Andrew Adamatzky · Leon Chua *Editors*

Memristor Networks

Springer

Memristor Networks

Andrew Adamatzky • Leon Chua
Editors

Memristor Networks

Springer

Editors

Andrew Adamatzky
Computer Science Department
University of the West of England
Bristol, UK

Leon Chua
Electrical Engineering and
Computer Sciences Department
University of California
Berkeley, CA, USA

ISBN 978-3-319-02629-9 ISBN 978-3-319-02630-5 (eBook)
DOI 10.1007/978-3-319-02630-5
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013957702

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

A memristor is a two-terminal device whose resistance depends on one or more internal state variables of the device. A memristor is defined by a state-dependent Ohm's law. Its resistance depends on the entire past signal waveform of the applied voltage, or current, across the memristor. Using memristors one can achieve circuit functionalities that it is not possible to establish with resistors, capacitors and inductors, therefore the memristor is of great pragmatic usefulness. Potential unique applications of memristors are in spintronic devices, ultra-dense information storage, neuromorphic circuits, and programmable electronics.

Despite explosive growth of memristor related findings there are few results on networks of memristive devices. Our book fills the gap. There are four main parts in the book yet their boundaries are eroded and their contents might sometimes overlap.

The first part deals with foundations of the memristor theory and applications. We start the book with a canonical text by Chua ("The Fourth Element"), where existence of memristor was predicted. This is followed by Williams's vivid vision of the impact the discovery of material memristors made on electronics and computer engineering ("Aftermath of Finding the Memristor"). In his chapter "Resistance Switching Memories are Memristors" Chua introduces fundamental circuit-theoretic concepts and properties of memristors, clarifying and demystifying common misconceptions on memristive devices. The ubiquitous nature of memristors is highlighted, from a historical perspective by Gandhi et al. ("The Detectors Used in the First Radios were Memristors"). They prove that first wireless radio detectors, or cat's whiskers, are amongst the first engineered memristors. The chapter "Memristor, Hodgkin-Huxley, and Edge of Chaos" by Chua demonstrates memristive properties of the Hodgkin-Huxley equation. Two critical elements of the Hodgkin-Huxley circuit model are a potassium ion channel memristor, and a sodium ion channel memristor, both of which are locally active. Adhikari and Kim help readers to differentiate between two devices, which are usually confused by non-initiated people: memristors and memistor.

The second part uncovers a rainbow of neuromorphic network architectures based on memristor assembles. The memristor with adaptive thresholds mimic higher-order behaviour of synapses, show Cai and Tetzlaff in "Synapse as a Memristor". Sheridan and Wei introduce material systems, anion based memristive devices, which are used in the design of learning networks with neuromorphic architectures ("Memristors and Memristive Devices for Neuromorphic Computing"). Memristive networks showing a long-term potentiation and spike-time-dependent plasticity, and associative memory, are discussed by Thomas and Kaltschmidt in their chapter "Bio-inspired neural networks". Chapter "Self-organization and Emergence of Dynamical Structures in Neuromorphic Atomic Switch Networks" by Stieg et al. introduces networks of atomic switches, inorganic synapse-like devices. The atomic switching networks are proved to be a uniquely scalable physical platform capable of exploring the dynamical interface of complexity, neuroscience, and engineering.

Serrano-Gotarredona et al., in "Spike-Timing-Dependent-Plasticity with Memristors" give a tutorial on the realisation of learning networks with memristors. They illustrate their approach on a visual cortex layer capable of orientation extraction. A memristor bridge synapses, based on a Wheatstone bridge-like circuit consisting of four identical memristors with specifically oriented polarities, are implemented by Kim et al. in "Memristor Bridge-based Artificial Neural Weighting Circuit". Networks of memristor bridge synapses are employed in image processing tasks. Nano-scale cellular-nonlinear network implemented from memristors in "Cellular Nonlinear Networks with memristor synapses" by Corinto et al., and utilised in image processing and multiplication tasks. Variable memristor networks are analysed and their efficiency in robot control is evaluated by Howard et al. in "Evolving Memristive Neural Networks". Computer experiments conclude that the variable memristor synapses bestow more behavioural degrees of freedom to the networks, allowing them to outperform the comparative synapse types.

The third part deals with dynamic behavior of memristive networks. Budhathoki et al. investigate the relationships among flux, charge and memristance of diverse composite memristors, using both linear and nonlinear memristor models, and analyze the characteristics of complex memristor circuits ("Behavior of Multiple Memristor Circuits"). In the chapter "A Memristor-Based Chaotic System with Boundary Conditions", Hu et al. incorporate charge-controlled and flux-controlled memristors into Chen oscillator and categorise non-trivial dynamical behaviour, including chaotic attractors. Gale et al. ("Spiking in Memristor Networks") show emergence of an oscillatory behaviour in groups of two and three memristors. Polyaniline (PANI) is an essential material of the organic memristive device. In his chapter "Organic memristive devices and neuromorphic circuits" Erokhin shows how to make organic memristors of PANI polymers and experimentally studies types of learning implementable in the networks of polymer-based organic memristors.

The fourth part is about computing with memristive networks. Kavehei et al., "Memristive in situ Computing", overview designs and computational potential of resistive random access memories, phase change memories and spin-transfer torque magnetoresistive memories. Designs of through-silicon via on chip stackable memristor arrays and their applications in neuromorphic circuits, current and temperature sensors are discussed in the chapter "Memory Effects in Multi-Terminal solid state devices and their Applications" by Sacchetto et al. Binary arithmetic is at the heart of all general purpose computing devices. Bickerstaff and Swartzlander, in "Memristor-Based Addition and Multiplication", provide an overview of analog and digital implementations of binary additions and multiplications. The schemes are exemplified on a ripple carry adder and an array multiplier. Circuit designs of memristors emulators for practical laboratory experiments are presented by Biolek in "Memristor emulators".

Vourkas and Sirakoulis, in "Modeling memristor–based circuit networks on crossbar architectures", explore dynamics of regularly connected networks of memristors; they also model memristors using quantum-mechanical phenomenon of tunnelling and test universal logic schemes. Computing potential of two- and three-dimensional memristive networks is discussed in chapters "Computing shortest paths in 2D and 3D memristive networks" by Ye et al., and "Computing Image and Motion with 3-D Memristive Grids" by Kai et al. Both chapters provide viable architectural designs and computer models which could be used in future material implementations of massively-parallel memristive processor at nano-, micro- and meso-scales. Flak, in "Solid-State Memcapacitors and Their Applications", introduces the concept of a memcapacitor, evaluates their physical implementations, and analyses a potential of memcapacitors for memory and logic applications. Stateful logical operations and synthesis of Boolean functions using the memristive stateful operations are outlined in chapter "Memristive Stateful Logic" by Lehtonen et al. A two-dimensional excitable medium with memristive diffusion links is imitated via Oregonator model and analysed by Asai in "Reaction-Diffusion Media with Excitable Oregonators Coupled by Memristors". A range of spatio-temporal behavioural scenarios is discovered, including emergence of non-uniform spatial patterns of excitation determined by initial conditions and memristor polarities. Pham et al., in "Autowaves in a lattice of memristor-based cells", study two-dimensional cellular non-linear networks, where every cell is equipped with memristors, and FPGA implementations of these networks. Generation and propagation of excitation waves is demonstrated. Iconic designs of memristive cellular automata are presented by Itoh and Chua, in "Memristor Cellular Automata and Memristor Discrete-Time Cellular Neural Networks". Cellular non-linear networks equipped with non-liner passive memristors show non-trivial behaviour, and perform logical and image processing operations.

The book is a unique self-contained compendium of results on memristor research developed by top world experts in the field. All aspects of memristor networks are presented in detail, in a fully accessible, often tutorial-like style. Mathematics, physics, engineering and computing of memristor devices are tackled in detail. The book is an indispensable source of information and an inspiring reference text for future generations of computer scientists, mathematicians, physicists and engineers.

Bristol, UK
Berkeley, USA
September 2013

Andy Adamatzky
Leon Chua

Contents

Contributors

Andrew Adamatzky University of the West of England, Bristol, UK

Shyam Prasad Adhikari Division of Electronics Engineering, Chonbuk National University, Jeonju, Jeonbuk, Republic of Korea; Intelligent Robots Research Center, Division of Electronics and Information Engineering, Chonbuk National University, Jeonju, Republic of Korea

Varun Aggarwal mLabs, New Delhi, India

Renato Aguilera Department of Chemistry and Biochemistry, University of California, Los Angeles, USA

Masakazu Aono WPI Center for Materials Nanoarchitectonics (MANA), National Institute for Materials Science (NIMS), Tsukuba, Japan

Tetsuya Asai Hokkaido University, Sapporo, Hokkaido, Japan

Alon Ascoli Technische Universität Dresden, Dresden, Germany

Audrius V. Avizienis Department of Chemistry and Biochemistry, University of California, Los Angeles, USA

K'Andrea Bickerstaff KenQuest LLC, Austin, TX, USA

Dalibor Biolek Dept. of Electrical Engineering/Microelectronics, University of Defence/Brno University of Technology, Brno, Czech Republic

Ram Kaji Budhathoki Intelligent Robots Research Center, Division of Electronics and Information Engineering, Chonbuk National University, Jeonju, Republic of Korea

Larry Bull University of the West of England, Bristol, UK

Arturo Buscarino Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania, Catania, Italy

Weiran Cai Fakultaet Elektrotechnik und Informationstechnik, Technische Universitaet Dresden, Dresden, Germany

Guanrong Chen Departments of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong SAR, China

Leon O. Chua Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA; University of California, Berkeley, USA; Technische Universität München, Munich, Germany

Fernando Corinto Politecnico di Torino, Torino, Italy

Ben De Lacy Costello University of the West of England, Bristol, UK

Giovanni De Micheli EPFL IC ISIM LSI1, Lausanne, Switzerland

Shukai Duan School of Electronics and Information Engineering, Southwest University, Chongqing, China

Victor Erokhin CNR-IMEM (National Council of the Researches—Institute of Materials for Electronic and Magnetism), Parma, Italy

Kamran Eshraghian iDataMap Corporation, Perth, Australia

Gang Feng Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Kowloon, Hong Kong SAR, China

Jacek Flak VTT Technical Research Centre of Finland, Espoo, Finland

Luigi Fortuna Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania, Catania, Italy

Mattia Frasca Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania, Catania, Italy

Pierre-Emmanuel Gaillardon EPFL IC ISIM LSI1, Lausanne, Switzerland

Ella Gale University of the West of England, Bristol, UK

Gaurav Gandhi mLabs, New Delhi, India

A. Gelencser Interdisciplinary Technical Sciences Doctoral School, Pázmány Péter Catholic University, Budapest, Hungary

James K. Gimzewski California NanoSystems Institute (CNSI), University of California, Los Angeles, USA; WPI Center for Materials Nanoarchitectonics (MANA), National Institute for Materials Science (NIMS), Tsukuba, Japan; Department of Chemistry and Biochemistry, University of California, Los Angeles, USA; School of Physics, Centre for Nanoscience and Quantum Information, University of Bristol, Bristol, UK

Gerard David Howard University of the West of England, Bristol, UK

Xiaofang Hu Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Kowloon, Hong Kong SAR, China

Makoto Itoh Fukuoka, Japan

Christian Kaltschmidt Biology Department, Bielefeld University, Bielefeld, Germany

Omid Kavehei Centre for Neural Engineering, Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Australia

Hyongsuk Kim Division of Electronics Engineering, Chonbuk National University, Jeonju, Jeonbuk, Republic of Korea; Intelligent Robots Research Center, Division of Electronics and Information Engineering, Chonbuk National University, Jeonju, Republic of Korea

Young-Su Kim School of Electrical Engineering, Kookmin University, Seoul, South Korea

Mika Laiho BID Technology, University of Turku, Turku, Finland

Yusuf Leblebici EPFL STI IEL LSM, Lausanne, Switzerland

Eero Lehtonen BID Technology, University of Turku, Turku, Finland

Chuan Kai Kenneth Lim Centre for Bio-inspired Technology, Department of Electrical and Electronic Engineering, Imperial College London, London, UK

B. Linares-Barranco Instituto de Microelectronica de Sevilla, Consejo Superior de Investigaciones Cientificas, Universidad de Sevilla, Sevilla, Spain

Wei Lu Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

Cristina Martin-Olmos California NanoSystems Institute (CNSI), University of California, Los Angeles, USA; Centre Suisse de Electronique et de Microtechnique SA (CSEM), Neuchatel, Switzerland

T. Masquelier Laboratory of Neurobiology and Adaptive Processses, University Pierre et Marie Curie, Paris, France

Kyeong-Sik Min School of Electrical Engineering, Kookmin University, Seoul, South Korea

Viet-Thanh Pham Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania, Catania, Italy; Hanoi University of Science and Technology, Hanoi, Viet Nam

Jonne K. Poikonen BID Technology, University of Turku, Turku, Finland

Jussi H. Poikonen Department of Communications and Networking, Aalto University, Espoo, Finland

Themistoklis Prodromakis School of Electronics and Computer Science, University of Southampton, Southampton, UK

Tamás Roska Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest, Hungary; Faculty of Information Technology, Pázmány P. Catholic University, Budapest, Hungary

Davide Sacchetto EPFL STI IEL LSM, Lausanne, Switzerland

Maheshwar P. Sah Intelligent Robots Research Center, Division of Electronics and Information Engineering, Chonbuk National University, Jeonju, Republic of Korea

Eric J. Sandouk Department of Chemistry and Biochemistry, University of California, Los Angeles, USA

T. Serrano-Gotarredona Instituto de Microelectronica de Sevilla, Consejo Superior de Investigaciones Cientificas, Universidad de Sevilla, Sevilla, Spain

Patrick Sheridan Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

Hsien-Hang Shieh California NanoSystems Institute (CNSI), University of California, Los Angeles, USA

Henry O. Sillin Department of Chemistry and Biochemistry, University of California, Los Angeles, USA

Georgios C. Sirakoulis Democritus University of Thrace, Xanthi, Greece

Efstratios Skafidas Centre for Neural Engineering, Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Australia; National ICT Australia (NICTA), Victoria Research Laboratory, Melbourne, VIC, Australia

R. Stanley Williams Hewlett-Packard Labs, Palo Alto, CA, USA

Adam Z. Stieg California NanoSystems Institute (CNSI), University of California, Los Angeles, USA; WPI Center for Materials Nanoarchitectonics (MANA), National Institute for Materials Science (NIMS), Tsukuba, Japan

Earl E. Swartzlander University of Texas at Austin, Austin, TX, USA

Ronald Tetzlaff Fakultaet Elektrotechnik und Informationstechnik, Technische Universitaet Dresden, Dresden, Germany

Andy Thomas Physics Department, Bielefeld University, Bielefeld, Germany; Physics Department, Osnabrück University, Osnabrück, Germany

Ioannis Vourkas Democritus University of Thrace, Xanthi, Greece

Shi Hong Marcus Wu Systems Engineering Advancement Research Initiative (SEAri), Massachusetts Institute of Technology, Cambridge, MA, USA

Changju Yang Intelligent Robots Research Center and Division of Electronics and Information Engineering, Chonbuk National University, Jeonju, Republic of Korea

Zhanyou Ye Centre for Bio-Inspired Technology, Department of Electrical and Electronic Engineering, Imperial College London, London, UK

The Fourth Element

Leon Chua

Abstract This tutorial clarifies the axiomatic definition of $(v^{(\alpha)}, i^{(\beta)})$ circuit elements via a look-up-table dubbed an, of admissible (v, i) signals measured via Gedanken Probing Circuits. The $(v^{(\alpha)}, i^{(\beta)})$ elements are ordered via a complexity metric. Under this metric, the *memristor* emerges naturally as the *fourth element* (Tour and He in Nature 453:42–43, 2008), characterized by a state-dependent Ohm's law. A logical generalization to memristive devices reveals a common *fingerprint* consisting of a dense continuum of *pinched hysteresis loops* whose area decreases with the frequency ω and tends to a straight line as $\omega \to \infty$, for all bipolar periodic signals and for all initial conditions. This common fingerprint suggests that the term memristor be used henceforth as a moniker for memristive devices.

1 Axiomatic Definition of Circuits Elements

How do you *characterize* a 2 terminal "black box" B such that its response to any electrical signal can be predicted? Since you are not allowed to peek inside B your only recourse is to carry out measurements by probing B with *all possible* electrical circuits, containing arbitrary interconnections of circuit elements, such as resistors, capacitors, inductors, diodes, transistors, op amps, batteries, voltage and current sources with arbitrary time functions, etc. We will henceforth call such circuits "*Gedanken Probing Circuits*", as depicted in the *Gedanken* Experimental Setup shown in Fig. 1. Let us insert an instrument called an ammeter in series with the top wire to record a time function $i(t)$ called the *current* in Amperes entering the top terminal (labeled by a plus $(+)$ sign). Next let us connect an instrument called a *voltmeter* across B to record a time function $v(t)$ called the *voltage* in Volts across

L. Chua (✉)
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA
e-mail: chua@eecs.berkeley.edu

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_1,

the *plus-minus* terminals of B.[1] Let us call $(v(t), i(t))$ an *admissible* (v, i) signal of B. The recorded list

$$B(v,i) \triangleq \{(v_1(t), i_1(t)), (v_2(t), i_2(t)), \ldots, (v_n(t), i_n(t)), \ldots\} \tag{1}$$

of *admissible* (v, i) *signals* (***AVIS***) from *all possible Gedanken Probing Circuits* constitutes the *complete* characterization of the 2-terminal black box B in the sense that given any voltage signal or current signal, one can search the ***AVIS*** "memory bank", henceforth called the ***AVIS**-pad* of B or just *A-pad*, and identify the unique admissible signals $(\tilde{v}(t), \tilde{i}(t))$ being sought. The ***A**-pad must* contain this entry in its memory bank because the signal is associated with *some* circuit connected to B, and this circuit is a Gedanken probing circuit, *by definition*. The ***A**-pad* is just a *look-up-table* containing *all* admissible (v, i) signals of B. Observe that the ***A**-pad* is in general an infinitely long pad containing infinitely-many pairs of admissible signal waveforms $(v(t), i(t))$ of B, as depicted in Fig. 1.

The above Gedanken experiment is only a thought experiment. However, for a large number of real-world 2-terminal devices, the ***A**-pad* for B can be generated via equations.

Example 1 (Ohm's Law) A very small subset of all 2-terminal black boxes are characterized by an ***A**-pad* that satisfies Ohm's Law; namely,

$$v = Ri \quad \text{or} \quad i = Gv \tag{2}$$

where R is called the *resistance* in Ohms (Ω) of B and G is called the *conductance* in Siemens (S) of B. In this case

$$\boldsymbol{AVIS} = \{(Ri_1(t), i_1(t)), (Ri_2(t), i_2(t)), \ldots, (Ri_n(t), i_n(t)), \ldots\} \tag{3}$$

can be reconstructed by (2). When Ohm's law is written with i as the independent variable, namely; $v = Ri$, it is called *current controlled.* If it is written in the form $i = Gv$, it is called *voltage controlled.* Often it is more convenient to recast (2) in the *implicit form*

$$f_R(v,i) = v - Ri = 0 \tag{4}$$

Since (4) is neither a function of v, nor of i, it is called a *relation* in mathematics. In *nonlinear circuit theory*, it is called a *constitutive relation* [2–4]. Observe that the constitutive relation is just a compact formula, or algorithm, for generating the ***A**-pad* of B.

[1] Observe that the voltage v and the current i are defined axiomatically via two instruments called voltmeter and ammeter, without invoking any physical concepts such as electric field, magnetic field, charge, flux linkages, etc. One does not even have to know how a voltmeter, or an ammeter, works. They are just names assigned to the instruments.

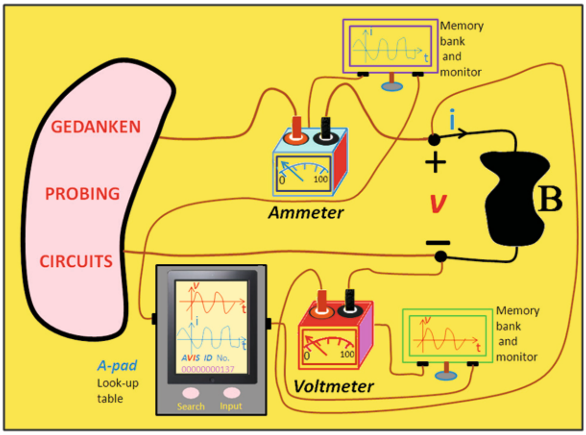


Fig. 1 Axiomatic definition of a 2-terminal circuit element

Example 2 Suppose the ***A****-pad* of the 2-terminal black box B in Fig. 1 can be written in the form

$$\boldsymbol{AVIS} = \left\{ \left(v_1, v_1 + \frac{1}{3} v_1^3 \right), \left(v_2, v_2 + \frac{1}{3} v_2^3 \right), \ldots, \left(v_n, v_n + \frac{1}{3} v_n^3 \right) \ldots \right\} \tag{5}$$

for all possible voltage signals

$$v(t) = v_1(t), v(t) = v_2(t), \ldots, v(t) = v_n(t) \ldots$$

then the *A-pad* of B can be generated by the much more compact constitutive relation

$$f_R(v, i) = v + \frac{1}{3} v^3 - i = 0 \tag{6}$$

Since both (4) of Example 1 and (6) of Example 2 involve the same pair of circuit variables (voltage, current), and since all 2-terminal devices that can be characterized by a constitutive relation

$$f_R(v, i) = 0 \tag{7}$$

between the variable pair (v, i) can be proved to be dissipative (or passive) if $v \times i > 0$ for all (v, i) listed in the ***A****-pad*, this class of 2-terminal elements are called *resistors* [2–4].

Example 3 Most 2-terminal black boxes can *not* be described by a constitutive relation between the variable pair (v, i). However, another important subclass can be expressed by a relationship between the variable pair (v, q), where

$$q(t) = \int_{-\infty}^{t} i(\tau) d\tau = q_0 + \int_{t_0}^{t} i(\tau) d\tau \tag{8}$$

and

$$q_0 \triangleq \int_{-\infty}^{t_0} i(\tau)d\tau \tag{9}$$

is called the initial state[2] of $q(t)$ at the initial time $t = t_0$. This subclass of 2-terminal black boxes can be characterized by a collection of admissible signals between the variable pair (v, q), namely,

$$B(v, q) = \big\{\big(v_1(t), q_1(t)\big), \big(v_2(t), q_2(t)\big), \ldots, \big(v_n(t), q_n(t)\big), \ldots\big\} \tag{10}$$

where,

$$q = Cv \tag{11}$$

and C is a constant called the *Capacitance* of B. Equation (11) is the constitutive relation of B because we can generate the corresponding ***AVIS*** $(v(t), i(t))$ via (8); namely

$$i(t) = \frac{dq(t)}{dt} \tag{12}$$

Indeed, any relationship

$$q = f_C(v) \tag{13}$$

is a valid constitutive relation and this class of 2-terminal devices are called *capacitors.*

By the same reasoning, the constitutive relation

$$\varphi = f_L(i) \tag{14}$$

involving the variable pair (i, φ) defines a third subclass of 2-terminal devices called *inductors*, where

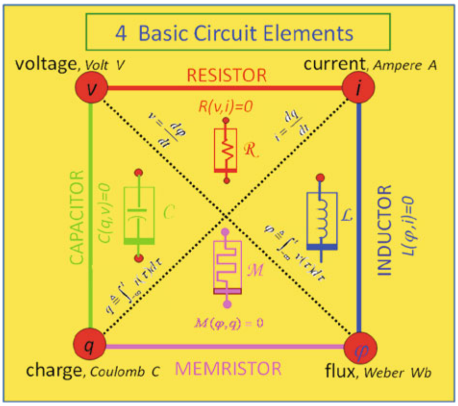
$$\varphi(t) = \int_{-\infty}^{t} v(\tau)d\tau = \varphi_0 + \int_{t_0}^{t} v(\tau)d\tau \tag{15}$$

Observe that the above 3 classes of basic circuit elements, called resistors, capacitors and inductors, are defined *axiomatically*, via a constitutive relation between a pair of variables chosen from $\{v, i, q, \varphi\}$. There are 6 different pairs that can be formed from these 4 variables; namely

$$\big\{(v, \varphi), (i, q), (v, i), (v, q), (i, \varphi), (\varphi, q)\big\} \tag{16}$$

[2] In practice one can never know the precise signal $i(t)$ over the infinite past. Rather we can only set up our measurements to begin at some initial time $t = t_0$. Consequently, the initial condition q_0 in Eq. (8) represents a summary of the past memory of $q(t)$ measured at $t = t_0$.

Fig. 2 Four axiomatically-defined Circuit Elements



The first 2 pairs (v, φ) and (i, q) are already related via (15) and (8), respectively, and are *not* constitutive relations because they can not predict the corresponding current $i(t)$ and voltage $v(t)$. However, the last pair (φ, q) defines yet another constitutive relation since given any *admissible* signals $(\varphi(t), q(t))$, one can recover the corresponding $(v(t), i(t))$ via (15) and (8). For logical consistency, and symmetry considerations, it is necessary to define a 4*th circuit element* [1] via the constitutive relation

$$f_M(\varphi, q) = 0 \tag{17}$$

between the variables φ and q. This element was postulated and named the *memristor* (acronym for *mem*ory *resistor* in [5]. A physical approximation of such an element has been fabricated in 2008 as a TiO_2 nano device by Dr. Stanley Williams group at hp [6]. The above axiomatic definition of the 4 basic circuit elements are summarized in Fig. 2, along with their respective symbols [7]. Note that the standard symbols for resistor, capacitor and inductor are enclosed by a thin rectangle with a dark band at the bottom because it is essential to distinguish the reference polarity of each nonlinear element if its constitutive relation is *not* odd-symmetric.

We wish to stress that although the symbols of q and φ in Fig. 2 are given the names *charge* and *flux*, respectively, *they need not* be associated with a real physical *charge* as in the case of a classical *capacitor* built by sandwiching a pair of parallel metal plates between an insulator, or a real physical *flux* as in the case of a classical *inductor* built by winding a copper wire around an iron core.

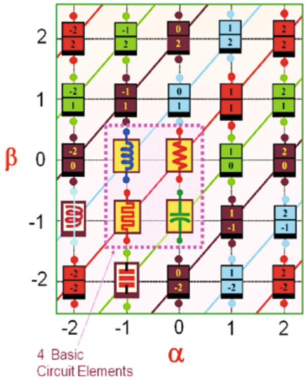


Fig. 3 The first 25 (α, β) circuit elements, $-2 \le \alpha \le 2$, $-2 \le \beta \le 2$

2 $(v^{(\alpha)} - i^{(\beta)})$ Circuit Elements

Let us introduce the notations [4]

$$\boxed{v^{(\alpha)}(t) \triangleq \begin{cases} \frac{d^\alpha v(t)}{dt^\alpha}, & \text{if } \alpha = 1, 2, \ldots, \infty \\ v(t), & \text{if } \alpha = 0 \\ \int_{-\infty}^{t} v(\tau)d\tau, & \text{if } \alpha = -1 \\ \int_{-\infty}^{t} \int_{-\infty}^{\tau_{|\alpha|}} \cdots \int_{-\infty}^{\tau_2} v(\tau_1)d\tau_1 d\tau_2 \cdots d\tau_{|\alpha|}, & \text{if } \alpha = -2, -3, \ldots, \infty \end{cases}} \tag{18}$$

and

$$\boxed{i^{(\beta)}(t) \triangleq \begin{cases} \frac{d^\beta i(t)}{dt^\beta}, & \text{if } \beta = 1, 2, \ldots, \infty \\ i(t), & \text{if } \beta = 0 \\ \int_{-\infty}^{t} i(\tau)d\tau, & \text{if } \ \beta = -1 \\ \int_{-\infty}^{t} \int_{-\infty}^{\tau_{|\beta|}} \cdots \int_{-\infty}^{\tau_2} i(\tau_1)d\tau_1 d\tau_2 \cdots d\tau_{|\beta|}, & \text{if } \beta = -2, -3, \ldots, \infty \end{cases}} \tag{19}$$

where $|\alpha|$ and $|\beta|$ are integers. Let us identify a $(v^{(0)}, i^{(0)})$ element as a *resistor*, a $(v^{(0)}, i^{(-1)})$ element as a *capacitor*, a $(v^{(-1)}, i^{(0)})$ element as an *inductor*, and a $(v^{(-1)}, i^{(-1)})$ element as a *memristor*. Using this notation, we can define an infinite family of circuit elements, each one identified by its element code $(v^{(\alpha)} - i^{(\beta)})$ and referred to simply as an (α, β) element.

The first 25 (α, β) elements are listed in Fig. 3, each coded by an integer pair (α, β), and identified by a rectangular box where "α" and "β" are printed on the "top", and at the "bottom" respectively. Each (α, β) element is located at the intersection between a vertical line through α, and a horizontal line through β. The four circuit element symbols shown in Fig. 2 are printed in their corresponding locations in Fig. 3. The 2 elements $(\alpha, \beta) = (-1, -2)$ and $(\alpha, \beta) = (-2, -1)$ are called *memcapacitor*, and *meminductor* respectively [11], and are identified by their corresponding symbols.

The above infinite family of circuit elements are defined *not* for the sake of generality. Rather, they are *essential* for developing a rigorous mathematical theory of nonlinear circuits in the sense that if one excludes all elements with $|\alpha| > k$ and $|\beta| > k$, for any finite integer k, then one can construct hypothetical circuits whose solutions do *not* exist after certain finite times $t \geq T_k$ due to the presence of a "singularity" called an *impasse point* [2, 3, 10]. It is unlikely however that (α, β) elements with $|\alpha| > 2$ and $|\beta| > 2$ will be needed in modeling most real world devices.

It can be proved that any (α, β) element with $|\alpha| + |\beta| > 2$ is *active* in the sense that it can be built only with active components, such as transistors and op amps, which requires a power supply. Finally, we remark that every (α, β) element can be built by the same procedure illustrated in [2, 5, 12] using a family of linear active 2-ports called *mutators*. They can also be *emulated* via various off-the-shelf digital components [13], or by programmable softwares interfaced with analog-to-digital (A/D) and digital-to-analog (D/A) converters.

3 Complexity Metric of Circuit Elements

For each (α, β) element, let

$$\chi \triangleq |\alpha| + |\beta| \tag{20}$$

be its associated *complexity metric* [9]. For example $\chi(0, 0) = 0$ for a *resistor*, $\chi(0, -1) = 1$ for a *capacitor*, $\chi(-1, 0) = 1$ for an *inductor*, $\chi(-1, -1) = 2$ for a *memristor*, $\chi(-1, -2) = 3$ for a *memcapacitor* and $\chi(-2, -1) = 3$ for a *meminductor*. If one associates the vertical and horizontal lines passing through the elements in Fig. 3 as streets of Manhattan, New York city, then the complexity metric χ of an (α, β) element gives a measure of its distance from the resistor $(\alpha, \beta) = (0, 0)$. The larger the metric $\chi(\alpha, \beta)$, the farther it is from the resistor.

The complexity metric measures not just the distance of (α, β) element from the resistor, but also the *minimum number* of capacitors (or inductors) needed to build an (α, β) element using off-the-shelf components. For example, a minimum of one capacitor along with active elements such as transistors and op amps, are needed to build a *memristor* while a minimum of two capacitors are needed to build a meminductor. From a mathematical perspective, the larger the complexity metric, the higher the dimension of the *state space* and the larger the number of nonlinear differential equations and exotic dynamical phenomena that can emerge.

Based on any of the above measures of complexity, the 4 elements depicted in Fig. 3 are indeed the simplest circuit elements, with the memristor ranked as the 4th element in increasing complexity.

4 Fingerprint of Memristors

The formal mathematical definition of the memristor is given in [5], along with its circuit-theoretic properties. Here we recall that the memristor is defined by a collection of all admissible signals, namely, an *A-pad* listing all signals measured from all admissible “Gedanken Probing Circuits” (Fig. 1) and which can be completely reproduced by the constitutive relation (17).

For example, a charge-controlled memristor can be defined by

$$\varphi = f_M(q) \tag{21}$$

where f_M is a *piecewise-differentiable* function [9]. In this case, we can generate all $(v(t), i(t))$ from the *A-pad* via the following q-dependent Ohm’s law:

$$\boxed{\begin{aligned} & v = R(q)i && (22a) \\ & R(q) \triangleq \frac{df_M(q)}{dq} && (22b) \end{aligned}}$$

The function $R(q)$ is called the memristance (acronym for Memory Resistance) where

$$R(q) \geq 0 \tag{23}$$

for all passive *memristors* [2].

Now observe from (8) that since

$$\frac{dq}{dt} = 0 \quad \text{when } i = 0 \tag{24}$$

the memristor can assume a *continuous* range of distinct equilibrium states

$$q = q(t_0), \quad t \geq t_0 \tag{25}$$

when the power is switched off at any time $t = t_0$. It follows that the *memristor* can be used as a *non-volatile analog memory*. In particular, it can be used as a non-volatile *binary* memory where two sufficiently different values of resistance are chosen to code the binary states “0” and “1”, respectively. Because the hp *memristor* reported in [6] as well as in many other nano devices [14] can be scaled down to atomic dimensions, the *memristor* offers immense potentials for an ultra low-power and ultra dense non-volatile memory technology that could replace flash memories and DRAMS.

An incisive analysis of (22a), (22b) reveals that the *non-volatile* memory property possessed by the memristor is a direct consequence of its *state-dependent* Ohm’s law. Moreover, all circuit-theoretic properties possessed by the *memristor* are preserved if we generalize Eqs. (22a), (22b) to the form [8].

$$\boxed{\begin{aligned} v &= R(x,i)i \\ dx/dt &= \mathbf{f}(x,i) \end{aligned}} \qquad \begin{aligned} &(26a) \\ &(26b) \end{aligned}$$

The generalized *memristor* defined in (26a), (26b) is dubbed a memristive *device* in [8] where $x = (x_1, x_2, \ldots, x_n)$ denotes *n states variables*, which do not depend on any external voltages or currents. However, since both (22a), (22b) and (26a), (26b) are endowed with the same circuit-theoretic properties, it is more convenient and logical to refer to both equations as defining a *memristor*. In the rare events where a distinction may be desirable, one can refer to (22a), (22b) as defining an "ideal memristor".

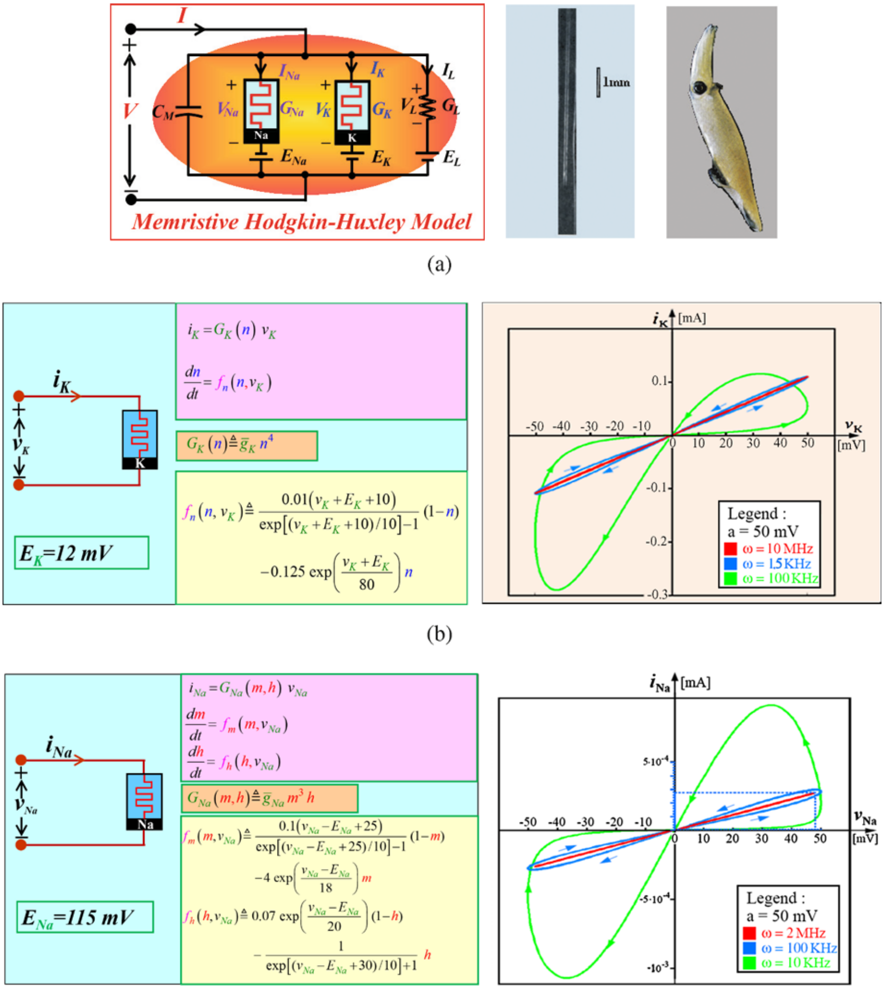
The most important common property of (22a), (22b) and (26a), (26b) is that the loci (i.e. Lissajous figure) of $(v(t), i(t))$ due to *any* periodic current source, or periodic voltage source, which assumes both positive and negative values, must always be *pinched* at the origin in the sense that $(v, i) = (0, 0)$ must always lie on the (v, i)-loci, called a *pinched hysteresis loop* in the literature [14]. We wish to stress that (22a), (22b) and (26a), (26b) imply that the pinched hysteresis loop phenomenon of the memristor must hold for *any periodic signal*, $v(t)$ or $i(t)$, that assumes both positive and negative values, as well as for any initial condition used to integrate the differential equations to obtain the corresponding steady state $i(t)$ and $v(t)$, respectively.

Another unique property shared by all memristor hysteresis loops is that for every given periodic function $i = f(t)$ (where $f(\bullet)$ assumes both positive and negative values), and for any initial state $\mathbf{x}(0)$ the area enclosed within the part of the pinched hysteresis loop in the first quadrant, and the third quadrant, of the v–i plane shrinks continuously as the frequency ω increases, and the hysteresis loop tends to a *single-valued function* through the origin as ω tends to ∞.

The above dense continuum of pinched hysteresis loops, as well as their *single-valued function* limiting phenomenon as $\omega \to \infty$ must hold for *all memristors*. Any purported system which may exhibit a pinched hysteresis loop but which violates the above continuum and frequency-dependent limiting *memristor fingerprint* is *not* a memristor, the reader is referred to [15] for several contrived examples which fails the above "*memristor fingerprint* test".

We end this tutorial by pointing out that not all memristors are non-volatile memories. In fact there is an even *larger class* of *locally-active* memristors [2, 4, 10] which exhibit many exotic nonlinear dynamical phenomena. A very interesting and scientifically significant example is the classic Hodgkin-Huxley Axon circuit model of the squid giant axon.[3] Notwithstanding the immense importance of their

[3] Hodgkin and Huxley were awarded the 1965 Nobel Prize in physiology for their derivation of the circuit shown in Fig. 4a, where the 2 memristors were drawn as time-varying resistors in Fig. 1 (p. 501) of [16].



I
V
C_M
I_{Na}
V_{Na}
G_{Na}
Na
E_{Na}
I_K
V_K
G_K
K
E_K
I_L
V_L
G_L
E_L
Memristive Hodgkin-Huxley Model
1mm
$i_K = G_K(n)\, v_K$
$\frac{dn}{dt} = f_n(n, v_K)$
$G_K(n) \triangleq \bar{g}_K n^4$
$f_n(n, v_K) \triangleq \frac{0.01(v_K + E_K + 10)}{\exp[(v_K + E_K + 10)/10] - 1}(1-n) - 0.125 \exp\left(\frac{v_K + E_K}{80}\right) n$
i_K
v_K
K
E_K=12 mV
i_K [mA]
0.1
-0.1
-0.2
-0.3
-50 -40 -30 -20 -10
0 10 20 30 40 50
v_K [mV]
Legend :
a = 50 mV
🟥 ω = 10 MHz
🟦 ω = 1.5 KHz
🟩 ω = 100 KHz
$i_{Na} = G_{Na}(m, h)\, v_{Na}$
$\frac{dm}{dt} = f_m(m, v_{Na})$
$\frac{dh}{dt} = f_h(h, v_{Na})$
$G_{Na}(m, h) \triangleq \bar{g}_{Na} m^3 h$
$f_m(m, v_{Na}) \triangleq \frac{0.1(v_{Na} - E_{Na} + 25)}{\exp[(v_{Na} - E_{Na} + 25)/10] - 1}(1-m) - 4 \exp\left(\frac{v_{Na} - E_{Na}}{18}\right) m$
$f_h(h, v_{Na}) \triangleq 0.07 \exp\left(\frac{v_{Na} - E_{Na}}{20}\right)(1-h) - \frac{1}{\exp[(v_{Na} - E_{Na} + 30)/10] + 1} h$
i_{Na}
v_{Na}
Na
E_{Na}=115 mV
i_{Na} [mA]
$5 \cdot 10^{-4}$
$-5 \cdot 10^{-4}$
-10^{-3}
-50 -40 -30 -20 -10
0 10 20 30 40 50
v_{Na} [mV]
Legend :
a = 50 mV
🟥 ω = 2 MHz
🟦 ω = 100 KHz
🟩 ω = 10 KHz

(a)

(b)

(c)

Fig. 4 Hodgkin-Huxley Axon. (**a**) Memristive Hodgkin-Huxley Circuit model of giant axon (*center*) of North Atlantic squid Loligo (*right*). (**b**) Potassium ion-channel memristor and its pinched hysteresis loops (**c**) Sodium ion-channel memristor and its pinched hysteresis loops [39]

circuit model, Hodgkin and Huxley had erroneously named 2 circuit elements in their model associated with the potassium ion, and the sodium ion, respectively, as *time-varying conductances*. This mistaken identity has led to numerous confusions and paradoxes ever since the publications of their classic axon circuit model [16]. Well-known physiologists were puzzled by experimentally observed *rectification phenomenon* as well as *gigantic inductances* that could not exist within the soft tissues of the brain. The following quotation from Cole (see p. 78 of [18]), an eminent physiologist and the recipient of the 1967 USA *National Medal of Science*, is a case in point:

> *"The suggestion of an inductive reactance anywhere in the system was shocking to the point of being unbelievable".*

We have solved the above conundrum, and many other hitherto unresolved paradoxes associated with the Hodgkin-Huxley Axon, by showing the Hodgkin-Huxley *time-varying* potassium conductance is in fact a 1st-order memristor, and the Hodgkin-Huxley *time-varying* sodium conductance is in fact a 2nd-order memristor, as defined in Figs. 4b and 4c, respectively [39]. Also depicted in Fig. 4 are the pinched hysteresis loops associated with each memristor. Observe that they are all pinched at the origin, and that the lobe area in the first and third quadrants shrinks continuously to a straight line as ω increases, both being the fingerprint of memristors.

We conclude this tutorial by stressing that memristors are not inventions. They are *discoveries* and are ubiquitous. Indeed, many devices, including the "*electric arc*" dating back to 1801, have now been identified as memristors [19, 38]. Aside from serving as non-volatile memories [20], *locally-passive* memristors, have been used for switching electromagnetic devices [21], for field programmable logic arrays [22–26], for synaptic memories [27–29], for learning [30–32], etc.

In addition, *locally-active* memristors have been found to exhibit many exotic dynamical phenomena, such as *oscillations* [33], *chaos* [34, 35], *Hamiltonian vortices* [36] and *autowaves* [37], etc.

5 Concluding Remarks

Any 2-terminal device which exhibits a pinched hysteresis loop in the v-i plane when driven by *any* bipolar periodic voltage or current waveform, for *any* initial conditions, is a *memristor*. In the case where the *memristance* $R(x_1, x_2, \ldots, x_n)$ does not depend on the current i, the loop shrinks to a straight line whose slope depends on the excitation waveform, as the excitation frequency tends to infinity.

Except in ideal cases, memristors, memcapacitors, and meminductors do *not* behave like resistors, capacitors, and inductors, respectively. For example, the potassium and sodium ion channel memristors in the Hodgkin-Huxley axon circuit model behave like R-L circuits [17, 39]. It is conceptually wrong and misleading to identify memristors, memcapacitors, and meminductors with resistors, capacitors, and inductors. Each (α, β) element is a distinct circuit element because it can not be built from the other elements.

Readers who may have been mislead by some erroneous commentary in the popular press which associates an earlier *gadget* called a *memistor* with the *memristor* are referred to a technical clarification in [15, 40].

We end this tutorial with the following succinct signature of a *memristor* [14]:

> *If it's pinched it's a memristor.*

References

1. Tour, J.M., He, T.: The fourth element. Nature **453**, 42–43 (2008)
2. Chua, L.O.: Introduction to Nonlinear Network Theory. McGraw-Hill, New York (1969)
3. Chua, L.O.: Device modeling via basic nonlinear circuit elements. IEEE Trans. Circuits Syst. **CAS-27**, 1014–1044 (1980)
4. Chua, L.O.: Nonlinear circuit foundations for nano devices, Part I: the four-element torus. Proc. IEEE **91**, 1830–1859 (2003)
5. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **CT-18**, 507–519 (1971)
6. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
7. Chua, L.O.: Introduction to memristors. IEEE Expert Now Educational Course (2009)
8. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**, 209–223 (1976)
9. Bartle, R.G.: The Elements of Real Analysis, 2nd edn. Wiley, New York (1976)
10. Chua, L.O.: Dynamic nonlinear networks: state of the art. IEEE Trans. Circuits Syst. **CAS-27**, 1059–1087 (1980)
11. Di Ventra, M., Pershin, Y.V., Chua, L.O.: Circuit elements with memory: memristors, memcapacitors, and meminductors. Proc. IEEE **97**, 1717–1723 (2009)
12. Biolek, D., Bioleova, V.: Mutator for transforming memristor into memcapacitor. Electron. Lett. **46**, 1428–1429 (2010)
13. Pershin, Y.V., Di Ventra, M.: Teaching memory circuit elements via experiment-based learning. arXiv:1112.5427v1 [physics.ins-det]
14. Chua, L.O.: Resistance switching memories are memristors. Appl. Phys. A **102**, 765–783 (2011)
15. Kim, H., Sah, M.P., Adhikari, S.P.: Pinched hysteresis loop is the fingerprint of memristive devices. arXiv:1202.2437v1 [cond-mat.mes-hall]
16. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to the conduction and excitation in nerve. J. Physiol. **117**, 500–544 (1952)
17. Chua, L., Sbitnev, V., Kim, H.: Neurons are poised near the edge of chaos. Int. J. Bifurc. Chaos **22**(4), 1250098 (2012)
18. Cole, K.S.: Membranes, Ions and Impulse. University of California Press, Berkeley (1972)
19. Ayrton, H.: The Electric Arc. Van Nostrand, London (1902)
20. Valov, I., Waser, R., Jameson, J.R., Kozicki, M.N.: Electrochemical metallization memories-fundamentals, applications, prospects. Nanotechnology **2**, 254003 (2011)
21. Bray, M.G., Werner, D.H.: Passive switching of electromagnetic devices with memristors. Appl. Phys. Lett. **96**, 0735041 (2010)
22. Borghetti, J., Snider, G.S., Kukes, P.J., Yang, J.J., Stewart, D.R.: Williams memristive' switches enable 'stateful' logic operations via material implication. Nature **464**, 873–876 (2010)
23. Strukov, D.B., Williams, R.S.: Four-dimensional address topology for circuits with stacked multilayer crossbar arrays. Proc. Natl. Acad. Sci. USA **106**, 20155–20158 (2009)
24. Lehtonen, W., Laiho, M.: Stateful implication logic with memristors. In: Proc. of IEEE/ACM Int. Symposium on Architectures, pp. 33–36 (2009)
25. Borghetti, J., Li, Z., Straznicky, X., Li, X., Ohlberg, A., Wu, W., Stewart, D.R., Williams, R.S.: A hybrid nanomemristor/transistor logic circuit capable of self-programming. Proc. Natl. Acad. Sci. USA **106**, 1699–1703 (2009)
26. Kim, K., Shin, S., Kang, S.-M.: Field programmable stateful logic array. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **30**, 1800–1813 (2011)
27. Linares-Barranco, B., Serrano-Gotarredona, T.: Memristance can explain spike-time-dependent-plasticity in neural synapses. Nature Precedings. hdl:10101/npre. 2009.3010.1: 31 Mar. 2009
28. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. **10**, 1297–1301 (2010)

29. Pershin, Y.V., Di Ventra, M.: Neuromorphic, digital and quantum computation with memory circuit elements. arXiv:1009.6025v3 [cond-mat.mes-hall]
30. Snider, G.S.: Spike-timing dependent learning in memristive nanodevices. In: IEEE/ACM Int. Symposium on Nanoscale Architecture, pp. 85–92 (2008)
31. Liu, T., Kang, Y., Verma, M., Orlowski, M.: Novel highly nonlinear memristive circuit elements for neural networks. In: Proceedings, 2012 IJCNN International Joint Conference on Neural Networks, Brisbane, Australia, June 2012. doi:10.1109/IJCNN.2012.6252460
32. Chang, T., Jo, S.-H., Lu, W.: Short-term memory to long-term memory transition in a nanoscale memristor. Am. Chem. Soc. (ACS) Nano **5**, 7669–7676 (2011)
33. Itoh, M., Chua, L.O.: Memristor oscillators. Int. J. Bifurc. Chaos **18**, 3183–3206 (2008)
34. Muthuswamy, B., Chua, L.O.: Simplest chaotic circuit. Int. J. Bifurc. Chaos **20**, 1567–1580 (2010)
35. Ginoux, J.-M., Letellier, C., Chua, L.O.: Topological analysis of chaotic solution of a three-element memristive circuit. Int. J. Bifurc. Chaos **20**, 3819–3827 (2010)
36. Itoh, M., Chua, L.O.: Memristor Hamiltonian circuits. Int. J. Bifurc. Chaos **21**, 2395–2425 (2011)
37. Pham, V.-T., Buscarino, A., Frasca, M., Fortuna, L.: Autowaves in memristive cellular neural networks. Int. J. Bifurc. Chaos **22**, 12300271 (2012). 9 pp.
38. Prodromakis, T., Toumazou, C., Chua, L.: Two centuries of memristors. Nat. Mater. **11**, 478–481 (2012)
39. Chua, L., Sbitnev, V., Kim, H.: Hodgkin Huxley axon is made of memristors. Int. J. Bifurc. Chaos **22**(3), 1230011 (2012)
40. Kim, H., Adhikari, S.P.: Memistor is not memristor. IEEE Circuits Syst. Mag., First Quart. 75–78 (2012)

Aftermath of Finding the Memristor

R. Stanley Williams

Abstract In this paper, a personal guide to the set of Leon Chua's papers that I have found most helpful in my research will be provided in the hope that this will encourage others to study them and answer the questions they have. Then, the paper will be finished with some observations and comments about Prof. Leon Chua's definition of memristor in mathematics.

Observing the response to our paper "The missing memristor found" [1] over the past four years in both the popular press and the scientific literature has been fascinating. A significant part of the scientific process is to vet descriptions of new ideas or objects, and the bigger the potential impact of a concept, the more rigorous that scrutiny should be. However, intertwined with this process are many human issues of desire for recognition and priority of discovery, as well as an often strong bias to reject anything new without actually understanding it. There are a lot of misconceptions about memristors floating around that are difficult to correct with only a few explanatory pages. Real understanding requires a great deal of hard work, and the resources essential to achieve that understanding already exist in the literature. However, for the vast majority of us, skimming over a few papers is completely insufficient; I spent years reading and re-reading Prof. Leon Chua's papers before I started to really get an appreciation of what he was saying. I have several copies of many different papers completely covered with highlighter of many colors and with my scrawled notes—each time I read one of his paper, and I continually refer back to them, I learn something new and my appreciation deepens. Although he has written some wonderful tutorials, most of Prof. Chua's writings are formal and dense with information, and thus can be intimidating; they absolutely require a level of mathematical sophistication to comprehend, but to those who persevere, they are marvels of rigor and, eventually, clarity. Here I will provide a personal guide to the set of papers that I have found most helpful in my research in the hope that this will encourage others to study them and answer the questions they have, and then I will finish with some observations and comments.

R. Stanley Williams (✉)
Hewlett-Packard Labs, 1501 Page Mill Road, Palo Alto, CA 94304, USA
e-mail: stan.williams@hp.com

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_2,

During the 1960's, Prof. Chua established the mathematical foundations for nonlinear circuit theory, which was the basis for his classic 1969 textbook *Introduction to Nonlinear Network Theory* as well as a large number of papers in refereed journals. As a result of this work, Prof. Chua made an interesting observation that led to his discovery of the memristor as a mathematical entity, reported in 1971 [2]. For completely linear circuits (which is highly restrictive, since real physical systems will display nonlinearity beyond some operating range), there are only three independent two-terminal passive circuit elements: the resistor R, the capacitor C and the inductor L. However, when he generalized the mathematical relations to be nonlinear, there was another independent differential relationship that in principle coupled the charge q that flowed through a circuit and the flux ϕ in the circuit, $d\phi = M\,dq$, that was mathematically different from the nonlinear resistance that coupled the voltage v to the current i, $dv = R\,di$. He mathematically explored the properties of this new model nonlinear circuit element, and found that it was essentially a resistor with memory—it was a device that changed its resistance depending on the amount of charge that flowed through the device, and thus he called this hypothetical circuit element M a memristor. This conclusion was independent of any physical mechanism that might couple the flux and charge, and none was postulated. Moreover, the memristor definition did not require causality. In other words, the mathematical relationship between flux and charge could be the result of some other cause—any mechanism that led to the constraint embodied by the equation $d\phi = M\,dq$ would lead to a device with the circuit properties of a memristor. This prediction of the properties of a new circuit element from symmetry principles was totally unique and revolutionary, and did not depend on any experimental observation. He published these initial findings essentially as a curiosity—it was not obvious at that time that a physical analog of such a circuit element existed, and thus he called it the "missing element".

In 1976, with his then student Sung Mo Kang, he published a critical generalization of the original memristor concept [3], but this has not been cited with the frequency of the 1971 paper, so fewer people seem to be aware of it. Chua and Kang introduced the fact that a 'memristive device' has a state variable (or variables), indicated by w, that describes the physical properties of the device at any time. A memristive system is characterized by two equations, the 'quasi-static' conduction equation that relates the voltage across the device to the current through it at any particular time via a generalized resistance,

$$v = R(w, i)i$$

and the dynamical equation, which explicitly asserts that the derivative of the state variable w is a function f of itself and the current through the device,

$$dw/dt = f(w, i).$$

Neither the flux ϕ nor the charge q explicitly appears in either of these two equations, but if $w = q$, $R(w, i) = R(w)$ and $f(w, i) = i$, the two equations reduce to the original definition of a memristor. Furthermore, the quasi-static conduction equation places a requirement on the current-voltage characteristic of the device—if a memristive system is driven with some type of cyclic excitation, such as a sinusoidal current, the plot of the voltage vs. the current will be a Lissajous curve for which the voltage is always equal to zero when the current is zero, and vice versa. Chua called this curve a 'pinched hysteresis loop', and it has an important physical interpretation—neither a memristor nor a memristive system stores either charge or energy (like a capacitor, for example), but they do 'remember' their history because of their changing resistance. This 1976 paper showed many other properties of the generalized memristor and also discussed possible examples, but again this was a mathematical exercise that was independent of any physical mechanism known at the time. The importance to real systems is that if one can identify the state variable with a physical property of a device and experimentally determine the dynamical and quasistatic equations, then one has a useful model for the element that can be used for designing a circuit that would utilize the device.

There is another pair of papers that are critical for not only understanding how memristors stand as independent devices, but how to appropriately understand a nonlinear circuit element model, how to construct one from a physics-based mechanism or black-box electrical measurements, and how the model relates to an actual nonlinear circuit [4, 5]. The two papers are best read together; the 1980 paper [4] is mathematically thorough, broad in coverage and filled with deep insights, whereas the 1984 paper is more tutorial and descriptive. I often find myself going back and forth between the two for the complementary viewpoints they express. We learn that no circuit model is an exact equivalent because no physical device can be exactly mimicked by a mathematical equation. A particular physical device may be best described by different models depending on the operating range, with the 'best' model being the simplest one that produces realistic results. There are several properties that a realistic model should have, including well-posedness (no mathematical artifacts that cause an unphysical situation), the capability to be simulated, qualitative similarity to the physical system (e.g. same initial and asymptotic behavior), the ability to predict previously unexplored operating modes, and structural resilience (stability under small perturbations of the model parameters). These concepts are made clear through mathematical definitions and examples. Thus, one needs an appropriate set of models (I think of them as basis functions) that are as complete as possible to describe a real system. Creating a device model is an art that can utilize a wide range of inputs and insights—there is no unique way to define the best possible model; was it useful in enabling a circuit to be designed and did it predict the properties of the circuit to within some desired accuracy? If there is a physical device for which the properties are well described by a particular model, then we can call that device by the name of the model, understanding that a more complete description may require some attribute of a different model. For example, all physical inductors have an intrinsic resistance, which is usually described as a model resistance in series with a model inductance.

The final two papers are both tutorials and are written in a much more informal style [6, 7]. They are very useful for people who just want to get a light overview of memristors without digging into the mathematical details, but also contain significant new insights and are therefore valuable even for people who have mastered the first four papers. However, no one should read only [6, 7] and think that they comprehend the subject—any word description can be misconstrued or misrepresented; the actual definitions are all mathematical. In his 2002 publication [6], Prof. Chua correctly realized that as electronic device dimensions shrink into the nanometer scale, their properties will become more nonlinear and thus the issues for understanding nonlinear circuits are becoming increasingly more relevant and critical. He used a fascinating analogy working up from the 'Laws' of motion postulated by Aristotle, Newton and Einstein to illustrate the necessity of choosing the right variables for a model in the first place and then what happens when the model progresses from an initial linear approximation to a more realistic nonlinear formulation. There follows several completely worked out examples to illustrate nonlinear circuit element modeling from his previous papers, including memristors. In the final paper [7], he describes memristors and memristive systems, and makes the observation that in fact the latter are a relatively straightforward generalization of the former, and recommends that from now on to simplify the nomenclature that both should be called memristors. By creating memristor models for the pinched i–v hysteresis loops of each, he shows that specific physical examples of memristors include several devices that are the subject of contemporary research: bipolar and unipolar resistive switches, often called RRAM or ReRAM; 'atomic switches'; spin-transfer torque (STT) RAM devices; and phase-change memory devices; which are based on a wide variety of materials and physical mechanisms [7].

An important issue to understand is that the discovery of the memristor mathematical model does not conflict with nor compete for priority against the various realizations of physical devices that exhibit this circuit behavior. It is complementary, in that it provides a mathematical framework for designing and actually using the devices in circuits. It also provides an important mathematical constraint for those who are interested in the physics of their devices—any mechanism proposed for how the device operates needs to be in agreement with the memristor equations or it is not valid. Thus, researchers who are working on various types of resistive switching devices need not fear the memristor, but rather should embrace it. It is a high-level mathematical model that can be used to predict the circuit behavior of a wide variety of physical devices, it provides a unifying framework to put the circuit properties of all the devices into context, and therefore provides insight into how the various devices may substitute for each other in a wide variety of (especially nonmemory) applications originally developed for a different device.

Another issue is that no matter how careful one tries to be, any word description of a mathematical model will likely be incomplete, just as the model itself is only an approximation of the properties of a physical system. Thus, papers or discussions that argue about the meaning of a particular word or phrase often miss the point, since words can be ambiguous and interpreted (or twisted) in different ways. That is why we use mathematics in science—when a question arises about the specific meaning of a concept, we must go back to the defining equations. This is where Prof. Leon Chua's work stands out—precise, complete, insightful and totally rigorous.

References

1. Strukov, D., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80 (2008)
2. Chua, L.O.: Memristor—missing circuit element. IEEE Trans. Circuit Theory **CT-18**(5), 507–519 (1971)
3. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**, 209–223 (1976)
4. Chua, L.O.: Device modeling via basic nonlinear circuit elements. IEEE Trans. Circuits Syst. **CAS-27**, 1014–1044 (1980)
5. Chua, L.O.: Nonlinear circuits. IEEE Trans. Circuits Syst. **CAS-31**, 69–87 (1984)
6. Chua, L.O.: Nonlinear circuit foundations for nanodevices, Part I: the four-element torus. Proc. IEEE **91** (2003)
7. Chua, L.: Resistance switching memories are memristors. Appl. Phys. A **102**, 765–783 (2011)

Resistance Switching Memories Are Memristors

Leon Chua

Abstract All 2-terminal non-volatile memory devices based on *resistance switching* are *memristors*, regardless of the device material and physical operating mechanisms. They all exhibit a distinctive "fingerprint" characterized by a *pinched hysteresis loop* confined to the first and the third quadrants of the v–i plane whose contour shape in general changes with both the amplitude and frequency of any periodic "sine-wave-like" input voltage source, or current source. In particular, the pinched hysteresis loop shrinks and tends to a straight line as frequency increases. Though numerous examples of voltage vs. current pinched hysteresis loops have been published in many unrelated fields, such as biology, chemistry, physics, etc., and observed from many unrelated phenomena, such as gas discharge arcs, mercury lamps, power conversion devices, earthquake conductance variations, etc., we restrict our examples in this *tutorial* to solid state and/or nano devices where copious examples of published pinched hysteresis loops abound. In particular, we sampled arbitrarily, one example from each year between the years 2000 and 2010, to demonstrate that the memristor is a device that does not depend on any particular material, or physical mechanism. For example, we have shown that *spin-transfer magnetic tunnel junctions* are examples of memristors. We have also demonstrated that both *bipolar* and *unipolar* resistance switching devices are memristors. The goal of this *tutorial* is to introduce some fundamental circuit-theoretic concepts and properties of the memristor that are relevant to the analysis and design of *non-volatile* nano memories where binary bits are stored as resistances manifested by the memristor's continuum of equilibrium states. Simple pedagogical examples will be used to illustrate, clarify, and demystify various misconceptions among the uninitiated.

1 Pinched Hysteresis Loops

The *memristor* [1] is a 2-terminal circuit element characterized by a *constitutive relation* between two mathematical variables q and φ representing the time integral

L. Chua (✉)
Department of EECS, University of California, Berkeley, CA 94720, USA
e-mail: chua@eecs.berkeley.edu

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_3,

of the element's current $i(t)$, and voltage $v(t)$; namely,

$$q(t) \triangleq \int_{-\infty}^{t} i(\tau)d\tau \tag{1}$$

$$\varphi(t) \triangleq \int_{-\infty}^{t} v(\tau)d\tau \tag{2}$$

It is important to stress that "q" and "φ" are defined mathematically and need not have any physical interpretations. Nevertheless, we call q the *charge* and φ the *flux* of the memristor since Eqs. (1) and (2) coincide with the formula relating charge to current, and flux to voltage, respectively. We say the memristor is *charge-controlled*, or *flux-controlled,* if its constitutive relation can be expressed by

$$\varphi = \hat{\varphi}(q) \tag{3}$$

or

$$q = \hat{q}(\varphi) \tag{4}$$

respectively, where $\hat{\varphi}(q)$ and $\hat{q}(\varphi)$ are *continuous* and *piecewise-differentiable* functions[1] with *bounded slopes*.

Differentiating Eqs. (3) and (4) with respect to time t, we obtain

$$v = \frac{d\varphi}{dt} = \frac{d\hat{\varphi}(q)}{dq}\frac{dq}{dt} = R(q)i \tag{5}$$

where,

$$R(q) \triangleq \frac{d\hat{\varphi}(q)}{dq} \tag{6}$$

is called the *memristance*[2] at q, and has the unit of Ohms (Ω), and

$$i = \frac{dq}{dt} = \frac{d\hat{q}(\varphi)}{d\varphi}\frac{d\varphi}{dt} = G(\varphi)v \tag{7}$$

where

$$G(\varphi) \triangleq \frac{d\hat{q}(\varphi)}{d\varphi} \tag{8}$$

is called the *memductance* at φ, and has the unit of Siemens (S). Observe that Eqs. (5) and (6) can be interpreted as *Ohm's law* except that the resistance $R(q)$ at any time $t = t_0$ depends on the entire *past* history of $i(t)$ from $t = -\infty$ to $t = t_0$.

[1] A function is *piecewise-differentiable* if its derivative is uniquely defined everywhere except possibly at isolated points.

[2] Just as *memristor* is an acronym for *memory resistor*, *memristance* is an acronym for *memory resistance*. Similarly *memductance* is an acronym for *memory conductance*.

Similarly the memductance $G(\varphi)$ in Eq. (8) depends on the entire *past* history of $v(t)$ from $t = -\infty$ to $t = t_0$. It follows from Eq. (5) that the charge-controlled memristor defined in Eq. (3) is equivalent to the charge-dependent Ohm's law

$$v = R(q)i \tag{9}$$

where $R(q)$ is just the *slope* of the curve $\varphi = \hat{\varphi}(q)$ at q. To show Eqs. (3) and (5) are equivalent representations, we can recover Eq. (3) by integrating both sides of Eq. (5) with respect to t:

$$\begin{aligned}
\varphi \triangleq \int_{-\infty}^{t} v(\tau)d\tau &= \int_{-\infty}^{t} R\big(q(\tau)\big)i(\tau)d\tau \\
&= \int_{-\infty}^{t} R\big(q(\tau)\big)\frac{dq(\tau)}{d\tau}d\tau \\
&= \int_{q(-\infty)}^{q(t)} R\big(q(\tau)\big)dq(\tau) \\
&= \int_{q(-\infty)}^{q(t)} R(q)dq \\
&= \hat{\varphi}(q)
\end{aligned} \tag{10}$$

It follows from Eq. (10) that

$$\hat{\varphi}(q) = \int R(q)dq \tag{11}$$

Similarly, a flux-controlled memristor is equivalent to the flux-dependent Ohm's law

$$i = G(\varphi)v \tag{12}$$

where

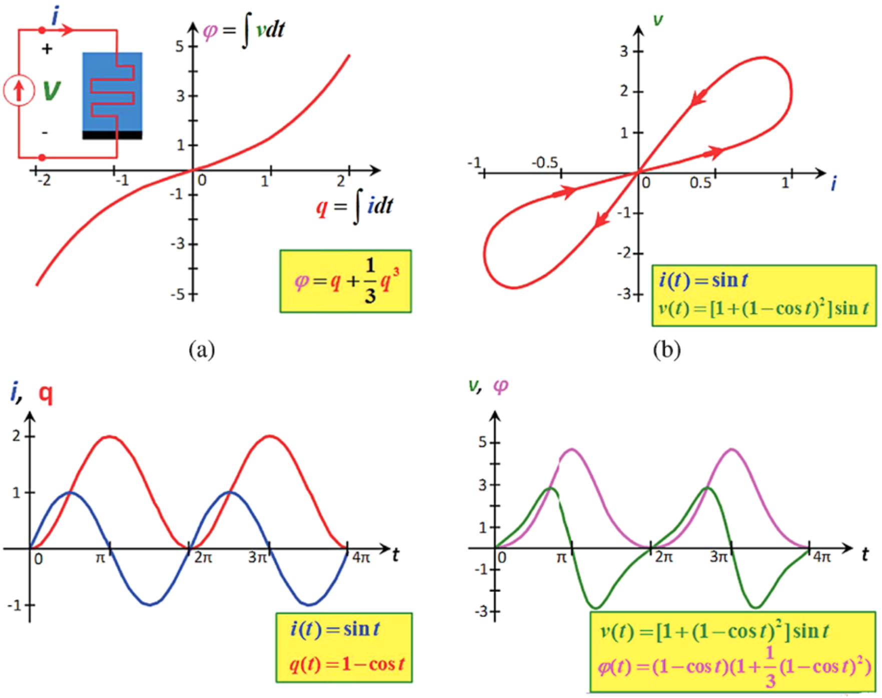
$$\hat{q}(\varphi) = \int_{\varphi(-\infty)}^{\varphi(t)} G(\varphi)d\varphi \tag{13}$$

Example 1 Consider the charge-controlled memristor shown in Fig. 1a along with the memristor symbol in the upper left corner. The memristor constitutive relation, shown in red, is described analytically by a cubic polynomial

$$\varphi = q + \frac{1}{3}q^3 \tag{14}$$

Let us apply a sinusoidal current source (blue sine wave in Fig. 1c) defined by

$$i(t) = \begin{cases} A \sin \omega t, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{15}$$



(a) (b)

(c) (d)

Fig. 1 Memristor symbol, constitutive relation, and pinched hysteresis loop associated with $(v(t), i(t))$ plotted for $A = 1$ and $\omega = 1$. (**a**) Memristor symbol and $\varphi = \hat{\varphi}(q)$ characteristic curve. (**b**) Pinched hysteresis loop: double-valued Lissajous figure of $(v(t), i(t))$ for all times t except when it passes through the origin, where the loop is pinched. (**c**) Periodic waveforms associated with $i(t)$ and $q(t) = \int_0^t i(\tau) d\tau$ plotted with $A = 1$ and $\omega = 1$. (**d**) Periodic waveforms associated with $v(t)$ and $\varphi(t) = \int_0^t v(\tau) d\tau$

across this memristor, as shown in Fig. 1c for $A = 1$ and $\omega = 1$. To determine the corresponding voltage response $v(t)$ from the constitutive relation (14), we must calculate first the corresponding charge (shown in red in Fig. 1c). Assuming the initial charge $q_0 = q(0) = 0$, we obtain upon integrating Eq. (15) the following equation for $q(t)$:

$$q(t) = \int_0^t A \sin(\omega\tau) d\tau = \frac{A}{\omega}[1 - \cos \omega t], \quad t \geq 0 \tag{16}$$

Substituting Eq. (16) into Eq. (14), we obtain the corresponding flux (shown in magenta in Fig. 1d)

$$\varphi(t) = \frac{A}{\omega}(1 - \cos \omega t)\left[1 + \frac{1}{3}\left(\frac{A^2}{\omega^2}\right)(1 - \cos \omega t)^2\right] \tag{17}$$

Differentiating Eq. (17) with respect to t, we obtain

$$v(t) = A\left[1 + \frac{A^2}{\omega^2}(1 - \cos\omega t)^2\right]\sin\omega t \tag{18}$$

Plotting the loci[3] of $(i(t), v(t))$ in the v–i plane, via Eqs. (15) and (18), we obtain the *pinched hysteresis loop* shown in Fig. 1b for $A = 1$ and $\omega = 1$. The hysteresis occurs because the maxima and minima of the sinusoidal input current $i(t)$ in Fig. 1c do not occur at the same time as the corresponding memristor voltage $v(t)$ in Fig. 1d. The *pinching* at the origin in Fig. 1b occurs because both $i(t)$ and $v(t)$ become zero *at the same time.* To show that the hysteresis loop is always pinched at the origin $(v, i) = (0, 0)$, let us calculate the *memristance* $R(q)$ from Eqs. (6) and (14):

$$R(q) = \frac{d\hat{\varphi}(q)}{dq} = 1 + q^2 \tag{19}$$

Substituting $q(t)$ from Eq. (16) into Eq. (19), we obtain

$$R\big(q(t)\big) = 1 + \left[\frac{A}{\omega}(1 - \cos\omega t)\right]^2, \quad t \geq 0 \tag{20}$$

Observe from Eq. (19) and Fig. 2c that

$$R(q) > 0 \tag{21}$$

Substituting Eqs. (20) and (15) in Eq. (9), we obtain the same expression for the memristor voltage $v(t)$ derived earlier in Eq. (18).

Now since $R(q)$ is *finite* for all finite q, it follows that

$$v(t) = 0 \quad \text{whenever } i(t) = 0 \tag{22}$$

for *any* input current $i(t)$. Similarly, for any φ-controlled memristor whose memductance $G(\varphi)$ is finite for all finite φ, we have

$$i(t) = 0 \quad \text{whenever } v(t) = 0 \tag{23}$$

for *any* input voltage $v(t)$.

The waveform of $R(t)$ given by Eq. (20) is shown in Fig. 2d for $A = 1$ and $\omega = 1$. The loci traced out by $(R(t), i(t))$ is shown in Fig. 2b. Again we obtain a hysteresis loop, but it is *not* pinched since $R(t) > 0$ for all times.

It is important to observe from Figs. 1d and 2d that while $v(t)$ and $i(t)$ assume both positive and negative values, both $\varphi(t)$ and $q(t)$ are *non-negative*. It follows that only the memristor φ–q curve in the first quadrant is visited during every period of $i(t)$. Observe also that the pinched hysteresis loop in Fig. 1b is *odd symmetric* with respect to the origin.

[3] Also known as a *Lissajous figure*.

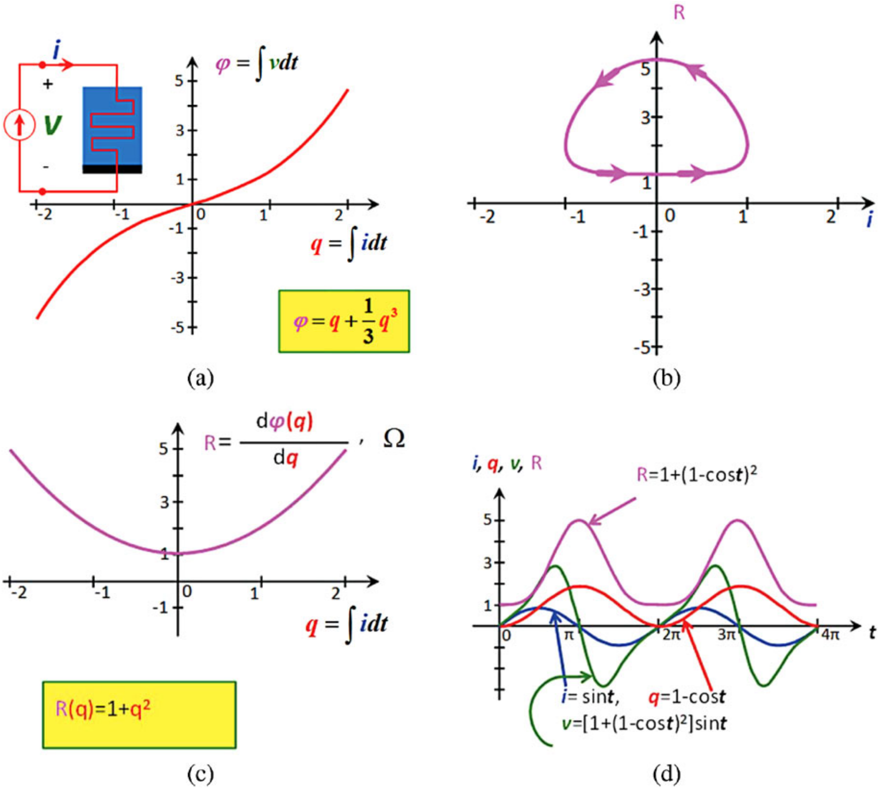


Fig. 2 (**a**) Memristor constitutive relation. (**b**) Resistance hysteresis loop associated with $(R(t), i(t))$. (**c**) Memristance plotted as a function of q. (**d**) Periodic waveforms of $i(t)$, $q(t)$, $v(t)$, and $R(t)$, plotted for $A = 1$, $\omega = 1$

Finally, observe that except for the *memristor constitutive relation* $\varphi = \hat{\varphi}(q)$ in Fig. 1a, and its associated *memristance* $R(q)$ in Fig. 2c, which remain unchanged, all other waveforms of Figs. 1 and 2 will change when we vary the amplitude A or the frequency ω of the input signal $i(t) = A \sin \omega t$. In particular, note from Eqs. (16) and (17) that $q(t) \to 0$ and $\varphi(t) \to 0$ as $\omega \to \infty$. This makes perfect sense since as we increase the frequency ω of the sinusoidal input current $i(t) = A \sin \omega t$, while keeping the amplitude constant, the "area" accumulated from $t = 0$ to the first half period $t = \frac{\pi}{\omega}$ diminishes with ω. It follows therefore that the memristance

$$R\big(q(t)\big) \to R(0) = 1\ \Omega, \quad \text{as } \omega \to \infty \tag{24}$$

We can confirm this prediction via Eq. (18) by noting that

$$v(t) \to A \sin \omega t, \quad \text{as } \omega \to \infty \tag{25}$$

In fact, this is one of the signature properties of a memristor, which we formalize as follows:

Memristor pinched hysteresis loop fingerprint

The loci (Lissajous figure) in the v–i plane of any *passive memristor* with positive memristance

$$R(q) = \frac{d\hat{\varphi}(q)}{dq} > 0 \tag{26}$$

and driven by a sinusoidal current source $i(t) = A \sin \omega t$ is always a *pinched hysteresis loop*, whose area shrinks with frequency and tends to a linear resistance equal to $R(0) =$ slope of the constitutive relation $\varphi = \hat{\varphi}(q)$ at $q = 0$.

We remark here that there exist degenerate cases where the v–i Lissajous figure is a single-valued function, such as the example shown in Fig. 3 when we drive the same memristor from Fig. 1 with the special input $i(t) = \cos t$ for $t \geq 0$. In fact, we can interpret the loci shown in Fig. 3b as a degenerate case where the hysteresis loop collapsed into a single-valued function, passing through the origin. Hence, the loci is still *pinched*, even in this degenerate scenario.

Another degenerate scenario can occur when the slope $R(q) = 0$ at some points on $\varphi = \hat{\varphi}(q)$ function, as illustrated in Fig. 4 for the constitutive relation

$$\varphi = \frac{1}{3} q^3 \tag{27}$$

In this case $R(0) = 0$. For the same input current source $i(t) = \cos t$ as in Fig. 3, we obtain a single-valued function in the v–i plane which touches the i-axis, as shown in Fig. 4b. This represents another degenerate situation where the v–i Lissajous figure actually includes points on the i-axis, as it is impossible to cross the i-axis for any passive memristor where $R(q) \geq 0$.[4] In such situations, the v–i Lissajous figure must still pass through the origin (i.e., it is pinched), but it makes contact with the i-axis as well. In either case, the Lissajous figure ($R(q) \geq 0$) of a *passive* memristor must be confined to the *first* and the *third quadrants*, including possibly the i-axis, of the v–i plane [3].

2 Continuum of Non-volatile Memories

A cursory examination of the charge-controlled memristor constitutive relation $\varphi = \hat{\varphi}(q)$ in Fig. 2a shows that its memristance $M(q)$ varies from[5] $R(q_1) =$

[4] Note the preceding memristor fingerprint property is stated for the case $R(q) > 0$.

[5] To avoid clutter, we will often write Memristance $M(q)$ and Resistance $R(q)$ interchangeably. Likewise, we will often write Conductance $G(\varphi)$ for Memductance $W(\varphi)$. Similarly, we use the terms memristance and resistance, as well as memductance and conductance, to mean the same thing.

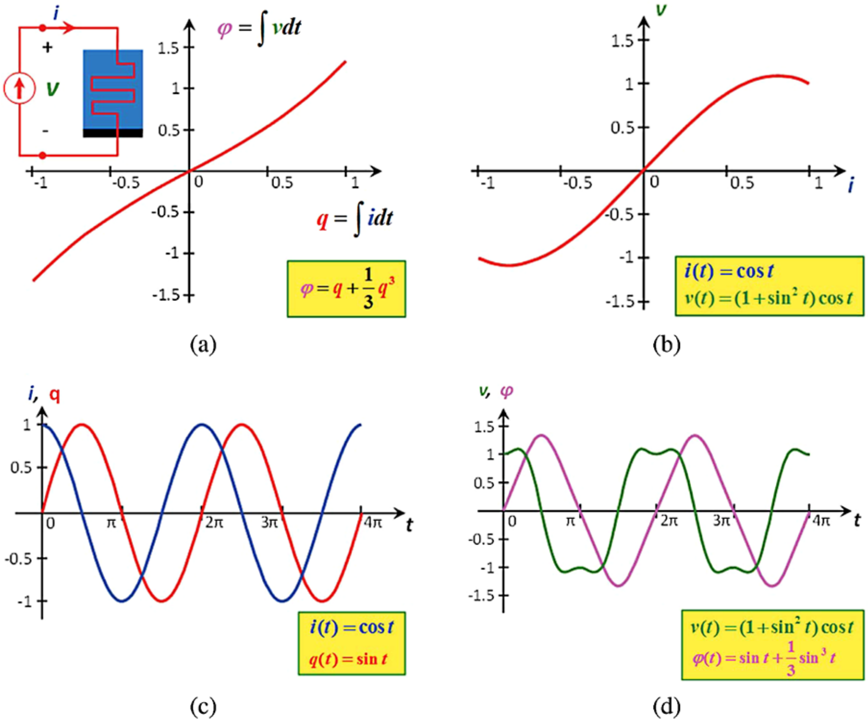
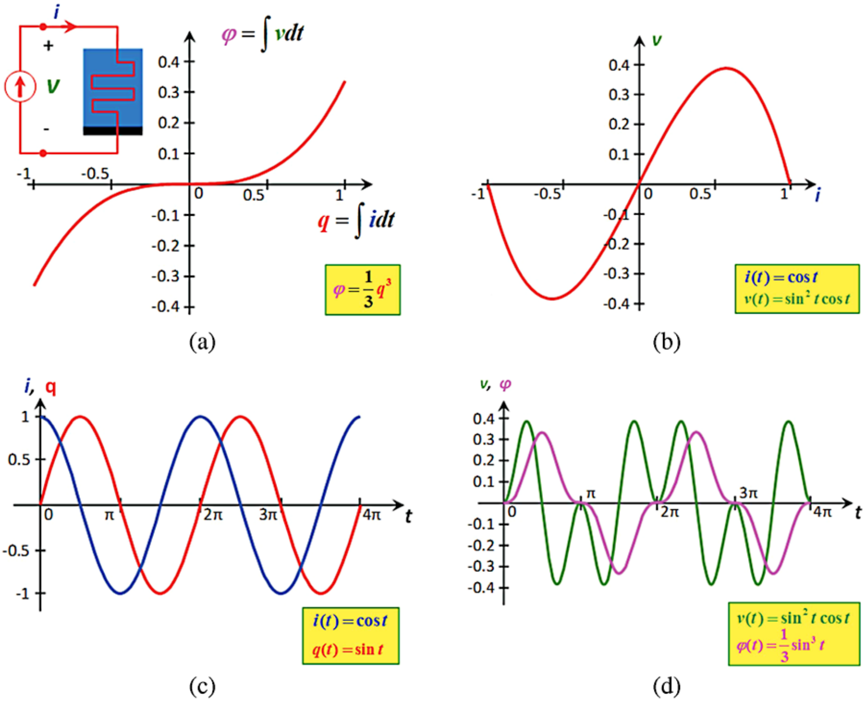


Fig. 3 Illustration of a *degenerate* scenario where the pinched hysteresis loop collapsed into a single-valued function when driven in this case with $i(t) = \cos t$, with the same $\varphi = \hat{\varphi}(q)$constitutive relation as Fig. 1a

l Ω to ∞, as depicted in the "Resistance vs. charge" curve in Fig. 2c, henceforth called the *Resistance vs. State map*. In Fig. 2c, *charge* is the *state variable*.

The *Resistance vs. State map*is a very useful graph because it shows how to navigate from one memristance R_0 at state $q = q_0$ on the memristor φ vs. q curve to another memristance R_1 at state $q = q_1$ by simply applying a short current pulse $\Delta i(t)$ whose area is equal to the increment Δq needed to be added to the latest value of $q(t_0) = q_0$ in order to move from R_0 to R_l. The memristance vs. state map in Fig. 2c therefore allows one to tune the memristor's resistance continuously from $R = 1$ Ω to $R = \infty$.

It is important to observe that if one opens or short circuits a memristor having a resistance R_0 at $t = t_0$ so that the memristor is *in equilibrium*, i.e., $v = 0$, and $i = 0$, at $t = t_0$, the memristor does not lose the value of φ and q when both voltage v and current i became zero at the instant when the power is switched off, but rather held the value unchanged at q_0 and φ_0, forever! Hence the passive memristor exhibits *non-volatile* memory.



(a) (b)

(c) (d)

Fig. 4 Example illustrating the second degenerate scenario where the Lissajous figure in the v–i plane actually reaches the i-axis. This limiting case can occur when the memristor constitutive relation has $R(q) = 0$ at some q, as in Fig. 4a where $R(q) = 0$ at $q = 0$ [3]

3 φ–q Curve and Memristance vs. State Maps Are Equivalent Memristor Representations

Both the φ vs. q *constitutive relation* (such as Fig. 2a) and its associated *resistance vs. state map* (such as Fig. 2c) with the state equation $dq/dt = i$ are equivalent representations of a memristor in the sense that given any applied current source input signal $i(t)$ for all times from $t = -\infty$, or equivalently, for positive times from $t = 0$, plus the initial charge $q(0)$ which represents the time-integral of $i(t)$ from $t = -\infty \rightarrow t = 0$, one can calculate the corresponding voltage $v(t)$. Conversely, given any $v(t)$, one can calculate the corresponding $i(t)$, assuming $R > 0$ so that the inverse constitutive relation $q = \hat{q}(\varphi)$ is a continuous function.

In contrast, all of the waveforms and hysteresis loops depicted in Figs. 1 and 2 are only *manifestations* of a *memristor*, and cannot be used to predict the voltage response given any other excitation waveforms different from the waveform i = $A \sin \omega t$, with $A = 1$ and $\omega = 1$, in Fig. 1c. The reader should verify that changing the parameter A, or ω, or changing the waveforms of $i(t)$ would result in completely different responses. For example, it follows from Eqs. (16), (18) and (19) that if we hold the amplitude $A = 1$ while increasing the frequency $\omega \rightarrow \infty$, we would find that $q(t)$ tends to zero, $v(t)$ tends to 0, and $R(t)$ tends to 1 Ω, as the hysteresis loop in Fig. 1b shrinks until it collapses into a unit-slope straight line through the origin. Indeed, as ω tends to ∞, the charge $q(t)$, and flux $\varphi(t)$ would both tend to the origin in Fig. 1a, and remain motionless thereafter. Under this limiting situation, the memristor degenerates into a linear R Ω resistor where R is just the slope of the φ–q curve at the origin in Fig. 1a; namely, $R = 1$ Ω.

Memristor Lesson 1. Pinched Hysteresis Loops Are Not Models! While a *pinched v–i* hysteresis loop measured from an experimental 2-terminal device implies that the device is a memristor, the pinched loop itself is useless as a model since it cannot be used to predict the voltage response to arbitrarily applied current signals, and vice-versa. The only way to predict the response of the device is to derive either the *φ–q constitutive relation*, or the *memristance* vs. state *map*.

4 Resistance vs. State Map and State Equation

When we write, or utter, the term *resistance*, or *conductance*,[6] we must always subconsciously remind ourselves that we are referring to a 2-terminal electrical device that obeys a *linear equation* called *Ohm's Law*; namely,

Ohm's Law:

$$v = Ri \tag{28}$$

where R is a constant, called the resistance of the resistor, where R has the unit of Ω. It is conceptually important to distinguish between the two words *resistance* and *resistor*: resistor is a device, while resistance is the slope of the straight line defined by Ohm's law. No harm is done when the device is linear-hence the sloppiness in current usage. However, for nonlinear devices, it is crucial to distinguish them! The *resistance vs. state map* of a memristor also obeys Ohm's Law, except that the resistance R is not a constant, as illustrated by the example in Fig. 2c, but depends on a dynamical state variable x ($x = q$ in the *ideal* memristor case considered so far) which evolves according to a prescribed ordinary differential equation, called the state equation. An ideal memristor is therefore defined by:[7]

State-dependent Ohm's Law:

$$v = R(x)i \tag{29a}$$

[6] To avoid clutter, we usually write only the term resistance, or conductance, with the understanding, *mutatis mutandis,* that the same follows for the dual case.

[7] We henceforth adopt the standard notation x to denote a *state variable* in *mathematical system theory*, where x may be a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. This will be the case for many *non-ideal* memristors found in practice.

Memristor State equation:

$$\frac{dx}{dt} = i \tag{29b}$$

Memristor Lesson 2 A *memristor* is defined by a *state-dependent Ohm's Law*.

5 Correspondence Between *Small-Signal Memristance* and *Chord Memristance*

Let us apply a sinusoidal current source $i(t) = A \sin \omega t$ across a charge-controlled memristor as in Fig. 1. The memristance $R(q(t_k))$ at $t = t_k$ as calculated from Eq. (6) is equal to the *slope* of the φ–q curve at $q = q(t_k)$. The slope at $q(t_k)$ will in general vary with the time-evolution of $\varphi(t)$. However, we can keep the slope at $q(t_k)$ approximately constant over time by choosing a sufficiently small amplitude A while fixing the frequency ω, assuming the φ–q curve is continuous at $q = q(t_k)$. Under this small-signal condition, the memristance, henceforth called the *small-signal memristance*, would be indistinguishable from that of a linear resistance, which obeys Ohm's Law with a constant resistance equal to $R(t_k)$ at all times. It follows that by applying a short current pulse signal of appropriate height, we can tune the memristance over a continuous range of values without introducing a third terminal, and without applying a continuous supply of power via a biasing circuit. For the example shown in Fig. 2c, any small-signal memristance greater than 1 Ω can be easily programmed. In particular, observe that we have aligned the vertical axis of Figs. 2a and 2c so that the value of R (height of the resistance vs. state map) is equal to the slope of the φ–q curve in Fig. 2a at the point $(q(t_k), \varphi(t_k))$, i.e., both points must fall on the vertical projection line through $q = q(t_k)$.

In other words, the memristor can be designed to function as a *non-volatile* and continuously tunable resistance. Let us consider next the large-signal case where $A \gg 0$, e.g. $A = 1$ and $\omega = 1$, as shown in Fig. 2 In this case, a quick calculation using Eq. (17) shows that the flux $\varphi(t)$ oscillates between $\varphi = 0$ and $\varphi = 14/3$, as shown in Fig. 1d. The corresponding memristance calculated from Eq. (20) ranges from $R = 1$ to $R = 5$ Ω, as shown in Fig. 2d. The corresponding v–i Lissajous figure is the pinched hysteresis loop shown in Fig. 1b. At any time $t = t_k$, the memristance is equal to $R(t_k) = \frac{v(t_k)}{i(t_k)}$. This number can be interpreted simply as the slope of a straight line, i.e., a *chord*, connecting the origin to the point $(i(t_k), v(t_k))$ in the i–v plane. We will henceforth call this large-signal resistance at time $t = t_k$ the "chord memristance" at $t = t_k$.[8]

Observe that the chord memristance at $t = t_k$ is simply the memristance calculated from the pinched hysteresis loop in Fig. 1b at the point where $t = t_k$. This

[8] The terminology "chord resistance" had been widely used by neuro-biologists, including Hodgkin and Huxley [5], for similar geometrical interpretations.

number is equal to the slope of a corresponding point on the φ–q curve in Fig. 1a, traversed at the same time $t = t_k$; namely, the *small-signal memristance* calculated at the same point. In fact, had we plotted Figs. 1a and 1b on the same scale, the *chord* connecting the point $(i(t_k), v(t_k))$ to the origin at $t = t_k$ will be parallel to a corresponding line drawn tangent to the φ–q curve in Fig. 1a.

For example, at $t = \frac{\pi}{2}$, $(i(\frac{\pi}{2}), v(\frac{\pi}{2})) = (1, 2)$, and the chord resistance is given by $R(\frac{\pi}{2}) = 2/1 = 2\ \Omega$, and the corresponding small-signal memristance is given by Eq. (19) for $q(\frac{\pi}{2}) = 1$, namely, $R(\frac{\pi}{2}) = 1 + 1 = 2$, as predicted and shown in Fig. 2c. Let us summarize the above results as follow:

Small-Signal and Chord Memristance Correspondence Property The large-signal *chord memristance* calculated at any point $(i(t_k), v(t_k))$ at time $t = t_k$ of a pinched hysteresis loop in the v–i plane is equal to the small signal memristance at a corresponding point on the φ–q curve traversed at the same time. In particular, the slope of the chord connecting $(0, 0)$ to $(i(t_k), v(t_k))$ is equal to the slope of the line drawn tangent to the φ–q curve at the corresponding point $(q(t_k), \varphi(t_k))$.

Recall that the small-signal memristance $R(q(t))$ remains constant under any sufficiently small odd-symmetric periodic current input signals, such as $i(t) = A \sin \omega t$ where $i(-t) = -i(t)$ because every value of the state variable x (charge in Fig. 1) is a *stable equilibrium point*[9] and because the memristor is *locally passive* when $R(q) \geq 0$ [3]. The *local passivity property* is essential for small-signal memristor circuit analysis to make sense because a *locally active* memristor [3] could give rise to oscillations, and even chaos [6].

In contrast to the small-signal memristance, which does not depend on the input waveform of $i(t)$ other than it being sufficiently small, the chord memristance is always associated with a particular Lissajous figure, such as a pinched hysteresis loop corresponding to a periodic input signal. However, once the input current waveform is given, we can derive the associated pinched hysteresis loop, such as that shown in Fig. 1b when $i = A \sin \omega t$ with $A = 1$ and $\omega = 1$. In this case, we can interpret the meaning of the two limiting chord memristances associated with the two hysteretic branches through the origin. In particular, the chord memristance of the lower limiting branch is equal in value to the small-signal memristance at the origin of the φ–q curve in Fig. 2a, namely, $R(0) = 1$. This also follows upon substitution of $q = 0$ in Eq. (19) at time $t = 0$. The second chord memristance associated with the limiting upper branch through the origin in Fig. 2b is associated with the small-signal memristance at the point $q = q(\pi) = 2$, namely, $R(2) = 5$.

[9] A state $x = x_o$ is said to be an *equilibrium point* of a dynamical circuit if $\frac{dx(t)}{dt} = 0$ at $x = x_o$. It is said to be *locally asymptotically stable* if it always returns to its original position whenever subjected to small perturbations, such as a small current pulse. An equilibrium point is said to be *stable* if any drift from its original position due to any perturbation to the state variable x is confined to a neighborhood of radius of about the same size as that of the perturbation. In other words, it does not diverge to infinity, as would be the case for an unstable equilibrium point. Neither does it return to its original position, as would be the case if the equilibrium point is asymptotically stable [3].

For the pinched hysteresis loop shown in Fig. 1b, the chord memristance will sweep from the lower limit $R(0) = 1$ to the upper limit $R(2) = 5$ in a counterclockwise direction in the 1st quadrant during the first half cycle, and then reversing the sweep in a symmetrical manner in the 3rd quadrant during the second half cycle, resulting in an odd-symmetric pinched hysteresis loop. The motion of the chord memristance in the 1st quadrant Fig. 1b is similar to that of an automobile windshield wiper except that the length of the blade changes continuously in accordance to the square root of the sum of squares of $i(t)$ and $v(t)$, from $t = 0$ to $t = \pi$ in the v–i plane.

6 Ideal Memristor φ–q Curves for Binary Memories

For digital computer applications requiring only two memory states, the memristor needs to exhibit only two sufficiently distinct equilibrium states R_0 and R_1 where $R_0 \gg R_1$, and such that the high resistance state R_0 can be easily switched to the low resistance state R_1, and vice versa, as fast as possible while consuming as little energy as possible. In contrast to conventional memories, the memristor does not dissipate any power except during the brief switching time intervals because $v(t) = d\varphi(t)/dt = 0$, and $i(t) = dq(t)/dt = 0$ at both equilibrium states R_0 and R_1. Our goal in this section is to present two ideal memristors for mimicking two, among many, recently published resistance switching memories.

Memristor Switching Memory 1 Figure 5 shows a charge-controlled memristor characterized by a 3-segment odd-symmetric φ–q curve (Fig. 5b). This piecewise-linear function can be described by the equation

$$\varphi = R_0 q + \left(\frac{1}{2}(R_1 - R_o)\right)\big[|q + B| - |q - B|\big] \tag{30}$$

where R_1 denotes the slope of the middle segment in Fig. 5b, R_0 denotes the slope of the outer segments in Fig. 5b, $q = -B$ denotes the left charge breakpoint in Fig. 5b, and $q = B$ denotes the right charge breakpoint in Fig. 5b. The corresponding memristance function $R(q)$ is derived by differentiating Eq. (30) with respect to q; namely,

$$R(q) = R_0 + \frac{1}{2}(R_1 - \mathrm{R}_0)\big[\operatorname{sgn}(q + B) - \operatorname{sgn}(q - B)\big] \tag{31}$$

where sgn$(\cdot)$ is defined by

$$\operatorname{sgn} x = \begin{cases} 1, & \text{if } x > 0 \\ -1, & \text{if } x < 0 \end{cases} \tag{32}$$

A graph of *the memristance vs. state map* is shown in Fig. 5d for the parameter values $R_0 = 6000\ \Omega$ and $R_1 = 2500\ \Omega$.

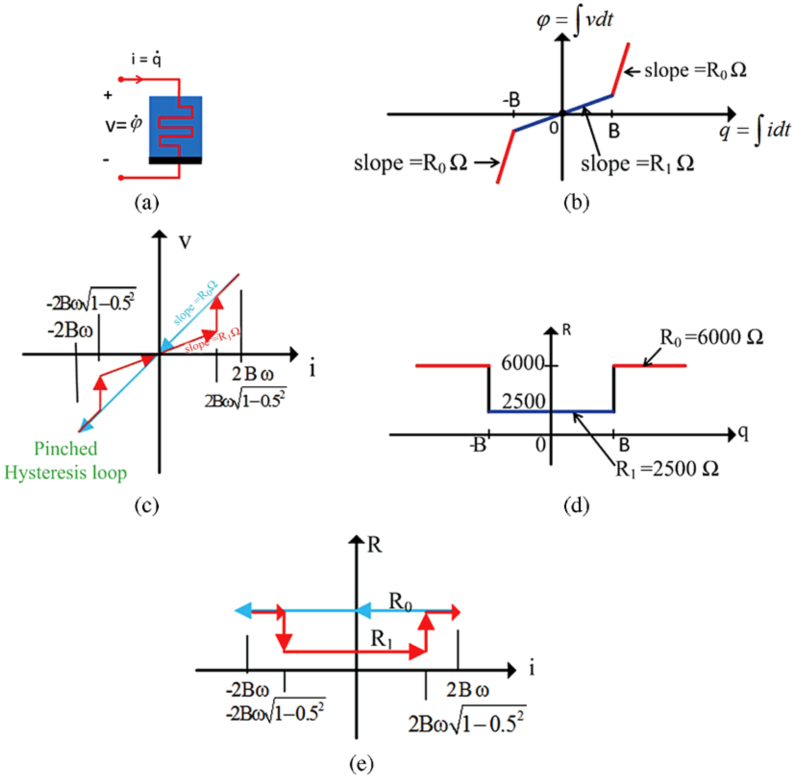


Fig. 5 A two-state pinched hysteresis loop resulting from driving a piecewise-linear charge-controlled memristor with a sinusoidal current source $i(t) = A \sin \omega t$, where $A > \omega B$, and B denotes the numerical value of the breakpoint in (**b**). Notice the horizontal axis is "q" in (**b**) and "i" in (**c**), which are the vertical axis in Figs. 6c and 6(f), respectively. Consequently, the slope of the piecewise-linear segments in (**b**) represents *memristance in* Ω. *(**d**) shows the relationship between the memristance as a function of q, assuming the slopes are given by* $R_o = 6000\ \Omega$ *and* $R_1 = 2500\ \Omega$

Applying the sinusoidal current source defined in Eq. (16) with $A = 2B\omega$ across the memristor, the corresponding memristor charge is given by

$$q(t) = \begin{cases} 2B(1 - \cos \omega t), & t > 0 \\ 0, & t < 0 \end{cases} \tag{33}$$

In this case, the memristor φ–q curve in Fig. 5b traverses from $q = 0$ at $t = 0$ to $q = 4B$ at $t = \frac{\pi}{\omega}$. Observe that starting from $q(0) = 0$ in Fig. 5b at $t = 0$, the memristor charge $q(t)$ increases along the lower branch while maintaining a constant memristance value of R_1 until it reaches the right breakpoint at $q = B$ where it switches abruptly to the upper branch and continues to increase, with the constant high memristance value of R_0, until it reaches the maximum value of $q(t) = 4B$ at $t = \frac{\pi}{\omega}$ corresponding to the end of the first half cycle of the sinusoidal current input $i(t)$. The corresponding chord memristance also remains constant at R_1 before the breakpoint $q = B$, and at R_0 after the breakpoint. During the next half cycle, the memristor input current $i(t)$ changes sign, and so does the corresponding memristor voltage $v(t)$. The loci in Fig. 5b then retraces the same route from $q = 4B$ with a constant memristance R_0 at $t = \frac{\pi}{\omega}$ until it reaches the right breakpoint $q = B$ again, where the memristance switches to R_1, and continue to decease until it returns to the initial departure point $q = 0$ at $t = 2\frac{\pi}{\omega}$. Since both $i(t)$ and $v(t)$ are negative during the return trip, the plot of the corresponding Lissajous figure in the v–i plane is an odd-symmetric pinched hysteresis loop, as shown in Fig. 5c. Observe that it consists of only two chord memristances equal to R_1 for the lower branch, and R_0 for the upper branch. Observe also that the switching occurs instantaneously, in both directions, in this case in view of the discontinuity in slope of the φ–q curve at the two breakpoints $q = B$ and $q = -B$.

The corresponding memristance vs. state map shown in Fig. 5d for $R_1 = 2500\ \Omega$, and $R_0 = 6000\ \Omega$, also shows a discontinuous jump at the same breakpoints, as expected. If we transcribe the corresponding loci of the memristance $R(t)$ from the pinched hysteresis loop in Fig. 5c into the R vs. i plane, we would obtain the square resistance hysteresis loop shown in Fig. 5e. This plot is the piecewise-linear analog of the smooth differentiable φ-q curve in Fig. 2b.

A cursory glance at the figures from [7] reveals similarities in the respective rectangular resistance hysteresis loops. From a circuit-theoretic perspective, the nonvolatile resistance switching memory device reported in [7] bears the fingerprint of a memristor, and should be modeled as a memristor. This example suggests that spin-transfer magnetic tunnel junctions are memristors. Indeed, unless a memristive device is properly identified and modeled as a memristor, no deep physical understanding of the rectangular *resistance hysteresis* mechanisms, let alone the development of a reliable commercial product, would be possible.

So far we have chosen charge-controlled memristors for illustrations. Let us now consider the dual case of a flux-controlled memristor where the flux φ is the independent variable.[10]

Memristor Switching Memory 2 Consider the flux-controlled memristor q–φ curve shown in Fig. 6f where q (vertical axis) is the charge in nano Coulomb (nC), and φ (horizontal axis) is the flux in Webers (Wb). This odd-symmetric piecewise-linear function can be described exactly by an equation involving two absolute-value functions; namely,

$$q = \frac{1}{2}G_1\big\{2\varphi + |\varphi - B| - |\varphi + B|\big\} \tag{34}$$

[10] For a strictly-passive memristor, defined by $R(q) > 0$, there is no mathematical difference between a charge-controlled memristor and a flux-controlled memristor except for the choice of the independent variable. However, for a *locally-active* memristor, defined by $R(q) < 0$ at some point on the φ–q curve, the difference becomes important because the φ–q curve in this case is no longer a single-valued function, and therefore does not have an inverse function.

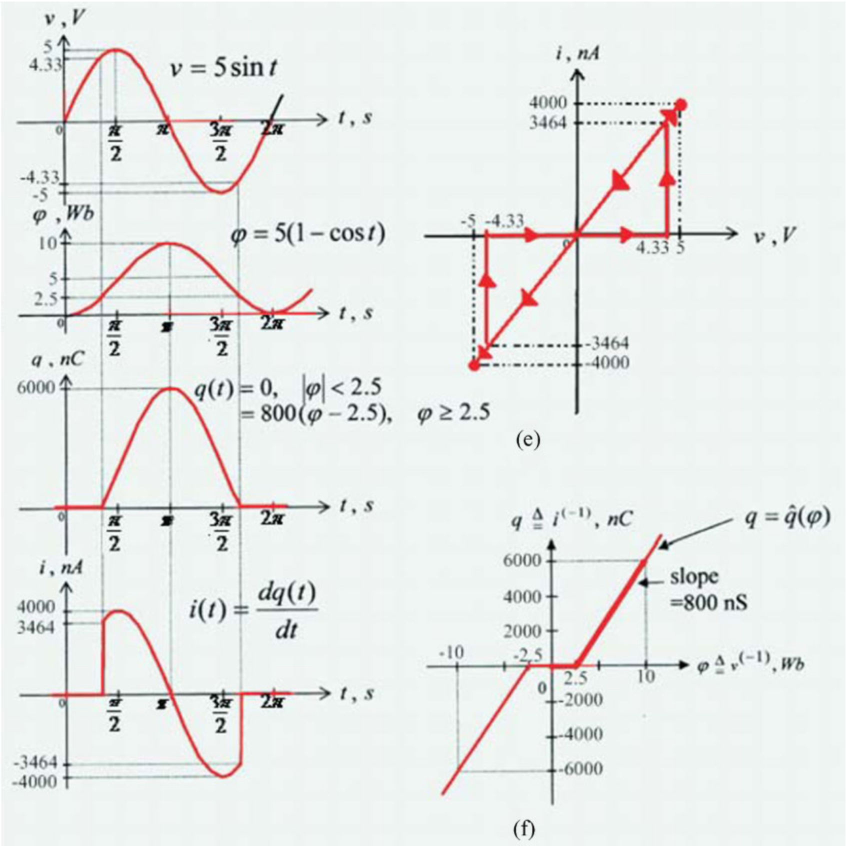


Fig. 6 A two-state *pinched hysteresis loop* resulting from driving a piecewise-linear *flux-controlled* memristor with a sinusoidal voltage source $v = 5\sin(t)$. The horizontal segment has a memductance $G(\varphi) = 0$ nS, and the two parallel outer segments have a memductance of $G(\varphi) = 800$ nS (Reproduced from Fig. 26 of [4], except for a revision of the original *cartoon sketch* (**e**) which was drawn distorted in order to unfold portions of the pinched hysteresis loop, as well as to exhibit a typical return loci for other periodic input signals)

where $G_1 = 800$ nS, and $B = 2.5$ Wb.[11]

Let us apply a sinusoidal voltage source

$$v(t) = \begin{cases} 5\sin t, & t > 0 \\ 0, & t < 0 \end{cases} \tag{35}$$

[11] This memristor is not charged-controlled because its memristance is infinite at all points on the horizontal segment where the memductance G_o is equal to zero.

shown in Fig. 6a, across the memristor. Integrating Eq. (35) we obtain the flux

$$\varphi(t) = \begin{cases} 5(l - \cos t), & t > 0 \\ 0, & t < 0 \end{cases} \tag{36}$$

as shown in Fig. 6b. Substituting Eq. (36) into Eq. (34), we obtain the corresponding charge

$$q(t) = 400\{10(1-\cos t) + |5(1-\cos t) - 2.5| - |5(1-\cos t) + 2.5|\} \tag{37}$$

as shown in Fig. 6c. Differentiating $q(t)$ from Eq. (37), we obtain

$$i(t) = 4000 \cdot \sin t \cdot \{1 + \theta(5(1-\cos t) - 2.5) - \theta(5(1-\cos t) + 2.5)\} \tag{38}$$

as shown in Fig. 6d, where

$$\theta(z) = \begin{cases} 1, & z > 0 \\ 0, & z < 0 \end{cases} \tag{39}$$

Plotting the Lissajous figure of $i(t)$ from Fig. 6d, or Eq. (38), and $v(t)$ from Fig. 6a, or Eq. (35), we obtain the *pinched hysteresis loop* shown in Fig. 6e. Since the current i is chosen as the vertical axis, and the voltage v is chosen as the horizontal axis, we must now use the dual terminology of *chord memductance*, instead of chord memristance. Observe that the memductance in Fig. 6e switches abruptly from $G_o = 0$ (horizontal segment) at the two breakpoint voltages $v = 4.33$ V, and $v = -4.33$ V, to $G_1 = 800$ nS. This switching is instantaneous because the slope of the q–φ curve in Fig. 6f changes abruptly at the corresponding breakpoints at $\varphi = 2.5$ Wb, and at $\varphi = -2.5$ Wb, respectively.

Observe that the pinched hysteresis loop in Fig. 6e has only two chord memductances. They correspond to the two small-signal memductances $G_o = 0$ and $G_1 = 800$ nS of the flux-controlled q–φ curve in Fig. 6f.

Let us now compare the dynamical behaviors of this memristor with the recent non-volatile nano-wire memory device reported by Professor Lieber's group from Harvard [8]. There seems to be little resemblance at first sight. This is because Lieber's group uses a square wave instead of a sinusoidal voltage source in their experiments. We have therefore repeated their experiments by applying the same voltage source, and parameters, across the flux-controlled memristor with the q–φ curve shown in Fig. 6f, and enlarged in Fig. 7a. Lieber's bipolar 10-volt square-wave input voltage $v(t)$ is shown in Fig. 7b. Integrating $v(t)$, we obtain the flux waveform $\varphi(t)$ shown in Fig. 7c, which is a triangular wave of the same frequency. Observe from Fig. 7a that the memductance is equal to zero for $|\varphi(t)| < 2.5$ Wb, and is equal to 800 nS elsewhere. It follows from Fig. 7c that the memductance $G(t)$ corresponding to the square wave voltage $v(t)$ in Fig. 7b will be a square wave of the same frequency, but delayed by 0.25 seconds. The memductance waveform predicted from the flux-controlled memristor constitutive relation is Fig. 7a is virtually identical to the experimental results reported in [8]. Moreover, by massaging the

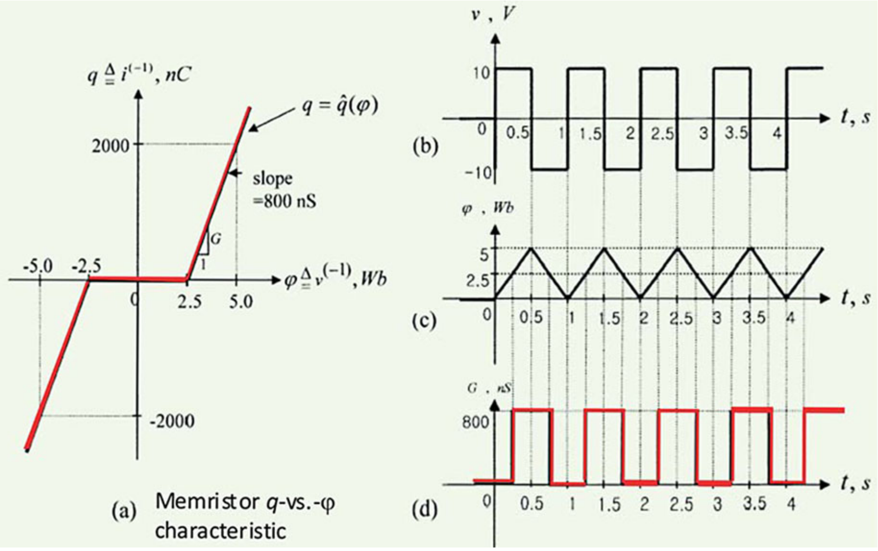


Fig. 7 Voltage and flux waveforms associated with the same memristor from Fig. 6f, but enlarged in (**a**). The memristor is driven by a ±10-volt square wave in (**b**), whose associated flux is the *triangular* wave shown in (**c**). The conductance waveform is a positive 800 nS square wave of the same frequency but shifted in time by 0.25 s. Observe that the conductance is zero over all times when $\varphi(t)$ in (**c**) falls below 2.5 Wb

q–φ curve into a smooth function, it is easy to obtain almost the same pinched hysteresis loop in the 1st quadrant as reported in [8]. There is one discrepancy, however, between our memristor prediction, and the experimental pinched hysteresis loop in [8]; namely, the pinched hysteresis loop predicted from the memristor in Fig. 7a is odd-symmetric, whereas that reported in [8] is not. In the next section, we will show how to *unfold* our ideal memristor model into a more general form that would allow us to model non-symmetric pinched hysteresis loops as well. Finally, we remark that, although not reported in [8], a private conversation with Prof. Lieber had confirmed that their hysteresis loop will shrink in size as the frequency of the input voltage signal increases, consistent with one of the fingerprints of a memristor.

7 Unfolding the Memristor

In order to develop a more precise quantitative model of non-volatile resistance switching memory devices, such as the nano-wire device cited in the preceding section, let us *unfold* the memristor's state-dependent Ohm's Law , and its associated state equation, defined earlier in Eqs. (29a), (29b), by introducing additional nonlinear terms, and parameters, while preserving the key properties of the memristor. Our approach is based on the mathematical theory of *unfoldings* of functions [9].

The foremost characteristic property of the memristor which distinguishes it from the other basic circuit elements defined axiomatically in [4] is its *pinched* hysteresis loop. The adjective "pinched" is chosen to emphasize that the loci, i.e., the Lissajous figure, of any bipolar current (resp., voltage) source waveform $i(t)$ (resp., $v(t)$), including chaotic signals, that is applied across the memristor, and its associated voltage (resp., current) response $v(t)$ (resp., $i(t)$), must pass through the origin $(v,i)=(0,0)$. This mathematical constraint can be generalized by introducing additional state variables, and the current i, into the state-dependent Ohm's Law and its associated state equation, defined in Eqs. (29a), (29b) as follow:

State-dependent Ohm's Law:

$$v = R(\mathbf{x}, i)i \tag{40a}$$

State Equation:

$$d\mathbf{x}/dt = \mathbf{f}(\mathbf{x}, i) \tag{40b}$$

where

$$R(\mathbf{x}, 0) \neq \infty \tag{41}$$

and

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \tag{42}$$

denotes a vector with n internal *state variables* $(x_1, x_2, \ldots, x_n)$. We stress here that the state variables are internal variables associated with the device material and its physical operating mechanisms, and *must not* be influenced by any external variable, such as a voltage or current applied to a third terminal, or a magnetic field generated from an external source. Observe that Eq. (41) is needed to ensure that $v=0$ whenever $i=0$. Indeed, if $R(\mathbf{x},i)$ tends to infinity when $i=0$, then $v=R(\mathbf{x},0)(0)\neq 0$ and the hysteresis loop would not be pinched at the origin.

We will illustrate the mathematical concept of *unfolding* with the following example of memristor Eqs. (40a), (40b) where x is a scalar:

$$v = R(x)i \tag{43a}$$

$$\frac{dx}{dt} = a_1x + a_2x^2 + \cdots + a_mx^m + b_1i + b_2i^2 + \cdots + b_ni^n + \sum_{j,k=1}^{p,r} c_{jk}x^j i^k \tag{43b}$$

By assigning different numerical values to the parameters a_j, b_k, c_{jk}, we can generate a very large family of distinct memristors, all of them originating from the same ancestor, namely, the original memristor defining Eqs. (29a), (29b). Just like the unfolding of flower petals, different parameter values gives rise to a memristor with a different pinched hysteresis loop. We will henceforth call these parameters the *memristor unfolding parameters*. Let us look at some special choices of these unfolding parameters.

Memristor Unfolding Example 1

$$\begin{aligned} &a_j = 0, \quad j = 1, 2, \ldots, m \\ &b_1 = 1, \qquad b_k = 0, \quad k = 2, 3, \ldots, n \\ &c_{jk} = 0, \quad j = 1, 2, \ldots, p, \ k = 1, 2, \ldots, r \end{aligned}$$

In this case, Eq. (43b) reduces to the original memristor equations (29a), (29b).

Memristor Unfolding Example 2 Let us choose the same unfolding parameters as above except b_1, where

$$b_1 = \mu_v \left[\frac{R_{ON}}{D} \right]$$

In this case, Eq. (43b) reduces to Eq. (6) from [10] describing the famous HP Titanium-Dioxide memristor reported in a seminal paper in the May 1 2008 issue of Nature [10].

Memristor Unfolding Example 3 Let us choose

$$a_j = 0, \quad j = 1, 2, \ldots, m$$

and

$$c_{jk} = 0, \quad j = 1, 2, \ldots, p, \ k = 1, 2, \ldots, r$$

In this case, the memristor unfolding assumes the following form:

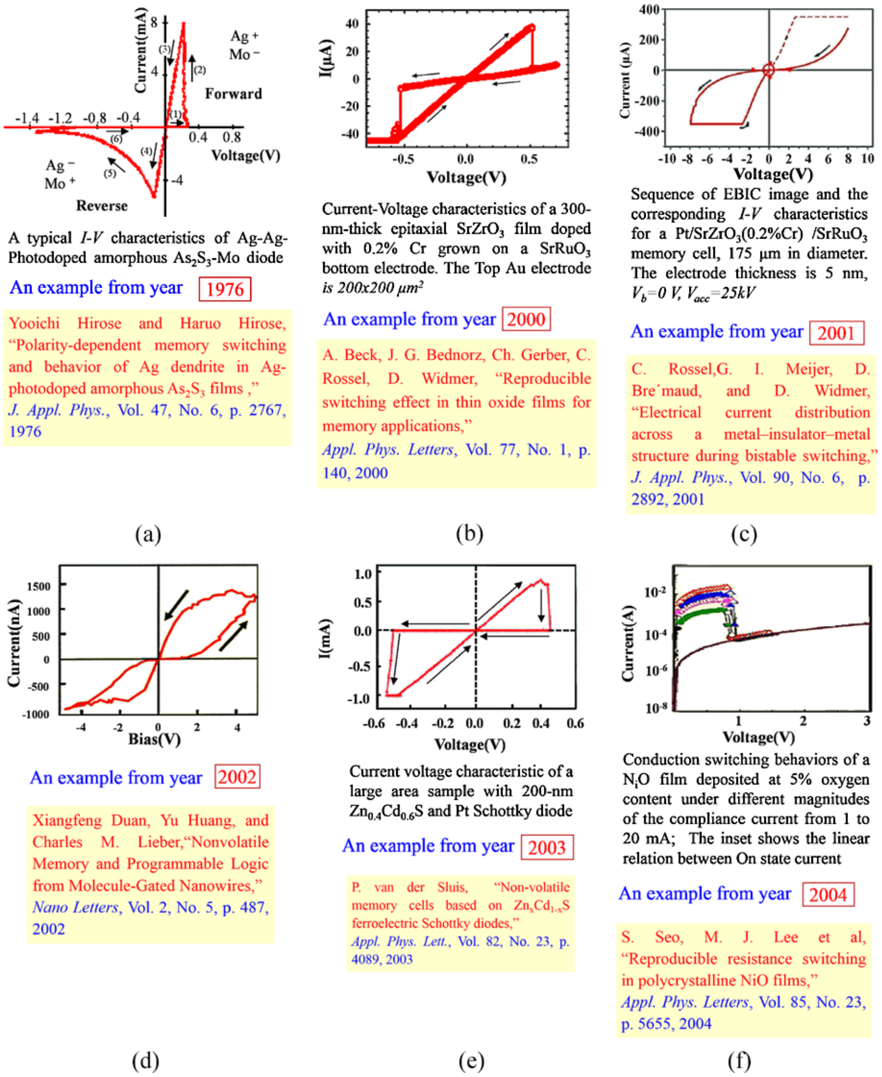
State-dependent Ohm's Law:

$$v = R(x) i \tag{44a}$$

State Equation:

$$dx/dt = m(i) \tag{44b}$$

By choosing different values for the unfolding parameters b_k, the resulting nonlinear scalar function $m(i)$ in Eq. (44b) can be used to massage the corresponding pinched hysteresis loop into almost any shape which best approximates the experimental data. In particular, the odd-symmetric pinched hysteresis loops shown in Figs. 1c, 5c, and 6e can be deformed and morphed into other non-symmetrical shapes, such as the one alluded to [8] in the previous section. We will henceforth call the function $m(i)$ in Eq. (44b) the "memristor morphing function" since it can be chosen to approximate numerous non symmetrical pinched hysteresis loops measured experimentally from real resistance- switching devices, such as those exhibited in Figs. 8a–8l, which were sampled from the literature on non-volatile resistance switching devices.



A typical *I-V* characteristics of Ag-Ag-Photodoped amorphous As_2S_3-Mo diode

An example from year 1976

Yooichi Hirose and Haruo Hirose, "Polarity-dependent memory switching and behavior of Ag dendrite in Ag-photodoped amorphous As_2S_3 films ," *J. Appl. Phys.*, Vol. 47, No. 6, p. 2767, 1976

Current-Voltage characteristics of a 300-nm-thick epitaxial $SrZrO_3$ film doped with 0.2% Cr grown on a $SrRuO_3$ bottom electrode. The Top Au electrode *is 200x200 μm^2*

An example from year 2000

A. Beck, J. G. Bednorz, Ch. Gerber, C. Rossel, D. Widmer, "Reproducible switching effect in thin oxide films for memory applications," *Appl. Phys. Letters*, Vol. 77, No. 1, p. 140, 2000

Sequence of EBIC image and the corresponding *I-V* characteristics for a Pt/$SrZrO_3$(0.2%Cr) /$SrRuO_3$ memory cell, 175 μm in diameter. The electrode thickness is 5 nm, V_b=0 V, V_{acc}=25kV

An example from year 2001

C. Rossel,G. I. Meijer, D. Bre´maud, and D. Widmer, "Electrical current distribution across a metal–insulator–metal structure during bistable switching," *J. Appl. Phys.*, Vol. 90, No. 6, p. 2892, 2001

An example from year 2002

Xiangfeng Duan, Yu Huang, and Charles M. Lieber,"Nonvolatile Memory and Programmable Logic from Molecule-Gated Nanowires," *Nano Letters*, Vol. 2, No. 5, p. 487, 2002

Current voltage characteristic of a large area sample with 200-nm $Zn_{0.4}Cd_{0.6}S$ and Pt Schottky diode

An example from year 2003

P. van der Sluis, "Non-volatile memory cells based on $Zn_xCd_{1-x}S$ ferroelectric Schottky diodes," *Appl. Phys. Lett.*, Vol. 82, No. 23, p. 4089, 2003

Conduction switching behaviors of a N_iO film deposited at 5% oxygen content under different magnitudes of the compliance current from 1 to 20 mA; The inset shows the linear relation between On state current

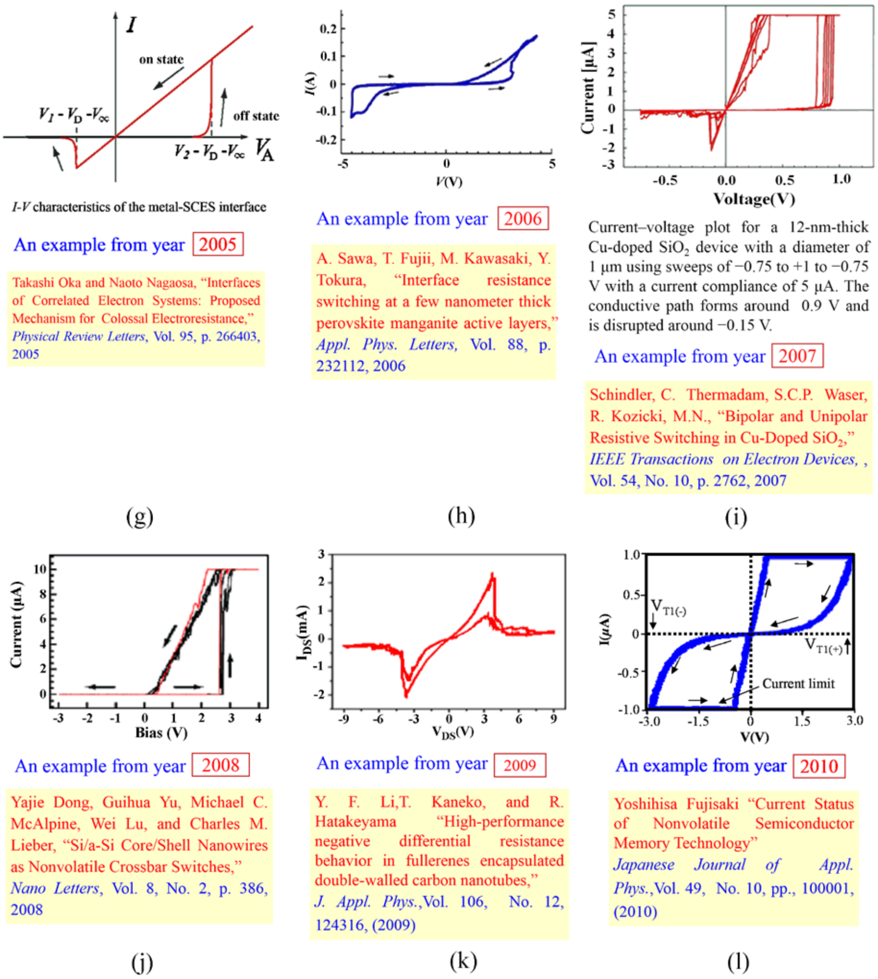
An example from year 2004

S. Seo, M. J. Lee et al, "Reproducible resistance switching in polycrystalline NiO films," *Appl. Phys. Letters*, Vol. 85, No. 23, p. 5655, 2004

Fig. 8 A sample of 12 experimentally measured pinched hysteresis loops extracted from dozens of similar loops published in the literature on a large variety of resistance switching devices, made from different materials, processes and physical mechanisms

7.1 *Non-volatile Memristors*

A careful examination of the 12 memristor pinched hysteresis loops exhibited in Figs. 8a to 8l shows that except for Figs. 8a, 8d, and 8h, most of the loops can be reproduced approximately by the preceding simpler memristor Eqs. (44a), (44b). A few of the pinched hysteresis loops, such as Figs. 8a, 8d, 8i and 8k contains



I-V characteristics of the metal-SCES interface

An example from year 2005

Takashi Oka and Naoto Nagaosa, "Interfaces of Correlated Electron Systems: Proposed Mechanism for Colossal Electroresistance," *Physical Review Letters*, Vol. 95, p. 266403, 2005

(g)

An example from year 2006

A. Sawa, T. Fujii, M. Kawasaki, Y. Tokura, "Interface resistance switching at a few nanometer thick perovskite manganite active layers," *Appl. Phys. Letters,* Vol. 88, p. 232112, 2006

(h)

Current–voltage plot for a 12-nm-thick Cu-doped SiO_2 device with a diameter of 1 µm using sweeps of −0.75 to +1 to −0.75 V with a current compliance of 5 µA. The conductive path forms around 0.9 V and is disrupted around −0.15 V.

An example from year 2007

Schindler, C. Thermadam, S.C.P. Waser, R. Kozicki, M.N., "Bipolar and Unipolar Resistive Switching in Cu-Doped SiO_2," *IEEE Transactions on Electron Devices,* , Vol. 54, No. 10, p. 2762, 2007

(i)

An example from year 2008

Yajie Dong, Guihua Yu, Michael C. McAlpine, Wei Lu, and Charles M. Lieber, "Si/a-Si Core/Shell Nanowires as Nonvolatile Crossbar Switches," *Nano Letters*, Vol. 8, No. 2, p. 386, 2008

(j)

An example from year 2009

Y. F. Li,T. Kaneko, and R. Hatakeyama "High-performance negative differential resistance behavior in fullerenes encapsulated double-walled carbon nanotubes," *J. Appl. Phys.*,Vol. 106, No. 12, 124316, (2009)

(k)

An example from year 2010

Yoshihisa Fujisaki "Current Status of Nonvolatile Semiconductor Memory Technology" *Japanese Journal of Appl. Phys.*,Vol. 49, No. 10, pp., 100001, (2010)

(l)

Fig. 8 (Continued)

small oscillatory or noisy signal components superimposed upon them. Since the cited authors did not provide details on how their pinched hysteresis loops were measured, we can only conjecture that these small signal components were either artifacts of their measurement systems, or they may represent genuine nonlinear dynamical phenomena. In the latter case, it may be necessary to use the generic memristor Eqs. (40a), (40b) to reproduce them. We wish to stress, however, that even this seemingly complex case would represent only the tip of an iceberg of vast nonlinear dynamical phenomena, such as chaos, which is not considered in this tutorial. Indeed, to build a non-volatile resistance switching memory exhibiting the fine details depicted in some of the pinched hysteresis loops shown in Fig. 8, we only need to consider a subclass of the memristor morphing function $\mathbf{f}(\mathbf{x}, i)$ in

Eq. (40b), namely, the class satisfying the condition

$$\mathbf{f}(\mathbf{x}, i) = \mathbf{0}, \quad \text{whenever } i = 0 \tag{45a}$$

Under the constraint imposed by Eq. (45a), the memristor state equation is thereby endowed with the following *non-volatility property*:

$$d\mathbf{x}/dt = \mathbf{f}(\mathbf{x}, i) = \mathbf{0}, \quad \text{when } i = 0 \tag{45b}$$

In other words, $(\mathbf{x}, i) = (\mathbf{x}, 0)$ is an equilibrium point of the memristor state Eq. (40b), for *any* value of $\mathbf{x}$. Hence, we have a continuum of *stable equilibrium points*, when $i = 0$, just as in the case of the ideal memristor of yore. This means that when we switch off the power at $t = 0$, such that $i(t) = 0$, for $t > 0$, the *state* vector $\mathbf{x}$ in Eq. (42) does not have to tend to zero, but is held unchanged at $\mathbf{x}(t) = \mathbf{x}(0)$ for all $t > 0$, where $\mathbf{x}(0)$ can be set by applying an appropriate input switching signal. But since we can choose many state variables, along with their numerous unfolding parameters, the device engineer has many degrees of freedom to massage his memristor model and optimize a memristance function $R(\mathbf{x}, i)$, and a corresponding memristor morphing function $\mathbf{f}(\mathbf{x}, i)$, to develop a memristor model capable of reproducing almost any fine details observed from their experiments.

7.2 Negative Resistance

Let us observe next that the pinched hysteresis loops shown in Figs. 8h and 8k contain a non-monotonic current-controlled region with a "negative" slope (i.e., a negative small-signal resistance), implying that the device is locally-active [3], and is capable of oscillation under dc bias. Such a pinched hysteresis loop could not be realized by any ideal passive memristor [1], but can be realized by connecting a locally-active current-controlled nonlinear resistor in series with a passive *memristor* described by Eq. (44a), as shown in Fig. 9a. Note that the resulting one-port in Fig. 9a is equivalent to a memristor described by the generic memristor Eqs. (40a), (40b).

To prove this equivalence property, let the memristor be described by

$$v_1 = R(\mathbf{x})i_1 \tag{46a}$$

Let the locally-active current-controlled nonlinear resistor be described by

$$v_2 = h(i_2) \tag{46b}$$

Applying Kirchoff Current Law (KCL), we obtain

$$i = i_1 = i_2 \tag{47}$$

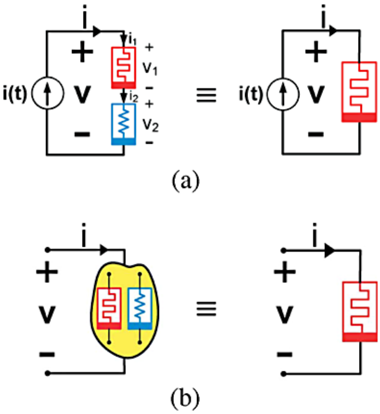


Fig. 9 The memristor-resistor series connected circuit in (**a**) is equivalent to another memristor with a transformed characteristic. In general, a one-port (2-terminal black box) made of arbitrary interconnections of arbitrary assortments of memristors and resistors is also equivalent to a memristor characterized by a more complex constitutive relation [3]

Applying Kirchoff Voltage Law (KVL), we obtain

$$v = v_1 + v_2 \tag{48}$$

Substituting Eqs. (46a), (46b), into Eq. (48), and making use of Eq. (47), we obtain the following equation for the one-port:

$$v = R(\mathbf{x})i + h(i) \tag{49}$$

Since Eq. (49) is a special case of Eq. (40a), the composite one-port in Fig. 9a is a memristor. The above example is but a special case of the following general result:

7.2.1 Memristor-Resistor Interconnection Theorem

Any one-port made of an arbitrary interconnection of *memristors* and *passive* nonlinear *resistors*, is equivalent to a memristor described by either Eqs. (40a), (40b), or by an implicit system of equations, whose behavior seen from outside the composite one-port shown in Fig. 9b bears all of the fingerprints of a memristor [2].

7.3 Is Memristor Negative Resistance Real or Artifact?

A careful examination of Figs. 8a, 8d, 8h and 8k reveals that these pinched hysteresis loops contain a small region with a negative slope. Assuming these regions are real measurements pertaining to the device, and not artifacts introduced via the measuring instruments, and/or their inflexible softwares, can we conclude that these devices are endowed with a *small-signal* (i.e., *differential*) *resistance* operating region, and hence is *locally-active*, and can be designed to amplify small signals, and/or to function as an oscillator [3] via an external biasing circuit?

The answer is *no*! Indeed, in many cases, the negative slope is merely a manifestation of a *phase-lag* between the *maxima* (or, *peak*) of the response voltage $v(t)$ (resp., current $i(t)$) and the *peak* of its excitation current waveform $i(t)$ (resp., voltage waveform $v(t)$). This phenomenon is best seen in Figs. 1b, 1c, and 1d where the voltage peak in Fig. 1d lags behind the input current peak in Fig. 1c. Observe that there is a short time interval where the voltage $v(t)$ in Fig. 1d increases while the input current $i(t)$ decreases. This phenomenon occurs after the pinched hysteresis loop in Fig. 1b reaches its peak at $i = 1$, and is the sole mechanism which gives rise to the negative slope. It has nothing to do with local activity [3]!

So how can we determine which of the pinched hysteresis loops in Fig. 8 with a negative-slope region is a *bona fide* small-signal resistance? The generic answer is we do not know unless we have already derived a realistic memristor circuit model, such as Fig. 9, or a memristor state equation, such as Eq. (40b), where we can find a point (V, I) on the negative-slope region of the pinched hysteresis loop which can be proved analytically to be an equilibrium point (otherwise known as a dc operating point in electronic circuit jargon), namely,

$$d\mathbf{x}/dt = \mathbf{f}(\mathbf{x}, I) = \mathbf{0}, \qquad V = R(\mathbf{x}(I), (I)I) \tag{50$'$}$$

for some state variable $\mathbf{x} = \mathbf{x}(I)$, which depends on I. This means that there exists a dc operating point (V, I) where, in the absence of noise, there is a state variable $\mathbf{x} = \mathbf{x}(I)$ where the *composite* memristor-biasing circuit is in *equilibrium*. This situation is usually not observable experimentally because the memristor small-signal resistance would usually make the circuit unstable, resulting in an *oscillation*. This alone suffices to conclude that the memristor is locally active. However, for pedagogical reasons, we can design an appropriate external stabilizing biasing circuit such that the composite circuit is *locally asymptotically stable* [3], whereupon the dc operating point (V, I) on the memristor pinched hysteresis loop can actually be measured. Alternately, we can determine whether the memristor is *locally active* by deriving first either a memristor circuit model, or a memristor state equation, and then apply standard nonlinear circuit analysis methods to determine whether there exists a *locally-active* equilibrium point [3].

Observe that for an ideal memristor we have $x = q$, and the equilibrium state equation

$$dq/dt = I = 0 \tag{50$''$}$$

does not have a solution if $I \neq 0$. It follows that *an ideal memristor can have only one dc operating point*; namely, the origin $(v, i) = (V, I) = (0, 0)$. If the small-signal resistance at the origin is negative, this would imply that the pinched hysteresis loop has a branch which crosses the origin into the 2nd and the 4th quadrants of the v–i plane, implying that the memristor is *not passive*. It follows therefore that *an ideal memristor cannot exhibit a small-signal negative resistance* unless it is locally active at the origin , which is possible only if the memristor has an internal source of power, such as light, chemical or nuclear reactions, or batteries, as demonstrated in Fig. 4(f), p. 511 of [1], where a locally-active memristor exhibiting a negative slope at the origin of the q vs. φ curve was built using transistors and op amps (see Fig. 2, p. 509 of [1]), powered by batteries. We can conclude therefore that if the pinched hysteresis loop of a physical device without internal power source exhibits a *bona fide* small-signal negative resistance, then that device cannot be an *ideal* memristor, and must therefore be an unfolded memristor sibling, characterized by Eqs. (40a), (40b).

8 Switching and Sensing Resistance Memory

We have presented in the preceding section a very special subclass, albeit of great interests to the theme of this special issue, of memristors whose members are endowed with the priceless, and timeless, gift of *non-volatile memories*. This subclass is defined by the memristor constitutive relation

$$v = R(\mathbf{x}, i)i \tag{51}$$

$$d\mathbf{x}/dt = \mathbf{f}(\mathbf{x}, i) \tag{52}$$

where the *memristance function* $R(\mathbf{x}, i)$ satisfies the memristor passivity condition

$$R(\mathbf{x}, i) \geq 0 \tag{53}$$

and where the *memristor dynamical function* $\mathbf{f}(\mathbf{x}, i)$ satisfies the following condition:

Continuous Non-volatility Condition

$$\mathbf{f}(\mathbf{x}, i) = \mathbf{0}, \quad \text{if } i = 0 \tag{54}$$

The non-volatility condition (54) ensures that *any* state variable $\mathbf{x}$ is a stable, non-isolated, *equilibrium point* of the memristor state equation (52) when $i = 0$, or equivalently, when the power is switched off. In other words, Eq. (54) is the genesis of the memristor's memory non-*volatility*. Observe that since every $\mathbf{x}$ is an equilibrium state of Eq. (52) when $i = 0$, the subclass of memristors defined by Eq. (54) has a *continuum of equilibrium states*, where every equilibrium state is stable, but not asymptotically stable [3] in the sense that while small perturbations around each equilibrium state may perturb its location slightly, it will never diverge beyond its perturbed boundary [3]. Hence, in principle, every memristor satisfying Eq. (54) is endowed with an infinite memory store. In the context of this special issue, we will consider only the special case of *binary* memory where only two sufficiently distant memory states are of interest because they will be used to store the "0" and "1" states for digital electronics. In this case, the ideal memristor φ vs. q curve only needs to have two approximately linear regions, where one region should have as small a slope as possible, while the other region should have as large a slope as possible.

The *Continuous Non-Volatility Condition* (Eq. (54)) guarantees a *continuum* of *tunable resistances*, which is essential for *synaptive learning applications*. For *non-volatile binary memory applications*, we can replace Eq. (54) by the following less restrictive condition:

Discrete Non-volatility Condition

$$\mathbf{f}\big(\mathbf{x}', 0\big) = \mathbf{0} \quad \text{and} \quad \mathbf{f}\big(\mathbf{x}'', 0\big) = \mathbf{0} \tag{54$'$}$$

where $\mathbf{x}'$ and $\mathbf{x}''$ denote two *locally-asymptotically-stable equilibrium points* of the memristor state equation (40b) with $i = 0$, i.e.,

$$d\mathbf{x}/dt = \mathbf{f}(\mathbf{x}, 0) \tag{40b$'$}$$

As an example, consider the state equation:

$$dx/dt = x - x^3 - i \tag{40b$''$}$$

Here, $x = x' = 1$ and $x = x'' = -1$ are two *isolated* locally-asymptotically-stable equilibrium points of

$$dx(t)/dt = x - x^3 \tag{40b$'''$}$$

obtained by setting $i = 0$ in Eq. (40b$''$).

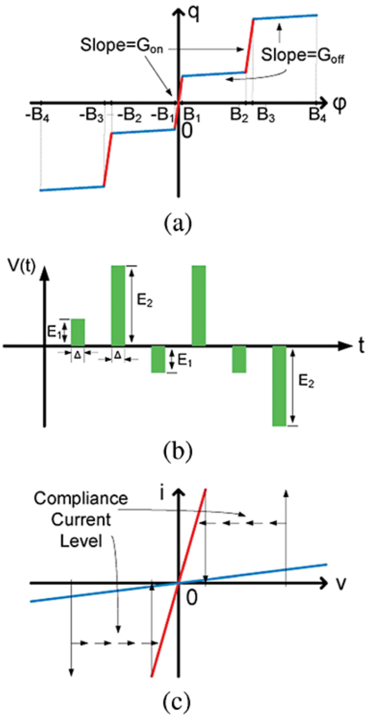
Let us now pause to consider some examples.[12]

Any device capable of non-volatile memory is useless unless it is relatively easy and inexpensive to sense its memory state. One of the great virtues of the memristor is that since its memristance function in Eq. (44a) is a state-dependent resistance obeying Ohm's law, one only needs to inject a small sensing voltage (resp, current), and observe its response. Since in practice, the two resistance memory states R_{OFF} and R_{ON} are chosen so that their ratio is sufficiently large, one can easily determine the memory state by observing the magnitude of the current (resp., voltage) response, to a small ac sensing voltage (resp., current) signal, or a small doublet-like pulse signal with a zero average area. The reason for requiring the sensing signal to have a zero dc average is to prevent the location of a *non-isolated* memory state from slowly drifting away.

Memristor Switching Example 1: Bipolar Switching Let us revisit the two-state charge-controlled memristor in Fig. 5. To switch from the low-resistance state R_1 corresponding to the middle segment with a small slope to the high resistance state R_0 corresponding to the upper segment with the much steeper slope, we simply apply a sufficiently large current pulse so that its corresponding charge $q(t)$ would traverse beyond the charge breakpoint $q = B$. To switch back from a point on the

[12] The two memory states are chosen sufficiently far apart in practice to enhance robustness and reliability.

Fig. 10 A “staircase-like” flux-controlled memristor can switch from a high conductance to a low conductance state using voltage pulses of the same polarity, somewhat reminiscent of the “unipolar” switching characteristic depicted in Fig. 1(a) of [7]



upper segment (*high* resistance state R_0, simply apply a similar pulse of the opposite polarity. This method of switching is usually referred to as *bipolar* resistance switching. Our next example illustrates how switching can be achieved by applying switching pulses of the same polarity, but of different amplitudes, often referred to in practice as *unipolar* resistance switching.

Memristor Switching Example 2: Unipolar Switching Consider the flux-controlled memristor depicted in Fig. 10 with a 5-segment piecewise-linear φ–q curve (Fig. 10a). Here the 3 parallel red segments with a steep slope have a high conductance state G_{ON}, whereas the 4 parallel green segments with a much smaller slope have a much smaller conductance state G_{OFF}. For the memristor constitutive relation shown in Fig. 10a, we can switch from a high conductance state to a low-conductance state with a relatively small-amplitude voltage pulse since it only needs a small increment $\Delta\varphi$ in φ to cross the breakpoint B_1 into the low conductance state. In contrast, a much larger-amplitude voltage-pulse, but of the *same polarity*, and the *same pulse width*, would be needed in order to reach the next breakpoint B_2, and beyond, in order to switch back to a high conductance state G_{ON} again. The same switching sequence with the opposite polarity can also be executed to achieve the same results, as illustrated in Fig. 10b. The corresponding switching loci plotted in the v–i plane is shown in Fig. 10c. Here, to prevent the excessive current jump from a small current to a very high current, thereby damaging the device, measuring instruments are normally programmed to clamp the current at a maximum safe value, called the “compliance current level” in industry, as illustrated in Fig. 10c. The above mode of using voltage pulses of the same polarity to switch between low and high resistance states has been reported in some so-called “*unipolar*” devices in industry [11].

9 Concluding Remarks

Any electronic device with only two electrical terminals is usually referred to in the semiconductor industry as a non-volatile resistance-switching memory device if the device can exhibit one of two resistance values over a sufficiently long time period, without consuming any power, and can be switched from a low-resistance state to a high-resistance state , and vice-versa, by applying either a short voltage pulse, or a short current pulse, of appropriate amplitude and polarity, across the two device terminals, and such that the resistance state at any time , either low or high, can be sensed by applying a relatively much smaller sensing voltage pulse , or current pulse, of some preset waveform, across the same terminals.

Implicit in the above definition is that at any time, the device can be modeled as a linear resistor obeying *Ohm’s Law*, when the sensing signal amplitude is sufficiently small, for otherwise, the word resistance would be meaningless. The *linearity* property implies that the sensing voltage, or current, and its corresponding voltage response, or current response, have identical waveforms, and have the same zero-crossings in time. It follows that the loci in the v–i plane during sensing when observed from an oscilloscope will appear as a short linear segment through the origin whose slope will be small if the resistance being sensed is low, or much larger, if the resistance being sensed is high. In other words, the two resistance states can be depicted as two short straight line segments of slopes R_1 and R_2, crossing each other at the origin of the v–i plane. These two segments can be emulated exactly by an *ideal* memristor having the memristance $R(q) = R_1$ at the origin, and $R(q) = R_2$ at another point , say $q = q_2$ of a smooth φ vs. q curve in the φ vs. q plane. By uncovering the physical operating mechanisms taking place internal to the device, one could construct a model that not only exhibits these two memristances, but also faithfully reproduces one or more pinched hysteresis loops, measured using different large-amplitude periodic signals [12]. The resulting mathematical expressions may be extremely complex, and may often be expressible only by implicit mathematical equations. Nevertheless, they would define a memristor of the generic form given by Eqs. (40a), (40b), by virtue of the characteristic property of the memristor.

The take-home lesson from this tutorial can be summarized succinctly as follow:

> Any 2-terminal electronic device devoid of internal power source and which is capable of switching between two resistances upon application of an appropriate voltage or current signal, and whose resistance state at any instant of time can be sensed by applying a relatively much smaller sensing signal, is a *memristor*, defined either by the ideal memristor equation, or by one of its unfolded siblings via Eqs. (40a), (40b).

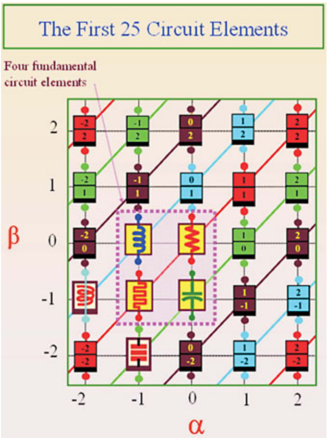

The First 25 Circuit Elements
Four fundamental circuit elements
β
2
1
0
-1
-2
-2
-1
0
1
2
α

Fig. 11 An enlargement of the first 25 axiomatically-defined circuit elements from the periodic table of circuit elements (Fig. 31 of [3]) where the 4 basic circuit elements (resistor, capacitor, inductors and memristor) are replaced by their symbols. Observe that since these two elements require a double time-integrals of voltage and current, their dynamics are of a higher order than those of the 4 basic circuit elements enclosed inside the *dotted red box*

Our final remark is concerned with the significance of the pinched hysteresis loop in the modeling of non-volatile resistance switching memories. Let us recall that while the *memristance vs. state map* tells us the complete set of *small-signal memristances* endowed upon a memristive device, it is rather difficult to measure them experimentally unless the memristor can be modeled by the ideal memristor equation $v = R(q)i$, where $dq/dt = i$. To extract such information from the generic memristor Eqs. (40a), (40b), we have to identify first the relevant state variable, or state variables in cases demanding a higher-order memristor state space. In contrast, the *chord memristances* associated with a pinched hysteresis loop can be readily extracted since it is simply the set of all slopes of a straight line anchored at the origin whose tips traces along the loci of a measured pinched hysteresis loop. Each such chord resistance is a true resistance indistinguishable from a linear resistor having the same resistance. The set of all such chord memristances associated with a pinched hysteresis loop therefore provides a subset of the memristor's endowed small-signal memristances. Since measuring pinched hysteresis loops associated with different periodic input voltage, or current, waveforms applied across a memristive device is a relatively simple task that could be automated,[13] it is a useful tool for uncovering a memristive device's nonlinear physical operating mechanisms, and for validating its memristive models. In the case of an *ideal memristor*, it is important to bear in mind that the *small-signal memristance*, and its corresponding *chord memristance*, represent exactly the same information. The main difference is that while the chord memristance is a long vector pinned at the origin of the v–i plane, its corresponding small-signal memristance is an infinitesimal tangent attached at each point on an ideal memristor's constitutive relation in the φ vs. q plane. It is also useful to note that unlike classical electronic circuit analysis, the small-signal memristor voltage associated with an applied small-signal memristor current represents the *actual total* solution, and is not superimposed upon some dc bias. We

[13] Measurement instrument companies could exploit the high market potentials of automated pinched-hysteresis-loop measuring instrumentations, and their memristance extractions.

end this tutorial with the following terse characterization of a resistance-switching memory device:[14]

> *"If it's pinched, it's a memristor."*

References

1. Chua, L.O.: IEEE Trans. Circuit Theory **CT-18**, 507 (1971)
2. Chua, L.O., Kang, S.M.: Proc. IEEE **64**, 209 (1976)
3. Chua, L.: Nonlinear circuit theory. In: Moschytz, G.S., Neirynck, J. (eds.) Modern Network Theory—An Introduction: Guest Lectures of the 1978 European Conference on Circuit Theory and Design, p. 81. PGeorgi, St. Saphorin (1978)
4. Chua, L.O.: Proc. IEEE **91**, 1830 (2003)
5. Jack, J.J.B., Noble, D., Tsien, R.W.: Electric Current Flow in Excitable Cells. Oxford University Press, Oxford (1975)
6. Itoh, M., Chua, L.: Int. J. Bifurc. Chaos **18**, 3183 (2008)
7. Diao, Z., Pakala, M., Panchula, A., Ding, Y., Apalkov, D., Wang, L.-C., Chen, E., Huai, Y.: J. Appl. Phys. **99**, 086510 (2006)
8. Duan, X., Huang, Y., Lieber, C.M.: Nano Lett. **2**, 487 (2002)
9. Bruce, J.W., Giblin, P.J.: Functions and Singularities. Cambridge University Press, Cambridge (1999)
10. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: Nature **45380** (2008)
11. Waser, R., Aono, M.: Nat. Mater. **6**, 833 (2007)
12. Chua, L.: IEEE Trans. Circuits Syst. **CAS-27**, 1014 (1980)
13. Chua, L.O.: Introduction to Memristors, IEEE Expert Now Education Course. IEEE Press, New York (2010)
14. Di Ventra, M., Pershin, Y.V., Chua, L.O.: Proc. IEEE **97**, 1717 (2009)

[14] Exactly the same theory of the memristor can be used to identify a memcapacitor (acronym for memory capacitor), and a meminductor (acronym for a memory inductor), from the table of axiomatically-defined circuit elements [3] and [4], as depicted in Fig. 11, and presented at the 2008 Berkeley Symposium on Memristors and Memristive Systems (Part 1, towards the end of the opening lecture) on Nov. 21, 2008 (proceedings of this symposium were videotaped and available via YouTube), as well as further elucidated in [13] and [14].

The Detectors Used in the First Radios Were Memristors

Gaurav Gandhi, Varun Aggarwal, and Leon O. Chua

Abstract The recent discovery of memristor has sparked renewed interest in the scientific community about state dependent resistances. In the current paper, we show that the detector used in the first radios, called cat's whisker, had memristive properties. We have identified the state variable governing the resistance state of the device and can program it to switch between multiple stable resistance states. Our observations are valid for a larger class of devices called coherers, including cat's whisker. We further argue that these constitute the missing canonical physical implementations for a memristor.

1 Introduction

There are two general types of components used in electronics: passive (e.g. resistors, capacitors and inductor) and active (e.g. transistors and integrated circuits). Passive components are incapable of power gain i.e. they cannot amplify signals while active circuit components can amplify signals. The three most commonly used passive components are resistor, capacitor and inductor. A simple wire acts as resistor, two parallel plates separated by a dielectric work as a capacitor and a coiled metallic wire functions as an inductor. No such implementation is there for a memristor.

G. Gandhi · V. Aggarwal
mLabs, New Delhi, India

G. Gandhi
e-mail: gaurav@mlabs.in

V. Aggarwal
e-mail: varun@mlabs.in

L.O. Chua (✉)
University of California, Berkeley, USA
e-mail: chua@eecs.berkeley.edu

L.O. Chua
Technische Universität München, Munich, Germany

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_4,

Table 1 Passive components

Component	Canonical implementation	Behavior
Resistor	Simple wire	Dissipates energy
Capacitor	Parallel plates	Store energy in electric field
Inductor	Coiled wire	Store energy in magnetic field
Memristor	**?**	State dependent resistor

Memristors are a new addition to this family. It was proposed by Leon Chua in 1971 [6] that there must be a fourth passive circuit element and it was discovered by HP Labs in 2008 [22].

A passive memristor is defined as any two-terminal electronic device that is devoid of an internal power-source and is capable of switching between two resistance states upon application of an appropriate voltage or current signal that can be sensed by applying a relatively much smaller sensing signal [8, 9, 11].[1] When one plots a curve of voltage versus current applied to the memristor, it shows pinched hysteresis loop [1]. Thus a pinched hysteresis loop in the voltage vs. current characteristics of the device serves as the fingerprint for memristor [1]. Despite the simplicity of symmetry argument that predicts the existence of memristor [6, 9], a simple macro memristor device, such as resistance wires, capacitor plates and inductor coils, does not exist [20] (refer Table 1). Current memristor implementations use specialized materials such as transition metal oxides, chalcogenides, perovskites, oxides with valence defects, or a combination of an inert and an electrochemically active electrode.

We look at devices invented in the quest to build a radio wave detector more than 100 years back, which ultimately culminated in the vacuum tube. Coherer, invented by Edouard Branly [12, 13, 19] in the 19th century, in many of its embodiments such as ball bearings, metallic filings (also referred to as granular media) in a tube or a point-contact, exhibits an initial high-resistance state and coheres to a low-resistance state on the arrival of radio waves. The device attains its original resistance state on being tapped mechanically. These were used as detector for wireless wave. The first electrically reset-able coherer, comprising a metal-mercury interface and named as an auto-coherer, [2–5] did not require tapping and resets automatically in the absence of radio waves.

Coherers and autocoherers worked unreliably for the purpose of radio wave detection, which actually required a diode for rectification. Cat’s whisker was the first metal-semiconductor point contact device patented by J.C. Bose and was actively used in early radio research [5, 14]. A cat’s whisker detector, also called a crystal detector, is an antique electronic component consisting of a thin metal wire that lightly

[1] In this paper, we are referring to memristor as defined in [9, 20], which includes the ideal memristor as defined in Chua’s 1971 paper [6] as a subclass. Throughout the paper, the reference is to memristor and not ideal memristor [1, 9].

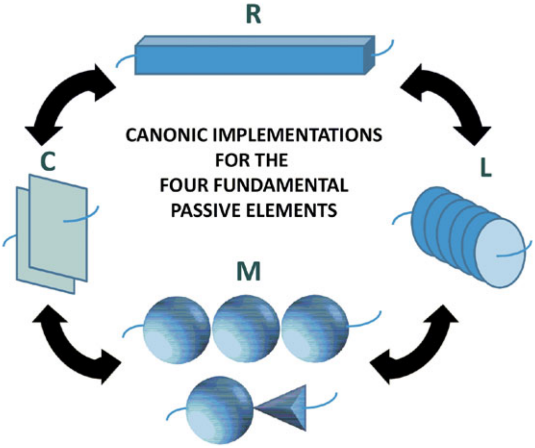


Fig. 1 Complete set of canonical discrete implementations of the four fundamental circuit elements

touches a crystal of semiconducting mineral (usually Galena) to make a crude point-contact rectifier (ref. Fig. 2). The symbol of diode is derived from this point-contact setup. The current flows from metal whisker to crystal but not the other way round making it function like a diode and hence a detector for wireless waves. Only certain sites (also referred as 'hot' spots) on the crystal function as a rectifying junction and are very sensitive to exact geometry and the pressure applied.

In the present work we have established that the crystal detector in the cat's whisker setup shows memristive properties. Our observations and results are valid for a larger class of devices (coherers) that also include the cat's whisker. These devices form a pinched hysteresis loop in their voltage-current plane, the fingerprint of a memristor, on the application of specific set of bipolar periodic inputs. The device can also be programmed in various resistance states, erased and read making it a suitable memristive device to study. Note that for convenience of exposition, this paper uses terms like "state", "state variable", and "average resistance" in an intuitive sense and they may not adhere to their definition as in literature in the strictest sense.

The discovery of memristive phenomenon in a device as simple and pervasive as cat's whisker provides students with a simple, inexpensive and easy to use memristor for research, experimentation and building applications. The lack of availability of such a simple discrete memristor has been an impediment in the research being pursued by many in the field of memristor.

The existence of the memristive phenomena in the larger class of devices, coherers, whose signature is an imperfect metal-metal contact show the ubiquity of the memristive phenomena. It shows that memristor phenomena is not limited to specific materials assembled at small geometries, but is present in a large class of metals put together as a point contact [15, 16, 18]. We believe that this constitutes the canonical "macro" implementation for a memristor (ref. Fig. 1).

This paper is organized as follows: The next section discusses about the cat's whisker setup used for studying the memristive properties of the device and the terminology used to explain the observations. It is followed by a section, which

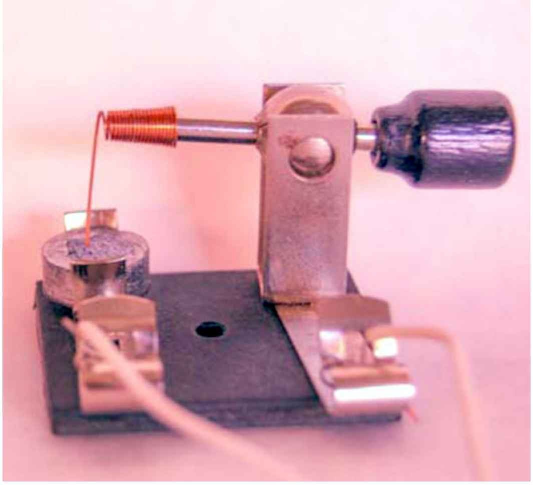


Fig. 2 Philmore Cat's Whisker in contact with a Galena Crystal

describes in detail the electrical properties of the cat's whisker setup and its behavior under different electrical stimulations. Based on the observed behavior, we postulate an electrical model and identify the state-variable controlling the resistance change. The paper concludes with a discussion on the observations and results.

2 Cat's Whisker Detector Setup

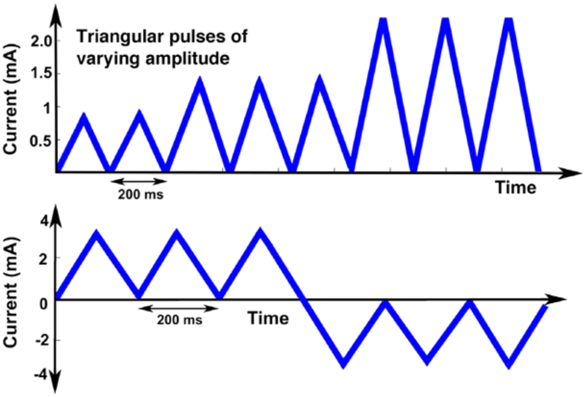
We used a setup comprising of a Galena crystal in contact with a Phosphorous bronze wire, also famously known as Philmore's 7003 cat's whisker setup, which was actively used in the past.[2] It consists of fixed whisker (phosphorous bronze) to control the pressure and contact area between the Galena crystal and the wire, which is a critical parameter to be setup for the device to work as a diode or memristor (refer Fig. 2).

For our experiments, we provided the Cat's whisker setup with different rising triangular-shaped current mode inputs (ref. Fig. 3) and recorded the voltage across the setup. We built a programmable system where we could digitally input the amplitude and time period properties for the input waveform, generate the input, apply it to our setup and automatically record the output voltage waveform.

This was achieved through the setup described in Fig. 4. The waveform specifications were fed into a micro-controller based system, which created a digital voltage waveform having the prescribed properties. This was passed to a digital-to-analog converter (DAC) to generate an analog voltage signal followed by a level shifter (such that negative voltage values can be generated). This voltage mode signal was

[2] Philmore 7003 cat's whisker is easily available from various antique radio or radio hobbyists shops and can also be ordered online through sites like eBay. On the other hand, it is easy to build it oneself using one of the semiconducting crystals and a metallic wire.

Fig. 3 Current input used to unravel various memristive behaviors. (**a**) Triangular pulse with rising amplitude used to unravel the multi-state resistive behavior of cat's whisker. (**b**) Bipolar input provided to demonstrate the memristive behavior



converted to a current mode signal using an AD844 current conveyor chip. The AD844 chip was connected in the configuration shown in Fig. 4. Finally, the current mode signal from the chip was fed to the phosphorous bronze metal of the Cat's whisker setup with the other terminal being grounded. The voltage across the Cat's whisker setup was captured and stored back in the micro controller.

The setup was activated by different current-mode input signals and its transient behavior was recorded. Here, we report the results for two representative input waveforms. In the first, the current was varied from 1 mA upto 2.5 mA as rising triangular bursts as shown in Fig. 3(a). In the second, repeated bursts of fixed amplitude triangular waveforms (amplitude of 3.5 mA) in the positive and negative direction were given. This is as shown in Fig. 3(b). For all our inputs, the time period of one triangular waveform is 200 ms to ensure a slowly varying signal.

We found that the devices exhibit three distinct behaviors: cohering action, multistable memristive behavior, and bistable resistive random access memory (RAM) type behavior apart from the usual rectifying behavior.

These results are for Galena crystal in contact with a phosphorous bronze wire. Similar qualitative results were observed for a wide class of crystals like carborundum, iron pyrite, etc. and metals including iron filing, aluminum, Ni, etc. [15]. The memristive properties are also observed if the semiconducting metal is replaced by a metal, say iron. Such a metal-metal setup comprised the coherer.

3 Terminology

The behavior of our device is non-linear in the V–I plane. In such case, the term resistance is only loosely defined since the value of resistance changes at each point on the V–I curve. To dispel any ambiguity, we define the following terms to explain our results. We define *Non-linear DC resistance, $R(I)$*, as the resistance at a given DC current input calculated as the ratio of the DC voltage and DC current at that point, assuming the steady-state voltage response is DC. As shown in Fig. 5(a), this is the slope of the line which joins the given point to origin [7].

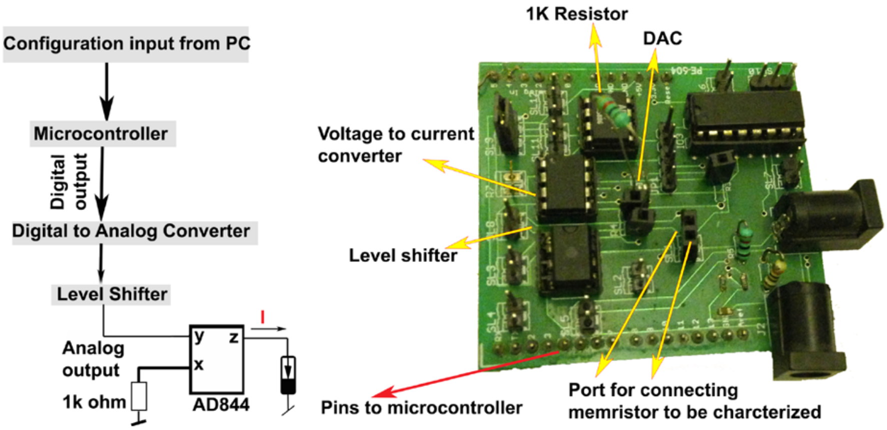

Configuration input from PC
Microcontroller
Digital output
Digital to Analog Converter
Level Shifter
Analog output
1k ohm
y
z
x
AD844
I
1K Resistor
DAC
Voltage to current converter
Level shifter
Pins to microcontroller
Port for connecting memristor to be charcterized

Fig. 4 Setup used to generate the input current waveforms described in Fig. 3

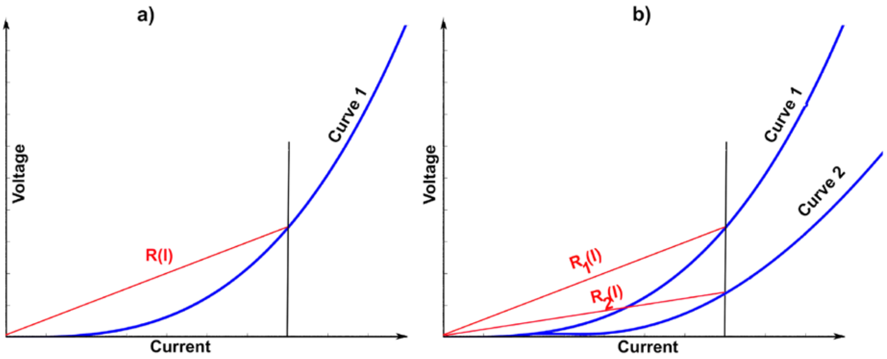


Fig. 5 Non-linear DC Resistance R(I). For a *non-volatile* memristor, the DC V–I curve must be a *multi-valued* function of I which nevertheless must pass through the origin. Here, we assume a hypothetical case where the DC V–I curve is *double-valued* and can be modeled by 2 curves, as depicted in (**b**)

Whenever we refer to an increase or decrease in Non-linear DC resistance, we imply its change as a function of the DC input current values. This is depicted in Fig. 5(b). Curve 2 has a lower Non-linear DC resistance than curve 1, since the Non-linear DC resistance is either equal or less at all points in the given current range.

In the "dual case" one can also define Non-linear DC conductance, $G(V)$, as the conductance at a given voltage input calculated as the ratio of the DC current and DC voltage at that point. Whenever we refer to an increase or decrease in Non-linear conductance, we imply its change across present values. This is depicted in Fig. 6(b). Curve 2 has a lower non-linear DC conductance than curve 1. In this paper we will be focusing on the Non-linear DC resistance, $R(I)$, terminology since our input signal is current.

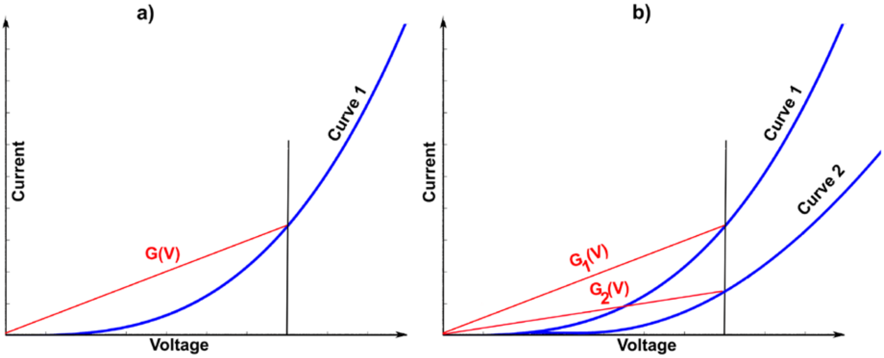


Fig. 6 Non-linear DC Conductance $G(V)$

We define *average resistance* as the mean of the Non-linear DC resistance over a range of input currents. The idea of average resistance is similar to what is loosely stated in literature as the resistance of a non-linear system. For instance, when it is stated that the resistance of diode is lower for positive input and higher for negative input, it actually refers to average resistance. To be strictly correct, we would use the new average resistance definition to explain our results. Whenever we provide the quantitative value of average resistance, we shall state the range of currents it was averaged upon.

4 Experimental Results

The device can be programmed into two possible states: firstly, touching the wire on the "hot" spot[3] on the crystal leads to rectification and the other, where the device is touched at places other than hotspots. The following observations are more dominant when the device is set at non-hotspots. The memristor behavior becomes feeble (but not non-existent) at hotspots where rectification takes place. This could be one reason why the memristive properties of cat's whiskers were not discovered to date.

4.1 Cohering Action

For any input current leading to a voltage below a specific threshold voltage, V_{th}, the devices exhibit a high non-linear DC resistance.

[3] Hotspots are found by the following process. The device is either connected to an oscilloscope or a radio receiver circuit. Then the point of contact between the crystal and wire is changed till a diode-like characteristics is observed. The point(s) on the crystal where rectification is observed are hotspots. Finding hotspots is generally a time consuming task for non-experts [14].

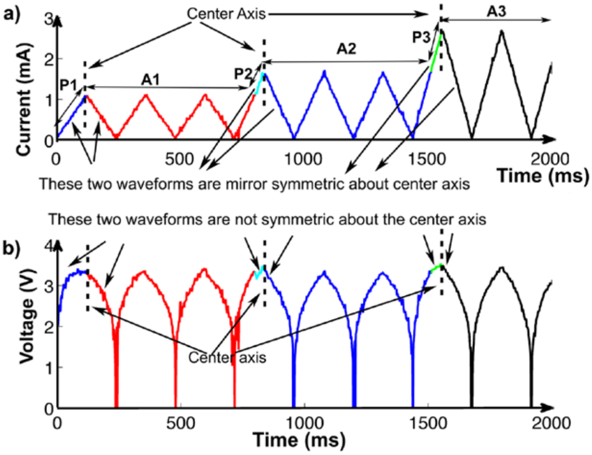


Fig. 7 *Device input current and output voltage*. (**a**) Input Current vs. Time and (**b**) Voltage across memristor vs. Time. It is observed that the maximum voltage across the device does not cross a threshold voltage, V_{th}. Here *A1*, *A2*, *A3* and *P1*, *P2* and *P3* denotes the time interval cited in the text. It is color coded to demonstrate the Multistable Memristive behavior of the device

At a current higher than I_{th}, corresponding to a voltage V_{th}, the average resistance of the device falls (refer P1 transition in Fig. 8). In our experiment, I_{th} had a value of 0.3 mA, whereas V_{th} was around 3.2 V.[4] Once the device takes this new state, it maintains the said non-linear DC resistance (Curve A1) on excitation by current values below I_{th} as well.

4.2 Multistable Memristive Behavior

Once in cohered state, the device exhibits a state-dependent non-linear DC resistance, the state variable being the maximum current ($I_{\max}$), i.e. $R_t(I) = f([I_{max}]_{0-t})$. Here $f([I_{max}]_{0-t})$ is defined as the maximum current that the device has experienced in the time period 0 to t.

As the device is exposed to pulses of subsequently larger peak current (refer Figs. 7 and 8) which shows the behavior for different time intervals, P1 to P3 and A1 to A3),[5] it sets itself to new lower non-linear DC resistance values. The resistance curve remains non-linear, nonetheless. The maximum voltage across the device remains practically constant at V_{th}. This behavior is akin to that of a diode, but unlike a diode the device remembers its changed resistance when taken to lower voltage

[4] These values may change according to the metal, the contact, pressure, etc.

[5] Note that the non-linear DC resistance changes appreciably only when the maximum current through the device has changed. This can be seen through color correspondence, where each color shows a new stable non-linear DC resistance-state and the transitions are marked by the first pulse of higher amplitude: P1, P2 and P3 being the time interval where these pulses are applied. In case the maximum current passed through the device does not change, the non-linear DC resistance feebly oscillates around the same value, as seen in the time-interval of A1, A2 and A3. Furthermore, we have observed that the non-linear DC resistance remains fixed even when the amplitude of the pulse is decreased, since the maximum current has not changed.

Fig. 8 *Device behavior as a state-dependent resistance: Current-Voltage Plot*. It is observed that the maximum voltage across the device does not cross a threshold voltage, V_{th}

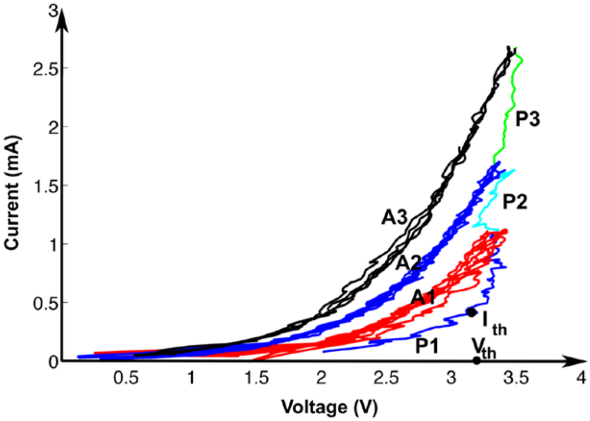


Table 2 Average resistance and nonlinear DC resistance for various regions in Multistable Memristive behavior of Cat's whisker

Region	Average resistance R	Nonlinear DC resistance ($I = 0.5$ mA)
A1	9.2 K	6 K
A2	7.3 K	4.6 K
A3	6.3 K	4.2 K

levels. For input current pulses of same or lower amplitude than the maximum current experienced, the device shows hysteresis loops around the already-achieved resistance value, with small oscillations.

The average resistance values and the non-linear DC resistance (at $I = 0.5$ mA) for curves A1, A2 and A3 are presented in Table 2. The average resistance values have been averaged over a current range of $I = 0.1$ mA to 0.5 mA. One clearly observes here that both these values decrease with changing I_{max}. We stress that our terminology of "average resistance" is not a new concept or phenomenon. It is introduced for the convenience of discussion.

4.3 Bistable Resistive RAM /Memristive Mode

When activated by any bipolar current input (i.e. current takes both positive and negative values in each cycle), the device gets programmed into one state in the positive cycle, and a different state in the negative cycle (refer Fig. 9).[6] It keeps oscillating between these two stable states, forming the famous eight-shaped pinched

[6] Whereas triangle current pulse is symmetrical with respect to center axis shown in Fig. 9, voltage response during that time interval is not symmetrical. This causes hysteresis loop. We refer to different stable V–I characteristics as different states of the device.

Fig. 9 *Bistable memristive behavior* (**a**) Input current vs. time, (**b**) Voltage across device versus time

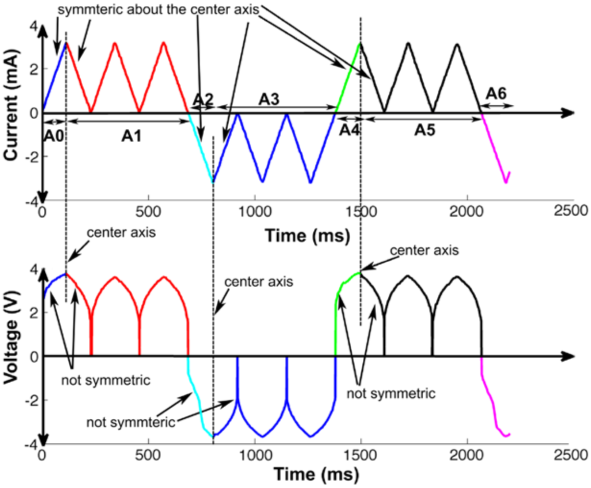
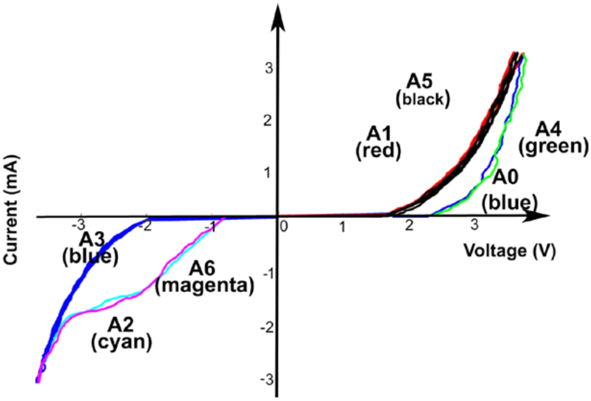


Fig. 10 *Bistable memristive behavior* Input current vs Voltage across device. One clearly observes pinched hysteresis loop for cat's whisker



hysteresis loop in its V–I characteristics (refer Fig. 10).[7] Figure 11(a)–(d) shows the complete cyclic behavior of pinched hysteresis loop. It has been established that "If it is pinched, it is memristor". Pinched hysteresis loop is the fingerprint of a memristor [8, 10].

The average resistance values and the non-linear DC resistance (at current magnitude of 0.5 mA) for curves A1... A6 are presented in Table 3. These values have been averaged over a current magnitude range of $I = 0.1$ mA to 3.5 mA.

Our observations further imply that the average resistance of the device is a function of the magnitude of I_{max} for either directions of current, but with a quantitatively different state-map. This can be mathematically stated as:

[7] It is evident by looking at regions depicted by A0 to A6 that the change in resistance happens at the first pulse of the transition. One may also note that these observations show recovery of resistance to a higher non-linear DC resistance: A4 resistance is higher than A5 resistance.

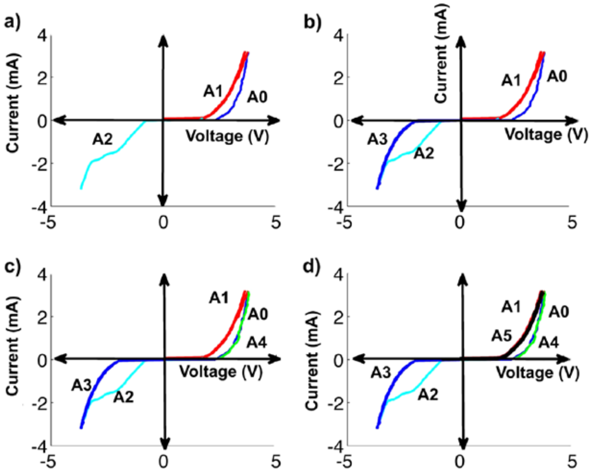


Fig. 11 *Bistable memristive behavior* Pinched Hysteresis Loop. (**a**) The plot between voltage and current when during *A0–A2* time interval. (**b**) The plot between voltage and current when during *A0–A3* time interval. Here *A0*, *A1* and *A2* in (**a**) and (**b**) are identical. It can be easily seen that pinched hysteresis loop is being formed. (**c**) Pinched hysteresis loop during *A0–A4* time interval. (**d**) Pinched hysteresis loop during *A0–A5* time interval. Note that (**c**) and (**d**) are identical since *A5* and *A1* are identical. These results clearly establish the memristive behavior of Cat's whisker

Table 3 Average resistance and nonlinear DC resistance for various regions in Bistable Memristive behavior of Cat's whisker

Region	Average resistance R	Nonlinear DC resistance ($\|I\| = 0.5$ mA)
A0	3.2 K	6 K
A1	3.6 K	4.6 K
A2	1.7 K	2.6 K
A3	3.0 K	5.2 K
A4	3.4 K	6.0 K
A5	2.8 K	4.6 K
A6	1.6 K	2.3 K

Let

$$R_{p1} = f\big(\text{magnitude}\big([I_{\max+}]_{0-t}\big)\big) = I_1, \tag{1}$$

$$R_{n1} = f\big(\text{magnitude}\big([I_{\max-}]_{0-t}\big)\big) = I_1, \quad \Rightarrow \tag{2}$$

$$R_{p1} \neq R_{n1} \tag{3}$$

where R_{p1} is the average resistance of the device when activated by a maximum current of I_1 in positive direction, and R_{n1} is the average resistance when activated by a maximum current of I_1 in the negative direction. $f(\text{magnitude}([I_{\max+}]_{0-t}))$

implies the maximum current the device has experienced between time = 0 and time = t.

A detailed circuit theoretic model for the non-linear DC resistance for the said memristor can also be achieved using the scheme presented recently by Chua [9]. Such a model is beyond the scope of the present discussion and will be treated in a separate publication.

5 Discussion

We show that devices as simple as metal-metal or metal-semiconductor point-contact, as in the famously known setup of coherer or cat's whisker, shows memristive properties. The famous pinched hysteresis curve between the voltage-current characteristics, which is the fingerprint of a memristor, is visible in the current set of experiments. We have demonstrated that by using various stimuli with different maximum amplitudes on either side, the device can be programmed to function in multiple stable resistance-states. When used as a resistive memory, the memory can be read in the memory state by providing an excitation of small amplitude (refer Sect. 4.1). This fulfills the conditions of Chua's definition of memristor [8], and qualifies the century-old cat's whisker (more generally coherer) as a canonical implementation of a memristor. Cat's whisker intrigued the science of that era as much as memristor is exciting the scientists of the present day [21].

There are certain differences between the behavior of our memristor and other present day memristors. Unlike Williams et al. memristor [22], they do not behave as a charge-flux based memristor. Irrespective of the increase or decrease of flux, their resistance does not change till the maximum current or current polarity changes. Of various memristors currently being studied, our devices, especially coherers, have similarities in behavior [16, 18] and construction [17]. However, none of these recent memristors have reported dependence on I_{max}.

Also, the point worth noting is that the observations are more dominant when the device is configured in the non-rectification mode. The memristor behavior becomes feeble (but not always non-existent) at hotspots where rectification takes place. As the main application of cat's whisker has been to rectify the signal and it shows feeble memristive behavior at that junction, this could be one reason why the memristive properties of Cat's whiskers were not discovered to date. Even though the memristor has only a feeble effect in optimal radio operation, one must note that the composite "cat whisker-crystal" 2-terminal device is a memristor because its internal physical mechanism, though not well-understood, is certainly "state-dependent", and hence is memristive [8].

By demonstrating the memristive properties of cat's whisker the present work not only fills an important gap in the study of switching devices, but also brings them into the realm of immediate practical use and implementation. Understanding the physical mechanism of the memristive behavior in this new class of device serves as an interesting case study for scientists.

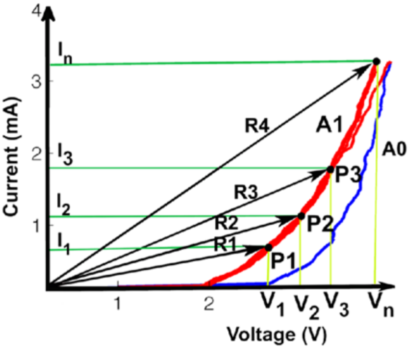


Fig. 12 *Calculating Average Resistance and Non-linear DC Resistance.* We stress here that we introduce the names "average resistance" and "states" merely to improve the clarity of our exposition. They are not new technical names or concepts

Acknowledgements This paper is supported in part by AFOSR grant no. FA9550-10-1-0290 and mLabs.

Appendix

5.1 Detailed Method to Find Average Resistance

Refer to Fig. 12 (a reproduction of Fig. 10). Here we will describe how to calculate the average resistance for the given curve during the time interval A1 for current range I_1 (0.1 mA) to I_n (3.5 mA).

For input current I_1, at point P_1, the voltage is V_1. Here, the non-linear DC resistance is r_1 (V_1/I_1), the inverse of the slope of the line segment joining P_1 to origin. Similarly, at current In, the non-linear DC resistance is r_n. For any current between I_1 and I_n, the non-linear DC resistance is similarly defined, as shown in the figure.

To calculate the average resistance, the non-linear DC resistance is measured at regular time interval starting from point P_1 to P_n. The average of these values is the average resistance. Mathematically this is equivalent to:

$$R_{avg} = \sum_{i=1}^{i=n} R_i/n, \quad \text{where } R_i = V_i(t = t_i)/I_i(t = t_i),$$

here $t_i = t_{p1} + \text{time interval} * (i - 1)$.

References

1. Adhikari, S.P., Sah, M.P., Kim, H., Chua, L.O.: IEEE Trans. Circuits Syst. I (2013). doi: 10.1109/TCSI.2013.2256171
2. Bondyopadhyay, P.: Proc. IEEE **86**(1), 259 (1998)
3. Bose, J.: Proc. R. Soc. Lond. **65**(413–422), 166 (1899)

4. Bose, J.: Originally presented to the British Association at Glasgow, September 1901
5. Bose, J.: Patent USA **755**, 840 (1904)
6. Chua, L.: IEEE Trans. Circuit Theory **18**(5), 507 (1971)
7. Chua, L.: Proc. IEEE **91**(11), 1830 (2003)
8. Chua, L.: Appl. Phys. A, Mater. Sci. Process. **102**(4), 765 (2011)
9. Chua, L.O.: Proc. IEEE **100**(6), 1920 (2012)
10. Chua, L.: Nanotechnology **24**(38), 3001 (2013)
11. Chua, L., Kang, S.: Proc. IEEE **64**(2), 209 (1976)
12. Dilhac, J.: IEEE Commun. Mag. **47**(9), 20 (2009)
13. Falcon, E., Castaing, B.: Invest. Cienc. **404**, 80 (2010)
14. Fleming, J.: In: The Principles of Electric Wave Telegraphy, New York (1908)
15. Gandhi, G., Aggarwal, V.: mLabs report (2010). www.mlabs.in
16. Jo, S., Lu, W.: Nano Lett. **8**(2), 392 (2008)
17. Kim, T., Jang, E., Lee, N., Choi, D., Lee, K., Jang, J., Choi, J., Moon, S., Cheon, J.: Nano Lett. **9**(6), 2229 (2009)
18. Kim, K., Jo, S., Gaba, S., Lu, W.: Appl. Phys. Lett. **96** (2010)
19. Lodge, O.: The Electrician **40**, 86 (1897)
20. Pershin, Y., Ventra, M.: IEEE Circuits Syst. Mag. **12**(1), 64 (2011)
21. Prodromakis, T., Toumazou, C., Chua, L.: Nat. Mater. **11**(6), 478 (2012)
22. Strukov, D., Snider, G., Stewart, D., Williams, R.: Nature **453**(7191), 80 (2008)

Memristor, Hodgkin-Huxley, and Edge of Chaos

Leon Chua

Abstract From a pedagogical point of view, the memristor is defined in this tutorial as any 2-terminal device obeying a *state-dependent Ohm's law*. This tutorial also shows that from an experimental point of view, the memristor can be defined as any 2-terminal device that exhibits the *fingerprints* of *"pinched" hysteresis loops* in the v–i plane. It also shows that memristors endowed with a continuum of equilibrium states can be used as non-volatile analog memories. This tutorial shows that memristors span a much broader vista of complex phenomena and potential applications in many fields, including neurobiology. In particular, this tutorial presents toy memristors that can mimic the classic habituation and LTP learning phenomena. It also shows that sodium and potassium ion-channel memristors are the key to generating the action potential in the Hodgkin-Huxley equations, and that they are the key to resolving several unresolved anomalies associated with the Hodgkin-Huxley equations. This tutorial ends with an amazing new result derived from the new *principle of local activity*, which uncovers a minuscule life-enabling *Goldilocks zone*, dubbed the *edge of chaos*, where complex phenomena, including creativity and intelligence, may emerge. From an information processing perspective, this tutorial shows that synapses are locally-passive memristors, and that neurons are made of locally-active memristors.

1 Introduction

Memristor is a 2-terminal electrical circuit element that has attracted immense worldwide interests from both industry and academia ever since an operational device was reported in 2008 [1] by a team of scientists from the *hp* Information and Quantum Systems Lab, headed by Stanley Williams. An acronym for *memory resis-*

L. Chua (✉)
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley
CA 94720 USA
e-mail: chua@eecs.berkeley.edu

L. Chua
Technische Universität München, Munich, Germany

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_5,

tor, the memristor was postulated in 1971 [2] but has received little attention even though a working device made from op amps and discrete nonlinear resistors had been built and demonstrated in [2], because it is bulky and requires a power supply. In contrast, the *hp* memristor is a scalable *passive nano* device that can be used as a *non-volatile memory* that could someday replace flash memories and DRAMs.

From a circuit-theoretic perspective, the memristor can be derived from an *axiomatic approach* and considered logically as the *fourth* basic circuit element [3–6].

In addition to serving as nonvolatile memories, memristors are in fact the right components for building brain-like machines. Our goal in this *tutorial* is to explain and illustrate, via memristor *toy* examples, how *synapses* are memristors, and why the classic Hodgkin-Huxley axon circuit model is made of memristors.

2 Definition, Symbol, and Fingerprints

2.1 Memristor Is Defined by a State-Dependent Ohm's Law

For pedagogical reasons, any 2-terminal black box (Fig. 1a) with 2 electrical terminals is called a *memristor* if it obeys the following state-*dependent* Ohm's law

Current-Controlled state-dependent Ohm's Law:

$$v = R(\mathbf{x})i \tag{1a}$$

State Equations:

$$\frac{d\mathbf{x}}{dt} = \mathrm{f}(\mathbf{x}, i) \tag{1b}$$

if the current i is the input, or

Voltage-Controlled state-dependent Ohm's Law:

$$i = G(\mathbf{x})v \tag{2a}$$

State Equations:

$$\frac{d\mathbf{x}}{dt} = \mathbf{g}(\mathbf{x}, v) \tag{2b}$$

if the voltage v is the input.

The scalars $R(\mathbf{x})$ in (1a) and $G(\mathbf{x})$ in (2a) are called the *memristance* (acronym for memory resistance) and the *memductance* (acronym for memory conductance), respectively. The unit of the memristance $R(\mathbf{x})$ is the Ohm (Ω). The unit of the memductance $G(\mathbf{x})$ is the Siemens (S).

The *state vector* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ has $n \geq 1$ components $x_1, x_2, \ldots, x_n$, called *state variables*. They represent *internal* physical parameters, such as *temperature, pressure, impurity concentration, chemical moiety*, etc., and may *not* depend on any *external* variables, such as the voltage or current associated with another device.

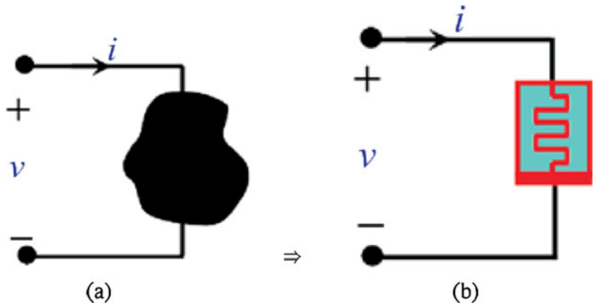


Fig. 1 (**a**) A 2-terminal device with external voltage *v* and external current *i* defined with the *associated reference* voltage polarity and current direction (**b**) symbol of memristor

To avoid pathological situations, we assume $R(\mathbf{x})$ and $G(\mathbf{x})$ are *piecewise-continuous* functions of $\mathbf{x}$ [7, 8], and that $\mathbf{f}(\mathbf{x}, i)$ and $\mathbf{g}(\mathbf{x}, v)$ are Lipschitz functions [7, 8].

2.1.1 Example 1: Discharge Tube [9]

Consider the following equations of a *current-controlled memristor* describing a *discharge tube* in [9, 10]:

State-dependent Ohm's law:

$$v = \left(\frac{F}{n}\right) i \tag{3a}$$

State Equation:

$$\frac{dn}{dt} = -\beta n + \alpha \left(\frac{F}{n}\right) i^2 \tag{3b}$$

where α, β and F are device constants depending on the dimensions of the tube and gas fillings, and n is the internal *state variable*. Here we can identify the *memristance*

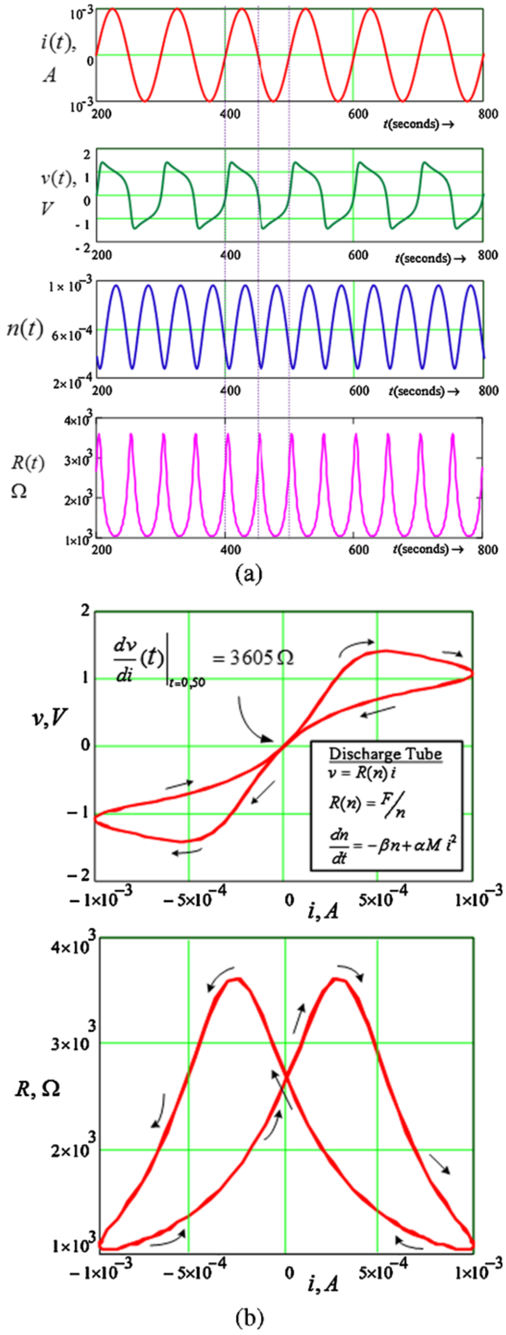
$$R(n) = \frac{F}{n} \tag{3c}$$

Figure 2a shows the steady-state (periodic) voltage solution $v(t)$ of (3b) calculated in [11] with the input current $i(t) = I \sin(\omega t)$, where $I = 1$ mA, along with the state variable $n(t)$ and the *time-varying* memristance $R(t)$. The loci plotted in the v *versus* i plane in Fig. 2b consists of a double-valued hysteresis loop which passes through *the origin*, henceforth called a *pinched hysteresis loop*. The *memristance R versus i* loci is also a doubled-valued hysteresis loop (Fig. 2b).

2.1.2 Example 2: Thermistor [12]

Consider the following equations of a *voltage-controlled memristor* describing a *thermistor* in [10, 12]:

Fig. 2 (**a**) Waveforms of input current $i(t) = I\sin(\omega t)$, voltage $v(t)$, state variable $n(t)$, and memristance $R(t)$ of the discharge tube and (**b**) pinched hysteresis loop plotted in the v *versus* i plane and the corresponding double-valued R *versus* i memristance hysteresis loop. The parameters used for simulations are $\alpha = 0.1$, $F = 1$, $B = 0.1$, $\omega = 0.063$ rad/s and $I = 1$ mA



(a)

(b)

State-dependent Ohm's law:

$$i = \left(R_0 e^{\beta(\frac{1}{T} - \frac{1}{T_0})}\right)^{-1} v \tag{4a}$$

State Equation:

$$\frac{dT}{dt} = \frac{\delta}{C}(T_0 - T) + \frac{G}{C}v^2 \tag{4b}$$

where T is the state variable (absolute temperature), R_0 is the resistance at the ambient temperature T_0, C is the heat capacitance, δ is the dissipation constant, and β is a material specific constant. Here we can identify the *memductance*

$$G(T) = \left(R_0 e^{\beta(\frac{1}{T} - \frac{1}{T_0})}\right)^{-1} \tag{4c}$$

Figure 3a shows the steady-state (periodic) current solution $i(t)$ of (4b) calculated in [11] with the input voltage $v(t) = E\sin(\omega t)$, where $E = 1$ V, along with the state variable $T(t)$ and the *time-varying memductance* $G(t)$. The loci plotted in the i *versus* v plane in Fig. 3b consists of a double-valued *pinched* hysteresis loop. The memductance G *versus* v loci is also a double-valued hysteresis loop (Fig. 3b).

2.2 *If It's Pinched It's a Memristor*

The preceding examples are illustration of the following *generic memristor characteristics*:

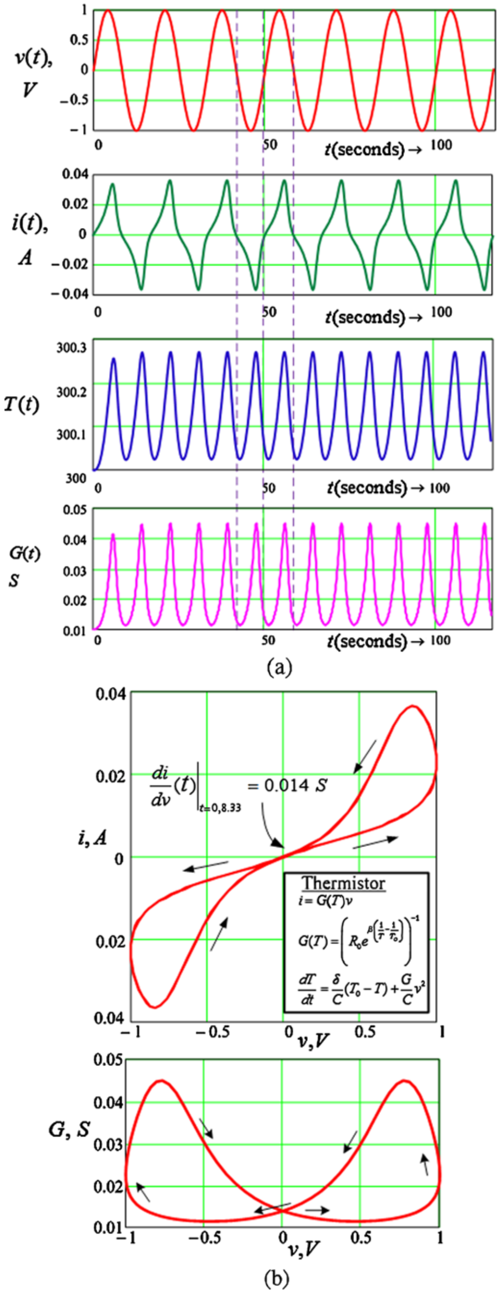
2.2.1 Memristor Fingerprints [11]

The v–i loci of a *memristor* driven by any periodic current source, or voltage source, which spans both positive and negative values over each period, is always a *pinched hysteresis loop* passing through the origin. The area bounding each sub-loop deforms as the frequency, or the amplitude, of the input waveform changes. In particular, at DC ($\omega \to 0$) the v–i loci shrinks to a *DC V–I characteristic*, which may be multi-valued, but always includes the origin $(v, i) = (0, 0)$. Likewise, as the frequency ω increases the lobe area shrinks to a linear function.[1]

It is important to understand that the above characterization of the memristor is *not a model* because its associated pinched hysteresis loops are not unique but vary with the input waveforms, as well as the amplitude and frequency. The significance of the above characterization is that it identifies any device that exhibits a *pinched hysteresis loop* over any bipolar periodic testing input signals as a memristor! Indeed

[1] For a generalized memristor defined by $v = R(\mathbf{x}, i)i$, or $i = G(\mathbf{x}, v)v$, the v–i loci tends to a *single-valued nonlinear* function as $\omega \to \infty$ [13].

Fig. 3 (**a**) Waveforms of input voltage $v(t) = E\sin(\omega t)$, current $i(t)$, state variable (temperature) $T(t)$, and memductance $G(t)$ of a thermistor (**b**) pinched hysteresis loop plotted in the i *versus* v plane and the corresponding double-valued G *versus* v memductance hysteretic loop. The parameters used for simulations are $R_0 = 100$, $T_0 = 300$, $C = 0.1$, $\delta = 0.1$, $\beta = 5 \times 10^5$ and $\omega = 0.377$ rad/s



many such pinched hysteresis loops had been published in numerous journals during the past decades and have been misidentified as something else. We now know they are memristors [14].

2.3 Ideal Memristor

Consider the *ideal* case where $\mathbf{f}(\mathbf{x}, i) = i$ and $\mathbf{g}(\mathbf{x}, v) = v$ in the state equations (1b) and (2b), respectively; namely,

$$\frac{dx}{dt} = i \tag{1b$'$}$$

$$\frac{dx}{dt} = v \tag{2b$'$}$$

Integrating both sides of (1b′) and (2b′), respectively, we obtain

$$x(t) = \int_{-\infty}^{t} i(\tau)d\tau \triangleq q(t) \tag{5a}$$

$$x(t) = \int_{-\infty}^{t} v(\tau)d\tau \triangleq \varphi(t) \tag{5b}$$

Substituting (5a) and (5b) for x in (1a) and (2a), respectively and integrating both sides, we obtain

$$\begin{aligned}\varphi(t) &\triangleq \int_{-\infty}^{t} v(\tau)d\tau = \int_{-\infty}^{t} R\big(q(\tau)\big)\frac{dq(\tau)}{d\tau}d\tau \\ &= \int_{q(-\infty)}^{q(t)} R(q)dq \triangleq \hat{\varphi}\big(q(t)\big)\end{aligned} \tag{6a}$$

$$\begin{aligned}q(t) &\triangleq \int_{-\infty}^{t} i(\tau)d\tau = \int_{-\infty}^{t} G\big(\varphi(\tau)\big)\frac{d\varphi(\tau)}{d\tau}d\tau \\ &= \int_{\varphi(-\infty)}^{\varphi(t)} G(\varphi)d\varphi \triangleq \hat{q}\big(\varphi(t)\big)\end{aligned} \tag{6b}$$

Equations (6a) and (6b) show that in the degenerate special scalar case defined by (1b′) and (2b′), the two equations (1a) and (1b) (resp., (2a) and (2b)) defining a current-controlled (resp., voltage-controlled) memristor are equivalent to a single equation:

$$\varphi = \hat{\varphi}(q) \tag{7a}$$

$$q = \hat{q}(\varphi) \tag{7b}$$

Equations (7a), (7b) are precisely the 4th *constitutive relationship* shown in Fig. 4 defining, via an *axiomatic* approach, the fourth circuit element dubbed the memristor.

We have adopted the symbols "q" and "φ" in (5a), (5b) and dubbed them "*charge*" and "*flux*", respectively, in keeping with tradition. In fact, they are mere

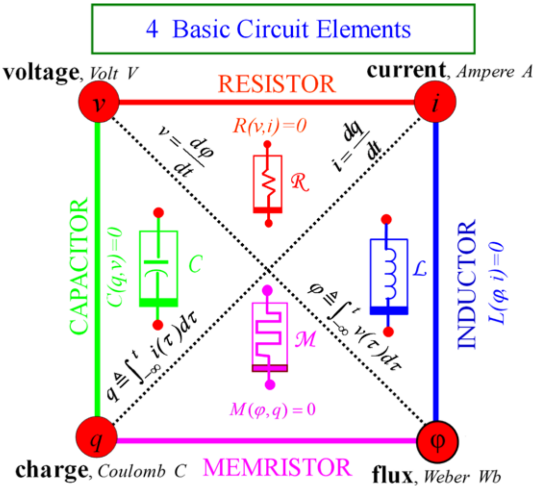


Fig. 4 The ideal memristor is defined by a constitutive relation between the flux $\varphi(t) \triangleq \int_{-\infty}^{t} v(\tau)d\tau$ and the charge $q(t) \triangleq \int_{-\infty}^{t} i(\tau)d\tau$

symbols and names defined *axiomatically* that need not have any physical significance.

Since real-world physical devices obeying the ideal constitutive relation (7a) or (7b) are rather rare, we believe that most memristive devices [10] will be modeled by either (1a) and (1b), or (2a) and (2b). For economy and efficiency of terminology, we will henceforth refer to all such devices as *memristors*,[2] and call the 4th circuit element defined in Fig. 4 as an *ideal memristor* on the rare occasions where a distinction is needed.

3 When Is a Memristor Non-volatile?

Most memristors are volatile in the sense that after the power is turn-off, the device reverts to a fixed memristance (resp., memductance) state. However, an *ideal* memristor is *non-volatile* and remembers its most recent memristance (resp., memductance) state when the power is shut off. To understand this *non-volatile* memory mechanism, consider a typical *ideal memristor* characterized by the $q = \hat{q}(\varphi)$ curve shown in Fig. 5. Let us assume, without loss of generality, that $\varphi(t_0) \triangleq \varphi_0 = 0$ at $t = t_0$ so that the memristor is operating at $q = 0$ at time $t = t_0$when a Δ-second voltage pulse $v_s(t)$ is applied across the memristor. The current $i(t)$ entering the memristor corresponding to the square voltage pulse shown in the upper left corner

[2] We will use the term *generalized memristor* to refer to the most general case defined in [10], where the *memristance* (resp., *memductance*) $R(\mathbf{x}, i, t)$ (resp., $G(\mathbf{x}, v, t)$) may depend also on the *input* current i (resp., voltage v) and the *time* t.

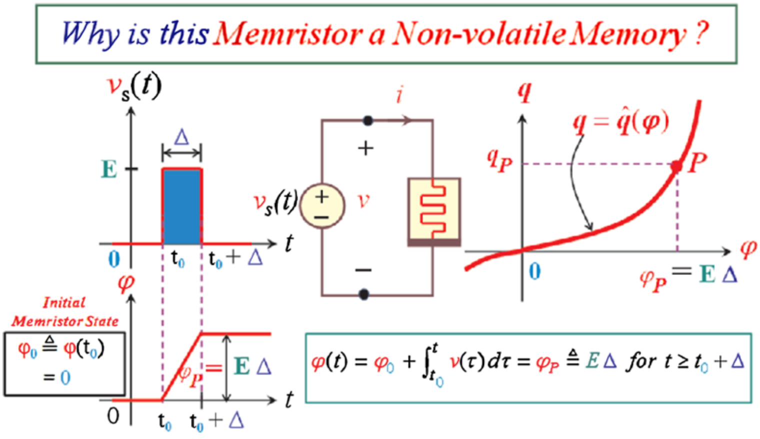

Why is this Memristor a Non-volatile Memory ?
$v_s(t)$
Δ
E
0
t_0
$t_0 + \Delta$
t
φ
Initial Memristor State
$\varphi_0 \triangleq \varphi(t_0) = 0$
$\varphi_P = E\Delta$
0
t_0
$t_0 + \Delta$
t
i
+
$v_s(t)$
v
−
q
$q = \hat{q}(\varphi)$
q_P
P
0
$\varphi_P = E\Delta$
φ
$\varphi(t) = \varphi_0 + \int_{t_0}^{t} v(\tau)d\tau = \varphi_P \triangleq E\Delta$ for $t \geq t_0 + \Delta$

Fig. 5 The operating point P remains invariant after power is off [15], i.e. $v_s(t) = 0$ and $i = 0$

of Fig. 5 is given by

$$i(t) = \frac{dq(t)}{dt} = \underbrace{\frac{d\hat{q}(\varphi)}{d\varphi}}_{G(\varphi)} \underbrace{\frac{d\varphi(t)}{dt}}_{v} = G\big(\varphi(t)\big)v(t) \tag{8}$$

It follows that

$$\left.\begin{aligned} i(t) &= G(0) \cdot 0 = 0, && t \leq t_0 \\ &= G(0) \cdot E, && t = t_0^+ \\ &= G(\varphi(t)) \cdot E, && t_0^+ \leq t \leq t_0 + \Delta \\ &= G(\varphi_P) \cdot 0 = 0, && t \geq t_0 + \Delta \end{aligned}\right\} \tag{9}$$

where $\varphi(t)$ is the saturated ramp waveform shown in the lower left of Fig. 5, and $G(\varphi(t))$ is the *slope* of the q *versus* φ curve at $\varphi = \varphi(t)$. Hence $i(t) = 0$ whenever $v(t) = 0$, and no information is retained before and after the application of the voltage pulse, assuming that we are coding the binary state “0” and “1” by the magnitude of the voltage v (“low voltage” or “high voltage”, respectively). However, if we code the low memductance $G(0)$ at the origin in Fig. 5 as binary state “0” and the high memductance $G(\varphi_P)$ at P, where $\varphi = E\Delta \triangleq \varphi_P$, as binary state “1”, then the memristor has remembered its memductance for all times $t \geq t_0 + \Delta$ even long after the voltage pulse had elapsed.

Moreover, observe that the ideal memristor in Fig. 5 is not only a non-volatile discrete binary memory, it can function even as an *analog* memory because the value of the memductance at any point along the q *versus* φ curve is distinct from all other points. In other words, an ideal memristor with a strictly monotonically-increasing q *versus* φ constitutive relation can store *any* memductance value represented by the continuously tunable slope along the q *versus* φ curve. We will see in Sect. 4 that this *non-volatile continuum* of memductances is precisely the property required for *learning* by tuning the "*weight*" of a *synapse*. But what characteristics must a *non-ideal* memristor possess in order to exhibit the "non-volatile memory" property endowed in all *ideal* memristors? The answer is given by the following theorem [14, 15].

Theorem 1 (Non-volatile memory criteria) *A memristor described by* (1a), (1b) (*resp*., (2a), (2b)) *has a continuum of non-volatile memory states if, and only if,*

$$\mathbf{f}(\mathbf{x}, i) = \mathbf{0} \quad \textit{whenever } i = 0 \textit{ for all } \mathbf{x} \in \mathbb{R}^n \tag{10a}$$

resp.,

$$\mathbf{g}(\mathbf{x}, v) = \mathbf{0} \quad \textit{whenever } v = 0 \textit{ for all } \mathbf{x} \in \mathbb{R}^n \tag{10b}$$

Proof It follows from

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, 0) = \mathbf{0} \qquad \left(\text{resp., } \frac{d\mathbf{x}}{dt} = \mathbf{g}(\mathbf{x}, 0) = \mathbf{0}\right) \quad \text{for all } \mathbf{x} \in \mathbb{R}^n,$$

that *any state* $\mathbf{x} \in \mathbb{R}^n$ is an *equilibrium state* of any memristor satisfying (10a) (resp., (10b)) when the power is turn-off, i.e. when $i = 0$ (resp., $v = 0$).

As a trivial application observe that for an *ideal memristor*, we have, *by definition*,

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, i) = i = \mathbf{0} \quad \text{whenever } i = 0 \text{ for all } q \in \mathbb{R}^1 \tag{11a}$$

resp.,

$$\frac{d\mathbf{x}}{dt} = \mathbf{g}(\mathbf{x}, v) = v = \mathbf{0} \quad \text{whenever } v = 0 \text{ for all } \varphi \in \mathbb{R}^1 \tag{11b}$$

Hence the *non-volatile memory criteria* (10a) and (10b) are satisfied trivially, implying *all ideal memristors are non-volatile memories*. □

4 Synapses Are Memristors

4.1 Learning with Memristors: Habituation

Eric Kandel was awarded the Nobel Prize in medicine and physiology in 2000 for uncovering the *molecular basis of memory*. In the Nobel lecture Kandel delivered at the Karolinska Institute on December 8, he flashed on the screen the picture of a sea snail, dubbed *Aplysia Californica,* with a Nobel Prize medal draped around its neck (Fig. 6), in recognition of the crucial role the Aplysia's *synapses* had in providing

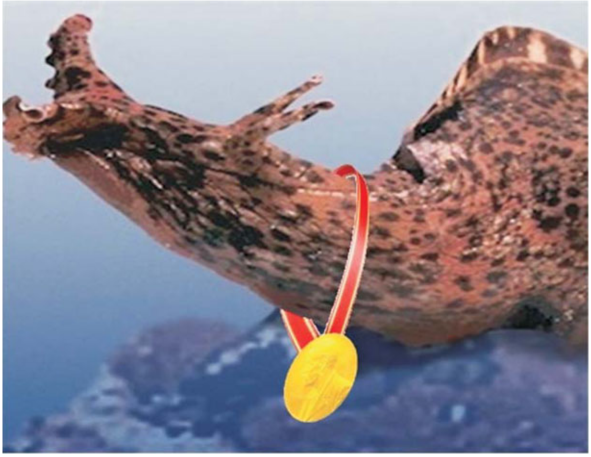
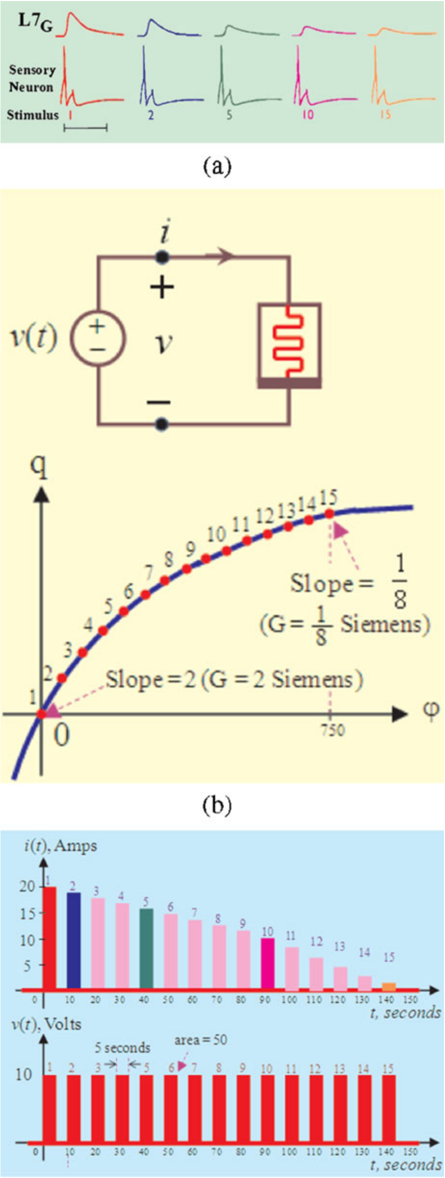


Fig. 6 Aplysia with a Nobel Prize medal

Kandel with the key to understanding the simplest form of *learning*; namely, *habituation* [16]. All of us use habituation everyday to help us learn *not* to attend or respond to irrelevant stimuli. The Aplysia is an invertebrate endowed with a retractable *gill* and tube-like extension, called the *siphon*, which are delicate organs that can be easily damaged. Consequently, when there are signs of danger, the Aplysia would retract its gill and siphon inside a protective mantle shelf. This is called the *gill-withdrawal reflex*, which displays habituation after repeated presentation of a stimulus, such as a periodic squirt of a stream of water jet, or a periodic gentle touch of a soft calligraphy brush, which was what Kandel used in his classic experiments on the Aplysia. In such experiments, sensory information (action potential) from the siphon travels along a nerve until it enters a region of the Aplysia nervous system called the abdominal ganglion [17]. Here the information is distributed to motor neurons, some of which in turn generate pulse-like electrical voltage signals and travel via nerves to the muscles that produce gill withdrawal, whose withdrawing rate depends on the strength of the motor signals. One of the motor neurons that receives direct monosynaptic sensory input from the siphon is identified in the neurophysiology literature as L7, and this cell innervates the muscles that produce gill withdrawal.

In a series of such carefully designed experiments, Kandel had observed that *repeated stimulations* of the siphons skin leads to *progressively less contraction* of the gill-withdrawal muscles, which resulted from a corresponding weakening signals, as depicted in the top row of Fig. 7a for stimulus number 1, 2, 5, 10, and 15. The corresponding "*action potentials*" which emerges from the sensory neuron on the siphon skin is shown in the bottom. Observe that the amplitude of the action potentials remains *undiminished*. How then does the motor neurons innervating the gill-withdrawal muscle *learn* the diminishing degree of danger and respond accordingly with a corresponding series of motor signals of diminishing strength? In other words, how does the relevant *synapse* learn to reduce its synaptic weight which mirrors the diminishing motor signal strength shown in Fig. 7a?



(a)

(b)

(c)

Fig. 7 (**a**) A sample of 5 identical action potentials nos. 1, 2, 5, 10, and 15 (*bottom*) recorded by Kandel, along with the corresponding motor response signals of diminishing strength recorded at the motor neuron (identified by L7_G) (*top*) (**b**) An ideal memristor with its convex piecewise-linear q vs. φ curve consisting of 15 equally-spaced linear segments of uniformly decreasing slopes. In particular the slope is equal to $G = 2$ at segment 1 and $G = \frac{1}{8}$ at segment 15 (**c**) The waveform of the excitation voltage source $v(t)$ consists of a square wave of 10-Volt amplitude and 5-second duration. The corresponding memristor current response $i(t)$ shown in the top row consists of a pulse train of diminishing pulse amplitudes

We now present in Fig. 7b a "toy" memristor circuit consisting of an *ideal* memristor endowed with a monotonically-increasing piecewise-linear q *versus* φ constitution relation $q = \hat{q}(\varphi)$ with diminishing slopes. Since $i = G(\varphi_k)v$ at segment k, where $G(\varphi_k)$ is the memductance evaluated at $\varphi = \varphi_k$ at segment k, it follows that[3] $\varphi(t) = \int_0^t v(\tau)d\tau$ for $t \geq 0$ is a monotonically-increasing step-wise staircase signal, resulting in a corresponding monotonically-decreasing memductance

$$G(t_k) = \left.\frac{d\hat{q}(\varphi)}{d\varphi}\right|_{\varphi=\varphi(t_k)} \tag{12}$$

Since the amplitude of the voltage pulse in the bottom of Fig. 7c is fixed at 10 Volts, the corresponding memristor current pulses must mirror the decreasing memductance $G(t_k)$=slope of the $q - \varphi$ curve at segment K as depicted in the top row of Fig. 7c. For the readers convenience, we have highlighted the memristor current response $i(t)$ at $t = 0$, 10s, 40s, 90s, and 140s, respectively, (labelled as pulse no. 1, 2, 5, 10 and 15) with the same color as the 5 sampled motor signals depicted in the top row of Fig. 7a. Note the striking *qualitative* resemblance between the motor signal response in Fig. 7a and the memristor current response in Fig. 7c. It follows that the ideal memristor defined in Fig. 7b is capable of the simplest form of learning; namely habituation [16, 17].

4.2 Learning with Memristors: LTP

In 1973, just two years after the memristor was postulated [2], Timothy Bliss and Terje Lomo had discovered that a brief high-frequency electrical stimulation of an excitatory pathway to the *hippocampus* produced a long-lasting enhancement in the strength of the stimulated *synapses* [17–19]. This phenomenon is now known as *long-term potentiation*, or *LTP*. This *memory* phenomenon has attracted a lot of interest because it appears that *plasticity* at many *synapses* in the cerebral cortex may be governed by the same *Hebbian-like* "learning rules" and might use the same mechanism. We will show the LTP phenomenon can be emulated by the same "toy" memristor circuit in Fig. 7b, albeit with a "concave" q *versus* φ memristor constitutive relation. It follows that the LTP phenomenon can be considered as a "*biological memristor*".

The LTP phenomenon discovered by Bliss and Lomo can be summarized in the simplified experimental set-up shown in Fig. 8 where a square voltage pulse $x_1(t)$ is applied to the *Schaffer collateral synapse* on the CA1 pyramidal neurons in brain slice preparations,[4] labelled simply by the yellow Hippocampus box in Fig. 8. The response is measured by the EPSP signal at the postsynaptic CA1 neuron [19]. This initial part of the experiment is depicted in Fig. 8 with the red switch contact connecting the voltage source $x_1(t)$ at switch position ① directly to the input of the Hippocampus neuron. The output $y(t)$ of the Hippocampus neuron in this case is

[3] Here we assume "*zero*" initial state, i.e. $\varphi(0) = 0$. Hence $\varphi(t) = \int_{-\infty}^t v(\tau)d\tau = \int_0^t v(\tau)d\tau$.

[4] The original Bliss-Lomo experiment was carried out at the *perforant path synapses on the dentate gyrus*.

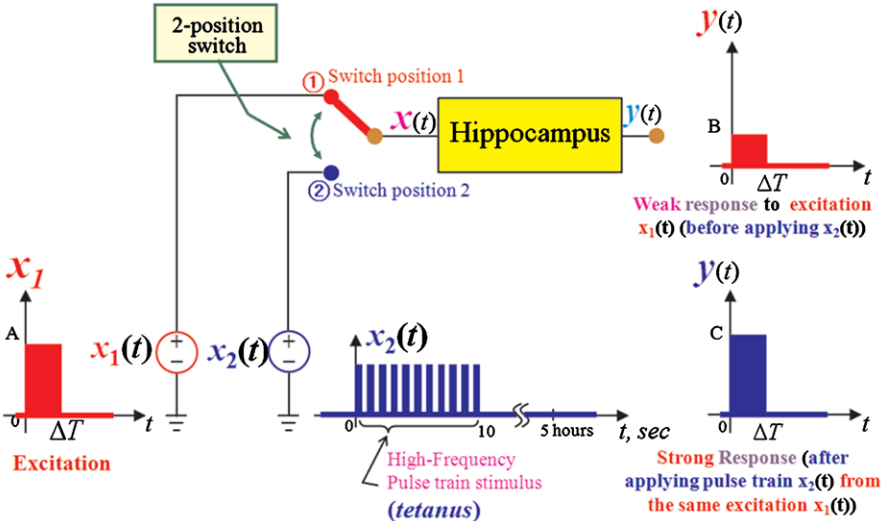


Fig. 8 Simplified schematic diagram depicting the classic Bliss-Lomo experimental set-up leading to the LTP phenomenon

represented by the “weak” red square pulse with amplitude B. The crux of the Bliss-Lomo experiment is to “prime” the Hippocampus neuron with a brief burst of high-frequency stimulation (typically 50–100 stimuli at a rate of about 100 Hz) called *tetanus* in biophysical parlance. This priming process is executed in the schematic diagram of Fig. 8 by first moving the switching contact to position ②, thereby connecting the blue *tetanus* voltage pulse train to the input of the Hippocampus neuron. Shortly after this priming procedure is implemented, Bliss and Lomo then applied *the same* red input square pulse in Fig. 8 to the input of the Hippocampus neuron by flipping the red switch contact to position ①, whereupon they were stunned to observe a much stronger response at the output of the Hippocampus neuron, as depicted by the much stronger output voltage pulse with an amplitude $C \gg B$. In other words, the *tetanus* has caused a modification of the stimulated synapse so it has become more effective. In modern language, the LTP phenomenon can be interpreted as a form of *Hebbian learning* where a neuron can tune its weight after repeated excitations and store its optimized value for long-term memory.

We will now demonstrate the LTP phenomenon can be emulated by the memristor circuit shown in Fig. 9a with the q *vs.* φ constitutive relation defined by the 2-segment piecewise-linear curve shown in Fig. 9b. For pedagogical reasons, we have chosen the numbers indicated in the figure to simplify calculation. In this idealized example, the memristor is equivalent to a 2 Ohm resistor (corresponding to a 0.5 Siemens memductance) so long as $\varphi(t) \leq 10$ Webers. It drops to a 1 Ohm resistor when $\varphi(t) > 10$ Webers. Applying the 5-Volt square voltage pulse (Fig. 9c) across the memristor in Fig. 9a results in the relatively “weak” 2.5-Ampere current response shown in Fig. 9d, assuming $\varphi(0) = 0$. If, however, we apply first the red 5-Volt pulse train shown in Fig. 9e across the memristor, with the same initial state

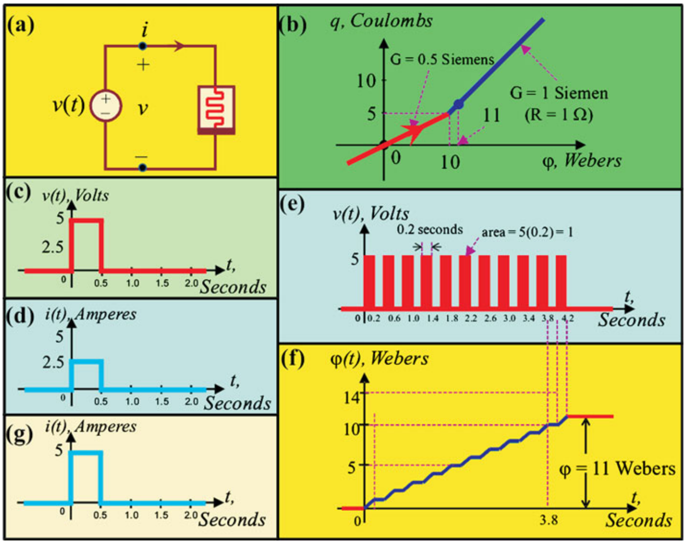


Fig. 9 A toy memristor circuit which can mimic the LTP phenomenon described in Fig. 8

$\varphi(0) = 0$, we would obtain the monotonically-increasing blue staircase waveform $\varphi(t)$ shown in Fig. 9f. Note that $\varphi(t) > 10$ Webers for $t > 3.8$ seconds. In other words, after applying a *tetanus* pulse train with more than 10 pulses, the memristor in Fig. 9a changes its resistance from 2 Ohms to 1 Ohm, thereby doubling the current amplitude to 5 Amperes, as shown in Fig. 9g, when the same red 5-Volt square voltage pulse in Fig. 9c is applied.

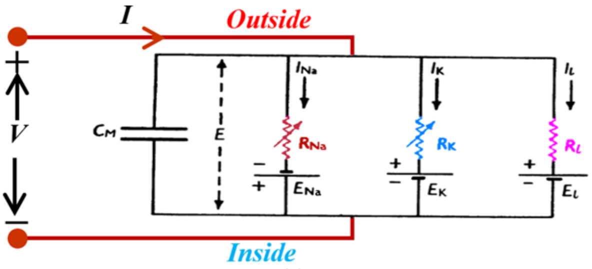
Finally observe that if we double the amplitude of the *tetanus* pulse train in Fig. 9e, while decreasing the pulse width from 0.2 seconds to 0.1 seconds, we would obtain exactly the same current response shown in Fig. 9g. It is quite remarkable that the "pulse amplitude" and "pulse width" tradeoff behavior predicted from our toy memristor circuit in Fig. 9a was also found to occur in laboratory LTP experiments [19]!

5 Hodgkin-Huxley Axon Is Made of Memristors

5.1 Anomalies of the Hodgkin-Huxley Axon Circuit Model

The *Hodgkin-Huxley* electrical *circuit model* (Fig. 10a) of the squid giant axon membrane and its associated *Hodgkin-Huxley equations* (Fig. 10b) has stood the test of time and has served as a classic reference [20] in neurophysiology and brain

Fig. 10 (**a**) The original Hodgkin-Huxley circuit model, reproduced from [20], (**b**) Hodgkin-Huxley equations



(a)

$$\frac{dV}{dt} = -\frac{1}{C_M}\left[\bar{g}_K n^4 (V - E_K) + \bar{g}_{Na} m^3 h (V - E_{Na}) + \bar{g}_l (V - E_l) - I\right]$$

$$\frac{dn}{dt} = \left[\frac{0.01\,(V+10)}{\left(\exp\frac{V+10}{10} - 1\right)}(1-n)\right] - \left[0.125\,\exp\left(\frac{V}{80}\right) n\right]$$

$$\frac{dm}{dt} = \left[\frac{0.1\,(V+25)}{\left(\exp\frac{V+25}{10} - 1\right)}(1-m)\right] - \left[4\,\exp\left(\frac{V}{18}\right) m\right]$$

$$\frac{dh}{dt} = \left[0.07\,\exp\left(\frac{V}{20}\right)(1-h)\right] - \left[\frac{1}{\left(\exp\frac{V+30}{10} + 1\right)} h\right]$$

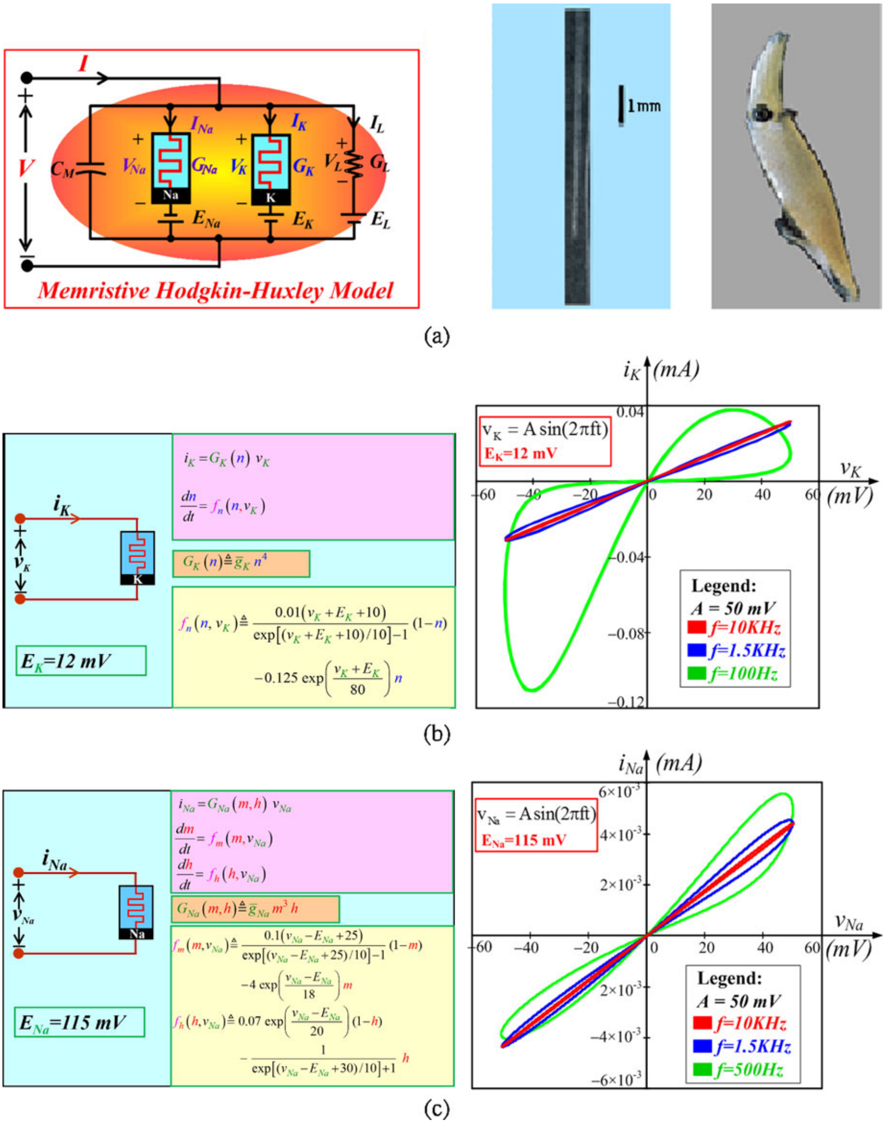
(b)

science research for over 70 years.[5] The *squid* was chosen by Hodgkin-Huxley because they are endowed with enormous axons, the largest of them in a large Atlantic squid (*Loligo pealii*) being as much as one millimeter in diameter (Fig. 11a). The Hodgkin-Huxley equations were derived strictly by *empirical* methods, and therefore did not have a sound circuit-theoretic foundation. In spite of its great success in emulating the *action potential*, there are three anomalies that had remained unresolved until recently [21, 22].

5.1.1 Anomaly 1

The *first anomaly* is concerned with the two circuit elements labeled R_K and R_{Na} in Fig. 10a which Hodgkin and Huxley had identified as *time-varying* potassium and sodium conductances, respectively. The anomaly arises from the fact that the time variations of $G_K = \frac{1}{R_K}$ and $G_{Na} = \frac{1}{R_{Na}}$ can *not* be specified a priori, but

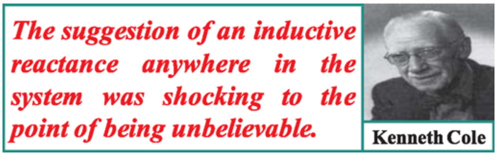
[5] Hodgkin and Huxley were awarded the 1961 Nobel Prize in Physiology for this seminal contribution.



I
V
C_M
V_{Na}
G_{Na}
V_K
G_K
I_{Na}
I_K
I_L
V_L
G_L
Na
K
E_{Na}
E_K
E_L
Memristive Hodgkin-Huxley Model
1mm
(a)
i_K
v_K
K
E_K=12 mV
$i_K = G_K(n)\, v_K$
$\frac{dn}{dt} = f_n(n, v_K)$
$G_K(n) \triangleq \bar{g}_K\, n^4$
$f_n(n, v_K) \triangleq \frac{0.01(v_K + E_K + 10)}{\exp[(v_K + E_K + 10)/10] - 1}(1-n) - 0.125 \exp\left(\frac{v_K + E_K}{80}\right) n$
i_K (mA)
v_K (mV)
$v_K = A\sin(2\pi ft)$
E_K=12 mV
0.04
−0.04
−0.08
−0.12
−60
−40
−20
0
20
40
60
Legend:
$A = 50$ mV
🟥 f=10KHz
🟦 f=1.5KHz
🟩 f=100Hz
(b)
i_{Na}
v_{Na}
Na
E_{Na}=115 mV
$i_{Na} = G_{Na}(m,h)\, v_{Na}$
$\frac{dm}{dt} = f_m(m, v_{Na})$
$\frac{dh}{dt} = f_h(h, v_{Na})$
$G_{Na}(m,h) \triangleq \bar{g}_{Na}\, m^3 h$
$f_m(m, v_{Na}) \triangleq \frac{0.1(v_{Na} - E_{Na} + 25)}{\exp[(v_{Na} - E_{Na} + 25)/10] - 1}(1-m) - 4\exp\left(\frac{v_{Na} - E_{Na}}{18}\right) m$
$f_h(h, v_{Na}) \triangleq 0.07\exp\left(\frac{v_{Na} - E_{Na}}{20}\right)(1-h) - \frac{1}{\exp[(v_{Na} - E_{Na} + 30)/10] + 1} h$
i_{Na} (mA)
v_{Na} (mV)
$v_{Na} = A\sin(2\pi ft)$
E_{Na}=115 mV
6×10^{-3}
4×10^{-3}
2×10^{-3}
2×10^{-3}
-4×10^{-3}
-6×10^{-3}
−60
−40
−20
0
20
40
60
Legend:
$A = 50$ mV
🟥 f=10KHz
🟦 f=1.5KHz
🟩 f=500Hz
(c)

Fig. 11 Hodgkin-Huxley Axon. (**a**) Memristive Hodgkin-Huxley Circuit model of the giant axon (*center*) of North Atlantic squid Loligo (*right*) (**b**) potassium ion-channel memristor and its pinched hysteresis loops (**c**) sodium ion-channel memristor and its pinched hysteresis loops

rather must be calculated numerically from empirically-derived nonlinear differential equations, thereby violating the fundamental circuit-theoretic concept of a *time-varying* circuit element [23, 24], where all time-varying parameters *must be* prescribed as an *explicit function of time*. We have resolved this anomaly in [21, 22] by showing the two elements R_K and R_{Na} in Fig. 10a are in fact *memristors* as



The suggestion of an inductive reactance anywhere in the system was shocking to the point of being unbelievable.

Kenneth Cole

Fig. 12 A quotation from page 78 of [25]

shown in Fig. 11a. In particular Hodgkin-Huxley's time-varying potassium conductance G_K is a *voltage-controlled memristor* associated with one state variable "n", dubbed the potassium gate-activation variable whose associated state equation is given in Fig. 11b. Similarly, Hodgkin-Huxley's time-varying sodium conductance G_{Na} is a *voltage-controlled memristor* associated with two state variables "m" and "h" dubbed the sodium gate-activation and inactivation variable, respectively, whose associated state equations are given in Fig. 11c. To confirm that both G_K and G_{Na} in the memristive Hodgkin-Huxley model in Fig. 11a are memristors, we show two pinch hysteresis loops corresponding to two frequencies $f_1 = 100$ Hz for G_K and 500 Hz for G_{Na} and $f_2 = 1.5$ KHz for both G_K and G_{Na}, when the input voltage is a sine wave with amplitude $A = 50$ mV.

Moreover, we show these hysteresis loops degenerate into a straight line whose slope depends on f for $f \geq 10$ KHz [11].[6] All of these pinched hysteresis loops therefore pass the *fingerprint test* for memristors, as prescribed in Sect. 2.

5.1.2 Anomaly 2

Both Hodgkin and Cole [25], and many other well-known physiologists, have measured the small-signal impedance $Z(i\omega)$ and admittance $Y(i\omega)$ of the squid giant axon membrane, at various DC bias (operating points) and found that over certain range of frequencies, and DC bias, the axon impedance exhibits a gigantic inductance exceeding several Henries. Since such inductors would call for winding a very long copper wire around iron cores weighing several Kilos, and would create a huge magnetic field, it is an inexplicable mystery! To understand the conundrum and bewilderment created by this anomaly, we reproduce a sentence in Fig. 12 from page 78 of the classic book [25] by Kenneth Cole, one of the most distinguished physiologists during the Hodgkin-Huxley era.

5.1.3 Anomaly 3

Both Hodgkin and Cole [25] had also measured the large-signal static V–I characteristic curve of the squid giant axon, and found that it resembles the DC V–I curve

[6] We caution the readers that these pinched hysteresis loops are different from those shown in Fig. 4 of [4], which were calculated with $E_K = 0$, and $E_{Na} = 0$, respectively, in the state equations given in Figs. 11b and 11c. We take this opportunity to alert the readers of [21] that the pinched hysteresis loops in Figs. 11, 12, 17 and 18 are calculated with $E_K = 0$ and $E_{Na} = 0$.

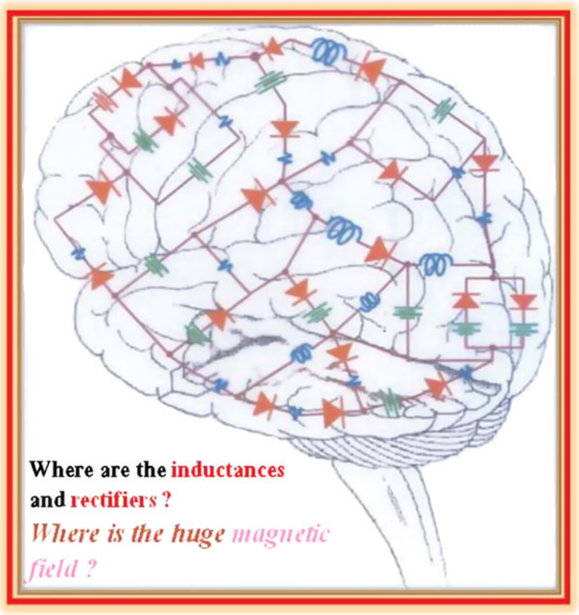


Fig. 13 A cartoon dramatizing the futile search for non-existing diodes, inductors, and their associated magnetic fields in the brain during the Hodgkin-Huxley era

of vacuum tube diodes (i.e. rectifiers). But how can neurons behave like vacuum tube diodes? Many physiologists were endlessly perplexed by both anomaly 2 and anomaly 3, searching in vain for the missing inductors and diodes depicted in the cartoon in Fig. 13, which was drawn to dramatize the plights during the Hodgkin-Huxley era. In Sects. 5.2.2 and 5.2.3, we will resolve both *anomaly 2* and *anomaly 3* by analyzing the memristive Hodgkin-Huxley model in Fig. 11a, obtained by substituting Hodgkin-Huxley's misidentified *time-varying* potassium and sodium conductances with a *time-invariant* potassium memristor, and a *time-invariant* sodium memristor, respectively. Hodgkin-Huxley's blunder was caused by their erroneous classification of R_K and R_{Na} in Fig. 10a as *time-varying* circuit elements, when in fact they are *time-invariant*, because no device parameters in Figs. 11b and 11c vary with time.

5.2 Deriving the DC V–I Characteristic of the Hodgkin-Huxley Axon

The set of all *steady state* (i.e., $\frac{di}{dt} = 0$ and $\frac{dv}{dt} = 0$) DC current solutions I (resp., DC voltage solutions V) corresponding to *all admissible*[7] DC voltage sources (resp.,

[7] For certain *ideal* 2-terminal circuit elements [23, 24], only a DC voltage source (resp., current source) restricted to a limited range of terminal voltages (resp., currents) is *admissible*. For ex-

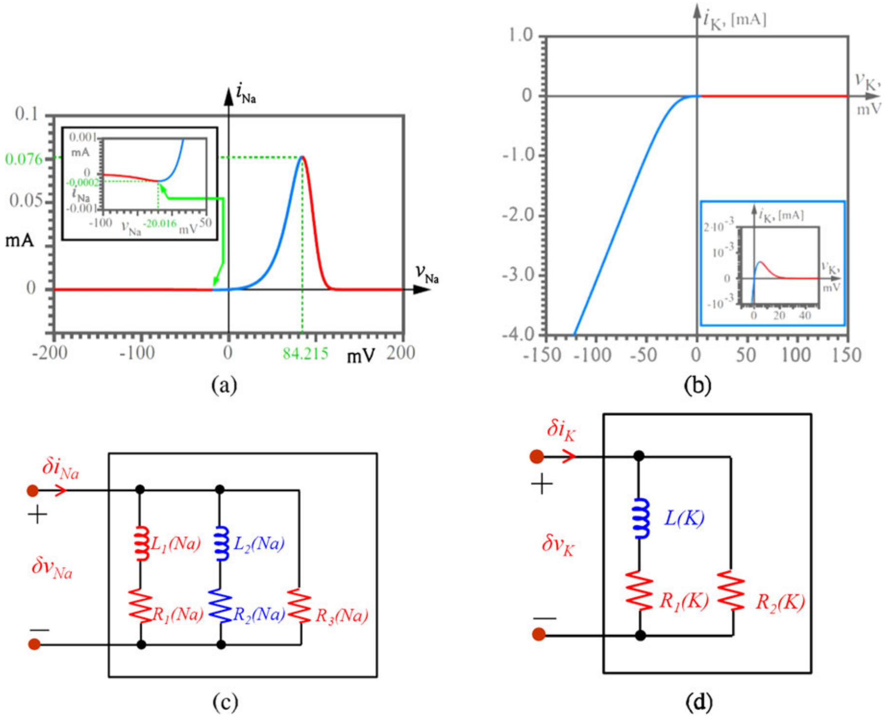


Fig. 14 (**a**) DC V_{Na}–I_{Na} Curve of sodium Ion- Channel Memristor (**b**) V_K–I_K Curve of potassium Ion- Channel Memristor (**c**) sodium ion-channel memristor small-signal equivalent circuit at equilibrium point $v_{Na}(Q)$ (**d**) potassium ion-channel memristor small-signal equivalent circuit at equilibrium point $v_K(Q)$

DC current sources) applied across a 2-terminal device is called the device's DC V–I characteristic.

5.2.1 Example: DC V_K–I_K Curve of Potassium Memristor

To derive the DC V_K–I_K characteristic of the potassium memristor, we assign *all* admissible values of $v_K \in \mathbb{R}^1$ to the function $f(n, v_K)$ defined in Fig. 11b and solve numerically for the solution $n = \hat{n}(V_K)$ of $f(n, V_K) = 0$, and then substituting it into $i_K = G_K(n)v_K$ to obtain the DC V_K–I_K curve $I_K = G_K(\hat{n}(V_K))V_K$ shown in Fig. 14b. Note the part of the curve with a *negative* slope is printed in red.

ample, for an *ideal diode* [23, 24], only a *non-positive* voltage source (resp., *non-negative* current source) is allowed (by the *definition* $v = 0,\ i \geq 0$, and $i = 0,\ v \leq 0$ of an *ideal diode*) to be connected across the ideal diode, in order to avoid the pathological situation where the circuit does not have a solution!

5.2.2 Example: DC V_{Na}–I_{Na} Curve of Sodium Memristor

To derive the DC V_{Na}–I_{Na} characteristic of the *sodium memristor*, we assign *all* admissible values of $v_{Na} \in \mathbb{R}^1$ to the two functions $f_m(m, v_{Na})$and $f_h(h, v_{Na})$ defined in Fig. 11c, and solve numerically for the solutions $m = \hat{m}(V_{Na})$ of $f_m(m, V_{Na}) = 0$, and $h = \hat{h}(V_{Na})$ of $f_h(h, V_{Na}) = 0$, and then substituting them into $i_{Na} = G_{Na}(m, h)v_{Na}$ to obtain the DC V_{Na}–I_{Na} curve $I_{Na} = G_{Na}(\hat{m}(V_{Na}), \hat{h}(V_{Na}))V_{Na}$ shown in Fig. 14a. Note the part of the curve with a *negative* slope is printed in red.

5.2.3 DC V–I Curve of Hodgkin-Huxley Axon

To derive the DC V–I curve of the Hodgkin-Huxley axon, we delete the membrane capacitor C_M in Fig. 11a and add the three currents I_{Na}, I_K, and $I_L$as a function of $V = V_{Na}–E_{Na}$, $V = V_K + E_K$, and $V = V_L + E_L$, respectively, where $E_{Na} = 115$ mV, $E_K = 12$ mV, and $E_L = -10.613$ mV are given in [20]. By aligning the $V = 0$ vertical axis, we can use the graphical construction method depicted in Fig. 15 to obtain the DC V–I curve shown in the bottom of Fig. 15. This DC V–I curve resembles the V–I curve of a vacuum tube *diode* (*rectifier*), as reported by Cole and others [25].[8] Hence we have resolved *Anomaly 3* by deriving the DC V–I curve explicitly, which use the DC V_K–I_K curve of the potassium memristor in Fig. 14b, and the DC V_{Na}–I_{Na} curve of the sodium memristor in Fig. 14a respectively.

5.3 Deriving the Small-Signal Admittance of the Hodgkin-Huxley Axon

Applying standard linear circuit theory, we can derive a *small-signal equivalent circuit* about each DC operating point Q of the potassium memristor, and the sodium memristor, as shown in Figs. 14d, and 14c, respectively [21]. The parameter value of each circuit element in Fig. 14 is given by an *explicit* formula in [21], as a function of the DC potassium memristor voltage V_K and sodium memristor voltage V_{Na}, respectively, at the operating point Q. The *small-signal equivalent circuit* of the Hodgkin-Huxley axon at any DC voltage $V = V_Q$ is simply obtained by replacing the potassium memristor G_K and battery E_K in Fig. 16a by the potassium memristor small-signal equivalent circuit in Fig. 14d, and by replacing the sodium memristor G_{Na} and battery E_{Na} in Fig. 16a by the sodium memristor small-signal equivalent

[8] Our reference voltage polarity for V and current direction for I follow Hodgkin-Huxley's 1952 paper [20], which are opposite to the prevailing reference convention. The corresponding DC V–I curve in conventional reference polarity and direction is obtained by rotating the V–I curve in Fig. 15 by 180° degrees.

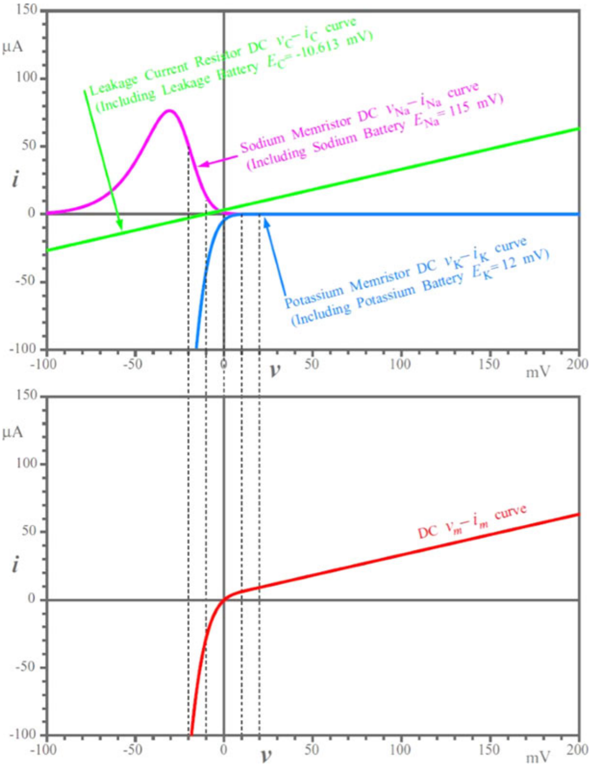


Fig. 15 Graphical procedure [23] for deriving the DC $V_m - =$ Im curve. Each point in the lower graph is obtained by adding the three component currents located on each dotted projection line. For details, see [22]

circuit in Fig. 14c, as shown in Fig. 16b, where we have combined the three parallel resistors G_L, $G_2(K) = \frac{1}{R_2(K)}$, and $G_3(Na) = \frac{1}{R_3(Na)}$ into an equivalent resistor with conductance $G_T = G_L + G_2(K) + G_3(Na)$ [21].

The admittance[9] evaluated at the DC current I_{ext} at the DC operating (equilibrium) point Q of the small-signal equivalent circuit in Fig. 16b is given by

$$Y(s; I_{ext}) \triangleq \frac{L\{\delta i(t)\}}{L\{\delta v(t)\}} \tag{13}$$

$$Y(s; I_{ext}) = \frac{b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{14}$$

where $s = \sigma + i\omega$ is the complex frequency.

The parameters $\{a_0, a_1, a_2, a_3; b_0, b_1, b_2, b_3, b_4\}$ are calculated from explicit formulas derived in [21]. Observe that the DC independent variable I_{ext} in (13) and (14) is a DC *current source* $I = I_{ext}$, where the subscript "ext" is our code for "ex-

[9] In view of the parallel network topology of Fig. 16b, it is more convenient to analyze the *admittance* $Y(s) \triangleq \frac{1}{Z(s)}$ instead of the *impedance* $Z(s)$. The independent *small-signal* variable in this case is $\delta v(t)$.

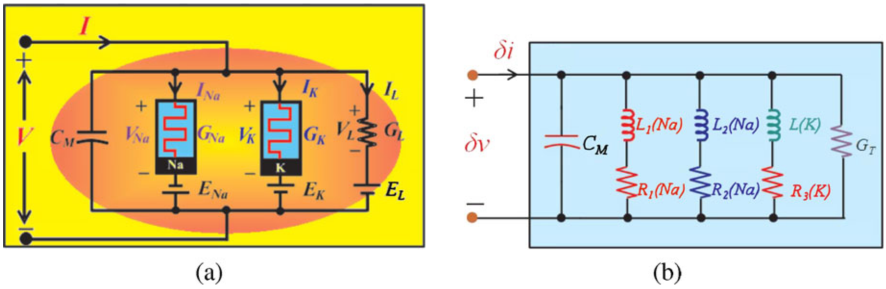


Fig. 16 (**a**) Memristive Hodgkin-Huxley model (**b**) small-signal Hodgkin-Huxley model about a DC equilibrium point Q at $(V, I) = (V_Q, I_{ext})$

ternal". In other words, $I = I_{ext}$ denotes an externally applied DC current source. By substituting $s = i\omega$ in (14), we can calculate the admittance

$$Y(i\omega; I_{ext}) = \operatorname{Re} Y(i\omega; I_{ext}) + i \operatorname{Im} Y(i\omega; I_{ext}) \tag{15}$$

due to a sinusoidal input signal at any angular frequency $\omega = 2\pi f$. We can calculate the value of an equivalent *inductance* $L(\omega)$ by the formula

$$L(\omega) = \frac{-1}{\omega[\operatorname{Im} Y(i\omega; I_{ext})]} \tag{16}$$

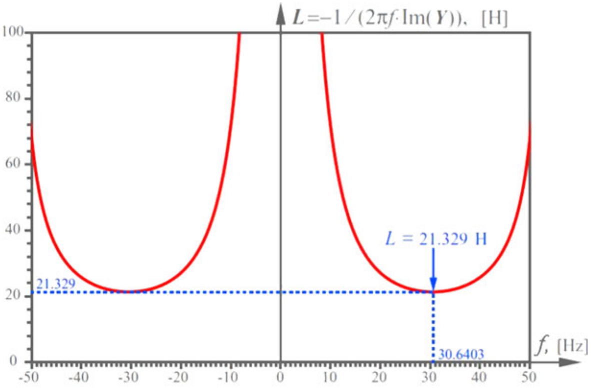
that would give the same *imaginary value* $\operatorname{Im} Y(i\omega; I_{ext})$ at any DC equilibrium point (V_Q, I_{ext}), as shown in Fig. 17, with $(V_Q, I_{ext}) = (0, 0)$. Note the inductance $L(\omega) > 0$ because $\operatorname{Im} Y(i\omega; I_{ext}) < 0$ over the frequency range $0 \leq f \leq 50$ Hz. Observe that the equivalent inductance predicted by the Hodgkin-Huxley axon model exceeds 21.329 Henries, which is consistent with the experimentally measured values reported by Hodgkin, Cole and others. We have therefore resolved *Anomaly 2* by showing the potassium and sodium memristors are the *source* of such huge inductances! No gigantic iron-core inductors are needed at all.

6 Neurons Are Poised Near the Edge of Chaos

6.1 Eigenvalues of Hodgkin-Huxley Axon Are Zeros of $Y(s)$

The Hodgkin-Huxley Equations have so far been studied via numerical integrations. No *analytical* results have been derived to date. Our analysis via the scalar admittance function $Y(s)$ as a function of the *complex variable* "s" is the first rigorous analysis of the Hodgkin-Huxley equations without involving any approximations. To understand the complex bifurcation phenomena exhibited by the Hodgkin-Huxley equations, we must calculate the *4 eigenvalues* of the *Jacobian matrix* associated with the Hodgkin-Huxley equations (Fig. 10) at each external DC current $I = I_{ext}$.

Fig. 17 Small-signal inductance predicted from the Hodgkin-Huxley axon model at rest ($V_m = 0$ mV, $I_{ext} = 0$)



21.329 H is a gigantic *inductance* manifesting a huge magnetic field if it were built with copper wire wound around an iron core.

The Jacobian matrix of the Hodgkin-Huxley equations is a 4 × 4 matrix involving many nonlinearities. Calculating the *4 eigenvalues* of the Jacobian matrix is a formidable task which has discouraged researchers from undertaking a concerted effort to analyze the global nonlinear dynamics and rich bifurcation phenomena of the Hodgkin-Huxley equations. The small-signal equivalent circuit of the potassium memristor and the sodium memristor has enabled us to derive the admittance $Y(s; I_{ext})$ of the Hodgkin-Huxley small-signal memristive circuit model in Fig. 16(b). Our next theorem is a major breakthrough because it allows us to undertake a rigorous global nonlinear analysis of the Hodgkin-Huxley equations via the eigenvalues associated with its small-signal equivalent circuit.

Theorem 2 (Fast eigenvalue calculation algorithm [22]) *The eigenvalues of the Hodgkin-Huxley Jacobian matrix are identical to the zeros of the admittance, i.e. the solution of the scalar polynomial equation.*

$$b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0 = 0 \tag{17}$$

6.2 Action Potential Originates Near the Edge of Chaos

The *action potential* generated from the Hodgkin-Huxley equations is obtained by numerical simulations. Although some explanations via the influx and outflow of sodium and potassium ions through the axon membrane provides an intuitively plausible mechanism for the emergence of an action potential when certain threshold membrane voltage is exceeded, no one really knows how the action potential is generated from a *rigorous nonlinear dynamics* perspective, until now. In a recent paper

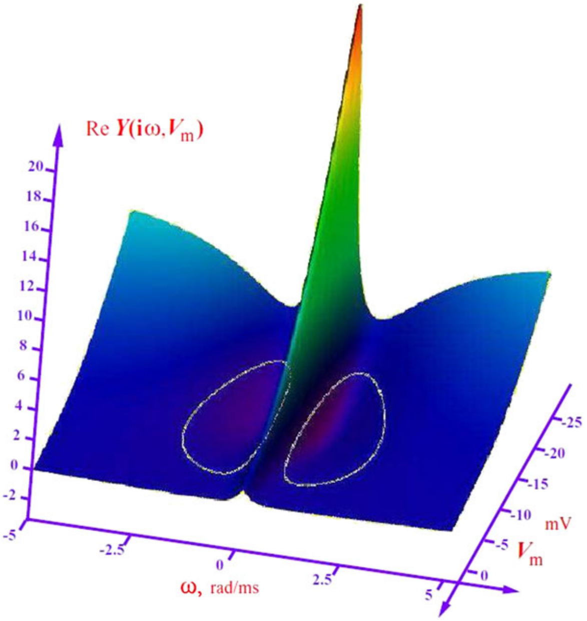


Fig. 18 The 2-dimensional surface $\text{Re}\, Y(i\omega;\, V_m)$ plotted as a function of ω and the DC axon membrane voltage $V_m(I_{ext})$. The area enclosed inside the white contour represents the region in the $(\omega - V_m)$ plane, where $\text{Re}\, Y(i\omega;\, V_m) < 0$

[22],[10] it was shown that the action potential emerges near the *edge of chaos*, which represents a very tiny subset of the *local activity* parameter regime of the Hodgkin-Huxley axon [22]. The theory of *local activity* and its "pearl", the *edge of chaos* [26–28], is a deep mathematical theory for explaining rigorously the *emergence of* complex phenomena [29, 30], including the mechanism responsible for the emergence of the action potential. In the context of this tutorial, we summarize the main mathematical criteria for *edge of chaos* as follow.

Theorem 3 (Hodgkin-Huxley edge of chaos criteria) *The Hodgkin-Huxley neuron under synaptic input I_{ext} is on the edge of chaos if, and only if,*

1. *All zeros $z_k(I_{ext})$ of $Y(s;\, I_{ext})$ are in the open left half-plane, i.e.* $\text{Re}\, z_k(I_{exit}) < 0, k = 1, 2, 3, 4$
2. $\text{Re}\, Y(i\omega;\, I_{ext}) < 0$ *for at least one frequency $\omega = \omega_0$.*

Note that, luckily, the above two mathematical conditions requiring the derivation of the admittance $Y(i\omega;\, I_{ext})$ and its *zeros* have already been achieved in (14) and Theorem 2. In particular the region bounded inside the white closed contour in Fig. 18 is precisely the *local activity domain* of the Hodgkin-Huxley equations. Since complex phenomena, such as the emergence of an action potential, can only

[10] We take this opportunity to alert the readers of [22] of consistently repeated errata in Figs. 14–26, and Fig. 35, where the unit of ω should be *rad/ms*, and not *kHz*. This error also occurs in the text on pp. 26–30.

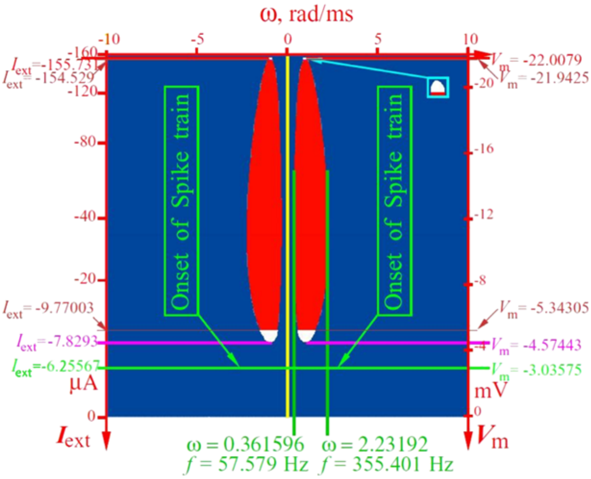


Fig. 19 The *red* area along with the *white* tips constitute the locally-active domain of the Hodgkin-Huxley equations. The *white* tips (drawn in enhanced scale) is the *edge of chaos* regime. The *blue* seas covers the *locally-passive* regions

occur inside the *locally-active-regime* of the Hodgkin-Huxley axon, we end this tutorial with a "zoom" diagram in Fig. 19 which shows the locally-active regime in *red*, and the much smaller *white tips* at the end of the 2 mirror-symmetric islands where the *edge of chaos regime* is located. The *blue* seas represent the *locally-passive regime* of the Hodgkin-Huxley equations where, except for a very small area adjacent to the "white" edge of chaos domain, no complex phenomena can emerge. A careful analysis of the range of external DC current I_{ext}, or its equivalent DC axon membrane voltage V_m, which satisfies condition 2 of Theorem 3 shows that the *edge of chaos regime* of the Hodgkin-Huxley axon is a tiny subset of the local activity domain; namely,

Edge of Chaos Regime

$$\boxed{\begin{array}{l} -5.34305 \text{ mV} \le V \le -4.577443 \text{ mV} \\ -9.77003\ \mu\text{A} \le I_{ext} \le -7.8293\ \mu\text{A} \end{array}} \tag{18}$$

Observe that $|\Delta V| < 1$ mV and $|\Delta I_{ext}| < 2$ μA represent a very tiny tunable parameter range (less than 1 mV and 2 μA). Extensive computer simulations had confirmed that *action potential* emerges either on, or nearby the above *edge of chaos regime*. It is truly remarkable that *life* exists within such a tiny parameter domain, spanned by less than 1 *millivolt* and 2 *microamperes*! It is even more gratifying that our *edge of chaos theory* [27, 28] is precise enough to identify this minuscule life-enabling *Goldilocks zone*.

7 How Did I Connect Memristor to Hodgkin-Huxley?

After searching in vain for two years for an electrical device that would mimic the behavior of the *memristor* [2], I started looking for *non-electrical* devices when

I came across an inspiring memoir from biology, entitled *Membranes, Ions, and Impulses* by Kenneth Cole [25], who had been shocked by the huge inductances which he, Hodgkin, and others, had measured from biological membranes, and had been searching in vain for several years for an analogous *non-biological* device. In his candid memoir, Cole had cited the *thermistor* as an electrical device that seems to exhibit a similar anomalous *impedance*.

Since I was very familiar with the *thermistor*, and had read some years back a little gem by Francis [9] who had suggested that *discharge tubes* behave qualitatively like *thermistors*, it suddenly dawned upon me that these two devices, as well as the *memristor*, had one common property; namely, they all obey *Ohm's law*, except that their resistance depends on a *state variable*; namely, *charge* in the case of the *memristor*, *temperature* in the case of the *thermistor*, and *ionization state* in the case of the *discharge tube*. It became obvious that they are all *generalized memristors*!

Almost immediately afterward, I found myself chewing up Francis' book [9] and was flabbergasted to discover that the *v–i loci* of all types of *discharge tubes* presented in this little gem had all passed through the origin. They are all *pinched hysteresis loops*!

It took me only 2 months in 1973 to develop my theory of the generalized *memristor* as presented in Parts I and II of [10]. But in order to illustrate my theory of the *generalized* memristor, I needed to exhibit an explicit set of memristor equations which could reproduce any family of *pinched* hysteresis loops in the v–i plane. This task took Dr. S.M. Kang two years to work out for his Ph.D. thesis, and a condensed version of which finally saw its light of day as Part III of [10].

References

1. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: Nature **453**, 80 (2008)
2. Chua, L.O.: IEEE Trans. Circuit Theory **18**, 507 (1971)
3. Tour, J.M., He, T.: Nature **453**, 42 (2008)
4. Chua, L.O.: Proc. IEEE **100**, 1920 (2012)
5. Chua, L.O.: Proc. IEEE **91**, 1830 (2003)
6. Chua, L.O.: IEEE Trans. Circuits Syst. **27**, 1014 (1980)
7. Bartle, R.G.: The Elements of Real Analysis, 2nd edn. Wiley, New York (1976)
8. Pugh, C.: Real Mathematical Analysis. Springer, New York (2000)
9. Francis, V.J.: Fundamentals of Discharge Tube Circuits. Methuen, London (1948)
10. Chua, L.O., Kang, S.M.: Proc. IEEE **64**, 209 (1976)
11. Adhikari, S.P., Sah, M.P., Kim, H., Chua, L.O.: IEEE Trans. Circuit Syst. I **60**, 3008 (2013)
12. Sapoff, M., Oppenheim, R.M.: Proc. IEEE **51**, 1292 (1963)
13. Corinto, F., Ascoli, A.: Electron. Lett. **48**, 824 (2012)
14. Chua, L.: Appl. Phys. A, Mater. Sci. Process. **102**, 765 (2011)
15. Chua, L.O.: Introduction to Memristor. IEEE Expert Now Educational Course (2009)
16. Kandel, E.R.: In Search of Memory, vol. 400. Norton, New York (2006)
17. Bear, M.F., Connors, B.W., Paradiso, M.A.: Neuroscience: Exploring the Brain. Williams & Wilkins, Philadelphia (2007)
18. Bliss, T.V.P., Lomo, T.: J. Physiol. **232**, 331 (1973)
19. Bliss, T.V.P., Collingridge Morris, R.: LTP Long Term Potentiation. Oxford University Press, Oxford (2004)

20. Hodgkin, A.L., Huxley, A.F.: J. Physiol. **117**, 500 (1952)
21. Chua, L., Sbitnev, V., Kim, H.: Int. J. Bifurc. Chaos **22**, 1230011 (2012)
22. Chua, L., Sbitnev, V., Kim, H.: Int. J. Bifurc. Chaos **22**, 1250098 (2012)
23. Chua, L.O.: Introduction to Nonlinear Network Theory. McGraw-Hill, New York (1969)
24. Chua, L.O., Desoer, C.A., Kuh, E.S.: Linear and Nonlinear Circuits. McGraw-Hill, New York (1987)
25. Cole, K.S.: Membranes, Ions and Impulses. University of California Press, Berkeley (1972)
26. Chua, L.O.: CNN: A Paradigm for Complexity. World Scientific, Singapore (1998)
27. Chua, L.O.: Int. J. Bifurc. Chaos **15**, 3435 (2005)
28. Mainzer, K., Chua, L.: Local Activity Principle: The Cause of Complexity and Symmetry Breaking. Imperial College Press, London (2013)
29. Turing, A.M.: Philos. Trans. R. Soc. Lond. B, Biol. Sci. **237**, 37 (1952)
30. Smale, S.: Lecture in Applied Mathematics, vol. 6, p. 15. American Mathematically Society, Providence (1974)

Why Are Memristor and Memistor Different Devices?

Shyam Prasad Adhikari and Hyongsuk Kim

Abstract This paper clarifies why the “memristor” is fundamentally different from a 3-terminal device with a similarly-sounding name called the “memistor”. It is shown that the memristor is a basic 2-terminal circuit element based on classic nonlinear circuit theory but the memistor is an ad hoc 3-terminal devise for one specific application. The memistor is difficult to predict its behavior when it is connected with other circuit elements.

1 Introduction

In the early 1970’s, Leon Chua of UC Berkeley presented a logical and scientific basis for the existence of a new two-terminal device called the *memristor* (a contraction of *memory resistor*) characterized by a relationship between the *charge* and the *flux linkage* [1]. The memristor exhibited some peculiar behavior different from that exhibited by resistors, inductors or capacitors, and it was introduced as the *fourth basic circuit element*. Though a physical memristor device was not discovered then, many unique applications beyond the realm of RLC circuits alone were envisioned. It was only recently when Stanley Williams group from Hewlett Packard (HP) built a nano-scale TiO_2 device [2], and provided a physical explanation and proved analytically that the HP device is in fact a memristor, and that many other devices have since been identified also as memristors [3] thereby opening up possibilities for unique applications such as artificial synapses and ultra-dense non-volatile memories.

On the other hand, Widrow and Hoff had developed a three-terminal device called the *memistor* for use in artificial neural networks [4]. Although both Widrow’s memistor and Chua’s memristor can emulate synapses, these two devices are fundamentally different as will be explained later in this paper. In particular, a memistor

S.P. Adhikari · H. Kim (✉)
Division of Electronics Engineering, Chonbuk National University, Jeonju, Jeonbuk 561-756, Republic of Korea
e-mail: hskim@jbnu.ac.kr

S.P. Adhikari
e-mail: all.shyam@gmail.com

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_6,

has 3 electrical terminals, whereas a memristor has only 2. Moreover, [5] has shown that a memistor can be built using only 2 memristors implying that the memistor is an *ad hoc* device that can be derived from the memristor. In fact, the memistor is an *ad hoc* gadget developed for emulating synapses, and not as a circuit element that can be connected with an arbitrary external circuit for general applications.

This paper is organized as follows: Sect. 2 is an introductory section and describes the memristor and the memistor, and the fundamental differences between these two devices. The memristor-based memistor is described in Sect. 3. Section 4 describes, with examples, why Widrow's memistor is not well-posed, and more distinctions between the memristor and memistor are discussed in Sect. 5. Some concluding remarks are provided in Sect. 6.

2 Memristor and Memistor are Different

2.1 Memristor

Memristor (memory resistor) is a two-terminal circuit element that exhibits a "pinched" hysteretic relationship [3] between current and voltage. Memristor is a *nonlinear* device satisfying Ohm's law with a time-varying resistance $R(t)$ whose value at any time T depends on the entire current waveform $i(t)$, $t \leq T$. In particular, the resistance of a memristor depends on the *time integral* of the memristor's current or voltage waveform. For example, the resistance (memristance) of the HP memristor increases when current flows in one direction and decreases when the direction of flow is reversed. When no current is flowing through the memristor, it retains the last resistance it had.

The existence of the "memristor" as the fourth basic circuit element was predicted from theory by Leon Chua in 1971 [1]. The prediction was based on symmetry and a missing relationship between an independent pair of the four fundamental circuit variables, namely; *current* (i), *voltage* (v), *charge* (q), and *flux linkage* (φ). The three basic circuit elements, resistor, inductor and capacitor are defined by a relationship between v and i, φ and i, and q and v, respectively, but the relationship between q and φ was missing. For the sake of symmetry and completeness, the memristor was postulated in [1] as the *fourth basic* two-terminal circuit element characterized by a relationship between φ and q, as shown in Fig. 1.

For a charge-controlled memristor defined by

$$\varphi = \hat{\varphi}(q) \tag{1}$$

we have

$$v = M(q)i \tag{2}$$

where

$$M(q) = \frac{d\hat{\varphi}(q)}{dq} \tag{3}$$

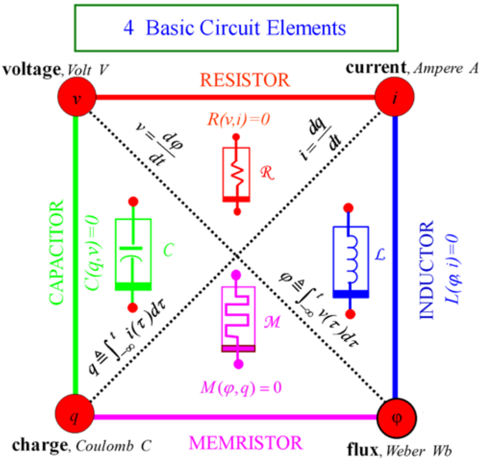


Fig. 1 The four fundamental circuit elements: resistor, capacitor, inductor and memristor, defined using the four fundamental circuit variables *current* (i), *voltage* (v), *charge* (q), and *flux linkage* (φ)

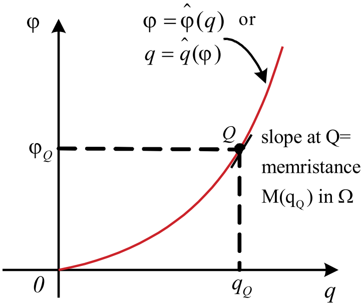


Fig. 2 Nonlinear function defining a memristor, where $\hat{q}(\cdot)$ is the inverse function of $\hat{\varphi}(\cdot)$

is called the *memristance* (contraction of memory resistance).

Thus, the memristance $M(q)$ at $q = q_Q$ of a memristor can be interpreted as the slope at an operating point Q (q_Q, φ_Q) on the φ–q curve, and has the same unit (Ohm) as the resistance, in view of Ohm's law. Since $R = M(q_Q)$ is not a constant, but depends on the "past history" of $i(t)$, it is called the *memristance*. If the φ–q curve is non-linear, as shown in Fig. 2, the memristance will vary with the operating point Q. The memristance of a memristor can be controlled by applying a voltage signal $v(t)$ across the memristor, such that Q has the co-ordinate (q_Q, φ_Q) shown in Fig. 2.

Memristors exhibit a hysteretic relationship between current and voltage. When a sinusoidal input voltage, or current, $A\sin(\omega t)$ or any *periodic bipolar* waveform, is applied to a memristor, the v–i curve exhibits a *pinched* hysteresis loop whose shape varies with the frequency ω. This "pinched" characteristic is the "fingerprint" of memristors [3]. Consequently, any device that does not exhibit a *pinched hysteresis loop* in the v–i plane when driven by a *periodic* bipolar voltage, or current waveform, is *not* a memristor. The hysteresis loop of a *passive* memristor is confined to the first and the third quadrant of the v–i plane. The shape of the pinched hysteresis loop changes with the amplitude and frequency of the periodic input. As

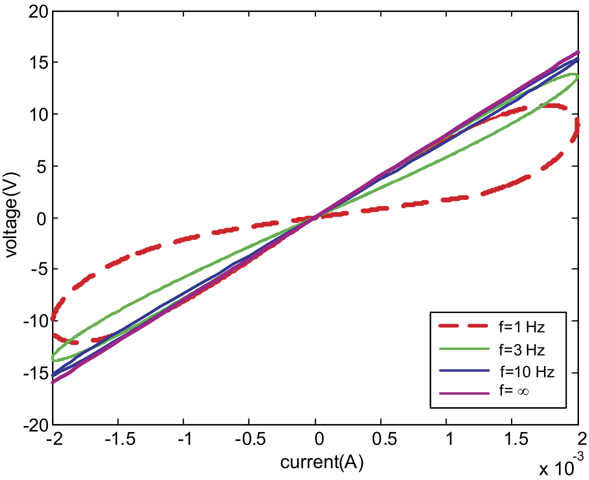


Fig. 3 Pinched hysteresis loop exhibited by a memristor when a sinusoidal input $A\sin(2\pi f t)$ is applied. The shape of the loop depends upon the amplitude A and frequency f of the applied sinusoid. In contrast, we will see in Sect. 3 that the v–i curve of a memistor consists of a straight line with a *fixed slope*, independent of the frequency of the applied sinusoid. The memistor therefore fails the "pinched hysteresis loop" *fingerprint test* for memristors

the frequency of the input signal increases, the loop shrinks, as shown in Fig. 3, and tends to a straight line as the frequency ω tends to infinity.

2.2 Memistor

Memistor (a resistor with memory) is a three-terminal device in which the resistance between two terminals is programmed by the current flowing in the *third* terminal [4]. Unlike transistors, the resistance between two terminals of a memistor depends upon the time integral of the current in the *third* terminal. In particular the resistance of Widrow's memistor is a linear function of the charge flowing from the third terminal. Although the resistance of Widrow's memistor is increased when the flow of current in the third control (programming) terminal is in one direction and decreased when the direction of flow is reversed, and retains the last resistance when no current flows through the control (programming) terminal, there is *no hysteresis* in the v–i relationship, let alone the "pinched" fingerprint, associated with any change in direction of the programming current in the third terminal.

The memistor is a *gadget* developed for use in an artificial neural network called the ADALINE (ADAptive LINEar) neuron by Widrow and Hoff in 1960. The first memistor developed by Widrow and Hoff was a chemical memistor, as shown in Fig. 4. The phenomenon of electroplating was used to control the resistance across the 2 terminals of an ordinary pencil lead, electroplated with *copper*, via a *third* copper terminal immersed in a copper sulphate-sulfuric acid bath. The thickness T of the deposited copper on the pencil lead is given by Eq. (4)

$$T = \frac{M_w}{nFAD}\int I dt \tag{4}$$

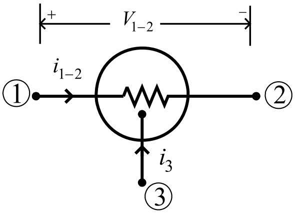


Fig. 4 Widrow's memistor consists of 3 terminals. Terminals 1–2 obeys Ohms law and $V_{1-2} = Ri_{1-2}$, where R is a *constant* for $t \geq t_0$, if $i_3(t) = 0$, $t \geq t_0$. Terminal 3 is used to program the resistance R between terminals 1–2. The relationship between i_3 and the other terminal variables such as V_{1-2}, V_{2-3}, V_{3-1}, i_{1-2} etc; is not specified in [4]. This "missing" characterization makes it impossible to predict the memistor's behavior when it is connected to an arbitrary external circuit environment

where M_w is the atomic weight of the deposited metal, n is the moles of electrons required to reduce one mole of the given metal, F is the Faraday constant, A is the area of the deposit, D is the density of the metal, and $\int Idt$ is the total charge used in the deposition.

The thickness of the electroplated copper and hence the resistance between terminals 1–2 is a function of the time integral of current flowing through the third programming terminal 3 and not the instantaneous current flowing into terminal 1–2. The resistance R_{1-2} between terminals 1–2 can be programmed with a direct current applied via terminal 3, and the programmed resistance is sensed nondestructively by passing an alternating current i_{1-2}. It follows from the above "electroplating" mechanism that the resistance R_{1-2} can only be changed by applying a voltage, or current, at a different terminal. Moreover, Widrow's memistor is a *linear* device because the resistance R_{1-2} remains unchanged *regardless of the magnitude* of the voltage across terminals 1–2, and consequently when its value is sensed with an alternating current, while open circuiting (floating) the control terminal 3, *no hysteresis* in the v–i relationship will occur. This proves that Widrow's memistor is not a memristor, which must exhibit a "*pinched hysteresis loop*" in the v–i plane when driven by any periodic "dual polarity" alternating current, or voltage, and whose hysteretic loci changes with the *frequency* ω of the sensing signal [3, 5].

3 Memristor-Based Memistor

3.1 Two Back-to-Back Series-Connected Memristors

Recently it was shown that the Widrow's memistor can be realized by connecting two HP memristors, as shown in Fig. 5. Since the memistor can be built using two memristors, and since the memristor cannot be built using 2-terminal resistors, capacitors, and inductors [1], it follows that the memristor is a fundamental circuit element, whereas the memistor is only a derived element [6].

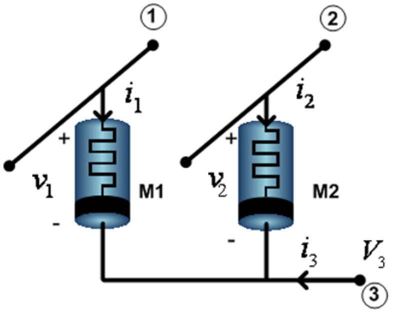


Fig. 5 Widrow memistor realized by connecting two identical HP memristors $M1$ and $M2$ connected in a back-to-back series configuration with a common bottom electrode 3. The resistance between terminals 1–2 (when terminal 3 is open circuited) can be varied by applying a programming voltage at terminal 3 with terminals 1 and 2 grounded. The resistance can be measured with a sensing signal applied to terminals 1–2 while open circuiting (floating) terminal 3

Two HP TiO_2 memristors [2], $M1$ and $M2$, are connected to a common programming electrode 3, and the top two electrodes 1 and 2 are used to access the variable resistance. The resistance of the device can be varied by applying a programming signal at the common electrode 3. When a negative voltage signal is applied across terminal 3 and ground, with terminals 1–2 grounded, the memristance of both HP memristors decreases. The overall resistance between terminals 1–2 of the two back-to-back series-connected HP memristors (when terminal 3 is open circuited) also decreases. Similarly, when a positive voltage is applied at terminal 3, the memristance of both the HP memristors increases. Hence, the total resistance between terminals 1 and 2 also increases. In this way the resistance of the memistor can be programmed by applying an appropriate input voltage signal at the third programming terminal. Simulation results of memistor programming using two TiO_2 memristors with low resistance value $R_{on} = 116\ \Omega$, high resistance value $R_{off} = 16\ \text{K}\Omega$, thickness of the sandwiched TiO_2 layer $D = 10$ nm, and dopant mobility $\mu_v = 10^{-14}\ \text{m}^2\,\text{V}^{-1}\,\text{S}^{-1}$ are shown in Fig. 6. The initial values of the memristance $M1$ and $M2$ are 8 KΩ each and the programming voltage amplitude is 1 V. A desired value of the resistance can be programmed in the memistor by controlling the width of the voltage input at the third terminal.

The resistance between terminals 1–2 is read out by applying a sensing pulse of magnitude 1 mV and duration 1 ms. The total resistance between terminals 1–2 of the memistor remains constant even when the readout signal is applied due to the complementary action of the two identical memristors. This follows from the observation that the two memristors $M1$ and $M2$ are connected in series but with opposite polarity. Hence, there is an equal drift in each memristor but in opposite directions. So the net drift in the memristance between terminals 1–2 is zero and the total memristance remains constant, as shown in Fig. 7.

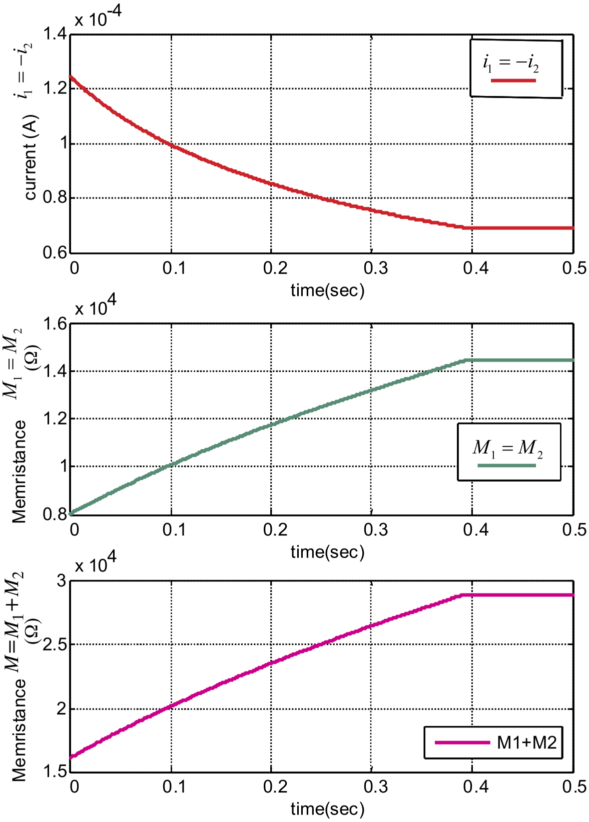


Fig. 6 Memistor programming: variation of current $i_1(t) = -i_2(t)$, memristance $M_1(t) = M_2(t)$, and total memristance $M(t) = M_1(t) + M_2(t)$ between the terminals 1–2 when a programming voltage $V_3(t) = 1\ V$ is applied to terminal 3, while keeping terminals 1 and 2 grounded

3.2 Charge vs. Flux Relationship of the Composite Device with Two Back-to-Back Series-Connected Memristors

Let us assume that the initial states of the two memristors $M1$ and $M2$ after programming are (M_{01}, φ_{01}) and (M_{02}, φ_{02}) respectively. Then, the flux as a function of charge for memristor $M1$ is given from [7] by,

$$\varphi_1(t) = R_{off}\left\{q_1(t)\left[1 + \frac{w_{01}}{D}\left(\frac{R_{on}}{R_{off}} - 1\right)\right] - \frac{\mu_v R_{on}}{2D^2}\left(1 - \frac{R_{on}}{R_{off}}\right)q_1(t)^2\right\} + \varphi_{01},$$
$$-\,Q_{\max n1} \le q_1(t) \le Q_{\max p1} \tag{5}$$

where, w is the thickness of the doped area in the TiO2 memristor, $Q_{\max pi} = \frac{D^2}{\mu_v R_{on}}(1 - \frac{w_{0i}}{D})$ for positive polarity and $Q_{\max ni} = \frac{D^2}{\mu_v R_{on}}\frac{w_{0i}}{D}$ for negative polarity.

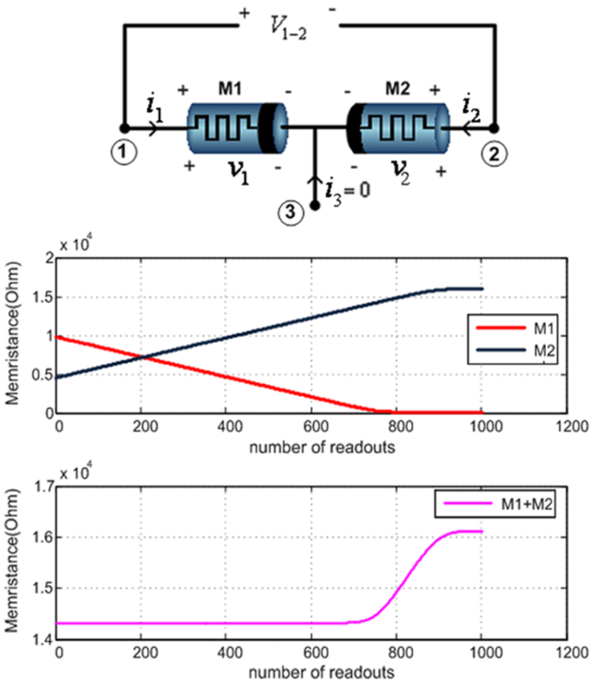


Fig. 7 Memistor readout: Drift in memristance of $M1$ and $M2$ when a readout pulse of amplitude $V_{1\text{–}2} = 1$ V and duration 1 ms is applied between terminals 1–2 while open circuiting (floating) terminal 3. Although $M1 = M2$, $\varphi_{01} = \varphi_{02}$ and $q_{01} = q_{02}$ in the programming scheme specified above, for sake of generality let us pick 2 different initial operating points for the 2 memristors in Fig. 5; namely, let us pick, arbitrarily, the following initial values: $M_1 = 9805\ \Omega$, $\varphi_{01} = 0.554$ Wb and $M_2 = 4513\ \Omega$, $\varphi_{02} = 0.76$ Wb respectively. The results shows the total resistance between terminals 1 and 2 remains constant for multiple readouts even though there is a drift in the memristance in the individual memristors

Similarly, the flux for memristor $M2$ is given by,

$$\varphi_2(t) = R_{\mathit{off}}\left\{q_2(t)\left[1 + \frac{w_{02}}{D}\left(\frac{R_{on}}{R_{\mathit{off}}} - 1\right)\right] - \frac{\mu_v R_{on}}{2D^2}\left(1 - \frac{R_{on}}{R_{\mathit{off}}}\right)q_2(t)^2\right\} + \varphi_{02},$$
$$-\,Q_{\max n2} \le q_2(t) \le Q_{\max p2} \tag{6}$$

For the two memristors (where $v_1 = \dot{\varphi}_1$, $v_2 = \dot{\varphi}_2$, $i_1 = \dot{q}_1$, and $i_2 = \dot{q}_2$ in Fig. 5) in series the *composite* flux is given by,

$$\varphi_c(t) = \varphi_{c1}(t) + \varphi_{c2}(t) \tag{7}$$

where $\dot{\varphi}_c = \dot{\varphi}_{1\text{–}2} = V_{1\text{–}2}$, $q_c = q_1$, $\varphi_{c1} = \varphi_1$, and $\varphi_{c2} = -\varphi_2$, $q_{c1} = q_1$, $q_{c2} = -q_2$ in Fig. 5.

Assuming that the polarity of the composite device is the same as that of $M1$, then the polarity of the composite device is opposite to that of $M2$. If charge $q(t)$ is injected to the composite device, it acts as a negative charge in $M2$. Thus,

$$q_2(t) = -q(t) \tag{8}$$

and Eq. (6) can be written as

$$\varphi_2(t) = R_{off}\left\{-q(t)\left[1+\frac{w_{02}}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right]-\frac{\mu_v R_{on}}{2D^2}\left(1-\frac{R_{on}}{R_{off}}\right)q(t)^2\right\}+\varphi_{02} \tag{9}$$

Also, the flux $\varphi_2(t)$ contributes to the composite memristor in an opposite direction. Thus,

$$\begin{aligned}\varphi_{c2}(t) &= -\varphi_2(t)\\ &= R_{off}\left\{q(t)\left[1+\frac{w_{02}}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right]+\frac{\mu_v R_{on}}{2D^2}\left(1-\frac{R_{on}}{R_{off}}\right)q(t)^2\right\}-\varphi_{02}\end{aligned} \tag{10}$$

The flux $\varphi_{c2}(t)$ in Eq. (10) is the transformed expression of the flux of $M2$ in the composite form. The lower part of Fig. 8 shows the flux $\varphi_{c2}(t)$ where initial states are taken as $M_{02} = 10043\ \Omega$, $\varphi_{02} = -0.538$ Wb, $q_{02} = -3.97 \times 10^{-5}$ C. In contrast, in case of $M1$, both the signs of flux and charge are identical to that of the composite device. Therefore, $\varphi_{c1}(t)$ is given by

$$\begin{aligned}\varphi_{c1}(t) &= \varphi_1(t)\\ &= R_{off}\left\{q(t)\left[1+\frac{w_{01}}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right]-\frac{\mu_v R_{on}}{2D^2}\left(1-\frac{R_{on}}{R_{off}}\right)q(t)^2\right\}+\varphi_{01}\end{aligned} \tag{11}$$

where $q_1(t)$ in Eq. (5) is replaced by $q(t)$. The composite portion of the flux graph of $M1$, $\varphi_{c1}(t)$, is as shown in the upper part of Fig. 8. In the graph, the initial state of $M1$ is taken as, $M_{01} = 10043\ \Omega$, $\varphi_{01} = 0.538$ Wb, $q_{01} = 3.97 \times 10^{-5}$ C. Plugging Eqs. (10) and (11) into Eq. (7), we obtain

$$\begin{aligned}\varphi_c(t) = {} & R_{off}\left\{q(t)\left[1+\frac{w_{01}}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right]-\frac{\mu_v R_{on}}{2D^2}\left(1-\frac{R_{on}}{R_{off}}\right)q(t)^2\right\}+\varphi_0\\ & + R_{off}\left\{q(t)\left[1+\frac{w_{02}}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right]+\frac{\mu_v R_{on}}{2D^2}\left(1-\frac{R_{on}}{R_{off}}\right)q(t)^2\right\}-\varphi_0\\ = {} & 2R_{off}q(t)\left[1+\frac{w_0}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right]\end{aligned} \tag{12}$$

where $-\min\{Q_{\max n1}, Q_{\max p2}\} \le q(t) \le \min\{Q_{\max p1}, Q_{\max n2}\}$ and $w_{01} = w_{02} = w_0$, since $M1 = M2$. The expression in Eq. (12) confirms with the result obtained

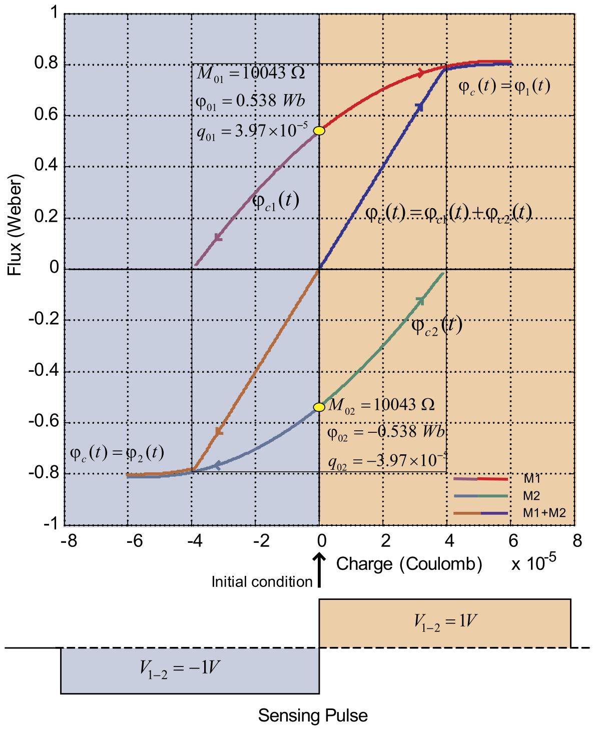


Fig. 8 Flux versus charge curve for a composite device with two identical TiO2 memristors with $R_{on} = 116\ \Omega$, $R_{off} = 16\ \text{K}\Omega$, $D = 10$ nm and $\mu_v = 10^{-14}\ \text{m}^2\,\text{V}^{-1}\,\text{S}^{-1}$ connected back-to-back in series. The initial states of the two memristors from the viewpoint of terminal 1 of the composite device are ($M_{01} = 10043\ \Omega$, $\varphi_{01} = 0.538$ Wb, $q_{01} = 3.97 \times 10^{-5}$ C) and ($M_{02} = 10043\ \Omega$, $\varphi_{02} = -0.538$ Wb, $q_{02} = -3.97 \times 10^{-5}$ C). The charge axis is the input charge to the composite device and does not represent the charge of the individual memristors. The charge of the individual memristors corresponding to the flux is obtained by adding its corresponding initial charge to the horizontal-axis value of the charge

graphically by adding the upper and lower curves as shown in Fig. 8. Furthermore, the memristance of the composite memristor can be obtained by differentiating Eq. (12) with respect to $q(t)$ as

$$\frac{d\varphi_c(t)}{dq(t)} = 2\left\{R_{off}\left[1 + \frac{w_0}{D}\left(\frac{R_{on}}{R_{off}} - 1\right)\right]\right\} = 2M_0; \tag{13}$$

where, $-\min\{Q_{\max n1}, Q_{\max p2}\} \leq q(t) \leq \min\{Q_{\max p1}, Q_{\max n2}\}$ and $M_0 = R_{off}[1 + \frac{w_0}{D}(\frac{R_{on}}{R_{off}} - 1)]$.

Equation (13) demonstrates the fact that the composite memristor exhibits a constant resistance in the range of $[-\min\{Q_{\max n1}, Q_{\max p2}\} \leq q(t) \leq \min\{Q_{\max p1}, Q_{\max n2}\}]$ and it emulates terminals 1–2 of a *memistor* over this range.

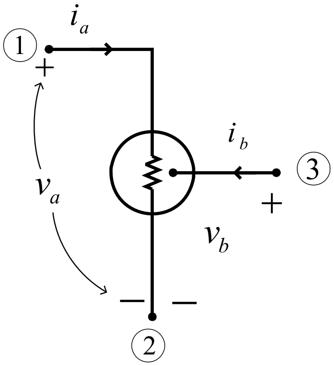


Fig. 9 Definition of 4 variables of Widrow's memistor

4 Widrow's Memistor Is Not Well-Posed

An n-terminal device, or circuit element is said to be "*well posed*" if, and only if, it has sufficient information to predict (via KCL or KVL) the current and voltage associated with all n terminals of the device when the device is connected or imbedded to an *arbitrary* external circuit. To characterize and model a well-posed three-terminal device, 2 *set* of relationships involving 2 sets of variables chosen from

1. $i_1, v_1, \frac{di_1}{dt}, \frac{dv_1}{dt}, \int_{-\infty}^{t} i_1 dt, \int_{-\infty}^{t} v_1 dt, \ldots$, higher order derivatives and/or integrals of v_1 and i_1 for terminals 1-3, and
2. $i_2, v_2, \frac{di_2}{dt}, \frac{dv_2}{dt}, \int_{-\infty}^{t} i_2 dt, \int_{-\infty}^{t} v_2 dt, \ldots$, higher order derivatives and/or integrals of v_2 and i_2 for terminals 2-3, must be specified [8, 9].

Figure 9 shows the three terminals of Widrow's memistor, where terminal 2 is chosen-without loss of generality- as the common third terminal, v_a and i_a are defined as the voltage and current between terminals 1 and 2, and v_b and i_b denote the voltage and the current between terminals 3 and 2 respectively.

For the Widrow memistor to be "well posed" two sets of relationships involving two sets of variables must be given. However, Widrow [4] had specified *only one* relationship; namely,

Relationship 1

$$v_a = R(q_b) i_a \tag{14}$$

where, $q_b = \int_{-\infty}^{t} i_b(\tau) d\tau$.

We need a *second* equation to specify how the current i_b and the voltage v_b are related to each other, and/or v_a, i_a and possibly their time derivatives and integrals; namely

Relationship 2

$$f\left(i_b, v_b, i_a, v_a, \frac{di_b}{dt}, \frac{dv_b}{dt}, \ldots, \int_{\infty}^{t} i_b dt, \int_{-\infty}^{t} v_b dt, \ldots\right) = 0 \tag{15}$$

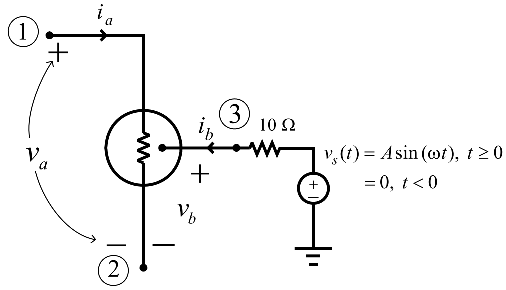


Fig. 10 To predict the waveforms of $i_a(t)$, $i_b(t)$, $v_a(t)$ and $v_b(t)$ in this circuit, for a specified input across terminals 1–2, it is necessary, to specify a second relationship involving $v_b(t)$, $i_b(t)$ and possibly other variables

This second relationship may involve one or more of the variables indicated in the general case, the precise specification depends on the *internal* physical mechanisms inside the device. Since Widrow did not specify Relationship 2, the device is not well-posed, i.e. the model is *incomplete*. The HP back-to-back series-connected arrangement of two memristors is *well posed*, but it is only one of many possible realization of a memistor.

There are many different possible ways in which the *Relationship 2* can be specified for a hypothetical memistor. Following are some examples:

Let us consider the memistor circuit with a voltage source $v_s(t)$ applied via a 10 Ω resistor as shown in Fig. 10. The waveforms of the 4 memistor variables v_a, i_a, v_b, and i_b in Fig. 9 will *depend* on the constitutive relation across terminals 2–3, even if we assume the same "memistor" constitutive relation $v_a = R(q_b)i_a$ across terminals 1–2, and driven by the same input current source, or voltage source, across terminals 1–2.

In the following examples, the constitutive relation between terminals 1–2 is given by Eq. (14).

Example 1

$$v_b = 0 \tag{16}$$

In this example the second constitutive relation is just an "internal" device relationship modeled by a *short circuit*, and hence in this case, i_b depends only on the external circuit; namely,

$$i_b = \frac{v_s(t)}{10} \tag{17}$$

Hence,

$$q_b(t) = \frac{1}{10}\int_{-\infty}^{t} v_s(t)dt = \frac{1}{10}\int_{0}^{t} A\sin(\omega t)dt, \quad t \geq 0 \tag{18}$$

Example 2

$$i_b = 0 \tag{19}$$

In this example, the second constitutive relation is just an “internal” device relationship modeled by an *open circuit*, and hence $i_b = 0$, $t \geq 0$, regardless of the external circuit. Hence,

$$q_b(t) = q_b(0) + \int_0^t (0)dt = q_b(0), \quad t \geq 0 \tag{20}$$

Example 3

$$v_b = R_b i_b \tag{21}$$

In this example, the second constitutive relation between terminals 3 and 2 is just an “internal” device relationship modeled by an R_b Ohm resistor. For example, suppose

$$v_b = 40 i_b \tag{22}$$

then,

$$i_b(t) = \frac{v_s(t)}{40+10} = \frac{v_s(t)}{50}, \quad t \geq 0 \tag{23}$$

It follows that

$$q_b(t) = \frac{1}{50}\int_{-\infty}^t v_s(t)dt = q_b(0) + \frac{1}{50}\int_0^t A\sin(\omega t)dt, \quad t \geq 0 \tag{24}$$

Example 4

$$v_b = L_b \frac{di_b}{dt} \tag{25}$$

In this example, the second constitutive relation is just a hypothetical “internal” device relationship modeled by an inductor. For example, suppose

$$v_b = 2\frac{di_b}{dt} \tag{26}$$

then,

$$v_s(t) = 10 i_b + 2\frac{di_b}{dt} \tag{27}$$

Let $i_b(t) = f(t)$ be the solution of the differential equation (27) with $i_b(0) = 0$. Then,

$$q_b(t) = q_b(0) + \int_0^t f(t)dt, \quad t \geq 0 \tag{28}$$

Note that Eq. (25) may have no relationship at all with Widrow’s gadget. But since Widrow did not specify a second relationship, it is impossible to predict the current $i_b(t)$ when Widrow’s memistor is connected to the external power supply shown in

Fig. 11 A hypothetical model of the second constitutive relationship between terminals 2 and 3 (internal to the device) of Widrow's memistor. The resistor on the left is defined by Eq. (14)

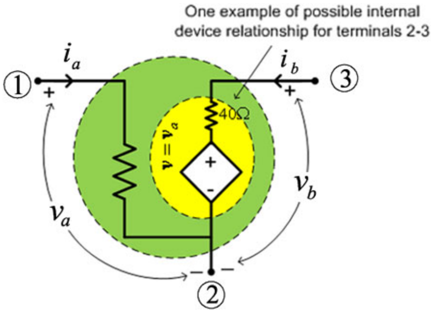

One example of possible internal device relationship for terminals 2-3
i_a
i_b
①
③
②
+
+
−
−
40Ω
$v = v_a$
+
-
v_a
v_b

Fig. 10. Note that since different specifications of the second relationship will give different $i_b(t)$, Widrow's memistor is not well posed.

Example 5

$$v_b = v_a + 40 i_b \tag{29}$$

This second hypothetical constitutive relation describing the internal physical mechanism inside the 3-terminal memistor can be modeled as in Fig. 11. Assuming a *voltage source* $v_a = g(t)$, $t \geq 0$ is applied across terminals 1–2, we obtain, upon substituting $v_b = v_s(t) - 10 i_b$ from Fig. 10,

$$\begin{aligned} i_b &= \frac{1}{40}(v_b - v_a) \\ &= \frac{1}{40}\big(v_s(t) - 10 i_b - g(t)\big), \quad t \geq 0 \end{aligned} \tag{30}$$

Solving Eq. (30) for $i_b(t)$, we obtain

$$i_b(t) = \frac{1}{50}\big(A \sin(\omega t) - g(t)\big), \quad t \geq 0 \tag{31}$$

$$q_b(t) = q_b(0) + \frac{1}{50}\int_0^t \big(A \sin(\omega t) - g(t)\big) dt, \quad t \geq 0 \tag{32}$$

Example 6

$$v_b = v_a + 40 i_b \quad \text{(same as Example 5)} \tag{33}$$

Assuming a *current source* $i_a = h(t)$ is applied across terminal 1–2, then

$$\begin{aligned} v_b &= R(q_b) h(t) + 40 i_b \\ &= -10 i_b + v_s(t) \end{aligned} \tag{34}$$

$$R(q_b) h(t) + 50 i_b = v_s(t) = A \sin(\omega t), \quad t \geq 0 \tag{35}$$

Since, $\dfrac{dq_b}{dt} = i_b$

$$\frac{dq_b(t)}{dt} + \frac{R(q_b)}{50}h(t) = \frac{1}{50}A\sin(\omega t), \quad t \geq 0 \tag{36}$$

In this case, we must solve the nonlinear differential equation (36), with initial charge $q_b(0)$, for $q_b(t)$, $t \geq 0$ to obtain

$$v_a(t) = R\big(q_b(t)\big)h(t), \quad t \geq 0 \tag{37}$$

The preceding examples illustrate why the second constitutive relation across terminals 2–3 must be specified to predict the solution waveforms, for arbitrary external circuit interconnections. The charge q_b in $R(q_b)$ for a memistor is the charge associated with the current i_b entering the terminal 3 of Widrow's 3-terminal memistor. In this sense, Widrow's memistor is incomplete and not "well posed".

In contrast, the memristor, being a 2-terminal element, does not need a second constitutive relation. This is another illustration showing why Widrow's memistor is not a memristor, since for a memristor, $v_a = R(q_a)i_a$ is uniquely defined when the waveform of i_a is given as a function of time. This is because q_a is the charge associated with the current i_a entering terminal 1 of the 2-terminal memristor.

Finally, note that *none* of the second hypothetical constitutive relation specified above would qualify Widrow's 3-terminal memistor as a basic n-terminal circuit element from nonlinear circuit theory [9, 10].

5 More Distinctions Between Memistor and Memristor

An ideal memristor [1] can be easily generalized to a broader class of nonlinear dynamical systems called *memristive* devices [3, 5], whereas the memistor has no such generalizations. The current and voltage relationship in a memristive device is defined by

$$v = \big[M(x_1, x_2, \ldots, x_n)\big]i \tag{38}$$

where M is the memristance, x_i's are the state variables, and v and i are the voltage and current, respectively; of the *2-terminal* device. The state variables of a memristive device are *internal* variables associated with the device material and its physical operating mechanism, and can *not be influenced* by any *external* voltage or current applied to a third terminal [1]. But in Widrow's memistor, if the thickness of plating is considered to be a state variable, then it is influenced by the integral of the current in the third (control) terminal, which makes it different from a memristor.

The two-terminal structure of a memristor bears a close resemblance to a biological synapse and can be used to emulate STDP (Spike Time Dependent Plasticity) [11], an important synaptic modification rule for competitive Hebbian learning, whereas the three terminals of a memistor make it unsuitable for realizing this learning rule in hardware.

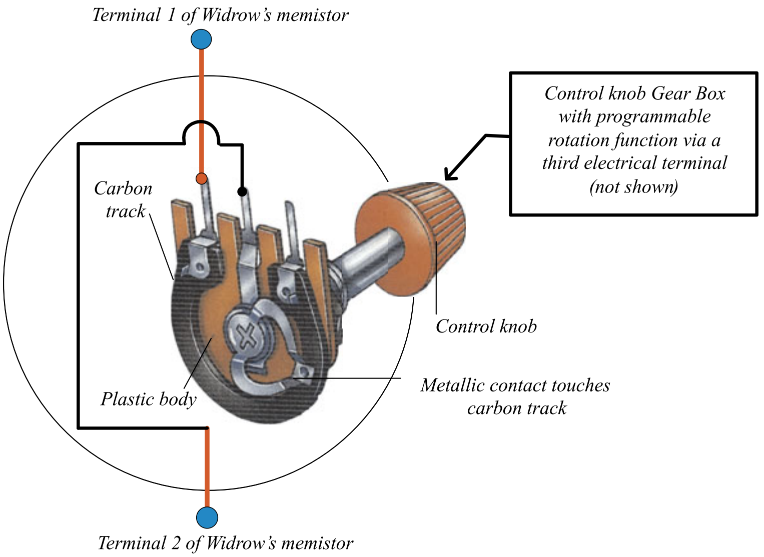

Terminal 1 of Widrow's memistor
Control knob Gear Box with programmable rotation function via a third electrical terminal (not shown)
Carbon track
Control knob
Plastic body
Metallic contact touches carbon track
Terminal 2 of Widrow's memistor

Fig. 12 Potentiometer analog of Widrow's memistor. The gear box is programmed via an *external third terminal* (not shown) to emulate the resistance $R(q_b)$ across terminals 1–2 where q_b is the time integral of $i_b(t)$ at terminal 3

It has also been shown analytically that two-element memristive circuits made of a passive linear inductor in parallel with a passive memristor, or an active memristive device, can be described explicitly by a Hamiltonian equation [12]. Moreover, an active memristive device has been shown to generate complex dynamics, such as chaos [13], and are essential components for understanding the circuit-theoretic phenomenon of the "*time-varying*" sodium conductance, and the potassium conductance in the classic Hodgin-Huxley nerve membrane model [5, 14]. None of the above unique nonlinear circuit dynamics can be emulated with memistors.

Memistor is a *linear* device irrespective of the frequency of operation, whereas a memristor is a *non-linear* device (except at infinite frequency $\omega \to \infty$) which exhibits a frequency-dependent pinched hysteresis loop. This feature of the memristor where the resistance varies under low frequency excitation but remains relatively unchanged at much higher frequencies has been utilized in memristor-based passive electromagnetic switches [15], whereas the memistor has no such capability since it does not exhibit a frequency-dependent resistance.

As a matter of fact, Widrow's memistor is conceptually equivalent to the two terminals of the potentiometer shown in Fig. 12, where the third controlling terminal is just a "gear box" coupled to the rotating shaft of the potentiometer, where the gear box is programmed to emulate $R(q_b)$, via a third electrical terminal where an external programming voltage $v_3 = v_b(t)$, or programming current $i_3 = i_b(t)$, can be applied. Such a gadget clearly bears no resemblance to a memristor.

6 Conclusion

Widrow's memistor is not a memristor because:

1. The *memistor* is a *three-terminal* device that is not well-posed, whereas the memristor is a *two-terminal* circuit element.
2. The input v–i characteristic of a *memistor* across terminal 1–2 when driven by a sinusoidal voltage *is not a pinched hysteresis loop*- which is the signature of a memristor.
3. Even if we can derive a second relationship which gives a realistic model of the internal device characteristic (i.e., the metallic sources immersed in a copper sulphate- sulfuric acid bath) inside terminal 3 of Widrow's memistor, thereby making it a well-posed circuit model, the memistor nevertheless has very limited applications because terminal 1–2 of Widrow's memistor is equivalent to a linear positive resistance for any "programmed" charge at terminal 3. In particular, unlike "locally-active" memristive devices, such as thermistor, and the Sodium memristive device in the Hodgin-Huxley model [5], *it is impossible to build an amplifier, a flip flop, or an oscillator using only resistors, inductors, capacitors, batteries, and memistors*.
4. The behavior across terminals 1–2 of Widrow's memistor can be built with 2 memristors, so it is not a "basic" circuit element, whereas it is impossible to build a memristor using only 2-terminal resistors, inductors, capacitors and Widrow's 3-terminal memistors.
5. Unlike the memristor, there is no logical circuit-theoretical definition and foundation for Widrow's *memistor*. In particular, Widrow's memistor can *not* be classified as a fundamental 3-terminal circuit element from nonlinear circuit theory [9, 10], regardless of what hypothetical constitutive relation one postulates for the third terminal.
6. The *memistor* is an *ad hoc* gadget invented for only one purpose; namely to emulate a synapse. In contrast, the memristor is defined from first principles, via an axiomatic approach involving the four fundamental circuit variables v, i, q and φ, and is applicable for many other signal processing applications.
7. The *memistor* is analogous to a *potentiometer* with a *"programmable"* control knob serving as the third terminal.
8. No one with an elementary understanding of circuit theory would regard Widrow's memistor as a "circuit element", because it is not well posed. Like the potentiometer, it is just a gadget [16].

References

1. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
2. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)

3. Chua, L.O.: Resistance switching memories are memristors. Appl. Phys. A **102**, 765–783 (2011)
4. Widrow, B.: An Adaptive ADALINE neuron using chemical memistors. Stanford Electronics Laboratories Technical Report, 1553-2 (1960)
5. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
6. Xia, Q., Pickett, M.D., Yang, J.J., Li, X., Wu, W., Riberio, G.M., Williams, R.S.: Two- and three-terminal resistive switches: nanometer-scale memristors and memistors. Adv. Funct. Mater. **21**, 2660–2665 (2011)
7. Kim, H., Sah, M.P., Yang, C., Roska, T., Chua, L.O.: Neural synaptic weighting with a pulse-based memristor circuit. IEEE Trans. Circuits Syst. I **59**(1), 148–158 (2012)
8. Chua, L.O.: Introduction to Nonlinear Network Theory, pp. 86–106. McGraw-Hill, New York (1964)
9. Chua, L.O.: Device modeling via basic nonlinear circuit elements. IEEE Trans. Circuits Syst. **CAS-27**(11), 1014–1044 (1980)
10. Brayton, R.K., Chua, L.O., Rhodes, J.D., Spence, R.: Modern Network Theory—An Introduction. PGeorgi, St. Saphorin, pp. 65–172 (1978)
11. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. **10**, 1297–1301 (2010)
12. Itoh, M., Chua, L.O.: Memristor Hamiltonian circuits. Int. J. Bifurc. Chaos **21**(9), 2395–2425 (2011)
13. Muthuswamy, B., Chua, L.O.: Simplest chaotic circuit. Int. J. Bifurc. Chaos **20**(5), 1567–1658 (2010)
14. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction in nerve. J. Physiol. **117**, 500–544 (1952)
15. Bray, M.G., Werner, D.H.: Passive switching of electromagnetic devices with memristors. Appl. Phys. Lett. **96**, 073504 (2010)
16. Kim, H., Adhikari, S.P.: Memistor is not memristor. IEEE Circuits Syst. Mag. **12**(1) (2012)

Synapse as a Memristor

Weiran Cai and Ronald Tetzlaff

Abstract The memristor, the fourth fundamental electric element, was conceptually proposed by L. Chua in 1971 and was found in laboratory late in 2008. Recently a special type of memristor was considered to be able to mimic the behavior of neural synapses. In particular, attributed to the long-term memory of weight changes, the memristor can reproduce the spike-timing-dependent plasticity (STDP) protocol of a synapse, displaying a synaptic modification related to the time interval of pre- and post-synaptic spikes. Not limited to it, we found that the memristor with adaptive thresholds can even mimic higher-order behavior of synapses, realizing the well-known suppression principle of Froemke. This type of memristor can actually express both long-term and short-term plasticities in synapses, which are responsible for the excitation level and the refractory time, respectively. The corresponding dynamical process is governed by a set of ordinary differential equations. Interestingly, the Froemke's model and our memristor-like model, based on two completely different mechanisms, are found to be quantitatively equivalent. In this chapter we would like to provide this new perspective of looking at synaptic dynamics.

1 Introduction

Memristors seem to have been existing in the nervous system for long time. In 1952, Hodgkin and Huxley proposed a compact circuit for modeling the nerve axon membrane [1], including two nonlinear electric conductances, a capacitance and a linear leakage resistance. The two nonlinear conductances are used to describe the behavior of the potassium and sodium channels in the lipid bilayer of the axon. These two elements are unusual from the existing electric elements, which exhibit a history-dependence. It was not until Chua proposed the concept of memristor in 1971 that their identities were determined: the two axon channels are memristors [2, 3]! Neurons are consist of axons and dendrites (possessing similar properties of axons) and

W. Cai (✉) · R. Tetzlaff
Fakultaet Elektrotechnik und Informationstechnik, Technische Universitaet Dresden,
Mommsenstrasse 12, 01069 Dresden, Germany
e-mail: weiran.cai@tu-dresden.de

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_7,

interconnected with junctions, known as synapses. While axons/dendrites are responsible for transmitting signals, synapses are terminals for controlling the signals. Axons and synapses are history-dependent. A natural question would then be posed: is the synapse also memristive?

Recently, an important memristor-based model for the spike-timing-dependent plasticity (STDP) in neuronal communication was proposed [4, 5], answering positively to the above question. The STDP rule is known as one of the fundamental rules of learning and memory in the brain, which explains how synapses in the nervous system adapt to the surrounding environment [6–9]. It is pointed out in [4] that the long-term plasticity of the synapse can also be expressed by a memristor, attributed to its long-term memory of state-variable (weight). The model provides a completely new way in modeling the synapse, differed from the existing behavioral model of Gerstner.

This is however not the whole story. Biological evidence has shown that the nature of the synapse is both long-term plasticity and short-term plasticity. In this chapter, we will propose an advanced memristive synapse model, which realizes both types of plasticity. Our model is an embodiment of the well known Froemke's principle for the synapse. In the following sections, we will first review Gerstner's pair-based STDP rule, as the starting point, and Lineares' memristive synapse model. We will then introduce our memristive synapse model as a realization of Froemke's principle, and prove its quantitative equivalence to his suppression model. In the discussion section, we will try to address certain relation of our model to the concept of short-term plasticity, from which the mechanism of adaptive thresholds used in our model is originated.

2 How do Neurons Work?

The brain cortex is known to be marvelously efficient. A human brain, containing 10^{11} neurons and 10^{15} synapses, consumes 20 W on average. Even a brain of a lower vertebrate excels the ultimate computer cluster of the time. A simulated mouse cortex on Blue Gene L based on Von-Neumann architecture, containing 8×10^4 virtual neurons and 5×10^{10} virtual synapses, consuming 40 kW at the speed of 1 GHz, is however still 10 times slower than a real mouse. The fact is that the individual real neurons are as slow as 10 Hz locally. It is the parallel architecture of computing and the dynamics related to neural encoding and decoding that makes the miracle.

2.1 Synapse: Bridge for Neurons

Neurons are specific cells comprising the major component of a biological cognitive system. Through a large range, signals in the form of spikes, known as the action potentials, propagate down the membrane of neural fibers in a network with a high interconnectivity to perform a style of parallel computation. Each neuron branches out two types of such fibers: axons and dendrites. Relatively long fibers branching out from a neuron, in a number of several hundred for each, are named as axons. Each axon makes connections with shorter fibers of other neurons, named as dendrites, which are in a much greater number. In fact, the contact between an axon and a dendrite is separated with a tiny cleft of around 20 nm, known as the synapse, which are responsible for controlling the signal traffic between the pre- and postsynaptic neurons. Instead of a constant conjunction, the synapse acts as an adaptive controller, depending on the history of the traffic in both directions. This adaptation in a large spatio-temporal scale is believed to be the origin of learning and memory in the brain. A pioneering work in this regard was done by the Canadian psychologist Hebb, who proposed a conjecture for the learning process in 1949 [10]:

> When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

The Hebbian rule tried to explain the synaptic modification in a sense of causality. A biologically confirmable embodiment of this rule was, however, proposed by Gerstner till 1996 [6]. His theory suggested that relative timing of pre- and postsynaptic action potentials is critical in determining synaptic modification. The above adaptation process is actually a spike-timing-dependent plasticity (STDP) : presynaptic spikes that precede postsynaptic action potentials produce long-term potentiation (LTP) , and presynaptic spikes that follow postsynaptic action potentials produce long-term depression (LTD), as illustrated in Fig. 1. The STDP rule agrees with the Hebbian rule and hence is an expression of causality in the nervous system. Confirmation of the relation between the timing of spikes and synaptic modification was thereafter made in experiments [11, 12].

2.2 *Gerstner's Pair-Based STDP Model*

The pair-based STDP model, proposed by Gerstner, et al., in 1996, is mostly considered as the first stage in the study of STDP, which considers pairs of pre- and postsynaptic action potentials as *independent* contributing units to synaptic modification. This model considers the long-term plasticity in the synapse. Spikes are here shaped as δ functions, as in the usual treatment in computational neuroscience. The change of the synaptic weight Δw depends on the arriving time of presynaptic spikes t_{pre}^{f} ($f = 1,\ 2,\ 3, \ldots, N$ counts the presynaptic spikes) and of postsynaptic spikes t_{post}^{n} ($n = 1,\ 2,\ 3, \ldots, M$ counts the postsynaptic spikes). The total weight change induced by a protocol with pairs of pre- and postsynaptic spikes is then given by the sum of all the *independent* contributions

$$\Delta w = \sum_{f=1}^{N} \sum_{n=1}^{M} W\left(t_{post}^{n} - t_{pre}^{f}\right) \tag{1}$$

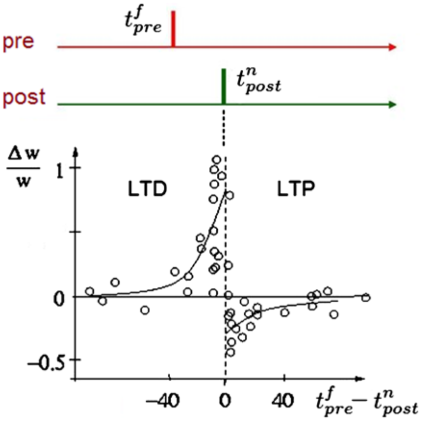


Fig. 1 Schematic of spike-timing-dependent plasticity. The STDP function expresses the change of synaptic weight as a function of the relative timing of pre- and postsynaptic spikes. The *circles* are experimental recordings, and the *solid curve* is a fitting STDP function (from [11])

where $W(x)$ denotes a STDP update function. A typically chosen STDP function $W(x)$, the contribution from one single pair, is piecewise exponential

$$W(x) = \begin{cases} A_{+}e^{-x/\tau_{+}} & \text{for } x > 0 \\ -A_{-}e^{x/\tau_{-}} & \text{for } x < 0 \end{cases} \tag{2}$$

the parameters of which can be fitted by experimental data. The synaptic modification is plotted in Fig. 1. Noticeably this is a purely empirical model. The model, however, does not explain the origin of the form of the STDP function, which is imposed empirically. We will address this problem in the next section when we introduce the memristive model.

3 Memristor Acting as a Synapse

Is the synapse memristive? Let's consider what the mathematical form of the memristor implies. Some of the properties can actually be outlined here: 1. work as a passive element (like a resistor); 2. history-dependent (depends on the state-variable); 3. polarized element (current direction cares); 4. frequency-dependent (time derivative of the state variable). All these properties exist in a synapse!

3.1 Linares' Pair-Based Memristive STDP Model

Indeed, the first memristive synaptic model was established based on such an analogy between the nonvolatility in the memristor and the long-term plasticity of the synaptic weight. In the following, we will consider the asynchronous memristive STDP model proposed by Linares, et al. [4]. The essence of this memristive STDP model is a specific type of memristor with two thresholds. The emphasis lies on

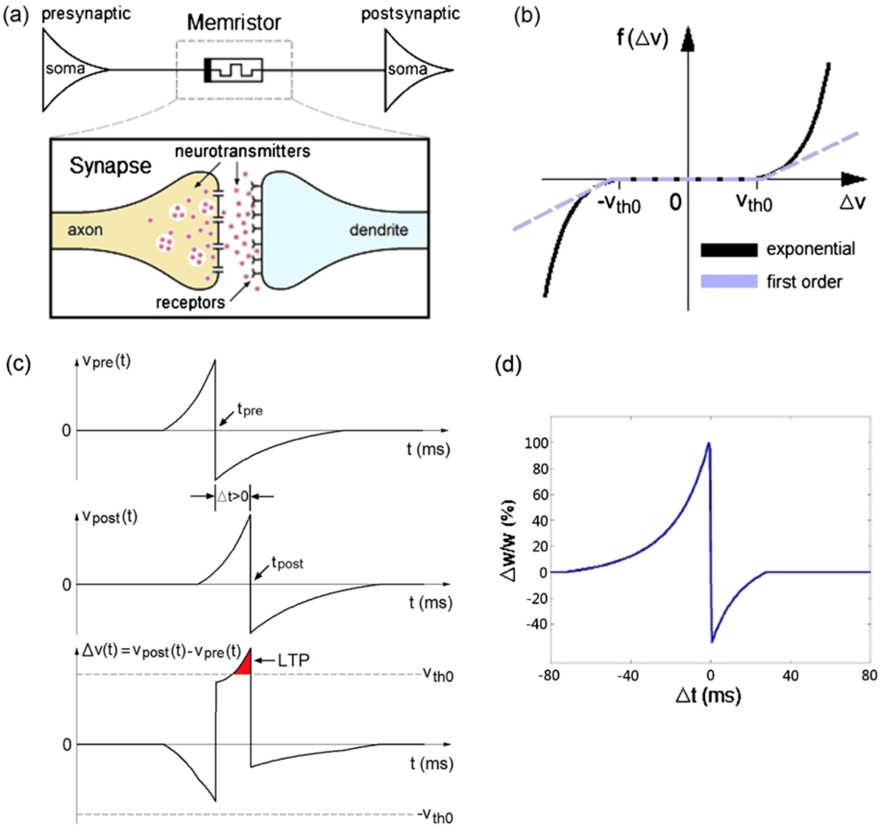


Fig. 2 Linares' STDP model. Plots are modified from [4]. (**a**) Analogy between a synapse and a memristor. (**b**) The piecewise rate function f of the synaptic modification and its first order approximation for small $\Delta v \pm v_{th0}$. (**c**) Demonstration of the pair-based model for the pre-post case (LTP). (**d**) The predicted STDP update function

two factors: the shape of action potential (non-δ function) and the thresholds determining the synaptic modification. The model assumes that the functionality of a synapse can be modeled as a memristor with two thresholds, as demonstrated in Fig. 2. The synapse is supposed to possess a memristance, which is a function of the intrinsic physical state variable of the memristor. We can suppose here that the memristance is linearly controlled by its state variable w, *i.e.*, $R(w) = k_R(w + w_0)$. The modification rate of the state variable is governed by a nonlinear function f:

$$\frac{\mathrm{d}w}{\mathrm{d}t} = -f\big(w, \Delta v(t)\big) \tag{3}$$

where $\Delta v(t)$ is the voltage across the memristor, which is analogous to the difference between the post- and presynaptic action potentials

$$\Delta v(t) = v_{pos}(t) - v_{pre}(t) \tag{4}$$

The minus sign in Eq. (3) is for the excitatory synapse, which means that the weight decreases when the pre-synaptic action potential arrives in advance of the post-synaptic action potential (vice versa for the inhibitory synapse).

The action potentials $v_{pre/pos}$ are chosen to have the form of piecewise exponential functions, as shown in Fig. 2(b) (see [4] for the definitions of parameters)

$$spk(t) = \begin{cases} A^{+}_{mp} \frac{\mathrm{e}^{t/\tau_p} - \mathrm{e}^{-t^{+}_{ail}/\tau_p}}{1 - \mathrm{e}^{-t^{+}_{ail}/\tau_p}} & \text{if } -t_p < t < 0 \\ -A^{-}_{mp} \frac{\mathrm{e}^{-t/\tau_m} - \mathrm{e}^{-t^{-}_{ail}/\tau_m}}{1 - \mathrm{e}^{-t^{-}_{ail}/\tau_m}} & \text{if } 0 < t < t_m \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The function f in [4] is chosen to be an piecewise function solely of the voltage difference

$$f(\Delta v) = \begin{cases} I_0 \operatorname{sign}(\Delta v)(\mathrm{e}^{|\Delta v|/v_0} - \mathrm{e}^{v_{th}/v_0}) & \text{if } |\Delta v| > v_{th} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Thus, the memristance (synpatic weight) can be modified only if the voltage across the memristor (synapse) exceeds the thresholds. By integrating Eq. (3), we have the total change in the state variable of the memristor

$$\Delta w(\Delta t) = \int_{\Delta t} f\big(\Delta v(t)\big) dt \tag{7}$$

where Δt is a time span of integration around the arrival of the spikes.

With these facilities, the STDP update function can be plotted by fitting to the experimental records [4, 11]. One should note that, unlike in Gerstner's model, the STDP update function is not imposed empirically in this model, but produced as a result of the thresholded memristance and the piecewise exponential action potentials. This result is based on the common property in the synapse and in the memristor: the long-term plasticity. To this point, we can say that the synapse *is* memristive! However, biological synapses behave in a more complicated way, involving both long-term and short-term plasticity. In the following section, we will consider a higher order behavior in the synapse and try to find a memristor which acts as a more real synapse.

3.2 Froemke's Triplet-Based STDP Model

Do the spike pairs really contribute independently to the total synaptic modification, as predicted by Gertner's model? The fact is that although Gerstner's pair-based STDP rule predicts synaptic change successfully for a pair of spikes, it fails to predict the case for a triplet of spikes. The contributions of spike pairs are not independent. A triplet of spikes can be a combination of either 2 pre- and 1 postsynaptic spikes (the '2/1' case) or 1 pre- and 2 postsynaptic spikes (the '1/2' case).

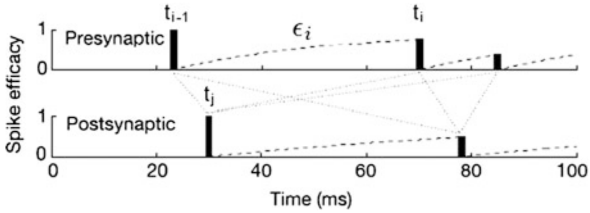


Fig. 3 The suppressed synaptic efficacy in the suppression model for triplet STDP rule of Froemke and Dan (adapted from [13]). The presynaptic spike efficacy ε_i is suppressed by the preceding presynaptic spike at t_{i-1}

According to Gerstner's model, the net synaptic modification due to a triplet of spikes should be predicted as the sum of two pairs of spikes, *e.g.*, for the pre-post-pre case, the synaptic modification should be sum of the changes from the pre-post pair, related to the time interval $\Delta t_1 = t_{post} - t_{pre,1}$, and from the post-pre pair, related to the time interval $\Delta t_2 = t_{post} - t_{pre,2}$. However, Froemke and Dan discovered from experiments that the two spike pairs do not contribute independently to synaptic modification [13]. The first spike pair plays a dominant role in the synaptic modification, and the contribution of the second pair is strongly suppressed by the presence of the preceding postsynaptic *spike*. Only when the time interval of the first pair is sufficiently large, the synaptic modification induced by the triplet could be mainly predicted by the second pair. We call this conclusion as the principle of suppression.

Froemke and Dan proposed a suppression model, which is a realization of the principle of suppression. It assumes that each spike has an 'efficacy'. The contribution of each pre/post spike pair depends not only on the interval between the pair, but also on the spike efficacy, which is suppressed by the preceding spike in each neuron (also suppose the spikes have a form of δ function). The spike efficacy is reduced to zero immediately after the preceding spike, and recovers exponentially towards one (see Fig. 3). Quantitatively, each pre- and postsynaptic spike is assigned an efficacy, which depends only on the interval from the preceding spike in the same neuron:

$$\varepsilon_i = 1 - e^{-(t_i - t_{i-1})/\tau_s} \tag{8}$$

where ε_i is the efficacy of the ith spike, t_i and t_{i-1} are the timings of the ith and $i-1$th spike, respectively, and τ_s is the suppression time constant. The contribution of each pre-post spike pair to synaptic modification is estimated as

$$w_{ij} = \varepsilon_i^{pre} \varepsilon_j^{post} F(\Delta t_{ij}) \tag{9}$$

where w_{ij} is the synpatic modification due to the ith presynaptic spike and the jth postsynaptic spike, ε_i^{pre} and ε_j^{post} are the efficacies of the two spikes, respectively, and Δt_{ij} is the interval between the two spikes, $t_j^{post} - t_i^{pre}$. The function F is the synaptic modification for an isolated spike pair:

$$F(\Delta t) = \begin{cases} A^{+}\mathrm{e}^{-\Delta t/\tau_{+}} & \text{if } \Delta t > 0 \\ -A^{-}\mathrm{e}^{\Delta t/\tau_{-}} & \text{if } \Delta t < 0 \end{cases} \tag{10}$$

where $A^{+/-}$ is the scaling factor, τ is the time constant. F is actually the pair-based STDP function. With these definitions, the synaptic modification for a triplet of spikes can be correctly predicted, which is in good accordance with experimental results. As pointed out in [13], suppression during synaptic modification may have a relation with short-term plasticity in the synapse.

3.3 Conflict with the Triplet Rule

Linares' model, the pair-based memristive STDP model, is incompatible with the triplet STDP protocol, since it is based on Gerstner's model. Actually, the memristance (synaptic weight) is additive with such choice of function f (whereas the synaptic efficacy is multiplicative). This implies that the total change of memristance is simply the sum of independent changes in all time intervals, with each consisting change depending only on $\Delta v(t)$. It has an intuitive geometric implication. To the first order of Eq. (6), the function $f(\Delta v)$ is a piecewise linear function of Δv (the dashed line in Fig. 2(c))

$$f\big(\Delta v(t)\big) \approx \frac{I_0}{v_0}\Big(\big[\Delta v(t) - v_{th}\big]_{+} + \big[\Delta v(t) + v_{th}\big]_{-}\Big) \tag{11}$$

where the notation $[\cdots]_{\pm}$ stands for the half-wave rectification functions

$$[z]_{\pm} = \begin{cases} z & \text{if } z \geq (\leq) 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Hence, for small extra voltage over the thresholds $[\Delta v(t) - v_{th}]_{+}$ and $[\Delta v(t) + v_{th}]_{-}$, by integrating Eq. (7), the total change of memristance can be viewed as the net area enveloped by the curve $\Delta v(t)$ and the thresholds, multiplied by a factor $k_R I_0/v_0$

$$\Delta R \propto A^{LTD} - A^{LTP} \tag{13}$$

where $A^{LTP} \equiv \int_{\Delta t}[\Delta v(t) - v_{th}]_{+} dt$ and $A^{LTD} \equiv -\int_{\Delta t}[\Delta v(t) + v_{th}]_{-} dt$.

With the property of additivity, let us consider the synaptic modification corresponding to a triplet of spikes, involving two pre-synaptic and one post-synpatic spikes (the 2/1 case). According to Linares' model, the total change of the memristance is the sum of an LTP produced by the pre-post pair and an LTD produced by the post-pre pair (see Fig. 4). As a special case, when the time interval between the first pair is equal to that of the second pair, the netto change of memristance sums to zero. This prediction is not in accordance with the principle of suppression we

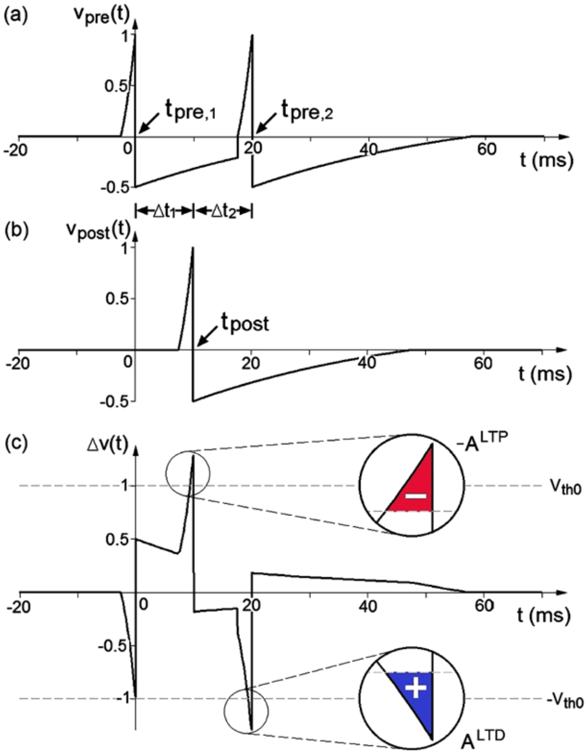


Fig. 4 The synaptic modification predicted by Linares' STDP model (the pre-post-pre case). (**a**) Pre-synaptic spikes $v_{pre}(t)$: it is assumed that the first spike turns to the second spike abruptly at the presence of the second spike; (**b**) post-synaptic spike $v_{post}(t)$; (**c**) the voltage difference function $\Delta v(t) = v_{post}(t) - v_{pre}(t)$. The parameters for the spike function $spk(t)$ are chosen as follows: $A^+_{mp} = 1$, $A^-_{mp} = 0.5$, $t^+_{ail} = 2.5$ ms, $t^-_{ail} = 37.5$ ms, $\tau_p = 3$ ms, $\tau_m = 40$ ms, the time intervals $\Delta t_1 = 10$ ms, $\Delta t_2 = 10$ ms, and the threshold $v_{th0} = 1$

introduced in the previous section. In this pre-post-pre case, the synaptic modification should mainly express as LTP, and especially, should be negative for the above special case of equal intervals ($\Delta t_1 = \Delta t_2$). The discordance implies that Linares' model needs a modification to be compatible with the triplet STDP rule.

4 Memristor Acting as a More Real Synapse

Short term plasticity, addressed by Froemke, is expressed by a virtual spike efficacy in his model. The concept solved the incompatibility problem for the case of triplet spikes. However, the spikes should by themselves just be electric signals and how could they possess certain efficacy? One would expect that this adaptation occurs rather in the synapse than in the spikes. We propose a memristive STDP model to realize the principle of suppression for the triplet rule by introducing a mechanism of adaptive thresholds.

4.1 Memristive STDP Model with Adaptive Thresholds

We introduce adaptation process in the synapse itself [14]. In this model, the type memristor with two thresholds is still adopted. However, the thresholds are not sup-

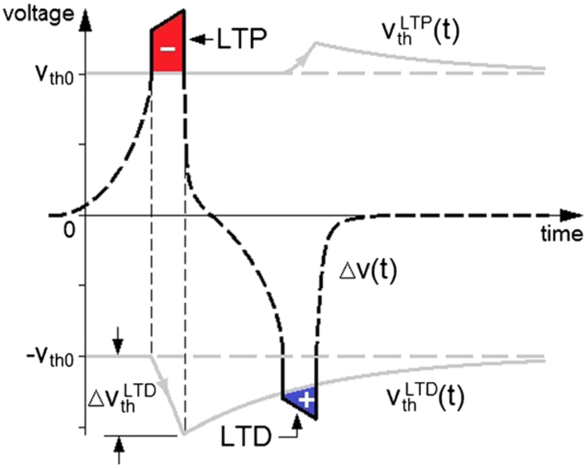


Fig. 5 The mechanism of the memristive STDP model with adaptive thresholds. The LTD area is suppressed by the raised threshold $v_{th}^{LTD}(t)$, which changes proportionally to the extent of the preceding LTP and returns exponentially to the resting value $-v_{th0}$. The suppression hence depends on the preceding LTP and the time interval between the LTD and the LTP. The function $\Delta v(t)$ is arbitrarily chosen for explanation

posed to be rigid, but can vary according to the synaptic modification. Specifically, we denote the positive threshold as the LTP threshold $v_{th}^{LTP}(t)$, and the negative threshold as the LTD threshold $v_{th}^{LTD}(t)$, both as functions of time. It is supposed that the magnitude of the LTD threshold rises by the presence of an LTP, while the LTP threshold rises by the presence of an LTD. Both thresholds restore exponentially to the resting values in the absence of any LTP or LTD. The synaptic modification is then controlled not only by $\Delta v(t)$ but also critically by the actual value of the threshold $v_{th}^{LTP}(t)$ or $v_{th}^{LTD}(t)$. In this way, an LTP suppresses the following LTD to a certain extent by raising the LTD threshold, while an LTD suppresses the following LTP in the same manner, as demonstrated in Fig. 5. This dynamical process can be described by the following ordinary differential equations:

$$\frac{\mathrm{d}v_{th}^{LTP}}{\mathrm{d}t} = \frac{1}{\tau_{th}^{LTP}}\left(-v_{th}^{LTP}(t) + v_{th0}\right) + k_{PD}\left[\frac{\mathrm{d}w}{\mathrm{d}t}\right]_+ \tag{14}$$

$$\frac{\mathrm{d}v_{th}^{LTD}}{\mathrm{d}t} = \frac{1}{\tau_{th}^{LTD}}\left(-v_{th}^{LTD}(t) - v_{th0}\right) + k_{DP}\left[\frac{\mathrm{d}w}{\mathrm{d}t}\right]_- \tag{15}$$

with the update equation for the synaptic weight

$$\tau_w \frac{\mathrm{d}w}{\mathrm{d}t} = -\left[\Delta v(t) - v_{th}^{LTP}(t)\right]_+ - \left[\Delta v(t) - v_{th}^{LTD}(t)\right]_- \tag{16}$$

where $\pm v_{th0}$ denotes the resting LTP and LTD thresholds, respectively; τ_{th}^{LTP} and τ_{th}^{LTD} are the restoring time constants for the two thresholds. The changing rates of the thresholds are supposed to be linearly dependent on dw/dt with factors k_{PD} and k_{DP} (both positive). Hence, by integration, the change in v_{th}^{LTP} or v_{th}^{LTD} is proportional to the change of w, *i.e.*, $\Delta v_{th}^{LTP(D)} \propto \Delta w$, as we have wanted:

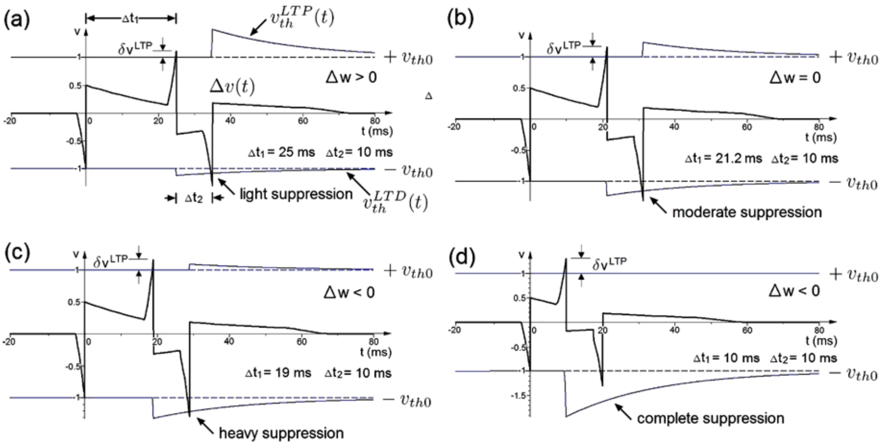


Fig. 6 Suppression by the adaptive thresholds in the 2/1 triplet spikes for different time intervals. t_2 is fixed at 10 ms, while t_1 is chosen at 25 ms (**a**), 21.2 ms (**b**), 19 ms (**c**) and 10 ms (**d**), respectively. *Black lines*: $\Delta v(t)$; blue solid: $v_{th}^{LTP}(t)$ and $v_{th}^{LTD}(t)$; *blue dashed*: $\pm v_{th0}$. Four typical suppression levels are shown: light, moderate, heavy and complete suppression, which are produced by extra voltages ordered as $\delta v_{(a)}^{LTP} < \delta v_{(b)}^{LTP} < \delta v_{(c)}^{LTP} < \delta v_{(d)}^{LTP}$. The net synaptic modification is depression for (**a**), null for (**b**) and potentiation for (**c**) and (**d**). The parameters are chosen as: $\tau_{th}^{LTP} = \tau_{th}^{LTD} = 25$ ms, $k_{PD} = k_{DP} = 10$ ms^{-1}

larger synaptic modification will cause larger variation in the threshold. For simplicity and geometric intuition, a linear synaptic update equation is used in Eq. (16).

For a triplet of spikes, e.g., in the pre-post-pre case, the first pair (pre-post) produces a positive extra voltage $\delta v^{LTP}(t) \equiv \Delta v(t) - v_{th}^{LTP}(t)$. This LTP raises the LTD threshold $v_{th}^{LTD}(t)$, so that the LTD caused by the second pair (post-pre) is suppressed by the raised threshold. Hence, the LTP caused by the first pair expresses dominantly in synaptic modification for triplet spikes.

The numerical simulation result is plotted in Fig. 6 for typical cases. An extreme case occurs when the raised threshold $v_{th}^{LTD}(t)$ grows beyond $\Delta v(t)$ so that it produces no LTD. Similar process occurs in the post-pre-post case, which causes LTD dominantly. The net potentiation and depression regions are plotted in Fig. 7 by numerical simulations with chosen parameters. It exhibits accordance with Froemke's experimental records and the prediction by the suppression model. The critical dashed lines are boundaries between LTP and LTD, corresponding to null synaptic modification or geometrically, cancellation of LTP area by LTD area. Only when Δt_1 is sufficiently large, the net synaptic modification can be predicted by the second pai aloner, because the LTD threshold is raised to a too low level to suppress the following LTD. The boundary at large Δt_1 and Δt_2 approaches the asymptotic line with a slope of -1, due to the fact that the LTP and LTD are independently additive, out of the time range of suppression.

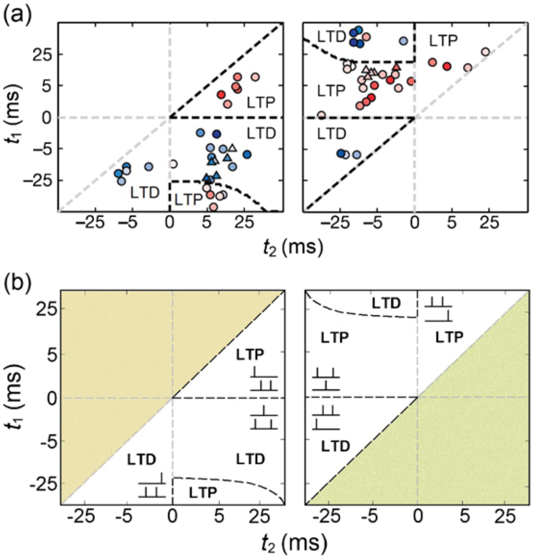


Fig. 7 Synaptic modifications for triplet spikes from experimental records and prediction of models. (**a**) Froemke's experimental data and prediction from the suppression model for '1/2' (*left*) and '2/1' (*right*) triplets. *Red symbols*: potentiation; *blue symbols*: depression. *Circles*: normal ACSF (artificial cerebrospinal fluid); *triangles*: high divalent ACSF with bicuculline. Dashed lines are the borders between potentiation and depression predicted by the suppression model. (**b**) The regions for potentiation and depression predicted by the memristive model with adaptive thresholds for '1/2' (*left*) and '2/1' (*right*) triplets. *Dashed lines* are the borders between LTD and LTP by the model. All the axes are plotted in logarithmic scale. The parameters are chosen as in Fig. 6

4.2 Quantitative Equivalency of the Models

The proposed model differs from Froemke's model in the mechanism, but they both realize the suppression principle. The following comparison proves that they are equivalent quantitatively. For our model, we estimate the suppression level on the second pair quantitatively (taking the pre-post-pre case for example). As we explained above, the change of the LTD threshold caused by an LTP, denoted as Δv_{th}^{LTD}, is approximately proportional to the change of the synaptic weight Δw, which is a function of the time interval of the first pair Δt_1. If the piecewise exponential spike function $spk(t)$ is adopted as in Eq. (5), the STDP update function $\Delta w(\Delta t)$ can be estimated in a piecewise exponential form (see Fig. 2(d))

$$\Delta w(\Delta t) = \begin{cases} -a^{+}\mathrm{e}^{-\Delta t/\tau^{+}} & \text{if } \Delta t > 0 \\ a^{-}\mathrm{e}^{\Delta t/\tau^{-}} & \text{if } \Delta t < 0 \end{cases} \tag{17}$$

Thus the total change of the LTD threshold integrated during the LTP is proportional to Δt_1 exponentially

$$\Delta v_{th}^{LTD} \propto \Delta w(\Delta t_1) = -a^{+}\mathrm{e}^{-\Delta t_1/\tau^{+}} \tag{18}$$

Thereafter, the LTD threshold restores exponentially to the resting value, which is described by

$$v_{th}^{LTD}(t) - (-v_{th0}) = \Delta v_{th}^{LTD} \mathrm{e}^{-\Delta t'/\tau_{th}^{LTD}} \propto -\mathrm{e}^{-(\Delta t_1/\tau^+ + \Delta t'/\tau_{th}^{LTD})} \tag{19}$$

where $\Delta t'$ denotes the time difference $t - t_{post}$. Suppose that the time span of the extra voltage $\delta v^{LTD}(t) \equiv \Delta v(t) - v_{th}^{LTD}(t)$ is narrow (a sharp peak), the amount of LTD is only proportional to the extra voltage at the time $t_{pre,2} = \Delta t_1 + \Delta t_2$ in Eq. (19), so we have

$$\begin{aligned} \Delta w^{LTD} &\propto -\delta v^{LTD}(t_{pre,2}) \equiv -\Delta v(t_{pre,2}) + v_{th}^{LTD}(t_{pre,2}) \\ &= -\big[\Delta v(\Delta t_1 + \Delta t_2) - (-v_{th0})\big] - S(\Delta t_1, \Delta t_2) \end{aligned} \tag{20}$$

with

$$S(\Delta t_1, \Delta t_2) = S_p \mathrm{e}^{-(\Delta t_1/\tau^+ + \Delta t_2/\tau_{th}^{LTD})}$$

The term $-[\Delta v(\Delta t_1 + \Delta t_2) - (-v_{th0})]$ in the right-hand-side of Eq. (20) is proportional to the amount of LTD according to the pair-based memristive model, and the second term $S(\Delta t_1, \Delta t_2)$ gives the suppression on the LTD according to this model. The suppression amount is determined by both Δt_1 and Δt_2.

Interestingly, this exponential dependence on the two time intervals agrees with Froemke's suppression model. According to the suppression model defined in Eqs. (8)–(10), in the pre-post-pre case, the LTD corresponding to the second pair is

$$\begin{aligned} \Delta w^{LTD} &= \varepsilon_2^{pre} \varepsilon_1^{post} F(\Delta t_2) \\ &= \big[1 - e^{-(\Delta t_1 + \Delta t_2)/\tau_s}\big] \cdot A^- \mathrm{e}^{\Delta t_2/\tau_-} \\ &= F(\Delta t_2) - \tilde{S}(\Delta t_1, \Delta t_2) \end{aligned} \tag{21}$$

with

$$\tilde{S}(\Delta t_1, \Delta t_2) = A^- e^{-(\Delta t_1/\tau_s + \Delta t_2/\tau_s')}$$

where $\tau_s' \equiv \tau_s \tau_- / (\tau_- - \tau_s)$. The first term $F(\Delta t_2)$ in Eq. (21) is the LTD according to the pair-based model, and the second term $\tilde{S}(\Delta t_1, \Delta t_2)$ gives the amount of suppression on the LTD according to the suppression model. The suppression amounts S and $\tilde{S}$ are exponentially dependent on the two time intervals Δt_1 and Δt_2 in both models. Hence, our memristive model and Froemke's suppression model are equivalent in this sense. It provides another expression of the suppression principle for the triplet STDP rule. However, the difference from the suppression model is that the memristive model imposes suppression on the following LTD or LTP caused by the adaptation process in the synapse, instead of directly on the spikes. And as another merit, the memristive model points out the origin of the STDP update function, the shape of the action potentials and the thresholds, whereas in the suppression model an ad hoc piecewise exponential STDP function is imposed.

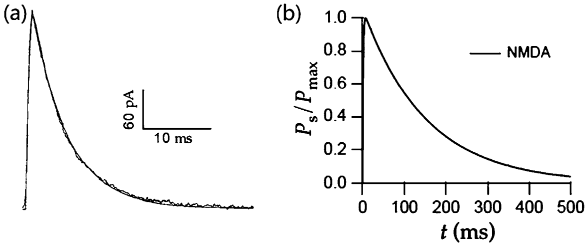


Fig. 8 Variable thresholds in the biological synapse: (**a**) the probability of release of neurotransmitters P_{rel} and (**b**) the probability of the postsynaptic channel opening P_s (from [9])

5 Short-Term Plasticity Revisited

We are not intended to identify specific biological quantities for the variable thresholds in this model, but certain relations may be addressed. Since STDP depends on NMDA receptor activation or glutamate bound for presynaptic events, and on the rise in the voltage-dependent influx Ca^{2+} concentration level and NMDA channels for postsynaptic events, the variable thresholds are correlated with the probability of release of neurotransmitters P_{rel} and the probability of the postsynaptic channel opening P_s (see Fig. 8). Both of the two probabilities reveal short-term plasticity by exhibiting a rapid rise to a high level and restoring exponentially to the resting value [9]. The short-term plasticity, pointed out by Froemke and Dan in [13], has now been expressed by the adaptive thresholds in our model (compare to the trajectories of the thresholds in Fig. 5).

We would argue that synapses with gap junctions may be regarded as a more direct correspondance to the memristive model [9, 15]. Gap junction is an electrically conductive link between two abutting neurons that is formed at a narrow gap between the pre- and postsynaptic. Especially, like the memristor, current carried by ions could travel in either direction through this type of synapse. The gap junction produces a synaptic current proportional to the difference between the pre- and post-synaptic membrane potentials, which possesses a memristive nature. The variable thresholds may hence be determined directly by inner electric properties of the synapse.

What is the common point shared by these advanced STDP models, including Froemke's synaptic efficacy model, our adaptive threshold model and also the second order memristive model proposed by Pershin and Di Ventra [16]? All these models consider both long-term plasticity and short-term plasticity. The latter determines the refractory time after spiking, and is responsible for producing specific dynamics in the nervous system. The ubiquity of the existence of refractory time in biological mechanisms is simply due to the fact that organisms need a duration of time to recover from the release of energy or discharging and to prepare for the succeeding event. In Froemke's model, the refractory time is expressed by the synaptic efficacy, while this time is expressed by the restoring time of the thresholds in our model. In the second-order memristive model in [16], the refractory time is realized by the second state variable of the memristive system. Their formulation reads as follows:

$$i = x^{-1} \Delta v \tag{22}$$

$$\dot{x} = \gamma \left[\theta(\Delta\tilde{v} - 1)\theta(\tilde{y} - 1) + \theta(-\Delta\tilde{v} - 1)\theta(-\tilde{y} - 1)\right] y \tag{23}$$

$$\dot{y} = \tau^{-1}\left[-\Delta v\, \theta(\Delta\tilde{v} - 1)\theta(1 - \tilde{y}) - \Delta v\, \theta(-\Delta\tilde{v} - 1)\theta(\tilde{y} + 1) - y\right] \tag{24}$$

where x and y are the first and second state variables, respectively. The memristance can change when y decays but is greater than a threshold y_t. The variable y excited by a certain polarity of the voltage Δv applied to the memristive system cannot be reexcited by a pulse of opposite polarity if $|y| \geq y_t$. This introduced a refractory time that forbids an immediate change in the memristance (synaptic weight) in the opposite direction, which shares a common spirit of our model that raises the opposite threshold temporarily.

The latter two models based on memristive systems are very promising for the use in the memristive neural networks since additional hardware is largely saved. But such memristive systems need to be developed. At the current time, memristor emulators, as used in [17, 18], can be regarded as a kind of straightforward implementation. However, our model gives a clear physical meaning about the device to be implemented. The necessary property of such a memristor is controllable thresholds, which can be changed with external electric signals. This can possibly be realized based on an existing type of memristor or on a new physical mechanism.

References

1. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. **117**(4), 500–544 (1952)
2. Chua, L.O.: Memristor-the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
3. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**, 209–223 (1976)
4. Zamarreño-Ramos, C., Camuñas-Mesa, L.A., Pérez-Carrasco, J.A., Masquelier, T., Serrano-Gotarredona, T., Linares-Barranco, B.: On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. Front. Neurosci. **5**(26) (2011)
5. Pérez-Carrasco, J.A., Zamarreño-Ramos, C., Serrano-Gotarredona, T., Linares-Barranco, B.: On neuromorphic spiking architectures for asynchronous STDP memristive systems. In: Proceedings of IEEE ISCAS, pp. 1659–1662 (2010)
6. Gerstner, W., Kempter, R., van Hemmen, J.L., Wagner, H.: A neuronal learning rule for sub-millisecond temporal coding. Nature **383**, 76–78 (1996)
7. Gerstner, W.: Spiking neurons. In: Maass, W., Bishop, C.M. (eds.) Pulsed Neural Networks. MIT Press, Cambridge (1999)
8. Kempter, R., Gerstner, W., van Hemmen, J.L.: Hebbian learning and spiking neurons. Phys. Rev. E **59**, 4498–4514 (1999)
9. Dayan, P., Abbott, L.F.: Theoretical Neuroscience. MIT Press, Cambridge (2001)
10. Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory. Wiley, New York (1949)
11. Bi, G.Q., Poo, M.M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. J. Neurosci. **18**(10), 464–472 (1998)
12. Wang, H.X., Gerkin, R.C., Nauen, D.W., Bi, G.Q.: Coactivation and timing-dependent integration of synaptic potentiation and depression. Nat. Neurosci. **8**, 187–193 (2005)

13. Froemke, R.C., Dan, Y.: Spike-timing-dependent synaptic modification induced by natural spike trains. Nature **416**, 433–438 (2002)
14. Cai, W., Tetzlaff, R.: Advanced memristive model of synapses with adaptive thresholds. In: International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), Turin (2012)
15. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: Principles of Neural Science, 4th edn. McGraw-Hill, New York (2000)
16. Pershin, Y.V., Di Ventra, M.: Neuromorphic, digital, and quantum computation with memory circuit elements. Proc. IEEE **100**, 2071–2080 (2012)
17. Pershin, Y.V., Di Ventra, M.: Experimental demonstration of associative memory with memristive neural networks. Neural Netw. **23**, 881–886 (2010)
18. Pershin, Y.V., Di Ventra, M.: Practical approach to programmable analog circuits with memristors. IEEE Trans. Circuits Syst. I, Regul. Pap. **57**, 1857–1864 (2010)

Memristors and Memristive Devices for Neuromorphic Computing

Patrick Sheridan and Wei Lu

Abstract Memristors are an important emerging technology for memory and neuromorphic computing applications. In this chapter, we review the fundamentals of the memistor framework developed by Leon Chuan nearly 40 years ago, and examine resistive switching phenomena as the quintessential example of physical memristive systems. A special focus is given to the hardware emulation of biological synapses using memristors and groundbreaking results in the field are reviewed. Future research directions with spiking neural networks are outlined and the exciting prospect of emergent behavior in memristor networks is discussed.

1 Introduction

The exponential growth in computing power over the past half century has revolutionized nearly every aspect of society from communication to medicine. This impact has largely been fueled by the continued shrinking of transistors that control the flow of information in computing hardware. However, this trend is increasingly challenged by the technological limits related to device nonidealities, power consumption and the ability of traditional hardware designs to tackle the data deluge and massively parallel requirements of today's computing tasks. Going beyond *Moore's Law*, which has guided the doubling of transistor counts approximately every two years, novel nanoelectronic technologies offer not only a possibility of continued transistor scaling with decreased size and power consumption, but also new computing paradigms like *neuromorphic* designs, which take inspiration from the incredible computing power of the brain.

In this chapter, we will review the mathematical underpinnings of the memristor and more generally, memristive systems in order to better understand the framework in which resistive switching phenomena is best understood. We then discuss

P. Sheridan · W. Lu (✉)
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA
e-mail: wluee@eecs.umich.edu

P. Sheridan
e-mail: sheridp@umich.edu

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_8,

two classes of material systems that display continuously tunable, analog memristive behavior, focusing particularly on anion (e.g. O_2^- or oxygen vacancy) based devices. Potential uses for memristive devices are discussed and key experimental results are presented that demonstrate the core concepts in emulating biological computing/learning rules for neuromorphic applications. Two important network architectures are reviewed to understand how memristors can be interfaced with conventional systems. Finally, emergent and self-organizing behaviors on the network level that suggest a radical paradigm shift in computing are reviewed.

2 Mathematical Definition

2.1 Memristor—Strict Definition

The memristor—a portmanteau of *memory* and *resistor*—was conceptualized in 1971 by Leon Chua [1], who noticed that, while there were circuit elements relating voltage and current, charge and voltage, and current and magnetic flux—namely the resistor, capacitor, and inductor, respectively—there was no circuit primitive relating charge and magnetic flux. Chua termed this hypothetical element the *memristor* and described it mathematically as:[1]

$$\frac{d\phi(q)}{dq} = M(q) \tag{1}$$

$$\frac{dq}{dt} = I(t) \tag{2}$$

where

ϕ is the magnetic flux, q is the electric charge, and $I(t)$ is the current
M is called the *incremental memristance*

Although it remains experimentally challenging to find a device that fits this linear memristor description, the central concept—that the device resistance is determined by an internal state variable (in this case q) which is not solely governed by the instantaneous inputs (voltage or current) but rather evolves according to a dynamic rate equation (Eq. (2))—allows the development of the unified memristive systems framework. Nearly 40 years later, this framework was found to be able to explain various resistive switching phenomena [2, 3]. It now provides a robust mathematical foundation for the understanding, design, and application of memristive devices.

[1] The equation listed is for a charge-controlled memristor; [1] includes a definition for flux-controlled memristors as well.

2.2 Memristive Systems

A few years after Chua's original publication, the concepts of memristors were generalized to describe an expanded class of devices known as *memristive systems* by Chua and Kang [4]. Rather than being restricted to a single state-variable and a linear, flux or charge-driven equation, the concept of memristive systems provides a much more generalized framework to describe any physical system that is governed by an (or a set of) internal state-variable(s), $\vec{s}$ (denoted as a vector). So the input-output relation can be written as

$$\vec{y}(t) = g(\vec{s}, \vec{u}, t)\vec{u}(t) \tag{3}$$

while the rate equations can be written as

$$\frac{d\vec{s}}{dt} = f\left(\vec{s}, \vec{u}(t), t\right) \tag{4}$$

where

$\vec{u}(t)$ represents some input (e.g. voltage) to the system, and
$\vec{y}(t)$ represents the output (e.g. current).

A key characteristic of the memristive systems is that the internal state variable, $\vec{s}$, cannot normally be determined directly from the instaneous inputs. Rather, only the dynamic rate equation (4), of the state variable can be determined. While superficially simple, by choosing the proper dynamic rate equation (4) (which depends only on the current state, the instantaneous input—and not previous inputs, and possibly the time), and the input-output relation (3), systems described by this framework can exhibit complex, hysteretic behaviors that were only explained phenomenologically previously.

It has now been accepted that all resistive switching devices [5] can be modeled in the memristive system framework by choosing the right state variable and input-output and rate equations. This allows for an intuitive understanding of the switching dynamics and the ability to predict the evolution of the state variable as well as a straightforward way to develop device models for circuit-level simulation programs such as SPICE. Following the spirit of [4] and for convenience purposes, we will use the name "memristor" to represent the generalized memristive systems throughout the discussion.

3 Material Systems

Resistive switching systems can be characterized by the physical phenomena that give rise to their memristive behavior and can be broadly categorized as unipolar and bipolar, depending on the voltage polarity of the set and reset processes. While there are many interesting results and applications in both categories, in the interest

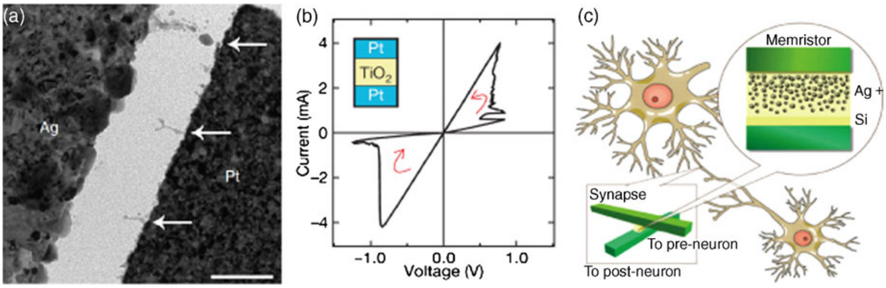


Fig. 1 Example of single memristive device formed at the intersection of two nanowire electrodes. (**a**) TEM image showing conductive filaments in nanoscale gap between electrodes (scale bar is 200 nm) [14]. (**b**) Current-voltage hysteresis loop, characteristic of memristor devices [2]. (**c**) Schematic of device used to emulate biological synapse [15]

of space, this chapter focuses on bipolar memristive devices where resistive switching is driven by electrical inputs (although sometimes facilitated by heat as well) [6]. In such devices, the resistive switching effect can be explained by the formation/dissolution of a conducting region inside an otherwise insulating medium, and the state variable is typically the size (e.g. area or length) of the conducting region. Below we discuss two different types of devices where the conducting regions are formed by the injection and accumulation of cations (e.g. metal ions) and anions (e.g. oxygen ions or oxygen vacancies in a metal oxide), respectively.

3.1 Cation Migration

Under the influence of a strong electric field (on the order of 10^7 V/cm) metal atoms can be ionized and become mobile within an insulating material. This forms the basis of operation for memristors based on cation (metal ion) migration. The device structure used in these memristors is typically that of a metal-insulator-metal (MIM) or metal-insulator-semiconductor (MIS) stack as shown in Fig. 1. The insulating material serves to isolate the top and bottom electrodes when the device is in the *OFF*, or high-resistance, state and can be composed of many materials, notably chalcogenides [7, 8], SiO_2 [9], HfO_2 [10], ZrO_2 [11], or amorphous-Si [12, 13]. When sufficient voltage is applied to the top active metal electrode, metal ions can be disassociated and migrate into the insulating matrix and eventually form a conductive path that electrically connects the two electrodes either through a physical conductive bridge, or by narrowing the gap enough to allow electron tunneling.

Devices based on metal ion migration typically require an initial forming step as well as higher programming voltages because of the relatively high energy barriers that must be overcome to cause ionization and ion movement [8, 16]. In operation, the devices often exhibit *digital* behavior with abrupt resistance changes. Digital switching is often advantageous for conventional memory applications; such devices are termed conductive-bridge random access memory (CBRAM) [17]. However, by careful material engineering (e.g. by doping an amorphous Si film with Ag atoms), incremental *analog* switching, which is more useful for neuromorphic application has also been obtained [18].

3.2 Anion Migration

In transition metal oxides, the mobility of oxygen vacancies (missing oxygen atoms, analogous to 'holes' in semiconductor electronics) can be much higher than that of the metal ions, so the resistive switching can be dominated by anion migration in such materials. In this case, an active metal electrode is no longer necessary and in fact should be avoided. Like the metal ion migration based devices, a MIM or MIS structure is used with the insulating material typically a transition-metal oxide such as TiO_2, WO_3, HfO_2, or Ta_2O_5, among many others [2, 18–20]. These materials often contain structural defects where an oxygen atom is not available to bond with the metal, formed either during deposition or through an electroforming process. Materials with a significant number of defects are denoted using an 'x' in the chemical formula, HfO_{2-x}, for example. The redistribution of the oxygen vacancies changes the local conductivity of the film [18], and creates the conduction channels in the oxide.

Similar to the cation migration based devices, both abrupt digital switching and incremental analog switching devices can be obtained in the anion migration based devices as seen in Fig. 2. However, given that the oxygen vacancies are 'intrinsic' to the film, which eliminates the harsh forming process, and that the activation energy for anion migration is typically lower than that for cations, controlled analog switching is more easily obtained in anion migration based devices which makes them well suited for memristor-based neurmorphic systems.

3.3 Modeling

All resistive switching devices can be modeled by mapping the physical processes that underpin their operation to the memristive framework set out in Eqs. (3), (4) [4]. The key is in identifying a state variable that accurately captures the important aspects of the system and then describing its dynamic, time evolution in response to the memristor's inputs. The first use of the memristive framework to describe resistive switching appeared in [2], characterizing the behavior of TiO_2 devices. Here, the memristor was composed of two adjacent regions, doped and undoped with oxygen vacancies, with a movable boundary between the regions while keeping the overall film thickness fixed as shown in Fig. 3(a). This was modeled as two resistors, R_1, R_2, connected in series, with $R_1 < R_2$ where R_1 represents the resistance of the doped region and R_2, the undoped region. The state variable, w,

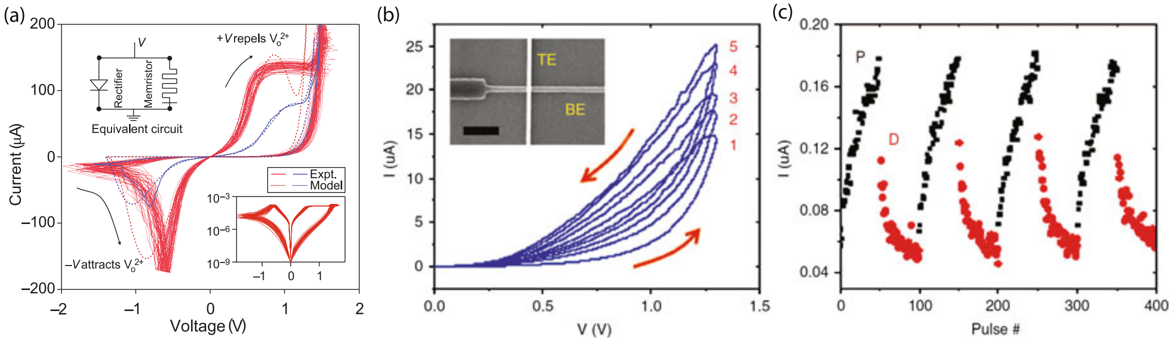


Fig. 2 Analog switching in anionic devices. (**a**) Hysteretic DC sweeps in TiO_2device [21]. (**b**) Incremental programming in WO_3device, SEM image in inset [22]. (**c**) Positive (*black*) and negative (*red*) pulse programming of device in (**b**) [22]

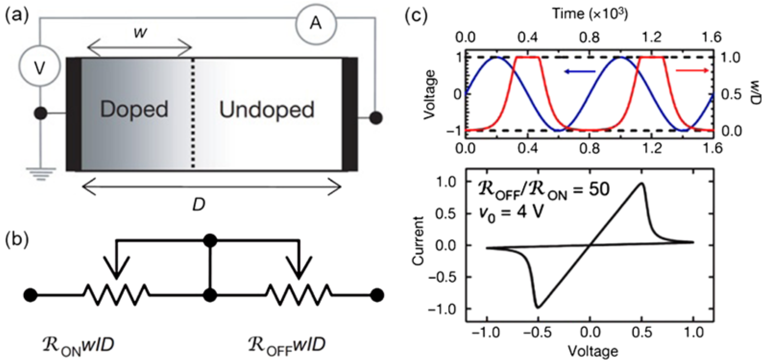


Fig. 3 (**a**) Schematic device model of doping-modulated device, (**b**) modeled as two series resistors. (**c**) Time dependent voltage-current relationship. Reprinted with permission from [2]

represents the length of the doped region. The governing equations for the system are given as:

$$v(t) = \left(R_{ON}\frac{w(t)}{D} + R_{OFF}\left(1 - \frac{w(t)}{D}\right)\right)i(t) \tag{5}$$

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{ON}}{D} i(t) \tag{6}$$

where:

$v(t)$ and $i(t)$ are the voltage and current, respectively
R_{ON} and R_{OFF} are respectively the doped and undoped resistances
D is total device length, and μ_v is the voltage dependent vacancy mobility

Although the original HP model successfully linked the resistive switching phenomena with the memristor model and captures many key features such as the pinched-hysteresis and frequency dependence, the two-resistor-in-series model is too simplified and did not quantitatively capture the switching behavior well. In particular, for metal-oxide memristors based on anion migration, evidence suggests that the incremental conductance increase is caused by the increase of the conduction region area, rather than its length and an improved model based on conduction region area is given in [23]. Because of the relatively high mobility of oxygen vacancies in transition metal oxides, it is easy for vacancies to traverse the oxide, and thus, many conductive paths are formed in parallel. In aggregate, this can be viewed as a conductive region with variable width (or equivalently, area or number of conduction paths, Fig. 4). This is essentially modeled as two resistors in parallel (Fig. 4(b)), one with a high conductance representing the oxygen vacancy-rich region, and the other a low conductance for the oxygen vacancy-poor region, with the state variable representing the relative width of the vacancy-rich region:

$$i(t) = (1 - w)\alpha\big(1 - \exp\big(-\beta v(t)\big)\big) + w\gamma \sinh\big(\delta v(t)\big) \tag{7}$$

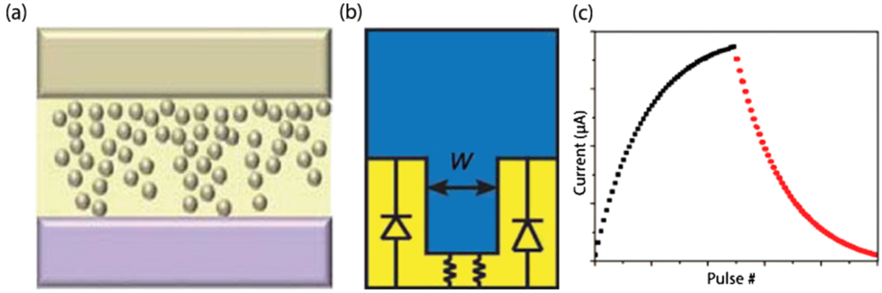


Fig. 4 Model of conductive path width modulation. (**a**) Schematic and (**b**) model showing parallel conductive paths [23]. (**c**) Simulated programming using positive (*black*) and negative (*red*) pulses [24]

$$\frac{dw}{dt} = \lambda \sinh\bigl(\eta v(t)\bigr) \tag{8}$$

where

w is the state variable
$\alpha, \beta, \gamma, \delta, \lambda, \eta$ are fitting parameters

Note here the conduction region growth rate, $\lambda \sinh(\eta v(t))$, does not depends on the state variable itself due to the parallel growth nature, in contrast with the length-growth model. Additionally, a decay term, $-\frac{w}{\tau}$, can be introduced to the dynamic equation (8) to account for the spontaneous decay of the conduction region (e.g. due to diffusion), so Eq. (8) can be written as

$$\frac{dw}{dt} = \lambda \sinh\bigl(\eta v(t)\bigr) - \frac{w}{\tau} \tag{9}$$

This spontaneous decay term causes the state variable to lose its state and can be naturally used to emulate short-term memory in biological systems, as will be discussed in Sect. 4.2.2

4 Synaptic Plasticity

The two-terminal nature of memristor devices and their ability to change conductance when stimulated with voltage pulses and evolve into states based on input history make them almost ideal electronic analogues of biological synapses. In addition, memristors can be readily implemented in a crossbar architecture, which offers the adaptivity, high density, and large connectivity needed for neuromorphic applications [25]. While memristors can be used to store a static weight as an analog resistance value in, for example, conventional artificial neural networks, anionic based devices offer more complex internal dynamics that can be used to implement different bio-mimicking learning rules as well. Therefore, in the following section, following a brief overview of synaptic weight storage in memristor memories, we will focus primarily on *spiking* neural networks, which are more efficient than non-spiking networks and whose incorporation of time allows the model to naturally make use of the rich internal dynamics found in vacancy based devices.

4.1 Memristors as Weight Storage

A key feature of neural networks is the ability to compute with memory, where computation is distributed and performed with localized memory storage. Memristor memory arrays, with the ability to provide high density, random access and analog storage, can be integrated locally with computing elements thus providing the most flexibility in network learning algorithms. In this context, the memristive array behaves like a conventional memory: weight values are calculated outside of the array and then programmed to the correct addresses; the weights are then retrieved when needed by the algorithm. If the cells have a natural decay, the weight values must be periodically refreshed in a manner similar to DRAM. The benefit to this approach is its flexibility as virtually all existing neural network algorithms can be implemented with the right design. For example both local and non-local learning algorithms, such as the popular *backpropagation of error*, can be implemented in a straightforward way. The approach, however, does not take full advantage of the high connectivity, or internal dynamics offered by these devices.

4.2 Synapse Emulation

A more elegant but challenging approach to memristor-based neuromorphic systems is to use memristors to directly implement the learning rules. In general, biologically plausible network operation requires that the learning rules use only information that is local to the synapse. For example, in spiking neural networks, the weight update algorithms should depend only on the firing times and rates of the pre- and post-synaptic neurons and internal synaptic variables. The idea, known as the *Hebb Postulate*, after Donald Hebb [26], is commonly summarized as 'cells that fire together, wire together.' The results are attractive from a circuit integration standpoint since it limits the length of signal routing for parameter calculations and precludes the need for global clock distribution.

A general model for localized spiking neuronal plasticity is given by [27]:

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{ji}(t) = a_0 + a_1 S_i(t) + a_2 S_j(t) + a_3 S_i(t)\bar{S_j}(t) + a_4 \bar{S_i}(t) S_j(t) \qquad (10)$$

where:

w_{ji} is the synaptic weight between neuron i and neuron j

$S_k(t)$ is the spike train of the k'th neuron (given as a sum of time-delayed Dirac deltas)
$\bar{S_k}(t)$ is the low-pass filtered version of $S_k(t)$

The particular choices of the constants $a_0, \ldots, a_4$ determine whether the network exhibits Hebbian or anti-Hebbian learning, memory decay, facilitation/inhibition, *etc*. It should be noted that these could also be functions of w_{ji} in order to implement long-term potentiation/depression, or other complex behaviors.

4.2.1 Spike Timing Dependent Plasticity

One prominent example of a biologically plausible learning rule is that of *spike timing dependent plasticity (STDP)*. This Hebbian-derived learning rule specifies that the synaptic efficacy is increased when a pre-synaptic neuron spikes before a post-synaptic neuron, while the efficacy is decreased if, in the reverse situation, the pre-synaptic neuron spikes after the post-synaptic neuron. Furthermore, the closer in time the two spikes, the greater the change as seen in Fig. 5.

Using a common exponentially decaying STDP model, the change in synaptic strength for a single pair of pulses can be calculated with the following equation, given by [28]:

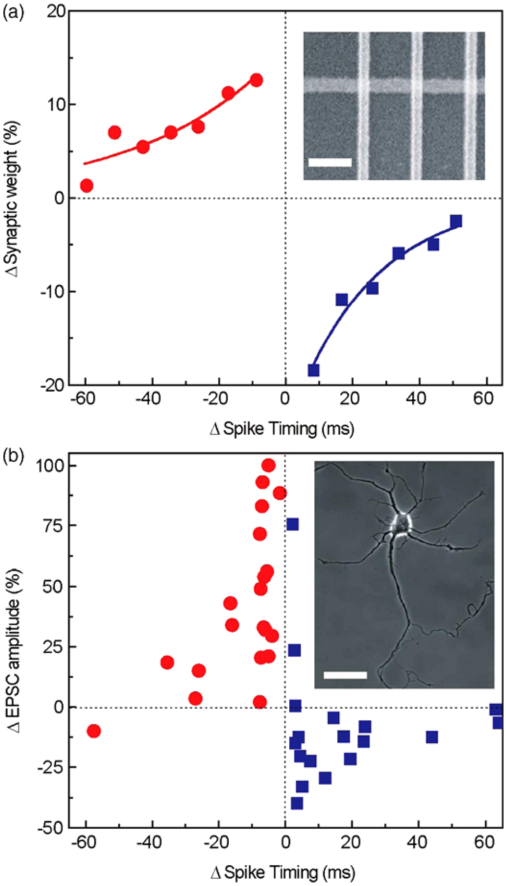
$$\Delta w_{ij} = \begin{cases} A_+ \exp(-\Delta t/\tau_+) & \text{if } \Delta t > 0 \\ A_- \exp(-\Delta t/\tau_-) & \text{if } \Delta t < 0 \end{cases} \tag{11}$$

where

Δw_{ij} is the amount by which weight w_{ij} is changed
$\Delta t = t_{post} - t_{pre}$, the delay between the post- and pre-synaptic neuron firing
$A_+, A_-, \tau_+, \text{and} \tau_-$ are fixed parameters, with $A_+ \geq 0$ and $A_- \leq 0$

STDP has been demonstrated in memristive systems using carefully designed pulse patterns to effect the proper conductance change [29]. In general, pulses are sent from the pre- and post-synaptic neurons to both terminals of the memristor. The pulses are of such an amplitude that, arriving alone, each will have minimal effect on the state of the memristor; arriving together, however, their overlap in time will cause a measurable change. The task is then to design circuitry that converts the relative timing of the pre- and post-synaptic spikes into pulses, modulating either the pulse width or pulse shape, to create an overlap which changes the memristor to different conductance states. Additionally, in order for this approach to work, the memristance change needs to be nonlinear in voltage, i.e. exhibit a threshold effect. This non-linear behavior is indeed typical in experimentally demonstrated memristor devices, as the ion motion is, to first order, exponentially dependent on the applied electric field, as seen from Eq. (8) To convert the relative timing information into different pulse shapes, an approach like the one shown in Fig. 6 is typically used [30, 31], which shows the pre- and post-synaptic pulses as well as the resultant waveform as seen by the memristor. The relative timing between the pulses

Fig. 5 Demonstration of STDP in WO_x (**a**), and rat hippocampal neurons (**b**). Here ΔSpike Timing = $t_{pre} - t_{post}$. Reprinted with permission from [15] which sourced (**b**) from Bi G.Q., Poo M.M., J. Neurosci. 18(24) (1998) and (**b**, inset) from Kaech S., Banker G., Nat. Protoc. 1(5) (2006)



determines the polarity and degree of overlap, and thus the direction and magnitude of the conductance change. By choosing different pulse shape designs, previous work has investigated the implementation of different types of STDP rules including complementary-square, triangle, and exponentially decaying wavelets [25, 31].

4.2.2 Short-Term Plasticity

Beyond first-order changes in their conductance in response to an applied external signal, some memristor devices exhibit complex internal dynamics that demonstrate interesting and potentially useful behaviors that occur on a shorter timescale. Ag_2S and WO_3 memristors, for example, have been shown capable of reproducing short-term dynamic responses known as *short-term plasticity* [32, 33].

In biological systems, short-term plasticity, both potentiation and depression, usually occurs on the timescale of milliseconds, in contrast to the more permanent changes associated with long-term potentiation and depression where the effects can

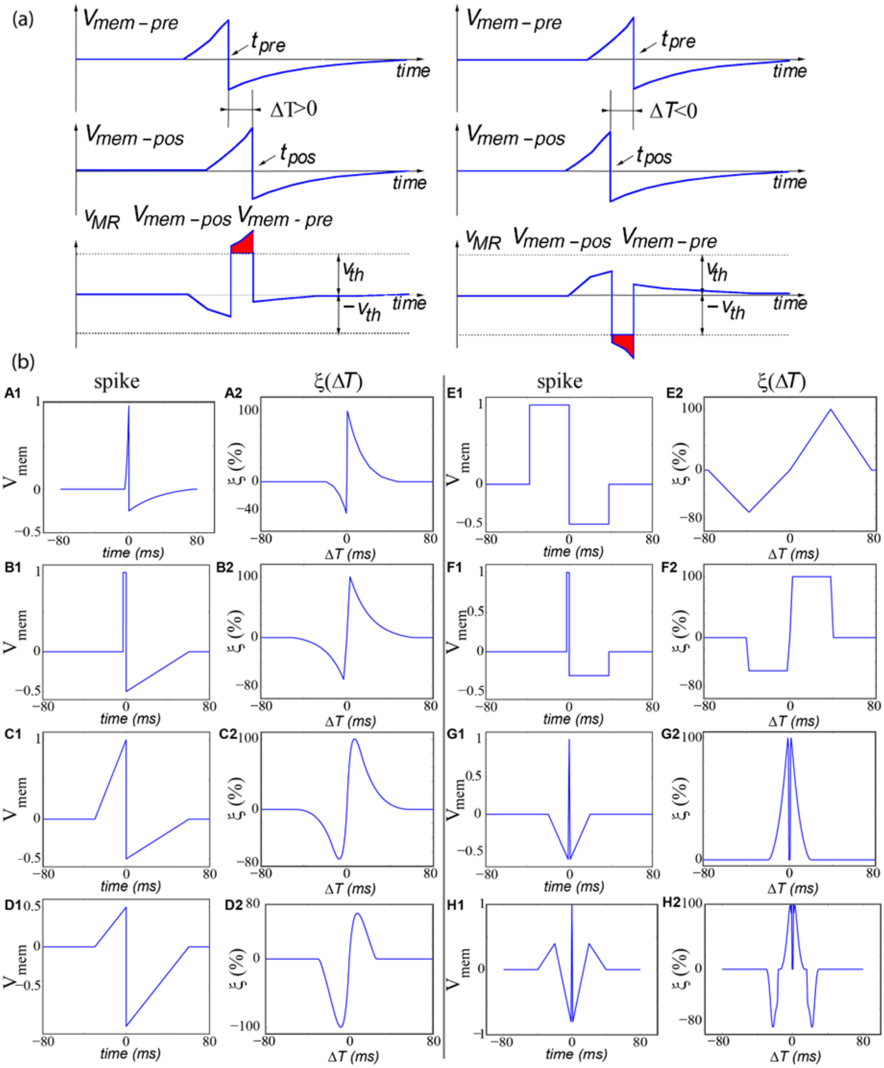


Fig. 6 Example waveforms implementing STDP on a memristive element. (**a**) The *first* and *second rows* show pulses emitted by pre- and postsynaptic neurons, respectively, while the *third row* shows the combined voltage signal seen by the memristor. The *dotted line* represents a threshold beyond which the memristor changes its state significantly. Potentiation results when the presynaptic neuron fires before the postsynaptic neuron, while depression occurs in the reverse situation. (**b**) Effect of waveform shape on weight change ($\xi(\Delta T)$) Reprinted with permission from [31]

range on the order of minutes, days, or even years, in the case of long-term memory. Specific examples of dynamic responses involving short-term plasticity include post-tetanic potentiation and paired-pulse facilitation [33]. The general concept in both examples is that, given a series of closely grouped pre-synaptic pulses (even

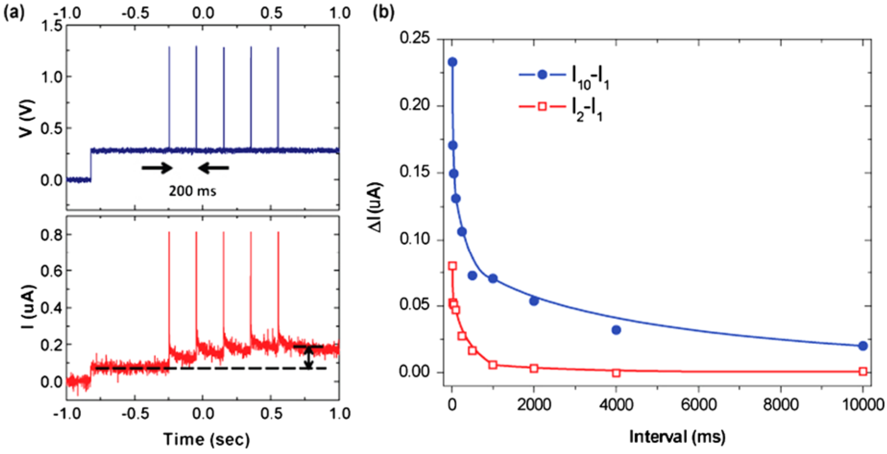


Fig. 7 Dynamic synapse responses. (**a**) Rate dependent plasticity and (**b**) paired pulse programming. Adapted with permission from [33]

just two, in the paired-pulse case), the pulses occurring later in the group have a greater effect on the post-synaptic potential than do early pulses. Figure 7 shows this experimentally; latter pulses deliver more current to the post-synaptic neuron than earlier pulses. This is commonly measured as the ratio of the post-synaptic neuron's response to a latter pulse over the its response to an earlier pulse. Though more often reported as an increase in the post-synaptic neuron's firing probability (a ratio greater than one), post-tetanic and paired-pulse depression have also been observed.

In biology, short-term plasticity strongly affects learning effects. For example, if the information is conveyed by the mean spike rate, the influences of short-term plasticity is found to critically depend on the range of spike frequencies.

The key to implementing dynamic synapses in neuromorphic hardware is in identifying material systems that have a natural weight decay mechanism. In oxygen vacancy based memristors, the decay results from the spontaneous diffusion of vacancies in the absence of an applied voltage. This has the effect of dissolving the conductive channels and the redistribution of vacancies within the insulating matrix. This effect can be modeled with the inclusion of a decay term in the memristive equations, as shown in Eq. (9). In general, if a_0 in Eq. (10) is negative, the memristive synapse gradually loses its programmed state.

Both facilitation and inhibition can be obtained in memristors exhibiting short-term plasticity. Facilitation emerges as a competition between the decaying weight and increase in synaptic strength as a result of repeated programming. Therefore, synaptic efficacy will exhibit a frequency dependence with higher frequency spike trains having a greater effect on post-synaptic membrane potential than lower frequency trains, even when the number of spikes is constant, as shown in Fig. 7. Inhibition can be demonstrated in some material systems if the undisturbed state of the memristor is not the lowest conductance state. In this case, negatively biased spikes will decrease the conductance of the device, while diffusion will bring the conductance back to an intermediate level.

4.2.3 Long-Term Potentiation and Depression

An obvious problem with synaptic decay is that the neural network will eventually forget what it has learned from the training phase. Fortunately, both biological and memristive systems can transition from the short-term plasticity effects discussed in the previous section, to a long term regime known as *long-term potentiation/depression (LTP/LTD)* so useful information can be stored more permanently in the network. In oxygen vacancy based materials, this phenomenon is attributed to an accumulation of oxygen vacancies to a level such that diffusive decay is less effective as migrating vacancies are likely to enter other conductive channels so outdiffusion from different channels is effectively balanced out [33]. Figure 8 demonstrates that as the device becomes more programmed (as a function of the number and frequency of input pulses), the decay time-constant increases and thus the device will lose its state more slowly.

5 Hardware Topology

5.1 Crossbar Architecture

To realize the benefits of memristor devices, ideally they will be implemented in the *crossbar array* architecture which offers high density, random access and large connectivity, as shown in Fig. 9. The configuration offers the highest packing density—$4F^2$ where F is the minimum feature size—and can be fabricated over underlying circuity to save space. Further, the regular pattern lends itself to novel fabrication techniques such as nanoimprint lithography [34] or self-organization using diblock copolymers [35].

The topology of the crossbar is particularly well suited to feed-forward artificial neural networks. The network can be mapped into hardware by connecting an input neuron at each top electrode (horizontal lines in Fig. 9) and an output neuron at each bottom electrode (vertical lines in Fig. 9). In this configuration, each input neuron is individually connected to every output neuron, with each connection implemented as a memristive element. The memristors regulate the connection between the neurons, serving as the synapses in the network.

One challenge for implementing the crossbar configuration is the leakage current around devices in the high-resistance state; this is known as the *sneak-path problem* as shown in Fig. 9(b). Given a crossbar array of devices, suppose a high-resistance memristor, device A, is to be measured by applying a voltage across the device and measuring the current. Undesirably, an additional current is able to flow through

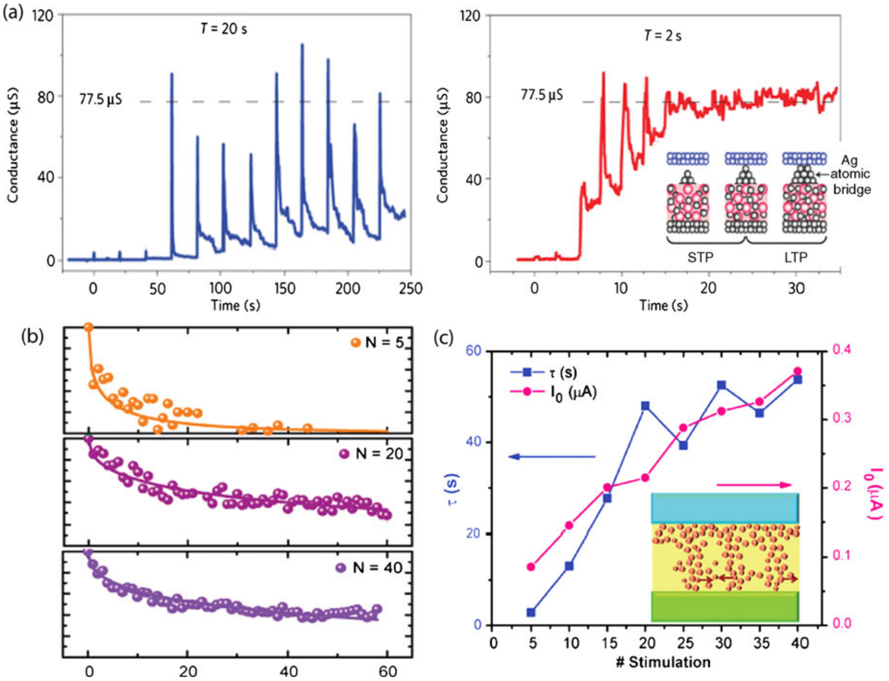


Fig. 8 Transition from short term to long term memory. (**a**) Frequency dependence of filament strengthening in Ag_2Smemristor, schematic shown in inset [32]. (**b**) Retention readings after a variable number (N) programming pulses; data scaled by initial read current. (**c**) Intrinsic decay constant, τ, versus stimulation number resulting from structural changes (inset). Adapted with permission from [33]

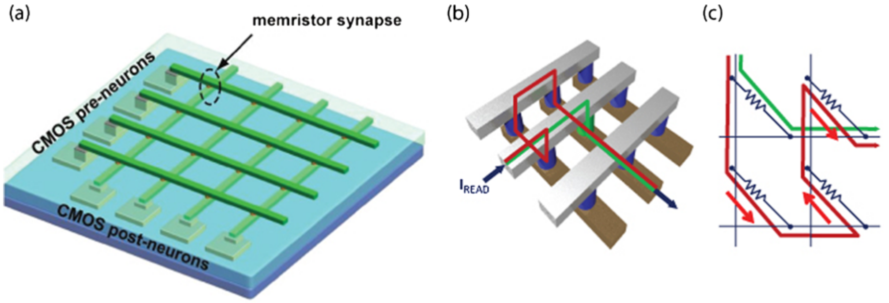


Fig. 9 (**a**) Crossbar array. Each memristor forms a synapse between a pre- and postsynaptic neuron [15]. (**b**) Sneak path problem in crossbar and (**c**) electrical equivalent. Desired current path shown in *green*, while sneak path shown in *red* [37]

devices having low-resistance in the interconnected passive array; (one such possible path is highlighted in Fig. 9(b)). This additional current can yield an erroneous measurement of the state of *A*. Possible solutions to the sneak-path problem include a selection device to block current in the reverse direction [36], complementary device configurations [37], current nonlinearity [38, 39], and various read schemes [8, 40]. It should be noted that the sneak path problem worsens with increased array size and the problem is more severe for analog arrays where different resistance values need to be read out accurately (digital memory arrays can tolerate small read margins more easily).

5.2 Hybrid Memristor/CMOS Circuitry

Memristors can be integrated directly on top of traditional complementary metal oxide semiconductor (CMOS) technologies and connected through vias in an approach known at CMOL [41], shown in Fig. 10(a), (b). By building up, rather than out, the circuit footprint is reduced while at the same time shortening signal path lengths, which, by decreasing capacitance, reduces power consumption and increases operating speed. Furthermore, using this approach allows for smaller arrays which can be tiled while minimizing the amount of external decoding circuitry needed [41, 42]. Using smaller arrays partially alleviates sneak path problem because it limits the possible number of parallel sneak paths that can form while also increasing read margins by limiting sources of off-state leakage current. Prototype hybrid memristor/CMOS circuits have been demonstrated successfully (Fig. 10(c), (d)) for memory and logic [34, 43], although multi-layer stacks with distributed interconnect pins remain to be developed.

5.3 Emergent Behavior

Under certain conditions, even a network consisting of simple elements interacting through a set of limited rules can display complex and interesting behaviors; the results are said to *emerge* from the network. The behavior is typically visible only at a macroscale and is often hard to predict from the microscale interactions. Emergence is found often in natural systems with the brain being a leading example. The cooperative actions of the brain's billions of neurons, each acting locally, self-coordinate to produce the complex range of behaviors observable in organisms.

The synaptic emulation capabilities of memristors make them an ideal platform to implement neuromorphic systems that exhibit emergent behavior. In particular, simulation studies have shown the use of memristive devices in self organizing maps [45], associative memories [46], and cellular automata [47]. Using a broad definition, these examples demonstrate emergence through the self-coordinated efforts of many devices using simple rules like STDP, and are able to produce complex behavior that is 'more than the sum of its constituent parts' in an undirected way.

Emergence evolves out of responses to the rules of microscale interactions as well as macroscale pressures exerted from the network. Requirements like sparsity

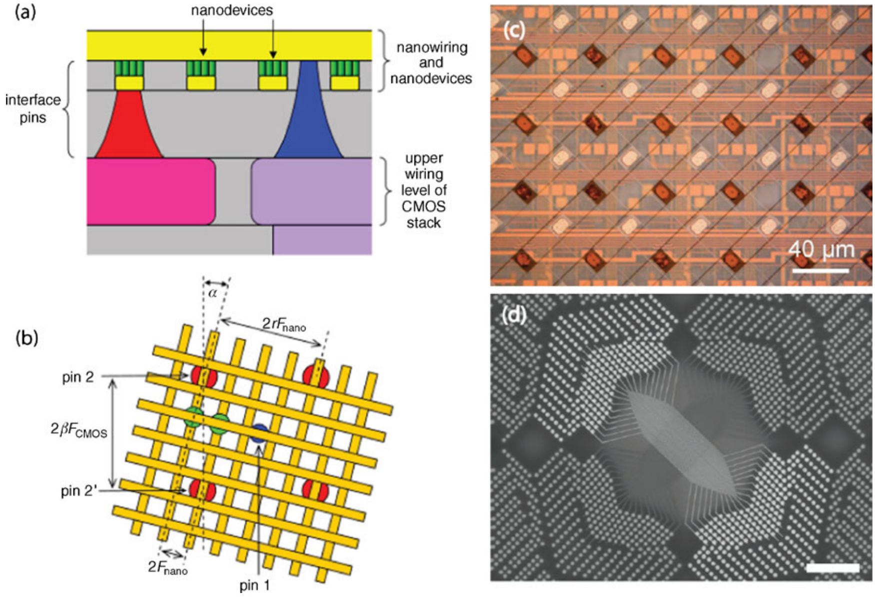


Fig. 10 CMOL integration of nanoscale devices on top of CMOS. Schematic (**a**) cross-sectional and (**b**) top view with underlying CMOS addressing [41]. Physical realizations in (**c**) TiO_2 [44] and (**d**) Ag/a-Si [43] (5 μm scale bar) material systems

[48], homeostasis [45], or competition and interference [49], coupled with the right interactions, can create an environment such that the network will tend to evolve to a useful or desired state. By defining an energy or error function, network evolution can often be cast as an optimization problem.

The advent of so-called 'third generation' neural networks, those that incorporate neuron spike timing in an integral manner, promise to greatly increase the computing power of neural networks, especially for temporal processing problems like audition or motion analysis. Dynamic synapses, whose state and behavior is defined by more than a single-valued weight, are thought to be an essential part of these networks. Using oxygen vacancy based memristors with complex internal dynamics that mimic the behavior of biological synapses offers a way to realize the computing power of the brain in low power, emergent manner. However, though great strides have been made in simulation [45, 47, 50], demonstrating emergent behaviors in memristor-based neural networks remains to be realized and will likely become the next milestone in memristor research.

Going forward, a key challenge is identifying applications that can benefit from the added realism that dynamic synapses provide. Earlier work has demonstrated that networks using dynamic synapses are more computationally powerful than networks with static (changing slowly and only during learning) weights [51], but adoption has been slow for several reasons. Networks with complex synaptic behavior are harder to train because there are more tunable parameters per synapse. Dynamic synapses often incorporate several variables, modeling Ca^{+} dynamics, for example, each with an associated decay time-constant. While [52] has shown how gradient descent can be used to tune these parameters, it is not yet clear how to do this is in a biologically plausible manner. Since synapses are far more numerous than neurons in a neural network, calculating synaptic dynamics also drastically increases the amount of computation required to simulate a network on conventional computing resources.

6 Conclusion

Memristors are an important emerging technology that will have a strong impact on future computing development, both at the device, circuit, and architectural levels. In this chapter, we have reviewed the mathematical principles of memristors in general and then focused on anion migration-based devices as an example of experimentally-demonstrated devices. This is followed by a discussion of the interesting, synapse-like behaviors that arise from the internal dynamics of material systems displaying analog behavior.

The further development of memristor technologies for conventional applications driven by an obvious market need, like non-volatile memory, will in turn aid novel architecture research. Improvements in device performance in key metrics like switching time, power consumption, on-off ratios, and retention will make neuromorphic design easier and more amenable to conventional analysis. On the other hand, research for memory applications may tend to overlook the unique advantages offered by material systems with intrinsic internal decay for applications like dynamic synapses. Nonetheless, continued interest in bio-inspired computing will likely make resistive switching technologies an important area of research during the next decade.

We believe memristor-based neuromorphic computing is well positioned to revolutionize computer architectures and the way we handle non-traditional computing problems. Emergent behavior suggests a way forward in artificial intelligence design where, taking inspiration from the brain, neural networks are self-organized with massively parallel processing and orders of magnitude greater connectivity than is possible with traditional transistor technology alone. Progress will be made through collaborative efforts of neuroscientists, network theorists, material scientists, electrical and computer engineers, and computer scientists to bring together diverse fields for gains in computing and neuromodeling.

References

1. Chua, L.: Memristor-the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
2. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**(7191), 80–83 (2008)

3. Prodromakis, T., Toumazou, C., Chua, L.: Two centuries of memristors. Nat. Mater. **11**(6), 478 (2012)
4. Chua, L.O., Mo Kang, S.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
5. Chua, L.: Resistance switching memories are memristors. Appl. Phys. A **102**(4), 765–783 (2011)
6. Waser, R., Aono, M.: Nanoionics-based resistive switching memories. Nat. Mater. **6**(11), 833–840 (2007)
7. Kozicki, M.N., Park, M., Mitkova, M.: Nanoscale memory elements based on solid-state electrolytes. IEEE Trans. Nanotechnol. **4**(3), 331–338 (2005)
8. Valov, I., Waser, R., Jameson, J.R., Kozicki, M.N.: Electrochemical metallization memories—fundamentals, applications, prospects. Nanotechnology **22**(25), 254003 (2011)
9. Schindler, C., Puthen Thermadam, S.C., Waser, R., Kozicki, M.N.: Bipolar and unipolar resistive switching in Cu-doped SiO_2. IEEE Trans. Electron Devices **54**(10), 2762–2768 (2007)
10. Wang, Y., Liu, Q., Long, S., Wang, W., Wang, Q., Zhang, M., Zhang, S., Li, Y., Zuo, Q., Yang, J., et al.: Investigation of resistive switching in Cu-doped HfO_2 thin film for multilevel non-volatile memory applications. Nanotechnology **21**(4), 045202 (2010)
11. Guan, W., Long, S., Liu, Q., Liu, M., Wang, W.: Nonpolar nonvolatile resistive switching in Cu doped ZrO_2. IEEE Electron Device Lett. **29**(5), 434–437 (2008)
12. Jafar, M., Haneman, D.: Switching in amorphous-silicon devices. Phys. Rev. B **49**(19), 13611 (1994)
13. Hyun Jo, S., Lu, W.: Cmos compatible nanoscale nonvolatile resistance switching memory. Nano Lett. **8**(2), 392–397 (2008)
14. Yang, Y., Gao, P., Gaba, S., Chang, T., Pan, X., Lu, W.: Observation of conducting filament growth in nanoscale resistive memories. Nat. Commun. **3**, 732 (2012)
15. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. **10**(4), 1297–1301 (2010)
16. Jo, S.H., Kim, K.-H., Lu, W.: Programmable resistance switching in nanoscale two-terminal devices. Nano Lett. **9**(1), 496–500 (2009)
17. Kund, M., Beitel, G., Pinnow, C.-U., Rohr, T., Schumann, J., Symanczyk, R., Ufert, K.-D., Muller, G.: Conductive bridging ram (cbram): an emerging non-volatile memory technology scalable to sub 20 nm. In: Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International. pp. 754–757. IEEE Press, New York (2005)
18. Yang, J.J., Strukov, D.B., Stewart, D.R.: Memristive devices for computing. Nat. Nanotechnol. **8**(1), 13–24 (2013)
19. Baek, I.G., Lee, M.S., Seo, S., Lee, M.J., Seo, D.H., Suh, D.-S., Park, J.C., Park, S.O., Kim, H.S., Yoo, I.K., et al.: Highly scalable nonvolatile resistive memory using simple binary oxide driven by asymmetric unipolar voltage pulses. In: Electron Devices Meeting, 2004. IEDM Technical Digest. IEEE International. pp. 587–590. IEEE Press, New York (2004)
20. Govoreanu, B., Kar, G.S., Chen, Y., Paraschiv, V., Kubicek, S., Fantini, A., Radu, I.P., Goux, L., Clima, S., Degraeve, R., et al.: 10×10 nm^2 Hf/HfO_x crossbar resistive RAM with excellent performance, reliability and low-energy operation. In: Electron Devices Meeting (IEDM), 2011 IEEE International, pp. 31–36. IEEE Press, New York (2011)
21. Yang, J.J., Pickett, M.D., Li, X., Ohlberg, D.A.A., Stewart, D.R., Williams, R.S.: Memristive switching mechanism for metal/oxide/metal nanodevices. Nat. Nanotechnol. **3**(7), 429–433 (2008)
22. Chang, T., Yang, Y., Lu, W.: Building neuromorphic circuits with memristive devices. IEEE Circuits Syst. Mag. **13**(2), 56–73 (2013)
23. Chang, T., Jo, S.-H., Kim, K.-H., Sheridan, P., Gaba, S., Lu, W.: Synaptic behaviors and modeling of a metal oxide memristive device. Appl. Phys. A **102**(4), 857–863 (2011)
24. Chang, T., Sheridan, P., Lu, W.: Modeling and implementation of oxide memristors for neuromorphic applications. In: 13th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), 2012, pp. 1–3. IEEE Press, New York (2012)
25. Snider, G.S.: Cortical computing with memristive nanodevices. SciDAC Rev. **10**, 58–65 (2008)

26. Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory, new edn. Wiley, New York (1949)
27. Ponulak, F., Kasinski, A.: Introduction to spiking neural networks: information processing, learning and applications. Acta Neurobiol. Exp. **71**(4), 409 (2011)
28. Song, S., Miller, K.D., Abbott, L.F.: Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nat. Neurosci. **3**(9), 919–926 (2000)
29. Snider, G.S.: Spike-timing-dependent learning in memristive nanodevices. In: IEEE International Symposium on Nanoscale Architectures, 2008. NANOARCH 2008, pp. 85–92. IEEE Press, New York (2008)
30. Kuzum, D., Jeyasingh, R.G.D., Lee, B., Wong, H.-S.P.: Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. Nano Lett. **12**(5), 2179–2186 (2011)
31. Zamarreño-Ramos, C., Camuñas-Mesa, L.A., Pérez-Carrasco, J.A., Masquelier, T., Serrano-Gotarredona, T., Linares-co, B.: On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. Front. Neurosci. **5** (2011)
32. Ohno, T., Hasegawa, T., Tsuruoka, T., Terabe, K., Gimzewski, J.K., Aono, M.: Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. Nat. Mater. **10**(8), 591–595 (2011)
33. Chang, T., Jo, S.-H., Lu, W.: Short-term memory to long-term memory transition in a nanoscale memristor. ACS Nano **5**(9), 7669–7676 (2011)
34. Xia, Q., Yang, J.J., Wu, W., Li, X., Williams, R.S.: Self-aligned memristor cross-point arrays fabricated with one nanoimprint lithography step. Nano Lett. **10**(8), 2909–2914 (2010)
35. Muthukumar, M., Ober, C.K., Thomas, E.L.: Competing interactions and levels of ordering in self-organizing polymeric materials. Science **277**(5330), 1225–1232 (1997)
36. Park, W.Y., Kim, G.H., Seok, J.Y., Kim, K.M., Song, S.J., Lee, M.H., Hwang, C.S.: A Pt/TiO$_2$/Ti Schottky-type selection diode for alleviating the sneak current in resistance switching memory arrays. Nanotechnology **21**(19), 195201 (2010)
37. Linn, E., Rosezin, R., Kügeler, C., Waser, R.: Complementary resistive switches for passive nanocrossbar memories. Nat. Mater. **9**(5), 403–406 (2010)
38. Kim, K.-H., Jo, S.H., Gaba, S., Lu, W.: Nanoscale resistive memory with intrinsic diode characteristics and long endurance. Appl. Phys. Lett. **96**(5), 053106 (2010)
39. Liang, J., Wong, H.-S.P.: Cross-point memory array without cell selectors–device characteristics and data storage pattern dependencies. IEEE Trans. Electron Devices **57**(10), 2531–2538 (2010)
40. Yu, S., Liang, J., Wu, Y., Wong, H.P.: Read/write schemes analysis for novel complementary resistive switches in passive crossbar memory arrays. Nanotechnology **21**(46), 465202 (2010)
41. Likharev, K.K., Strukov, D.B.: Cmol: devices, circuits, and architectures. In: Introducing Molecular Electronics, pp. 447–477. Springer, Berlin (2005)
42. Strukov, D.B., Williams, R.S.: Four-dimensional address topology for circuits with stacked multilayer crossbar arrays. Proc. Natl. Acad. Sci. **106**(48), 20155–20158 (2009)
43. Kim, K.-H., Gaba, S., Wheeler, D., Cruz-Albrecht, J.M., Hussain, T., Srinivasa, N., Lu, W.: A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. Nano Lett. **12**(1), 389–395 (2012)
44. Xia, Q., Robinett, W., Cumbie, M.W., Banerjee, N., Cardinali, T.J., Yang, J.J., Wu, W., Li, X., Tong, W.M., Strukov, D.B., et al.: Memristor-cmos hybrid integrated circuits for reconfigurable logic. Nano Lett. **9**(10), 3640–3645 (2009)
45. Querlioz, D., Bichler, O., Gamrat, C.: Simulation of a memristor-based spiking neural network immune to device variations. In: The 2011 International Joint Conference on Neural Networks (IJCNN), pp. 1775–1781. IEEE Press, New York (2011)
46. Pershin, Y.V., Di Ventra, M.: Experimental demonstration of associative memory with memristive neural networks (2009). arXiv:0905.2935
47. Itoh, M., Chua, L.O.: Memristor cellular automata and memristor discrete-time cellular neural networks. Int. J. Bifurc. Chaos **19**(11), 3605–3656 (2009)

48. Zylberberg, J., Murphy, J.T., DeWeese, M.R.: A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1 simple cell receptive fields. PLoS Comput. Biol. **7**(10), e1002250 (2011)
49. Hermiz, J., Chang, T., Du, C., Lu, W.: Interference and memory capacity effects in memristive systems. Appl. Phys. Lett. **102**(8), 083106 (2013)
50. Zhao, W., Querlioz, D., Klein, J.-O., Chabi, D., Chappert, C.: Nanodevice-based novel computing paradigms and the neuromorphic approach. In: IEEE International Symposium on Circuits and Systems (ISCAS), 2012, pp. 2509–2512. IEEE Press, New York (2012)
51. Maass, W., Zador, A.M.: Dynamic stochastic synapses as computational units. Neural Comput. **11**(4), 903–917 (1999)
52. Natschlger, T., Maass, W., Zador, A.: Efficient temporal processing with biologically realistic dynamic synapses. Netw. Comput. Neural Syst. **12**(1), 75–87 (2001)

Bio-inspired Neural Networks

Andy Thomas and Christian Kaltschmidt

Abstract We describe a biological network and the principal mechanisms that are responsible for learning and memory. We start with a description of the morphology of these networks and their components, such as neurons and synapses. Then, we will identify crucial components of the information processing, such as ion flux and the induced mechanisms, e.g., long-term potentiation and depression. Next, we will compare the behaviour of a memristive system with the mechanisms identified in biological systems and present corresponding experiments and a few simulations. Finally, we will present more abstract ways of using memristors to solve complex problems.

1 Introduction

Memristors have attracted great interest for a variety of applications in recent years. An obvious use would be as a memory device [17, 50, 52] or, more ambitiously, a reconfigurable logic device [10, 11, 64, 88, 89]. However, the most interesting implementation of memristive devices is *neuromorphic computing*.

Neuromorphic computing aims to use biological mechanisms operating within the brain as a blueprint to construct novel computer architectures. Carver Mead built the foundation of this field and proposed large-scale adaptive analogue systems because of their robustness as well as good power efficiency [61]. The efficiency of these systems is particularly promising, as shown in Table 1.

Despite its power efficiency and robustness, some tasks are very challenging for the human brain, e.g., solving coupled differential equations. However, it is very

A. Thomas (✉)
Physics Department, Bielefeld University, Bielefeld, Germany
e-mail: andy.thomas@uni-bielefeld.de

A. Thomas
Physics Department, Osnabrück University, Osnabrück, Germany

C. Kaltschmidt
Biology Department, Bielefeld University, Bielefeld, Germany
e-mail: c.kaltschmidt@uni-bielefeld.de

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_9,

Table 1 Comparison of the power consumption of three different ‘technologies’ [73]. A biological neuron draws less power and consumes less area than a digital computer or silicon neuron

	Digital computer	Silicon neuron	Biological neuron
Energy consumption (J/spike)	10^{-5}	10^{-8}	10^{-11}
Size (μm^2)	10^8	3×10^3	10

easy to find a solution for this problem with the help of a von-Neumann/Zuse computer [85, 92]. However, some tasks are also difficult for a state-of-the-art computer system, e.g., driving a car through traffic in Rome. Therefore, a novel, fundamentally different computer system would be useful. These systems would be based on more brain-like principles and could excel at certain tasks. In particular, the new computer system could have implemented parallel analogue processing.

2 Biological Mechanisms

In the following sections, we describe a biological network and the principal mechanisms that are responsible for learning and memory. We start with a description of the morphology of these networks and their components, such as neurons and synapses. Then, we will identify crucial components of the information processing, such as ion flux and the induced mechanisms, e.g., long-term potentiation and depression. Next, we will compare the behaviour of a memristive system with the mechanisms identified in biological systems and present corresponding experiments and a few simulations. Finally, we will present more abstract ways of using memristors to solve complex problems.

2.1 Connectome (Wires and Neurons)

One aspect of the complexity of nervous systems is their intricate morphology, particularly the interconnectivity of their neuronal processing elements. Neural connectivity patterns have long attracted the attention of neuroanatomists and are now the focus of large-scale projects such as the human connectome project [84]. Researchers hope to unravel neuronal connections to understand personality. Two methods have been used: measuring neuronal parameters such as axon diameters in vivo by nuclear magnetic resonance (NMR) or reconstruction of connections from serial sections magnified by electron microscopy.

2.2 Synapses (Resistance and Biochemistry)

Synapses are the ends of the connections within the nervous system. There are two types of synapses: chemical synapses, which use neurotransmitters, and electrical

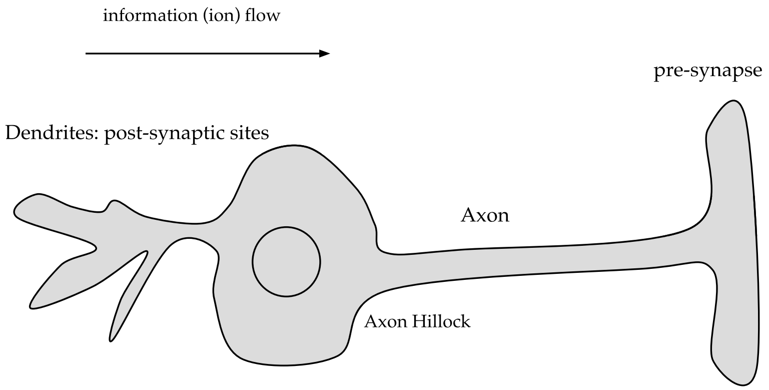


Fig. 1 Neuron in a schematic view: Dendrites (input, via post-synaptic sites) are on the *left*; the long cable is called the axon and ends with a pre-synapse

synapses, which provide direct electrical coupling to the synapsed cell. Neurons are very polarised cells, with long (μm to m) and thin (in the range of 100 nm) extensions (Fig. 1).

In a cable model, the neuronal extensions would be axons and dendrites originating from the neurons. At the end of these cables, chemical synapses could be formed. The underlying biochemistry involves neurotransmitter-gated ion channels. The neurotransmitters are stored in synaptic vesicles (see Fig. 2) from which they are released into the synaptic cleft. This special type of connection has a direction of use. Input is sensed mainly via the dendrites, and output is transmitted via the axon. Depolarisation in the form of ion flux (e.g., sodium ions) travels from the soma along the axon to its endpoint, the pre-synapse.

Vesicles are stored within this pre-synapse region that contain neurotransmitters, such as glutamate for excitatory, positively acting neurons. When a neuron fires, it generates action potentials, which can be detected as spikes with a height of several tens of millivolts. These potentials ultimately lead to vesicle fusion and neurotransmitter release, which diffuse out into the synaptic cleft. The neurotransmitter can then interact with a receptor at the postsynaptic side. Many of these neurotransmitter receptors are ligand-gated ion channels. Thus, the release of a transmitter leads to ion flux from the extracellular site to the post-synapse. Numerous biochemical feedback mechanisms operate on the synapse.

Activation of the gene expression programme via transcription factors ultimately leads to long-lasting changes that are thought to be responsible for memory formation. Analogous to computer memory, transcription factors activate specific addresses at the genome. The genes at these addresses could function to initiate further gene expression programmes, such as a subprogramme. Alternatively, gene products could be directly involved in memory formation by directing the growth of new connections. For example, NF-kappa B activation is necessary for the growth of an axon in vitro and in vivo [36].

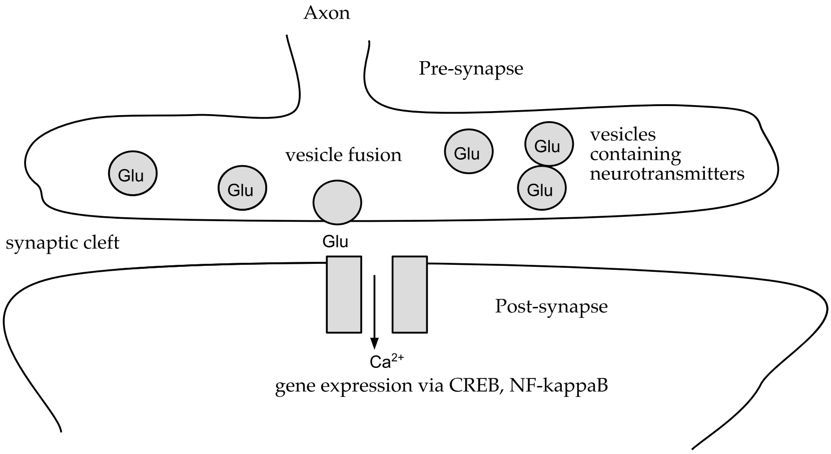


Fig. 2 Details of a chemical synapse. Neurotransmitters such as glutamate (Glu) are stored in synaptic vesicles at the pre-synapse, which is located at the end of an axon. An axon can form more than one synapse; these additional synapses are omitted for the sake of simplicity. When an action potential travels through the axon towards the synapse, vesicle release results in the diffusion of neurotransmitters within the synaptic cleft. These neurotransmitters can interact with neurotransmitter receptors such as the NMDA receptor (*grey rectangle*), which opens a selective ion channel, in this case, for calcium ions. During learning, long-term potentiation of the synapse activates this calcium signalling, which in turn activates kinases. These activate transcription factors such as CREB and NF-kappaB [41, 53, 59]

In addition to calcium, the post-synaptic glutamate receptors are permissive for sodium ions, which generate a membrane potential. Thus, the influx of ions (e.g., sodium ions) can lead to the propagation of a changed membrane potential (see charge propagation section below). In a reductionist view, synapses can be viewed as resistors with tuneable resistance (memristors).

2.3 Charge Propagation and Re-amplification (Ion Flux)

Charge propagation in neurons can be described by the Hodgkin-Huxley (H-H) theory of the action potential. Hodgkin and Huxley were the first to record an action potential using squid axons. Using fine glass electrodes, they recorded potentials within the squid axon, which was facilitated by the large size of these axons (approximately 1 mm in diameter) [34].

An action potential is described as a short voltage change from negative membrane potential to positive values, followed by a rapid return to the base level (see Fig. 3). This action potential is elicited when the potentials at the post-synapses (e.g., dendrites) reach a given threshold.

Signal flows from the dendrites to the cell body to a region at which an axon initiates (axon hillock, see Fig. 1). There are at least two types of electrical signals

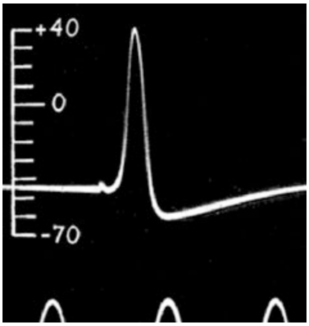


Fig. 3 Action potential recorded by Hodgkin and Huxley from a squid axon. Note the rapid depolarisation from approximately -40 mV to approximately $+40$ mV and the rapid decline with an overshooting baseline. The time marks on the lower part of the image are two milliseconds. Adapted by permission from Macmillan Publishers Ltd: Nature [34], copyright 1939

in the neuron: *graded potentials* and all-or-none *action potentials*. Graded potentials travel short distances, are additive, and degrade with distance from the point of origin. Action potentials can self-regenerate and thus do not degrade as they travel down an excitable cell surface and are not additive. Voltage-gated sodium channels are responsible for this property of action potentials.

The cell body and dendrites lack voltage-gated sodium channels, and thus, they have only graded post-synaptic potentials. Because they have the potential for many synapses, their effects can be cumulative (graded potentials are additive). If the signal is strong enough when it reaches the axon hillock, voltage-gated sodium channels are opened, initiating an action potential that travels faithfully down the axon. If the signal is not strong enough to open these channels, the signal dies. Hodgkin and Huxley deduced that the first phase of the action potential is a rapid inward current of Na^+ ions. This phase is followed by a slower outward current of K^+ ions. Furthermore, they concluded that the membrane had a selective permeability for both ions and presented their data in a quantitative model (see below). Most spiking models continue to use H-H equations.

More recently, Erwin Neher and Bert Sakmann (Nobel Prize in Physiology and Medicine, 1991) developed the patch clamp technique, which allows the measurement of pA ion flux through single channels. However, the model presented here and the H-H equations are over-simplifications of the real situation in a neuron. One can think of a neuron as a machine that performs computations at the subcellular level. Thus, the dynamics of spiking activity and the interaction of electric charges provide further levels of information processing. During the generation of an action potential, electric charges that locally interact perform computations at a molecular level within active neurons. A refined model of computation incorporates physical interactions, which could develop during spikes (both axonal and dendritic), internal molecular processes, and the effects of neurotransmitters and generated electric fields, which provide a continuous form of communication. Therefore, some scientists consider the computation developed by electric interactions and that "memories" could be stored within densities of electric charges in molecular structures; see Aur et al. for more discussion [3].

2.4 Two Protypes of Neurons: Excitatory (Glutamatergic) and Inhibitory (GABAergic) Neurons

There are many types of neurons, which can be characterised by their positions, e.g., sensory neurons, which are the first neurons in a neuronal layer. Furthermore, interneurons are neurons that connect to other neurons (the most frequent type). Finally, motoneurons, which activate the muscles, can be considered the last layer in a given network. Neurons can be classified by transmitter types. Initially, it was thought that one transmitter is used for each neuron, but it is now known that some neurons contain more than one transmitter. Embryonic stem cells can differentiate into two major neuronal cell types [80]: activating (excitatory) neurons, which use glutamate as a transmitter, and repressing (inhibitory) neurons, which use gamma amino butyric acid (GABA) as a transmitter. Excitatory neurons can release glutamate, which elicits an *excitatory postsynaptic potential* (EPSP). This is counteracted by the action of the inhibitory neuron, which could induce an inhibitory postsynaptic potential (IPSP) [29]. The postsynaptic potentials are characterised by the potentials of different ions: Na^+ for EPSP and the counteracting Cl^- for IPSP. In a simple view, one part of neural computation occurs at the synapse, where a positively charged EPSP might be summed for the induction of an action potential.

2.5 Long-Term Potentiation

Long-term potentiation describes the process of a long-lasting change in the synaptic plasticity caused by stimulation of an excitatory pathway in, e.g., the hippocampus. The NMDA receptor (see Fig. 2) at the post-synapse seems to play a critical role in this process. In 1973, Bliss, Lømo and Gardner-Medwin reported on this process in anaesthetised [9] as well as unanaesthetised rabbits [8]. Long-term potentiation is believed to be responsible for storing information in the biological neural network, i.e., for (part of) the memory function [7, 28, 63]. Figure 4(b) depicts an example of this process [60]. The stimulus generates a long-lasting change in the efficiency of the synaptic transmission. This change follows three characteristics, identified by Bliss and Collingridge [7]: *cooperativity*, *associativity*, and *input-specificity*:

Input-specificity is the most obvious quality and will also be present in any artificial system. If a stimulus is sent through a fibre that is part of a network, only the active pathway will be potentiated, and the plasticity of the other pathways will remain the same [2, 54]. This effect is reproduced in Table 2. The active pathway is potentiated to approximately 350 %, while the control pathway remains roughly constant. The same behaviour will be observed in an artificial system as dictated

Table 2 Population spike amplitude of an active as well as a control pathway before and after stimulation. The amplitude was probed after 5, 10, and 15 minutes. Up to 17 matched-pair experiments were performed. Adapted by permission from Macmillan Publishers Ltd: Nature [54], copyright 1977

Post tetanus (min)	Pre	5	10	15
Tetanised pathway (%)	100	390	380	332
Control pathway (%)	100	74	67	73

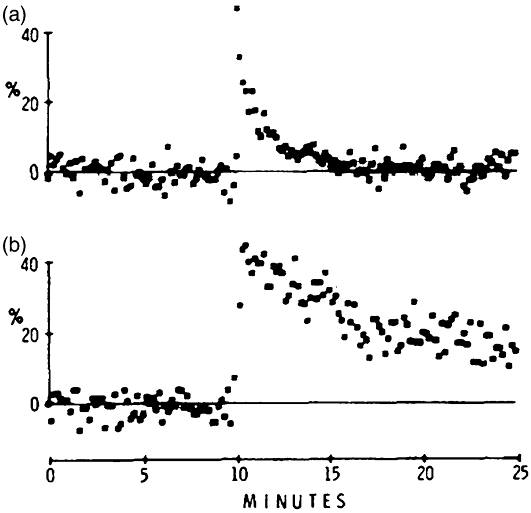


Fig. 4 Response to low- and high-intensity spike trains delivered after 10 min. (**a**) The low-intensity train (250 µV, 30 µs, 128 repetitions at 500 Hz) caused a quickly decaying effect. (**b**) The high-intensity train (250 µV, 255 µs, 16 repetitions at 100 Hz) gives rise to a long-lasting response. Reprinted from Brain research [60], with permission from Elsevier

by general causality. The morphology/topology of the network is determined by the electrical connections, and only the parts in which a current flows can be modified by that current.

Cooperativity designates an intensity threshold to initiate the long-lasting response in the synaptic pathway [60]. This response is depicted in Fig. 4. Two spike trains with a spike amplitude of 250 µV are delivered to a biological system. One train consists of spikes with a width of 30 µs (low intensity, Fig. 4(a)); the other train is made of 255-µs spikes (high intensity, Fig. 4(b)). The response to the high-intensity train is long-lasting, while the response to the low-intensity train decays quickly, although the initial increase is approximately of the same amplitude. The high-intensity train consists of fewer spikes (16) than the low-intensity train (128). In a biological system, the exact response to a given spike train is a complex function of parameters such as spike amplitude and width as well as repetition and repetition rate [56].

The intensity threshold discussed in the preceding paragraph does not consider two sub-threshold signals in the same channel. An example is illustrated in Fig. 5. After four control measurements, a weak tetanus (spike train) arrives but does not lead to LTP. The same is still true for a strong tetanus. We can only observe LTP if

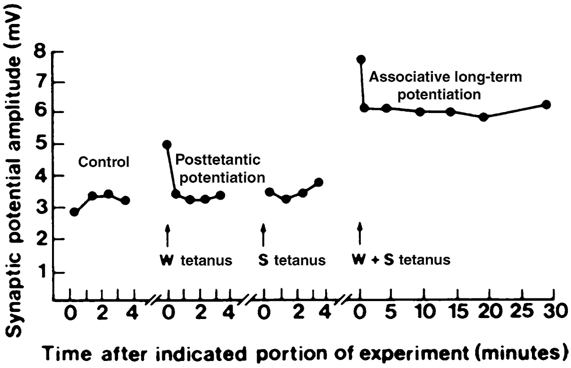


Fig. 5 Associative long-term potentiation: Neither the weak tetanus W nor the strong tetanus S alone lead to LTP. Only W+S combined induce a long-lasting response. Copyright (1983) National Academy of Sciences, USA [5]

both tetani are applied at the same time. This behaviour is called *associative* long-term potentiation [51].

2.6 Long-Term Depression

We have seen in the last paragraph, that we can increase the synaptic plasticity by stimulation of the excitatory pathways; thus, the opposite process should also be possible, i.e., a decrease in the synaptic plasticity. This response is shown in Fig. 6 by Goda and Stevens, which is similar to Fig. 4(b). The connection strength is changed after the arrival of the stimulus. The plasticity increased in the case of LTP (Fig. 4(b)) and decreased in the case of LTD (Fig. 6); thus, we could say that LTD is the antagonist of LTP.

The process of adjusting the synaptic plasticity in a (biological) neural network is generally associated with learning and memory [12, 30, 44]. In this context, Manahan-Vaughan and Braunewell performed experiments in which a punchboard with specific hole patterns was presented to rats, which revealed that exploratory learning is connected to LTD [57]. Experiments by Ge et al. also indicate the importance of LTD in the learning process [32]. Two different LTD-suppressing agents disturbed the spatial memory consolidation, while the LTP-blocking GluN2A antagonist had no effect.

The described responses were observed at the single-neuron level, i.e., one neuron connected to another single neuron. Other processes, such as long-term and short-term memory and data retention in general, might not be explainable by investigating single neurons and their connections. For an overview on data retention in humans, please refer to Rubin and Wenzel [74, 75]. Gene expression is particularly important for memory processes in hour-lasting LTP and is activated by calcium-ion-activated transcription factors (see the description of synapses above).

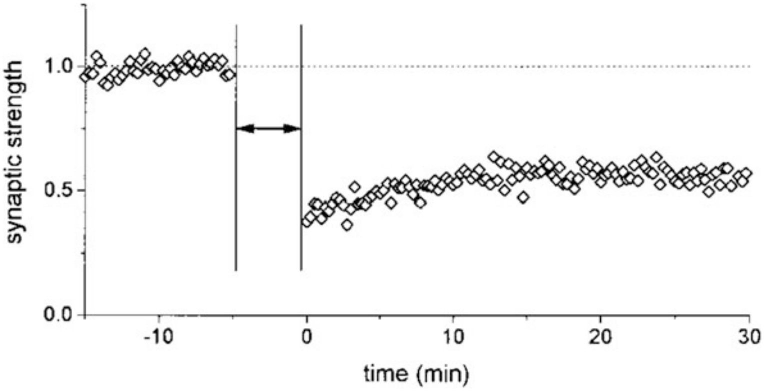
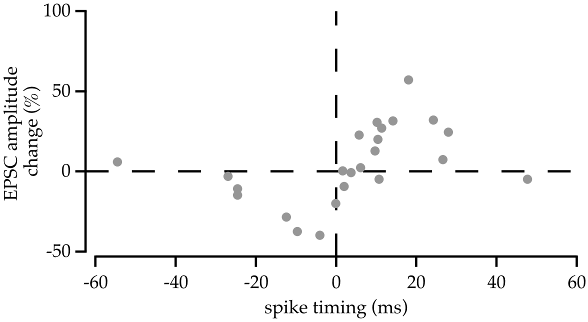


Fig. 6 Long-term depression in hippocampal neurons. Low-frequency stimulation (900 pulses@5 Hz) results in a long-lasting depression. Reprinted from Neuron [33], with permission from Elsevier

Fig. 7 STDP: The EPSC amplitude change is plotted against the time difference of the post- and the presynaptic spike. Positive differences potentiate, while negative differences depress. Adapted by permission from Macmillan Publishers Ltd: Nature [15], copyright 2007



2.7 *Spike-Time-Dependent Plasticity*

To understand the change in synaptic plasticity, we must identify a so-called learning rule. This learning rule determines when the connection strength is increased and in which case the connection strength is decreased. The learning rule is a crucial component of a neural network. In this context, Bi and Poo investigated the plasticity change in interconnected neurons by analysing the timing of spikes [6]. They compared the spike timing of the pre- and postsynaptic neurons and plotted the plasticity change vs. the time difference. The results of a similar experiment by Cassenaer and Laurent are depicted in Fig. 7 [15].

We will examine three distinct examples to understand the meaning of Fig. 7. First, we assume a postsynaptic spike a few moments after the arrival of the presynaptic spike, i.e., a positive spike timing. The postsynaptic spike was likely induced (at least partly) by the presynaptic spike, and the plasticity is potentiated. Now, we presume the opposite case: a negative spike timing, which translates into a postsynaptic spike preceding the presynaptic spike. In this case, the postsynaptic spike cannot be caused by the presynaptic spike, and the plasticity is depressed. Finally, we cannot draw a conclusion if the two spikes occur more than a few milliseconds apart. This overall behaviour is called spike-time-dependent plasticity (STDP) and functions as a *causality detector*.

Spike-time-dependent plasticity is not the only possible learning rule, and STDP alone might not be sufficient to implement a learning algorithm [78]. For more information, please refer to, e.g., Fusi et al. [31] or Carpenter et al. [14]. However, in short, we could describe the requirements for a neural network [81]:

1. Neurons are connected to each other via synapses.
2. The neurons exhibit excitatory and inhibitory inputs and integrate the incoming signals. They generate a spike once a threshold is reached, which also zeroes the integration.
3. The synapses express long-term depression (LTD), long-term potentiation (LTP) and spike-timing-dependent plasticity (STDP).

3 Implementations Using Memristive Systems and Conventional Electronics

If we think about memristors in the context of artificial neural networks, we have to consider the functionality of the memristor. Which part of the network will be emulated by the use of memristors? The basic functionality of a neuron can be easily mimicked by an electronic circuit, i.e., the integration of incoming signals until a certain threshold is exceeded and a pulse is generated. However, the adjustment of the synaptic plasticity according to particular learning rules can only be achieved by employing many (100–400) transistors. Consequently, we present the use of a memristor as an elaborated synapse.

3.1 (Leaky) Integrate and Fire Model

Before we turn to the memristors, we briefly recapitulate the functionality of the (leaky) integrate-and-fire model. Investigations of information processing in a biological neural network date back more than 100 years, to research pioneered by Louis Lapicque (1866–1952) [48, 49]. Thus, his model is known as *Lapicque's model* as well as the *voltage threshold model*: The information is transported via short voltage pulses known as *spikes*. The incoming spikes are integrated in the neuron, and the neuron releases an outgoing spike once a certain threshold is exceeded, which also zeroes the integrator. This model has been improved over the last 100 years and is now known as the *integrate-and-fire model* [39, 42, 79, 82].

This basic behaviour can be reproduced by a simple electronic circuit, which is depicted in Fig. 8 [43]. A capacitor is used to integrate the incoming signals. Then, as an example, a 1-bit analogue-digital converter can be used to fire the spike

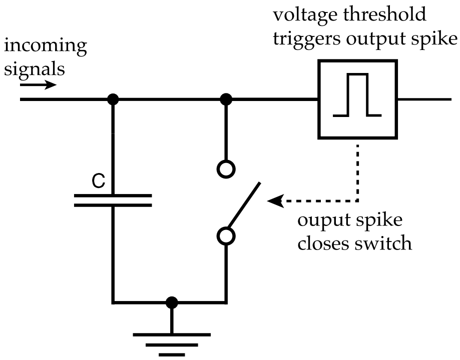


Fig. 8 A basic integrate-and-fire circuit consists of a capacitor that integrates the incoming signals and a voltage threshold detector that produces the output spike and discharges the capacitor

and close the switch. A possible implementation would be an oversampled $\Delta\Sigma$-modulator [4, 86]. Furthermore, the model circuit can be modified to mimic the biological behaviour even more closely. A leakage current through the membrane can be reproduced by a resistor connected in parallel to the capacitor, which is then called a *leaky integrate-and-fire model*.

Because the described behaviour of the neuron can easily be achieved with conventional electronics, we focus on the use of a memristor as a synapse in the following sections of this chapter.

3.2 Long-Term Depression and Potentiation

In this section, we will examine the electronic implementations of long-term potentiation and depression. The memristive system is generally used to provide synaptic functionality. The change in conductivity can naturally be associated with the change in synaptic plasticity in a biological system.

We investigated the resistance change in metal-insulator-metal (MIM) junctions with an insulator thickness of approximately 1 nm [45, 46]. These devices can also be prepared with two ferromagnetic electrodes (FIF) and exhibit tunnel magnetoresistance [62]. This is a mature technology that is used in read heads of hard disk drives [55] or in prototypes of magnetic random access memory [76] and can be prepared on an industrial scale on 300 mm wafers [87]. Two conductance traces of memristive magnetic tunnel junctions are shown in Fig. 9.

We observe long-term depression as well as long-term potentiation in our junctions depending on the polarity of the delivered voltage pulse. Subsequent pulses lead to a subsequent change in the resistance until an upper or lower limit of 6–10 % is reached, while more recent experiments promise larger changes in optimised materials. However, a general limitation of varying magnitude is present in all experimental implementations of memristive systems.

The next example is research by Jo et al. [40], who used CMOS neurons in two tiers: postsynaptic and presynaptic neurons, which also resulted in a global

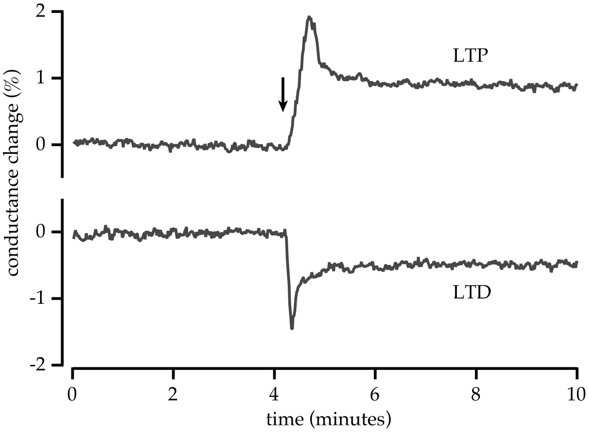


Fig. 9 Long-term potentiation and long-term depression in memristive magnetic tunnel junctions. The positive and negative voltage pulses are delivered after 4 min to induce LTP and LTD, respectively

clock signal. The neurons were connected in a cross-bar geometry, i.e., every neuron in the pre-tier was connected to every neuron in the post-tier through an individual memristor. This arrangement also means that the synaptic plasticity can be adjusted individually for every connection. The adjustment was performed using programming voltage pulses, which were potentiating (3.2 V, 300 µs) or depressing (3.2 V, 300 µs). They observed a continuous increase in conductivity, sending approximately 100 potentiating pulses, followed by a continuous decrease in the conductivity, which was caused by 100 decreasing pulses.

A similar experiment was conducted by Chang et al. [16]. They used tungsten oxide as the functional layer in a device and explained the memristive operation by the migration of oxygen vacancies at the tungsten-oxide/electrode interface. They further presented a SPICE model to examine the results quantitatively. Other material combinations are also possible, e.g., Cu_2O was used by Choi et al. [18] to realise LTP, LTD and STDP, which combines depression and potentiation following a particular learning rule and leads to spike-time-dependent plasticity, which is presented in the next section.

3.3 Spike-Time-Dependent Plasticity

The idea to implement the STDP learning rule by the use of memristors developed immediately after interest in memristors was renewed. In 2008, Greg Snider suggested that STDP could be realised by using memristive nanodevices to emulate synapses [78]. This continued an earlier work by the same author [77]. The synapses were combined with CMOS-based neurons to obtain a neuromorphic circuit, and two electronic circuit symbols were used to indicate the neuron as well as the synapse (Fig. 10). The same symbols are used throughout this chapter.

Snider's approach is based on large-scale adaptive analogue systems and uses a global clock signal, i.e., it is a synchronous implementation. Afifi et al. returned to

Fig. 10 The circuit symbols for a neuron with excitatory and inhibitory inputs and the circuit symbol for a (memristive) synapse

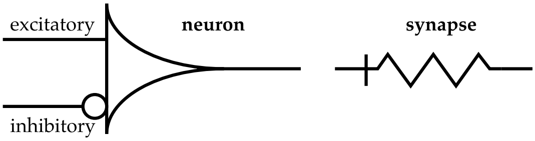
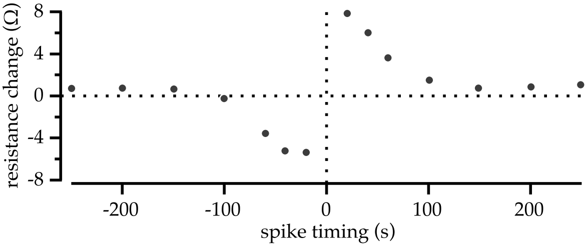


Fig. 11 Spike-time-dependent plasticity in magnetic tunnel junctions



the idea to use STDP in artificial neural networks but suggested a system without a global clock signal, i.e., an asynchronous approach [1], which matches a biological system more closely. In the described case, an artificial neural network was realised using the CMOL platform [83]. CMOL is a crossbar-based CMOS architecture, and established integrate-and-fire neurons were used (see Sect. 3.1 on the integrate-and-fire model) [37, 38, 66]. A special spike form was essential for this particular realisation of the STDP learning rule, although the implementation was robust against variations of the ideal shape of the spikes.

We also observed STDP (Fig. 11) in our magnetic tunnel junctions [47], the potentiation and depression of which are depicted in Fig. 9. The observed behaviour is equivalent to that presented in Fig. 7, although the time scale is different: A simple two terminal device exhibits partial synaptic functionality. Thus, the described experiment and other experiments by Jo et al. [40], Chang et al. [16] and Choi et al. [18] are very promising for the future application of memristive systems in more complex neural networks that are robust against component variability and degradation [61]. Consequently, we will introduce more complex systems in the following paragraphs, starting with Pavlov's dog, which is arguably the smallest system exhibiting *intelligent* behaviour.

3.4 Pavlov's Dog

Before we place everything in the context of neural networks, we will repeat the main features of Pavlov's experiment [68]: when food is presented to a dog, the dog salivates. This is an unconditioned reflex. If a bell sound is played to the same dog, no salivation occurs. Next, the bell is played every time the food is given. With time, the bell becomes a conditioned reflex and leads to salivation, even if presented by itself. The 1904 Nobel Prize in physiology or medicine was awarded to Ivan Petrovich Pavlov

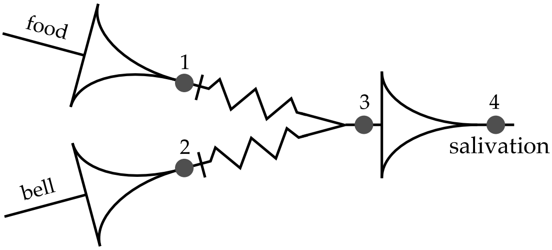


Fig. 12 Basic electronic circuit mimicking the behaviour of Pavlov's dog with only five functional components, as proposed by Pershin and DiVentra. The output at position 4 depends on the input signals and the plasticity of the two synapses

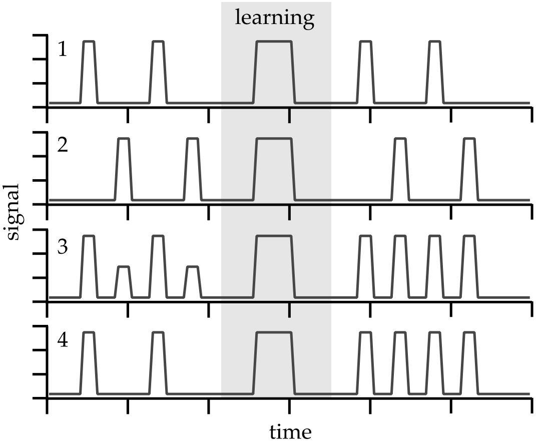


Fig. 13 The signals at the positions 1 to 4 as indicated in Fig. 12. First, the plasticity of the second synapse is low and suppresses the bell signal, which is changed after the learning period in which both inputs are on at the same time

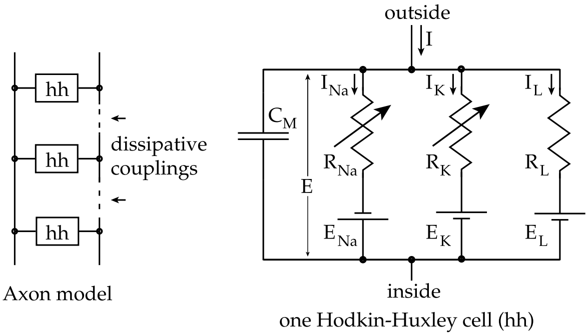
> in recognition of his work on the physiology of digestion, through which knowledge on vital aspects of the subject has been transformed and enlarged [67].

There is a very elegant experiment to demonstrate one important function of the brain—associative memory—with just five functional components in an electrical circuit: two synapses and three neurons. It was described in 2010 by Pershin and DiVentra, who simulated the memristive behaviour using microcontrollers [69, 70]. The principal circuit diagram is depicted in Fig. 12.

Two input neurons are connected via two synapses to an output neuron. The data are taken at four distinct parts in the circuit: (1) after the food neuron, signalling if food is present; (2) at the bell neuron output, signalling if the bell is played; (3) at the input of the salivation neuron but behind the synapses connecting the input neurons; and (4) at the output of the salivation neuron, indicating if salivation occurs. Now, we can observe the previously described behaviour of the actual dog if we examine the signals at the indicated positions in the circuit. This response is illustrated in Fig. 13.

First, we present the food and play the bell alternatively. The food induces a large response signal at position 3, while the bell signal is suppressed. The behaviour changes after a learning phase. In this phase, the food and bell are presented simultaneously. Subsequently, the synaptic plasticity changes according to the STDP

Fig. 14 The axon model of Hodgkin and Huxley consists of several dissipatively coupled hh-cells. Each cell is composed of a capacitor representing the membrane capacitance, the sodium and potassium channels and additional leakage



Axon model

one Hodkin-Huxley cell (hh)

learning rule. Now, even the bell input signal alone is sufficient to trigger a response, i.e., salivation.

The connection between the bell and salivation can also be unlearned if the food is presented without the bell several times. This effect can also be simulated using the presented mechanisms, as demonstrated by Cantley et al. in 2012 using SPICE (UC Berkeley, CA, USA) and MATLAB (Natick, MA, USA) [13]. A similar experiment was performed by Ziegler et al. [91].

In the next section, we would like to introduce another key system in biology, namely the Hodgkin-Huxley model and its relation to memristors as proposed by Chua [21, 23, 24].

3.5 *Hodgkin-Huxley Model*

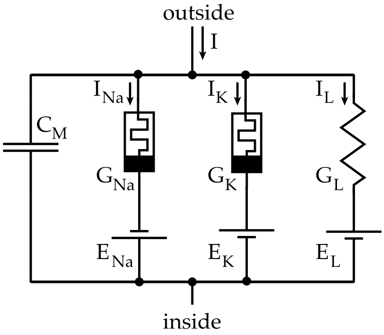
Before we present the connection of memristive systems to the Hodgkin-Huxley model, we will introduce the model itself. The Hodgkin-Huxley model was developed by Alan Lloyd Hodgkin and Andrew Fielding Huxley, who shared the Nobel prize in 1963 with Sir John Carew Eccles

> for their discoveries concerning the ionic mechanisms involved in excitation and inhibition in the peripheral and central portions of the nerve cell membrane [35].

The Hodgkin-Huxley model describes how a spike is initiated and conducted through neural connections. Originally, they chose a giant squid for their studies because of the large size of its axons. The diameter of the axon used to expel water for locomotion is approximately 1 mm [90]. The Hodgkin-Huxley model is an electrical circuit model that explains how the membrane potentials are conducted from one cell to another cell through the axon, whose size facilitates the connection of probes to measure the electrical potentials. Figure 14 depicts the principal electrical circuit of the Hodgkin-Huxley model.

There are a total of four channels in the model, and we will describe Fig. 14 from left to right. The first channel corresponds to the membrane capacity and is represented by a single capacitor. The next two channels represent the sodium and potassium ions and can each be described as an electrical voltage and a variable

Fig. 15 A modified axon model with two memristors replacing the variable resistors of the sodium (2nd order memristor) and potassium (1st order memristor) channels



resistor. The last channel is symbolised by a resistor and a third voltage source, which corresponds to a leakage channel. In total, the external current and the circuit lead to differential equations and, finally, to the Hodgkin-Huxley equations [35].

The axon can now be described as chains of Hodgkin-Huxley cells (hh-cells) in which each cell is dissipatively coupled to the neighbouring cells. The dissipative coupling follows (in an approximation) standard diffusion equations and can be modelled quantitatively [20].

In the Hodgkin-Huxley-model, the values of the circuit elements such as resistances and capacitances are determined experimentally, and these resistances in the potassium and sodium channels are time-varying resistances, i.e., the resistance changes with time but not as a function of time, which is a general assumption in basic circuit theory [19, 22, 23]. Another observation also puzzled researchers: the squid axon exhibits inductive reactance in some experiments [25, 27, 58], which was considered implausible [26].

In 2012, Chua proposed a modified Hodgkin-Huxley model to reconcile the paradox of the positive reactance and the missing magnetic fields due to the inductance. This model replaces the two variable resistors in the potassium and sodium channel with first- and second-order memristors, respectively. The modified circuit is shown in Fig. 15 [23, 24].

3.6 Complex Problem Solving Using Memristors

There are more complex challenges that can be solved with the help of memristive systems. Maze solving is an example that is not based on mimicking synaptic functionality. A maze is a tour puzzle, i.e., a complex arrangement of pathways in which the correct path must be found, e.g., the shortest path to enter and exit the maze. Mazes have fascinated people since ancient times; the famous Cretan labyrinth is described by Homer in the Iliad (Please note that we do not distinguish between mazes and labyrinths in this chapter).

The primitive biological organism Physarum polycephalum can solve a maze in which food has been placed at the entrance and exit. The maze solving was presented by Nakagaki et al. in 2000 [65] and is illustrated in Fig. 16.

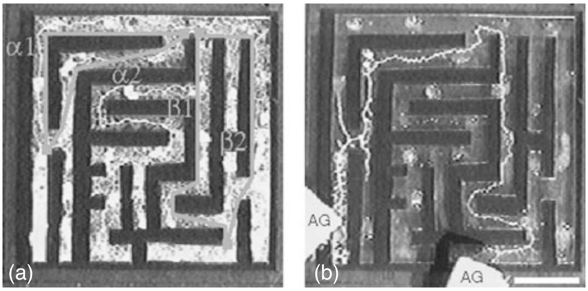


Fig. 16 (**a**) The P. polycephalum was shaped into the form of a maze. There are 2 × 2 possible solutions: $\alpha 1$ (41 mm); $\alpha 2$ (33 mm); $\beta 1$ (44 mm); and $\beta 2$ (45 mm); all ±1 mm. (**b**) After 8 hours, the organism selected the shortest path between the exit and entry points marked with food. Adapted by permission from Macmillan Publishers Ltd: Nature [65], copyright 2000

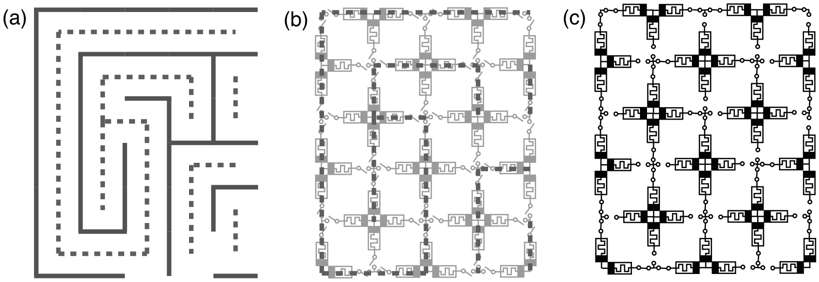


Fig. 17 (**a**) A part of a simple maze in 2 dimensions with *grey*, *solid walls* and *red*, broken, possible pathways. (**b**) An overlay of the memristor/switch checkerboard pattern and the pathways. (**c**) The switches are on for possible pathways and off otherwise. Image redrawn from Pershin et al. [71]

First, the slime mould was cut into pieces and assembled in the shape of the desired maze. The mould merged into a single piece while maintaining its shape because plastic pieces of the desired shape were put on a food source (agar) and the mould avoided the plastic. Next, the start and end points were marked with agar blocks. The organism could then choose between 4 routes to reach both food sources: ($\alpha 1$, $\beta 1$),($\alpha 2$, $\beta 1$),($\alpha 2$, $\beta 1$), and ($\alpha 2$, $\beta 2$). After 8 hours, the mould changed its shape to that of a single tube connecting the two agar blocks via the shortest possible path ($\alpha 2$, $\beta 1$).

Pershin et al. used Physarum polycephalum as inspiration to solve problems with the help of memristors [71, 72]. Here, we will examine their interesting maze-solving scheme [71], as depicted in Fig. 17.

First, we examine the delineation of a given maze with memristors. The maze and the possible pathways are shown in Fig. 17(a). The maze is superimposed on a head-to-head and tail-to-tail checkerboard pattern of memristors in Fig. 17(b). Each memristor is connected to the others via a switch at the 'backside'. To encode the pattern, the switch is closed for possible pathways and closed for blocked passages. This leads to a configuration of memristors as illustrated in Fig. 17(c). A complete maze is prepared in the same way, and a voltage is applied across the entrance and exit points of the maze to solve it. The solution is determined by a read-out of each of the memristor resistances (all switches are put in the off state). The network finds the shortest path, which is given by the lowest resistances, as well as all other solutions. The resistance for each path is proportional to its length. The network has no external clock and finds all of the solutions in parallel in one 'step', although the additional read-out requires subsequent resistance measurements. It is a nice example of a large-scale adaptive analogue system.

4 Conclusion and Outlook

In this chapter, we presented some bio-inspired neural networks that directly mimic mechanisms such as long-term potentiation and spike-time-dependent plasticity and more abstract processes such as associative memory. All of the presented work can be implemented on a hardware level, which is a complementary approach to simulations of the same mechanisms. In the future, more complex hardware implementations might be advantageous to a software approach because of the low energy consumption, robustness and parallel processing capability. In all presented systems, memristors were considered as the main functional element. Memristors have been the focus of attention because a simple two-terminal device already exhibits partial synaptic functionality and consequently promises good scalability. Other future applications might include assistance systems or the tracking of robotic movement with imperfect 'muscles' (motors).

Acknowledgements The authors would like to thank Barbara Kaltschmidt and Elisabetta Chicca for many helpful discussions. Furthermore, we are grateful to Patryk Krzysteczko, Jana Münchenberger, Stefan Niehörster, Matthias Schürmann, Marius Schirmer, Olga Simon, Savio Fabretti, and Joachim Sterz. Additionally, we would like to thank Günter Reiss and Andreas Hütten for the continuous support of our work. The Ministry of Innovation, Science and Research (MIWF) of North Rhine-Westfalia funded this research.

References

1. Afifi, A., Ayatollahi, A., Raissi, F.: STDP implementation using memristive nanodevice in CMOS-nano neuromorphic networks. IEICE Electron. Express **6**(3), 148–153 (2009)
2. Andersen, P., Sundberg, S., Sveen, O., Wigström, H.: Specific long-lasting potentiation of synaptic transmission in hippocampal slices. Nature **266**(5604), 736–737 (1977)
3. Aur, D., Jog, M., Poznanski, R.R.: Computing by physical interaction in neurons. J. Integr. Neurosci. **10**(04), 413–422 (2011)
4. Aziz, P., Sorensen, H., van der Spiegel, J.: An overview of sigma-delta converters. IEEE Signal Process. Mag. **13**(1), 61–84 (1996)
5. Barrionuevo, G., Brown, T.: Associative long-term potentiation in hippocampal slices. Proc. Natl. Acad. Sci. USA **80**(23), 7347–7351 (1983)
6. Bi, G.q., Poo, M.m.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. J. Neurosci. **18**(24), 10464–10472 (1998)

7. Bliss, T., Collingridge, G.: A synaptic model of memory—long-term potentiation in the hippocampus. Nature **361**(6407), 31–39 (1993)
8. Bliss, T., Gardner-Medwin, A.R.: Long-lasting potentiation of synaptic transmission in dentate area of unanesthetized rabbit following stimulation of perforant path. J. Physiol., Lond. **232**(2), 357–374 (1973)
9. Bliss, T., Lømo, T.: Long-lasting potentiation of synaptic transmission in dentate area of anesthetized rabbit following stimulation of perforant path. J. Physiol., Lond. **232**(2), 331–356 (1973)
10. Borghetti, J., Li, Z., Straznicky, J., Li, X., Ohlberg, D.A.A., Wu, W., Stewart, D.R., Williams, R.S.: A hybrid nanomemristor/transistor logic circuit capable of self-programming. Proc. Natl. Acad. Sci. USA **106**(6), 1699–1703 (2009)
11. Borghetti, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R., Williams, R.S.: 'Memristive' switches enable 'stateful' logic operations via material implication. Nature **464**(7290), 873–876 (2010)
12. Cajal, S.R.: Histology of the Nervous System of Man and Vertebrates. Oxford University Press, London (1995)
13. Cantley, K.D., Subramaniam, A., Stiegler, H.J., Chapman, R.A., Vogel, E.M.: Neural learning circuits utilizing nano-crystalline silicon transistors and memristors. IEEE Trans. Neural Netw. Learn. Syst. **23**(4), 565–573 (2012)
14. Carpenter, G., Milenova, B., Noeske, B.: Distributed ARTMAP: a neural network for fast distributed supervised learning. Neural Netw. **11**(5), 793–813 (1998)
15. Cassenaer, S., Laurent, G.: Hebbian STDP in mushroom bodies facilitates the synchronous flow of olfactory information in locusts. Nature **448**(7154), 709–712 (2007)
16. Chang, T., Jo, S.H., Kim, K.H., Sheridan, P., Gaba, S., Lu, W.: Synaptic behaviors and modeling of a metal oxide memristive device. Appl. Phys. A **102**(4), 857–863 (2011)
17. Chen, X., Wu, G., Dinghua, B.: Resistive switching behavior of Pt/Mg0.2Zn0.8O/Pt devices for nonvolatile memory applications. Appl. Phys. Lett. **93**, 093501 (2008)
18. Choi, S.J., Kim, G.B., Lee, K., Kim, K.H., Yang, W.Y., Cho, S., Bae, H.J., Seo, D.S., Kim, S.I., Lee, K.J.: Synaptic behaviors of a single metal–oxide–metal resistive device. Appl. Phys. A **102**(4), 1019–1025 (2011)
19. Chua, L.O.: Introduction to Nonlinear Network Theory. McGraw-Hill, New York (1969)
20. Chua, L.O.: CNN: A Paradigm for Complexity. World Scientific, Singapore (1998)
21. Chua, L., Kang, S.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
22. Chua, L.O., Desoer, C.A., Kuh, E.S.: Linear and Nonlinear Circuits. McGraw-Hill, New York (1987)
23. Chua, L., Sbitnev, V., Kim, H.: Hodgkin-Huxley axon is made of memristors. Int. J. Bifurc. Chaos **22**(3), 1230011 (2012)
24. Chua, L., Sbitnev, V., Kim, H.: Neurons are poised near the edge of chaos. Int. J. Bifurc. Chaos **22**(4), 1250098 (2012)
25. Cole, K.: Rectification and inductance in the squid giant axon. J. Gen. Physiol. **25**(1), 29–51 (1941)
26. Cole, K.: Membranes, Ions and Impulses. University of California Press, Berkeley (1972)
27. Cole, K., Baker, R.: Longitudinal impedance of the squid giant axon. J. Gen. Physiol. **24**(6), 771–788 (1941)
28. Doyere, V., Laroche, S.: Linear relationship between the maintenance of hippocampal long-term potentiation and retention of an associative memory. Hippocampus **2**(1), 39–48 (1992)
29. Eccles, J.: The Ferrier lecture: the nature of central inhibition. Proc. R. Soc. Lond. B, Biol. Sci. **153**, 445–476 (1961)
30. Eccles, J., McIntyre, A.: Plasticity of mammalian monosynaptic reflexes. Nature **167**(4247), 466–468 (1951)
31. Fusi, S., Abbott, L.F.: Limits on the memory storage capacity of bounded synapses. Nat. Neurosci. **10**(4), 485–493 (2007)
32. Ge, Y., Dong, Z., Bagot, R.C., Howland, J.G., Phillips, A.G., Wong, T.P., Wang, Y.T.: Hippocampal long-term depression is required for the consolidation of spatial memory. Proc.

Natl. Acad. Sci. USA **107**(38), 16697–16702 (2010)

33. Goda, Y., Stevens, C.: Long-term depression properties in a simple system. Neuron **16**, 103–111 (1996)
34. Hodgkin, A., Huxley, A.: Action potentials recorded from inside a nerve fibre. Nature **144**, 710–711 (1939)
35. Hodgkin, A., Huxley, A.: A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol., Lond. **117**(4), 500–544 (1952)
36. Imielski, Y., Schwamborn, J.C., Lüningschrör, P., Heimann, P., Holzberg, M., Werner, H., Leske, O., Püschel, A.W., Memet, S., Heumann, R., Israel, A., Kaltschmidt, C., Kaltschmidt, B.: Regrowing the adult brain: NF-κB controls functional circuit formation and tissue homeostasis in the dentate gyrus. PLoS ONE **7**(2), e30838 (2012)
37. Indiveri, G., Chicca, E., Douglas, R.: A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. IEEE Trans. Neural Netw. **17**(1), 211–221 (2006)
38. Indiveri, G., Stefanini, F., Chicca, E.: Spike-based learning with a generalized integrate and fire silicon neuron. In: 2010 IEEE International Symposium on Circuits and Systems, pp. 1951–1954 (2010)
39. Jack, J.J.B., Noble, D., Tsien, R.W.: Electric current flow in excitable cells. OUP Australia and New Zealand (1975)
40. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. **10**(4), 1297–1301 (2010)
41. Kaltschmidt, B., Kaltschmidt, C.: NF-κB in the nervous system. Cold Spring Harb. Perspect. Biol. **1**(3), a001271 (2009)
42. Knight, B.: Dynamics of encoding in a population of neurons. J. Gen. Physiol. **59**(6), 734 (1972)
43. Koch, C.: Biophysics of Computation. Oxford University Press, London (1999)
44. Konorski, J.: Conditioned Reflexes and Neuron Organization. Cambridge University Press, Cambridge (1948)
45. Krzysteczko, P., Kou, X., Rott, K., Thomas, A.: Current induced resistance change of magnetic tunnel junctions with ultra-thin MgO tunnel barriers. J. Magn. Magn. Mater. **321**, 144 (2008)
46. Krzysteczko, P., Reiss, G., Thomas, A.: Memristive switching of MgO based magnetic tunnel junctions. Appl. Phys. Lett. **95**(11), 112508 (2009)
47. Krzysteczko, P., Münchenberger, J., Schäfers, M., Reiss, G., Thomas, A.: The memristive magnetic tunnel junction as a nanoscopic synapse-neuron system. Adv. Mater. **24**, 762–766 (2012)
48. Lapicque, L.: Lapicque: recherches quantitatives sur l'excitation des nerfs traitée comme une polarisation. J. Physiol. Pathol. Gén. **9** (1907)
49. Lapicque, L.: L'excitabilité en fonction du temps. Presses Universitaires de France, Paris (1926)
50. Lee, M.J., Lee, C.B., Lee, D., Lee, S.R., Chang, M., Hur, J.H., Kim, Y.B., Kim, C.J., Seo, D.H., Seo, S., Chung, U.I., Yoo, I.K., Kim, K.: A fast, high-endurance and scalable non-volatile memory device made from asymmetric Ta_2O_{5-x}/TaO_{2-x} bilayer structures. Nat. Mater. **10**(8), 625–630 (2011)
51. Levy, W., Steward, O.: Synapses as associative memory elements in the hippocampal-formation. Brain Res. **175**(2), 233–245 (1979)
52. Linn, E., Rosezin, R., Kuegeler, C., Waser, R.: Complementary resistive switches for passive nanocrossbar memories. Nat. Mater. **9**(5), 403–406 (2010)
53. Luscher, C., Malenka, R.C.: NMDA receptor-dependent long-term potentiation and long-term depression (LTP/LTD). Cold Spring Harbor Perspectives in Biology **4**(6), a005710 (2012)
54. Lynch, G., Dunwiddie, T., Gribkoff, V.: Heterosynaptic depression—postsynaptic correlate of long-term potentiation. Nature **266**(5604), 737–739 (1977)
55. Maekawa, S., Shinjo, T. (eds.): Spin Dependent Transport in Magnetic Nanostructures. Advances in Condensed Matter Science. CRC Press, Boca Raton (2002)

56. Malenka, R.: Postsynaptic factors control the duration of synaptic enhancement in area Ca1 of the hippocampus. Neuron **6**(1), 53–60 (1991)
57. Manahan-Vaughan, D., Braunewell, K.: Novelty acquisition is associated with induction of hippocampal long-term depression. Proc. Natl. Acad. Sci. USA **96**(15), 8739–8744 (1999)
58. Mauro, A.: Anomalous impedance, a phenomenological property of time-variant resistance—an analytic review. Biophys. J. **1**(4), 353–372 (1961)
59. Mayford, M., Siegelbaum, S.A., Kandel, E.R.: Synapses and memory storage. Cold Spring Harbor Perspectives in Biology **4**(6), a005751 (2012)
60. McNaughton, B., Douglas, R., Goddardd, G.: Synaptic enhancement in fascia dentata—cooperativity among coactive afferents. Brain Res. **157**(2), 277–293 (1978)
61. Mead, C.: Neuromorphic electronic systems. Proc. IEEE **78**(10), 1629–1636 (1990)
62. Moodera, J., Mathon, G.: Spin polarized tunneling in ferromagnetic junctions. J. Magn. Magn. Mater. **200**(1–3), 248–273 (1999)
63. Morris, R., Davis, S., Butcher, S.: Hippocampal synaptic plasticity and NMDA receptors—a role in information-storage. Philos. Trans. R. Soc. B **329**(1253), 187–204 (1990)
64. Muenchenberger, J., Krzysteczko, P., Reiss, G., Thomas, A.: Improved reliability of magnetic field programmable gate arrays through the use of memristive tunnel junctions. J. Appl. Phys. **110**(9), 096105 (2011)
65. Nakagaki, T., Yamada, H., Toth, A.: Maze-solving by an amoeboid organism. Nature **407**(6803), 470 (2000)
66. Neftci, E., Chicca, E., Indiveri, G., Douglas, R.: A systematic method for configuring VLSI networks of spiking neurons. Neural Comput. **23**, 2457–2497 (2011)
67. Norrby, E.: Nobel Prizes. World Scientific, Singapore (2010)
68. Pavlov, I.: Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex. Oxford University Press, London (1927)
69. Pershin, Y.V., Di Ventra, M.: Experimental demonstration of associative memory with memristive neural networks. Neural Netw. **23**(7), 881–886 (2010)
70. Pershin, Y.V., Di Ventra, M.: Neuromorphic, digital, and quantum computation with memory circuit elements. Proc. IEEE **100**(6), 2071–2080 (2010)
71. Pershin, Y.V., Di Ventra, M.: Solving mazes with memristors: a massively parallel approach. Phys. Rev. E **84**(4), 046703 (2011)
72. Pershin, Y., La Fontaine, S., Di Ventra, M.: Memristive model of amoeba learning. Phys. Rev. E **80**(2), 021926 (2009)
73. Poon, C.S.: Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities. Front. Neurosci. 1–3 (2011)
74. Rubin, D., Wenzel, A.: One hundred years of forgetting: a quantitative description of retention. Psychol. Rev. **103**(4), 734–760 (1996)
75. Rubin, D., Hinton, S., Wenzel, A.: The precise time course of retention. J. Exp. Psychol. Learn. **25**(5), 1161–1176 (1999)
76. Sbiaa, R., Meng, H., Piramanayagam, S.N.: Materials with perpendicular magnetic anisotropy for magnetic random access memory. Phys. Status Solidi RRL **5**(12), 413–419 (2011)
77. Snider, G.S.: Self-organized computation with unreliable, memristive nanodevices. Nanotechnology **18**(36), 365202 (2007)
78. Snider, G.: Spike-timing-dependent learning in memristive nanodevices. Nanoscale Archit. 85–92 (2008)
79. Stein, R.: Frequency of nerve action potentials generated by applied currents. Proc. R. Soc. Lond. B, Biol. Sci. **167**(1006), 64 (1967)
80. Strübing, C., Ahnert-Hilger, G., Shan, J., Wiedenmann, B., Hescheler, J., Wobus, A.M.: Differentiation of pluripotent embryonic stem cells into the neuronal lineage in vitro gives rise to mature inhibitory and excitatory neurons. Mech. Dev. **53**, 275–287 (1995)
81. Thomas, A.: Memristor-based neural networks. J. Phys. D, Appl. Phys. **46**(9), 093001 (2013)
82. Tuckwell, H.C.: Introduction to Theoretical Neurobiology. Cambridge University Press, Cambridge (2008)

83. Turel, O., Lee, J., Ma, X., Likharev, K.: Neuromorphic architectures for nanoelectronic circuits. Int. J. Circuit Theory Appl. **32**(5), 277–302 (2004)
84. Van Essen, D.C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T.E.J., Bucholz, R., Chang, A., Chen, L., Corbetta, M., Curtiss, S.W., Della Penna, S., Feinberg, D., Glasser, M.F., Harel, N., Heath, A.C., Larson-Prior, L., Marcus, D., Michalareas, G., Moeller, S., Oostenveld, R., Petersen, S.E., Prior, F., Schlaggar, B.L., Smith, S.M., Snyder, A.Z., Xu, J., Yacoub, E., Consortium, W.M.H.: The human connectome project: a data acquisition perspective. NeuroImage **62**(4), 2222–2231 (2012)
85. von Neumann, J.: First draft of a report on the EDVAC. Tech. rep., University of Pennsylvania (1945)
86. Wong, P., Gray, R.: Sigma-delta modulation with I.I.D. Gaussian inputs. IEEE Trans. Inf. Theory **36**(4), 784–798 (1990)
87. Wrona, J., Langer, J., Ocker, B., Maass, W., Kanak, J., Stobiecki, T., Powroźnik, W.: Low resistance magnetic tunnel junctions with MgO wedge barrier. J. Phys. Conf. Ser. **200**(5), 052032 (2010)
88. Xia, Q., Robinett, W., Cumbie, M.W., Banerjee, N., Cardinali, T.J., Yang, J.J., Wu, W., Li, X., Tong, W.M., Strukov, D.B., Snider, G.S., Medeiros-Ribeiro, G., Williams, R.S.: Memristor-CMOS hybrid integrated circuits for reconfigurable logic. Nano Lett. **9**(10), 3640–3645 (2009)
89. Yan, H., Choe, H., Nam, S., Hu, Y., Das, S.: Programmable nanowire circuits for nanoprocessors. Nature **470**, 240–244 (2011)
90. Young, J.: The structure of nerve fibres in cephalopods and crustacea. Proc. R. Soc. Lond. B, Biol. Sci. **121**(822), 319–337 (1936)
91. Ziegler, M., Soni, R., Patelczyk, T., Ignatov, M., Bartsch, T., Meuffels, P., Kohlstedt, H.: An electronic version of Pavlov's dog. Adv. Funct. Mater. **22**(13), 2744–2749 (2012)
92. Zuse, K.: Der Computer—Mein Lebenswerk: 100 Jahre Zuse. Springer, Berlin (2010)

Self-organization and Emergence of Dynamical Structures in Neuromorphic Atomic Switch Networks

Adam Z. Stieg, Audrius V. Avizienis, Henry O. Sillin, Renato Aguilera, Hsien-Hang Shieh, Cristina Martin-Olmos, Eric J. Sandouk, Masakazu Aono, and James K. Gimzewski

Abstract The self-organization of dynamical structures in complex natural systems is associated with an intrinsic capacity for computation. Beginning from the context of modern trends in neuromorphic engineering, this work introduces an effort toward the construction of purpose-built dynamical systems. Known as atomic switch networks (ASN), these systems consist of highly interconnected, physically recurrent networks of inorganic synapses (atomic switches). By combining the advantages of controlled design with those of self-organization, the functional topology of ASNs has been shown to produce emergent system-wide dynamics and a diverse set of complex behaviors with striking similarity to those observed in many natural systems including biological neural networks and assemblies. Numerical modeling and experimental investigations of their operational characteristics and intrinsic dynamical properties have facilitated progress toward implementation in neuromorphic reservoir computing. These achievements demonstrate the utility of ASNs as a uniquely scalable physical platform capable of exploring the dynamical interface of complexity, neuroscience, and engineering.

A.Z. Stieg (✉) · H.-H. Shieh · C. Martin-Olmos · J.K. Gimzewski
California NanoSystems Institute (CNSI), University of California, Los Angeles, USA
e-mail: stieg@cnsi.ucla.edu

J.K. Gimzewski
e-mail: gimzewski@cnsi.ucla.edu

A.Z. Stieg · M. Aono · J.K. Gimzewski
WPI Center for Materials Nanoarchitectonics (MANA), National Institute for Materials Science (NIMS), Tsukuba, Japan

A.V. Avizienis · H.O. Sillin · R. Aguilera · E.J. Sandouk · J.K. Gimzewski
Department of Chemistry and Biochemistry, University of California, Los Angeles, USA

C. Martin-Olmos
Centre Suisse de Electronique et de Microtechnique SA (CSEM), Neuchatel, Switzerland

J.K. Gimzewski
School of Physics, Centre for Nanoscience and Quantum Information, University of Bristol, Bristol, UK

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_10,

1 Introduction

Networks are universal [15, 40]. At all known scales of space and time, complex patterns emerge from the dynamic relationships between basic building blocks [35, 123]. Whether random or ordered, static or dynamic, or somewhere in between, natural systems can be described as networks. These networks, which by definition are open and correlated, have garnered substantial inquiry into their structure, function, and evolution. When considering networks that describe the behavior of the natural world, particular classes of complex topologies tend to predominate, such as those with non-trivial connectivity termed either 'small world' or 'scale free' [14, 142, 144]. Complex topology alone however does not beget complex behavior. To date, research in the field of complex networks has largely focused on either the dynamical aspects of network topology or the dynamical processes generated from fixed, non-adaptive networks [22, 49, 52]. Concomitantly, the perpetual interplay between structure and function is known throughout the natural world [113, 123]. While it is common for scientific inquiry to begin either from a phenomenological or reductionist approach [21, 24], a restricted view of any complex system by definition fails to capture essential aspects of the structure. This is especially true for nonlinear dynamical systems, where feedback and memory effects can amplify seemingly minor perturbations into large-scale changes of behavior.

Complex networks are inextricably linked to self-organization as a consequence of interactions amongst their constituent parts [40, 96, 114], which not only produce emergent correlations, patterns, and dynamical properties [7, 34, 35, 135] but also render these systems adaptive and evolutionary [49, 52]. One great mystery of humankind is how a self-organized collection of relatively simple, nonlinear dynamical elements known as the brain demonstrates such extraordinary efficiency in topology [25], energy consumption [137] and capacity to integrate, segregate, and process information [2, 130, 136]. Extensive efforts to describe the structural [68, 118] and functional [18, 38, 43, 69, 92, 129] properties of the brain have exposed its complex, dynamical nature [29, 44, 116, 117]. Of particular interest are those properties considered emergent in that they belong to the system collectively rather than to any isolated element of it [51]. For example, consciousness has been posited as an emergent property of the brain, arising from nonlinear interactions between microscopic components (neurons, synapses) throughout a hierarchy of brain structures which produce macroscopic patterns of activity associated with conscious perception [29, 107, 128]. Accordingly, these patterns of activity have been analyzed in great detail to understand their causal relationship with emergent, system-level functions. A recurring theme in these patterns is the presence of scale-free, power-law distributions such as the distribution of spatially and temporally correlated electric spiking activity known as neuronal 'avalanches' [19].

For historical reasons, these observations of scale-free behavior are associated with the concept of criticality. Referring generally to states where the correlation length diverges, as during continuous phase transitions, critical systems represent an interdisciplinary nexus combining research ranging from superconductivity to cosmology [12, 66, 132]. Systems exhibiting power-law behaviors have characteristic critical exponents associated with general classes of interactions [120]. Experimental determination of critical exponents provides an important link between theory and observation, which is especially crucial when investigating dynamic networks with complex topologies [41]. The behavior of such complex systems are difficult to quantify, leaving a great deal to be inferred from simulations and models, including basic questions as to which measureable physical quantities contain useful information about the system [33, 126]. Despite the inherent challenges of characterizing critical systems, extraordinary interest in the spatiotemporal properties of brain activity has promulgated entirely new areas of research into the scaling laws of neuronal dynamics [28, 29, 37, 53, 72, 81, 103], where the patterns of critical activity known as neuronal avalanches can be viewed as representing maximal efficiency for information transmission and computation [19].

The concept of the brain as a computer has been part of the modern scientific landscape for over fifty years following the path laid out by early researchers in cybernetics [147] and Turing's pioneering work [134]. This path has led to new fields of research including artificial intelligence, neural networks, and neuromorphic engineering [58, 90, 91, 106]. Connections between computation and the brain have been studied extensively using artificial neural networks (ANN) [59]. Inspired by biological neural networks, ANNs consist of 'neuron' nodes connected by 'synapses' of variable strength and can be trained to perform a given task through algorithmic modification of the synaptic weights. A desired input-output relationship can be generated for a known set of examples, after which the ANN can be used to process unknown inputs. This general framework spans a vast parameter space: network size, network topology, neuron activity thresholds, update rules, training algorithms, etc., in which each independent task potentially has a unique optimal configuration [57]. ANN research is consequently diverse, with some efforts focusing on designing biologically plausible networks and determining their general properties (e.g. critical exponents), while others create and optimize ANN models to efficiently perform specific, high-level tasks such as pattern classification, time series prediction or nonlinear control [16, 152].

Some features of biological neural networks, such as their recurrent connectivity, are not readily implemented in ANNs. Recurrent neural network models are problematic due to the presence of coupled feedback, which complicates the update rules and destabilizes standard training algorithms [9, 77, 148]. These technical difficulties have motivated the development of a new approach to recurrent neural networks [83], termed reservoir computing (RC) [64, 86, 121]. The RC framework starts with a complex recurrent network of nonlinear elements and that serves a kind of filter that dynamically maps low-dimensional input signals to a high-dimensional representation: the state of the reservoir. The recurrent reservoir does not need to be trained in order to perform a useful task. Instead, linear classifiers external to the reservoir can be trained to correlate the observed reservoir response to task inputs with the desired task output. A sufficiently complex reservoir should be able to perform any computation, significantly outperforming conventional algorithms for certain classes of problems [84, 109, 139]. Though simulation of recurrent reservoirs remains the norm, it is also possible to build one. Any device built to emulate the brain, with nonlinear functional elements possessing memory-dependent activation thresholds connected recurrently in a complex network, should be usable as a hardware reservoir. The potential advantages of a hardware approach are quite clear since device physics will instantly produce the entire reservoir response, which a simulation will always need time to calculate.

Complex systems continue to inspire intense scientific inquiry, largely due to their capacity to confound rigorous examination. To comprehend and harness the dynamical properties of complex networks necessitates a new frame of reference: one that explores the interface of mechanistic insight and observed behavior [70]. Exponential trends in the accessibility of quantitative data, availability of statistical methods, and capacity for computational analysis make the present a time like no other to embrace such an ambitious endeavor [48, 88]. In this chapter, we have set out to describe the context, motivation, approach, and progress in efforts to design and implement a physical system capable of generating the class of dynamical properties found in complex networks, specifically the mammalian brain. By providing a brief background on the history, recent developments, and new directions in neuromorphic engineering in Sect. 2, we lay a foundation for the concept of generating a dynamical neuromorphic system known as ASNs using highly interconnected networks of inorganic synapses (atomic switches). Fabricated through a nanoarchitectonic approach blending traditional top-down microfabrication with bottom-up self-organization, these physical devices have been specifically designed to capture emergent, network-level functionality. ASNs enable fundamental examinations of complex systems as well as applications in neuromorphic computing. Section 3 describes recent progress in extending basic numerical modeling of individual atomic switches to examine their emergent properties when embedded in a complex network through simulation. The results of this study are compared to and validated by our experimental results throughout the subsequent sections. Discussion of the operational device characteristics (activation, memristance, and plasticity) and emergent properties (recurrent dynamics, harmonic generation, and criticality) provided in Sect. 4 generates the basis for efforts to harness system dynamics toward potential applications in memory, learning, and neuromorphic computation described in Sect. 5. Finally, we provide our perspectives and outlook for future developments.

2 Emergence of a Complex Neuromorphic Architecture

Modern computers can be functionally described as highly optimized Turing machines [134] operating in the Von Neumann paradigm [140]. Although impressive engineering achievements in complementary metal-oxide semiconductor (CMOS) technology have enabled nearly exponential increases in processing power and data storage, this approach remains subject to well-documented fundamental limits [45, 85]. Computational architectures produced in CMOS consume orders of magnitude more space and energy than intelligent biological systems such as the human brain [141]. In addition, their capacity for processing of multiple/distributed data sets in real-time is limited at best. Originally proposed by Carver Mead over twenty years ago [91], the concept of designing biologically inspired systems for information processing that could be realized in silicon-based CMOS technology launched an entirely new field of research now known as neuromorphic engineering.

Since the early days of neuromorphic engineering, and in conjunction with advances in high-performance supercomputing, a myriad of implementations have built upon very-large scale integration (VLSI) and/or parallel integration of neuromorphic components, circuits, and devices. The goal of such efforts revolved around the design and construction of purpose-built systems largely optimized for the execution of particular algorithmic operations or pre-defined symbol processing tasks [39, 60, 62, 80, 119, 124, 131, 133]. Innovative hardware designs have shown marked improvements in the density, flexibility, and performance of neuromorphic circuits. This has enabled the modeling of biologically relevant spiking neural networks at a fraction of the expense (time and energy) as compared to their simulation-based counterparts carried out on traditional hardware [3, 61, 104, 108, 119]. Extensions of these efforts to generate and simulate multi-scale systematic maps of functional connectivity in the mammalian brain, known as the connectome, have achieved a reasonable level of success considering the task complexity [6, 17, 38, 69, 88, 93, 115].

Despite substantial progress toward wafer-scale integration of neuromorphic hardware, divergent views on the significance of connectionist approaches to neuroscience persist [94, 138]. There remains a basic gap between using neuromorphic computing systems to simulate neurobiological function and the ability to harness the essential nonlinear dynamics that enable processing, computation, and storage of information in the natural cognitive systems. Within the past year, two highly ambitious large-scale, international projects have been announced that represent large-scale international collaborations which attempt to address these aspects of next-generation neuromorphic computing systems: the Human Brain Project [89] and BRAIN Initiative [4, 5]. These endeavors seek new fundamental insights into how the brain 'computes' by considering structural and functional characteristics that may represent the architectural basis of a new computational paradigm.

2.1 Inorganic Synapses

Recent focus in next-generation neuromorphic hardware [62, 67] has sought to diversify the efforts of beyond-CMOS technology [63] or the development of neuronal circuitry [87, 91] toward emulating the operational behaviors of biological synapses, which are the key players in mediating signaling within the vast network of neurons that underlies cognition. Various concepts of 'synthetic synapses' have been demonstrated or modeled which display long and short-term potentiation (LTP and STP respectively), long-term depression (LTD), spike timing dependent plasticity (STDP), and other neuroscientific phenomena [23, 26, 27, 31, 55, 56, 71, 74,

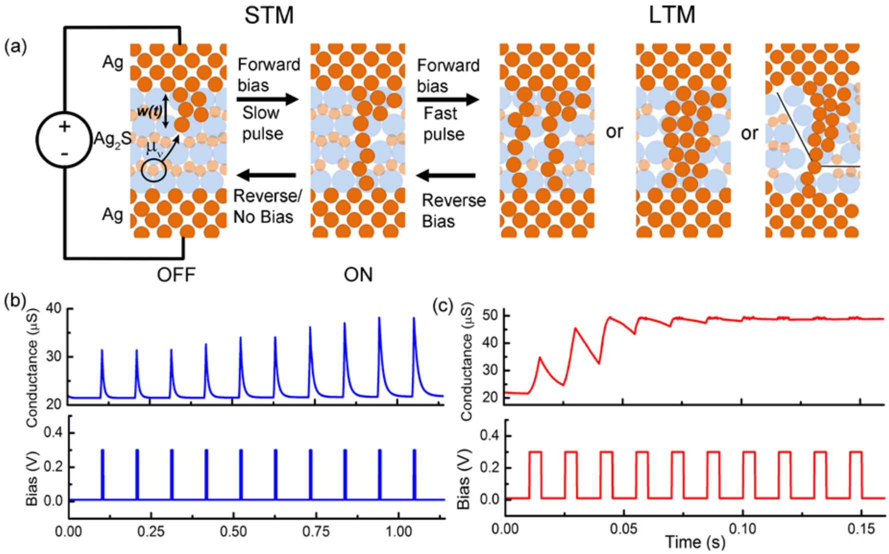


Fig. 1 *Memory behaviors of an atomic switch.* Resistance switching and memory effects in an Ag|Ag$_2$S|Ag atomic switch stems from (**a**) the increased mobility (μ_v) and migration of Ag cations under the influence of an externally applied electric field in simulation. (**b**) Sparse pulsing causes cation reduction at the cathode to form a conductive Ag filament which quickly redissolves in the absence of bias, and is observed as short term memory (STM). (**c**) Intensive pulsing (note time axes) can cause a long term filament formation (LTM) due to any combination of (1) formation of multiple filaments, (2) thicker, more stable filaments, or (3) forcing cations to irreversibly cross grain boundaries unless the electric field is reversed. Thus, atomic switches may operate between ON and OFF states, where the ON state has a slow (LTM) or fast (STM) decay time constant

75, 97, 102, 110, 119, 153]. Realization of the nanoscale memristor [32, 105, 125] has thrust this class of two-terminal circuit elements into the limelight for applications in high-density data storage and computation [151] due to their fundamental property, termed ‘memristance’. Defined as a relationship between charge and flux, memristance imparts these nanoscale elements with a capacity for both volatile and non-volatile memory capacity.

One member of this class of memristive systems is known as the atomic switch [127]. Generally comprised of a metal-insulator-metal (MIM) junction, atomic switches have been shown to clearly demonstrate quantized multi-state switching as well as input-dependent memory analogous to the short- and long-term memory observed in biological synapses [55, 95, 97, 98].

Atomic switches come in several forms, varying in the makeup of the electrodes, insulating/interfacial region, and number of terminals. In all cases, atomic switches represent a class of electroionic resistive switches known to operate through formation and annihilation of metallic filaments across an insulating gap. In particular, the most well-known Ag|Ag$_2$S system functions due to: (i) a bias-catalyzed phase transition between monoclinic acanthite (α) and body centered cubic argentite (β) Ag$_2$S and (ii) concurrent formation/dissolution of metallic filaments across the insulating gap [149] as seen in Fig. 1. Application of a bias voltage induces the formation of nanoscale channels of conducting Ag_2S across the interface through conversion of the α-Ag_2S matrix to the β-Ag_2S phase with high ionic mobility. This transition produces a weakly memristive or 'soft' switching behavior prior to the electrochemical formation of metallic Ag filaments. Removal of the applied bias voltage allows the metallic filament dissolve, restoring the thermodynamically favored, stoichiometrically balanced equilibrium state of Ag_2S, thereby returning the atomic switch to its initial high resistance 'OFF' state.

Sustained application of bias voltage generates increased flux across the interface, driving migration of mobile silver cations toward the cathode where they are reduced to Ag. This process results in the formation of a highly conductive nanowire junction, the completion of which produces a strong transition to a low resistance 'ON' state. Completion and dissolution of nanowire filaments causes large, sudden changes in conductivity, a strongly memristive behavior also called 'hard' switching. Reversing the polarity of the applied bias can oxidize the metallic Ag filament, producing mobile cations and shifting the switch back toward the weakly memristive regime. Under repetitive bipolar voltage stimulation, filaments continually form and dissolve resulting in memristive switching characterized by frequency dependent, pinched hysteresis conductance loops. Continued application of unipolar voltage stimulation reduces additional silver cations, causing both thickening of existing filaments with time and the formation of additional filaments. These processes have been shown to combine with the effect of grain boundaries within the Ag_2S to alter the decay time from the 'ON' to 'OFF' states. Large and/or quickly repeated stimulation can cause Ag^+ to migrate across grain boundaries, which then amount to a barrier against equilibration once the bias is removed, slowing the rate of conductivity decrease. These changes in volatility have been interpreted as memory analogues to the classic Atkinson and Schiffrin multi-store model of memory where sufficient rehearsal shifts stored information from short- to long-term memory [55, 97, 98].

2.2 *The Growth of a Concept*

Neuromorphic device implementations involving synthetic synapses with nanoscale dimensions are typically achieved in hybrid-CMOS/molecular CMOS [79, 82, 133, 153] architectures optimized for applications in memory and data storage. They employ either passive or active matrix crossbar-type arrays of isolated 2- and 3-terminal elements. Arrays of this type can be readily interfaced with standard CMOS circuits, allowing each synaptic element to be addressed individually in a programmatic fashion. As in the case of traditional CMOS, these configurations are designed to eliminate unwanted cross-talk by preventing the coupling of individual active units to either the environment or cross-talk to other device components. While the theoretical limits of modern lithographic fabrication methods are somewhat known, limits imposed by the functional materials themselves, e.g. gate dielectrics used to prevent leakage currents and inter-device electron tunneling, must be considered in the

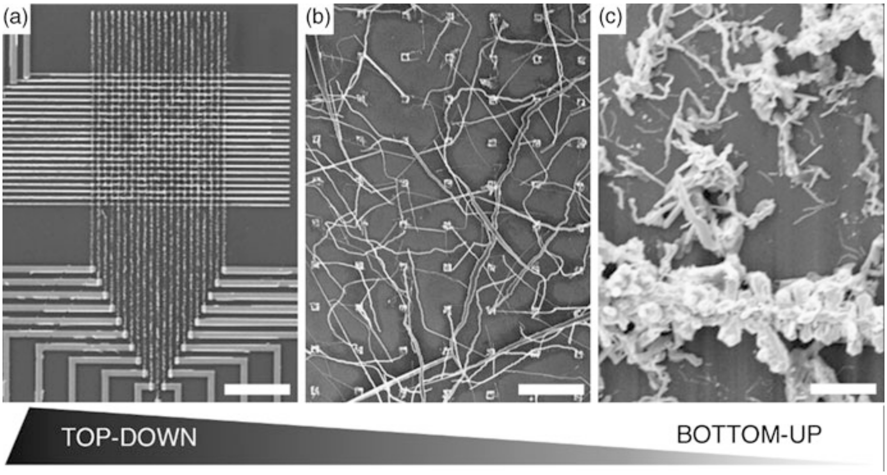


Fig. 2 *Top-down meets bottom-up in approaches to device fabrication.* Crossbar arrays prepared using top-down lithographic methods (**a**) provide independently addressable atomic switches in a highly controlled architecture. Random networks of fractals, dendrites and nanowires prepared by bottom-up self-organization (**c**) provide a complex interconnected network but with little control over topology. A blend of top-down lithography to form the basic device architecture with bottom-up self-organization through diffusion-limited electroless deposition (**b**) generates a designer architecture with complex topology

scaling limits of modern devices [143]. Thus, the development of high-density architectures remains at the mercy of not only the so-called 'cost of wiring' limit but also the fundamental limitations of designing a system that relies on functional units operating in isolation.

Atomic switches are no exception. Starting from their original form, which used the tip of a scanning tunneling microscope to form and dissolve the metallic filament across a controlled gap of nanometer dimensions, a diverse collection of operational devices have now been demonstrated using both 2- and 3-terminal configurations. While tremendous progress has been made in this effort, such highly controllable architectures (as seen in Fig. 2a) are not complex networks with respect to either physical interconnectivity or dynamical properties. They can only serve to emulate or simulate complex networks by implementing designer algorithms. However, unforeseen discoveries sometimes open doors to exciting new concepts.

Ongoing efforts to increase device density by reducing the inter-component spacing revealed interesting operational characteristics in some specific cases that prompted new ways of thinking about possible neuromorphic device architectures. Specifically, coupling between adjacent atomic switch junctions sharing a common ionic conducting layer indicated what appeared to be a competition for available metal cations between the switching junctions. These results suggested that the effects of local electroionic coupling might be applied in new device architectures that exploit the non-trivial outcomes of interconnectivity commonly seen in complex systems. What was needed was a feasible method for producing complex, highly interconnected networks of atomic switch interfaces that allowed individual units to couple and interact recurrently. What was needed was a physical analogue to this nanoarchitectonic concept [101].

Fabrication of such networks with nanoscale dimensions is not scalable from a practical and economic viewpoint using traditional lithographic methods. Fortunately, a large number of innovative, bottom-up approaches to nanofabrication based on rational design and self-organization of functional materials have been developed in recent years. Electrochemical deposition in particular has proven to be a versatile technique for the production of metallic nanostructures ranging from coatings to nanotubes. Given the makeup of the synaptic atomic switch ($Ag|Ag_2S|X$ where X represents a metal), an interest in exploiting self-organization, and a wealth of scientific literature on the generation of complex structures through the spontaneous electroless deposition of silver, early attempts to produce the ASN architecture focused on a purely bottom-up approach. Random dispersion of copper microspheres were used as seeds for a solution-phase galvanic replacement reaction with $AgNO_3$, generating dendritic silver micro- and nanostructures under diffusion limited reaction conditions. Subsequent functionalization of the network via reaction with gaseous sulfur was found to produce operational memristive switching devices with this simple benchtop protocol [112].

While refining this bottom-up approach, an unexpected observation led to a fundamental insight into the electroless deposition process and a new idea for optimizing the device fabrication process. Electroless deposition of silver from random dispersions of Cu microspheres produced not only the expected fractal and dendritic structures previously reported for identical process conditions using bulk Cu, but also interconnected nanowire networks extending over macroscopic distances as seen in Fig. 2c. Parametric examination using both dispersed microspheres and lithographically patterned Cu grids revealed a previously unreported morphological transition in the electroless deposition process as a function of the Cu seed size. At a given $AgNO_3$ concentration, larger Cu seeds consumed the available silver cations faster than diffusion replaced them, causing Mullins-Sekerka instabilities to form and propagate along the growth front, shaping the deposit into the familiar self-similar dendritic morphology [11]. For seeds smaller than the critical size, local crystalline anisotropy dominated the growth process, producing nanowires as the growth front advanced into the bulk solution fast enough to avoid Ag^+ depletion and dampen nascent instabilities. The pitch/spacing of the seeds themselves also showed a marked effect on both the density and apparent connectivity of the self-organized network structure as seen in Fig. 2b. These observed dependencies represent an important fundamental insight into nonequilibrium growth processes and provide new control parameters for the directed assembly of metallic nanostructures with specific morphology toward the goal of generating a complex, designer networks of interconnected metallic nanostructures. Functionalization of these optimized nanowire architectures has resulted in ASNs composed of more than 10^8 individual inorganic synapses/cm^2 and drastic improvements in operational device yield [10, 122].

Developments in network and device design have continued to focus on combining top-down lithographic patterning with bottom-up self-organization to create a fabrication scheme that maximizes the benefit of each approach. By extending this methodology, one can envision the design and fabrication of increasingly complex, hierarchical network architectures. In addition, the device architecture is readily scalable to the limits of current lithographic processing methods commonly used in VLSI microfabrication on 300 mm silicon wafers. Most recently, distributed electrode arrays similar to those used in experimental neuroscience to examine the electrical activity of neural cultures and cortical slices have been embedded in ASN device architectures. These electrodes provide the requisite capacity to probe the system using real-time electrical inputs and outputs with spatial specificity, enabling more detailed analysis of system dynamics.

2.3 Dynamical Circuits

Neuromorphic design strives to build the same adaptive, learning capacity found in biological neural networks into an electronic device. However, establishing specific connections between patterns of electrical activity and brain function is a difficult task. Indeed, there is as yet no Rosetta Stone of cognition. To construct a device capable of learning in a physical sense, the neuromorphic engineer must study general features of neuronal structure in order to determine which properties are essential. These features are believed to include at least synaptic plasticity, allowing physical reconfiguration of the network to enable functional differentiation and the development of hierarchical structures [1]. These hierarchies include recurrent amplifying loops [42] and feedforward transmission subassemblies [47], which all possess correlated memory distributed throughout the dynamically coupled synapses [50]. Therefore, it can be inferred that learning capacity is connected to dynamic activity within the brain. Specifically, a near-critical or ‘edge of chaos’ operational regime has been associated with the fast, correlated response to stimulation necessary for computation [76] and learning [20, 78]. Periodic oscillations in the magnitude of activity push the network back and forth across the critical-subcritical boundary, moving through and modifying an attractor landscape that appears to constitute an associative memory and action-generating mechanism [46, 49, 52, 73, 145, 146]. One possible explanation of this tendency for a complex dynamical system to remain near a critical state arises from the theory of self-organized criticality [13]. This construct proposes that a system of coupled elements with nonlinear thresholds, when subjected to slow energetic driving with respect to fast internal relaxation dynamics, will self-organize towards the critical, i.e. maximally correlated state.

Individual atomic switches demonstrate a range of nonlinear responses to input signals, including both short- and long-term memory. As in biological synaptic junctions, electric potentials drive ion transport coupled to thresholds triggering electrochemical reactions. The range of function available in a single switch gives them a technological utility in configurations where both switch terminals are directly controlled as seen in Fig. 2a. However, another class of function emerges, shown in Fig. 3, when atomic switches are massively interconnected as seen in Fig. 2b.

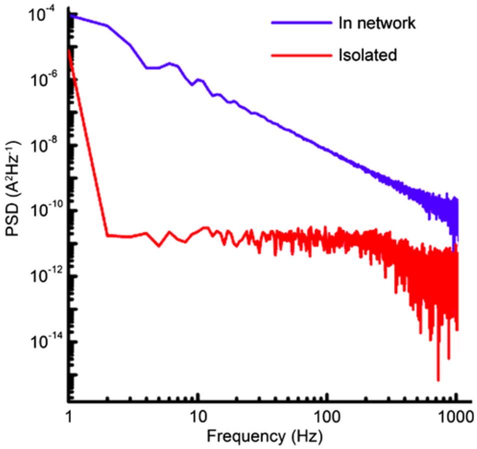


Fig. 3 *Spectral Density: Results of Networking.* Comparisons of the power spectrum of a simulated ASN network with simulation of a single, isolated atomic switch with identical parameters. The isolated atomic switch (*red*) shows a power spectrum dominated by white noise in contrast with a single element within network (*blue*) showing $1/f^{\gamma}$ power-law scaling. This demonstrates that elements in a network operate in a fundamentally different environment than isolated elements

The power spectral density of a single switch is white noise when isolated (red line), which changes to a $1/f^{\gamma}$ ($\gamma \approx 2$) power-law distribution when connected inside a network (blue line), indicating long-range correlations in switching activity. These correlations are fundamentally due to networking: an input bias which produces no frequency dependence in the activity of any given switch causes correlated activity in amongst all switches in the network. This is an example of a basic self-organizing trend in dynamical circuits possessing local Hebbian-style plasticity, enabling the strengthening of connections between correlated elements until the activity of a single node becomes indicative of the global dynamics. Ultimately, the network dynamics approach the 'edge of chaos' where global measurements of current response have demonstrated that ASNs operate in a critical regime, transitioning between macroscopic metastable states [54, 59] governed by power-law statistics [122].

These system-level similarities between ASNs and biological neural networks are interesting for studying the connection between concepts of computing and observations of brain activity. The aspiring neuromorphic engineer faces a further question: is the utility of a dynamical circuit expected to exceed the technical difficulties of designing a device that can capture that utility? At present, it is inevitable that dynamical circuits must be interfaced with more conventional electronics to produce and observe signals, and the challenge of creating some useful degree of control without excessively constricting the range of dynamic behavior is formidable. However, the incentives are palpable. Such systems would provide adaptability and learning at a rate determined by device physics, not algorithmic complexity. They would possess associative memory and are able to both classify and discerning important elements from confusing contexts. More practically, with high-speed operation and with low power consumption, all these qualities are potentially available from dynamical circuits operating in a critical or near-critical regime. From a technical perspective, the fact that the device fabrication process uses a bottom-up self-assembly approach to form the smallest features and create the architectural complexity in a manner that is readily scalable is a strong draw, especially as conventional device design struggles with the limits of miniaturization. At its core, the effort is based on the notion that technology we now possess is sufficient to harness the higher order utility that emerges when nonlinear functional elements are connected in complex dynamical circuits.

3 Modelling and Simulation of Atomic Switches: From Nodes to Networks

Despite substantial experimental study into the operational characteristics of atomic switches, they have received little attention from the perspective of modeling and simulation. The particular case of resistive switches and memristors involving a lack of junction asymmetry relative to the nature of the constituent metal electrodes prompted recent efforts to augment the standard memristive equations with additional terms that reflect the well characterized properties of individual $Ag|Ag_2S|Ag$ atomic switches [111]. Grounded upon physical observations, the simulation initially reconstructs an idealistic single switch whose memristive behavior stem from a gapless-type atomic switch. The length of the Ag filament was selected as the state variable for memristive elements and represented as $w(t) \in [0, w_0]$, where w_0 represents the junctions gap size. The voltage across each atomic switch junction is given by:

$$V(t) = \left[R_{on} \frac{w(t)}{w_0} + R_{off} \left(1 - \frac{w(t)}{w_0} \right) \right] I(t) \tag{1}$$

Here, R_{on} and R_{off} are the resistance values for the ON/OFF states and $I(t)$ the current across the interface. The rate of change in filament length can then be modeled according to:

$$\frac{dw(t)}{dt} = \left[\mu_v \frac{R_{on}}{w_0} I(t) \right] \Omega - \tau \bigl(w(t) - w_0 \bigr) + \eta(t) \tag{2}$$

where μ_v is the ionic mobility. The window function, given by:

$$\Omega = \frac{w(t)(w_0 - w(t))}{w_0^2} \tag{3}$$

was included in (2) to account for the properties of those elements with state variables at the extreme limits due to ionic drift [7, 34, 35, 99, 125, 135]. The term $[\mu_v \frac{R_{on}}{w_0} I(t)]\Omega$ provided the dependence of the filament growth rate on electronic flux, while $\tau(w(t) - w_0)$ represented a term related to filament dissolution due to the thermodynamic stability of the high resistance state. As a complement to the ionic mobility equation, a stochastic term noise term $\eta(t)$ defined as:

$$\eta = \alpha \Delta w(t) \tag{4}$$

determined the growth rate of nanowire filament dimensions, $\Delta w(t)$ being the change in filament length at time t and served to account for fluctuations in the density of available silver ions. Finally, a flux base dissolution term allowed the filament to spontaneously reduce back to its original state. This modified state equation has successfully modeled all the fundamental properties of an individual atomic switch (nonlinear hysteretic switching) as well synaptic properties such as STM and LTM seen in Fig. 1. In concert with substantial experimental investigations, effective modeling of the atomic switch provided a basis for further exploration of the underlying physics that render these devices an extremely attractive physical implementation of a biological construct.

At present, little is known regarding to the consequences of embedding memristive elements within an interconnected network architecture [99]. In this sense, the present volume is timely! To explore this concept through a merger of theory and experiment, a numerical simulation was constructed for the purpose of investigating network dynamics utilizing the well-documented physics of single atomic switches and known network topology of the ASN. The simulation scaled the system to network size and embedded atomic switches in a topology with features consistent with the known ASN architecture. Such physical networks were grown from spatially distributed nucleation sites in a lattice network. This method of construction allowed for the spontaneous construction of switches into a small-world topology. Connections occurring with their nearest neighboring nucleation sites were predominant as switches were more likely to connect to those grown from nearby. These networks exhibited characteristics of a random network with long wires extending across the entire topology connecting distant nodes.

Connectivity in the simulation was modeled similarly using an approach of bottom-up construction. Starting with a lattice of nodes, switches were assigned randomly with priority given to connecting neighboring nodes while long-range connections were assigned according to a distribution for a given network. Nearest neighbors were defined as nodes being exactly one unit length of a primitive cell away in the lattice while long-range connections could span the entire network. Normalizing the total number of connections allowed for direct comparison and classification of various network topologies. Physical parameters such as ionic mobility, R_{on}/R_{off} ratio, and average gap size were chosen according to experimental literature values [55, 97, 98] shown in Table 1, leaving only the network size and wiring density as free parameters.

Reproduction of behaviors observed in single atomic switches, i.e. memristive behavior, short- and long-term memory, and device activation served to validate the ASN simulation. In addition, the simulation faithfully reproduced the various emergent properties specific to the ASN architecture. These efforts allow for the study of

Table 1 Parameters used in the simulation were tested over ranges that are physically relevant to the Atomic Switch Network system: total gap width w_0; ionic mobility μ_v of Ag^+ in Ag_2S; R_{on}/R_{off} is the ratio of resistances at $w = 0$ and $w = w_0$; τ is the filament dissolution time constant; α modulates the level of noise in the $w(t)$ term with each time step; and N is the total number of connections

w_0 (nm)	μ_v (m^2 s$^-$1 V$^-$1)	R_{on}/R_{off}	τ (s^{-1})	α	N
Avg: 5 σ: 0–40 %	0.5×10^{-12}	Ave: 10^{-1} to 10^{-3} σ: 0–40 %	1 to 10^3	Avg: 0 σ: 0–30 % of $\Delta w(t)$	50 to 400

the internal dynamics of ASNs where it would otherwise have been impractical in experiment.

4 Characterization of the Atomic Switch Network

The physical production of such self-organized, complex network architectures, despite being an achievement, is not the final goal. Rather, it provides an opportunity to probe the characteristics of an intentionally designed complex system in hopes of harnessing its dynamical properties. To this end, substantial effort has been made to elucidate the physical mechanisms underlying operation of ASNs through a combination of experiment and simulation, the results of which have shown ASNs to retain the adaptive plasticity and memory of their component atomic switches while exhibiting emergent properties such as criticality and spontaneous switching between discrete metastable resistance states.

4.1 Device Activation

Similar to the electroforming step observed in individual memristive elements [142, 150], ASNs must undergo an activation process before they display memristive and emergent behaviors [10, 111, 122]. Freshly fabricated ASNs are essentially collections of corroded silver nanowires, since the Ag_2S interfaces are in their low-temperature insulating phase and function as nearly ohmic resistors. Bias voltage sweeps of the virgin-state networks devices exhibited weak memristive/soft switching behavior as silver cations begin to migrate into the junctions, characterized by pinched hysteresis current-voltage curves with a small R_{on}/R_{off} ratio and a smooth transition between the two states (Fig. 4a). Continued application of bias voltage produced an abrupt, nearly discontinuous jump to a state of higher conductance (Fig. 4b). Repeated stimulation with bipolar bias voltage sweeps produced strong memristive/hard switching behavior, typified by abrupt switching between two distinct resistance states (Fig. 4c). While the parameters such as threshold voltage and

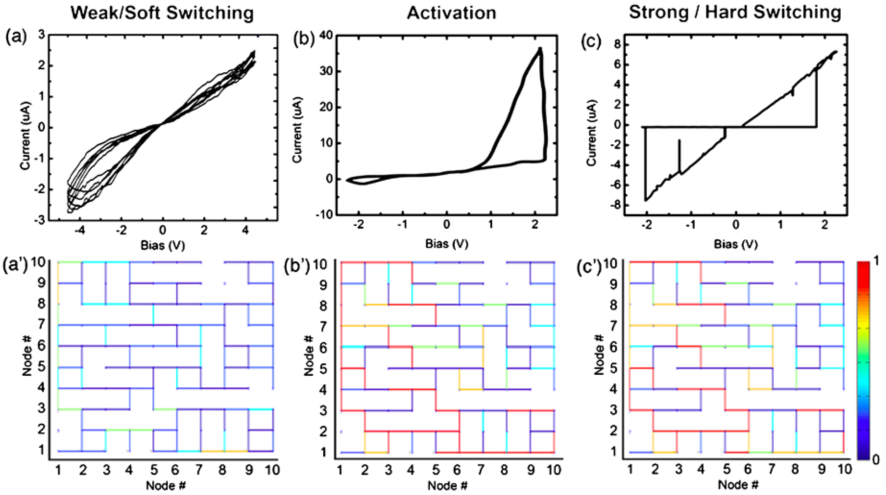


Fig. 4 *Understanding Activation.* Simulations of device activation using a 10 × 10 network under a triangle wave bias of +/−2 V at 10 Hz confirmed previously reported experimental results where (**a**) an initial weakly memristive switching($\alpha = 10$ %) repeats indefinitely until (**b**) a transition in behavior from weak (*blue*, $\alpha = 10$ %) to strong (*red*, $\alpha = 0$ %) switching occurs. Strongly memristive switching (**c**) persists indefinitely with $\alpha = 0$ %. Associated internal network connectivity maps (**a**′)–(**c**′) clearly reveal conductive pathways in the soft switching, transitional, and hard switching state as denoted by link color, with warmer color (*red*) corresponding to higher conductivity. The soft switching state (**a**′) comprises a network of low conductance switches, whereas a single dominating pathway is responsible for the high current through the system in the transitional state (**b**′). The hard switching regime (**c**′) follows, where the dominant pathway is no longer present and more distributed connections with increased conductance are observed

the R_{on}/R_{off} ratio varied from device to device due to inherent variability in self-assembled fabrication, the qualitative transition from weak to strong memristive behavior is a general property of the ASN.

This observed phase transition has been theoretically predicted in simulations of memristor networks [99] and was reproduced in ASN simulations [111]. The transition from soft to hard switching resulted from the emergence of distinct spatial patterns of individual hard and soft switching elements (Fig. 4a′). The initial weakly memristive state was characterized by a large fraction of soft switching junctions. As net flux through the network increased, connections became increasingly polarized and conductive. Continued stimulation eventually caused the formation of a percolative pathway comprised of conductive, hard switching elements path across the simulated network (Fig. 4b′). Completion of this pathway results in a dramatic change in conductance associated with the activated state and a concurrent shift from weak to strong memristive behavior. Subsequent hard switching was observed following the destruction of this highly conductive pathway, as strongly memristive elements were redistributed throughout the network, increasing the probability that connecting a given link would create an equivalent highly conductive path (Fig. 4c′). This is an example of a dynamical self-organization process: different ASNs can have very different initial conditions, yet the basic features of their functional units and network topology cause similar patterns of behavior to emerge during activation and subsequent operation.

Based on experimental and simulation results, a description of physical processes in the ASN was formulated to describe the activation process based on the two mechanisms described in Sect. 2.1: a bias-catalyzed phase transition of Ag_2S and the subsequent Ag filament formation. A weakly memristive effect is caused primarily by a distribution of phase-transition driven atomic switches, with a small fraction of filamentary driven switches. As overall conduction and the fraction of hard switching elements increases, the electric field intensifies across the remaining soft switching junctions, encouraging further filament formation. Network response changes from weak to strong memristive behavior when a percolative pathway of hard switching junctions forms across the network. Having undergone this transition, the continuously swept network operates as a hard switching memristor, since only a few local switching events are required to reconnect an equivalent path.

4.2 Memristive Properties

Following activation, ASN devices were tested to determine if their fundamental memristive properties were retained in a network configuration. Properties such as frequency dependent hysteresis and variable threshold voltages would confirm the existence of atomic switches, validate the self assembly fabrication process, and also draw attention to the interesting behavior of atomic switches in a network environment. ASN devices stimulated with repeated bipolar triangle wave bias showed a decrease in the switching threshold voltage needed to reach the low resistance state during consecutive sweeps. In individual switches, this behavior has been attributed to a progressive decrease in gap width whereby the silver ions are not provided sufficient time to fully re-dissolve into the insulator, leaving the filaments progressively more intact until the fundamental limit, where switching involves the migration of a single atom to complete or break the filament [127]. It is hypothesized that a similar process occurs in the percolative pathway which affects the entire network's conductance. Additionally, the frequency dependent switching expected for memristive systems was observed as seen in Fig. 5, whereby stimulation at increased frequency diminishes the degree of pinched hysteresis, resulting in ohmic I-V behavior at very high frequency [68, 125]. Higher input frequency offers charge carriers less time to migrate, produces more frequent reversal of direction, and causes them to simply fluctuate about their equilibrium point. In contrast, lower frequency stimulation provides sufficient time for cation migration that is sufficient not only to complete a conductive filament but to also thicken it and further increase conductance. In ASN devices, conductance has been seen to subtly decrease over an input bias frequency range of 1–20 Hz from 1.01 mS at 1 Hz to 0.57 mS at 20 Hz, suggesting that these mechanisms also mediated network device behavior [112].

Frequency dependent behavior was observed alongside changes in the R_{on}/R_{off} ratio. Occasionally during application of AC bias, devices fail to switch resistance

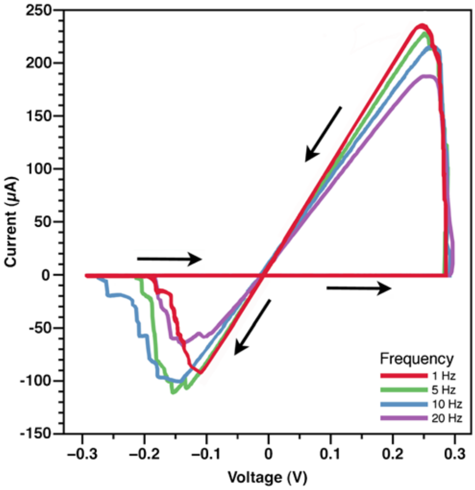


Fig. 5 *Frequency dependent hysteresis.* Investigation of ASN device operation by current-voltage spectroscopy confirmed robust hysteretic switching. As expected for a hard-switching memristive device, there is a dependence of the ON state resistance with respect to frequency, where increased hysteresis was associated with lower frequency sweeps

states, instead remaining in the OFF state for the duration of a complete voltage sweep. To quantify this behavior, a ‘switching event’ has been defined as a voltage sweep that produced a minimum current output threshold. The abundance of switching events measured as the device was stimulated repeatedly for 60 second trials at 1, 5, 10, 20, or 50 Hz decreased with increasing frequency of applied bias, in qualitative agreement with the expected trend of ohmic behavior at higher input frequency. A mechanistic explanation of this frequency dependent behavior must consider the order parameter of gap widths in the context of a network, where individual MIM junctions interact in a complex manner. At relatively low input frequencies, repeatable switching was dominated by abrupt transitions between two primary resistance states governed by the reformation or breakage of a completed pathway extending across the network. It is therefore inferred from ‘switching event’ results that at high frequency, cation migration is restricted, and there is progressively less time for a percolative pathway to completely form. Thus, the frequency dependent behavior in a network setting results not only from the thickness of Ag filaments in the percolative pathway, but also by the likelihood of forming a complete pathway in the timespan of a single sweep.

4.3 Network Plasticity

A general objective in designing a functional device platform included a direct interface the between memory/logic elements embedded in ASN architecture and externally controlled input/output contact electrodes allowing for neuromorphic data processing. After verifying the fundamental properties of atomic switches in the network setting, it was essential to understand how these external contact electrodes might be used to influence the network collectively in a manner useful for data processing.

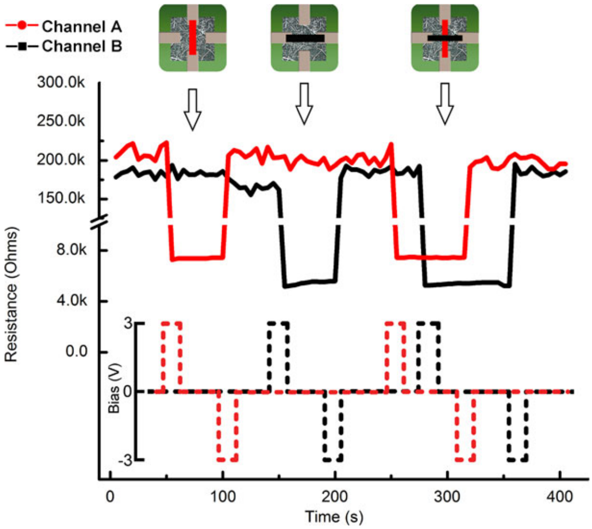


Fig. 6 *Independent Switching*. Resistance values across two spatially overlapping channels, *A* (*red*) and *B* (*black*), are plotted with a representative schematic of channel positions as insets. Applied voltage pulses across *channels A* and *B* produce distinct switching events, independently inducing low resistance values and enabling selective activation of regions within the ASN

While the dense, recurrent connectivity of ASNs has been shown to distribute switching activity throughout the network [122], the brain also exploits a capacity to form subassemblies within its neuronal network for rapid signal propagation [47]. To explore the potential formation of such feedforward assemblies through alteration of the input signal, an experiment was designed to investigate how network behavior changed as spatially overlapping regions were independently stimulated. In the current device configuration, ASNs can only be electrically probed using the macroscopic interface electrodes. It was therefore essential to confirm that these electrodes could be effectively coupled to local ensembles of atomic switches within particular spatial region of the network. As shown in Fig. 6, application of superthreshold voltage pulses (±3 V, 1 s) across different electrode pairs induced large alterations (R_{on}/R_{off} ratio $<$ 30) in conductivity. Further, controlled switching across different regions of the network can take advantage of the inherent memory capabilities of the component atomic switches, whereby the two functionally independent but spatially overlapping regions of the network could act as a simple 2-bit memory storage device.

Monitoring the conductance of all other electrode combinations throughout the stimulation regimen revealed dynamic patterns of activity in regions free from intentional manipulation. The coexistence of localized changes in network connectivity alongside complex system-wide correlations suggest a capability for autonomous, higher-dimensional information processing through formation of specialized functional regions. The recurrent structure and memory capacity of the ASN device provides operational conditions where the functional connectivity of the network itself is both dynamic and self-organized.

4.4 Emergent Properties—Harmonic Generation

The dynamical structures within ASNs produce behaviors distinct from those observed in single atomic switches. Following investigations into their general memristive behavior, experiments were performed to examine such emergent behaviors. The nonlinear response of resistive switches to charge flow can result in higher harmonic generation (HHG) for periodic input signals. Theoretical studies of memristor networks have indicated that the magnitude of HHG is strongly correlated to the fraction of network links exhibiting hard switching behavior, rising sharply once that fraction exceeds the percolation threshold [99]. This suggests HHG can be used as a quantitative measure of functional connectivity in network devices, a characteristic that is challenging to measure directly in a substantive fashion. Through experimental methods and numerical simulation, the potential for HHG in ASN devices has been examined as shown in Fig. 7. Specifically, HHG was observed to be substantially greater in functionalized ASN devices (Fig. 7b) as compared to unsulfurized Ag nanowire networks (Fig. 7a). The ratio of higher harmonic amplitudes relative to the fundamental also increased with applied input bias [10], likely due to an increase in the number of atomic switch junctions operating in the hard switching regime as seen in Fig. 7c (inset). The distribution of junction parameters (thickness, gap widths, etc.) resulting from the self-organized fabrication scheme inherently produces a similar distribution of threshold voltages across the network. These results indicated that increased bias stimulation served to activate additional atomic switch junctions within the network.

While the lithographic component of the fabrication scheme for ASNs provides some degree of control over network density, numerical simulation allows direct control and variation of network connectivity in order to explore the effects of density and topology on HHG. Simulations showed the bias magnitude required for significant HHG generation to decrease in denser networks, as increased coupling magnifies nonlinear threshold effects [111]. At increased network density, local clusters of atomic switches can form recurrent structures with enhanced sensitivity to smaller bias voltages, producing large, high frequency fluctuations. While initially explored to confirm the distributed nature of device activity and its mediation by interactions between interconnected atomic switches, measurement and analysis of HHG in ASNs has proven additionally useful as a quantitative probe into the functional connectivity of these complex devices, and will serve to guide the ongoing optimization of designer network topologies for desired operational characteristics and targeted applications.

4.5 Emergent Properties—Criticality

The presence of recurrent loops and dynamics within ASNs lead to behaviors distinct from that observed in single atomic switches. A clear example of such an emergent property is demonstrated in the persistent fluctuations in network conductivity

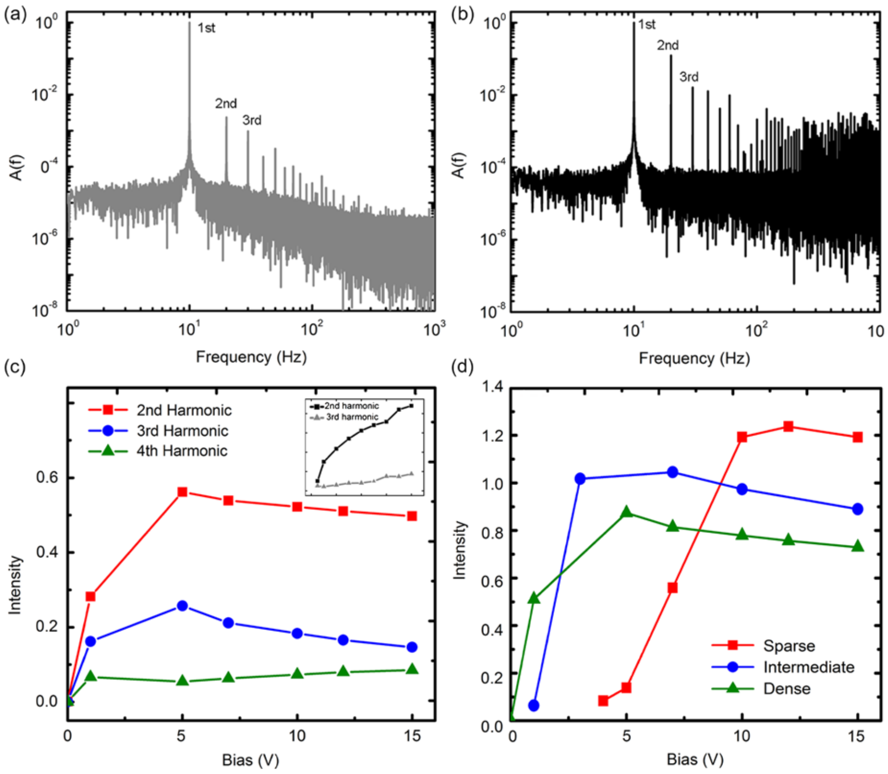


Fig. 7 *Harmonic generation.* Logarithmic plots of the Fourier transformed current outputs from AC sinusoidally driven (2 V) ASNs are shown for an (**a**) unsulfurized and (**b**) sulfurized device. Higher harmonic generation was observed for the sulfurized device suggesting increased activity. (**c**) Relative intensities of the first 3 harmonics above the 1st harmonic are plotted with respect to input amplitude for a simulated ASN. A sudden increase suggesting activation is seen at 5 V. Inset shows experimental data for comparison. (**d**) Plotting the sum of the intensities of these harmonics with respect to input amplitude for different network densities revealed a marked shift to lower input bias as density of connections/complexity of the network increased, allowing it to be more readily activated

observed in response to constant DC bias. In the case of a single switch under similar stimulation, conductivity would increase monotonically in initially quantized steps until reaching a maximum value determined by the conductivity of a completed silver filament. However, the current response of network devices has been observed to fluctuate for days under constant applied voltage, with fluctuation magnitudes on the order of the mean value [10]. This characteristic clearly illustrates how complex network connectivity inherently resists localized positive feedback of a kind that would lead to the creation of a dominant high conductivity pathway between the biased electrodes. The formation of a single filament does not simply lead to an increased potential drop across the next junction in a serial chain. Rather, it redistributes voltage across many recurrent connections that can ultimately pro-

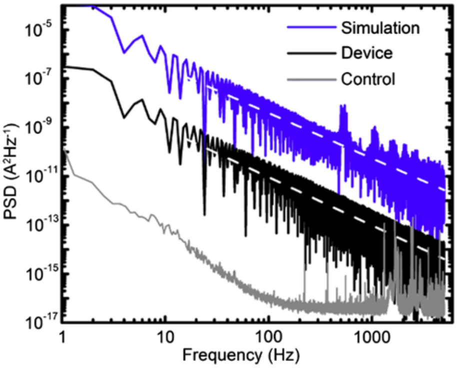


Fig. 8 *Spatiotemporal correlations.* Logarithmic plot of power spectral density indicate a power-law behavior that may lead to a critical regime of activity. Unsulfurized, purely ohmic devices such as the control (*grey*) do not exhibit critical behavior while device (*black*) and simulation (*blue*) corroboratively illustrate an emergent network dynamic due to interconnected nonlinear elements

duce a net decrease in network conductivity. The magnitude of these fluctuations is larger than any observed stochastic or random telegraph noise in an isolated conductive filament, meaning that switching behavior is correlated by network structure.

Comparing the Fourier transforms of the current response for an unsulfurized silver nanowire network with that of a functionalized device, as shown in Fig. 8, demonstrated how the formation of atomic switch junctions expands the degree of correlation in conductivity fluctuations, leading to $1/f$-like behavior across the entire sampled range. In contrast, the power spectrum of the control device spectrum flattens to white noise along with some high energy, high frequency components attributed to arcing between neighboring wires. The presence of increasing dynamic correlations suggested potential value in analyzing the system for critical behavior. By collecting time series data on the overall network current response, measured at a single (grounded) electrode, the extent of spatial correlation remained undetermined. Therefore, the analysis was focused on temporal features. Operation of ASN devices using voltage pulses produced output time traces (Fig. 9a) with characteristic plateaus, referred to as metastable conductance states, interrupted by high frequency fluctuations [122]. The duration of these metastable states was found to follow power-law statistics on timescales from sub-ms (intra-pulse) to seconds (tens of consecutive pulses) with a critical exponent of $\sim$1.8, as seen in Fig. 9b–c.

Interestingly, similar temporal correlations were not observed under DC bias, suggesting that a mechanism akin to SOC is involved, where the evolution into a critical state requires a separation of timescales between the slow external driving force and fast internal relaxation. When local dynamics contain nonlinear thresholds, the interplay of slow driving with fast relaxation causes the system to become increasingly correlated until it reaches a critical state. Given sufficient separation of time scales, an energetic balance will emerge at all the coupled interfaces such that the correlation length approaches the system size. This behavior has

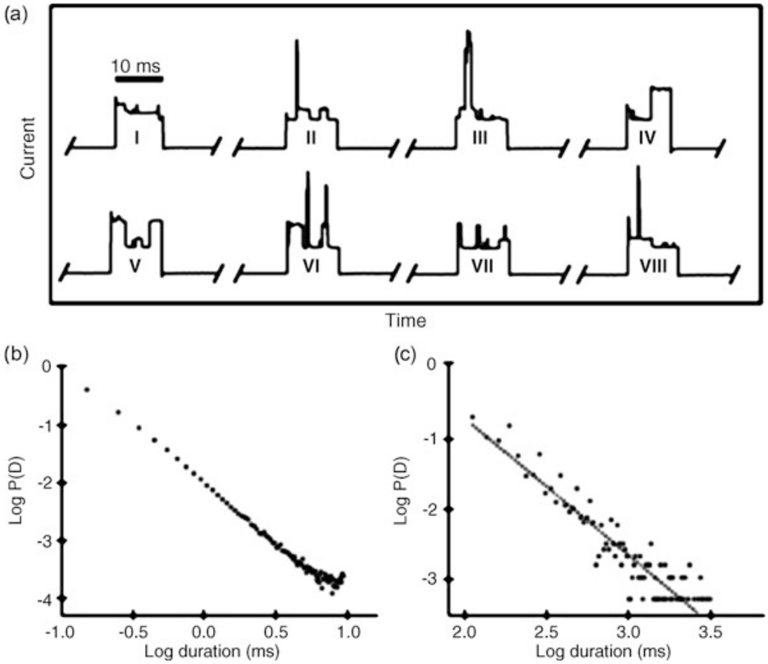


Fig. 9 *Power-law scaling of temporal metastability.* Representative examples of network current response during application of sub-threshold pulsed bias (2 V) stimulation demonstrates the existence of both increased and decreased conductivity during a single pulse (**a**) where breaks in the time axis correspond to the period between pulses (90 ms). Network reconfiguration events and temporally metastable conductive states are repeatedly observed. Temporal correlation of metastable states observed during pulsed stimulation demonstrated power-law scaling ($a \approx 1.8$) for residence time both within a single 10 ms pulse (**b**) and over 2.5 s during extended periods of pulsed stimulation (**c**)

been observed in devices fabricated from both Cu microsphere dispersions (fractals) as well as lithographically patterned Cu posts (nanowires), showing that a distinctly scale-free (fractal) physical architecture is not a prerequisite for scale-free functional network behavior. Observation of critical dynamics in the ASN provides a purpose-built experimental platform for precise examinations of criticality and its potential applicability in the fields of neuroscience and neuromorphic computation.

5 Harnessing System Dynamics

Moving beyond efforts to characterize their fundamental operation, specific classes of experimental implementations have been designed and executed which exploit the emergent dynamical properties of ASNs as introduced in Sect. 2.3 and described in detail here.

5.1 Resistance Control

While individual atomic switches have well-defined memory properties, examinations of network plasticity during the multichannel switching experiment (Sect. 4.2) indicated that functional memory might be evoked in the collective behavior of the device using stimulation in the form of an external bias voltage. A well-developed protocol for adjustable network resistance would be invaluable as the basis for information storage. Further, a survey of neuromorphic computing literature quickly reveals various models that utilize continually adjustable weights as a means to control dynamics and implement training through punishment and/or reward. If successfully executed, an adjustable network resistance protocol could serve to train targeted network regions to a particular resistance state, thereby opening the door to a multitude of neural network models that could be applied or adapted to harness the functional properties of the ASN. Unfortunately, the density of elements in the ASN makes controlling them individually impossible. To address this, an experiment referred herein as 'resistance control' was devised to evoke goal-specific behavior in response to external stimuli, namely using stimulation in the form of voltage pulses to drive the collective network resistance to an arbitrary target value.

The overarching purpose of the resistance control experiment was to first train a portion of the network to a particular resistance value, and then observe the subsequent decay to a thermodynamically stable state. This process was conducted by means of an iterative two-step cycle using the distributed electrode arrays described in Sect. 2.2. First, an electrode was chosen at random, as represented by A in Fig. 10a. The resistance between A and every other electrode was measured using a sub-threshold pulse (10 ms, 200 mV), and then used to calculate as a single global resistance for all connections as though they were resistors in parallel. Selection of a target resistance was entirely arbitrary. Second, a larger 'training' pulse (100 ms, >200 mV) was applied across the first electrode and another randomly selected electrode, denoted as B in Fig. 10b. All subsequent training pulses would be applied across these two electrodes. Following the first training pulse, the parallel resistance was calculated again, and the next training pulse was modified by polarity and/or magnitude based on whether or not the network resistance improved with respect to the target. Once the target was achieved, the training pulses ceased and the network was continually measured until the network resistance diverged to a thermodynamically stable value. This iterative training/convergence/divergence process was used to successfully train ASN devices using pulses in the range of $+/-10$ V. Network resistances could be altered with arbitrary precision (<0.5 % error) over a range from 200 Ω to 20,000,000 Ω. Training times varied widely from 1 s to 5 min or more, and target resistance states were maintained from 0.5 s to over 1 min.

While individual tests of the training/convergence/divergence process shown schematically in Fig. 11a provided proof of concept, repeating this process thousands of times yielded other interesting results. As the device cycled through trained and untrained states, close examination revealed that although the same global resistance state can be achieved multiple times, its composition was the result of one of several equivalent but markedly different microstates. Further, repeated training of an electrode pair has been seen to decrease the time required

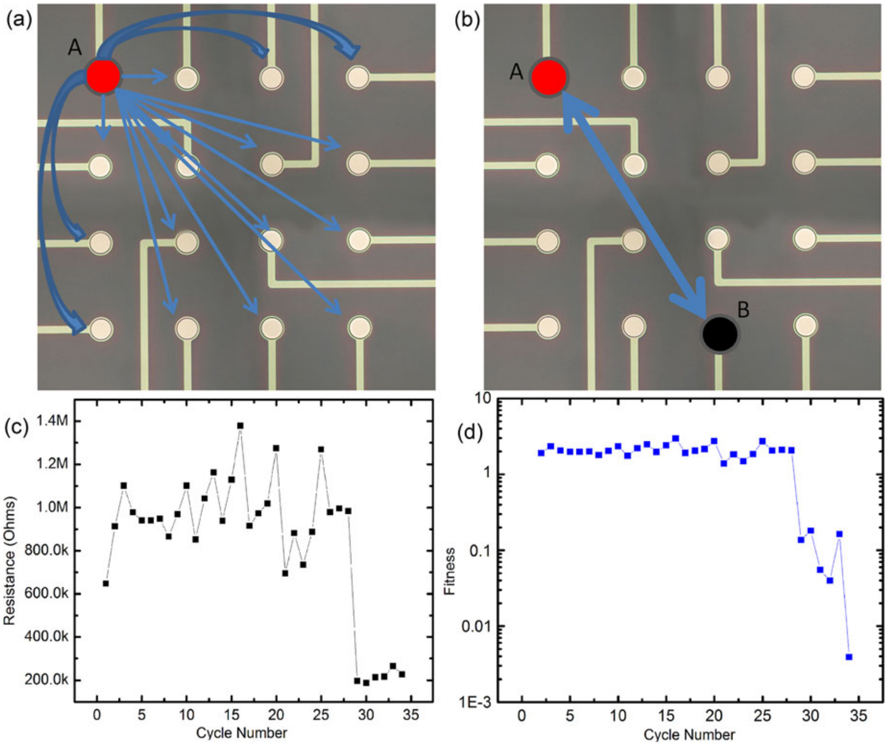


Fig. 10 *Resistance-control experiment.* The resistance optimization experiment repeats in a cycle where a stimulus pulse is administered between electrodes *A* and *B* (**b**). Using 200 mV pulses, the resistance (**c**) of electrode *A* is measured with every other electrode to monitor network reconfigurations (**a**). An automated algorithm determines if there was improvement or regression and modifies the next stimulus pulse accordingly, ending when the goal resistance is met, as determined by the fitness function (**d**)

for convergence, indicating a capacity of long-term memory. While the time a device maintains a target resistance value (dwell time) was not observed to increase after repeated training, dwell times tended to be longer at lower target resistance values as shown in Fig. 11b. This behavior is expected due to the domination of stable filament formation at lower resistance values, rather than the less thermodynamically stable α/β phase transition regime. The relationship of dwell time versus the probability of a given dwell time also followed a power-law distribution seen in Fig. 11c, where the slope varied as a function of target resistance. Collectively, these effects point toward the utility of long and short-term memories, which could be intentionally maximized by choosing a resistance state. Practically speaking, the resistance control experiment offers a way to consistently place the device into a particular resistance state, allowing for more reproducible activation or initialization of the network prior to performing other experiments.

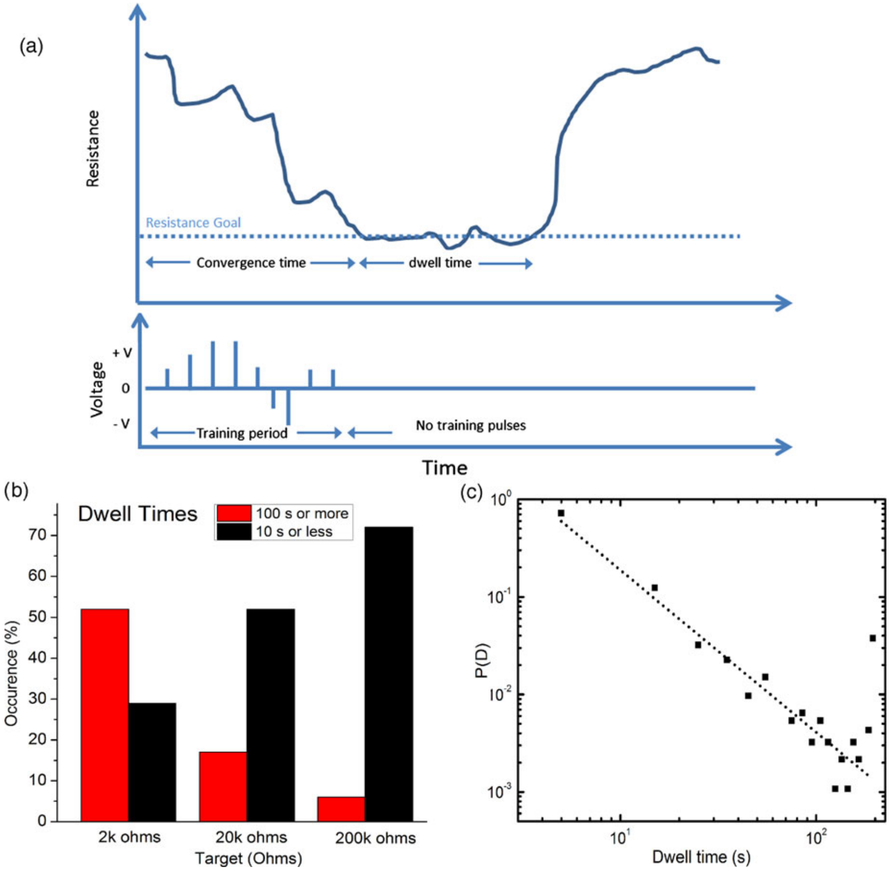


Fig. 11 *Memory capacity.* Assessing memory in ASNs through the resistance control experiment (**a**) employed both convergence times to a target and dwell times in a given state following training. (**b**) Dwell times clearly increase with decreased target resistance, e.g. increased metallization of the network and (**c**) power-law relationship for dwell time mimics that of the previously reported temporal metastable states, corresponding to specific network configurations

Based on the long times required to converge to resistance values, we may conclude that ASNs are inefficient as a ReRAM (resistive random access memory) device, or as a neural network with adjustable 'weights/resistances'. However, this experiment shows that the LTM/STM mechanisms are observable in the network setting and can be selectively emphasized by varying the resistance during operation. An exciting implication of these results is that the power-law distributions of dwell times can be interpreted as a temporally longer term manifestation of the metastable state power-law relationship, indicating that ASNs might display learning behavior in a critical regime [30, 36, 122, 140].

5.2 Reservoir Computing

Based on the knowledge gained from previous experiments, it was determined that an immediate use for ASNs lies in reservoir computing [122], where a dynamical neural-like network serves as a processing unit for solving complex systems [84]. As computational tasks become increasingly difficult, a computational paradigm with the ability to adapt and creatively solve complex systems is necessary. Works in machine-learning attempt to emulate the brain's topology of integrating neurons with random connections to understand how it does computation, specifically in RC. Unlike current computation models of explicitly programmable algorithms, RC relies on systems operating in a regime where they are able to 'learn' through experience, circumventing the need for intelligent programming. Echo State Networks (ESN) [65] and Liquid State Machines (LSM) [86] offer two distinct architectures for reservoir computation. While a brief introduction to RC will be provided here, a thorough survey of reservoir methods can be readily found in scientific literature.

Generally, RC utilizes a randomly connected network, dubbed the 'reservoir' or 'kernel', composed of coupled neurons with a topology based upon a specific neural network. Signals propagate through the network and are preferentially directed towards neurons with the greatest connective strength. A neuron integrates incoming signals via a prescribed transfer function *f*, and distributes the new signal to its outgoing connections, thereby allowing the signal to percolate throughout the network. Output neurons assigned during initialization constitute the global signal of the network, which is used to construct the desired output. Schematically, the global signal is fed to a layer of multipliers whose coefficients are determined by linear regression, then superimposed to construct the desired output. These coefficients are initially trained in a period of controlled trial runs then maintained for the duration of the set task. After the system is properly trained using sample tasks, the coefficients are fixed such that the system retains the knowledge and experience for future tasks. RC is both a simple and elegant construction that avoids the need for control over programmable elements within the recurrent network. Network performance is determined by its connectivity and the distribution of strong and weak connections, which are global on the macroscopic scale. Operational regimes are thus classified by the kernel's statistical characteristics and global parameters, focusing on emergent qualities of the network instead of individual elements. Reservoir methods are simple to execute and do not require subtle control of internal network dynamics, making it an appealing route to begin using neuromorphic devices to perform computational tasks. Despite these relative advantages, minimal progress has been made in the physical implementation of RC methods [8, 100].

Architectures for ASNs, fractals and otherwise, have been studied extensively [11] showing a level of control ideal for reservoir computation [122]. As explained in Sect. 2.2, connectivity and density of ASNs can be readily controlled by the size of the nucleation sites. Various operating regimes defined by network topology may be accessed during device fabrication in this method and further specialized in resistance control training as explained in Sect. 5.1. The robust nature

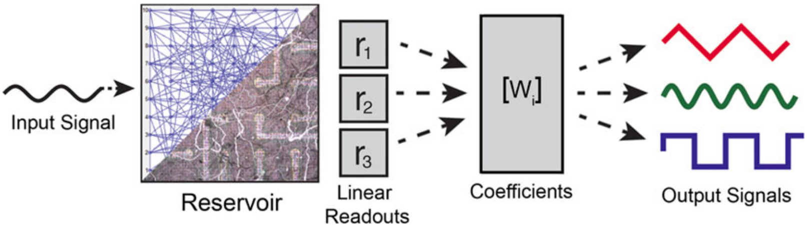


Fig. 12 *Reservoir Computing.* Schematic representation of RC. An input signal is used to excite the reservoir, which transforms the signal into higher dimensional waveforms. The activity is measured at different reservoir locations, known as linear readouts. The desired output is then constructed by using a linear combination of actual readout data using a weight matrix $[W_i]$

of complex critical systems does not require control over individual elements, making ASNs an ideal system for RC implementation. Preliminary results for RC's waveform generation task, a prerequisite for reservoir computation, revealed that embedded atomic switches have an emergent network behavior suitable for efficient kernel design in RC [111]. Performance dependencies on measurable global parameters such as higher harmonic generation were shown as potential metrics to investigate the controversies of exactly why specific reservoirs served as efficient kernels while others did not.

Standard implementation of the waveform generation task follows a simple three-step process as shown in Fig. 12. First, an input signal is used to drive the kernel into an active state. Next, the global output signal is collected and recorded from various readout nodes assigned during task initialization. Finally, the individual readout signals are used as basis vectors to construct the desired signal using least-squared linear regression. The steps are further separated into two phases of training and testing. A training period teaches the kernel using sample inputs and desired outputs, analogous to a student using an exam with solutions to study. During this time, the coefficients to be used in constructing the desired signal are calculated using least-squared linear regression and recorded. Performance is measured during the testing phase, where the actual scientific question is passed to the kernel. The three-steps are repeated using coefficients calculated during the training period for the construction of the signal. Using the student analogy, a student studies for an exam using sample questions and is subsequently tested on the actual exam. The general performance can then be measured by mean squared error, which quantifies the differences between the target and the generated waveforms:

$$MSE = \frac{\Sigma^{P_{n=1}} (y_{target} t_n - \Sigma_{i=1}^{m} W_i V_i(t_n))^2}{P} \tag{5}$$

where y_{target} is the target waveform, W_i are the weight coefficients to be trained, V_i is the signal at readout i, m total number of readouts, t$_n$ are the discreet time indices, P total time indexes.

In the waveform generation task, the RC paradigm utilizes the reservoir's ability to project the input signal into a higher dimensional representation space, thereby

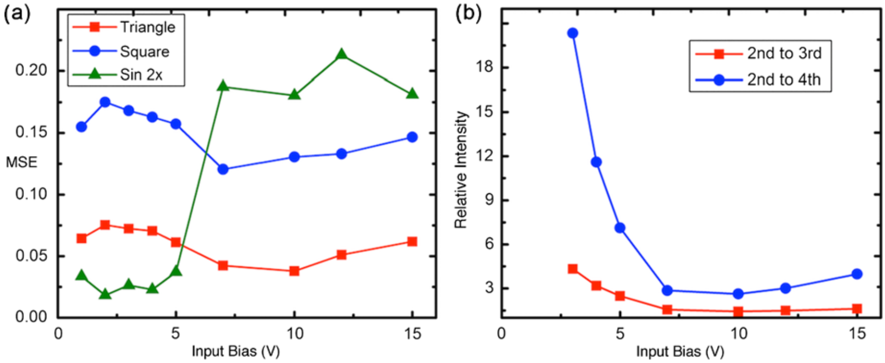


Fig. 13 *Performance of the waveform generation task* Quantified by mean-squared error, RC performance using a simulated ASN network was seen to vary with input voltage amplitude. (**a**) Periods of high relative intensity in the 2nd harmonic correlated (**b**) with high performance (low MSE) for generating double frequency sine whereas there was low performance in generating triangle and square waves

allowing for spatial temporal decomposition and reconstruction of the input signal into an arbitrary waveform. The network's complex, recurrent connections characterize its mathematical dimensionality, controlled by physical size and density of connections. However, exactly how the kernel decomposes the signal is controlled by the distribution of strong and weak connections as well as the overall topology. A solely reductionist or phenomenological viewpoint is insufficient to understand the emergent behavior of the overall network, which requires new metrics of kernel quality.

Simulation and hardware implementation of the RC waveform generation task have been conducted using the ASN platform in the aforementioned procedure [111]. The signal was defined to be the voltage with the external input as a sinusoidal voltage signal and voltage time traces were taken at the readout nodes in simulation and readout electrodes in device. Simulation efforts clarified the importance of the kernel's higher dimensional representation space in waveform generation. Variations in the amplitude of the input signal profoundly affected performance (Fig. 13a) in generating triangle, square, and higher frequency sinusoidal waveforms. The abrupt change in performance during construction of the higher frequency sinusoidal waveform was attributed to a sudden redistribution of higher harmonics. Loss in performance during higher frequency sinusoidal wave generation coincided exactly with a reduction in relative harmonic intensity of the same frequency (Fig. 13b). Conversely, kernel performance increased as the harmonic distribution shifted towards a higher range of frequencies.

Comparable results were seen in experimental implementations. Again, the distribution of higher harmonics described in Sect. 4.4 was shifted towards the higher frequencies at increased input gain (Fig. 7c inset). The redistribution of higher harmonics occurred at considerably lower voltage and progressed more continuously with respect to voltage. These trends were attributed to magnification of nonlinear

Table 2 *RC Task Performance Comparison.* Typical mean-squared error values for simulated and hardware ASNs in performing the waveform generation task. Values highlight each system's relative task difficulty, rating cosine wave generation the most difficult task for simulation and triangle wave generation most difficult for ASN device. RC implemented using the ASN device outperforms simulation in all cases

Waveform task	Simulation	Experiment
Cosine (ω_o)	0.2633	0.0028
Square (ω_o)	0.0467	0.0451
Triangle (ω_o)	0.1132	0.1071
Sine ($2\omega_o$)	0.0959	0.0910

effects resulting from a wider range of threshold voltages and higher interconnect densities in physical devices. Performance of the waveform generation task for ASN devices was compared with that achieved in simulation as shown in Table 2. Similarities in performance for most tasks re-confirm the validity of the simulation as a scaled down model of the ASN device. ASN devices performed each task with relative ease, including those requiring high frequency contributions to reservoir dynamics. In contrast, a marked difference in performance was observed for the cosine generation task, which requires a 90° phase shift transformation. The improved performance in the ASN device relative to simulation was attributed to coupling effects within its highly recurrent architecture as well as the contribution of stray capacitance, a parameter not included in the simulation.

It is evident that further implementation of ASN networks as kernels for RC will require continued investigation of higher harmonics, as well as other high-order parameters that utilize its recurrent topology. At its core, the RC paradigm utilizes a reservoir's ability to project the signal into high-dimension representation space thereby enabling the reservoir to decompose it into manageable elements and perform any mathematical operation. A joint effort of simulation and experiment is thus crucial to elucidate the complex mapping of synthetic synapses in neuromorphic devices such as ASNs. Preliminary results quickly showed the ASN as ideally suited for RC implementation and higher harmonic generation as a useful metric to characterize reservoir quality.

6 Conclusions and Outlook

Through a consideration of complex networks, self-organization, and emergent phenomena in natural systems, we have set out to lay a foundation for the use of dynamical systems as physical platforms for the implementation of emerging paradigms in neuromorphic computation and information processing. A brief presentation of historical developments in the fields of neuromorphic engineering and bioinspired computation set the stage for an introduction to recent developments in synaptic electronics using CMOS architectures. Building on these developments, we have described our recent efforts to leverage the advantages of structural control with those of self-organization through a nanoarchitectonic approach that have resulted in massively interconnected networks containing functional atomic switch elements at a density of 10^8 synthetic synapses/cm^2. These ASNs have structural characteristics similar to the biological neuropil, a synaptically rich connectivity matrix in the neocortex.

However, structural complexity alone is insufficient to produce complex behavior. As a purpose-built system designed specifically to generate emergent dynamical properties similar to those found in biological brains and neuronal networks, the operational characteristics of ASN devices were explored through simulation and experiment. Simulation efforts involving augmentation of the standard memristive equations with terms specific to the known properties of the $Ag|Ag_2S$ atomic switch have provided the capacity to not only examine the operational dependencies of individual elements on meaningful physical parameters, but to also explore the consequences of embedding atomic switches in a network setting. Validation of the numerical model and its extension to network architectures facilitates a synergy between experiment and theory in the characterization of device operation and the implementation of specific tasks.

Fundamental characterization of ASN device operation has confirmed various properties associated with atomic switches and other memristive systems, including but not limited to a requisite forming step and frequency-dependent hysteretic switching. Simulation of the forming step has provided direct insight into the nature of a theoretically predicted and experimentally observed transition between to distinct operational regimes, namely soft/hard switching. Simulated network connectivity maps of the activation process and subsequent hard/switching regime have confirmed prior indications of distributed network conductance through a collection of dynamically interacting elements. In addition, the complex structural and functional topology of ASNs has been shown to indeed produce a diversity of complex behaviors, ranging from distributed memory function to emergent critical dynamics similar to that found in both MRI/EEG of biological brains and multi-electrode array (MEA) studies of neuronal populations.

While one could imagine a broad repertoire of operations for such a complex, dynamical system, a sequence of two specific implementations has been described in order to illuminate the potential applicability of ASNs. Attempts to exploit the dynamical properties of ASNs have demonstrated a capacity to utilize controlled network plasticity in the implementation of various neural network learning models. Observations of power-law scaling in the residence (dwell) time of ‘learned’ states has additional implications for new approaches to learning using critical states. Further, the ‘fading memory’ property of the learned state has been implicated as an essential component for applications of reservoir computing. Initial progress in the use of ASNs as nonlinear reservoirs capable of task performance in the RC paradigm has been shown through simulation and experimental implementation of a benchmark task known as waveform generation. These recent efforts and associated results hold great promise for future application of the ASN as a physical platform for RC. Of particular interest are the speed, density, and scalability of the ASN, which in concert serve to overcome major hurdles in the RC paradigm.

The ability to generate synthetic device architectures comprising functional nonlinear elements such as atomic switches and other memristive systems will continue to foster an expansion of inquiry into the dynamics of complex networks, allowing for new capabilities and opportunities. Explorations that reside at the interface of complexity, nonlinear dynamics, neuroscience and engineering undoubtedly provide a vast arena for fundamental and applied research. Moreover, such interdisciplinary approaches represent an important step forward in our quest to understand nature's networks, including the ever-elusive human brain. Moving forward, efforts to leverage the promise of atomic switch networks as a neuromorphic platform will certainly included more advanced applications in memory and learning. In addition, the functional diversity demonstrated to date has strong implications for using the ASN as a universal approximator of dynamical systems, not only as a physical device for information processing and computation but also as a scalable experimental platform for investigating theoretical constructs of complexity and neuroscience.

Acknowledgements This work was partially supported by the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) World Premier International (WPI) Research Center for Materials Nanoarchitectonics (MANA), HRL Laboratories, and the Defense Advanced Research Projects Agency (DARPA)—Physical Intelligence Program (BAA-09-63), US Department of Defense. The authors acknowledge use of the Nanoelectronics Research Facility (NRF) and Nano and Pico Characterization Laboratory (NPC) at the University of California, Los Angeles.

References

1. Abbott, L.F., Nelson, S.B.: Synaptic plasticity: taming the beast. Nat. Neurosci. **3**, 1178–1183 (2000)
2. Achard, S., Bullmore, E.: Efficiency and cost of economical brain functional networks. PLoS Comput. Biol. **3**, e17 (2007)
3. Afifi, A., Ayatollahi, A., Raissi, F.S.TD.P.: Implementation using memristive nanodevice in CMOS-nano neuromorphic networks. IEICE Electron. Express **6**, 148–153 (2009)
4. Alivisatos, A.P., Chun, M., Church, G.M., Greenspan, R.J., Roukes, M.L., Yuste, R.: The brain activity map project and the challenge of functional connectomics. Neuron **74**, 970–974 (2012)
5. Alivisatos, A.P., Chun, M., Church, G.M., Deisseroth, K., Donoghue, J.P., Greenspan, R.J., McEuen, P.L., Roukes, M.L., Sejnowski, T.J., Weiss, P.S., et al.: The brain activity map. Science (2013)
6. Ananthanarayanan, R., Esser, S.K., Simon, H.D., Modha, D.S.: The cat is out of the bag: cortical simulations with 109 neurons, 1013 synapses. In: ACM Request Permissions (2009)
7. Anderson, P.: More is different. Science **177**, 393–396 (1972)
8. Appeltant, L., Soriano, M.C., Van der Sande, G., Danckaert, J., Massar, S., Dambre, J., Schrauwen, B., Mirasso, C.R., Fischer, I.: Information processing using a single dynamical node as complex system. Nat. Commun. **2**, 468 (2011)
9. Atiya, A.F., Parlos, A.G.: New results on recurrent network training: unifying the algorithms and accelerating convergence. IEEE Trans. Neural Netw. **11**, 697–709 (2000)
10. Avizienis, A.V., Sillin, H.O., Martin-Olmos, C., Shieh, H.H., Aono, M., Stieg, A.Z., Gimzewski, J.K.: Neuromorphic atomic switch networks. PLoS ONE **7**, e42772 (2012)
11. Avizienis, A.V., Martin-Olmos, C., Sillin, H.O., Aono, M., Gimzewski, J.K., Stieg, A.Z.: Morphological transitions from dendrites to nanowires in the electroless deposition of silver. Cryst. Growth Des. **13**, 465–469 (2013)

12. Bak, P., Paczuski, M.: Complexity contingency, and criticality. Proc. Natl. Acad. Sci. USA **92**, 6689 (1995)
13. Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality. Phys. Rev. A **38**, 364–374 (1988)
14. Barabási, A.L., Albert, A.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)
15. Barabási, A.L., Ravasz, E., Vicsek, T.: Deterministic scale-free networks. Physica A **299**, 559–564 (2001)
16. Basheer, I.A., Hajmeer, M.: Artificial neural networks: fundamentals, computing, design, and application. J. Microbiol. Methods **43**, 3–31 (2000)
17. Bassett, D.S., Greenfield, D.L., Meyer-Lindenberg, A., Weinberger, D.R., Moore, S.W., Bullmore, E.T.: Efficient physical embedding of topologically complex information processing networks in brains and computer circuits. PLoS Comput. Biol. **6**, e1000748 (2010)
18. Bassett, D.S., Wymbs, N.F., Porter, M.A., Mucha, P.J., Carlson, J.M., Grafton, S.T.: Dynamic reconfiguration of human brain networks during learning. Proc. Natl. Acad. Sci. USA **108**, 7641 (2011)
19. Beggs, J., Plenz, D.: Neuronal avalanches in neocortical circuits. J. Neurosci. **23**, 11167 (2003)
20. Bertschinger, N., Natschläger, T.: Real-time computation at the edge of chaos in recurrent neural networks. Neural Comput. **16**, 1413–1436 (2004)
21. Binder, P.M.: Computation: the edge of reductionism. Nature **459**, 332–334 (2009)
22. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.: Complex networks: structure and dynamics. Phys. Rep. **424**, 175–308 (2006)
23. Borghetti, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R., Williams, R.S.: 'Memristive', switches enable 'Stateful', logic operations via material implication. Nature **464**, 873–876 (2010)
24. Bornholdt, S., Roehl, T.: Self-organized critical neural networks. Phys. Rev. E **67**, 066118 (2003)
25. Bullmore, E., Sporns, O.: The economy of brain network organization. Nat. Rev. Neurosci. (2012)
26. Chang, T., Jo, S.-H., Kim, K.-H., Sheridan, P., Gaba, S., Lu, W.: Synaptic behaviors and modeling of a metal oxide memristive device. Appl. Phys. A **102**, 857–863 (2011)
27. Chang, T., Jo, Short-Term, S.: Memory to long-term memory transition in a nanoscale memristor. ACS Nano (2011)
28. Chialvo, D.: Critical brain networks. Phys. A, Stat. Mech. Appl. **340**, 756–765 (2004)
29. Chialvo, D.R.: Emergent complex neural dynamics. Nat. Phys. **6**, 744–750 (2010)
30. Chialvo, D.R., Bak, P.: Learning from mistakes. arXiv 1997, adap-org, 7006
31. Choi, H., Jung, H., Lee, J., Yoon, J., Park, J., Seong, D., Lee, W., Hasan, M., Jung, G., Hwang, H.: An electrically modifiable synapse array of resistive switching memory. Nanotechnology **20**, 345201 (2009)
32. Chua, L.O.: Memristor-the missing circuit element. IEEE Trans. Circuit Theory **18**, 507–519 (1971)
33. Clauset, A., Shalizi, C.R., Newman, M.E.J.: Power-law distributions in empirical data. SIAM Rev. **51**, 661–703 (2009)
34. Cross, M.C., Hohenberg, P.C.: Pattern formation outside of equilibrium. Rev. Mod. Phys. **65**, 851–1112 (1993)
35. Crutchfield, J.P.: Between order and chaos. Nat. Phys. **8**, 17–24 (2012)
36. De Arcangelis, L., Herrmann, H.: Learning as a phenomenon occurring in a critical state. Proc. Natl. Acad. Sci. USA **107**, 3977 (2010)
37. De Arcangelis, L., Perrone-Capano, C., Herrmann, Self-Organized, H.: Criticality model for brain plasticity. Phys. Rev. Lett. **96** (2006)
38. DeFelipe, J.: From the connectome to the synaptome: an epic love story. Science **330**, 1198–1201 (2010)

39. Diorio, C., Hasler, P., Minch, A., Mead, C.A.: A single-transistor silicon synapse. IEEE Trans. Electron Devices **43**, 1972–1980 (1996)
40. Dorogovtsev, S.N., Mendes, J.F.F.: Evolution of networks. Adv. Phys. **51**, 1079–1187 (2002)
41. Dorogovtsev, S., Goltsev, A., Mendes, J.: Critical phenomena in complex networks. Rev. Mod. Phys. **80**, 1275–1335 (2008)
42. Douglas, R., Koch, C., Mahowald, M., Martin, K., Suarez, H.: Recurrent excitation in neocortical circuits. Science **269**, 981–985 (1995)
43. Eguiluz, V.M., Chialvo, D.R., Cecchi, G.A., Baliki, M., Apkarian, A.V.: Scale-free brain functional networks. Phys. Rev. Lett. **94** (2005)
44. Fraiman, D., Balenzuela, P., Foss, J., Chialvo, D.: Ising-like dynamics in large-scale functional brain networks. Phys. Rev. E **79** (2009)
45. Frank, D.J.: Power-constrained CMOS scaling limits. IBM J. Res. Dev. **46**, 235–244 (2002)
46. Freeman, W.J.W., Kozma, R.R., Werbos, P.J.P.: Biocomplexity: adaptive behavior in complex stochastic dynamical systems. Biosystems **59**, 109–123 (2001)
47. Ganguli, S., Huh, D., Sompolinsky, H.: Memory traces in dynamical systems. Proc. Natl. Acad. Sci. USA **105**, 18970–18975 (2008)
48. Gao, J., Buldyrev, S.V., Stanley, H.E., Havlin, S.: Networks formed from interdependent networks. Nat. Phys. **8**, 40–48 (2011)
49. Garlaschelli, D., Capocci, A., Caldarelli, G.: Self-organized network evolution coupled to extremal dynamics. Nat. Phys. **3**, 813–817 (2007)
50. Goldman, M.S.: Memory without feedback in a neural network. Neuron **61**, 621–634 (2009)
51. Goldstein, J.: Emergence as a construct: history and issues. Emergence **1**, 49–72 (1999)
52. Gross, T., Blasius, B.: Adaptive coevolutionary networks: a review. J. R. Soc. Interface **5**, 259–271 (2008)
53. Haimovici, A., Tagliazucchi, E., Balenzuela, P., Chialvo, D.R.: Brain organization into resting state networks emerges at criticality on a model of the human connectome. Phys. Rev. Lett. **110**, 178101 (2013)
54. Haldeman, C., Beggs, J.: Critical branching captures activity in living neural networks and maximizes the number of metastable states. Phys. Rev. Lett. **94**, 058101 (2005)
55. Hasegawa, T., Ohno, T., Terabe, K., Tsuruoka, T., Nakayama, T., Gimzewski, J.K., Aono, Learning, M.: Abilities achieved by a single solid-state atomic switch. Adv. Mater. (2010)
56. Hasegawa, T., Terabe, K., Tsuruoka, T., Aono, Atomic, M.: Switch: atom/ion movement controlled devices for beyond Von-Neumann computers. Adv. Mater. (2011)
57. Hassoun, M.H.: Fundamentals of artificial neural networks. Proc. IEEE **84**, 906 (1996)
58. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA **79**, 2554 (1982)
59. Hopfield, J.J.: Artificial neural networks. IEEE Circuits Devices Mag. **4**, 3–10 (1988)
60. Husband, C., Husband, S., Daniels, J., Tour, J.: Logic and memory with nanocell circuits. IEEE Trans. Electron Devices **50**, 1865–1875 (2003)
61. Indiveri, G., Chicca, E., Douglas, R.J.: Artificial cognitive systems: from VLSI networks of spiking neurons to neuromorphic cognition. Cogn. Comput. **1**, 119–127 (2009)
62. Indiveri, G.G., Linares-Barranco, B.B., Hamilton, T.J.T., van Schaik, A.A., Etienne-Cummings, R.R., Delbruck, T.T., Liu, S.-C.S., Dudek, P.P., Häfliger, P.P., Renaud, S.S., et al.: Neuromorphic silicon neuron circuits. Front. Neurosci. **5**, 73 (2011)
63. International Technology Roadmap for Semiconductors (2003)
64. Jaeger, H.: The "Echo State" approach to analysing and training recurrent neural networks-with an erratum note'. Technology GMD Technical Report, 148, German National Research Center for Information, Bonn, Germany (2001)
65. Jaeger, H.: Adaptive nonlinear system identification with echo state networks. Networks **8**, 9 (2003)
66. Jensen, H.J., Self-organized criticality: emergent complex behavior in physical and biological systems 10 (1998)
67. Jeong, D.S., Kim, I., Ziegler, M., Kohlstedt, H.: Towards artificial neurons and synapses: materials point of view. RSC Adv. (2012)

68. Joglekar, Y.N., Wolf, S.J.: The elusive memristor: properties of basic electrical circuits. arXiv 2008, cond-mat.mes-hall
69. Johansen-Berg, H.: Human connectomics—what will the future demand? NeuroImage 1–5 (2013)
70. Kelso, J.A.S.: Dynamic Patterns. MIT Press, Cambridge (1997)
71. Kim, K.-H., Gaba, S., Wheeler, D., Cruz-Albrecht, J.M., Hussain, T., Srinivasa, N., Lu, W.: A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. Nano Lett. **12**, 389–395 (2012)
72. Kitzbichler, M.G., Smith, M.L., Christensen, S.R., Bullmore, E.: Broadband criticality of human brain network synchronization. PLoS Comput. Biol. **5**, e1000314 (2009)
73. Kozma, R., Puljic, M., Balister, P., Bollobás, B., Freeman, W.J.: Phase transitions in the neuropercolation model of neural populations with mixed local and non-local interactions. Biol. Cybern. **92**, 367–379 (2005)
74. Kuzum, D., Jeyasingh, R.G.D., Lee, B., Wong, H.-S.P.: Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. Nano Lett. **12**, 2179–2186 (2012)
75. Kuzum, D., Jeyasingh, R.G.D., Yu, S., Wong, H.-S.: Low-energy robust neuromorphic computation using synaptic devices (2012)
76. Langton, C.: Computation at the edge of chaos—phase-transitions and emergent computation. Physica D **42**, 12–37 (1990)
77. Lazar, A.: SORN: a self-organizing recurrent neural network. Front. Comput. Neurosci. **3** (2009)
78. Legenstein, R., Maass, W.: Edge of chaos and prediction of computational performance for neural circuit models. Neural Netw. **20**, 323–334 (2007)
79. Likharev, K., Strukov, D.C.MO.L.: Devices, circuits, and architectures. Introd. Mol. Electron. 447–477 (2005)
80. Likharev, K., Mayr, A., Muckra, I., Türel, Ö.: CrossNets: high-performance neuromorphic architectures for CMOL circuits. Ann. N.Y. Acad. Sci. **1006**, 146–163 (2003)
81. Linkenkaer-Hansen, K., Nikouline, V.V., Palva, J.M., Ilmoniemi, R.J.: Long-range temporal correlations and scaling behavior in human brain oscillations. J. Neurosci. **21**, 1370–1377 (2001)
82. Lu, W., Lieber, C.M.: Nanoelectronics from the bottom up. Nat. Mater. **6**, 841–850 (2007)
83. Lukosevicius, J.: Reservoir computing approaches to recurrent neural network training. Comput. Sci. Rev. **3**, 23 (2009)
84. Lukosevicius, M., Jaeger, H., Schrauwen, B.: Reservoir computing trends. Künstl. Intell. 1–7 (2012)
85. Lundstrom, M.: Applied physics: enhanced: Moore's law forever? Science **299**, 210–211 (2003)
86. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput. **14**, 2531–2560 (2002)
87. Mahowald, M., Douglas, R.: A silicon neuron. Nature **354**, 515–518 (1991)
88. Marconi, E., Nieus, T., Maccione, A., Valente, P., Simi, A., Messa, M., Dante, S., Baldelli, P., Berdondini, L., Benfenati, F.: Emergent functional properties of neuronal networks with controlled topology. PLoS ONE **7**, e34648 (2012)
89. Markram, H.: The human brain project. Sci. Am. **306**, 50–55 (2012)
90. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biol. **5**, 115–133 (1943)
91. Mead, C.: Neuromorphic electronic systems. Proc. IEEE **78**, 1629–1636 (1990)
92. Meunier, Hierarchical, D.: Modularity in human brain functional networks. Front. Neuroinform. **3** (2009)
93. Modha, D.S.D., Singh, R.R.: Network architecture of the long-distance pathways in the macaque brain. Proc. Natl. Acad. Sci. USA **107**, 13485–13490 (2010)
94. Morgan, J.L., Lichtman, J.W.: Why not connectomics? Nat. Chem. Biol. **10**, 494–500 (2013)

95. Nayak, A., Ohno, T., Tsuruoka, T., Terabe, K., Hasegawa, T., Gimzewski, J.K., Aono, M.: Controlling the synaptic plasticity of a Cu2S gap-type atomic switch. Adv. Funct. Mater. (2012)
96. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. **45**, 167–256 (2003)
97. Ohno, T.: Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. Nat. Mater. **10**, 591–595 (2011)
98. Ohno, T., Hasegawa, T., Nayak, A., Tsuruoka, T., Gimzewski, J.K., Aono, M.: Sensory and short-term memory formations observed in a Ag2S gap-type atomic switch. Appl. Phys. Lett. **99**, 203108 (2011)
99. Oskoee, N., Sahimi, M.: Electric currents in networks of interconnected memristors. Phys. Rev. E **83**, 031105 (2011)
100. Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., Massar, S.: Optoelectronic reservoir computing. Sci. Rep., 2 (2012)
101. Pask, G.: Physical analogues to the growth of a concept. In: Proceedings of a Symposium Held at the National Physical Laboratory on Mechanisation of Thought Processes, p. 2 (1958)
102. Pickett, M.D., Medeiros-Ribeiro, G., Williams, R.S.: A scalable neuristor built with Mott memristors. Nat. Mater. **12**, 114–117 (2012)
103. Plenz, D.: The critical brain. Physics **6**(47), 1–3 (2013)
104. Poon, C.-S.: Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities 1–3 (2011)
105. Prodromakis, T., Toumazou, C., Chua, L.: Centuries of memristors. Nat. Mater. **11**, 478–481 (2012)
106. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. **65**, 386 (1958)
107. Ryle, G.: The Concept of Mind. University of Chicago Press, Chicago (1949)
108. Schemmel, J., Bruderle, D., Grubl, A., Hock, M., Meier, K., Millner, S.: A wafer-scale neuromorphic hardware system for large-scale neural modeling 1947–1950 (2010)
109. Schrauwen, B., Verstraeten, D., Van Campenhout, J.: An overview of reservoir computing: theory, applications and implementations. In: Proceedings of the 15th European Symposium on Artificial Neural Networks, pp. 471–482 (2007)
110. Seo, K., Kim, I., Jung, S., Jo, M., Park, S., Park, J., Shin, J., Biju, K.P., Kong, J., Lee, K., et al.: Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device. Nanotechnology **22**, 254023 (2011)
111. Sillin, H.O., Aguilera, R., Shieh, H.H., Avizienis, A.V., Aono, M., Stieg, A.Z., Gimzewski, J.K.: A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. Nanotechnology **38**(24), 384004 (2013)
112. Sillin, H.O., Sandouk, E.J., Avizienis, A.V., Aono, M., Stieg, A.Z., Gimzewski, J.K.: Benchtop fabrication of memristive atomic switch networks. J. Nanosci. Nanotechnol. **24**, 1–7 (2013)
113. Simon, H.: The architecture of complexity. Proc. Am. Philos. Soc. 467–482 (1962)
114. Song, C., Havlin, S., Makse, H.A.: Self-similarity of complex networks. Nature **433**, 392–395 (2005)
115. Sporns, O.: Small-world connectivity, motif composition, and complexity of fractal neuronal connections. Biosystems **85**, 55–64 (2006)
116. Sporns, O., Tononi, G., Edelman, G.: Connectivity and complexity: the relationship between neuroanatomy and brain dynamics. Neural Netw. **13**, 909–922 (2000)
117. Sporns, O., Chialvo, D., Kaiser, M., Hilgetag, C.: Organization development and function of complex brain networks. Trends Cogn. Sci. **8**, 418–425 (2004)
118. Sporns, O., Tononi, G., Kötter, R.: The human connectome: a structural description of the human brain. PLoS Comput. Biol. **1**, e42 (2005)
119. Srinivasa, N.N., Cruz-Albrecht, J.J.: Neuromorphic adaptive plastic scalable electronics: analog learning systems. IEEE Pulse **3**, 51–56 (2012)

120. Stanley, H.E.: Introduction to Phase Transitions and Critical Phenomena (1987)
121. Steil, J.J.: Backpropagation-decorrelation: online recurrent learning with O(N) complexity. In: Proceedings IEEE International Joint Conference on Neural Networks, vol. 2, pp. 843 (2004)
122. Stieg, A.Z., Avizienis, A.V., Sillin, H.O., Martin-Olmos, C., Aono, M., Gimzewski, J.K.: Emergent criticality in complex turing B-type atomic switch networks. Adv. Mater. **24**, 286–293 (2011)
123. Strogatz, S.H.: Exploring complex networks. Nature **410**, 268–276 (2001)
124. Strukov, D.B., Likharev, K.K.: CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices. Nanotechnology **16**, 888–900 (2005)
125. Strukov, D., Snider, G., Stewart, D., Williams, R.: The missing memristor found. Nature **453**, 80–83 (2008)
126. Stumpf, M.P.H., Porter, M.A.: Critical truths about power laws. Science **335**, 665–666 (2012)
127. Terabe, K., Hasegawa, T., Nakayama, T., Aono, M.: Quantized conductance atomic switch. Nature **433**, 47–50 (2005)
128. Tononi, G.: Consciousness and complexity. Science **282**, 1846–1851 (1998)
129. Tononi, G., Sporns, O., Edelman, G.: A measure for brain complexity: relating functional segregation and integration in the nervous system. Proc. Natl. Acad. Sci. USA **91**, 5033 (1994)
130. Tononi, G., Edelman, G.M., Sporns, O.: Complexity and coherency: integrating information in the brain. Trends Cogn. Sci. **2**, 474–484 (1998)
131. Tour, J., Van Zandt, W., Husband, C., Husband, S., Wilson, L., Franzon, P., Nackashi, D.: Nanocell logic gates for molecular computing. IEEE Trans. Nanotechnol. **1**, 100–109 (2002)
132. Turcotte, D.L.S.-O.: Self-organized criticality. Rep. Prog. Phys. **62**, 1377–1429 (1999)
133. Türel, Ö., Lee, J.H., Ma, X., Likharev, K.K.: Neuromorphic architectures for nanoelectronic circuits. Int. J. Circuit Theory Appl. **32**, 277–302 (2004)
134. Turing, A.M.: Computing machinery and intelligence. Mind **59**, 433 (1950)
135. Turing, A.M.: The chemical basis of morphogenesis. Philos. Trans. R. Soc. Lond., Part B **237**, 37–72 (1953)
136. van den Heuvel, M.P., Stam, C.J., Kahn, R.S., Hulshoff Pol, H.E.: Efficiency of functional brain networks and intellectual performance. J. Neurosci. **29**, 7619–7624 (2009)
137. Versace, M., Chandler, B.: The brain of a new machine. IEEE Spectr. **47**, 30–37 (2010)
138. Verschure, P.: Connectionist explanation: taking positions in the mind-brain dilemma. In: Neural Networks and a New Artificial Intelligence, pp. 133–188 (1997)
139. Verstraeten, D., Schrauwen, B., D'Haene, M., Stroobandt, D.: Special issue: an experimental unification of reservoir computing methods. Neural Netw. **20**, 391–403 (2007)
140. Von Neumann, J.: The principles of large-scale computing machines. IEEE Ann. Hist. Comput. **3**, 263–273 (1981)
141. Von Neumann, J.: The Computer and the Brain. Yale University Press, New Haven (2012)
142. Wang, X., Chen, G.: Complex networks: small-world, scale-free and beyond. IEEE Circuits Syst. Mag. **3**, 6–20 (2003)
143. Waser, R., Aono, M.: Nanoionics-based resistive switching memories. Nat. Mater. **6**, 833–840 (2007)
144. Watts, D.J., Strogatz, S.H.: Collective dynamics of "Small-World" networks. Nature **393**, 440–442 (1998)
145. Werner, G.: Metastability, criticality and phase transitions in brain and its models. Biosystems **90**, 496–508 (2007)
146. Werner, G.: Viewing brain processes as critical state transitions across levels of organization: neural events in cognition and consciousness, and general principles. Biosystems **96**, 114–119 (2009)
147. Wiener, N.: Cybernetics, Second Edition: or the Control and Communication in the Animal and the Machine (1965)

148. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Comput. **1**, 270–280 (1989)
149. Xu, Z., Bando, Y., Wang, W., Bai, X., Golberg, D.: Real-time in situ HRTEM-resolved resistance switching of Ag2S nanoscale ionic conductor. ACS Nano **4**, 2515–2522 (2010)
150. Yang, J.J., Pickett, M.D., Li, X., Ohlberg, D.A.A., Stewart, D.R., Williams, R.S.: Memristive switching mechanism for Metal/Oxide/Metal nanodevices. Nat. Nanotechnol. **3**, 429–433 (2008)
151. Yang, J.J., Strukov, D.B., Stewart, D.R.: Memristive devices for computing. Nat. Nanotechnol. **8**, 13–24 (2013)
152. Yegnanarayana, B.: Artificial Neural Networks (2004)
153. Zhao, W.S., Agnus, G., Derycke, V., Filoramo, A., Bourgoin, J.-P., Gamrat, C.: Nanotube devices based crossbar architecture: toward neuromorphic computing. Nanotechnology **21**, 175202 (2010)

Spike-Timing-Dependent-Plasticity with Memristors

T. Serrano-Gotarredona, T. Masquelier, and B. Linares-Barranco

Abstract (This chapter is reprints material from Zamarreño-Ramos et al. in Front. Neurosci. 5:26, 2011 and Serrano-Gotarredona et al. in Front. Neurosci. 7:02, 2013, with permission.) Here we present a very exciting overlap between emergent nano technology and neuroscience, which has been discovered by neuromorphic engineers. Specifically, we are linking one type of memristor nano technology devices to the biological synaptic update rule known as Spike-Time-Dependent-Plasticity found in real biological synapses. Understanding this link allows neuromorphic engineers to develop circuit architectures that use this type of memristors to artificially emulate parts of the visual cortex. We focus on the type of memristors referred to as voltage or flux driven memristors and focus our discussions on behavioral macro models for such devices. The implementations result in fully asynchronous architectures with neurons sending their action potentials not only forwards but also backwards. One critical aspect is to use neurons that generate spikes of specific shapes. We will see how by changing the shapes of the neuron action potential spikes we can tune and manipulate the STDP learning rules for both excitatory and inhibitory synapses. We will see how neurons and memristors can be interconnected to achieve large scale spiking learning systems, that follow a type of multiplicative STDP learning rule. We will briefly extend the architectures to use three-terminal transistors with similar memristive behavior. We will illustrate how a V1 visual cortex layer can be assembled and how it is capable of learning to extract orientations from visual data coming from a real artificial CMOS spiking retina observing real life scenes. Finally, we will discuss limitations of currently available memristors. The results presented are based on behavioral simulations and do not take into account non-idealities of devices and interconnects. The aim here is to present, in a tutorial manner, an initial framework for the possible development

T. Serrano-Gotarredona (✉) · B. Linares-Barranco
Instituto de Microelectronica de Sevilla, Consejo Superior de Investigaciones Cientificas,
Universidad de Sevilla, Avda. Americo Vespucio s/n, 41092 Sevilla, Spain
e-mail: terese@imse-cnm.csic.es

T. Masquelier
Laboratory of Neurobiology and Adaptive Processses, University Pierre et Marie Curie,
9 quai St. Bernard, 75005 Paris, France

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_11,

of fully asynchronous STDP learning neuromorphic architectures exploiting two or three terminal memristive type devices. (A Supplemental Material compressed zip file containing all files used for the simulations can be downloaded from http://www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2011.00026/abstract.)

1 Introduction

Neuromorphic engineering [3] is a new interdisciplinary discipline that takes inspiration from biology, physics, mathematics, computer science and engineering to design artificial neural systems, such as vision systems, head-eye systems, auditory processors, and autonomous robots, the physical architecture and design principles of which are based on those of biological nervous systems. The term neuromorphic was coined by Carver Mead, in the late 1980s [4] to describe very-large-scale integration (VLSI) systems containing electronic analog circuits that mimic neurobiological architectures present in the nervous system. In recent times the term neuromorphic has been used to describe both analog, digital or mixed-mode analog/digital VLSI systems that implement models of neural systems (for perception, motor control, or sensory processing) and also software algorithms. A key aspect of neuromorphic design is understanding how the morphology of individual neurons, circuits, and overall architectures creates desirable computations, affects how information is represented, influences robustness to damage, incorporates learning and development, and facilitates evolutionary change.

It is obvious that interdisciplinary research broadens our view of particular problems yielding fresh and possibly unexpected insights. This is the case of neuromorphic engineering, where technology and neuroscience cross-fertilize each other. One example of this is the recent impact of fabricated memristor devices [5–8], postulated since 1971 [9–11], thanks to research in nanotechnology electronics. Another is the mechanism known as Spike-Time-Dependent-Plasticity (STDP) [12–26] which describes a neuronal synaptic learning mechanism that refines the traditional Hebbian synaptic plasticity proposed in 1949 [27]. These are very different subjects from relatively unrelated disciplines (nanotechnology, biology, and computer science), which have nevertheless recently been drawn together by researchers in neuromorphic engineering [1, 2, 28–30]. STDP was originally postulated as a family of computer learning algorithms [12], and is being used by the machine intelligence and computational neuroscience community [17–26]. At the same time its biological and physiological foundations have been reasonably well established during the past decade [31–38]. If memristance and STDP can be related, then (a) recent discoveries in nanophysics and nanoelectronic principles may shed new light on the intricate molecular and physiological mechanisms behind STDP in neuroscience, and (b) new neuromorphic-like computers built out of nanotechnology memristive devices could incorporate biological STDP mechanisms, yielding a new generation of self-adaptive ultra-high-dense intelligent machines. Here we explain how by combining memristance models with the electrical wave signals of neural impulses (spikes) converging from pre- and post-synaptic neurons into a synaptic junction, STDP behavior emerges naturally [30]. This helps us to understand how neural and memristance parameters modulate STDP, and may offer new insights to neurophysiologists searching for the ultimate physiological mechanisms responsible for STDP in biological synapses. At the same time, it also provides a direct means of incorporating STDP learning mechanisms into a new generation of nanotechnology computers employing memristors. Here we focus on this second aspect.

2 STDP

Spike-time-dependent plasticity (STDP) is a family of learning mechanisms originally postulated in the context of artificial machine learning algorithms (or computational neuroscience), exploiting spike-based computations (as in brains) with great emphasis on the relative timings of spikes. Gerstner started to report the first spike timing dependent learning algorithms [12, 13] in 1993. STDP has been shown to be better than Hebbian correlation-based plasticity at explaining cortical phenomena[25, 26], and has been proven successful in learning hidden spiking patterns [22] or performing competitive spike pattern learning [23]. Astonishingly, experimental evidence of STDP has been reported by neuroscience groups during the past decade[31–38], so today we can state that the physiological existence of STDP has been reasonably well established.[1]

However, the full implications of the molecular and electro-chemical principles behind STDP are still under debate [39]. Before describing STDP mathematically, let us first explain how neurons interchange information and what the synaptic connections are.

Figure 1 illustrates two neurons connected by a synapse. The pre-synaptic neuron is sending a pre-synaptic spike $V_{mem\text{-}pre}(t)$ through one of its axons to the synaptic junction. Neural spikes are membrane voltages from the outside of the cellular membrane V_{pre^+} with respect to the inside V_{pre^-}. Thus $V_{mem\text{-}pre} = V_{pre^+} - V_{pre^-}$ and $V_{mem\text{-}pos} = V_{pos^+} - V_{pos^-}$. The "large" membrane voltages during a spike (in the order of a hundred mV) cause a variety of selective molecular membrane channels to open and close allowing many ionic and molecular substances to flow, or preventing them from flowing through the membrane. At the same time, synaptic vesicles inside the pre-synaptic cell containing "packages" of neurotransmitters fuse with the membrane in such a way that these "packages" are released into the synaptic cleft (the inter cellular space between both neurons at the synaptic junction). Neurotransmitters are collected in part by the post-synaptic membrane, contributing to a change in its membrane conductivity. The cumulative effect of pre-synaptic spikes (coming from this or other pre-synaptic neurons) will eventually trigger the generation of a new spike at the post-synaptic neuron. Each synapse is characterized by a "synaptic strength" (or weight) w which determines the efficacy of a pre-synaptic

[1] For a historical overview on how STDP research evolved independently among computational and experimentalist groups, please refer to the last paragraph in [14].

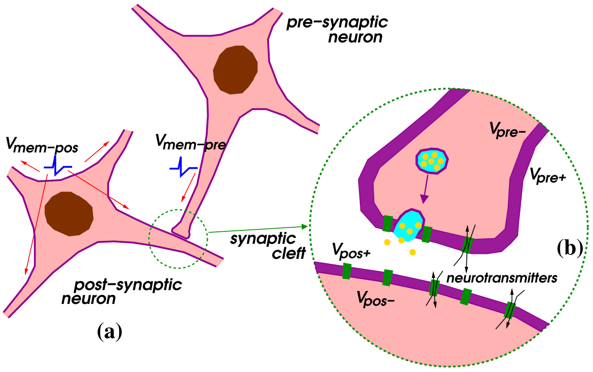
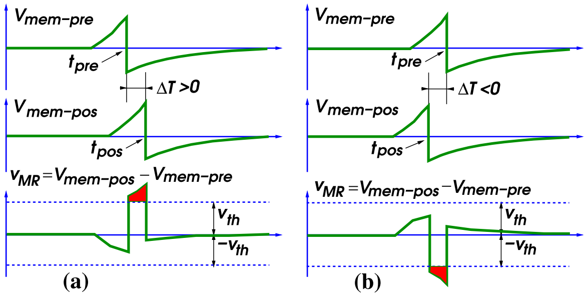


Fig. 1 Illustration of synaptic action. (**a**) A synapse is where a pre-synaptic neuron "connects" with a post-synaptic neuron. The pre-synaptic neuron sends an action potential $V_{mem\text{-}pre}$ traveling through one of its axons to the synapse. The cumulative effect of many pre-synaptic action potentials, generates a post-synaptic action potential at the membrane of the post-synaptic neuron, which propagates through all the neuron's terminations. (**b**) Detail of synaptic junction. The cell membrane has many membrane channels of varying nature which open and close with changes in the membrane voltage. During a pre-synaptic action potential vesicles containing neurotransmitters are released into the synaptic cleft

Fig. 2 Membrane voltage waveforms. Pre- and post-synaptic membrane voltages for the situations of positive ΔT (**a**) and negative ΔT (**b**). Voltage v_{MR} is the difference between the post-synaptic membrane voltage $V_{mem\text{-}pos}$ and the pre-synaptic membrane voltage $V_{mem\text{-}pre}$



spike in contributing to this cumulative action at the post-synaptic neuron. This weight w could well be interpreted as the size and/or number of neurotransmitter packages released during a pre-synaptic spike. However, for our analyses, we will interpret w more generally as some kind of structural parameter of the synapse (like the amount of one or more metabolic substances) that directly controls the efficacy of this synapse per spike. The synaptic weight w is considered to be non-volatile and analog in nature, but it changes in time as a function of the spiking activity of pre- and post-synaptic neurons. This phenomenon was originally observed and reported in 1949 by Hebb, who introduced his Hebbian learning postulate [27]: "*When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased*". Traditionally, this has been described by computational neuroscientists and machine learning computer engineers as producing an increment in synaptic weight Δw proportional

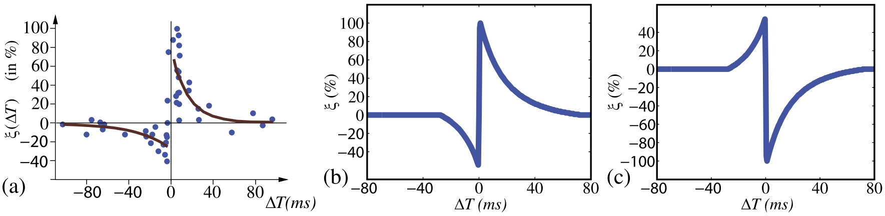


Fig. 3 (**a**) Experimentally measured STDP function $\xi(\Delta T)$ on biological synapses (data from Bi and Poo [32, 33]). (**b**) Ideal STDP update function used in computational models of STDP synaptic learning. (**c**) Anti-STDP learning function for inhibitory STDP synapses

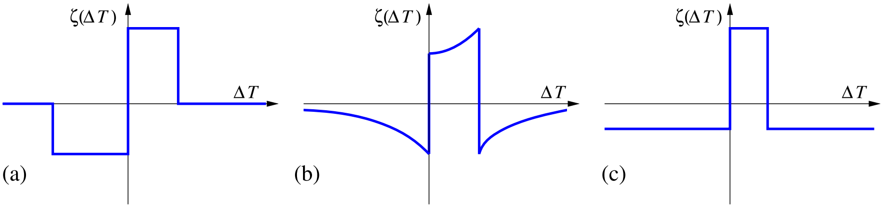


Fig. 4 Other STDP functions reported in literature, such as (**a**) square boxes simplification [18, 19, 46], (**b**) central-only potentiation [44], (**c**) square boxes simplification of central-only potentiation [41–44]

to the product of the mean firing rates of pre- and post-synaptic neurons. STDP is a refinement of this 1949 rule which takes into account the precise relative timing of individual pre- and post-synaptic spikes, and not their average rates over time. In STDP the change in synaptic weight Δw is expressed as a function of the time difference between the post-synaptic spike at t_{pos} and the pre-synaptic spike at t_{pre} (see Fig. 2). Specifically, as is shown in Fig. 3, $\Delta w = \xi(\Delta T)$, with $\Delta T = t_{pos} - t_{pre}$. The shape of the STDP function ξ can be interpolated from experimental data from Bi and Poo as shown in Fig. 3(a) [33]. For positive ΔT (that is to say, the pre-synaptic spike has a highly relevant role in producing the post-synaptic spike) there will be a potentiation of synaptic weight $\Delta w > 0$, which will be stronger as $|\Delta T|$ reduces. For negative ΔT (that is to say, the pre-synaptic spike is highly irrelevant for the generation of the post-synaptic spike), there will be a depression of synaptic weight $\Delta w < 0$, which will be stronger as $|\Delta T|$ reduces. Bi and Poo concluded that they had observed an asymmetric critical window for ΔT of about ±40–80 ms for synaptic modification to take place. Mathematically, this $\xi(\Delta T)$ STDP learning function is described by computational neuroscientists as

$$\xi(\Delta T) = \begin{cases} a^+ e^{-\Delta T/\tau^+} & \text{if } \Delta T > 0 \\ -a^- e^{\Delta T/\tau^-} & \text{if } \Delta T < 0 \end{cases} \tag{1}$$

Other STDP functions, like those shown in Fig. 4, have been used by computational neuroscientists, as well as observed in biological experimental [45].

2.1 STDP Versus Anti-STDP

The STDP learning functions $\xi(\Delta T)$ as defined in Figs. 3(a), (b) and 4 are useful for synapses with positive weights. In these cases, weight w is strengthened if it is increased ($\Delta w > 0$) when $\Delta T > 0$, and vice versa. However, if the weight is negative ($w < 0$), as in some inhibitory synapse implementations, the STDP learning functions in Figs. 3(b) and 4 are not appropriate because an increase in weight ($\Delta w > 0$) would weaken the strength of the synapse, and vice versa. For negative weight synapses an STDP learning function with a shape similar to that shown in Fig. 3(c) [40] is required. In this case, the synapse is strengthened by decreasing its weight ($\Delta w < 0$), which should happen for $\Delta T > 0$. Let us call this an Anti-STDP synaptic update or learning function. Other more exotic shapes for $\xi(\Delta T)$ are also possible, as we will discuss later in Sects. 4.1 and 5.1.

2.2 Additive Versus Multiplicative STDP

Most of the present day literature on STDP presents a learning function ξ which depends on ΔT but not on the actual weight value w. This type of weight-independent STDP learning rule is usually known as "additive STDP". Additive STDP requires the weight values to be bounded to an interval because weights will stabilize at one of their boundary values [46, 47].

On the other hand, in multiplicative STDP (mSTDP) [46–48] the learning function is also a function of the actual weight value $\xi_m(w, \Delta T)$. Furthermore, there usually appears a weight dependent factor which multiplies the original additive STDP learning function ξ_a, and which may generally be different for the positive ($\Delta T > 0$) and negative ($\Delta T < 0$) sides

$$\xi_m(w, \Delta T) = F\big(w, sign(\Delta T)\big)\xi_a(\Delta T) \tag{2}$$

In mSTDP weights can stabilize to intermediate values inside the boundary definitions. Thus, it is often not even necessary to enforce boundary conditions for the weight values [46]. As we will see later in Sects. 4.2 and 4.3, present day memristors can result in either additive or quadratic (multiplicative) STDP depending on whether their operation is better described by a "moving wall" model or a "filament creation annihilation" model.

3 Memristance

Memristance was postulated in 1971 by Chua [9] based on circuit theoretical reasonings. According to circuit theoretical fundamentals, there are four basic electrical quantities [11]: (1) voltage difference between two terminals "v", (2) current flowing through into a device terminal "i", (3) charge flowing through a device terminal

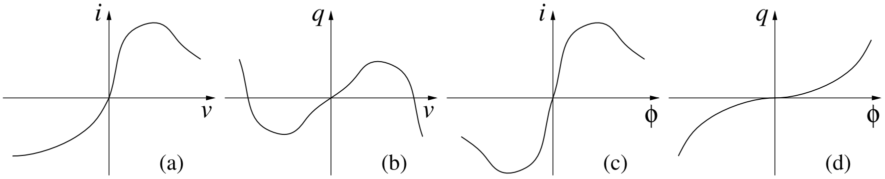


Fig. 5 Description of the four canonical two-terminal devices. (**a**) A resistor is defined by a static relationship between a device's voltage and current. (**b**) A capacitor is defined by a static relationship between a device's charge and voltage. (**c**) An inductor is defined by a static relationship between a device's current and flux. (**d**) And a memristor is defined by a static relationship between a device's charge and flux

or integral of current $q = \int i(\tau)d\tau$, and (4) flux or integral of voltage $\phi = \int v(\tau)d\tau$. A two-terminal device is said to be canonical [11] if either two of the four basic electrical quantities are related by a static[2] relationship, as shown in Fig. 5. A resistor has a static relationship between terminal voltage v and device current i, as shown in Fig. 5(a). A capacitor shows a static relationship between charge q and voltage v, as shown in Fig. 5(b). An inductor has a static relationship between its current i and flux ϕ, as shown in Fig. 5(c). These three devices have been very well known since the origins of Electronics and Electricity. However, there are other possibilities for combining the four basic electrical quantities: (q, i), (v, ϕ), and (q, ϕ). Ignoring the combinations of a quantity with its time derivative leaves us with one single additional possibility: (q, ϕ). This reasoning led Chua to postulate the existence of a fourth basic two-terminal element, which he called the *Memristor*. The memristor would show a static relationship between charge q and flux ϕ, as shown in Fig. 5(d). If the q vs. ϕ relationship is linear, the *memristor* degenerates into a linear resistor. *Memristors* behave as resistances in which the resistance changes through some of the basic electrical quantities, and is somehow *memorized*. The simple concept of *memristance* as defined in Fig. 5(d) can be extended to refer to any device exhibiting resistive behavior whose resistance can change through some of the four basic electrical quantities, but at the same time exhibiting *memory* for that resistance. In that case, more elaborate mathematical descriptions are required [10].

Memristance has recently been demonstrated (with extraordinary impact among the research community) in nanoscale two-terminal devices, such as certain titanium-dioxide [5, 6] and amorphous Silicon [8] cross-point switches. However, memristive devices were reported earlier by other groups [49, 50]. Memristance arises naturally in nanoscale devices because small voltages can yield enormous electric fields that produce the motion of charged atomic or molecular species, changing structural properties of a device (such as its doping profile) while it operates. Memristors are asymmetric two-terminal passive devices. Consequently, their

[2] By 'static' we mean it is not altered by changes of the above electrical quantities, or by their history, integrals, derivatives, etc. These 'static' curves can, however, be time-varying if the change is caused by an external agent. For example, a motor driven potentiometer would have a 'static' i/v curve that is time varying.

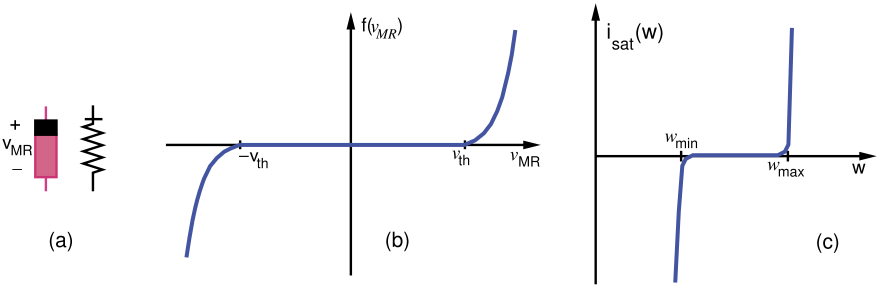


Fig. 6 (**a**) Memristor asymmetric symbols. (**b**) Memristor nonlinear weight update function with exponential growth and thresholding. (**c**) Memristor saturation function for limiting range of weight

circuit symbol must indicate somehow their polarity. Figure 6(a) shows two possible symbols. Here we will consider one particular subset of memristors described by [28, 29]

$$i_{MR} = g(w, v_{MR})v_{MR} \tag{3}$$

$$\dot{w} = f(w, v_{MR}) \tag{4}$$

where w is some physical (structural) parameter, i_{MR} is the current through the device, v_{MR} the voltage drop across it, and g is its (nonlinear) conductance. Since the change of structural parameter w is driven by voltage v_{MR}, we say this memristor is voltage (or flux) driven. The group at the Michigan University claims to have fabricated a memristor of this kind [7, 8]. If function $f()$ in Eq. (4) is driven by memristor current i_{MR}, then we say the memristor is current (or charge) driven. The HP group tends to model their memristor as one of this type [5, 6].

In memristive nanoscale devices, function f may describe ionic drift under electric fields. Although this may conceivably be modeled by a linear dependence of f with voltage v_{MR} [5], it is clear that in reality such dependence is more likely to grow exponentially and/or include a threshold barrier v_{th} [7]. For our discussions, let us assume the following dependence

$$f(v_{MR}) = I_o \operatorname{sign}(v_{MR})\left[e^{|v_{MR}|/v_o} - e^{v_{th}/v_o}\right] \quad \text{if } |v_{MR}| > v_{th} \tag{5}$$

$$f(v_{MR}) = 0 \quad \text{otherwise} \tag{6}$$

where I_o and v_o are parameters which may or may not depend on w. This shape of f is shown in Fig. 6(b). Many other mathematical formulations can be used [10]. In order to relate memristance to biological STDP, as will be done in Sect. 4, we need a voltage/flux controlled memristor with thresholding behavior, exponential behavior beyond threshold, and bidirectional behavior (i.e. to be able to increment and decrement w). Since a memristor has polarity, as indicated in Fig. 6(a), we need to establish a criterion to assign one of the terminals as the positive terminal. The criterion adopted to assign polarity is that if a sufficiently large *positive* voltage v_{MR} is applied to the memristor (i.e. larger than its positive threshold), it will increase its

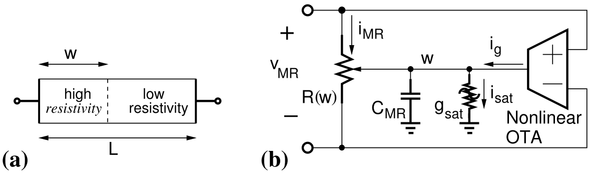


Fig. 7 (**a**) Memristor "moving wall" model. (**b**) Simple macro model circuit for electrical simulations

conductance. Otherwise, if a sufficiently large *negative* voltage v_{MR} is applied (i.e. increasing beyond its negative threshold), it will decrease its conductance.

3.1 Memristor Moving-Wall Macro Model for Two-Terminal Devices

A macro model of a device is a behavioral model made of circuit elements (ideal or not) that describes its behavior. Some circuit simulators allow a device to be defined mathematically using for example AHDL or Verilog-A circuit description languages. However, if the device can be described with a macro model circuit, this will have some advantages.[3] (1) First, it uses already built-in components providing faster simulations; (2) second, as it is made of circuit elements it gives (analog) circuit designers a richer intuitive insight into how it works and performs, and how to improve it for specific goals; (3) it is very intuitive when adding parasitic components (resistors and capacitors) to aid in the convergence of the simulator's internal algorithms; (4) and if care is taken to keep the operating voltages and currents of internal nodes to the levels the simulator expects from conventional circuits, simulations converge easier and faster.

Some reported memristors (as in [53–56]) adhere better to the "moving wall model" (see Figs. 7(a) and 9(b)), where the wall position w separates two different resistive regions in series, and moves depending on device current or voltage. A circuit macro model that implements Eqs. (3)–(6) is shown in Fig. 7(b). It comprises a controlled resistor in which resistance R is controlled linearly by internal state voltage w, $R(w) = k_R \times (w + w_o)$. Voltage w represents the "structural" parameter of the wall position, which is bounded to $[0, L]$ [51, 52]. Component NOTA is a nonlinear differential-input voltage-controlled current-source (transconductor), also known as nonlinear OTA (Operational Transconductance Amplifier) which provides an output current $i_g(v_{MR}) = f(v_{MR})$ controlled by input differential voltage v_{MR}, as given in Eqs. (5)–(6) and Fig. 6(b). Nonlinear element g_{sat} is a nonlinear resistor

[3] Note that our aim in providing a macro model circuit is to have a means of simulating large number of memristors efficiently in a circuit simulator, and hence take advantage of its computational efficiency and ease of use. Our aim is not to provide a means of building physical circuits out of such macro models (as Chua did in the past using mutator circuits [9]). A direct physical circuit realization of Fig. 7(b) would result, for example, in leaks of the memory value w due to unavoidable current leak paths in parallel with the capacitor.

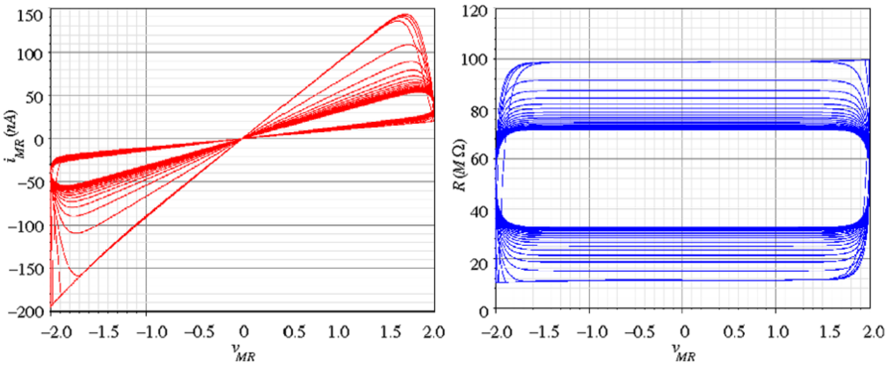


Fig. 8 Memristor simulations using the macro model. *Left*, i/v curves obtained. *Right*, dependence of memristor time varying resistance with memristor voltage

with a shape as shown in Fig. 6(c), which limits the range of the resistance $R(w)$ to $[R_{min}, R_{max}]$, thus keeping w inside its natural boundary $[0, L]$. Consequently, the macro model circuit in Fig. 7 is mathematically described by

$$v_{MR} = R(w)i_{MR} \tag{7}$$

$$R(w) = k_R \times (w + w_o) \tag{8}$$

$$C_{MR}\dot{w} = i_g(v_{MR}) - i_{sat}(w) \tag{9}$$

Parameter k_R scales between the voltage domain range of w (usually within a few volts, for proper simulator convergence) and the resistance domain range of R which can be as high as hundreds of Mega-ohms [7, 8]. Figure 8 shows the simulation results of a memristor connected in series with a 5 MΩ resistance stimulated with a 2 V sinusoid of decreasing frequency from 5 KHz to 0 Hz in 26 cycles. Maximum and minimum memristor resistance limits were $R_{max} = 100$ MΩ and $R_{min} = 10$ MΩ, symmetric threshold voltages were $|v_{th}| = 1$ V, the exponential $f(v_{MR})$ was characterized by $I_o = 10$ μA, $v_o = 0.1$ V, $v_{th} = 1$ V, and the other macromodel parameters were $w_{max} = 10$ V, $w_{min} = -10$ V, $k_R^{-1} = 222$ μA, $w_o = 12.2$ V, and $C_{MR} = 10$ mF.

3.2 Memristor Filament Model for Two-Terminal Devices

The previous *moving wall model* can approximate phenomena like migration of oxygen ions [53] and vacancies [54], the lowering of Schottky barrier heights by trapped charge carriers at interfacial states [55], and the phase-change in some PCM (phase change materials) devices [56].

However, resistive switching effects in dielectric-based devices have normally been assumed to be caused by conducting filament formation across the electrodes,

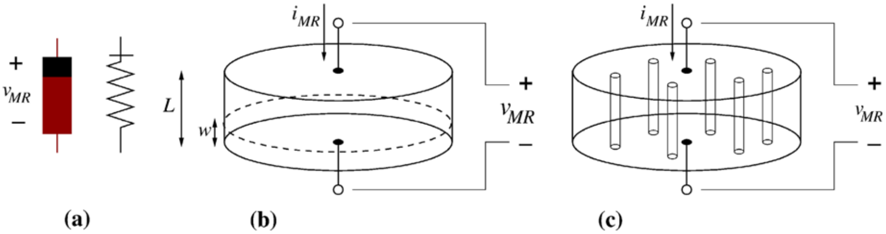


Fig. 9 (**a**) Memristor asymmetric symbols. (**b**) Moving wall model memristor operating as two variable resistors in series. (**c**) Illustration of filament formation/annihilation model describing memristor operation as two variable resistances in parallel

although the understanding and modeling of these phenomena remains controversial. As a matter of fact, some researchers are observing the formation and annihilation of nanoscale width conducting filaments in memristors [57, 58]. Precise modeling of this phenomenon is still under research [59]. However, let us here propose the following very simplified view to approximate this physical mechanism. Figure 9(c) illustrates schematically a memristor with several conducting filaments between the two electrodes. The number of filaments or their cross-sectional area would increase or decrease with memristor operation. Let us call now w the total cross sectional area of the effective conducting filaments at a given instant in time, and S the total cross section area of the memristor. The filaments present high conductivity (low resistivity), while the bulk presents much lower conductivity (high resistivity). All formed parallel filaments behave as one effective resistance of low resistance, while the rest of the bulk behaves as another higher resistivity resistor. Therefore, now the memristor behaves as two variable resistors in parallel. Consequently, its total conductance (inverse of resistance) could be described as

$$G = G_{ON}\frac{w}{S} + G_{OFF}\left(1 - \frac{w}{S}\right) \tag{10}$$

where G_{ON} is the conductance per effective cross section area of the filaments, and G_{OFF} is the conductance per effective cross section area of the filament-less bulk material. Parameter w would change from 0 to w_{max}, the maximum possible effective cross section area of total conducting filaments ($w_{max} \leq S$).

This changing cross section description not only approximates filament formation/annihilation phenomena, but also some other gradual cross section area variations observed in some phase-change or ferroelectric-domains-based materials [60].

As we will highlight later in Sect. 4, whether a memristor is better described by the moving wall model or the filament formation/annihilation model, impacts severely on the resulting type of STDP learning mechanism. The latter yields an additive type of STDP, while the former results in a quadratic type STDP. Note that a memristor can be either voltage/flux or current/charge driven, independently of whether it is a “wall” or a “filament” memristor.

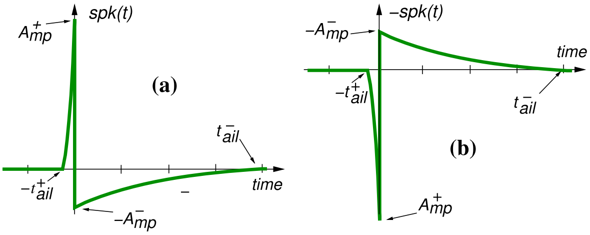


Fig. 10 (**a**) Details of membrane voltage action potential, as by Eq. (11). (**b**) Reversed version

4 Relation Between STDP and Memristance

How can STDP be related to memristance? The key is to consider carefully the shape of the electric neural spikes [30]. The exact shape of neural spikes, usually called "*action potentials*" among neuroscientists, is difficult to measure precisely since the experimental setup influences strongly. Furthermore, different action potential shapes have been recorded for different types of neurons, although in general they all display a certain resemblance. For our discussion it suffices to assume a generic action potential shape with the following properties (see Fig. 10(a)). During spike on-set, which happens during a time t^+_{ail}, membrane voltage increases exponentially to a positive peak amplitude A^+_{mp}. After this, it changes quickly to a peak negative amplitude $-A^-_{mp}$ and returns smoothly to its resting potential during a time t^-_{ail}. A shape of the type shown in Fig. 10(a) can be expressed mathematically, for example, as

$$spk(t) = \begin{cases} A^+_{mp} \frac{e^{t/\tau^+_{ail}} - e^{-t^+_{ail}/\tau^+_{ail}}}{1-e^{-t^+_{ail}/\tau^+_{ail}}} & \text{if } -t^+_{ail} < t < 0 \\ -A^-_{mp} \frac{e^{-t/\tau^-_{ail}} - e^{-t^-_{ail}/\tau^-_{ail}}}{1-e^{-t^-_{ail}/\tau^-_{ail}}} & \text{if } 0 < t < t^-_{ail} \\ 0 & \times\text{otherwise} \end{cases} \tag{11}$$

Parameters τ^+_{ail} and τ^-_{ail} control the curvature of the on-set and off-set sides of the action potential.

Consider the case of pre- and post-synaptic neurons in Fig. 1 being of the same type, and thus generating the same action potential shape, $spk(t)$ of Eq. (11), when they fire. Axons and dendrites operate as transmission lines, so it is reasonable to expect some attenuation when the spikes arrive at the respective synapses. Let α_{pre} be the attenuation for the pre-synaptic spike $V_{mem\text{-}pre}(t) = \alpha_{pre}\, spk(t - t_{pre})$, and α_{pos} for the post-synaptic spike $V_{mem\text{-}pos}(t) = \alpha_{pos}\, spk(t - t_{pos})$. When both spikes are simultaneously present at the two cell membranes of the synapse, then channels on both membranes are open. Consequently, in principle, it makes sense to assume that during such time there could be a path for substances in the inside of one cell to move directly to the inside of the other cell and vice versa. Furthermore, let us assume now that such motion of substances obeys a memristive law similar to those described by Eqs. (3)–(6). This means, that we would have a two-terminal memristive device between the inner sides of the two cells; more specifically, between V_{pos-} and V_{pre-} in Fig. 1(b). Consequently, the memristor voltage would be

$v_{MR} = V_{pre^-} - V_{pos^-}$. On the other hand, since the outside nodes of both membranes V_{pos^+} and V_{pre^+} are very close together, both voltages will be approximately equal, yielding

$$v_{MR}\big(t'\big) \approx V_{mem\text{-}pos}\big(t'\big) - V_{mem\text{-}pre}\big(t'\big) = \alpha_{pos}\, spk\big(t' - t_{pos}\big) - \alpha_{pre}\, spk\big(t' - t_{pre}\big) \tag{12}$$

A simple change of variables $t = t' - t_{pos}$ and recalling that $\Delta T = t_{pos} - t_{pre}$, results in

$$v_{MR}(t, \Delta t) = \alpha_{pos}\, spk(t) - \alpha_{pre}\, spk(t + \Delta t) \tag{13}$$

This memristor voltage v_{MR} is shown in Fig. 2 for the cases of ΔT being positive or negative. According to Eqs. (5)–(6), memristive update will take place only if v_{MR} exceeds threshold v_{th}, as indicated by the red shaded area in Fig. 2. As we postulated earlier, during this memristive update some amount of synaptic structural substance(s) Δw would be interchanged between the two sides of the synapse. The amount of substance Δw will ultimately affect the synaptic strength of this synapse. If this amount of synaptic structural substance interchanged between the two synaptic terminations obeys a memristive law as in Eqs. (3)–(6), then from Eq. (4)

$$\Delta w(\Delta T) = \int f\big(v_{MR}(t, \Delta T)\big)dt = \xi(\Delta T) \tag{14}$$

which is the red area of the shaded regions in Fig. 2, previously amplified exponentially through function $f()$ of Eqs. (5)–(6). Positive areas (above v_{th}, when $\Delta T > 0$) yield increments for w ($\Delta w > 0$), while negative areas (below $-v_{th}$, when $\Delta T < 0$) result in decrements for w ($\Delta w < 0$). As $|\Delta T|$ approaches zero, the peak of the red area in v_{MR} is higher. Since this peak is amplified exponentially, the contribution for incrementing/decrementing w will be more pronounced as $|\Delta T|$ is reduced. The resulting function $\Delta w(\Delta T)$, computed using the memristor model through Eq. (14) is shown in Fig. 3(b). It imitates the behavior of the STDP function ξ obtained by Bi and Poo from physiological experiments, which is shown in Fig. 3(a). For this numerical computation we used the following parameters: $\alpha_{pos} = 1$, $\alpha_{pre} = 0.9$, $v_{th} = A_{mp+} = 1$, $A_{mp-} = 0.25$, $v_o = 1/7$, $t_{ail+} = 5$ ms, $t_{ail-} = 75$ ms, $\tau^+ = 40$ ms, $\tau^- = 3$ ms. Making $\alpha_{pos} \neq \alpha_{pre}$ breaks the symmetry of function $\xi(\Delta T)$, and making them very different removes one of the branches in $\xi(\Delta T)$. It is possible to have more freedom to achieve a desired shape for $\xi(\Delta T)$ by setting $\alpha_{pos} = \alpha_{pre} = 1$, but instead shaping the spikes traveling forwards and backwards independently. Also, note that for ΔT values very close to zero $\Delta w(\Delta T)$ is approximately linear and crosses the origin. This is because as ΔT approaches zero, v_{MR} approaches zero for any t (see Eq. (13)).

This result shows that a memristive type of mechanism could be behind the biological STDP phenomenon.

4.1 Influence of Action Potential Shape

The shape of the action potential function $spk(t)$ strongly influences the shape of the resulting STDP function $\xi(\Delta T)$. As an illustration, Fig. 11 shows how for several shapes of action potentials ("spikes") different STDP learning functions $\xi(\Delta T)$ are obtained. For example, if the exponential shape (a1) degenerates into a triangular type of shape (c1), then the central region of $\xi(\Delta T)$ will display a smoother transition from the negative peak to the positive peak (c2). Note that this would weaken learning for cases with small $|\Delta T|$. Making the positive peak of the spike smaller than the negative peak, makes the negative branch for $\xi(\Delta T)$ stronger than the positive branch. If the action potential is substituted by a rectangular shape signal (d1), then the central region becomes linear and a saturation effect might occur (d2). If the rectangular spike is made more symmetric, then $\xi(\Delta T)$ degenerates into a triangular type of shape, which is very different from the original biological STDP learning function. In general, to obtain a biological-like STDP learning function a narrow short positive pulse of large amplitude and a longer relaxing slowly decreasing negative tail are required (a1, b1, c1, d1). However, from a computational point of view, it might be more interesting to massage the shape of the "spikes" and tune the STDP learning function as desired.

Figure 11(e1) shows a symmetric spike which results in a symmetric STDP learning function (e2). Inverting the forward spike in Fig. 11(f1) with respect to the backward one also results in a symmetric STDP learning function (see Fig. 11(f2)). By using different shapes for the forward and backward spikes, as in Fig. 11(g1), one can obtain the STDP learning function (g2), as in Fig. 4(c). If, instead, we use the shapes in Fig. 11(h1), we obtain Fig. 11(h2), which is the time limited version of Fig. 11(g2).

4.2 Wall Model Memristors Implement a Multiplicative Type of STDP

Memristors described by a "moving wall model" implement a multiplicative type of STDP. The reason is because, according to Eq. (14), the structural parameter updates $\Delta w(\Delta T)$ follow an additive type of STDP rule, independent of w. Parameter w is the memristor "wall" position separating the low and high resistivity regions (see Fig. 7(a)). According to Eq. (8), the memristor instantaneous resistance $R(w)$ is linear with w. Consequently, the memristive STDP resistance update $\Delta R(\Delta T) = k_R \times \Delta w(\Delta T) = k_R \xi(\Delta T)$ will follow an additive STDP update rule as well, independent of the actual value of R. However, as we will see in the next Section, when memristors (or resistors) are used as synapses in a neural circuit, the synaptic strength of such synapses is proportional to their conductance $G = 1/R$, because as conductance increases more current will be delivered to the post-synaptic

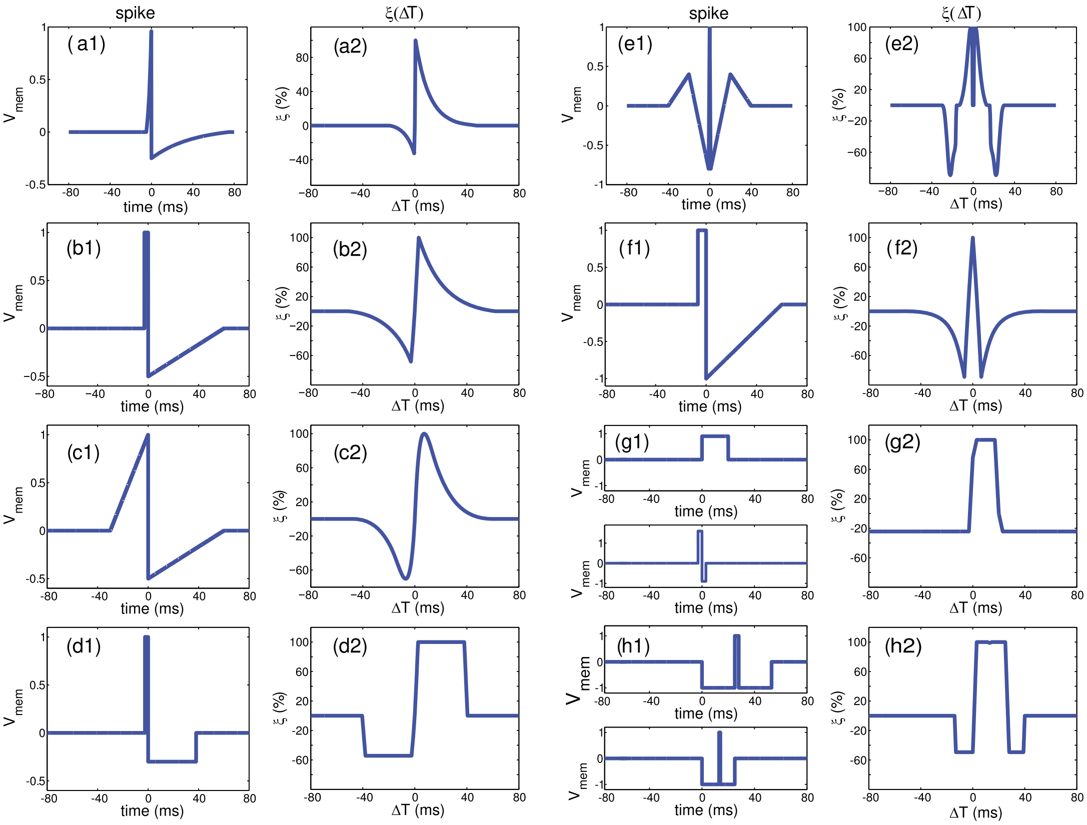


Fig. 11 Illustration of influence of action potential shape on the resulting STDP resistance update function $\xi(\Delta T)$

neuron. Consequently, synaptic strength update is given by

$$\Delta G(\Delta T) = -\frac{\Delta R(\Delta T)}{R^2} = -G^2 k_R \xi(\Delta T) \tag{15}$$

which is quadratically proportional to the actual conductance. Such memristors will therefore yield larger update steps for high conductances, but smaller steps for low conductances. This suggests that, before training, weights (conductances) should be initialized to rather high values, so that as learning progresses the updates tend to become smaller.

Note that the fact that memristors implement a multiplicative type of STDP derives from the fact that the wall position w is linear with resistance, and thus inversely proportional to synaptic strength. Consequently, we should expect mSTDP also from synchronous STDP realizations, using either current or voltage driven memristors, as the update would also be weight dependent. On the other hand, we have assumed that function $f()$ in Eqs. (5), (6), (14) does not depend on w. In practice, there might exist such dependence, and if true, the resulting mSTDP learning function might deviate from the one discussed here.

4.3 Filament Model Memristors Implement Additive STDP

When the memristor physics is better represented by the inter-electrode filament formation/annihilation model, then synaptic update would change parameter w of Eq. (10), which is now directly proportional to memristor conductance (synaptic strength),

$$\Delta G(\Delta T) = \frac{G_{ON}}{S} \Delta w(\Delta T) = \gamma \xi(\Delta T) \tag{16}$$

where γ is a constant. Therefore, synaptic update would be independent of actual weight (conductance) and the resulting STDP update rule is said to be of additive type. Note that Eqs. (11)–(14) and the resulting functions $\xi(\Delta T)$ in Fig. 11 are common for both "wall" and "filament" models.

5 Connecting Memristors with Spiking Neurons for Asynchronous STDP Learning

Synchronous memristive STDP learning architectures were proposed by Snider [28, 29], assuming voltage/flux driven memristors, and recently demonstrated by the group at Michigan University [7]. In that proposal each neural spike is mapped into a sequence of precisely spaced fixed amplitude digital pulses which must maintain global synchronization to separate the integration phase of neural activity from the synaptic weight update phase. This global synchronization requirement imposes severe difficulties when the system scales up to very large sizes.

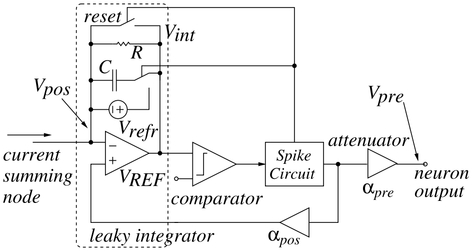


Fig. 12 Proposal of LIF neuron circuit implementation for memristance compatible STDP fully-asynchronous learning system

On the other hand, from the discussions in previous Sections, we present an alternative approach which is fully asynchronous [30]. Consequently, no global synchronizations will be required nor global separations into neuron-integration phases and synapse-learning phases, making this approach attractive for scaling up to very large numbers of neurons and synapses.

We first need a neural circuit that integrates spikes until a threshold is reached. At that moment, it should provide a spike of the desired shape. A possible schematic diagram for a leaky integrate-and-fire (LIF) neuron block is shown in Fig. 12. The neurons need to include a current summing and sinking input terminal so that in the absence of spike output the integral of input current spike signals can be computed, while maintaining the input node tied to a fixed voltage. This can be done by using a lossy integrator with a clamped voltage input. The output of this accumulated integral V_{int} is compared against a reference V_{REF}. If this reference is reached, the comparator output will trigger a spike generation circuit, which provides the output spike of the neuron. During spike generation, the integration capacitor is charged to refractory voltage V_{refr}, while the input opamp is configured as a voltage buffer, thus copying the spike waveform to the neuron input node. An attenuated version of the post-synaptic neural voltage $\alpha_{pos} V_{pos}(t)$ is thus made available to the synaptic memristors connected to this neuron input. Another attenuated version of the spike is fed forward to the output of the neuron $V_{pre}(t) = \alpha_{pre}\, spk(t)$. During the whole time of the spike (typically in the order of 20–100 ms) the neuron is not integrating (computationally inactive). This time is also called "*refractory time*". During the absence of spike output, the spike generation circuit provides a constant voltage V_{rest}.

For the *Spike Circuit* in Fig. 12 an analog circuit can be devised that generates a specific action potential shape with some tunable parameters [61]. However, for STDP experiments it is more desirable to allow for full programmability of arbitrarily shaped action potentials. Since all neurons should have the same spike shape (at least all neurons of a part of a whole system), one interesting option is to have a circuit at the chip periphery broadcasting digital samples of the action potentials at different phases to all neurons. The spike generation circuits would then capture the closest phase, delay it properly, and through a local, compact digital-to-analog converter provide the programmed action potential.

For the synapses, it is possible to fabricate very high density memristor crossbar structures [5, 6] which connect to the neural layers, as shown in Fig. 13(a). Neurons generate action potentials with a shape similar to those given in Fig. 10. Note that

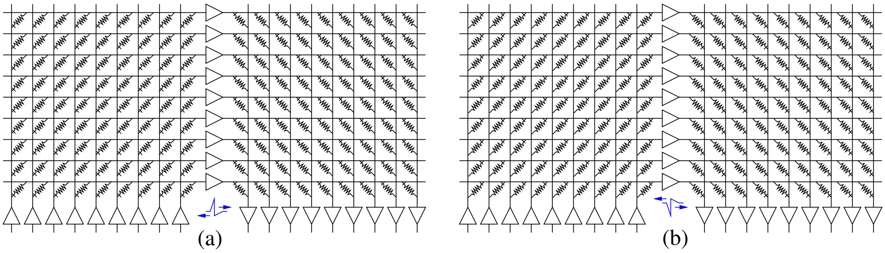


Fig. 13 Two possible interconnection schemes between memristors and neurons for STDP learning

the positive terminals of the memristors connect to the pre-synaptic neurons. This way, when $\Delta T > 0$ the memristors see a negative voltage beyond threshold and their resistance (inverse of synaptic strength) will decrease. Alternatively, the same result can be obtained by having the neurons generate an inverted spike action potential (as in Fig. 10(c)) but connecting the memristors with opposite polarity, as shown in Fig. 13(b).

In an excitatory synapse a pre-synaptic action potential spike should produce an increment in the neuron integral, i.e., make the integrator output voltage V_{int} approach V_{REF} (see Fig. 12). During neuronal spike integration a neuron simply accumulates the contributions of incoming spikes on its integrator. All synapses connected to its input node do not experiment any weight update and operate as resistances of constant value. This is guaranteed by making the action potential peaks lower than the threshold value v_{th} in Fig. 6. In order to have a constant positive resistance contribute a net positive charge packet during each incoming pre-synaptic spike, the net area under the spike waveform (see Fig. 10) should be positive. For the particular case of parameter selection that results in the action potential shapes in Fig. 10, it turns out that the spike in Fig. 10(a) presents a net negative area while the spike in Fig. 10(b) presents a net positive area. Consequently, using spikes with the shape and parameters as in Fig. 10(a) results in synapses delivering net negative charges, while using spikes with the shape and parameters as in Fig. 10(b) results in synapses delivering net positive charges. If neurons are set such that $V_{REF} > V_{rest}$, the incoming net negative charge packets make the integrator output V_{int} approach V_{REF}. In this case synapses delivering net negative charge packets operate as excitatory synapses. On the contrary, if neurons are set such that $V_{REF} < V_{rest}$ then synapses delivering net positive charge packets operate as excitatory synapses. The arrangement shown in Fig. 13(a), which uses spikes as in Fig. 10(a), therefore results in excitatory synapses delivering net negative charge packets if $V_{REF} > V_{rest}$. On the other hand, for Fig. 13(b) which uses spikes as in Fig. 10(b), synapses are excitatory, delivering net positive charge packets if $V_{REF} < V_{int}$.

Interestingly, the strength of STDP learning can be modulated by changing the amplitudes (or shapes) of the electric spikes in time. This would allow the implemention of faster learning at the beginning of a learning process, and progressively slow learning down as the system stores acquired knowledge, or even turn it off

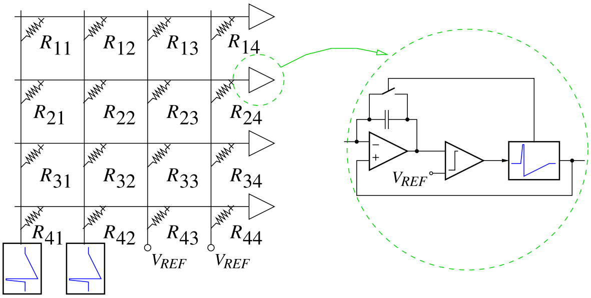


Fig. 14 Small feed forward memristive array simulated behaviorally with Cadence Spectre

completely after some time. This is a very desired feature for STDP machine learning systems [62].

This way of interconnecting memristors with neurons as in Figs. 12 and 13 avoids cross-coupling of spikes between rows and columns, because all lines are driven by (ideal) voltage sources. Using this arrangement with the memristor macro model of Fig. 7 we performed intensive behavioral simulations in Cadence-Spectre to test the concept on the 4 × 4 feed forward array shown in Fig. 14. Note that the neuron used (shown in the inset) is a particular case of the one in Fig. 12 with $V_{refr} = V_{rest} = 0$ (spike resting potential) and $R = \infty$. The results are shown in Fig. 15. Only the first 2 column synapses are stimulated, with 200 ms period spikes (of 45 ms duration) with a 25 ms relative delay between the two columns. As can be seen, only the synapses at the first two columns change their resistance, while those on the other two columns do not, confirming the correct operation of STDP without any crosstalk between columns or rows. This demonstrates that this architecture can be scaled up to arbitrary size, at least conceptually. Practical considerations that could limit maximum size are mainly fan-out of neurons, interconnect delays, and parasitic crosstalk. Note that in Fig. 15 memristor resistances do not always converge to their extreme values R_{min} or R_{max} (as in additive STDP) but that some of them (R_{12}, R_{21}, R_{31}, R_{34}) have converged to intermediate values (as is characteristic for multiplicative STDP).

5.1 STDP Variations

Standard STDP aims to implement the synaptic learning functions of the shape shown in Fig. 3(b). In the case of synapses with negative weights anti-STDP learning functions similar to the one shown in Fig. 3(c) need to be implemented. This is achieved with memristors by simply changing their polarity. Figure 16 shows how neurons and memristors can be interconnected to achieve anti-STDP learning. Memristors are reversed with respect to the cases in Fig. 13. Note that now, for anti-STDP, when $\Delta T > 0$ the memristors see a positive v_{MR} voltage beyond threshold (which will produce an increase in resistance and a decrease in synaptic strength), while for $\Delta T < 0$ they see a negative voltage beyond threshold (which will produce

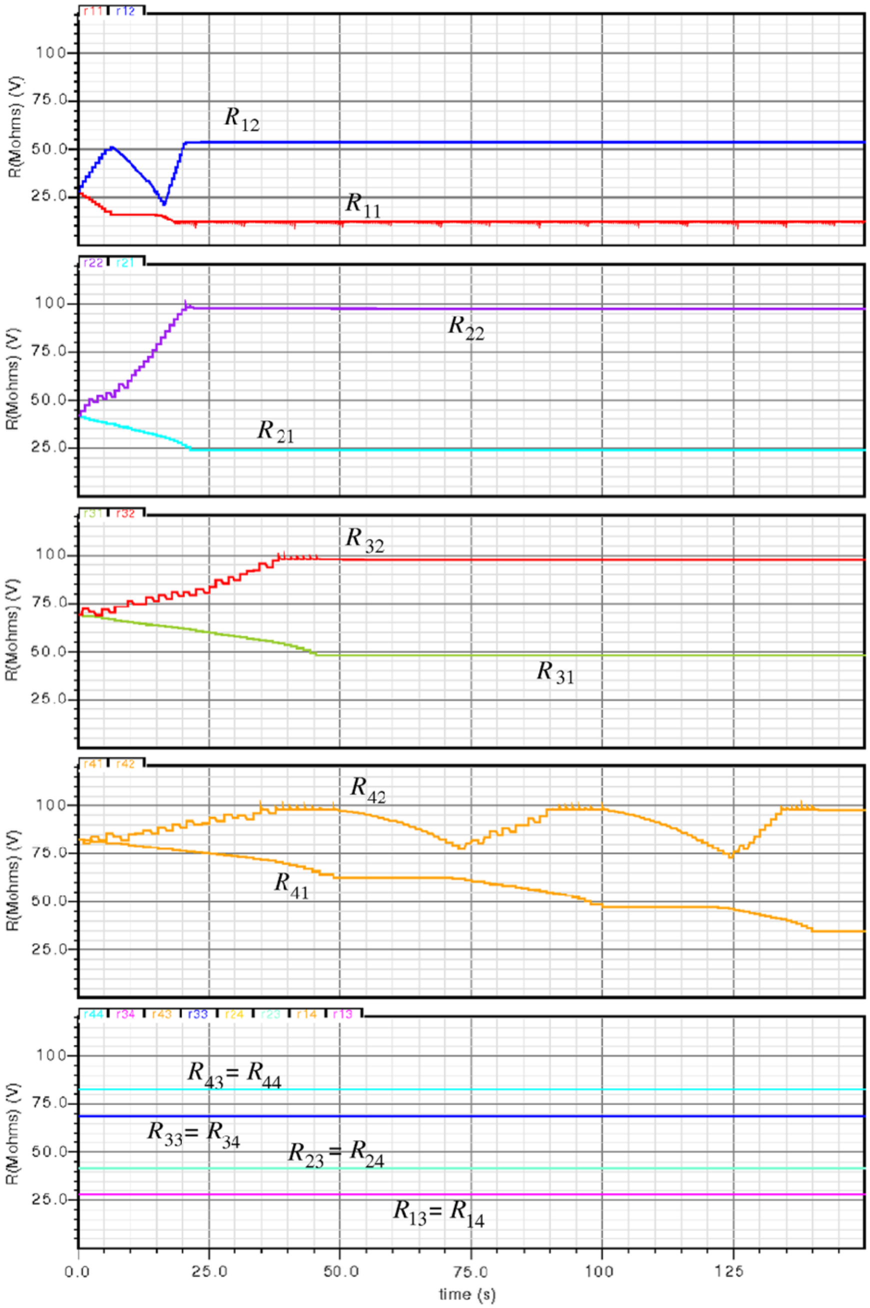


Fig. 15 Evolution of weights (resistances) in a 4 × 4 feed forward memristive perceptron network. The *bottom trace* shows the weights of memristors in the *third* and *fourth column*. The *other traces* show the evolution of weights in the *two columns* furthest to the *left*. Traces are grouped pair-wise with synapses in the same row, and with identical initial conditions

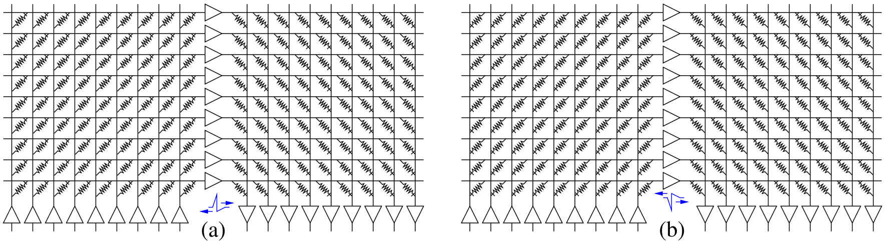


Fig. 16 Memristor connections for anti-STDP learning. (**a**) Using positive action potentials with negative net waveform area as in Fig. 10(a) resulting in synapses delivering net negative charge packets. (**b**) Using inverted action potentials with positive net waveform area as in Fig. 10(b) resulting in synapses delivering net positive charge packets

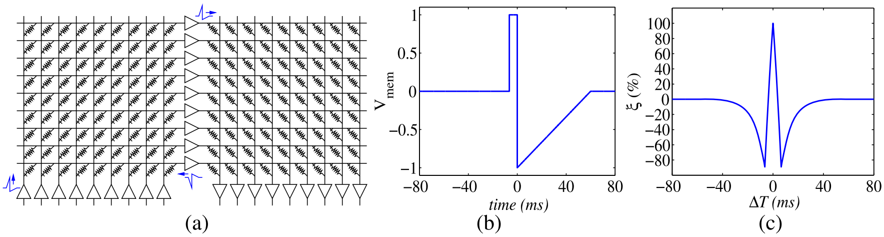


Fig. 17 Arrangement where neurons send back the inverted spike sent forward. (**a**) Feed forward Crossbar Example, (**b**) Spike shapes used, (**c**) and resulting STDP resistance update function

a synaptic strength increment). Memristors are physically positive resistances (of time varying values). Whether memristors act as excitatory or inhibitory synapses is determined by the combination of (1) net area under the action potential waveform (i.e. sign of net charge sent to the post-synaptic neuron) and (2) whether V_{REF} is above or below V_{rest}, as mentioned in the discussion around Fig. 13.

Another twist in STDP variations is obtained by having the neurons send back an inverted version of the spike sent forward, as shown in Fig. 17(a). In this case, it is possible to have the resulting STDP learning function show a positive learning window around $\Delta T = 0$ with positive increments for both positive and negative values of ΔT close to zero. Beyond a specific value of $|\Delta T|$ there are decrements in the weights ($\Delta w < 0$), for both positive and negative sides of ΔT. This is shown in Fig. 17(c). This type of learning is useful under some circumstances [62].

6 Address Event Representation (AER)

AER (Address-Event-Representation) is a well established technology among neuromorphic engineers. AER was originally proposed twenty years ago in Mead's Caltech research lab [63, 64]. For over ten years AER sensory systems were reported by

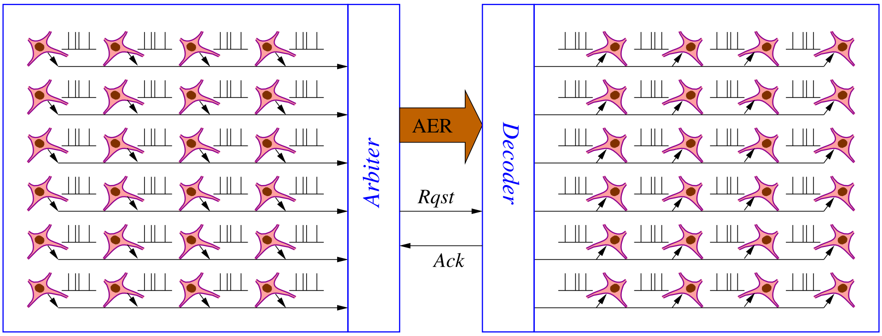


Fig. 18 Illustration of AER point-to-point communication link concept

only a handful of research groups, examples being Lazzaro's [65] and Johns Hopkins University [66] pioneering work on audition, or Boahen's early developments on retinas [67, 68]. However, during these years some basic progress was made. A better understanding of asynchronous design [69, 70] leading to robust unarbitrated [71] and arbitrated [72, 73] asynchronous event readout, combined with the availability of user-friendly FPGA external support for interfacing and new submicron technologies allowing complex pixels in reduced areas, heralded a new trend in AER bio-inspired Spiking Sensor developments. Since 2003 many researchers have embraced this trend. AER has been used fundamentally in vision (retina) sensors, for purposes such as simple light intensity to frequency transformations [74, 75], time-to-first-spike coding [76, 77], foveated sensors [78], spatial contrast [79–82], temporal contrast [74, 83–86], motion sensing and computation, [67], and combined spatial and temporal contrast sensing [87, 88]. AER has also been used for auditory systems [65, 66, 89–91], competition and winner-takes-all networks [92, 93], and even for systems distributed over wireless networks [94].

But AER has also been employed for post-sensing event-driven processing, emulating biological cortical structures. Venier developed AER convolutional filters with elliptic-like kernels [95] while Choi reported more sophisticated Gabor-like filters [96]. In 1999 Serrano reported an AER architecture [97] that would allow more generic kernels, although with some geometric symmetry restrictions. In 2006 the same group started to report working AER Convolution chips with arbitrary shape programmable kernels of size up to 32 × 32 [98–100].

Figure 18 explains the basic idea behind a point-to-point AER link. An emitter chip (or module) includes an array of neurons generating spikes. Each neuron is assigned an address, such as its (x, y) coordinate within the array. Neurons generate spikes at low frequency (10–1000 Hz), and these are arbitrated and put on an inter-chip (or inter-module) high-speed asynchronous AER bus. The AER bus is a multi-bit (either parallel, serial, or mixed) bus which transmits the addresses of the emitting neurons. Typical delays for transmitting Address Events between AER chips range from about 30 ns [81] to 1 μs [85] per event for parallel AER, and have been reported down to 24 ns per event for serial AER with potential to go as low as 8 ns per event [101]. These addresses are received, read, and decoded by the receiver chip (or module) and sent to the corresponding destination neuron or neurons. Figure 18 illustrates a point-to-point AER link with a single emitter and a single receiver. The use of AER splitters and mergers [102] allows extension to one-to-many, many-to-one, or many-to-many AER links. Inserting AER mappers [102] allows coordinate transformations (rotations, translations, etc.) to be performed while address events travel between modules. Current research is looking at how large numbers of AER convolutional modules can be combined through independent and multiple AER links to build high speed object and texture recognition systems [103–105].

7 Building a Self-learning Visual Cortex with Memristors and STDP-Ready AER Hardware

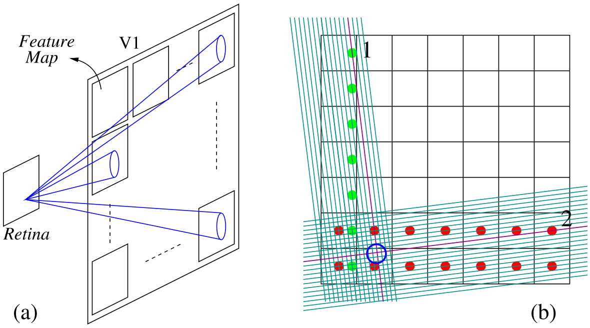
In previous Sections we have shown how to interconnect memristors with spiking neurons to achieve STDP learning systems. We have illustrated this with a very specific topology, a feed forward crossbar structure (Figs. 13, 16, and 17), where all neurons in one layer connect to all neurons in the next layer. However, the methodology is not restricted to this specific spatial topology, and can be extended to any generic neural network topology. In this Section we will apply those same concepts to a topology representing the first processing layer of the visual cortex, namely layer V1. We will first explain the V1 layer topology we will use, show how to build it physically, then we will describe the training data we will use from a real artificial AER retina, and finally we will show the receptive fields formed through STDP learning in the artificial memristive V1 layer with this training data. The biological V1 visual cortex layer is known to be sensitive to specific orientations [106]. We will show how such orientation sensitive receptive fields arise naturally when building an artificial memristive V1 layer with STDP learning and stimulated with real spiking data obtained with an artificial AER motion sensitive retina.

The spontaneous formation of orientation sensitive receptive fields through STDP learning has already been reported by other researchers [17, 28]. In those works static luminance images were used for training. Pixel intensities were coded into spikes through some kind of computational transformation: either a stochastic rate coding scheme [28], or a rank-order coding scheme [17]. Here we directly use the continuous AER output stream of events produced by a real motion sensitive retina CMOS sensor.

7.1 Topology of V1 Visual Cortex Layer and Physical Realization

The simplified V1 topology we want to emulate can be explained with the help of Fig. 19(a). The retina sends spikes to the V1 cortex layer through synaptic connections. The V1 layer is structured in a number of "*Feature Maps*". We can think of

Fig. 19 (**a**) Projection Field Topology of V1 layer in Visual Cortex. (**b**) Hybrid CMOS-memristor arrangement with CMOL style tilted lines



the retina as an array of "*pixels*", each with coordinate (x, y). Let us assume each "*Feature Map*" in V1 replicates the same coordinates (x, y), so that each pixel in the retina has a corresponding pixel in each "*Feature Map*" with the same coordinate. Each pixel (x_c, y_c) in a "*Feature Map*" receives inputs not only from pixel (x_c, y_c) in the retina, but also from all neighbors within a spatial neighborhood $(x_c + x_r, y_c + y_r)$. Alternatively, we may say that each pixel in the retina (x_c, y_c) connects to a *Projection Field* of pixels $(x_c + x_r, y_c + y_r)$ in each of the *Feature Maps*. Thus, *projection fields* include a number of synaptic connections, so that the spikes produced by one pixel in the retina are sent to the pixels of the *projection field* in each *feature map*. Feature maps operate as feature extractors. Specifically, the feature maps in V1 detect the presence of oriented edges at specific orientations and scales [106].

The physical implementation of one such *Feature Map* with AER CMOS neurons and a layer of memristor crossbar structure on top is shown in Fig. 19(b) [29]. The lower CMOS contains the array of neurons (or pixels) of one V1 *Feature Map*. Each neuron has coordinate (x, y), as its corresponding retina pixel. Address Event spikes of coordinate (x, y) coming from the retina are sent to pixel (x, y) in the *Feature Map*. This neuron then sends out a spike of the desired shape (for example, as in Fig. 10) through its output node. In Fig. 19(b) each neuron has an output node (green) and an input node (red). The output node connects to a nano wire tilted slightly with respect to the CMOS tile (as in CMOL [107]), so that it does not intersect with any other neuron output node in the CMOS tile. This nano wire has many others in parallel, each connecting to one neuron output node. Perpendicular to all these nano wires there are other nano wires (at a different altitude), each connecting to the input node of one neuron. The two sets of perpendicular nano wires form a "sandwich" with a separation layer formed by memristive material. This way, at the intersection of each perpendicular nano wire there is a memristor. For example, in Fig. 19(b) neuron 1 output node connects to the vertical pink nano wire, while neuron 2 input node connects to the horizontal pink nano wire. The synaptic memristor connecting neuron 1 output to neuron 2 input is at the intersection of the two nano wires (blue circle). The vertical pink nano wire (neuron 1 output) has memristive intersections with all horizontal nano wires. Consequently, neuron 1 output connects to all other neuron inputs. In the same manner, all neuron output nodes connect to all

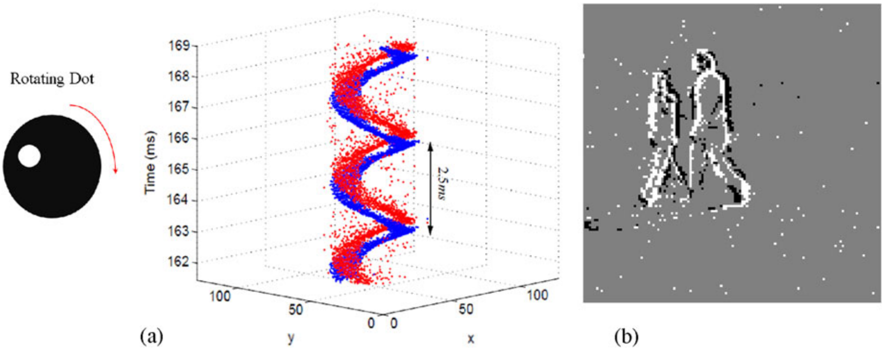


Fig. 20 Illustration of Temporal Derivative Retina Outputs. (**a**) Events produced by a rotating *black disk* with a *white dot*, represented in (x, y, t) coordinates. (**b**) Events collected during 20 ms when observing two people walking

neuron input nodes. For *projection field* based topologies, each neuron output does not connect to all other neuron inputs. Instead, connectivity is limited to a given spatial neighborhood. This is achieved by having nano wires of limited length (instead of one reaching over the full CMOS array). For square projection fields of size 10 k $= 100^2$, for example, each nano wire has to be extended to 50 cells on each side.

Below we present some simulation results from training a set of such AER *Feature Maps* with real stimuli coming from a temporal derivative AER retina watching life scenes. First, we briefly explain the AER temporal derivative retina and what kind of spikes it produces. We then describe how we used this data to stimulate a set of AER hybrid CMOS-memristor *Feature Maps* and what kind of selectivity these *Feature Maps* developed.

7.2 AER Temporal Difference Retina

We will use the spiking data obtained from an AER Temporal Difference Retina chip [74, 84–86] to train an artificial V1 STDP visual cortex layer. The retina has an array of 128 × 128 pixels. Each pixel (x, y) has a photo sensor that provides a continuous photo current $I_{ph}(x, y)$ plus a circuitry that generates a signed spike every time its photo current changes by a given relative amount $|\Delta I_{ph}/I_{ph}| > C_{th}$. Figure 20(a) shows the output events produced by the retina when observing a dot rotating at 400 Hz. Blue dots represent positive events (going from dark to bright) and red dots represent negative events (going from bright to dark). The address events collected during 7 ms are plotted in (x, y, t) coordinates. Figure 20(b) shows the events collected during 20 ms when observing two people walking. White pixels correspond to positive events ($\Delta I_{ph}/I_{ph} > 0$), while black pixels to negative events ($\Delta I_{ph}/I_{ph} < 0$).

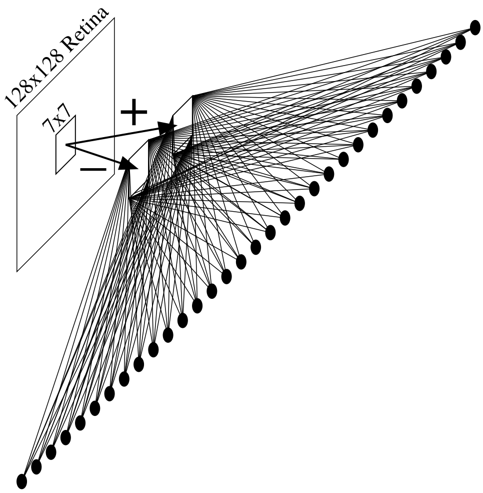


Fig. 21 Topology used for testing STDP on V1-like Network

7.3 STDP Training Results of V1 Layer

In this Section we will analyse the learning behavior a hybrid CMOS-memristive (wall model) V1-like system when it is trained through STDP using the architectural and circuital principles outlined throughout the Chapter and using real stimuli obtained from a 128 × 128 pixel AER temporal derivative retina. Specifically, we used a 521 sec recording with 20.5 million events showing scenes observed when driving in a car [108]. We used a simplified network structure to simulate and see what kind of receptive fields would naturally arise. The network structure is shown in Fig. 21. From the retina visual field of 128 × 128 pixels we cropped 324 non-overlapping patches of 7 × 7 pixels each, and concatenated all these events sequentially making a recording of 324 × 521 = 168804 sec (47 hours) with 19.6 million events. This concatenation was used for one training epoch, and we required a total of 5 epochs to observe convergence in the learned weights. The events from each patch are separated into two additional 7 × 7 fields depending on the event sign. The activity of these two subfields is projected onto 32 neurons.[4] Consequently, there are 32 × 2 × 7 × 7 trainable weights. Weights are always positive. Each of the 32 neurons inhibits the other neurons through lateral inhibitory connections, as in Ref. [23]. Each neuron is as shown in Fig. 12, with a leak and a refractory voltage. Inhibitory lateral connections have fixed weights, while the weights of the feed forward connections follow STDP learning. Weights were initialized either to random values, or to maximum values. The STDP learning functions were such that the ratio of the negative side area over the positive side area was (see Eq. (1)) $a^{-}\tau^{-}/a^{+}\tau^{+} = 1.25$,

[4] Compared to the arrangement in Fig. 19(a), each of the 32 neurons in Fig. 21 represents one of the *Feature Maps* in Fig. 19(a). Consequently, to implement the full V1 structure physically, each neuron with all its 7 × 7 input synapses in Fig. 21 has to be "cloned" in a 128 × 128 array.

meaning that STDP was biased towards depression. The time constant for the positive side was $\tau^+ = 13.6$ ms, while that of the negative side was $\tau^- = 15.2$ ms, and there is a central linear region for $|\Delta T| < 0.5$ ms. Memristor resistances were bounded to the interval $R_{min} = 10\,\mathrm{M}\Omega$, $R_{max} = 100\,\mathrm{M}\Omega$. We simulated this system theoretically in several ways: (1) by solving the differential equations of biological integrate-and-fire neurons via an Euler method with a time step update of 0.1 ms, using the Brian simulator [109] and a conventional additive STDP learning rule; (2) by using a dedicated event based simulator adapted from Ref. [23] implementing the quadratic STDP learning rule of the memristors and the neuron dynamics corresponding to the circuit in Fig. 10 with spikes as in Fig. 8; and (3) by simulating a simplified event driven matlab code with instantaneous neuron dynamics (but including a non-instantaneous leak) and with quadratic multiplicative STDP. In all cases, receptive fields became clearly orientation selective. The resulting receptive fields are biased to vertical edges, similar to the visual input stimuli we have used for this experiment.

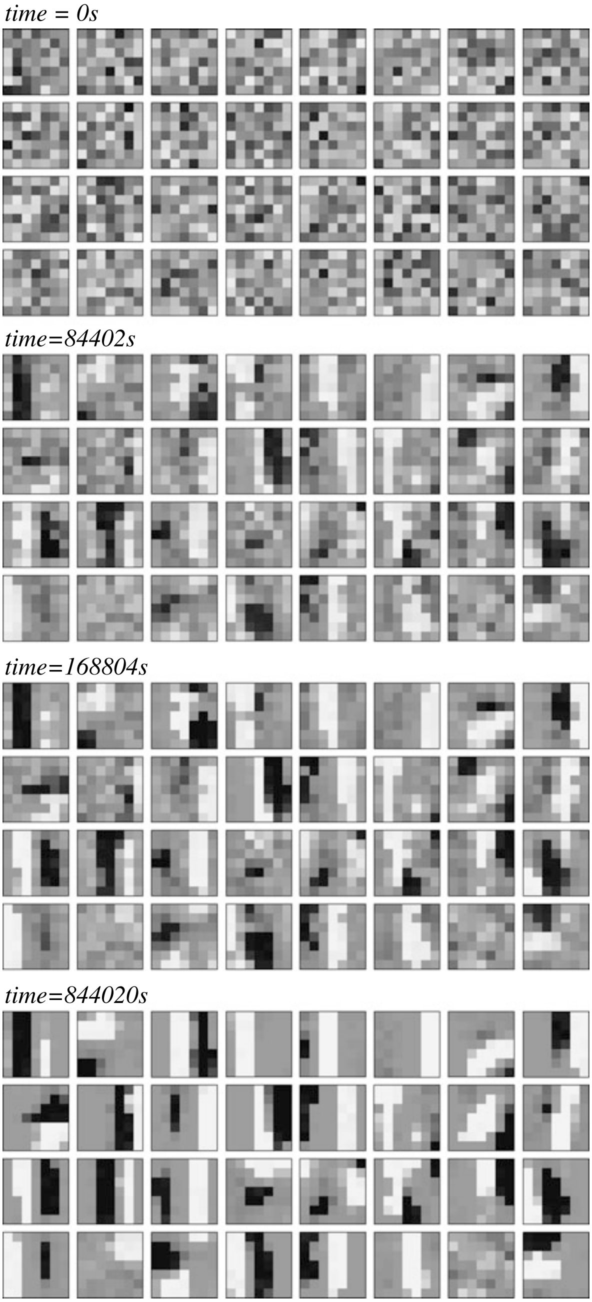
Figure 22 shows the evolution of the receptive fields when using the dedicated event based simulator (case (2) in previous paragraph). We see the receptive fields of the 32 neurons, where the positive and negative weights are grouped together in the same 7 × 7 square by assigning 'white' to positive weights and 'black' to negative weights. The central gray color means zero weight. As can be seen, there is a clear tendency for the receptive fields to become orientation selective.

It is worth mentioning that the type of continuous processing involved here differs from time-to-first-spike (or rank order) coding schemes, where a stimulus onset provides a reference time [17–21]. It also differs from Phase-of-Firing coding, where the peak of a population activity oscillation is used as a reference time [24]. Here there is no oscillation, nor stimulus onset, nor any reference time, but a continuous flow of spikes, and yet STDP is able to pick patterns that are consistently present in the training data, confirming previous results [22]. Future work, however, will evaluate the use of periodic resets in the AER retina, leading to time-to-first-spike coding with respect to those resets.

We also simulated the network in Spectre using the memristor macro model in Sect. 3.1, but for 16 neurons only. However, electrical circuit simulation was very slow. Simulating for just one of the 324 input patches (with about 154 K events) took 559 ksec CPU time (6.5 days) running on a SUN Fire X2200 M2 Linux cluster with dual cores at 2.2 GHz and 4 GB RAM. In this case we could only verify that the initial evolution of weights was similar to those of the software programs. The obtained results are shown in Fig. 23.

For the circuit simulations we used the topology and spike shapes shown in[5] Fig. 24. There are two input memristor arrays, one for the positive and one for the negative subfields in Fig. 21. The backward spikes are attenuated by $\alpha_{pos} = 0.97$. The output neurons forward spikes are rectified and sent back through non-trainable

[5] This topology and these spike shapes also correspond to the theoretical simulations of case (2) and Fig. 22. The only difference between the two cases is in the number of neurons used: 32 neurons for the theoretical simulations, and 16 neurons for the circuit simulations.



time = 0s

time=84402s

time=168804s

time=844020s

Fig. 22 Evolution of 7×7 pixel Receptive Fields through unsupervised STDP training with AER retina data observing life scenes. Weights are shown at different stages of training: initial random weights, after half a training epoch, after one training epoch, and after five training epochs

fixed value resistors $R_{inh} = 2$ MΩ to implement the lateral inhibitory connections. The parameters used for the memristors are $w_{max} = 10$ V, $w_{min} = -10$ V, $C_{MR} = 50$ mF, $R_{min} = 8$ MΩ, $R_{max} = 100$ MΩ, $k_R^{-1} = 217$ nA, $w_o = 11.74$ V,

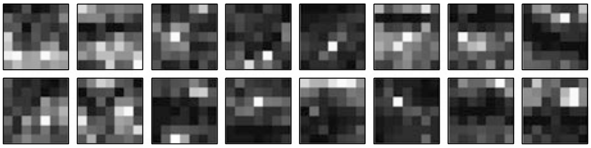


Fig. 23 Receptive fields weight distribution of memristor conductances after training for one patch out of 324 of one epoch, obtained through spectre circuit simulations

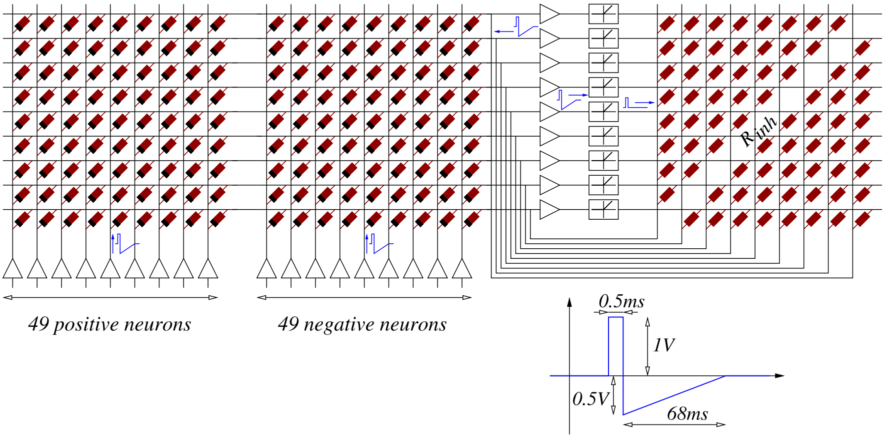


Fig. 24 Circuit Topology and Spike Shape used for the Spectre electrical circuit simulation of the simplified V1 network

$I_o = 10$ µA, $v_o = 0.1$ V, $v_{th} = 1$ V, and for the neuron $R = 1$ MΩ, $C = 19.2$ nF, $V_{refr} = 0.625$ V, $V_{REF} = 1$ V. With these memristor parameters and the spike shapes in Fig. 24 we were able to characterize the STDP learning function of the memristors and adjust them to the learning function in the event based simulation (case (2) above). Figure 25 shows the STDP learning function characterized through electrical spectre simulations (blue dots) to match the ideal function used in the theoretical simulations (red circles).

At this point we would like to highlight an important difference between the memristor-based network of integrate-and-fire neurons with STDP synapses presented here, and an equivalent network as used normally among neurocomputational researchers (see the integrate-and fire neuron model in [110]). In this latter case, the evolution of membrane voltage following an input spike at t_{spk} is as if the spike injects a current $I_{spk}(t > t_{spk}) = I_m e^{-(t-t_{spk})/\tau_{spk}}$ at node V_{pos} in Fig. 12. Parameters I_m and τ_{spk} defining this spike contribution are independent of the parameters a^+, a^-, τ^+, τ^- in Eq. (1) used to characterize the STDP learning function. However, in case of the memristor implementation, each spike injects a current at node V_{pos} in Fig. 12 proportional to the spike waveform delivered by the neurons. Since this waveform also determines the shape of the STDP learning function, it turns out that there is now a strong dependency between the parameters defining the evolu-

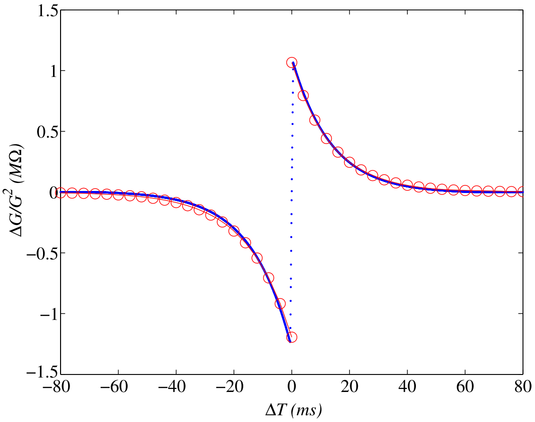


Fig. 25 Spectre characterization of STDP learning function of memristors. Blue dots are results obtained from Spectre electrical circuit simulations. Red circles correspond to the theoretical target function used in the theoretical behavioral simulations

tion of the membrane voltage and those defining the STDP learning function. They are no longer independent and it is consequently more difficult to adjust all true independent parameters properly for a desired behavior.

8 Practical Limitations, Realistic Sizes, Pitches, Density, Crosstalk and Power Considerations

Nanoscale memristor technology is still quite incipient and no realistic large scale systems have been reported at the time of writing (as far as we know). However, we can estimate an orientative scale and density of what may realistically be achieved in the near future, and the main limitations which may be encountered in a real physical implementation.

Regarding the wiring density of synaptic memristors, a pitch of 100 nm is conservatively realistic for present day technologies [111, 112], while the near future might bring us closer to 10 nm [113]. Assuming wafer scale dies of 100 nm pitch 2D memristor arrays capable of interfacing reliably with lower CMOS become available some time soon, this would result in a synaptic density of 10^{10} synapses per cm^2.

In the brain, the number of synapses per neuron is about 10^3 to 10^4. If we want to maintain the 10^4 ratio, we would need to fabricate CMOS neurons with a pitch of 10 μm, resulting in 10^6 neurons per cm^2. Such neuron sizes are quite realistic for present day nanometer scale CMOS (45 nm or 32 nm), given the complexity of the neurons needed.

Another problem is that of resistance value ranges of the memristors' R_{min} (synapse ON) and R_{max} (synapse OFF). Reported memristors present resistance values from the $k\Omega$ range up to the $M\Omega$ range [6–8]. The memristor resistance value range affects the performance, reliability, crosstalk and power dissipation of a full large scale system. For example, it affects the driving capability of the neurons and their power consumption. If one neuron needs to drive 10^4 synapses of average value 1 MΩ to an average 1 V level, it has to be able to provide an average current of 10 mA during a spike (of say 20 ms), delivering 10 mW per spike. If there are 10^6 neurons per cm^2 each firing at an average of 10 Hz (which is similar to biological neurons), the synapses would dissipate a power of about 2 kW. The neurons would need at least the same power, presumably more. It is obvious that such a structure would melt quickly. The resistance range needs to be increased by a minimum factor of 100, so that minimum resistances are at least 100 MΩ, or even larger. As pitches are lowered, resistances would need to increase quadratically with pitch decrease, to maintain the power limitation. Another option would be to scale down voltage, but there is not much range. Even our 1 V maximum voltage assumption is quite optimistic for available present day memristors, which tend to operate between 2–10 V [6–8]. Also, we have always assumed so far that voltage sources driving memristor terminals behave as ideal voltage sources, or at least, that the output resistance of such voltage sources is negligible compared to the total resistance they have to drive. Again, this will be achieved more easily if memristors present rather high resistance values. If driving voltage sources are no longer so ideal, then there will be crosstalk between lines. For example, if a spike is sent to a column then the voltage on all rows would change slightly. The consequence of this is that part of the charge provided by the incoming spike will be lost through non desired synapses and the impact of the spike on the target neurons will be weaker. During learning, the situation is less severe because for STDP update the memristor voltage has to exceed the learning threshold (v_{th} in Eq. (5)). The effect of having non-ideal voltage sources is that the terminal voltage difference on the memristors needing synaptic update would be slightly less than in the ideal situation and learning would be weaker than expected ideally. However, having non-ideal voltage sources would not induce STDP update in undesired synapses. Another issue related to crosstalk is parasitic capacitive crosstalk between lines, which can be more pronounced as pitch and line distances decrease.

Also, one highly critical aspect which needs to be evaluated is the influence of component mismatches. Nano scale devices suffer from high mismatch in general. Consequently, we should expect nano scale memristors too to suffer from great parameter variations from one to another. It is true that they will operate as adaptive devices that will learn their functionality hopefully compensating for (some) mismatches. However, their learning and adaptation rules will also suffer from mismatch, making some synapses learn faster than others, or in slightly different fashions. In any case, the main sources of mismatch in memristor devices still need to be identified, and then their influence in the overall system learning behavior evaluated. However, to undertake such an initiative, we first need ready access to large arrays of reliable memristors fabricated in a stable and repeatable manner.

In general, an important issue is precise memristor modeling. Throughout this Chapter we have assumed an idealized voltage-driven memristor macro model. This is useful to devise possible system architectures to achieve a desired functionality, such as STDP learning. However, to estimate realistic performance figures of resulting systems, it will be necessary to include non-ideal effects, both of the memristors and companion CMOS circuits. No high order effects have been modeled, such as those related to noise, mismatch, and other memristor non-idealities not yet reported.

9 Conclusions

We have shown that STDP learning can be induced by the voltage/flux driven formulation of a memristor device. We have used this formulation to develop fully asynchronous circuit architectures capable of performing STDP, by having neurons send their spikes not only forward but also backwards. We have seen that the STDP learning rule induced this way can be additive or quadratic, depending on the nature of the memristor. We have shown how the shape of spikes is critical to achieve and modulate a specific STDP learning function. We have provided a memristor macro model for simulating arrays of memristors efficiently in circuit simulators. Finally, we have studied an emulation of the V1 visual cortex capable of self-learning how to extract orientations from spiking inputs provided by a real physical AER spiking retina fabricated in CMOS. At the end we have also discussed possible limitations of present day memristors.

The presented results are ideal extrapolations based on behavioral simulations. As memristor devices are further developed and non-ideal effects become known, the impact of non-idealities in the presented architectures and methods can be further assessed. Future work has to evolve towards more realistic memristor models and improved memristor devices, specially devices with much higher resistivities. One critical property that memristors need to provide for efficient STDP and non-volatility is the central dead-zone in Fig. 6(b), which the already reported memristor from Michigan University [8] seems to present. Another issue relates to the quadratic type of multiplicative STDP followed by the presented devices and architectures. This is a quite unusual form of STDP, which needs to be further investigated from a theoretical point of view. In general, there might be stability issues with generic STDP when used in complex biological models [114, 115]. Similarly, since the presented approach allows the shape of the neural spikes, and therefore the shape of the STDP learning curves to be changed in time, further theoretical studies are required to incorporate time varying STDP learning functions for speeding up, stabilizing, or in general improving learning performance.

References

1. Zamarreño-Ramos, C., Camuñas-Mesa, L.A., Perez-Carrasco, J.A., Masquelier, T., Serrano-Gotarredona, T., Linares-Barranco, B.: On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. Front. Neuromorphic Eng., Front. Neurosci. **5**, 26 (2011). Available from www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2011.00026/abstract
2. Serrano-Gotarredona, T., Masquelier, T., Prodromakis, T., Indiveri, G., Linares-Barranco, B.: STDP and STDP variations with memristors for spiking neuromorphic learning systems. Front. Neuromorphic Eng., Front. Neurosci. **7**, 02 (2013). Available from www.frontiersin.org/Neuroscience/10.3389/fnins.2013.00002/abstract
3. http://en.wikipedia.org/wiki/Neuromorphic
4. Mead, C.: Analog VLSI and Neural Systems. Addison-Wesley, Reading (1989)
5. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**(1), 80–83 (2008)

6. Borghetti, J., Li, Z., Straznicky, J., Li, X., Ohlberg, D.A.A., Wu, W., Stewart, D.R., Williams, R.S.: A hybrid nanomemristor/transistor logic circuit capable of self-programming. Proc. Natl. Acad. Sci. USA **106**(6), 1699–1703 (2009)
7. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. **10**(4), 1297–1301 (2010)
8. Jo, S.H., Kim, K.-H., Lu, W.: High-density crossbar arrays based on a Si memristive system. Nano Lett. **9**(2), 870–874 (2009)
9. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**, 507–519 (1971)
10. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
11. Chua, L.O., Desoer, C.A., Kuh, E.S.: Linear and Nonlinear Circuits. McGraw-Hill, New York (1987)
12. Gerstner, W., Ritz, R., Hemmen, J.L.: Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. Biol. Cybern. **69**, 503–515 (1993)
13. Gerstner, W., Kempter, R., van Hemmen, J.L., Wagner, H.: A neuronal learning rule for sub-millisecond temporal coding. Lett. Nat. **383**, 76–78 (1996)
14. Sjöström, J., Gerstner, W.: Spike-timing dependent plasticity. In: Scholarpedia, the Peer-Reviewed Open-Access Encyclopedia, vol. 5(2), p. 1362 (2010). Available from http://www.scholarpedia.org/article/STDP
15. Rao, R.P.N., Sejnowski, T.J.: Spike-time-dependent Hebbian plasticity as temporal difference learning. Neural Comput. **13**, 2221–2237 (2001)
16. Porr, B., Wörgötter, F.: How the shape of pre- and postsynaptic signals can influence STDP: a biophysical model. Neural Comput. **16**, 595–625 (2004)
17. Delorme, A., Perrinet, L., Thorpe, S.J.: Networks of integrate-and-fire neurons using rank order coding B: spike timing dependent plasticity and emergence of orientation selectivity. Neurocomputing **38–40**, 539–545 (2001)
18. Guyonneau, R., VanRullen, R., Thorpe, S.J.: Temporal codes and sparse representations: a key to understanding rapid processing in the visual system. J. Physiol. Paris **98**(4–6), 487–497 (2004)
19. Masquelier, T., Thorpe, S.J.: Unsupervised learning of visual features through spike timing dependent plasticity. PLoS Comput. Biol. **3**(2), e31 (2007)
20. Weidenbacher, U., Neumann, H.: Unsupervised learning of head pose through spike-timing dependent plasticity. In: Perception in Multimodal Dialogue Systems. Lecture Notes in Computer Science, vol. 5078, pp. 123–131. Springer, Berlin (2008)
21. Masquelier, T., Thorpe, S.J.: Learning to recognize objects using waves of spikes and spike timing-dependent plasticity. In: Proc. of the 2010 IEEE Int. Joint Conf. on Neural Networks (2010). doi:10.1109/IJCNN.2010.5596934
22. Masquelier, T., Guyonneau, R., Thorpe, S.J.: Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. PLoS ONE **3**(1), e1377 (2008)
23. Masquelier, T., Guyonneau, R., Thorpe, S.J.: Competitive STDP-based spike pattern learning. Neural Comput. **21**(5), 1259–1276 (2009). doi:10.1162/neco.2008.06-08-804
24. Masquelier, T., Hugues, E., Deco, G., Thorpe, S.J.: Oscillations, phase-of-firing coding and spike timing-dependent plasticity: an efficient learning scheme. J. Neurosci. **29**(43), 13484–13493 (2009)
25. Young, J.M.: Cortical reorganization consistent with spike timing-but not correlation-dependent plasticity. Nat. Neurosci. **10**(7), 887–895 (2007)
26. Finelli, L.A., Haney, S., Bazhenov, M., Stopfer, M., Sejnowski, T.J.: Synaptic learning rules and sparse coding in a model sensory system. PLoS Comput. Biol. **4**(4), e1000062 (2008)
27. Hebb, D.O.: The Organization of Behavior, a Neuropsychological Study. Wiley, New York (1949)
28. Snider, G.S.: Self-organized computation with unreliable, memristive nanodevices. Nanotechnology **18**, 365202 (2007)
29. Snider, G.S.: Spike-timing-dependent learning in memristive nanodevices. In: IEEE Int. Symp. Nano Architectures, pp. 85–92 (2008)

30. Linares-Barranco, B., Serrano-Gotarredona, T.: Memristance can explain spike-time-dependent-plasticity in neural synapses. Available from Nature Precedings http://hdl.handle.net/10101/npre.2009.3010.1, March 2009
31. Markram, H., Lübke, J., Frotscher, M., Sakmann, B.: Regulation of synaptic efficacy by coincidence of postsynaptic APS and EPSPS. Science **275**(5297), 213–215 (1997)
32. Bi, G., Poo, M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. J. Neurosci. **18**(24), 10464–10472 (1998)
33. Bi, G., Poo, M.M.: Synaptic modification by correlated activity: Hebb's postulate revisited. Annu. Rev. Neurosci. **24**, 139–166 (2001)
34. Zhang, L., Tao, H., Holt, C., Harris, W., Poo, M.: A critical window for cooperation and competition among developing retinotectal synapses. Nature **395**(6697), 37–44 (1998)
35. Feldman, D.: Timing-based LTP and LTD at vertical inputs to layer II/III pyramidal cells in rat barrel cortex. Neuron **27**(1), 45–56 (2000)
36. Mu, Y., Poo, M.M.: Spike timing-dependent LTP/LTD mediates visual experience-dependent plasticity in a developing retinotectal system. Neuron **50**(1), 115–125 (2006)
37. Cassenaer, S., Laurent, G.: Hebbian STDP in mushroom bodies facilitates the synchronous flow of olfactory information in locusts. Nature **448**(7154), 709–713 (2007)
38. Jacob, V., et al.: Spike-timing-dependent synaptic depression in the in vivo barrel cortex of the rat. J. Neurosci. **27**(6), 1271–1284 (2007)
39. Rubin, J.E., Gerkin, R.C., Bi, G.-Q., Chow, C.C.: Calcium time course as a signal for spike-timing-dependent plasticity. J. Neurophysiol. **93**, 2600–2613 (2005)
40. Lubenov, E.V., Siapas, A.G.: Decoupling through synchrony in neuronal circuits with propagation delays. Neuron **58**, 118–131 (2008)
41. Querlioz, D., Bichler, O., Gamrat, C.: Simulation of a memristor-based spiking neural network immune to device variations. In: Proc. Int. Joint Conf. Neural Networks (IJCNN), pp. 1775–1781 (2011)
42. Bichler, O., Suri, M., Querlioz, D., Vuillaume, D., DeSalvo, B., Gamrat, C.: Visual pattern extraction using energy-efficient "2-PCM synapse" neuromorphic architecture. IEEE Trans. Electron Devices **59**(8), 2206–2214 (2012)
43. Bichler, O., Querlioz, D., Thorpe, S.J., Bourgoin, J.P., Gamrat, C.: Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. Neural Netw. **32**, 339–348 (2012)
44. Nessler, B., Pfeiffer, M., Maass, W.: STDP enables spiking neurons to detect hidden causes of their inputs. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems, pp. 1357–1365. MIT Press, Cambridge (2010)
45. Abbott, L.F., Nelson, S.B.: Synaptic plasticity: taming the beast. Nat. Neurosci. **3**, 1178–1183 (2000)
46. van Rossum, M.C.W., Bi, G.Q., Turrigiano, G.G.: Stable Hebbian learning from spike timing-dependent plasticity. J. Neurosci. **20**(23), 8812–8821 (2000)
47. Rubin, J., Lee, D.D., Sompolinsky, H.: Equilibrium properties of temporally asymmetric Hebbian plasticity. Phys. Rev. Lett. **86**(2), 364–367 (2001)
48. Gütig, R., Aharonov, R., Rotter, S., Sompolinsky, H.: Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity. J. Neurosci. **23**(9), 3697–3714 (2003)
49. Argall, F.: Switching phenomena in titanium oxide thin films. In: Solid-State Electronics, vol. 11, pp. 535–541. Pergamon, Elmsford (1968)
50. Swaroop, B., West, W.C., Martinez, G., Kozicki, M.N., Akers, L.A.: Programmable current mode Hebbian learning neural network using programmable metallization cell. In: Proc. of the IEEE Int. Symp. on Circ. and Syst. (ISCAS1998), pp. III.33–III.36 (1998)
51. Joglekar, Y.N., Wolf, S.J.: The elusive memristor: properties of basic electrical circuits. Eur. J. Phys. **30**, 661–675 (2009)
52. Biolek, Z., Biolek, D., Biolkova, V.: Spice model for memristor with nonlinear dopant drift. Radioengineering **18**(2), 210–214 (2009)

53. Nian, Y., Strozier, J., Wu, N., Chen, X., Ignatiev, A.: Evidence for an oxygen diffusion model for the electric pulse induced resistance change effect in transition-metal oxides. Phys. Rev. Lett. **98**(14), 146–403 (2007)
54. Yang, J.J., Pickett, M.D., Li, X., Ohlberg, D.A.A., Stewart, D.R., Williams, R.S.: Memristive switching mechanism for metal/oxide/metal nanodevices. Nat. Nanotechnol. **3**(7), 429–433 (2008)
55. Hur, J., Lee, M.-J., Lee, C., Kim, Y.-B., Kim, C.J.: Modeling for bipolar resistive memory switching in transition-metal oxides. Phys. Rev. B **82**(15) (2010)
56. Wuttig, M., Yamada, N.: Phase-change materials for rewriteable data storage. Nat. Mater. **6**(11), 824–832 (2007)
57. Kwon, D.H., et al.: Atomic structure of conducting nanofilaments in TiO_2 resistive switching memory. Nat. Nanotechnol. **5**, 148–153 (2010)
58. Yang, Y., Gao, P., Gaba, S., Chang, T., Pan, X., Lu, W.: Observation of conducting filament growth in nanoscale resistive memories. Nat. Commun. **3**, 732 (2012)
59. Shihong, M.W., Prodromakis, T., Salaoru, I., Toumazou, C.: Modelling of current percolation channels in emerging resistive switching elements. arXiv:1206.2746v1 [cond-mat.mes-hall]
60. Chanthbouala, A., Garcia, V., Cherifi, R.O., Bouzehouane, K., Fusil, S., Moya, X., Xavier, S., Yamada, H., Deranlot, C., Mathur, N.D., Bibes, M., Bartélémy, A., Grollier, J.: A ferroelectric memristor. Nat. Mater. **11**, 860–864 (2012)
61. Indiveri, G., et al.: Neuromorphic silicon neuron circuits. Inaugural issue of Frontiers in Neuromorphic Engineering. http://www.frontiersin.org/neuromorphic_engineering
62. Maass, W.: Private communication
63. Sivilotti, M.: Wiring considerations in analog VLSI systems with application to field-programmable networks. PhD, Computation and Neural Systems, Caltech, Pasadena California (1991)
64. Mahowald, M.A.: VLSI analogs of neuronal visual processing: a synthesis of form and function. PhD, Computation and Neural Systems, Caltech, Pasadena, California (1992)
65. Lazzaro, J., Wawrzynek, J., Mahowald, M., Silvilotti, M., Gillespie, D.: Silicon auditory processors as computer peripherals. IEEE Trans. Neural Netw. **4**, 523–528 (1993)
66. Cauwenberghs, G., Kumar, N., Himmelbauer, W., Andreou, A.G.: An analog VLSI chip with asynchronous interface for auditory feature extraction. IEEE Trans. Circuits Syst., Part II **45**, 600–606 (1998)
67. Boahen, K.: Retinomorphic chips that see quadruple images. In: Proc. Int. Conf. Microelectronics for Neural, Fuzzy and Bio-Inspired Systems (Microneuro99), Granada, Spain, pp. 12–20 (1999)
68. Boahen, K.: A retinomorphic chip with parallel pathways: encoding INCREASING, ON, DECREASING, and OFF visual signals. Int. J. Analog Integr. Circuits Signal Process. **30**, 121–135 (2002)
69. Martin, A.J., Nyström, M.: Asynchronous techniques for system-on-chip design. Proc. IEEE **94**(6), 1089–1120 (2006)
70. Sparsø, J., Furber, S.B.: Principles of Asynchronous Circuit Design: A Systems Perspective. Kluwer Academic, Dordrecht (2001)
71. Mortara, A., Vittoz, E.A., Venier, P.: A communication scheme for analog VLSI perceptive systems. IEEE J. Solid-State Circuits **30**(6), 660–669 (1995)
72. Boahen, K.: Retinomorphic vision systems. In: Microneuro'96: Fifth Int. Conf. on Neural Networks and Fuzzy Systems, Laousanne, Switzerland, February 1996
73. Boahen, K.: Point-to-point connectivity between neuromorphic chips using address events. IEEE Trans. Circuits Syst., Part II **47**(5), 416–434 (2000)
74. Posch, C., et al.: A QVGA 143 dB dynamic range asynchronous address-event PWM dynamic image sensor with lossless pixel-level video compression. In: ISSCC Dig. of Tech. Papers, pp. 400–401 (2010)
75. Culurciello, E., Etienne-Cummings, R., Boahen, K.A.: A biomorphic digital image sensor. IEEE J. Solid-State Circuits **38**, 281–294 (2003)

76. Ruedi, P.F., et al.: A 128 × 128, pixel 120-dB dynamic-range vision-sensor chip for image contrast and orientation extraction. IEEE J. Solid-State Circuits **38**, 2325–2333 (2003)
77. Chen, S., Bermak, A.: Arbitrated time-to-first spike CMOS image sensor with on-chip histogram equalization. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **15**(3), 346–357 (2007)
78. Azadmehr, M., Abrahamsen, J., Häfliger, P.: A foveated AER imager chip. In: Proc. of the IEEE Int. Symp. on Circ. and Syst. (ISCAS2005), Kobe, Japan, pp. 2751–2754 (2005)
79. Massari, N., et al.: A 100 uW 64 × 128-pixel contrast-based asynchronous binary vision sensor for wireless sensor networks. In: IEEE ISSCC Dig. of Tech. Papers, pp. 588–638 (2008)
80. Ruedi, P.F., et al.: An SoC combining a 132 dB QVGA pixel array and a 32 b DSP/MCU processor for vision applications. In: IEEE ISSCC Dig. of Tech. Papers, pp. 46–47, 47a (2009)
81. Costas-Santos, J., et al.: A contrast retina with on-chip calibration for neuromorphic spike-based AER vision systems. IEEE Trans. Circuits Syst. I, Regul. Pap. **54**(7), 1444–1458 (2007)
82. Leñero-Bardallo, J.A., Serrano-Gotarredona, T., Linares-Barranco, B.: A five-decade dynamic-range ambient-light-independent calibrated signed-spatial-contrast AER retina with 0.1 ms latency and optional time-to-first-spike mode. IEEE Trans. Circuits Syst., Part I **57**(10), 2632–2643 (2010)
83. Mallik, U., et al.: Temporal change threshold detection imager. In: IEEE ISSCC Dig. of Tech. Papers, pp. 362–363 (2005)
84. Posch, C., et al.: A dual-line optical transient sensor with on-chip precision time-stamp generation. In: IEEE ISSCC Dig. of Tech. Papers, pp. 500–618 (2007)
85. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128 × 128 120 dB 30 mW asynchronous vision sensor that responds to relative intensity change. IEEE J. Solid-State Circuits **43**(2), 566–576 (2008)
86. Leñero-Bardallo, J.A., Serrano-Gotarredona, T., Linares-Barranco, B.: A 3.6 μs latency frame-free event-driven dynamic vision sensor. IEEE J. Solid-State Circuits (2010)
87. Zaghloul, K.A., Boahen, K.: Optic nerve signals in a neuromorphic chip: part 1. IEEE Trans. Biomed. Eng. **51**, 657–666 (2004)
88. Zaghloul, K.A., Boahen, K.: Optic nerve signals in a neuromorphic chip: part 2. IEEE Trans. Biomed. Eng. **51**, 667–675 (2004)
89. Sarpeshkar, R., et al.: An analog bionic ear processor with zero-crossing detection. In: ISSCC Dig. of Tech. Papers, pp. 78–79 (2005)
90. Wen, B., Boahen, K.: A 360-channel speech preprocessor that emulates the cochlear amplifier. In: ISSCC Dig. of Tech. Papers, pp. 556–557 (2006)
91. Chan, V., Liu, S.-C., van Schaik, A.: AER EAR: a matched silicon cochlea pair with address event representation interface. IEEE Trans. Circuits Syst., Part I **54**, 48–59 (2007)
92. Chicca, E., Whatley, A.M., Lichtsteiner, P., Dante, V., Delbruck, T., Del Giudice, P., Douglas, R.J., Indiveri, G.: A multichip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity. IEEE Trans. Circuits Syst., Part I **54**(5), 981–993 (2007)
93. Oster, M., Wang, Y., Douglas, R., Liu, S.-C.: Quantification of a spike-based winner-take-all VLSI network. IEEE Trans. Circuits Syst., Part 1 **55**(10), 3160–3169 (2008)
94. Teixeira, T., Andreou, A.G., Culurciello, E.: Event-based imaging with active illumination in sensor networks. In: Proc. of the IEEE Int. Symp. on Circ. and Syst. (ISCAS2005), Kobe, Japan, pp. 644–647 (2005)
95. Vernier, P., Mortara, A., Arreguit, X., Vittoz, E.A.: An integrated cortical layer for orientation enhancement. IEEE J. Solid-State Circuits **32**(2), 177–186 (1997)
96. Choi, T.Y.W., Merolla, P., Arthur, J., Boahen, K., Shi, B.E.: Neuromorphic implementation of orientation hypercolumns. IEEE Trans. Circuits Syst., Part I **52**(6), 1049–1060 (2005)
97. Serrano-Gotarredona, T., Andreou, A.G., Linares-Barranco, B.: AER image filtering architecture for vision processing systems. IEEE Trans. Circuits Syst., Part I **46**(9), 1064–1071 (1999)

98. Serrano-Gotarredona, R., et al.: A neuromorphic cortical-layer microchip for spike-based event processing vision systems. IEEE Trans. Circuits Syst., Part I **53**(12), 2548–2566 (2006)
99. Serrano-Gotarredona, R., et al.: On real-time AER 2D convolutions hardware for neuromorphic spike based cortical processing. IEEE Trans. Neural Netw. **19**(7), 1196–1219 (2008)
100. Camuñas-Mesa, L., Acosta-Jiménez, A., Zamarreño-Ramos, C., Serrano-Gotarredona, T., Linares-Barranco, B.: A 32 × 32 convolution processor chip for address event vision sensors with 155 ns event latency and 20 Meps throughput. IEEE Trans. Circuits Syst., Part-I **58**(4), 777–790 (2011). doi:10.1109/TCSI.2010.2078851
101. Berge, H.K.O., Hafliger, P.: High-speed serial AER on FPGA. In: Proc. of the IEEE Int. Symp. on Circ. and Syst. (ISCAS2007), pp. 857–860 (2005)
102. Serrano-Gotarredona, R., et al.: CAVIAR: a 45 k neuron, 5 m synapse, 12 g connects/s AER hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking. IEEE Trans. Neural Netw. **20**(9), 1417–1438 (2009)
103. Pérez-Carrasco, J.A., Acha, B., Serrano, C., Camuñas-Mesa, L., Serrano-Gotarredona, T., Linares-Barranco, B.: Fast vision through frame-less event-based sensing and convolutional processing. application to texture recognition. IEEE Trans. Neural Netw. **21**(4), 609–620 (2010)
104. Camuñas-Mesa, L., Pérez-Carrasco, J.A., Zamarreño-Ramos, c., Serrano-Gotarredona, T., Linares-Barranco, B.: On scalable spiking ConvNet hardware for cortex-like visual sensory processing systems. IEEE Int. Symp. Circuits Syst. Proc. **2010**, 249–252 (2010)
105. Zamarreño-Ramos, C., Linares-Barranco, A., Serrano-Gotarredona, T., Linares-Barranco, B.: Multi-casting mesh AER: a scalable assembly approach for reconfigurable neuromorphic structured AER systems. Application to ConvNets. IEEE Trans. Biomed. Circuits Syst. **7**(1), 82–102 (2013)
106. Hubel, D.H., Wiesel, T.N.: Receptive fields of single neurones in the cat's striate cortex. J. Physiol. **148**, 574–591 (1959)
107. Strukov, D.B., Likharev, K.K.: CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices. Nanotechnology **16**, 888–900 (2005)
108. Delbruck, T.: Driving in Pasadena to the post office, from Vision Project jAER download area. Available at http://sourceforge.net/apps/trac/jaer/wiki/AER%20data
109. Goodman, D., Brette, R.: Brian: a simulator for spiking neural networks in python. Front. Neuroinform. **2**, 5 (2008). doi:10.3389/neuro.11.005.2008
110. Gerstner, W.: Spiking neurons. In: Maass, W., Bishop, C.M. (eds.) Pulsed Neural Networks. MIT Press, Cambridge (1999). Chap. 1
111. Green, J.E., Choi, J.W., Boukai, A., Bunimovich, Y., Johnston-Halperin, E., DeIonno, E., Luo, Y., Sheriff, B.A., Xu, K., Shin, Y.S., Tseng, H.-R., Stoddart, J.F., Heath, J.R.: A 160-kilobit molecular electronic memory patterned at 10^{11} bits per square centimetre. Nature **445**, 414–417 (2007)
112. Jung, G.-Y., Johnston-Halperin, E., Wu, W., Yu, Z., Wang, S.-Y., Tong, W.M., Li, Z., Green, J.E., Sheriff, B.A., Boukai, A., Bunimovich, Y., Heath, J.R., Williams, R.S.: Circuit fabrication at 17 nm half-pitch by nanoimprint lithography. Nano Lett. 351–354 (2006)
113. Jeon, H.-J., Kim, K.H., Baek, Y.-K., Kim, D.W., Jung, H.-T.: New top-down approach for fabricating high-aspect-ratio complex nanostructures with 10 nm scale features. Nano Lett. **10**(9), 3604–3610 (2010)
114. Izhikevich, E.M., Desai, N.S.: Relating STDP to BCM. Neural Comput. **15**, 1511–1523 (2003)
115. Watt, A.J., Desai, N.S.: Homeostatic plasticity and STDP: keeping a neuron's cool in a fluctuating world. Front. Syn. Neurosci. **2**, 5 (2010). doi:10.3389/fnsyn.2010.00005

Memristor Bridge-Based Artificial Neural Weighting Circuit

Hyongsuk Kim, Maheshwar P. Sah, Changju Yang, Tamás Roska, and Leon O. Chua

Abstract A novel memristor bridge circuit which is able to perform zero, negative and positive synaptic weightings in neuron cells is proposed. It is composed of four memristors and three transistors for weighting operation and voltage-to-current conversion, respectively. It is compact as it can be fabricated in nano meter scale. It is power efficient since it operates in pulse-based. Its input terminals are utilized commonly for applying both weight programming and weight processing signals via time sharing. By programming on each memristor of the memristor bridge circuit, the signed weighting values can be set on the memristor bridge synapses. The features of proposed architecture are investigated via various simulations.

H. Kim (✉) · M.P. Sah · C. Yang
Intelligent Robots Research Center and Division of Electronics and Information Engineering, Chonbuk National University, Jeonju 561-756, Republic of Korea
e-mail: hskim@jbnu.ac.kr

M.P. Sah
e-mail: maheshwarsah@hotmail.com

C. Yang
e-mail: ychangju@chonbuk.ac.kr

T. Roska
Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest, Hungary
e-mail: roska@sztaki.hu

T. Roska
Faculty of Information Technology, Pázmány P. Catholic University, Budapest, Hungary

L.O. Chua
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA
e-mail: chua@eecs.berkeley.edu

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_12,

1 Introduction

It is known that the human brain contains more than 100 billion neurons and each neuron has more than 20,000 synapses [1]. Efficient circuit implementation of synapses, therefore, is very important to build a brain-like machine. However, due to the lack of proper device to implement the synapses, research in this area has only limited success. Among them, the cellular neural network (CNN) [2–8] is known as one of successful approaches. However, even for the case of the CNN, more efficient and smaller devices for implementing its final goal, a brain-like visual computer, are needed.

Recently, the Stanley Williams group [9] has built a nano-scale TiO_2 device which is a nonvolatile and exhibits synaptic characteristics [10–13]. The memory application of the TiO_2 had been reported earlier in [14], but without any explanation on the physical nature of the device. The fundamental contribution by the Williams group is that they were the first to recognize their TiO_2 resistive switches is in fact a memristor, over a limited operating range, and that it can be modeled as a memristive device [15] in general. The memristor was postulated by Leon Chua in 1971 as the fourth basic element of electrical circuits [16] and later the concept was generalized to a class of dynamical devices (systems) called *memristive device* [15]. A common fingerprint of both the memristor, and the memristive device is their pinched hysteresis loops in the current vs. voltage plane, under sinusoidal excitations, or under any bipolar sine wave-like excitations, which is the unique qualitative phenomena that are absent from that of a resistor, inductor, and capacitor [15]. Due to these phenomena, the resistance depends upon the past history of input current or voltage, which enables this device to function as a non-volatile memory.

Many scientists have investigated the memristor for possible applications in analog [17, 18] and digital information processing [19, 20]. Others have applied memristive devices for memory and logic applications [21–23].

Also, the great potential for exploiting the memristor for neural network applications had been claimed by many scientists [9, 15, 24], and [25]. Analog signal storing capability of the memristor is very important for this research area. Recently, Jo et al., [24] and Kund et al., [26] built memristors using Ag and Si in a sandwiched layer and reported their performances as analog memories. Also, Snider had presented a memristor-based self-organized network employing dedicated connections for inhibitory (negative) weighting [25].

In this paper, a general structure to implement synapses efficiently using memristors is addressed. Specifically, we propose a *memristor-based synapse* consisting of a four-memristor bridge circuit, and three transistors. *Strong* (large and wide) pulses are used for programming the synaptic weights, while *weak* (small or narrow) pulses having negligible effects on the memristor operating point are used as processing input signals.

The operating mechanism of our memristor-bridge weighting circuit is derived analytically and illustrated by computer simulations. The synaptic *weight programming* capability of the memristor bridge is demonstrated by implementing two well-known CNN templates using the TiO_2 memristor model.

In Sect. 2, we review the principle of the memristor and the TiO_2 memristor model. Our memristor bridge synapse is introduced in Sect. 3. The programming characteristics of our memristor bridge neuron are analyzed in Sect. 4. Simulation results and conclusions are given in Sects. 5 and 6, respectively.

2 Memristors and Memristive Devices

In the memristive device, the current and voltage relationship can be defined by

$$v = \left[M(x_1, x_2, \ldots, x_n)\right]i \tag{1}$$

where M of memristance and the x_i's are state variables,[1] v and i are voltage and current, respectively. The state variables $x_1, x_2, \ldots, x_n$ are defined by "n" differential equations where $n \geq 1$, called the associated state equations, as follow:

$$\frac{dx_k}{dt} = f_k[x_1, x_2, \ldots, x_n, i], \quad k = 1, 2, \ldots, m \tag{2}$$

Specifically, if the device possesses only a single variable x and it is a function of only the charge q, the device is called "ideal memristor" or simply "memristor". In this sense, the memristor [16] is a subset of memristive devices [15].

One of the "ideal memristor" is defined [27] by

$$v(t) = R(t)i(t) = \frac{d\varphi}{dt} \cdot \frac{dt}{dq} i(t) \tag{3}$$

where $\varphi(t)$ and $q(t)$ denote the *flux* and *charge*, respectively, at time t. Thus, the resistance can be interpreted as the *slope* at the operating point $q = q_Q$ at time t on the memristor φ–q curve. If the φ–q curve is nonlinear; the resistance will vary with the operating point.

Since the flux φ is defined by $\varphi(t) = \int_{-\infty}^{t} v(\tau)d\tau$, the resistance of the memristor, called the *memristance M*, can be controlled by applying a voltage or current signal across the memristor, where

$$R = M = \left.\frac{d\varphi}{dq}\right|_{(q_Q, \varphi_Q)} \tag{4}$$

In the TiO_2 memristor, a thin titanium dioxide (TiO_2) layer and a thin oxygen-poor titanium dioxide (TiO_{2-x}) layer are sandwiched between two platinum electrodes. The TiO_2 layer and the TiO_{2-x} layer are referred to as un-doped, and doped layers, respectively. When a voltage or current is applied to the device, the dividing line between the TiO_2 and TiO_{2-x} layers shifts as a function of the applied voltage or current. As a result, the resistance between the two electrodes is altered.

[1] At least one state variable must appear *explicitly* in the definition of the memristance.

Let D and w denote the thickness of the sandwiched area and the doped area (oxygen deficient area) in the $\mathrm{TiO_2}$ memristor, respectively, and let R_{ON} and R_{OFF} denote the resistances at high and low dopant concentration areas, respectively.

Then, the relation between the voltage and the current in (3) is given by

$$v(t) = \left(R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right) \right) i(t) \tag{5}$$

where $w(t)/D$ is the state variable x defined in (2). In $\mathrm{TiO_2}$ memristor [9], the state variable w is defined as a function of current i as

$$\frac{dw(t)}{dt} = \mu_V \frac{R_{ON}}{D} i(t) \tag{6}$$

where μ_v is the dopant mobility. This model is called a linear drift model since the velocity of the width is linearly proportional to the current.

The relationship between the *flux* and the *charge* of the $\mathrm{TiO_2}$ memristor is given by [9]

$$\varphi(t) = R_{off} \left\{ q(t) \left[1 + \frac{w_0}{D} \left(\frac{R_{ON}}{R_{OFF}} - 1 \right) \right] - \frac{\mu_v R_{ON}}{2D^2} \left(1 - \frac{R_{ON}}{R_{OFF}} \right) q(t)^2 \right\} + \varphi_0 \tag{7}$$

From (4) and (7), we obtain

$$M = \frac{d\varphi}{dq} = R_{off} \left\{ \left[1 + \frac{w_0}{D} \left(\frac{R_{ON}}{R_{OFF}} - 1 \right) \right] - \frac{\mu_v R_{ON}}{D^2} \left(1 - \frac{R_{ON}}{R_{OFF}} \right) q(t) \right\} \tag{8}$$

Observe that the memristance in (8) is a linear function of the charge $q(t)$, where the theoretical range of the φ, q and M of the $\mathrm{TiO_2}$ memristor are $\varphi = [0, 0.804]$ Weber, $q = [0, 1 \times 10^{-4}]$ Coulomb and $M = [16, 0.1]$ KΩ, respectively for the range $w = [0, D]$.

Differently from the linear drift model described above, nonlinear phenomenon appears often at the boundaries of nano-scale devices; with even the small voltage applied across nanometer devices, a large electric field is produced and therefore, the ion boundary position is moved in a non-linear fashion in nano-scale devices [11].

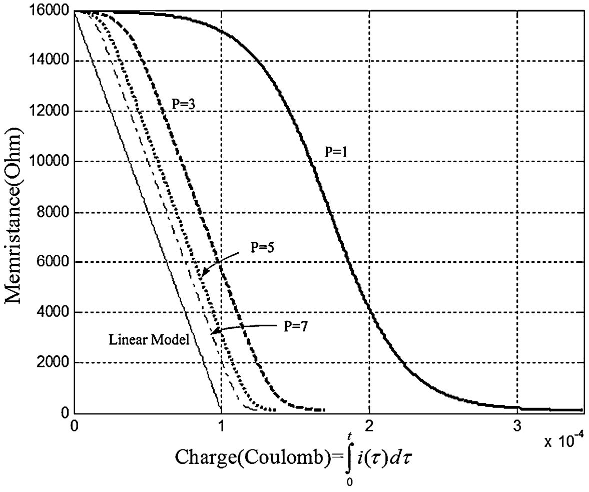
Several different types of nonlinear memristor models have been investigated [11, 28]. One of them is the window model in which the state equation is multiplied by window function $F_p(w)$, namely

$$\frac{dw(t)}{dt} = \mu_V \frac{R_{ON}}{D} i(t) F_p(w) \tag{9}$$

where p is an integer parameter and $F_p(w)$ is defined by

$$F_p(w) = 1 - \left(2\frac{w}{D} - 1 \right)^{2p} \tag{10}$$

Fig. 1 Relationship between memristance versus charge for the linear and the nonlinear models [11] of TiO2 memristors. As the integer p increases, the graphs tend to the linear model



This is called the nonlinear drift model or memristive model. Fig. 1 shows the graphs of the memristance vs charge of the linear and nonlinear models of memristors. As the number p becomes smaller, the nonlinearity increases. On the other hand, as the integer p increases, the model tends to the linear model.

3 Synaptic Multiplication via Memristor Bridge

Our memristor bridge synapse is a Wheatstone bridge-like circuit consisting of four identical memristors with the polarities indicated in Fig. 2. When a positive or a negative, strong pulse $V_{in}(t)$ is applied at the input, the memristance of each memristor is increased or decreased depending upon its polarity. For instance, when a positive pulse is applied as input, the memristances of M_1 and M_4 (whose polarities are forward-biased) will decrease. On the other hand, the memristances of M_2 and M_3 (whose polarities are reverse-biased) will increase. It follows that the voltage V_A at node A (with respect to ground) becomes larger than the voltage V_B at node B for a positive input signal pulse. In this case, the circuit produces a positive output voltage V_{out} representing a positive synaptic weight.

On the other hand, when a negative strong pulse is applied, the memristances are varied in the opposite direction and the voltage at node B becomes larger than that at node A. In this case, the circuit produces a negative output voltage V_{out} representing a negative synaptic weight.

3.1 Weighting of Input Signals via the Memristor Bridge

Suppose an input signal voltage v_{in} is applied to the memristor bridge circuit in Fig. 2 at time t. The input voltage will be divided via the well-known "*voltage-*

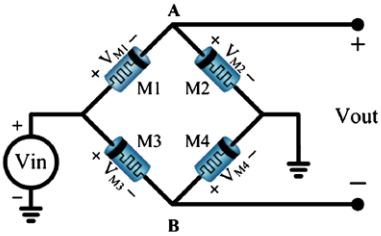


Fig. 2 Memristor bridge circuit. The synaptic weight is programmable by varying the input voltage v_{in}. The weighting (multiplication) of the input signal is also performed in this circuit

divider formula” as follows

$$v_{M1} = \frac{M_1}{M_1 + M_2} v_{in} \tag{11a}$$

$$v_{M2} = \frac{M_2}{M_1 + M_2} v_{in} = v_A \tag{11b}$$

$$v_{M3} = \frac{M_3}{M_3 + M_4} v_{in} \tag{11c}$$

$$v_{M4} = \frac{M_4}{M_3 + M_4} v_{in} = v_B \tag{11d}$$

where M_1, M_2, M_3, and M_4 denote the corresponding memristances of the memristors at time t, in Fig. 2. Note that the voltage divider formula for memristors is the same as that for resistors.

The output voltage V_{out} of the memristor bridge circuit is equal to the voltage difference between terminal A and terminal B; namely,

$$v_{out} = v_A - v_B = \left(\frac{M_2}{M_1 + M_2} - \frac{M_4}{M_3 + M_4} \right) v_{in} \tag{12}$$

Equation (12) can be rewritten as a relationship

$$v_{out} = \psi \times v_{in} \tag{13}$$

between a synaptic weight ψ and a synaptic input signal v_{in} where

$$\psi = \frac{M_2}{M_1 + M_2} - \frac{M_4}{M_3 + M_4} \tag{14}$$

Equations (13) and (14) define the synaptic weighting operation in the memristor bridge. Since the memristor bridge acts like a linear resistor network during short time intervals, the weighting operation is almost linear. The detailed analysis of the linearity in the synaptic weight programming will be carried out in the next section.

In contrast to our memristor bridge weighing (multiplication) circuit, multiplication in a typical analog multiplication circuit [7, 8] is usually implemented with at least eight transistors operating in a nonlinear and hence power-hungry regime. Therefore, significant nonlinearity is unavoidable in multiplication processing with conventional CMOS transistor circuits.

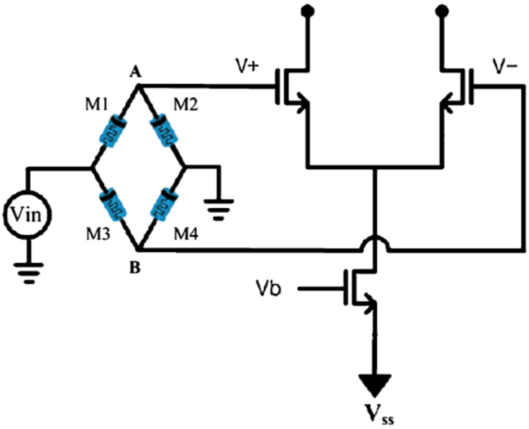


Fig. 3 Memristor bridge synaptic circuit. The memristor bridge on the *left performs* the weighting operation while the differential amplifier on the *right performs* the voltage to current conversion

4 Memristor-Bridge Neuron

Each neuron must add a set of weighted input signals from diverse sources. This is implemented in our memristor bridge neuron via summing input signals with current mode circuits. The differential amplifier with three transistors in Fig. 3 is the voltage-to-current converter which functions as a current source.

Figure 4 shows the complete circuit of our memristor-bridge neuron with multiple input synapses. All positive terminals of the input synapses are connected together, as are the negative terminals. The circuit at the bottom right of Fig. 4 is the cell biasing circuit that provides the DC bias voltage for the output.

The active load at the top of Fig. 4 is shared among all input synaptic circuits. The sum of all synaptic input signal currents appears at the output and is converted back to a *voltage* signal via a memristor load which operates at 7.2 kΩ. The transistor circuit as well as the memristor circuit operates based on voltage pulses, which enables the circuit to save power greatly.

Our memristor-bridge neuron can be used for any kind of neural networks, including *cellular neural networks* (CNN).

5 Weight Programming in Memristor Bridge Synapses

The synaptic weight *processing* introduced in the previous section was performed with very small or narrow pulses so that its effect on the change in memristance was negligible. In contrast, the pulses for synaptic weight *programming* must be strong (relatively large and wide) enough to change the charge operating point of the memristor.

When a positive programming pulse V_{in} is applied to the memristor bridge circuit in Fig. 2, the memristances M_2 and M_3 increase while the memristances M_1 and M_4 decrease. On the other hand, when a negative pulse V_{in} is applied, the memristances M_2 and M_3 decrease while M_1 and M_4 increase.

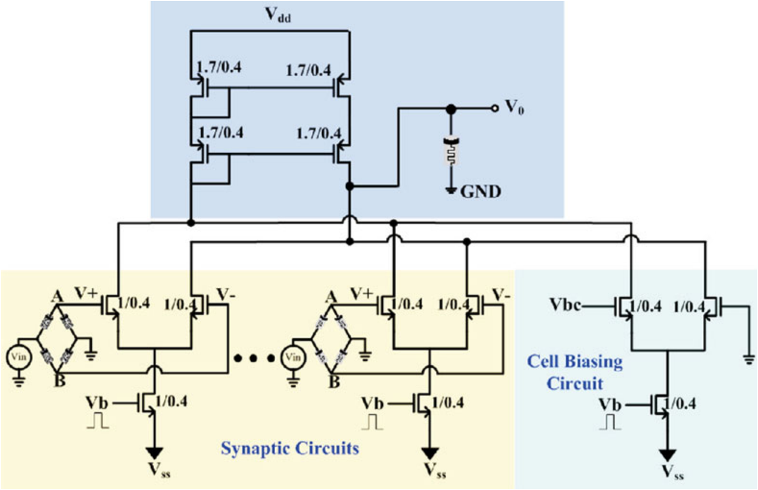


Fig. 4 Complete schematic of our memristor-bridge neuron. The synaptic circuits are located at the *left bottom* and the cell biasing circuit is at the *right bottom*. The current mirror at the *top acts* as an active load that is shared by all synapses

It follows that if the synaptic weight ψ is larger than 0, namely,

$$\frac{M_2}{M_1+M_2} - \frac{M_4}{M_3+M_4} > 0 \tag{15}$$

Our memristor bridge circuit represents as a *positive* synaptic weight. Rearranging (15), the condition for positive synaptic weight is as follow:

$$\text{Positive Synaptic Weight;} \quad \text{if} \quad \frac{M_2}{M_1} > \frac{M_4}{M_3}$$

Similarly, the conditions for negative weight, or zero synaptic weight, are given, respectively, as follow:

$$\text{Negative Synaptic Weight;} \quad \text{if} \quad \frac{M_2}{M_1} < \frac{M_4}{M_3}$$

$$\text{Zero Synaptic Weight;} \quad \text{if} \quad \frac{M_2}{M_1} = \frac{M_4}{M_3}$$

The state when the output is zero is henceforth called the *balanced state*. At the balanced state, the synaptic weight is zero.

Note that the synaptic weight programming signal and the synaptic input signal v_{in} for processing share the same input terminal in Fig. 2. The two different kinds of signals are discriminated by being assigned at different time slots. Very narrow pulses with negligible effect on memristance changes are used for input signals, while very strong pulses are used for programming synaptic weights.

6 Simulation

Computer simulations of our memristor bridge neuron in Fig. 3 were performed using the TiO_2 memristor model [9], where the initial doped-width of w_0 is assumed to be either $0.1D$ when $q = 0$ and $\varphi = 0$, or $0.9D$ when $\varphi = 0.64$ Weber and $q = 8 \times 10^{-5}$ Coulomb, which corresponds to the physically possible *maximum* (14.41 KΩ) and *minimum* (1.69 KΩ) memristances as in [9], respectively. Linearity in the synaptic weight programming of our memristor bridge synapses and the synaptic weight processing resulting from synaptic input signals were investigated via computer simulations. In addition, the feasibility of using our memristor bridge for implementing CNN templates has been examined.

The width of the synaptic weight processing pulse must be sufficiently narrow so that the memristances will not be affected. The simulations for the synaptic weight programming and the synaptic weight processing can be performed separately. For synaptic weight programming, the memristance variations are computed every 4×10^{-4} sec after application of a strong synaptic weight programming pulse. Each branch current in Fig. 2 is computed and during every 4×10^{-4} sec, followed by an update of all memristor charges. The charges are then used to compute the current memristance values via (8) of the TiO_2 memristor model.

After the memristance programming is completed, the memristors in the bridge circuit are replaced with resistors having identical resistances for the subsequent SPICE simulation. Synaptic weight processing of the combined resistor-CMOS circuit was performed using SPICE.

Two types of rectangular voltage pulses were used; for *synaptic weight programming*, we used pulses with fixed amplitude of 1 V and varying widths. For *synaptic weight processing*, pulses with a fixed width of 5 ns and varying amplitudes ranged $[-1.28, 1.28]$ V were used.

6.1 Linearity in Synaptic Weight Programming

Simulations to demonstrate the linearity in programming our memristor-bridge circuit have been performed. Figure 5(a) shows the changes in the *memristances* $M_1(t)$, $M_2(t)$, $M_3(t)$ and $M_4(t)$ as a function of time, obtained via computer simulations of the memristor bridge circuit in Fig. 2 with initial memristances, $M_1(0) = M_4(0) = 14.41$ KΩ, $M_2(0) = M_3(0) = 1.69$ KΩ. Figure 5(b) is the corresponding weight computed with the memristance values in Fig. 5(a) using (14). As shown in the figures, these numerically computed memristances in Fig. 5(a) and the corresponding weight in Fig. 5(b) are all very linear. In contrast, when the same voltage pulse as in Fig. 5 is applied to a non-bridge memristor circuit in parallel connection as shown in Fig. 6(a), the memristance changes are quite nonlinear as in Fig. 6(b). This reveals the fact that the linearity of Fig. 5 comes from the complementary connection of our memristor bridge circuit.

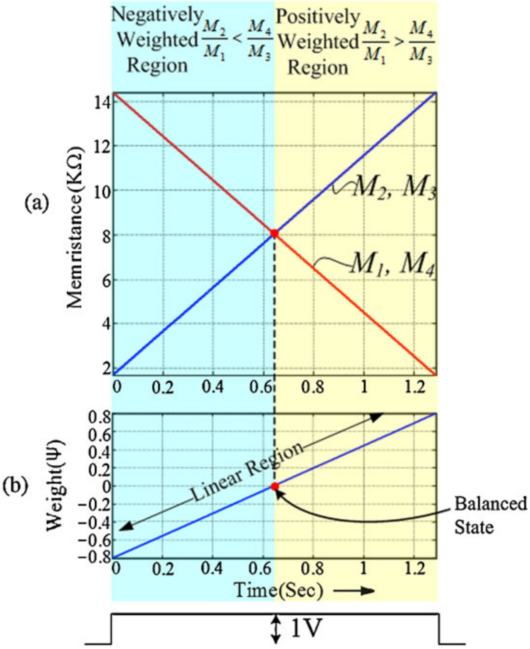


Fig. 5 Time variations of $M_1(t)$, $M_2(t)$, $M_3(t)$, $M_4(t)$, and $\psi(t)$with our memristor bridge circuit when a wide pulse is applied. The linear memristor model is assumed. The initial memristances are $M_1(0) = M_4(0) = 14.41$ KΩ, and $M_2(0) = M_3(0) = 1.69$ KΩ. (a) $M_1(t)$, $M_2(t)$, $M_3(t)$, and $M_4(t)$ (b) weight $\psi(t)$

The effect of the nonlinearity of memristor models to the performance of our memristor bridge circuit has also been investigated. Figure 7(b) shows the change of the memristances when the same simulation as in Fig. 5(a) is performed with the parallel memristor circuit as Fig. 6(a), where the nonlinear model of (10) with $p = 1$ is employed. The memristance changes as functions of time are highly nonlinear. In contrast, Fig. 7(a) shows the results with our memristor bridge circuit employing the nonlinear memristor model with $p = 1$ which is used for Fig. 7(b). Observe from this figure that the nonlinearity is reduced significantly. The graphs around the center are almost linear though some nonlinear regions can be seen at the end of the curves. Since weighting operations are performed only in the linear region at the center as indicated with a box, such nonlinear regions can also be avoided.

Since a similar linearity exhibited in a linear model can be obtained from a nonlinear model with our memristor bridge circuit when it is operated at the center area, all further simulations in this paper are performed only with the linear model to avoid redundancy.

6.2 Synaptic Weight (Multiplication) Processing

Simulations of synaptic weight (multiplication) processing were also performed. When an input signal is applied across the input terminal of the memristor-bridge

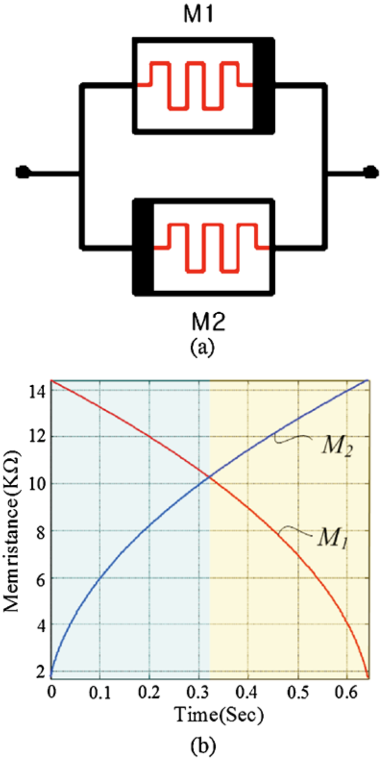
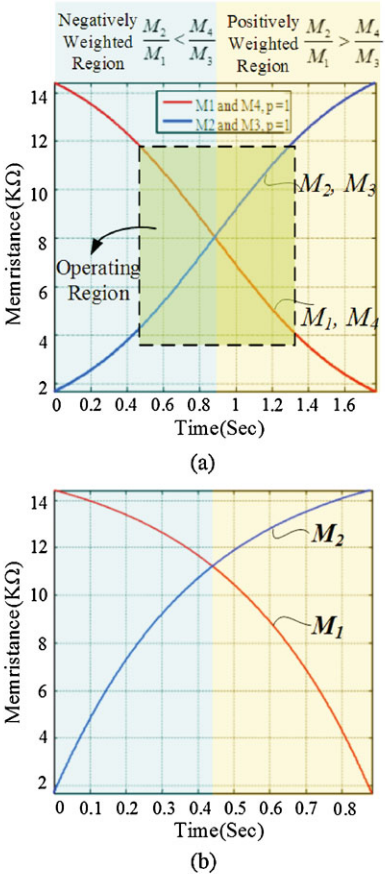


Fig. 6 Nonlinear variations of memristances about time when two linear memristors M_1 and M_2 are connected in parallel with opposite polarities. (**a**) two memristors connected in parallel, (**b**) time variations of memristances (Initial memristances $M_1(0) = 14.41$ KΩ, $M_2(0) = 1.69$ KΩ are assumed)

circuit, synaptic weighting is performed by a weighting factor created by the memristor bridge circuit. Simulations of the synaptic weight processing were performed in our memristor-bridge neuron with nine synaptic inputs. Various input signals were applied to a single synapse, while *zero* volts were applied to the remaining eight synapses. Figure 8(b) shows the linearity of the relationship between the input voltages, and the memristor bridge neuron output voltage, for various synaptic weights of the proposed circuit. To compare this result with the performance of a conventional multiplication circuit, the performance of the conventional multiplication (synaptic weight) circuit employed in the CNN [7, 8] is shown in Fig. 8(a). Note that the linearity of our memristor-based synaptic weighting is superior to that of the conventional multiplication circuit. The linearity of our memristor bridge synaptic circuit is due to the linear voltage assignment at the memristors and the operation at the center of the memristor dynamic range. The superiority of our memristor bridge circuit is demonstrated more clearly when the graph of Fig. 8 is redrawn with respect to the graph of output current variation ratio versus the synaptic weight change. The input voltage is fixed at 1 V. As seen in the Fig. 9, the current increment ratio of the conventional multiplier changes by almost 80 % over the synaptic weight range of $[-0.1, 0.1]$ while that of our memristor bridge circuit is almost constant.

Fig. 7 Time variations of $M_1(t)$, $M_2(t)$ when the nonlinear model of memristors $p = 1$ are employed. (**a**) improved linearity with the proposed memristor bridge circuit (**b**) nonlinear variation of the memristances when memristors are connected in parallel as in Fig. 6(a). Note that the time taken between (**a**) and (**b**) is different since the memristors in (**a**) are connected in series, while (**b**) are connected in parallel



6.3 Applications

Our memristor bridge neuron was used to implement a two-dimensional (2-D) image processing task. The basic architecture [7] of a feed forward CNN with zero **A** template was designed using our memristor bridge synapses as template elements. The circuit has nine synaptic inputs representing the **B** template and one bias voltage representing the threshold. The *average* template, and the *Laplacian* template, as shown in Fig. 10, were tested with our memristor bridge neurons. Note that Z is a bias template of the CNN and zero for the both cases. For the *average* template, the circuit of the CNN cells was simulated via SPICE, and the simulation with the input image (size: 16 × 16) in Fig. 11(a) was performed. The images shown in Figs. 11(b) and (c) are the results obtained by using the CNN software, and by using memristor bridge synapses, respectively. Also, our simulation of the *Laplacian* template was performed using the input image (size: 59 × 59) in Fig. 12(a). The images shown in Figs. 12(b) and (c) are results obtained by using the CNN software, and by using memristor bridge synapses, respectively. Although image processing using circuits

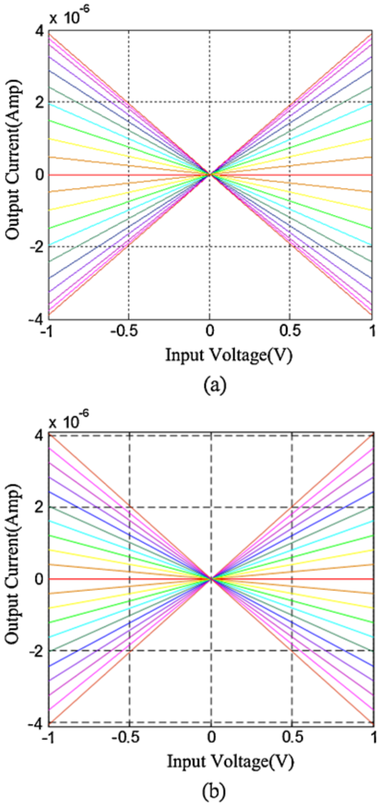


Fig. 8 Comparison of synaptic weight (multiplication) performance between (**a**) conventional multiplication circuit in [7, 8] and (**b**) memristor bridge synaptic circuit

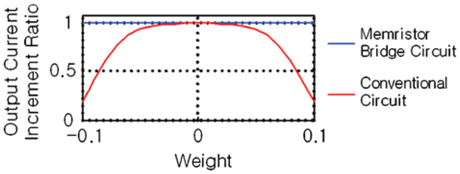


Fig. 9 Comparison of Fig. 8 with respect to the output current increment ratio versus the synaptic weight when the input is fixed at 1 V

in general has poorer performances than that obtained from CNN software, the result obtained from memristor bridge synapses is very close to the result obtained from CNN software: the average pixel difference between the results with the CNN simulator and that of the proposed circuit 0.89 % and 0.86 % for Figs. 11 and 12, respectively.

Circuit size is also the benefit of memristor bridge synapses. Since, the size of the memristor is projected to be less than 5 nm [9], the total size of the four memristors for each synapse would be much smaller than using CMOS transistors.

Ignoring the space occupied by memristors, a comparison of the number of required transistors between a memristor bridge neuron and a conventional CNN cell is shown in Table 1, where a full CNN cell with 18 synapses was assumed to be implemented. As Table 1 shows, a bridge synaptic implementation requires less than

$$\mathbf{B} = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 2 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \begin{array}{l} \\ A = 0 \\ Z = 0 \end{array} \qquad \mathbf{B} = \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} \begin{array}{l} \\ A = 0 \\ Z = 0 \end{array}$$

(a) (b)

Fig. 10 CNN templates tested for image processing with memristor bridge synapses (**a**) Average template (**b**) Laplacian template

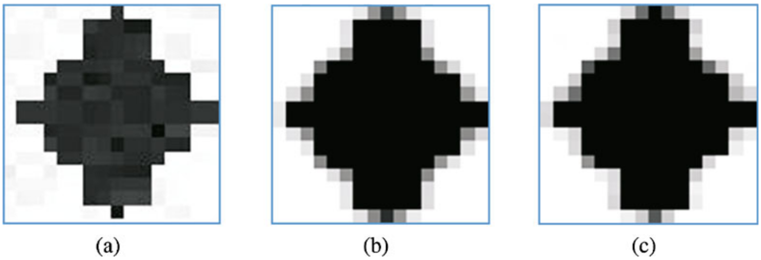

(a) (b) (c)

Fig. 11 Comparison of image processing performance using the average template. (**a**) Input image, (**b**) Processing implemented with CNN software (**c**) Processing implemented with memristor bridge synapses representing B template

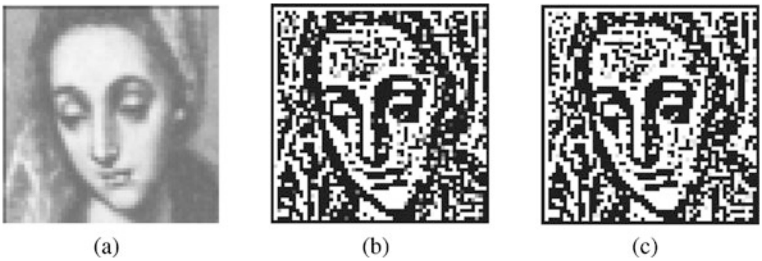

(a) (b) (c)

Fig. 12 Comparison of image processing performance using the Laplacian template. (**a**) Input image, (**b**) Processing implemented with CNN software. (**c**) Processing with memristor bridge synapses representing the B-template

Table 1 Comparison of the number of transistors needed to implement our memristor-based neuron and a conventional CNN cell [7, 8]

	Memristor Bridge neuron	Conventional CNN cell [7, 8]
Number of Transistors for a neuron	61	140

half of the number of transistors in a conventional CNN cell implementation. The size reduction would be even more impressive for implementing larger "$N \times N$ CNN templates", where $N \geq 3$.

Another benefit of our circuit comes from small power consumption due to the pulse-based operation of our memristor synapse. Figure 13 shows the variations of

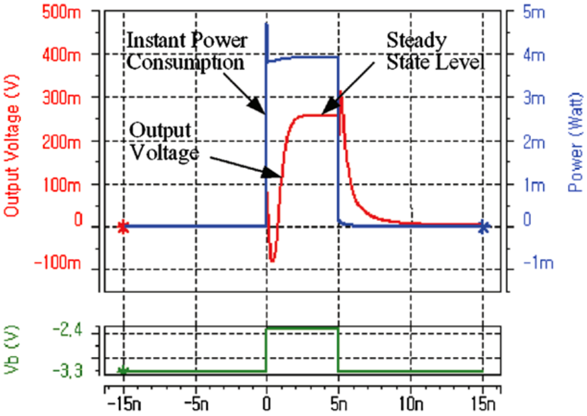


Fig. 13 Variations of the output voltage and the instant power consumption of our memristor bridge synapse circuit when all the signal input voltages and current source biasing voltages in Fig. 4 are given in pulse form

output voltage and instant power consumption when input voltage pulse (1 V, 5 ns,) and current source biasing pulse V_b ranged $[-3.3, -2.4]$ are presented at all the inputs of memristor bridge synapses and their current source biasing transistors, respectively. All the synaptic weights are programmed with 0.01. The periods and widths of the input signal pulses and those of current source biasing pulses are all identical. Observe that correct output voltages could be obtained if the output voltage is sampled after 2 ns from the time when a current source biasing voltage pulse V_b is given. Also, observe that the power consumption occurs during only the pulse width period. This fact shows the evidence of power saving with the proposed circuit though the average power consumption depends upon the frequency that the pulses are given to the circuit.

7 Conclusion

In this paper, we proposed a memristor bridge synaptic circuit for analog weighting and signal processing applications that could significantly outperform conventional approaches with respect to size and power consumption.

Our memristor bridge circuit is composed of four identical memristors. A weight programming was performed by applying strong positive or negative input pulses with the pulse width varying over the range [0, 1 ms] at amplitude of 1 V. The weight processing was performed on input signals ranging over the range $[-1.28\text{ V}, 1.28\text{ V}]$. Circuit simulation in which the TiO_2 memristor model is employed showed that our memristor synapse exhibits excellent linearity over both positive and negative weight ranges.

A neuron based on memristor bridge synapses was also proposed. All synapses were combined with an active load circuit and all the output of synapses were summed in current mode. The performance of memristor bridge synapses was investigated via two CNN templates (*average template* and *Laplacian template*). Circuit simulations using SPICE showed processing results that were very close to the ideal result obtained via software processing.

Finally, we remark that the two words "memristor" and "memristive device" are models, just as all other equations describing other devices, e.g., the "Ebers-Moll equations" of transistors are models. Since the term "memristor" as defined in 1971, and the term "memristive device " as defined in 1976 , share the same fingerprints, namely, a pinched hysteresis loop (under bipolar sine-wave like excitations) whose area shrinks with frequency, a unique circuit phenomenon completely different from the other three basic circuit elements (resistor, inductor, capacitor), Chua has proposed recently in [27] that there is no need to use two different words for the new family devices that behaves similarly. Hence, he proposed to use only one word (memristor) for both "memristor circa 1971", and "memristor circa 1976", to avoid clutter. In the few occasions where one may wish to refer to the "memristor circa 1971" specifically, Chua had suggested to simply add the adjective "ideal" to "memristor", namely, "ideal" memristor.

References

1. The Scientific American Book of the Brain. Scientific American, New York (1999)
2. Chua, L.O., Yang, L.: Cellular neural networks: theory. IEEE Trans. Circuits Syst. **35**(10), 1257–1272 (1988)
3. Chua, L.O., Yang, L.: Cellular neural networks: applications. IEEE Trans. Circuits Syst. **35**(10), 1273–1290 (1988)
4. Roska, T., Chua, L.O.: The CNN universal machine: an analogic array computer. IEEE Trans. Circuits Syst. II **40**(3), 163–172 (1993)
5. Kim, H., Son, H., Roska, T., Chua, L.O.: Optimal path finding with space- and time-variant metric weights with multi-layer CNN. Int. J. Circuits Theory Appl. **30**, 247–270 (2002)
6. Kim, H., Son, H., Roska, T., Chua, L.O.: High-performance Viterbi decoder with circularly connected 2-D CNN unilateral cell array. IEEE Trans. Circuits Syst. I **52**, 2208–2218 (2005)
7. Domínguez-Castro, R., et al.: A 0.8-μm CMOS two-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage. IEEE J. Solid-State Circuits **32**(7), 1013–1026 (1997)
8. Cruz, J.M., Chua, L.O.: A 16 × 16 cellular neural network universal chip: the first complete single-chip dynamic computer array with distributed memory and with gray-scale input-output. Analog Integr. Circuits Signal Process. **15**, 227–237 (1998)
9. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
10. Shin, S., Kim, K., Kang, S.M.: Compact models for memristors based on charge-flux constitutive relationships. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **29**(4), 590–598 (2010)
11. Joglekar, Y.N., Wolf, S.J.: The elusive memristor: properties of basic electrical circuits. Eur. J. Phys. **30**(4), 661–675 (2009)
12. Kavehei, O., Iqbal, A., Kim, Y.S., Eshraghian, K., Al-Sarawi, S.F., Abboti, D.: The fourth element: characteristics, modelling and electromagnetic theory of the memristor. Proc. R. Soc. A **466**(2120), 2175–2202 (2010)
13. Wang, F.Y.: Memristor for introductory physics (Aug. 4, 2008) arXiv:0808.0286v1 [physics. Class-ph]
14. Argall, F.: Switching phenomena in titanium oxide thin films. Solid-State Electron. **11**, 535–541 (1968)
15. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)

16. Chua, L.O.: Memristor-the missing circuit element. IEEE Trans. Circuit Theory **CT-18**(5), 507–519 (1971)
17. Itoh, M., Chua, L.O.: Memristor cellular automata and memristor discrete-time cellular neural networks. Int. J. Bifurc. Chaos (IJBC) **19**(11), 3605–3656 (2009)
18. Kim, H., Sah, M.P., Changju, Y., Chua, L.O.: Memristor-based multilevel memory. In: 12th International Workshop on Cellular Nanoscale Networks and Their Appl. (CNNA), Feb. 2010
19. Xia, Q., et al.: Memristor-CMOS hybrid integrated circuits for reconfigurable logic. Nano Lett. **9**(10), 3640–3645 (2009)
20. Borghettil, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R., Williams, R.S.: Memristive switches enable stateful logic operations via material implication. Nat. Lett. **464**(8), 873–876 (2010)
21. Aono, M., Hasegawa, T.: The atomic switch. Proc. IEEE **98**(12), 2228–2236 (2010)
22. Kozicki, M.N., Gopalan, C., Balakrishnan, M., Mitkova, M.: A low-power nonvolatile switching element based on copper-tungsten oxide solid electrolyte. IEEE Trans. Nanotechnol. **5**(5), 535–544 (2006)
23. Waser, R., Aono, M.: Nanoionics-based resistive switching memories. Nat. Mater. **6**, 833–840 (2007)
24. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. 1297–1301 (2010)
25. Snider, G.: Self-organized computation with unreliable, memristive nanodevices. Nanotechnology **18**(36), 1–13 (2007)
26. Kund, M., Beitel, G., Pinnow, C.U., Röhr, T., Schumann, R., Symanczyk, R., Ufert, K.D., Müller, G.: Conductive bridging RAM (CBRAM): An emerging non-volatile memory technology scalable to sub 20 nm. IEDM Tech. Dig. 754–757 (2005)
27. Chua, L.O.: Resistance switching memories are memristors. Appl. Phys. A **102**, 765–783 (2011)
28. Pickett, M.D., Strukov, D.B., Borghetti, J.L., Yang, J.J., Sinder, G.S., Stewart, D.R., Williams, R.S.: Switching dynamics in titanium dioxide memristive devices. J. Appl. Phys. **106**, 074508 (2009)

Cellular Nonlinear Networks with Memristor Synapses

Fernando Corinto, Alon Ascoli, Young-Su Kim, and Kyeong-Sik Min

Abstract Cellular Nonlinear/Nanoscale Networks (CNNs) that can provide parallel processing in massive scale are known to be suitable for neuromorphic applications such as vision systems. In CNNs, synaptic weights can be calculated by digital or analog multiplications. Though conventional CMOS digital circuits can be used in calculating these multiplications for CNN applications, they occupy very large area and consume a large amount of power, especially when multiplications should be calculated in parallel in massive scale. On the other hand, analog circuits seem to be very attractive for calculating multiplications for CNN applications. One possible approach is to multiply the input current by the programmable resistance of a memristor before applying the resulting voltage to a differential pair for the final voltage-to-current conversion. In this chapter we introduce some analog circuits for CNN applications that use the resistance of a memristor in calculating multiplications. In addition we discuss memristor models and some practical problems in CNN circuits that should be resolved using analog memristor-based implementations.

1 Introduction

The current-controlled *ideal memristor* is a passive bipole linking charge $q(t)$ and flux $\varphi(t)$ through a nonlinear relation, i.e. $\varphi(t) = \varphi(q(t))$. It follows that voltage

F. Corinto (✉)
Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
e-mail: fernando.corinto@polito.it

A. Ascoli
Technische Universität Dresden, Mommsenstraße 12, 01062 Dresden, Germany
e-mail: alon.ascoli@tu-dresden.de

Y.-S. Kim · K.-S. Min
School of Electrical Engineering, Kookmin University, Seoul, South Korea

Y.-S. Kim
e-mail: gongja85@kookmin.ac.kr

K.-S. Min
e-mail: mks@kookmin.ac.kr

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_13,

$v(t)$ depends upon current $i(t)$ through

$$v(t) = \frac{d\varphi(t)}{dt} = \frac{d\varphi(q(t))}{dq}\frac{dq(t)}{dt} = M\big(q(t)\big)\, i(t), \tag{1}$$

where $M(q) = \frac{d\varphi(q)}{dq}$ is the memristance (i.e. memory-resistance) of the bipole, i.e the resistance of the memristor depends upon the time history of the current flowed through it. This explains the memory capability of the memristor, theoretically envisioned by Chua in 1971 [1] and later classified by Chua and Kang in 1976 as the simplest element from a large class of nonlinear dynamical systems endowed with memristance, the so called memristive systems [2].

In [2] a memristive system (or memristor system[1]) is a nonlinear dynamical two–terminal circuit element defined by the following differential-algebraic system of equations:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}\big(\mathbf{x}(t), u(t)\big), \tag{2}$$

$$y(t) = \mathbf{g}\big(\mathbf{x}(t), u(t)\big)u(t), \tag{3}$$

where[2] $\mathbf{x} \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}$ refers to the input, $y \in \mathbb{R}$ describes the output, $\mathbf{f}(\mathbf{x}, u) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ stands for the *state evolution function*, while $\mathbf{g}(\mathbf{x}, u) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$, denotes the *memductance* (*memristance*) if input u is in voltage (current) form.

Since 2008, when its existence at the nano-scale was certified at Hewlett-Packard (HP) Labs [3], the memristor has attracted a strong interest from both industry and academia for its central role in the set up of novel integrated circuit (IC) architectures, especially in the design of high-density non-volatile memories [4], programmable analog circuitry [5], neuromorphic systems [6], and logic gates ([7, 8]).

Several applications of memristors such as memory, analog circuits, digital circuits, etc. have been intensively studied. Among these applications, neural network circuits with memristors can be considered very promising, because memristors can be used in synaptic circuits emulating the excitation-dependent change in synaptic coupling factors in neuronal networks.

One of key advantages of neural network systems is that massive parallelism can be more easily implementable thus the energy efficiency in neural network systems can be better than in conventional von Neumann systems that have been a theoretical ground of high-speed digital systems since over fifty years [9]. In addition, the Cellular Nonlinear/Nanoscale Network (CNN) paradigm [10–14] permits to overcome limits of VLSI neural network implementations. In CNNs local connectivity is realized by means of synapses whose weights are multiplied in the digital or analog domain by external excitations to produce signals to be transferred to neighboring

[1] In the following memristive systems are referred to as memristor systems, whereas the term ideal memristor is used for systems described by (1).

[2] For the sake of brevity the explicit time dependency is dropped where it is not strictly necessary.

cells. Though conventional CMOS digital multiplier circuits can be used in CNN applications, they occupy considerable layout area and consume large amount of power, especially when multiplications should be conducted in parallel in massive scale. Recently, new and simple analog multiplication circuits were developed by H. Kim et al. [15, 16], where the input signal currents are multiplied by the programmable resistances of memristors.

This chapter aims at introducing analog circuits for CNN applications that use memristance in calculating multiplications. The development of innovative strategies for the design of memristor-based CNNs (and general–purpose electronic systems as well) requires the development of circuital and mathematical memristor models and their implementation into a software package for computer aided integrated circuit design.

Several mathematical models ([3, 6, 17–22], to name but a few) for the memristor nano-structures were proposed in the literature. A good model should be as general as possible, i.e. it should be able to capture the memristor dynamics of a large number of nano-films. In this respect the Boundary Condition Memristor (BCM) model, recently-introduced in [20], was developed so as to meet this generality requirement. In fact the distinctive feature of the BCM model is the adaptability of the nano-device behavior at boundaries. In particular, the model makes use of adaptable[3] threshold voltages v_{th0} and v_{th1}, respectively defining[4] the magnitude of the limit value the input voltage (i.e. the voltage drop across the memristance) needs to cross after its negative-to-positive and positive-to-negative sign reversal before the memristor state may be released from its lower and upper bound.

It is straightforward to establish an optimization procedure, which, on the basis of observed data, sets the most suitable values for threshold voltages, i.e. those values, let us identify them as v^*_{th0} and v^*_{th1}, minimizing the mean squared error between observed and modeled data. This enables the BCM to stand out over other models available in the literature for the larger number of detectable dynamics, despite the extreme simplicity of the window function embedded into the state equation (when the state variable lies within its two bounds its time evolution is governed by the basic linear dopant drift model [3]). It is noteworthy that the class of detectable dynamics include not only all the behaviors observed in the HP memristor [3], but also phenomena exhibited by various other nano-structures where memristor behavior arises from distinct physical mechanisms ([25–28]). In order to enable the BCM model to support various neural learning rules, we recently developed a generalized version [29]—the so called Generalized Boundary Condition Memristor (GBCM)

[3] Note that by defining a time evolution rule for the threshold voltages, it was recently demonstrated [23] that an adaptable threshold voltage-based version of the memristor model from [6] may explain the Suppression Principle [24] of the Spike-Timing-Dependent-Plasticity (STDP) Rule [6], which may occur in the case of triplet spikes.

[4] Throughout the paper, unless stated otherwise and without loss of generality, we assume that the doped layer is spatially located to the left of the un-doped layer along the horizontal extension of the nano-film [20], and in this case we assign a value of +1 to the memristor polarity coefficient η (see Eq. (6)).

model—in which the activation threshold property characterizing the boundary behavior in the original BCM model [20] is extended to the whole admissible range of the state variable, thus allowing the modulation of the rate of evolution of the memristor dynamics in sub- and supra-activation threshold regimes.

In this chapter, we introduce some analog circuits for CNN applications that use memristance in calculating multiplications. In addition, we discuss memristor models and some practical image processing tasks which may be performed in memristor–based CNNs.

2 Brief Review of Memristor Models

Various memristor circuit models have been proposed in the literature. A large number of models assumes that the control waveform is in current form (the voltage v-current i relationship is expressed by (1)), views the memristance as the series between two variable resistances, associated with the insulating and conductive layers of the nano-film, and sets the width w of the conductive layer, normalized with respect to the entire length D of the device, as the state $x = \frac{w}{D} \in [0, 1]$ of the system. The linear drift model from Williams [3], where the time derivative of the state is proportional to the input waveform in current form, is valid under the assumption of low electric fields, and does not take into account the boundary behavior.

In the nonlinear drift models from [17, 18] and [30] the rate of change of the state is proportional to the product between the input waveform in current form and a window function accounting for nonlinear dynamical behavior and imposing suitable boundary conditions.

In Joglekar's model [17] the window function is defined as $f_J(x) = 1 - (2x - 1)^{2p}$ $(p \in \mathbb{Z}_+)$. Such window describes the suppression of dopant drift close to the extremities, but is not vertically scalable (i.e. its maximum value may not be up- or down-shifted) and introduces the so-called "terminal-state problem" [30], since if the state is at either of its two bounds it may not leave it for any subsequent time instant. Note that for $p = 1$ Joglekar's window is a scaled (by a factor of 4) version of yet another window previously derived by Strukov in [3], i.e. $f_S(x) = x(1 - x)$. Benderli [31] presented a circuit realization of Strukov's model [3], in which the terminal-state problem was resolved through the use of comparators and logic gates allowing the emulation of the state clipping at or release from either bound.

In Biolek's model the window function depends on both state x and input current i, being defined as $f_B(x, i) = 1 - (x - stp(-i))^{2p}$, where $stp(x) = 1$ for $x \geq 0$ and 0 otherwise $(p \in \mathbb{Z}_+)$. Such window resolves the "terminal-state problem", but has limited scalability (in particular, its maximum value may not exceed $+1$ [30]). PSpice implementations of Joglekar's and Biolek's models are reported in [18].

In the versatile model proposed by Prodomakis [30] the window function $f_P(x) = j(1 - ((x - 0.5)^2 - 0.75)^p)$ has two control parameters j and p lying in $\mathbb{R}_+$ and is vertically scalable, i.e. $0 \leq \max\{f_P(x)\} \gtreqless 1$. A PSpice version of such model may be easily derived by modifying the PSpice code [32] file available in [18].

Another model endowed with a PSpice circuit implementation was developed by Cserey [33]. In this model the state evolution function in Strukov's model [3] was augmented with an additive state-dependent linear term to resolve the "terminal-state problem".

One of the finest mathematical descriptions of memristor behavior is credited to Shin and Kang [19], which proposed a general model where the control waveform may be in either current or voltage form and the state is defined as the memristance. Their model, from which the charge-flux relationship of the memristor under modeling may be easily extracted, may be suitably tuned through the introduction of a window function depending on the memristor charge.

Kavehei [34] proposed a memristor model based upon the specification of a piecewise-linear charge q-flux φ relationship. Its PSpice implementation is based on Chua's [1] first circuit realization of a memristor through a type-1 memristor-resistor mutator.

An interesting model was presented in [35] to explain the memristor behavior of nanoparticle assemblies.

The nonlinear dependence of the time derivative of the state on the input signal is taken into account in Lehtonen's model [36], inspired by the experimental work from [37], where the current is related to the voltage by means of a rectifying exponential function in the off state (as in a diode) and of a sinh function in the on state (typical of electron tunneling). This model, where the control waveform is in voltage form, was implemented in PSpice to describe the neighborhood connections among CNNs ([10, 11, 38]).

An even more nonlinear function of the input governs the state equation in the voltage-controlled model from Poikonen [39], which studied the transition between non-programming and programming phases in memristor devices.

In the memristor emulator circuit from [40], used as basic building block of a 4-memristor bridge synapse for neuromorphic applications, the memristance, modeled by the input impedance of an operational amplifier, is made proportional to the time integral of the memristor current by constraining the voltage at one of the input terminals of the amplifier to be the analogue multiplication between the voltage across a resistor, proportional to the memristor current, and the voltage across a capacitor, proportional to the time integral of the memristor current.

In [41] Strukov and Williams demonstrated the exponential relationship between drift velocity and local electric field. Since this discovery a number of models have been introduced to support threshold-activated state dynamics.

Among them, one which merits mention, is the physics-based Pickett's model from [21], in which the dependency of the rate of change of the state on the current-form input is strongly nonlinear. In such model the memristor is seen as the series between a low resistance associated with the conductive layer of the nano-film and Simmons' electron tunneling barrier [42], whose width is chosen as the system state. A PSpice version of Pickett's model was presented in [43].

More recently Kvatinski developed a simplified version of Pickett's model [21] and named it as ThrEshold Adaptive Memristor (TEAM) model [22]. In such model for input current magnitude below a certain adaptable threshold no state change occurs, otherwise the state evolution rule may be tuned to the memristor element under modeling through specification of an appropriate set of control parameters and of suitable window and memristance functions. The PSpice architecture of the TEAM model is similar to the one originally presented in [19].

Another activation-type state model, where the state variable expresses the memristance and the control signal is in voltage form, embedded in the PSpice software program [44], enabled the reproduction of the adaptive behavior of a unicellular organism named amoeba through a simple memristor-based oscillator [45].

Another interesting model with threshold-activated state dynamics was proposed in [46] to explain Spike-Timing-Dependent-Plasticity (STDP) in neural synapses.

Most of these mathematical models have been classified in [47]. Another insightful discussion on the models available in the literature was recently published in [48], where a novel model inspired from Simmons' electron tunneling theory [42], endowed with programming threshold capability and PSpice circuit implementation, was also proposed.

The Boundary Condition Memristor (BCM) model is a simple yet accurate boundary condition-based mathematical model for memristor nano-structures made up of two layers with different conductivity levels, whose longitudinal extensions depend on the time history of the input. In comparison with the classical BCM model [20], the generalized version [29] is augmented with threshold-activated state dynamics [41].

Recently, in [49], assuming Pickett's model [21] as reference for comparison, various memristor models, including Biolek's, the TEAM and the BCM models, were first compared on the basis of the ability to reproduce (after an optimization process) the dynamics of the reference model in a particular simulation scenario, and secondly employed in a couple of memristor-based circuits to investigate the variance in the nonlinear dynamical behaviors they give rise to. The latter study revealed the model-dependency of the dynamics of memristor-based circuits, and thus raised a warning against a blind faith in the memristor models. The results of this investigation point out the necessity to develop an ad-hoc mathematical model for each physical memristor so as to explore its full potential and unfold its unique properties.

Section 2.1 briefly summarizes the recently proposed GBCM model and its PSpice-based circuit [29] (the PSpice emulator of the classical BCM is reported in [50]).

2.1 GBCM Model and Its Circuit Implementation

Let R_{on} and R_{off} stand for the on and off resistances of a memristor nano-film. The memristor state variable x is chosen as the length $w(t)$ of the conductive layer of the nano-film normalized with respect to the entire longitudinal extension D of the nano-film (i.e. $x = \frac{w(t)}{D} \in [0, 1]$). Denoting memristor current and voltage as i and v

respectively, the state-dependent input-output algebraic relationship of the GBCM model is expressed by

$$i(t) = W\big(x(t)\big)\, v(t), \tag{4}$$

where $W(x(t))$ describes the state-dependent memductance, expressed by

$$W\big(x(t)\big) = \frac{G_{on} G_{off}}{G_{on} - \varDelta G x(t)}, \tag{5}$$

with $G_{on} = R_{on}^{-1}$, $G_{off} = R_{off}^{-1}$, while $\varDelta G = G_{on} - G_{off}$.

The state equation of the GBCM model is defined as

$$\frac{dx(t)}{dt} = \eta\, k\ W\big(x(t)\big)\, v(t)\, f\big(x(t), \eta\, v(t), v_{th0}, v_{th1}, v_{t1}, v_{t2}, a, b\big), \tag{6}$$

where $k \in \mathbb{R}$ is a constant depending on physical properties of the memristor (its dimensions are C^{-1}), $\eta \in \{-1, +1\}$ is a coefficient denoting the polarity of the nano-device, while $f(x(t), \eta\, v(t), v_{th0}, v_{th1}, a, b) \in \{0, a, b\}$, a switching window function specifying boundary behavior and state evolution rates in sub- and supra-threshold regimes [41], is expressed as

$$f(x, \eta\, v, v_{th0}, v_{th1}, v_{t0}, v_{t1}, a, b) = \begin{cases} b & \text{if } C_1 \text{ or } C_2 \text{ holds,} \\ 0 & \text{if } C_3 \text{ or } C_4 \text{ holds,} \\ a & \text{if } C_5 \text{ holds ,} \end{cases} \tag{7}$$

where tunable conditions C_n $(n = 1, 2, 3, 4, 5)$ are mathematically described by

$$\begin{aligned}
C_1 &= \Big\{\big(x(t) \in (0,1) \text{ and } \big(\big(\eta v(t) > v_{t0}\big) \text{ or } \big(\eta v(t) < -v_{t1}\big)\big)\big)\Big\}, \\
C_2 &= \Big\{\big(x(t) = 0 \text{ and } \eta v(t) > v_{th0}\big) \text{ or } \big(x(t) = 1 \text{ and } \eta v(t) < -v_{th1}\big)\Big\}, \\
C_3 &= \Big\{x(t) = 0 \text{ and } \eta v(t) \le v_{th0}\Big\}, \\
C_4 &= \Big\{x(t) = 1 \text{ and } \eta v(t) \ge -v_{th1}\Big\}, \\
C_5 &= \Big\{\big(x(t) \in (0,1) \text{ and } \big(\big(\eta v(t) \le v_{t0}\big) \text{ and } \big(\eta v(t) \ge -v_{t1}\big)\big)\big)\Big\},
\end{aligned}$$

where $v_{th0} \in \mathbb{R}_+$, $v_{th1} \in \mathbb{R}_+$ represent the input thresholds at boundaries, $v_{t0} \in \mathbb{R}_+$, $v_{t1} \in \mathbb{R}_+$ define the state activation thresholds, ($v_{t0} \le v_{th0}$ and $v_{t1} \le v_{th1}$), while a and b are constants modulating the state evolution rate under sub- and supra-threshold excitation ($b \in \mathbb{R}_+$, $a \in \mathbb{R}_{0,+}$, $a < b$).

The PSpice implementation of the GBCM model, based on voltage-controlled voltage-switches, is reported in [29].

Note that it is possible to develop a more realistic implementation of the PSpice circuit from [29] by implementing the voltage-controlled voltage switches with suitable combinations of Complementary-Metal-Oxide-Semiconductor (CMOS) transistors.

3 Memristor Synaptic Weighting Circuits for Neuromorphic Applications

Classical Von Neumann architectures are known to be great in number crunching, nevertheless, they struggle with tasks like face recognition, real-time navigation control, object segmentation and depth perception. The parallelism can be realized by memristor cross-point array that can be stacked layer by layer as 3-dimensional structure. S.H. Jo et al. recently developed the synapse programming circuit that mimics Spike Timing Dependent Plasticity (STDP) using the digital pulse generator circuits and memristor array [51]. According to the generated pulse sequence, the synaptic weight that is stored at a cross-point memristor cell can be programmed to raise or lower its value in accord with the STDP rule for neuron-to-neuron interactions.

Physical implementations of neural networks exploit local connections among the cells, i.e. they are based on the Cellular Nonlinear/Nanoscale Networks (CNNs) architecture [10–14], where synaptic weights are multiplied in the digital or analog domain by external excitations to produce signals to be transferred to neighboring neuronal cells. Though the conventional CMOS digital multiplier circuits can be used in CNN applications, they occupy considerable layout area and consume a large amount of power, especially when multiplications should be conducted in parallel in massive scale. Recently, new and simple analog multiplication circuits were developed by H. Kim et al. [15, 16], where the input signal currents are multiplied by the programmable resistances of memristors. We can think in this multiplication that the input current acts as a multiplicand and the programmable memristance can be regarded a multiplier.

Figure 1 shows the memristor-based synaptic weighting circuit that was proposed for CNN applications by H. Kim et al [16]. The circuit is composed of nine weighting circuits, nine differential pairs, and one shared current mirror among nine circuits. In Fig. 1, I_{IN1} means the input current to the first weighting circuit. Similarly, I_{IN2} is the input current to the second one. M_{S1}, M_{S2}, M_{S3}, M_{S4} are memristors that act as the control switches to change polarity of multiplication. In this paper, M_S is called the switching memristor to distinguish it from the weighting memristor, M_W. If M_{S1} and M_{S4} are Low Resistance State (LRS) and M_{S2} and M_{S3} are High Resistance State (HRS), the polarity of multiplication is plus. When M_{S1} and M_{S4} are HRS and M_{S2} and M_{S3} are LRS, the polarity becomes minus. After the polarity is decided, I_{IN1} is applied to M_{W1} that represents the synaptic weight. It means that I_{IN1} is multiplied by M_{W1} thus we can obtain V_{A1B1} $(= V_{A1}–V_{B1})$ that corresponds to the multiplication result. V_{A1B1} that is the differential input is applied to the gates of M_{N1} and M_{N2} and it is converted to the output current that is delivered to the load resistor, R_L. This analog multiplication can be done easily by the voltage-current relationship of memristors. In Fig. 1, nine multiplications can be done in parallel and the output currents of nine multiplications that are obtained at the same moment can be added using the shared current mirror, as shown in Fig. 1. Here, nine weighting circuits with nine M_Ws are connected to nine differential pairs, respectively. If we look at the second weighting circuit and differential pair, I_{IN2} goes through M_{W2} to

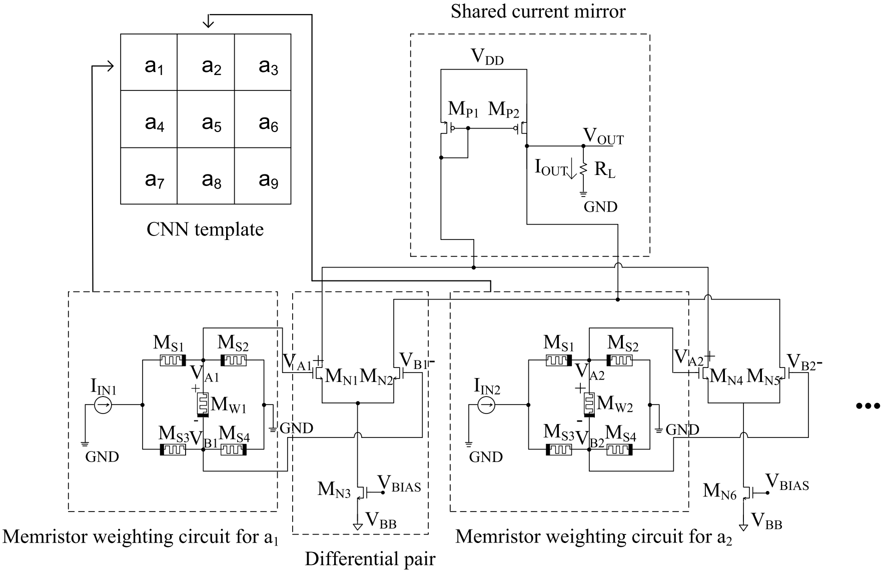


Fig. 1 The conventional synaptic weighting circuit proposed in [16] with the memristor weighting circuits and the differential pairs. This circuit has nine memristor weighting circuits, nine differential pairs, and one shared current mirror. Here V_{BB} is a negative voltage lower than GND

make a voltage drop between V_{A2} and V_{B2}. The second multiplication result as large as V_{A2B2} (= V_{A2}–V_{B2}) is applied to the gates of M_{N4} and M_{N5}. V_{A2B2} is converted to the output current to be added to the first output current from V_{A1B1} using the shared current mirror. Similarly, more multiplications can be done simultaneously and their output currents can be added by merging the differential-pair circuits and sharing the current mirror that has M_{P1} and M_{P2}, as shown in Fig. 1. M_{N3} and M_{N6} are the tail current biasing circuit that is connected to V_{BB}. Here V_{BB} should be a negative voltage that is lower than 0 V. V_{BIAS} is a biasing voltage to the gates of M_{N3} and M_{N6}. The simple analog multiplication circuit of Fig. 1 is very useful in CNN applications due to the fact that using memristors can make the multiplication circuit much simpler than not only the conventional digital multiplication circuits but also analog multiplication circuits [15, 16]. However, the accuracy of analog multiplication is affected by an offset at the output voltage, V_{OUT}. And, a negative V_{OUT} that should be generated by the on-chip voltage generator circuit is very prone to be fluctuated by supply noise, etc. For the on-chip voltage generator, we have to use a negative charge pump circuit that needs high-voltage devices that can add more complexity to process technology. One more problem in Fig. 1 can happen when the multiplier coefficient is zero. The multiplier coefficient that is programmed by the weighting memristance, M_{W1}, cannot be decreased to zero but as low as LRS. If we compare the switching memristance M_{S1} with the weighting memristance M_{W1} in the case where the multiplier coefficient is supposed to be 0, both of them have

LRS. Hence, the multiplication result by $M_{W1} = \text{LRS}$ is very different from our expectation that assumes $M_{W1} = 0\ \Omega$.

In the next part, another synaptic weighting circuit that is more advanced to mitigate the problems that are mentioned just earlier is shown. The synaptic circuit has Common-Mode Feed Back (CMFB) circuit that can minimize the offset at V_{OUT} and does not use any negative voltage. Moreover, the zero-multiplier-coefficient problem that is caused by the fact that LRS is larger than 0 Ω can be eliminated. The detailed analysis about how much the multiplication accuracy can be affected by process-V_{DD}-temperature variations in memristors is also given in this chapter.

3.1 Synaptic Weighting Circuit

Figure 2(a) shows the synaptic weighting circuit that has one shared current mirror, nine weighting circuits with memristors, nine differential pairs, and the common-mode feedback circuit. Here the common-mode feedback circuit is used to minimize the offset at the output voltage. The shared current mirror is used to add the results of nine multiplications. Each of multiplication is done by each weighting circuit using the voltage-current relationship of memristor. The result of multiplication, for example,V_{A1}–V_{B1} for the first weighting circuit in Fig. 2(a) is applied to the first differential pair to be converted to the output current. In the weighting circuit, M_{S1}, M_{S2}, M_{S3}, and M_{S4} are the switching memristors that can decide the multiplication polarity. M_{W1} is the weighting memristor whose memristance can be programmed according to the multiplier coefficient. S_1 and S_2 are used to solve a problem that is caused by zero multiplier coefficient. As mentioned earlier, the weighting memristor, M_{W1} cannot be programmed to 0 Ω. In Fig. 2(a), S_1 is open and S_2 is short when the multiplier coefficient is zero. By doing so, V_{A1} and V_{B1} become equal and the output current can be zero.

Figure 2(b) indicates that small amounts of current can flow through M_{S2} and M_{S3}, though M_{S2} and M_{S3} are as large as HRS. Though these I_2 and I_4 are much smaller than I_1, I_3, and I_5, they can affect V_{A1B1} that is the multiplication result of $I_{IN1} \times M_{W1}$. Because of this small amount of current in I_2 and I_4, multiplication accuracy can be degraded severely. To avoid the accuracy degradation in multiplication, we have to tune the weighting memristance, M_W's very precisely. Figure 2(c) shows the relationship of V_{A1B1} with the weighting memristance, M_{W1}, when $I_{IN1} = 10\ \mu\text{A}$. For the multiplier coefficient = 1, V_{A1B1} should be 10 mV that corresponds to a voltage drop at $M_{W1} = 1.02\ \text{K}\Omega$. This value 1.02 kΩ was obtained by CADENCE SPECTRE simulation. Similarly, V_{A1B1} as large as 80 mV can be obtained by $M_{W1} = 9.55\ \text{k}\Omega$, for the multiplier coefficient = 8.

One more thing to note in Fig. 2(a) is that we do not use the negative voltage, V_{BB} that is used in Fig. 1. Instead of the negative V_{BB}, $V_{DD}/2$ is used at the minus terminal of R_L, as shown in Fig. 2(a). V_{OUT} is kept by $V_{DD}/2$ by the common-mode feedback circuit when the input current is 0 A. It means that the offset voltage at V_{OUT} can be 0 V. For the plus polarity of multiplication, if the input current is

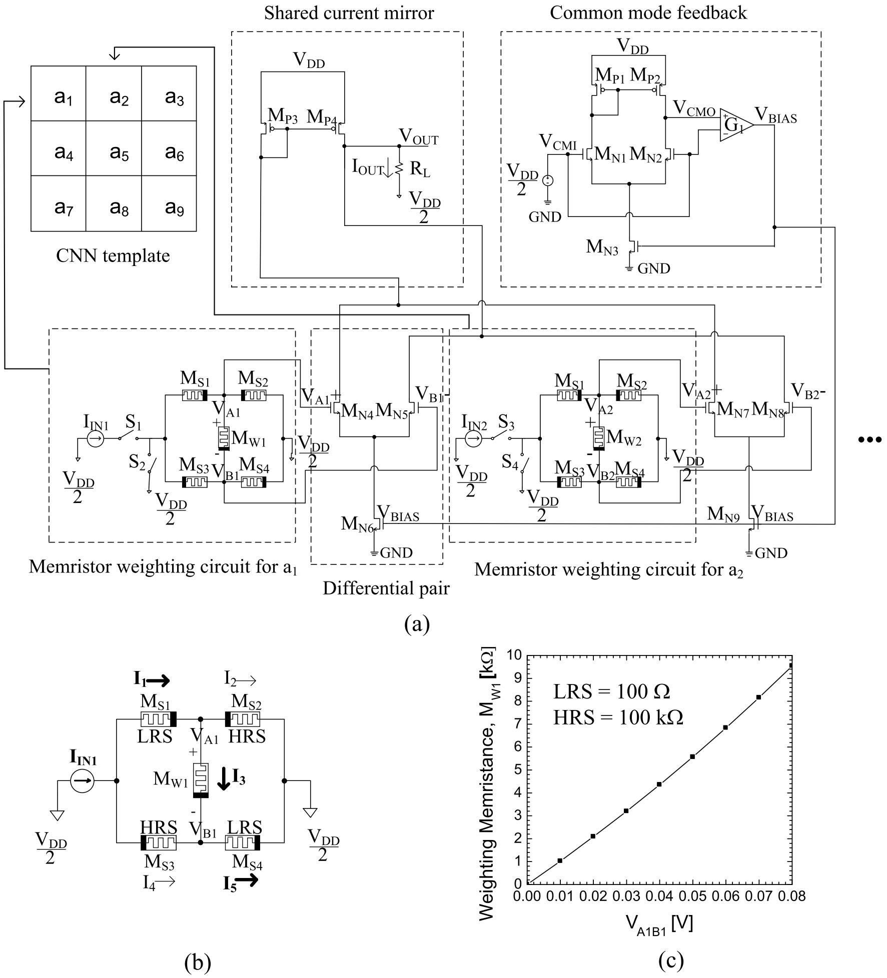
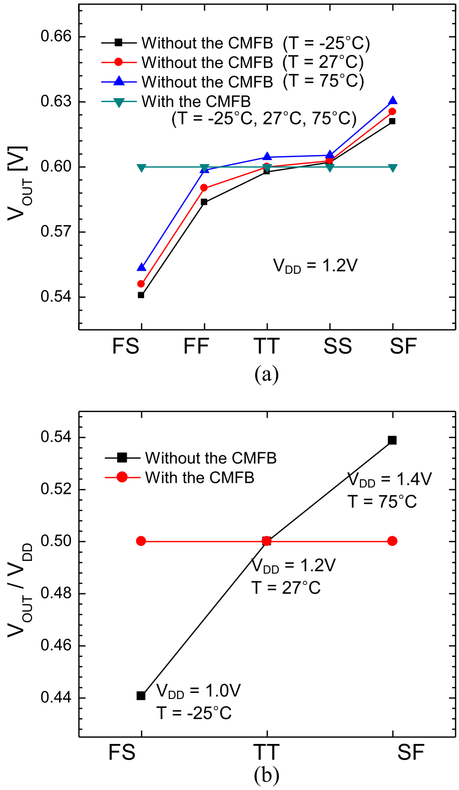


Fig. 2 (**a**) The proposed synaptic weighting circuit that includes the common-mode feedback circuit to minimize the output offset and the modified memristor weighting circuit to solve the problem of zero multiplying coefficient. This circuit has nine memristor weighting circuits, nine differential pairs, one shared current mirror, and one common-mode feedback circuit. Here the weighting circuit with M_{W1} is corresponding to multiplier coefficient, a_1, in the CNN template. The weighting circuit with M_{W2} is for the multiplication with a_2 in the template. Similarly, we can say that the other seven memristors from M_{W3} to M_{W9} are representing the seven multiplications with multiplier coefficients from a_3 to a_9. (**b**) The weighting circuit with four switching memristors and one weighting memristors. (**c**) The programmed weighting memristance, M_{W1} vs. V_{A1B1} $(= V_{A1}{-}V_{B1})$, as shown in (**b**)

higher than 0 A, the output current is going into R_L to make V_{OUT} higher than $V_{DD}/2$. If the input current is lower than 0 A, the output current is coming out from R_L thus V_{OUT} becomes lower than $V_{DD}/2$. The dynamic range of V_{OUT} is decided

Fig. 3 Comparison of the synaptic weighting circuit between with the Common-Mode Feed Back (CMFB) and without it (**a**) V_{OUT} simulation with different process corners and temperatures (**b**) V_{OUT}/V_{DD} simulation with different process corners, different temperatures, and different V_{DD}s



by R_L and I_{OUT}. For good accuracy in multiplication, V_{OUT} should be linear in the dynamic range of V_{OUT} with respect to the input current.

Now, let us see how the common-mode feedback circuit works in Fig. 1(a). Here the gate of M_{N6} is biased by V_{BIAS} that is made by the common-mode feedback circuit. The feedback is composed of the comparator, G_1 and the differential pair that has M_{N1} and M_{N2}. On the top of differential pair, the current mirror of M_{P1} and M_{P2} is placed. M_{N3} is the current sink. Here the differential-pair inputs that are connected to the gates of M_{N1} and M_{N2} are biased by not a differential signal but a common-mode input. The common-mode input, V_{CMI} is externally biased as high as $V_{DD}/2$. G_1 compares V_{CMI} and V_{CMO} thus it controls V_{BIAS} to make V_{CMI} and V_{CMO} equal. The gate voltage of M_{N3}, V_{BIAS} is also connected to the gate of M_{N6} and M_{N9} to minimize the offset at the output voltage, V_{OUT}.

To know how much effective the common-mode feedback circuit in Fig. 2(a) is, we compared V_{OUT} without the common-mode feedback circuit and with it in Figs. 3(a) and (b). Here, we assumed various process corners, supply voltages, and temperatures. As mentioned, the goal of the common-mode feedback circuit is to keep V_{OUT} by $V_{DD}/2$ when the input current is zero or the multiplier coefficient is zero, regardless of process-V_{DD}-temperature variations. First, in Fig. 3(a), we simulated V_{OUT} for different process corners and different temperatures. The simulated corners are FS, FF, TT, SS, and SF. Here, FS means that PMOS SPICE parameters are from the fast corner and NMOS parameters are from the slow corner. Similarly, SF is the case that PMOS and NMOS are at the slow corner and the fast corner, respectively. SS means that both PMOS and NMOS are slow and FF is that both PMOS and NMOS are fast. TT means that PMOS and NMOS are at the typical condition that means their performance is the same with the average value. Three different temperatures are −25 °C, 27 °C, and 75 °C in Fig. 3(a), respectively. The synaptic weighting circuit without the feedback shows different V_{OUT} values for different process corners and temperatures. However, the proposed circuit with the common-mode feedback does not allow V_{OUT} to move away from $V_{DD}/2$, when the input current is 0 A or the multiplier coefficient is zero. This fixed V_{OUT} means that the offset voltage at V_{OUT} can be zero. By having almost the zero offset voltage, the accuracy of analog multiplication can be improved compared with the conventional circuit without the common-mode feedback. Figure 3(b) shows the V_{OUT} simulation with varying V_{DD}. Though V_{DD} varies between 1.0 V and 1.4 V, V_{OUT} is fixed by $V_{DD}/2$, regardless of V_{DD} variation, as shown in Fig. 3(b).

3.2 Simulations

For simulating memristors, we used the simplest memristor model (based on the linear drift diffusion) described by the following equation (see also [3, 52])[5]

$$\begin{aligned} v(t) &= R_X(t)i(t) \\ &= \left(R_{LRS}\frac{w(t)}{D} + R_{HRS}\left(1 - \frac{w(t)}{D}\right)\right)i(t), \\ &= \left(R_{LRS}\frac{q(t)}{Q_{CRIT}} + R_{HRS}\left(1 - \frac{q(t)}{Q_{CRIT}}\right)\right)i(t) \end{aligned} \tag{8}$$

where

$$\frac{w(t)}{D} = \mu_v \frac{R_{SET}}{D^2} q(t) = \frac{q(t)}{Q_{CRIT}}$$

and

$$Q_{CRIT} = \frac{D^2}{\mu_v R_{SET}}$$

In (8), $w(t)$ is the effective width of nano-scale memristors and D is the maximum drift distance for $w(t)$, R_X(t) represents memristance, R_{LRS} and R_{HRS} are Low Resistance State and High Resistance Stats, respectively, and μ_v is the dopant mobility, Q_{CRIT} means an amount of critical charge to the HRS-to-LRS transition. When $q(t)$ reaches the value of Q_{CRIT}, R_X(t) is changed to R_{LRS} from R_{HRS}

[5] Nonlinear memristors models, including the GBCM model, will be considered in the forthcoming publications.

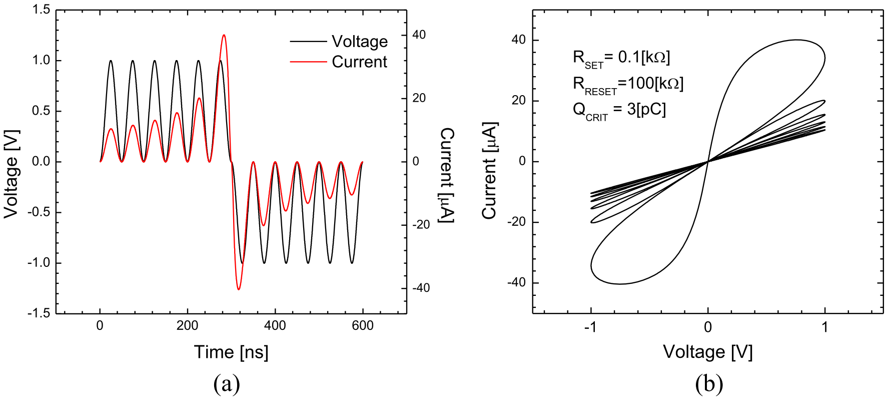
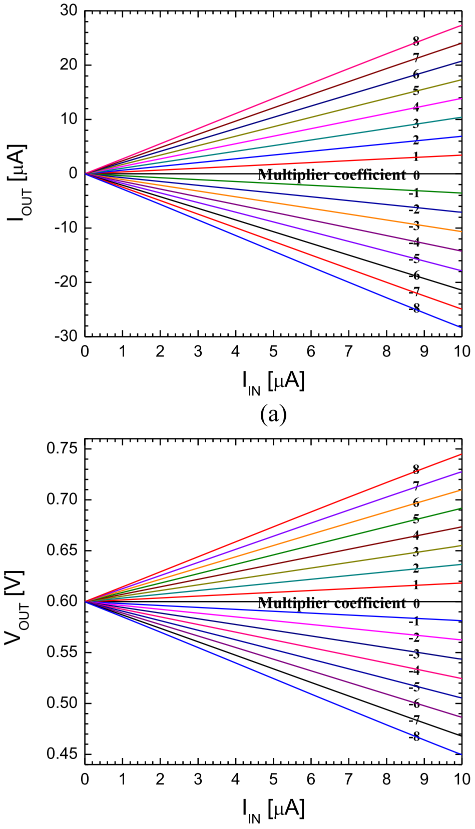


Fig. 4 (**a**) Current and voltage waveforms of a memristor when a nonlinear function of a sinusoidal voltage is applied to the memristor. The sinusoidal frequency is 20 MHz and the amplitude is 1.0 V [52]. (**b**) The current-voltage relationship shows that the memristance changes between R_{HRS} and R_{LRS}. Here, Q_{CRIT}, R_{LRS}, and R_{HRS} are 3 pC, 100 Ω, and 100 kΩ, respectively [52]

Figure 4(a) shows the current and voltage waveforms of memristor when a nonlinear function of a sinusoidal voltage is applied to the memristor [52]. The sinusoidal frequency is 20 MHz and the amplitude is 1.0 V [52]. In Fig. 4(a), we can know that the memristor current-voltage relationship shows that the memristance changes according to the charge flux through the memristor. Figure 4(b) shows typical butter-fly curves of memristor that indicate memristance changes with the applied current and voltage [52]. For simulating CMOS circuits, we used SPICE parameters from SAMSUNG 0.13-μm process with $V_{DD} = 1.2$ V.

Figure 5(a) and (b) show how much linear I_{OUT} and V_{OUT} are with respect to I_{IN}. I_{OUT} and V_{OUT} are the results of analog multiplication, where the input signal, I_{IN}, is multiplied by the multiplier coefficient that is programmed by the weighting memristance in Fig. 2(a). Here, the input current, I_{IN}, varies from 0 A to 10 μA. The multiplier coefficient can be programmed from −8 to +8 including 0, as shown in Figs. 5(a) and (b). The polarity is decided by the switching memristors of M_{S1}, M_{S2}, M_{S3}, and M_{S4} in Fig. 2(a). In the simulation, M_{S1} and M_{S4} are 100 Ω, M_{S2} and M_{S3} are 100 kΩ. I_{OUT} and V_{OUT} show good linearity with I_{IN} from 0 μA to 10 μA that indicates the multiplication accuracy can be good. The dynamic range of I_{OUT} is decided between −30 μA and +30 μA. In the simulation, the load resistance is 5.3 kΩ thus V_{OUT} can be varied from 0.45 V to 0.75 V. If we increase the load resistance, the dynamic range of V_{OUT} can be increased larger than 0.3 V.

The linearity of I_{OUT} and V_{OUT} can be affected by the variations of memristance. It is well know that memristors are very susceptible to process-V_{DD}-temperature variations that prevent memristors from being widely used instead of the current DRAM and FLASH memories. In Fig. 6, we analyzed that how much the variations affect the linearity of V_{OUT}. In Fig. 6, we defined two parameters to estimate the variation effect on the linearity of V_{OUT}. They are the Integral Non-Linearity (INL) and the Differential Non-Linearity (DNL), respectively. First, INL is defined



(a)

(b)

Fig. 5 The simulated (**a**) I_{OUT} and (**b**) V_{OUT} with I_{IN} from 0 μA to 10 μA for the multiplier coefficients that are changed from -8 to $+8$. Here V_{OUT} and I_{OUT} are the output voltage and current in Fig. 2(a), respectively. In the simulation, M_{S1} and M_{S4} are 100 Ω, M_{S2} and M_{S3} are 100 kΩ

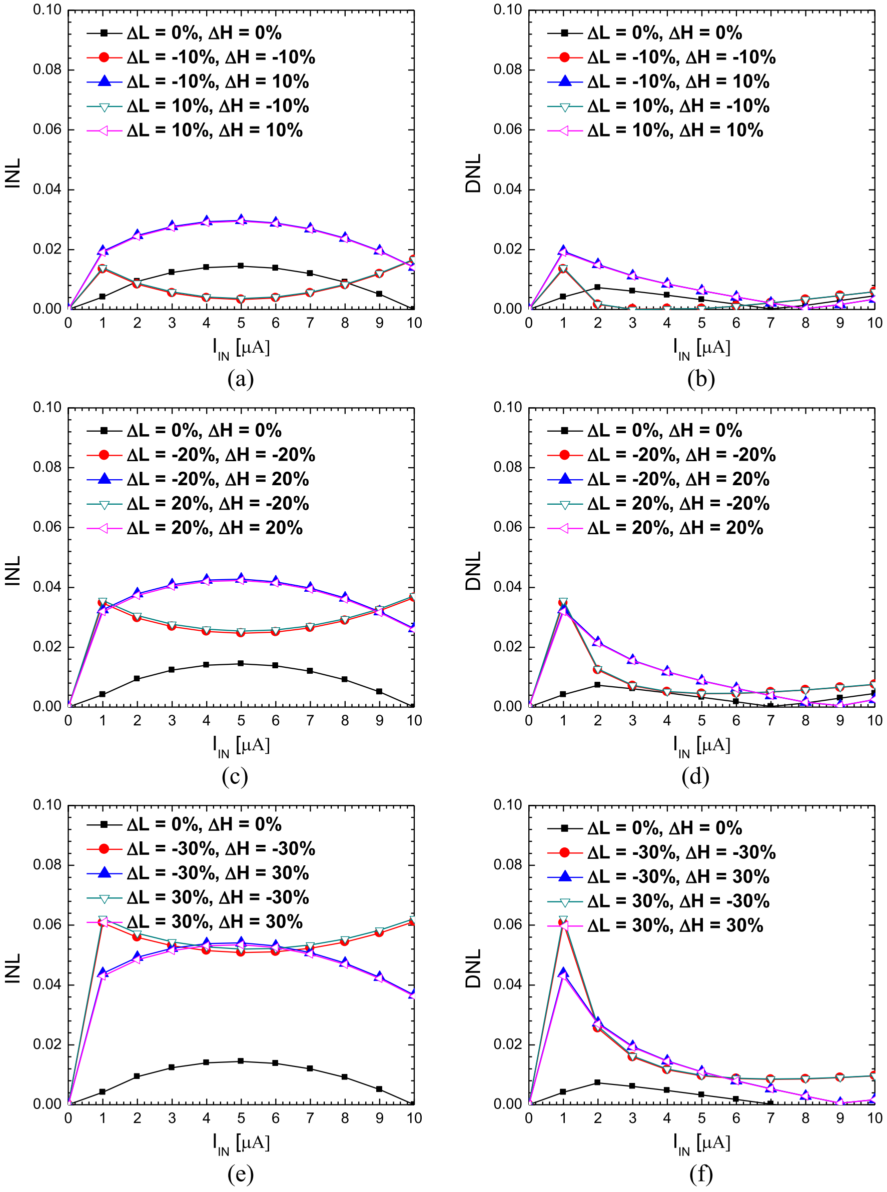
as below.

$$INL = \frac{V_{OUT,k} - V_{REF,k}}{V_{REF,k}} \tag{9}$$

In Eq. (9), $V_{OUT,k}$ is the V_{OUT} value that is obtained by the circuit simulation using Fig. 2(a), for the kth input current. The input current is given by $k \times 1$ μA. For example, the input current for $k = 1$ is 1 μA. If k is 10, the input current for $k = 10$ becomes as large as 10 μA. $V_{REF,k}$ means the ideal reference value of V_{OUT} for the kth input current. Similarly, we can define DNL by Eq. (10).

$$DNL = \frac{V_{OUT,k} - V_{OUT,k-1} - (V_{REF,k} - V_{REF,k-1})}{V_{REF,k}} \tag{10}$$

Here we assumed that the nominal LRS and HRS are 100 Ω and 100 kΩ, respectively. The LRS and HRS with variations can be expressed with LRS $= 100\ \Omega + \Delta L$ and HRS $= 100\ \text{K}\Omega + \Delta H$. ΔL and ΔH represent the percentage variations of LRS and HRS, respectively. In Figs. 6(a) and (b), we compared 5 cases of ΔL and



(a) (b)

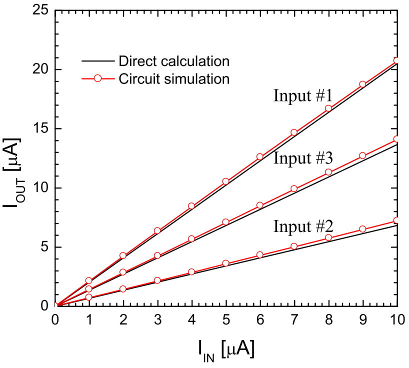
(c) (d)

(e) (f)

Fig. 6 The (**a**) INL and (**b**) DNL with LRS $= 100\ \Omega + \Delta L$ and HRS $= 100\ \mathrm{k}\Omega + \Delta H$ are simulated for $\Delta L = \pm 10\ \%$ and $\Delta H = \pm 10\ \%$. The (**c**) INL and (**d**) DNL with LRS $= 100\ \Omega + \Delta L$ and HRS $= 100\ \mathrm{k}\Omega + \Delta H$ are simulated for $\Delta L = \pm 20\ \%$ and $\Delta H = \pm 20\ \%$. The (**e**) INL and (**f**) DNL with LRS $= 100\ \Omega + \Delta L$ and HRS $= 100\ \mathrm{k}\Omega + \Delta H$ are simulated for $\Delta L = \pm 30\ \%$ and $\Delta H = \pm 30\ \%$

0	1	0
1	2	1
0	1	0

(a)



(b)

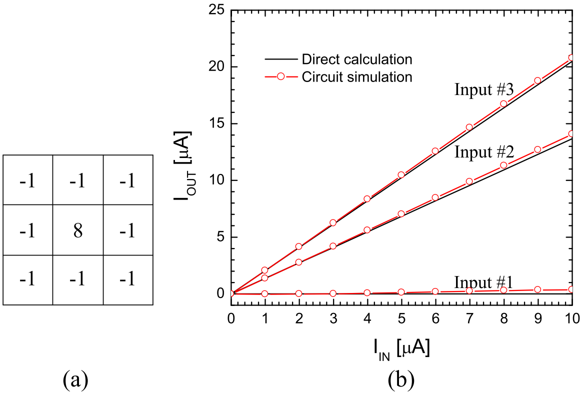
Fig. 7 (**a**) The Average template for CNN applications [15] (**b**) the comparison of I_{OUT} between the direct calculation and the circuit simulation for the three input vectors

ΔH variations. They are ($\Delta L = 0$ %, $\Delta H = 0$ %), ($\Delta L = -10$ %, $\Delta H = -10$ %), ($\Delta L = -10$ %, $\Delta H = +10$ %), ($\Delta L = +10$ %, $\Delta H = -10$ %), and ($\Delta L = +10$ %, $\Delta H = +10$ %). Similarly, in Figs. 6(c) and (d), ΔL and ΔH are given by ±20 %. In Figs. 6(e) and (f), ΔL and ΔH become as large as ±30 %. In the worst case of ΔL and ΔH variations as large as ±30 %, we can know that both INL and DNL remain always below 6.3 %.

Simulations of synaptic weighting circuit in Fig. 2(a) were performed for two CNN templates of the Average template and Laplacian template, respectively [15]. Figure 7(a) shows the Average template with nine multiplier coefficients. Each multiplier coefficient is implemented by the weighting circuit with memristors, as shown in Fig. 2(a). In Fig. 7(b), we showed the multiplication results of I_{OUT} that were simulated by the CADENCE SPECTRE circuit simulator, when we varied I_{IN} from 0 μA to 10 μA. The solid line represents the reference values of I_{OUT} that are calculated by direct multiplication using the multiplier coefficients of the Average template in Fig. 7(a). Comparing the circuit simulation with the direct calculation tells us that the discrepancy is less than 6.8 % in the worst case. In Fig. 7(b), we tested three cases of input vectors that are shown 'Input #1', 'Input #2', and 'Input #3', respectively.

Figure 8(a) shows the nine multiplier coefficients in the Laplacian template. Similarly, with Fig. 8(b), we applied three input vectors to the Laplacian template. With reference to Fig. 8(b), the worst discrepancy between the circuit simulation and direct calculation is lower than 3.2 %.

Figure 9(a) shows the original image data which has 60 × 60 pixels. Figure 9(b) is the image data which is mathematically calculated with Average Template in Fig. 7(a). Here the calculation is done by MATLAB. Figure 9(c) and (d) are the simulated image with Average Template, where the memristor weighting circuits are programmed according to the multiplier coefficients that are defined in Average Template, as shown in Fig. 7(a). Here the circuit simulation is done by CADENCE SPECTRE tool [53]. For Fig. 9(c), the process corner is at SF and CMFB circuit in Fig. 2(a) is not used. Here we can know that the output voltages of the synaptic weighting circuits in Fig. 9(c) can have large amounts of offset voltage because

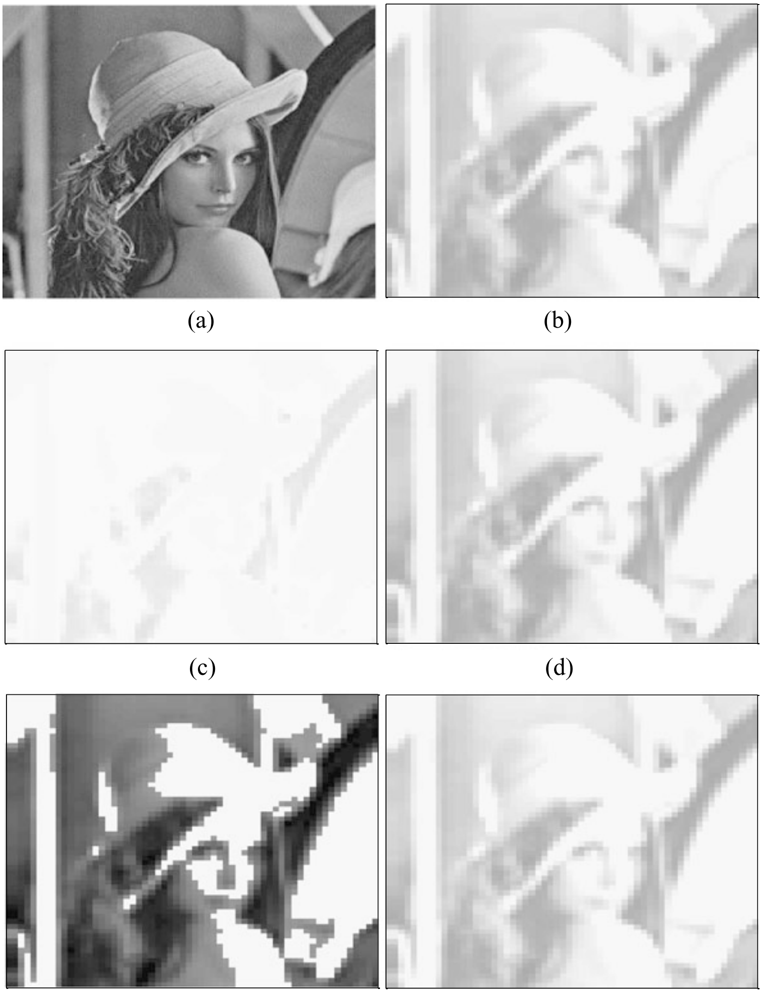


-1	-1	-1
-1	8	-1
-1	-1	-1

Fig. 8 (**a**) The Laplacian template for CNN applications [15] (**b**) the comparison of I_{OUT} between the direct calculation and the circuit simulation for the three input vectors

they are not compensated by CMFB circuit. As a result, the simulated image seems much lighter than the calculated [54] image data in Fig. 9(b) that is obtained by MATLAB simulation. On the contrary, Fig. 9(d) shows the simulated image that is much similar with the calculated image in Fig. 9(b), though the process corner is also at SF. The difference between Figs. 9(c) and (d) comes from CMFB circuit. In Fig. 9(d), CMFB circuit is used to compensate the offset voltages that may happen at the output nodes due to the process, temperature, and V_{DD} corners. In Fig. 9(e), the process corner is at FS and CMFB circuit is not used here. Similarly to Fig. 9(c) with SF corner, Fig. 9(e) with FS corner also shows large amounts of offset voltage at the output nodes. However, the offset voltages due to FS corner have opposite polarity to the Fig. 9(c), thereby the image in Fig. 9(e) seems much darker than the calculated image in Fig. 9(b). Figure 9(f) has very similar image with Fig. 9(b). This is because the offset voltages due to FS corner can be compensated by CMFB circuit that is used in Fig. 2(b).

Figure 10(a) shows the original image data which has 60 × 60 pixels that is the same with Fig. 9(a). Figure 10(b) is the image data which is calculated with Average Template in Fig. 8(a). Here Fig. 10(b) is obtained by MATLAB simulation. Figure 10(c) and (d) are the simulated image with Laplacian Template, where the memristor weighting circuits are programmed according to the multiplier coefficients in Laplacian Template that is defined in Fig. 8(a). Here the circuit simulation is done by CADENCE SPECTRE tool. For Fig. 10(c), the process corner is at SF and CMFB circuit is not used. Here we can know that the output voltages of the synaptic weighting circuits in Fig. 10(c) can have large amounts of offset voltage because they are not compensated by CMFB circuit. As a result, the simulated image seems much lighter than Fig. 10(b) that is the image calculated by MATLAB. On the contrary, Fig. 10(d) shows the simulated image that is much similar with the calculated image in Fig. 9(b), though the process corner is also at SF. The different between Figs. 10(c) and (d) comes from CMFB circuit. In Fig. 10(d), CMFB circuit is used to compensate the offset voltages that may happen at the output nodes due to the process corners. Figures 10(e) and (f) show the simulated images for the FS corner without and with CMFB circuit respectively. The advantage of using a circuit for output offset minimization is obvious



(a) (b)

(c) (d)

(e) (f)

Fig. 9 (**a**) The original image with 60 × 60 pixels (**b**) the image that is calculated with Average Template by MATLAB simulation (**c**) the simulated image with Average Template by CADENCE SPECTRE circuit simulator (SF corner and without CMFB circuit) (**d**) the simulated image with Average Template by CADENCE SPECTRE circuit simulator (SF corner and with CMFB circuit) (**e**) the simulated image with Average Template by CADENCE SPECTRE circuit simulator (FS corner and without CMFB circuit) (**f**) the simulated image with Average Template by CADENCE SPECTRE circuit simulator (FS corner and with CMFB circuit)

Figure 11(a) shows the original image data which has 60 × 60 pixels that is the same with Figs. 9(a) and 10(a). Figure 11(b) is the MATLAB image data which is calculated with Laplacian Template in Fig. 8(a). Figures 11(c) and (d) are the simulated image with Laplacian Template, where the memristor weighting circuits are programmed according to the multiplier coefficients in Laplacian Template. For

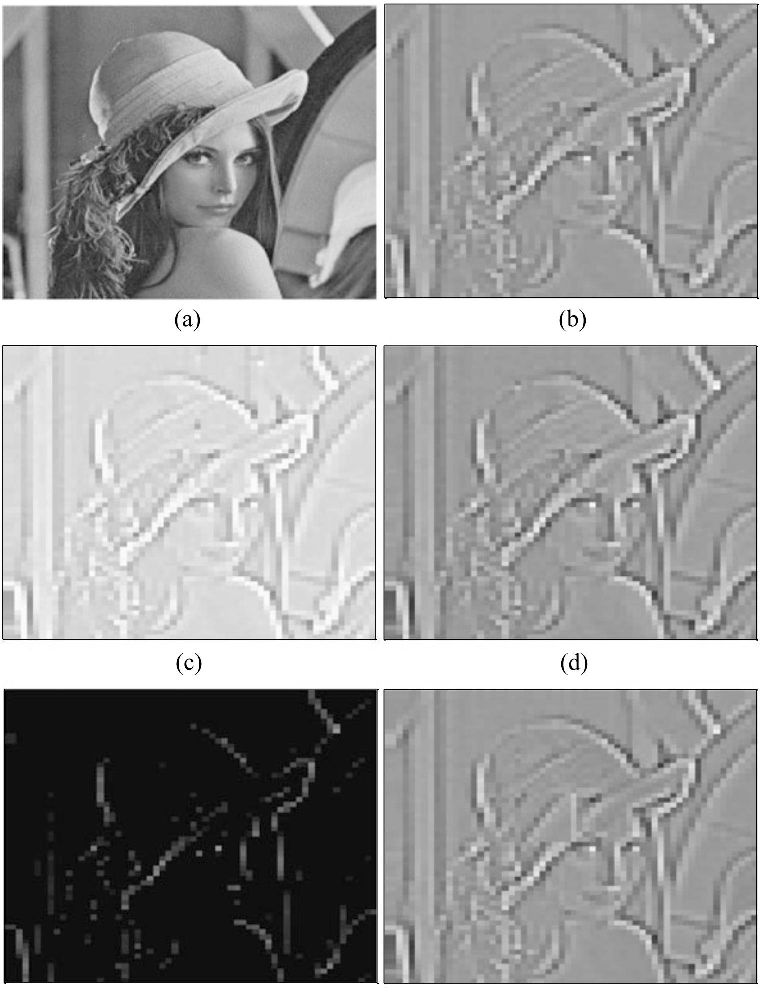


Fig. 10 (**a**) The original image with 60 × 60 pixels (**b**) the image calculated with Laplacian Template by MATLAB (**c**) the simulated image with Laplacian Template by CADENCE SPECTRE circuit simulator (SF corner and without CMFB circuit) (**d**) the simulated image with Laplacian Template by CADENCE SPECTRE circuit simulator (SF corner and with CMFB circuit) (**e**) the simulated image with Laplacian Template by CADENCE SPECTRE circuit simulator (FS corner and without CMFB circuit) (**f**) the simulated image with Laplacian Template by CADENCE SPECTRE circuit simulator (FS corner and with CMFB circuit)

Fig. 11(c), we did not use the precise tuning circuit to program weighting coefficients. In Fig. 11(c), we can know that multiplier coefficients in Laplacian Template are not so accurate because they are not programmed by the precise tuning circuit. As a result, Fig. 11(c) seems very different from the MATLAB calculated image in Fig. 11(b). On the contrary, Fig. 11(d) shows the simulated image that is much

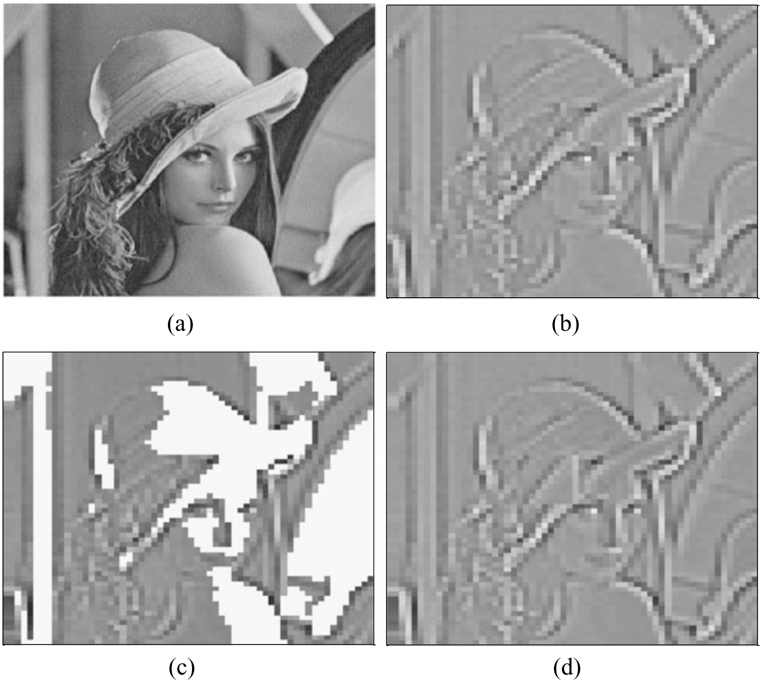


Fig. 11 (**a**) The original image with 60 × 60 pixels (**b**) the image calculated with Laplacian Template by MATLAB simulation (**c**) the simulated image with Laplacian Template by CADENCE SPECTRE circuit simulator (without the precise tuning circuit of weighing coefficients) (**d**) the simulated image with Laplacian Template by CADENCE SPECTRE circuit simulator (with the precise tuning circuit of weighting coefficients)

similar with the calculated image in Fig. 11(b). The accurate image in Fig. 11(d) is due to that the precise tuning circuit is used to program the weighting coefficients of Laplacian Template.

4 Conclusions

After a brief review of the memristor models available in the literature, this paper describes the PSpice-based implementation of the Generalized Boundary Condition Memristor (GBCM) model, which stands out over other models [17, 18] thanks to the adaptability of the boundary behavior and to the tunability of the state evolution rate in sub- and supra-threshold regimes. Reference [29] contains a couple of case studies where the use of the PSpice emulator, which is inspired to the circuit from Batas and Fiedler [55], sheds light into the synapse-alike behavior [56] of the memristor. More in general, such circuit may be of great help to researchers willing to investigate in the user-friendly PSpice environment the unique memristor features [57] and the extraordinary opportunities [58] this nano-device offers in integrated circuit design. Further, besides PSpice implementations of physical memristor models (such as the one proposed here for the GBCM model), there are a number of memristor emulators [59, 60] which may also be of added value for future research into non-conventional memristor [61]-based architectures [62].

The second part of the paper introduces the synaptic weighting circuits that could perform analog multiplication for CNN applications. The common-mode feedback circuit was used in the weighting circuits to minimize offset voltages that can be found at the output nodes. The multiplication accuracy of synaptic weighting circuits can be degraded because HRS is not infinitely large and LRS is not as small as 0 Ω in real memristors. To maximize the multiplication accuracy, two MOSFET switches were added to the memristor weighting circuits and the weighting memristance should be programmed very precisely considering the leakage current through HRS. Variations in memristance were analyzed to estimate the effect on accuracy of analog multiplication. Finally, Average and Laplacian Templates were tested and verified for both input vectors and input images by the circuit simulation using the memristor weighting circuits.

Acknowledgements This work was partially supported by the CRT Foundation, under the project no. 2012.1121 and by the Ministry of Foreign Affairs ”*Con il contributo del Ministero degli Affari Esteri, Direzione Generale per la Promozione del Sistema Paese*”.

The CAD tools were supported by the IC Design Education Center (IDEC), Korea. This work was financially supported by NRF-2013K1A3A1A25038533 through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning.

References

1. Chua, L.O.: Memristor: the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
2. Chua, L.O., Kang, S.-M.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
3. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
4. Vontobel, P.O., Robinett, W., Kuekes, P.J., Stewart, D.R., Straznicky, J., Williams, R.S.: Writing to and reading from a nano-scale crossbar memory based on memristors. Nanotechnology **20**(42), 425204 (2009)
5. Pershin, Y.V., Di Ventra, M.: Practical approach to programmable analog circuits with memristors. IEEE Trans. Circuits Syst. I **57**(8), 1857–1864 (2010)
6. Zamarreño-Ramos, C., Camuñas-Mesa, L.A., Pérez-Carrasco, J.A., Masquelier, T., Serrano-Gotarredona, T., Linares-Barranco, B.: On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. Front. Neuromorphic Eng. Front. Neurosci. **5**(26), 1–22 (2011)
7. Borghetti, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R., Williams, R.S.: ‘Memristive’ switches enable ‘stateful’ logic operations via material implication. Nat. Lett. **464**(7290), 873–876 (2010)
8. Xia, Q., Robinett, W., Cumbie, M.W., Banerjee, N., Cardinali, T.J., Yang, J.J., Wu, W., Li, X., Tong, W.M., Strukov, D.B., Snider, G.S., Medeiros-Ribeiro, G., Williams, R.S.: Memristor-CMOS hybrid integrated circuits for reconfigurable logic. Nano Lett. **9**(10), 3640–3645 (2009)
9. Snider, G.: Spike-timing-dependent learning in memristive nanodevices. In: International Symposium on Nanoscale Architecture, pp. 85–92 (2008)

10. Chua, L.O., Yang, L.: Cellular neural networks: theory. IEEE Trans. Circuits Syst. **35**(10), 1257–1272 (1988)
11. Chua, L.O., Yang, L.: Cellular neural networks applications. IEEE Trans. Circuits Syst. **35**, 1273–1290 (1988)
12. Domínguez-Castro, R., Espejo, S., Rodríguez-Vázquez, A., Carmona, R.A., Földesy, P., Zarándy, Á., Szolgay, P., Szirányi, T., Roska, T.: A 0.8-μm CMOS two-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage. IEEE J. Solid-State Circuits **32**, 1013–1026 (1997)
13. Kim, H., Son, H., Roska, T., Chua, L.O.: High-performance viterbi decoder with circularly connected 2-D CNN unilateral cell array. IEEE Trans. Circuits Syst. I **52**, 2208–2218 (2005)
14. Zheng, C.-D., Zhang, H., Wang, Z.: Improved robust stability criteria for delayed cellular neural networks via the LMI approach. IEEE Trans. Circuits Syst. II, Express Briefs **57**, 41–45 (2010)
15. Kim, H., Sah, P., Yang, C., Roska, T., Chua, L.O.: Memristor bridge synapses. Proc. IEEE **100**(6), 2061–2070 (2012)
16. Kim, H., Sah, M.P., Yang, C., Roska, T., Chua, L.O.: Neural synaptic weighing with a pulse-based memristor circuit. IEEE Trans. Circuits Syst. I **59**, 148–158 (2012)
17. Joglekar, Y.N., Wolf, S.T.: The elusive memristive element: properties of basic electrical circuits. Eur. J. Phys. **30**, 661–675 (2009)
18. Biolek, Z., Biolek, D., Biolková, V.: Spice model of memristor with nonlinear dopant drift. Radioengineering **18**(2), 210–214 (2009)
19. Shin, S., Kim, K., Kang, S.-M.: Compact models for memristors based on charge-flux constitutive relationships. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **29**(4), 590–598 (2010)
20. Corinto, F., Ascoli, A.: A boundary condition-based approach to the modeling of memristor nanostructures. IEEE Trans. Circuits Syst. I **59**(11), 2713–2726 (2012)
21. Pickett, M.D., Strukov, D.B., Borghetti, J.L., Yang, J.J., Snider, G.S., Stewart, D.R., Williams, R.S.: Switching dynamics in titanium dioxide memristive devices. J. Appl. Phys. **106**(7), 074508 (2009)
22. Kvatinski, S., Friedman, E.G., Kolodny, A., Weiser, U.C.: TEAM: threshold adaptive memristor model. IEEE Trans. Circuits Syst. I **60**(1), 211–221 (2013)
23. Cai, W., Tetzlaff, R.: Advanced memristive model of synapses with adaptive thresholds. In: Proc. IEEE Int. Workshop on Cellular Nanoscale Networks and Their Applications (2012)
24. Froemke, R.C., Dan, Y.: Spike-timing-dependent synaptic modification induced by natural spike trains. Nature **416**(6879), 433–438 (2002)
25. Oka, T., Nagaosa, N.: Interfaces of correlated electron systems: proposed mechanism for colossal electroresistance. Phys. Rev. Lett. **95**(26), 64031–64034 (2005)
26. Beck, A., Bednorz, J.G., Gerber, Ch., Rossel, C., Widmer, D.: Reproducible switching effect in thin oxide films for memory applications. Appl. Phys. Lett. **77**(1), 140 (2000)
27. Linn, E., Rosezin, R., Kügeler, C., Waser, R.: Complementary resistive switches for passive nanocrossbar memories. Nat. Mater. **9**, 403–406 (2010)
28. Chua, L.O.: Resistance switching memories are memristors. Appl. Phys. A **102**(4), 765–783 (2011)
29. Ascoli, A., Corinto, F., Tetzlaff, R.: Generalized Memristor Boundary Condition model. IEEE Trans. Circuits Syst. I (2013, under revision)
30. Prodromakis, T., Peh, B.P., Papavassiliou, C., Toumazou, C.: A versatile memristor model with nonlinear dopant kinetics. IEEE Trans. Electron Devices **58**(9), 3099–3105 (2011)
31. Benderli, S., Wey, T.A.: On spice macromodelling of TiO_2 memristors. Electron. Lett. **45**(7), 377–379 (2009)
32. Cadence Design Systems. In: OrCad PSpice User's Guide. OrCAD, Inc., USA. Available online as PSpice.pdf (1998). http://www.electronics-lab.com/downloads/schematic/013/
33. Rák, a., Cserey, G.: Macromodeling of the memristor in SPICE. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **29**(4), 632–636 (2010)

34. Kavehei, O., Iqbal, A., Kim, Y.S., Eshraghian, K., Al-Sarawi, S.F., Abbott, D.: The fourth element: characteristics, modelling, and electromagnetic theory of the memristor. Proc. R. Soc. A, Math. Phys. Eng. Sci. **466**(2120), 2175–2202 (2010)
35. Hee Kim, T., Jang, E.Y., Lee, N.J., Choi, D.J., Lee, K.-J., Jang, J.-t., Choi, J.-s., Moon, S.H., Cheon, J.: Nanoparticle assemblies as memristors. Nano Lett. **9**(6), 2229–2233 (2009)
36. Lehtonen, E., Laiho, M.: CNN using memristors for neighborhood connections. In: IEEE Int. Workshop on Cellular Nanoscale Networks and Their Applications, pp. 1–4 (2010)
37. Yang, J.J., Pickett, M.D., Li, X., Ohlberg, D.A.A., Stewart, D.R., Williams, R.S.: Memristive switching mechanism for metal/oxide/metal nanodevices. Nat. Nanotechnol. **3**(7), 429–433 (2008)
38. Roska, T., Chua, L.O.: The CNN universal machine: an analogic array computer. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **40**(3), 163–173 (1993)
39. Lehtonen, E., Poikonen, J., Laiho, M., Lu, W.: Time-dependence of the threshold voltage in memristive devices. In: IEEE Int. Symp. on Circuits and Systems, pp. 2245–2248 (2011)
40. Sah, M.Pd., Yang, C., Kim, H., Chua, L.O.: A voltage-mode memristor bridge synaptic circuit with memristor emulators. Sensors **12**(3), 3587–3604 (2012)
41. Strukov, D., Williams, R.S.: Exponential ionic drift: fast switching and low volatility of thin-film memristors. Appl. Phys. A, Mater. Sci. Process. **94**(3), 515–519 (2009)
42. Simmons, J.G.: Generalized formula for the electric tunnel effect between similar electrodes separated by a thin insulating film. J. Appl. Phys. **34**(6), 1793–1803 (1963)
43. Abdalla, H., Pickett, M.D.: Spice modeling of memristors. In: IEEE Int. Symp. on Circuits and Systems, pp. 1832–1835 (2011)
44. Pershin, Y.V., Di Ventra, M.: Spice model of memristive devices with threshold. Available online at arXiv (2012). http://lanl.arxiv.org/abs/1204.2600v4
45. Pershin, Y.V., La Fontaine, S., Di Ventra, M.: Memristive model of amoeba learning. Phys. Rev. E **80**(2), 021926 (2009)
46. Linares-Barranco, B., Serrano-Gotarredona, T.: Memristance can explain spike-time-dependent-plasticity in neural synapses. Available on-line at Nature Precedings. http://hdl.handle.net/10101/npre.2009.3010.1
47. Pazienza, G.E., Albo-Canals, J.: Teaching memristors to EE undergraduate students. IEEE Circuits Syst. Mag. **11**(4), 36–44 (2011)
48. Eshraghian, K., Kavehei, O., Cho, K.-R., Chappell, J.M., Iqbal, A., Al-Sarawi, S.F., Abbott, D.: Memristive device fundamentals and modeling: applications to circuits and systems simulation. In: Proceedings of the IEEE, vol. 100, pp. 1991–2007 (2012)
49. Ascoli, A., Corinto, F., Senger, V., Tetzlaff, R.: Memristor model comparison. IEEE Circuits Syst. Mag. (2013). doi:10.1109/MCAS.2013.2256272
50. Corinto, F., Ascoli, A., Gilli, M.: PSpice switch-based versatile memristor model. In: Proc. Int. Symp. Circ. Syst. I (2013)
51. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. 1297–1301 (2010)
52. Jung, C.M., Jo, K.H., Lee, E.S., Vo, H.M., Min, K.S.: Zero-sleep-leakage flip-flop circuit with conditional-storing memristor retention latch. IEEE Trans. Nanotechnol. **11**, 360–366 (2012)
53. Kim, Y.S., Min, K.S.: Synaptic weighting circuits for cellular neural networks. In: International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA2012), Turin, Italy (2012)
54. Abramowitz, M., Stegun, I.A.: Handbook of Mathematical Functions, with Formulas, Graphs and Mathematical Tables. Dover, New York (1972)
55. Batas, D., Fiedler, H.: A memristor SPICE implementation and a new approach for magnetic flux-controlled memristor modeling. IEEE Trans. Nanotechnol. **10**(2), 250–255 (2011)
56. Hebb, D.O.: The Organization of Behavior: A Neuropsychological Theory. Wiley, New York (1949)
57. Biolek, D., Biolek, Z., Biolkova, V.: Pinched hysteretic loops of ideal memristors, memcapacitors and meminductors must be ‘self-crossing’. Electron. Lett. **47**(25), 1385–1387 (2011)

58. Pershin, Y.V., Di Ventra, M.: Memory effects in complex materials and nanoscale systems. Adv. Phys. **60**(2), 145–227 (2011)
59. Corinto, F., Ascoli, A.: The simplest class of passive Memristor Emulators. IEEE Trans. Circuits Syst. I (2013, submitted)
60. Corinto, F., Ascoli, A.: Memristive diode bridge with LCR filter. Electron. Lett. **48**(14), 824–825 (2012)
61. Chua, L.O.: The fourth element. Proc. IEEE **100**(6), 1920–1927 (2012)
62. Corinto, F., Kang, S.-M., Ascoli, A.: Memristor based neural circuits. In: Int. Symp. Circ. and Syst. (2013)

Evolving Memristive Neural Networks

Gerard David Howard, Larry Bull, Ben De Lacy Costello, Ella Gale, and Andrew Adamatzky

Abstract Of the many network representations in which memristors can be modelled, neural networks are perhaps the most enticing as they open the possibility for neuromorphic computing—biologically-inspired brainlike information processing in hardware. Memristors are analogous to biological synapses; both feature nonvolatile resistance, charge-dependent plastic response to activity, and can provide adaptive learning when coupled with a Hebbian mechanism. In this chapter, various types of memristors are deployed as synapses in spiking networks.

Biological information processing implies autonomous learning control—a neuro-evolutionary approach provides this functionality and is used to search for beneficial network topologies. The main focus of this work extends the remit of the evolutionary algorithm to alter the conductance profiles of individual memristors, creating networks of heterogeneous *variable* synapses. These variable memristor networks are tested against networks of benchmark synapses in a robotic pathfinding scenario. Experimental findings conclude that the variable synapses bestow more behavioural degrees of freedom to the networks, allowing them to outperform the comparative synapse types.

1 Introduction

Memristors [4] partially comprise a class of devices known as Resistive Memories (RMs [49]), which harness one of several mechanisms—including ionic/electronic transport [45], spintronics [48], and magnetoresistance [28]—to realise hardware memory where the device exhibits a charge-dependent nonvolatile resistance somewhere between the device's lowest (R_{on}) and highest (R_{off}) possible resistances. Flow of charge through the device induces a resistance alteration; typically current flowing in one direction increases resistance, with the inverse being true for current flowing in the opposite direction. In this study we focus on memristors, which permit continuous resistance alteration, and another type of Resistive Memory called a

G.D. Howard (✉) · L. Bull · B. De Lacy Costello · E. Gale · A. Adamatzky
University of the West of England, Frenchay Campus, Coldharbour Lane, Bristol BS161QY, UK
e-mail: david.howard@csiro.au

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_14,

Resistive Switching Memory (RSM, [49]) which gives a strictly bi-state switching between R_{on} and R_{off}.

Much recent research has focused on adaptive networks of RMs, where adaptivity implies the autonomous generation of self-organised learning architectures—a powerful motif, especially given the opportunity to realise such designs at the nanoscale using dense, highly-connected networks of physical devices. In terms of a single memristive node, adaptivity allows the device to dynamically reconfigure it's output based on the history of activity that it has experienced [11] via a Hebbian [16] process. Scaling up to the level of the network, adaptivity may arise through input-output autocorellation of some external stimuli, plus the aforementioned Hebbian process, exploiting the chance physical makeup of a disorganised network of devices [9]. Another prospective approach harnesses an Genetic Algorithm (GA) [17]—a stochastic search algorithm—to discover beneficial network topologies. Previous studies have evolved various network representations using groups of single memristors [22], memristive logic gates [20], and memristors as synapses in Spiking Neural Networks (SNN, e.g. [15]) [21, 23], a process termed *neuroevolution* [12].

The main contribution of this chapter extends the traditional remit of the neuroevolutionary mechanism from topology-only search to alter the conductance profiles of individual synapses, creating a *variable RM*. As the conductance profile of the RM is responsible for its behaviour, variable RMs can potentially impart a variety of adaptive behaviours to the networks. Our hypothesis is that the additional degrees of functional freedom afforded to the variable RM networks can be harnessed by the evolutionary process to outperform a number of benchmark memristive network types. A simulated robotics navigation task is selected for this purpose. We note that, as the RM models used are based on the characterisation of physical devices, any evolved designs have the potential to be deployed in hardware as task-optimal subsystems of larger architectures.

1.1 Content Overview

The remainder of this chapter is ordered as follows: Sect. 2 introduces background research. Section 3 details the system. Section 4 describes the GA. Section 5 provides details on the variable RMs. Section 5.2 provides experimental results and the chapter concludes with a discussion and presentation of future research directions in Sect. 6.

2 Background

Although any connectionist network representation is implicitly amenable to evolutionary techniques, the evolution of memristive neural networks is prominently promising as it paves the way to the realisation of neuromorphic (NM) systems [31]. Neuromorphic computing is a biologically-inspired paradigm that aims to emulate brain-like functionality in hardware by distributing processing and memory across the entire architecture. A typical proposed neuromorphic system involves a collection of Complimentary Metal-Oxide Semiconductor units which function as neurons, connected by a mesh of synapse-mimicing memristors. Such an arrangement aims to exploit the physical properties of the memristor (nanoscale fabrication, nonvolatile state, implicit synapse-like functionality) to enable highly parallel, low-power, bio-inspired processing. A Hebbian mechanism harnesses the computational power of this configuration by allowing the RM synapses to alter their weights in a flexible manner during the lifetime of the network, depending on the temporal activation of the neurons they are connected to. In this chapter we demonstrate a prototype NM system that combines memristive synapses (both memristor and RSM) and a neuroevolutionary model to automatically create task-specific networks.

2.1 Spiking Networks

Spiking Neural Networks (SNNs) present a phenomenological model of neural activity in the brain. In an SNN, a network of neurons are linked via unidirectional, weighted connections. Each neuron has a measure of excitation, or *membrane potential* and communicates via the voltage spike, or *action potential*. A neuron spikes when it's membrane potential exceeds some threshold, which typically requires a cluster of incoming spikes arriving within a short time period. A spike emitted from a neuron is received by all connected postsynaptic neurons.

As the membrane potential may be considered a form of memory, such networks are able to produce temporally dynamic activation patterns, potentially providing increased computing power [29, 40] when considering temporal problems (e.g. robotics, time series) compared to stateless network models, such as the Multi Layer Perceptron (MLP) [39]. Two well-known formal SNN implementations are the Leaky Integrate and Fire model [15] and the Spike Response Model [15]. Neuroevolution involves the use of evolutionary techniques to alter the topology and/or weights of neural networks [12]. The literature also covers evolution of networks purely for robotics tasks [34]—such as the test problem used in this chapter.

2.2 Resistive Memory Synapses

Resistive memories are a broad class of devices that display nonvolatile state (resistance), which can be varied by applying an appropriate charge to the device. They typically consist of two electrodes separated by an insulating layer—it is the resistance of this insulating layer which varies depending on charge. Numerous Resistive Memories have been previously manufactured from a plethora of materials [3]. Resistive Switching Memories are predominantly metal oxides (HfO_2, Cu_2O, ZnO,

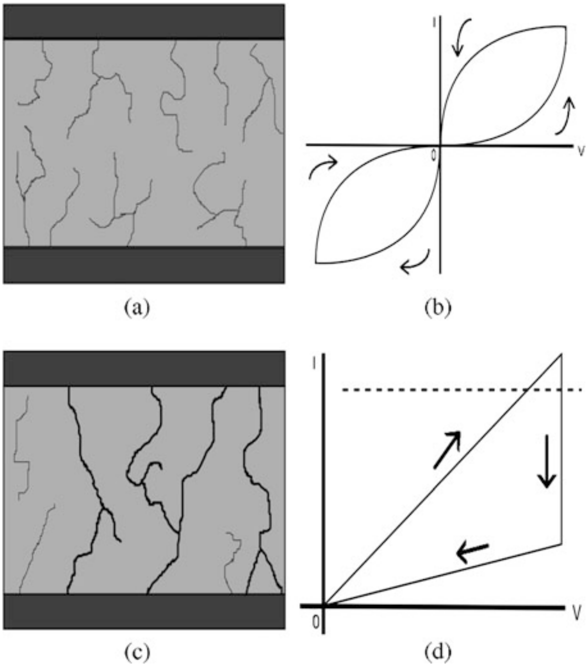


Fig. 1 (**a**) Incomplete filament formation (memristor)—as there are no complete filaments, ionic conductivity gives rise to (**b**) Generalised HP memristor I–V curve (**c**) Complete filament formation (RSM); *dark lines* show complete filaments (**d**) Generalised RSM I–V curve, *dashed line* showing a possible current compliance. The steeper gradient is the LRS, the shallower is the HRS

ZrO_2, TiO_2). Memristor materials are more varied and include conductive polymers [10], metal silicides [25], and crystalline oxides [7] in addition to certain metal oxides.

A popular theory [49] states that disparities between the two resistance profiles (memristive and RSM) are the result of the presence or absence of conductive filaments (pathways through the insulator that permit expedient electron flow) in the substrate. In our taxonomy, memristors (Fig. 1(a)) do not form complete filaments, giving rise to the characteristic nonlinear I–V curves of these devices as other mechanisms (such as ionic conductivity) play a more prominent role in electron transport (Fig. 1(b)). RSMs are able to form complete filaments (Fig. 1(c)) which connect the two electrodes, resulting in ohmic I–V profiles (Fig. 1(d)). We note that this distinction is not universal (e.g. under specific conditions an RSM may act like memristor, and *vice versa*).

2.2.1 Memristors

The memristor is a class of RM theoretically characterized by Chua [4] which has recently enjoyed a resurgence of interest from the research community after being manufactured at the nano scale by HP labs [44]. Memristors are the fourth fundamental circuit element, joining the capacitor, inductor and resistor. Figure 1(d) shows a typical I–V resistance profile for the HP device (termed a *hysteresis loop*).

As filaments are not formed, ionic conductivity alone gives rise to the nonlinear I–V curve seen in Fig. 1(d). A classic model of the HP memristor involves modelling the device as two variable resistors R_{on} and R_{off}. The instantaneous resistance of the device can be attributed to proportional contributions of R_{on} (which is doped with oxygen vacancies (charge carriers) and therefore has low resistance) and R_{off} (which displays higher resistance). Charge flowing through the memristor in a given direction causes the oxygen vacancies to migrate in that direction. This moves the boundary between R_{on} and R_{off}, thus modifying their proportional contribution to the total device resistance.

Formally, a memristor is a passive two-terminal electronic device that is described by the relation between the device terminal voltage, v, terminal current, i (which is related to the charge q transferred onto the device), and magnetic flux, φ, as Eq. (1) shows. Resistance can be made to increase or decrease depending on the polarity of the voltage. The nonvolatile resistance, M, is a nonlinear function of the charge—see Eq. (2).

$$v = M(q)i \tag{1}$$

$$M(q) = d\varphi(q)/dq \tag{2}$$

Previous applications of memristors within neural paradigms are ubiquitous: Titanium dioxide memristors [44] have been used as the constituent elements of nanoscale neural crossbars [41], and silver silicide memristors have been shown to function in neural architectures [25]. Other successful applications include the modelling of learning in amoeba [35], as well as pattern recognition by crossbar circuits for robotic control [33]. In particular, Mouttet [33] highlights the attractive prospect of applying evolutionary computation techniques directly to memristive hardware, as memristors can simultaneously perform the functions of both processor and memory.

2.2.2 Resistive Switching Memories

RSMs typically comprise an ion-conducting insulator which is sandwiched between two metal electrodes. RSMs are a bistable resistive switch; pulsing with a voltage over some threshold transfers the device from an initial Low Resistance State (LRS) to a High Resistance State (HRS). Successive voltage pulses (in SNN terminology, *spikes*) can switch between these states by successively forming and breaking filaments in the insulator—see Figs. 1(c), (d). RSMs follow Eqs. (1) and (2), under the proviso that M is now a linear function of charge. RSMs operate under voltage of a single polarity. Examples of RSMs are confined to binary operation, e.g. as Resistive Random Access Memory [3, 18]. It should be noted that neural implementations exist [50].

2.3 Synaptic Plasticity

Hebbian learning [16] is thought to account for adaptation and learning in the brain through a process whereby “neurons that fire together, wire together”—or more technically *in the event that a presynaptic neuron causes a postsynaptic neuron to fire, the synaptic strength between those two neurons is increased so that such an event is more likely to happen in the future*. Such a mechanism allows for input-dependent, self-organising activity.

Spike Time dependent Plasticity (STDP) [26] was originally formulated as a way of implementing Hebbian learning within artificial neural networks. It has been postulated that a memristance-like mechanism affects synaptic weights in biological neural networks [27], based on similarities between memristive equations and their neural counterparts. More recently, it has been proposed that the axon itself may be comprised of a chain of memristors [5]. This raises the possibility that biological neural processing may be intrinsically memristive in nature, although such research is outside the scope of this study.

The notion that varied plastic behaviours could be combined in a single network is an attractive one from a computing perspective, as more functional degrees of freedom may be afforded to the synapses. Integration of neuroevolution with heterogeneous neuromodulation rules has been investigated [43], and has been extended to robot controllers [8]. Increased behavioural diversity (and high-quality pathfinding in the latter case) is reported. Probabilistic spike emission which is governed by modulatory Hebbian rules has also been investigated [30]. The authors show a biologically-plausible mechanism capable of computing with short spike trains where the population of synapses display heterogeneous probabilities of transmitting/blocking a spike. A nodes-only encoding scheme has been presented [46], where synapses are affected by four versions of the Hebb rule to generate online path-finding behaviour from initially random actions. Synaptic weights are not evolved; instead evolution is performed on the rules which govern how synapses react to STDP. High adaptability to new environments is evidenced, however as synaptic weights are not directly modelled, and all synapses at a given node display homogeneous STDP behaviour, it is unclear how to transition such a scheme to memristive/hardware implementations.

Typically when implementing STDP with RMs, a bidirectional voltage spike is emitted by a neuron whose membrane potential exceeds some threshold (see Sect. 3.1). This spike can be approximated by either a continuous [2, 27] or discrete. References [25, 42] waveform through time, which is transmitted to all synapses that the neuron is connected to (presynaptic or postsynaptic). In the case of memristors, if the instantaneous voltage across a synapse surpasses some threshold—typically when the waveforms sufficiently overlap—the conductance of the synapse alters. For RSMs, multiple consecutive voltage spikes of a given polarity within a short time frame are required to switch the device from one resistance state to the other. Note that this removes the element of biological realism from RSM STDP whilst providing a fast-switching binary behaviour.

Memristive STDP has been shown [21] to vary the behaviour of a network during a trial, as well as a way to mitigate device variations that are currently intrinsic to hardware memristor implementations [36]. Previous research has compared the evolution of networks of HP, PEO-PANI, LIN, and GA synapse types [21]; the main findings were:

- Memristive STDP enabled the generation of highly fit solutions (statistically fitter than networks using non-plastic connections).
- The evolutionary algorithm assigned roles to the synapses based on their STDP behaviours (HP memristors were statistically more frequently connected to an inhibitory neuron, PEO-PANI were statistically more frequently attached to an excitatory neuron).
- Self-adaptive search parameters were found to be context-sensitive (statistically significant inter-parameter variances), and beneficial to the evolution of the networks.

In this study we use these synapse types as benchmarks, and extend the concept of memristive STDP by allowing for variable RMs (memristors and RSMs) whereby the effect that STDP has on the synapse can be tailored by evolution to suit it's role within the network.

A long-term goal of this research is the creation of adaptive hardware neural networks. Creating hardware RM synapses that match their simulated counterparts is likely to be a difficult obstacle to overcome. However, there are a number of possible methods to achieve this transition.

Part of the equation expressing the "physical properties" of the titanium dioxide memristor [44] is encapsulated in the variable β by Howard et al. [21] in Eq. (3) below.

$$\beta = \gamma^v / D^2 \tag{3}$$

Here, γ^v is the mobility of oxygen vacancies and D is the device thickness in nanometres. To create variable RM synapses, we allow the value of the physical properties parameter to vary.

Firstly, we note that the sizes of D given for current memristor models predict a scale that is small enough to permit sufficient synaptic density for neuromorphic hardware implementations—equivalent to that found in the human brain. Furthermore, Eq. (3) allows us to predict that highly variable synapses will be produceable at the required scales. Note that a small D will provide a higher behavioural variance; in other words, *the smaller the scale of the devices, the more variable they will be.* A benefit of the "physical properties" parameter is that, by matching simulated prediction to experimental observation, the properties of a memristor corresponding to a given β can be calculated and used as a metric for the selection of suitable device size, synthesis method, and material selection. RSMs could be categorised in the same way, based on their switching voltage sensitivities to create "best-fit" hardware networks based on simulated results. Creating an accurate predictive model from simulation to reality is unlikely to be a trivial process, however.

A more detailed memristor model [13] (with some experimental verification [14]) indicates that varying the electrode size also affects memristance—potentially providing another way of achieving variable RMs via tweaking the electrode size without having to alter the properties of the insulator.

A final method of varying the STDP response of an RM involves irradiation by an ion beam [47]. Simulations of irradiated titanium dioxide memristors (and hence titanium dioxide RSMs) were found to possess a reduced resistance in the doped region (R_{on}), as well as lower oxygen vacancy mobility (γ^{v}). Bombardment of specific synapses could therefore alter their β values in an online manner, although the precise method of targetted radiation delivery would depend upon the physical structure of the network. The main advantage of this approach is online behaviour modification, e.g. as a form of intrinsic hardware evolution.

In summary, we have shown RMs to be suitable synapse candidates, and introduced STDP as a means to achieving learning within RM spiking networks. Furthermore, we provide evidence showing the potential for evolutionary approaches to design variable memristive neural networks, and shown how we might create such networks in hardware.

3 The System

The system consists of a population of SNNs which are evaluated on a robotics test problem, and altered via GA operation which is detailed in Sect. 4. To introduce the terminology to be used throughout this paper: each experiment lasts for 1000 evolutionary *generations*; each generation involves two new networks being created and evaluated on the test problem (a *trial*). Each trial consists of 4000 *timesteps*, which begin with the reading of sensory information and calculation of action, and end with the agent performing that action. Every timestep consists of 21 *steps* of SNN processing, at the end of which the action is calculated. The state of the system was sampled every 20 generations and used to create the results. Results were averaged over 30 experimental repeats.

3.1 Neural Control Architecture

We base our SNN implementation on the Leaky Integrate and Fire [15] model. Neurons can be stimulated either by an external current or by connections from presynaptic neurons. Each neuron has a membrane potential, y, where $y > 0$, which slowly degrades over time. As spikes are received by the neuron, the value of y is increased in the case of an excitatory spike, or decreased if the spike is inhibitory. If y surpasses a positive threshold, θy, the neuron spikes and transmits an action potential to every neuron to which it is presynaptic, with strength relative to the

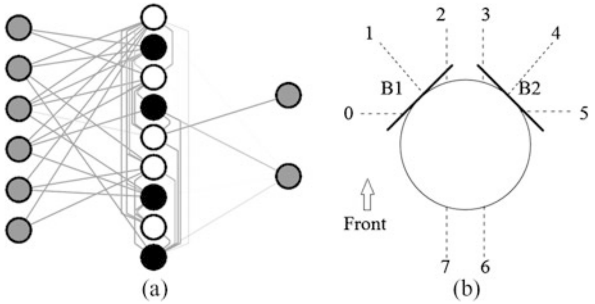


Fig. 2 (**a**) A typical SNN architecture. In the hidden layer, *white neurons* denote excitatory neurons and *black neurons* signify inhibitory neurons. (**b**) Khepera sensory arrangement. Three light sensors and 3 IR sensors share positions *0*, *2*, and *5* and form the network input. Two bump sensors, *B1* and *B2*, are shown attached at 45 degree angles to the front-left and front-right of the robot

connection weight between those two neurons. The neuron then resets its membrane potential to zero. The membrane potential of a neuron at time t is given in Eq. (4). Equation (5) shows the reset formula.

$$y(t+1) = y(t) + \big(I + a - by(t)\big) \tag{4}$$

$$\text{If } (y > \theta y) y = c \tag{5}$$

Here, $y(t)$ is the membrane potential at time t, I is the input current to the neuron, a is a positive constant, b is the degradation (leak) constant and c is the reset membrane potential of the neuron. A model of temporal delays is used so that, in the hidden layer only, a spike sent to a neuron not immediately neighbouring the sending neuron is received x steps after it is sent, where x is the number of neurons between the sending neuron and receiving neuron. SNN parameters are initial hidden layer nodes = 9, $a = 0.3$, $b = 0.05$, $c = 0.0$, $c_{ini} = 0.5$, $\theta_y = 1.0$, output window size = 21. A sample network is provided in Fig. 2(a).

The SNNs were used to control a simulated Khepera II robot with 8 light sensors and 8 distance sensors. At each timestep (64 ms in simulation time), the agent sampled its light sensors, whose values ranged from 8 (fully illuminated) to 500 (no light) and IR distance sensors, whose response values ranged from 0 (no object detected) to 1023 (object very close). All sensor readings were scaled to the range [0, 1] (0 being unactivated, 1 being highly activated) before being input to the SNN. Six sensors comprised the input state for the SNN, three IR and three light sensors at positions 0, 2, and 5 as shown in Fig. 2(b). Additionally, two bump sensors were added to the front-left and front-right of the agent to prevent it from becoming stuck against an object. If either bump sensor was activated, an interrupt was sent causing the agent to reverse 10 cm and the agent to be penalised by 10 timesteps. Movement values and sensory update delays were constrained by accurate modelling of physical Khepera agent. Sensory noise was added based on Webots Khepera data; ±2 % noise for IR sensors and ±10 % noise for light sensors, all randomly sampled from a uniform distribution. Wheel slippage was also included (10 % chance). The spike trains of the output neurons were discretised into *high* or *low* activated (*high* activation if more than half of the 21 SNN processing steps generated a spike at the neuron, *low* otherwise). Three actions were possible: forward, (maximum movement on both left and right wheels, *high* activation of both output neurons) and continuous turns to both the left (*high* activation on the first output neuron, *low* on the second) and right (*low* activation on the first output neuron, *high* on the second)—caused by halving the left/right motor outputs respectively.

3.2 Benchmark Synapses

The functionality of the SNN depends on the type of synapse used. Here we describe the equations governing the two comparitive memristor synapses (HP and PEO-PANI), the linear device and the constant connection—details on creating variable RMs are given in Sects. 4.3.1 and 4.3.2. These synapse types serve as a means of comparison to the variable devices that will be introduced later.

3.2.1 HP Memristor

The HP memristor is comprised of thin-film Titanium Dioxide (TiO_2) and oxygen-depleted Titanium Dioxide ($TiO_{(2-x)}$), which have different resistances. The boundary between the two compounds moves in response to the charge on the memristor, which in turn alters the resistance of the device. To allow for future self-adaptation, memristance equations are refactored from the original, as given in [44].

In the following equations, W is the scaled weight (conductance) of the connection, G is the unscaled weight of the connection, M is the memristance, $sf1$ and $sf2$ are scale factors, R_{off} is the resistance of the TiO_2, R_{on} is the resistance of the $TiO_{(2-x)}$, q is the charge on the device, q_{min} is the minimum allowed charge, and β encompasses the physical properties of the device. The original profiles, used for the benchmark memristors, are recreated using a rescaled $\beta = 1$, $R_{\text{on}} = 0.01$, $R_{\text{off}} = 1$, $q_{\text{min}} = 0.0098$.

$$sf_1 = 0.99 \Big/ \left(1 - \left(\frac{1}{-R_{\text{off}} R_{\text{on}} \beta q_{\text{min}} + R_{\text{off}}}\right)\right) \tag{6}$$

$$sf_2 = 1 \Big/ \left(\frac{-R_{\text{off}} R_{\text{on}} \beta (R_{\text{on}} - R_{\text{off}})}{-R_{\text{on}} R_{\text{off}} \beta + R_{\text{off}}} sf_1\right) - 1 \tag{7}$$

$$q = \left(\frac{1}{-R_{\text{off}} R_{\text{on}} \beta}\right)\left(\frac{1}{sf_1 (W + sf_2)} - R_{\text{off}}\right) \tag{8}$$

$$M = -R_{\text{off}} R_{\text{on}} \beta q + R_{\text{off}} \tag{9}$$

$$G = \frac{1}{M} \tag{10}$$

$$W = Gsf_1 - sf_2 \tag{11}$$

3.2.2 PEO-PANI Memristor

The polyethyleneoxide-polyaniline (PEO-PANI) memristor consists of layers of PANI, onto which Li$^+$-doped PEO is added [10]. We have phenomenologically recreated the performance characteristics of the PEO-PANI memristor in terms of the HP memristor, creating a memristance curve similar to that seen in [10]. Two additional parameters, $G_{q_{\min}}$ and $G_{q_{\max}}$, are the values of G when q is at its minimum ($q_{\min}$) and maximum ($q_{\max}$) values respectively. As with the HP equations, $\beta = 1$ will produce the original PEO-PANI profile.

$$q_{\max} = (R_{\text{on}} - R_{\text{off}}) / - R_{\text{on}} R_{\text{off}} \beta \tag{12}$$

$$G_{q_{\min}} = 1/(-R_{\text{off}} R_{\text{on}} \beta q_{\min} - R_{\text{on}}) + R_{\text{on}} \tag{13}$$

$$G_{q_{\max}} = 1/(-R_{\text{off}} R_{\text{on}} \beta q_{\max} - R_{\text{on}}) + R_{\text{on}} \tag{14}$$

The two scale factors are recalculated in Eqs. (15) and (16). Following this q (Eq. (17)) and M (Eq. (18)) are calculated, then G is calculated as in Eq. (10), and W as in Eq. (11).

$$sf_1 = 0.99/(G_{q_{\max}} - G_{q_{\min}}) \tag{15}$$

$$sf_2 = (G_{q_{\min}} sf_1) - 0.01 \tag{16}$$

$$q = \left(\frac{1}{-R_{\text{off}} R_{\text{on}} \beta}\right)\left(\frac{1}{((W + sf_2)/sf_1) - R_{\text{on}}} + R_{\text{on}}\right) \tag{17}$$

$$M = (-R_{\text{off}} R_{\text{on}} \beta q - R_{\text{on}}) + (1/R_{\text{on}}) \tag{18}$$

3.2.3 Linear Device

This hypothetical benchmark device provides linear behaviour so that each STDP event affects synaptic weighting equally. It is included to contrast the nonlinearity of the memristive conductance profiles whilst still providing gradual weight change under STDP. The connection weight of the device, W, is given in Eq. (19). Here, L is the number of STDP steps it takes to linearly increase W from R_{off} to R_{on}.

$$W = W + (1/L) \tag{19}$$

3.2.4 Constant Connection

The constant connection presents a static horizontal resistance profile similar to a resistor. The conductance of the connection is set random-uniformly in the range [0, 1] during initialisation and may be altered during application of the GA, but is unaffected by STDP and therefore constant during a trial. We use the constant connection as a static (e.g. non-switching) RSM, whose evolution is tailored via selection of appropriate weight selection rather than switching between two fixed weights.

3.3 STDP Implementation

In Sect. 2.3 a number of STDP implementations were reviewed. As our SNNs operate under discrete time quanta, we have elected to use discrete-time stepwise waveforms [25, 42]. Each neuron in the network is augmented with a “last spike time” variable *LS*, which is initially 0. When a neuron spikes, this value is set to some positive number—in this case 3. At the end of each of the 21 steps that make up a single timestep, each RSM is analysed by summating the *LS* values of its presynaptic and postsynaptic neurons. Following this, each *LS* value is decremented to a minimum of 0, creating a discrete stepwise waveform through time.

For memristors, all synapse weights are initially 0.5, and can vary in-trial via STDP. If the *LS* value exceeds some positive threshold θLS, memristance of the synapse occurs (see Fig. 3(a)). Whether the connection increases or decreases depends on which neuron has the highest *LS* value, providing pre- to postsynaptic temporal coincidence. If the *LS* values are identical, STDP does not occur as one spike cannot be said to have directly caused the other. Each STDP event either increases or decreases q by Δq as detailed in Eq. (20), which is then used to calculate memristor weight as detailed in Sects. 3.2.1, 3.2.2.

$$\Delta q = (q_{\max} - q_{\min})/L \tag{20}$$

From Fig. 4, we see that the values of β and W can have a pronounced effect on the functionality of the synapse—both HP and PEO-PANI synapses display larger ΔW per STDP event when β is a low number and either $W > 0.1$ (for HP synapses), or $W < 0.9$ (for PEO-PANI synapses).

To extend the concept of STDP to RSMs, we augment each synapse with parameters S_n, which represents the sensitivity of the device to voltage buildup (in the form of repeated STDP spikes), and S_c, which tracks the number of consecutive STDP events the synapse has experienced. All synapses are initially in the Low Resistance State ($W = 0.9$). At each step we check every synapse as before, incrementing S_c if an STDP event occurs at the synapse and decrementing S_c if no STDP event occurs at that step. If $S_n = S_c$, the RSM switches to the High Resistance State ($W = 0.1$) and S_c is reset to 0. The RSM can switch back and forth between the

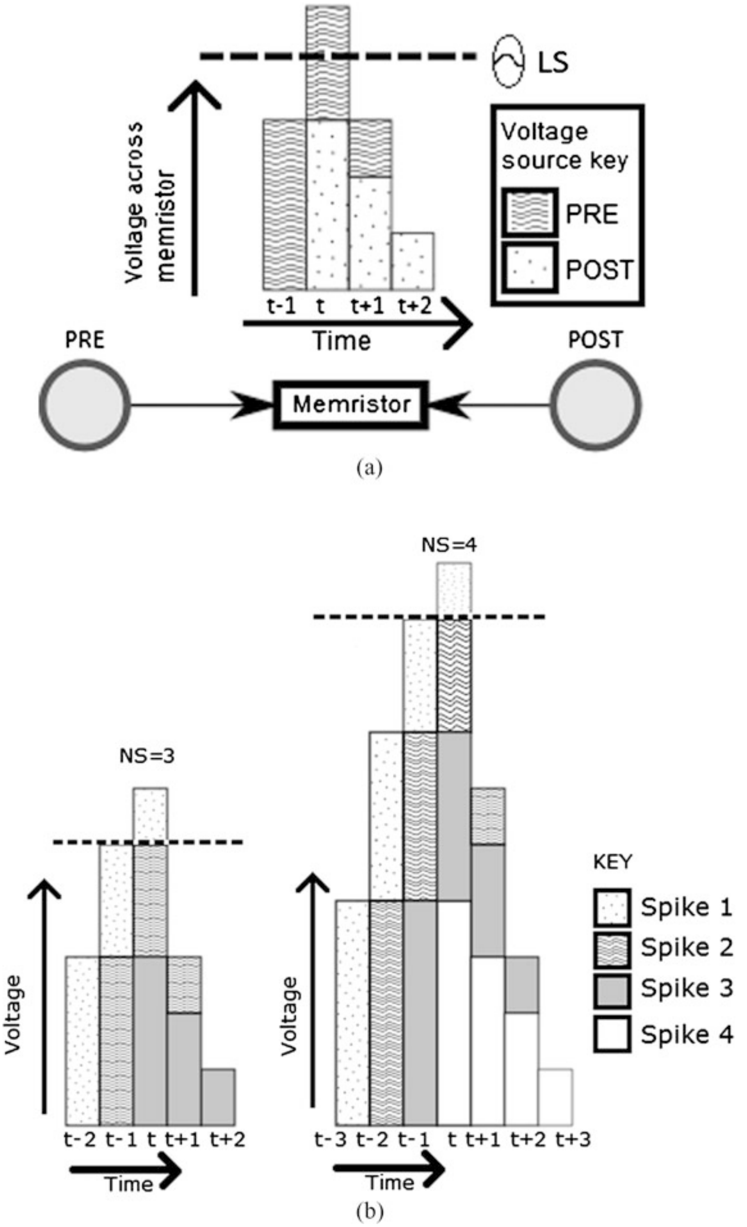


Fig. 3 Showing a positive STDP event for a memristor. A presynaptic voltage spike is received at time $t - 1$, with a postsynaptic voltage spike at time t. Combined, the voltage surpasses θLS, increasing the conductivity of the device. (**b**) If the memristor is replaced by an RSM, consecutive voltage spikes (l.h.s $S_n = 3$, r.h.s $S_n = 4$) serve to push the voltage past a threshold, causing a switch. *Dashed lines* show the derived voltage threshold. Voltage spikes are decremented at each subsequent time step

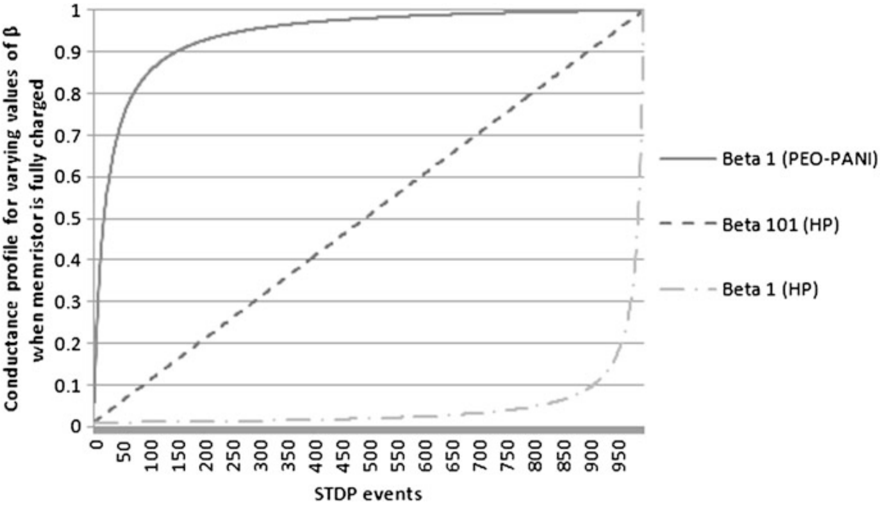


Fig. 4 Displaying resistance profiles attained with different values of β when fully charging a memristor. Here the x axis shows 1000 positive STDP events assuming $L = 1000$. Static HP and PEO-PANI memristors have $\beta = 1$

LRS and HRS during a trial. Note that the polarity of the voltage spike in RSM networks is always the same, regardless of the coincidence of presynaptic and postsynaptic spikes across the device. Figure 3(b) shows how this mechanism compares to regular STDP.

During a trial, the majority of RM synapses will have their weights altered via STDP. After a trial, every RM synapse is reset to its original weight of 0.5 (memristor) or 0.9 (RSM).

4 Genetic Algorithm

Having described the component parts of our networks, we now detail the implementation of the steady-state GA that acts on them. In our GA, two parents are selected fitness-proportionately, mutated, and used to create two offspring. We use only mutation to explore weight space; crossover is omitted as sufficient solution space exploration can be obtained via a combination of self-adaptive weight and topology mutations; a view that is reinforced in the literature [38]. The offspring are inserted into the population and two networks with the lowest fitness deleted.

The onus of this GA is on adaptivity, which justified when the application area of neuromorphic computing is considered—brainlike systems must be able to autonomously adapt to a changing environment and adjust their learning accordingly. This potentially allows increased structural stability in highly fit networks whilst enabling less useful networks to vary more strongly per GA application. Mechanistically, self-adaptation also permits the use of self-repair/self-modification, wherein the GA is able to (i) in a stable environment, lower mutation rates to enable homeostatis or provide incremental, gradual improvements or (ii) when the environment rapidly changes, or part of the network fails, increase learning rates to more quickly adapt to these new conditions. When coupled with neuroevolution, the effect is to tailor the evolution of the network to the complexity of the environment explicitly, e.g. each network controls its own architecture autonomously in terms of (i) amount of mutation that takes place in a given network at a given time (ii) adapting the hidden-layer topology of the neural networks to reflect the complexity of the problem considered by the network [24]. This mechanism was first used with SNNs by Howard [19]. We note that benefits of this approach will be more pronounced when dynamically controlling hardware implementations—this will be the topic of future research.

4.1 Self-adaptive Mutation

We utilise self-adaptive mutation rates as with Evolutionary Strategies [37]—to dynamically control the frequency and magnitude of mutation events taking place in each network. This potentially allows increased structural stability in highly fit networks whilst allowing less fit networks to vary more strongly per GA application. Here, the μ $(0 < \mu \leq 1)$ value (rate of mutation per allele) of each network is initialized uniformly randomly in the range $[0, 0.25]$. During a GA cycle, a parent's μ value is modified as in Eq. (21), the offspring then adopts this new μ, and mutates itself by this value, before being inserted into the population.

$$\mu \rightarrow \mu \exp^{N(0,1)} \tag{21}$$

Only non-memristive networks can alter their connection weights via the GA. Connection weights in this case are initially set during network creation, node addition, and connection addition randomly uniformly in the range [0, 1]. Memristive network connections are always set to 0.5, and cannot be mutated from this value. This aims to force the memristive networks to harness the plasticity of their connections during a trial to successfully solve the problem.

4.2 Topology Mechanisms

Given the desire for adaptive solutions, it would be useful if appropriate network structure is allowed to develop until some task-dependent required level of computing power is attained. In our system, each network has a varying number of hidden layer neurons (initially 9, and always > 0). Additional neurons can be added or removed from the single hidden layer based on two new self-adaptive parameters, ψ $(0 < \psi \leq 1)$ and ω $(0 < \omega \leq 1)$. Here, ψ is the probability of performing neuron addition/removal and ω is the probability of adding a neuron; removal occurs with probability $1-\omega$. Both have initial values taken from a random uniform distribution, with ranges [0, 0.5] for ψ and [0, 1] for ω. Offspring networks have their parents ψ and ω values modified using Eq. (21) as with μ, with neuron addition/removal taking part after mutation. Added nodes are initially excitatory with 50 % probability, otherwise they are inhibitory.

Automatic feature selection is a method of reducing the dimensionality of the data input to a process by using computational techniques to select and operate exclusively on a subset of inputs taken from the entire set. Feature selection in neural networks can disable synaptic (traditionally input) connections [6]. In this paper we allow each connection to be individually enabled/disabled, a mechanism termed "Connection Selection". During a GA cycle a connection can be enabled or disabled based on a new self-adaptive parameter τ (which is initialized and self-adapted in the same manner as μ and ψ). If a connection is enabled for a non-memristive network, its connection weight is randomly initialised uniformly in the range [0, 1], memristive connections are always set to 0.5. During a node addition event, new connections are set probabilistically, with $P = 0.5$ of the connection being enabled. Connection selection is particularly important to the memristive networks. As they cannot alter connection weights via the GA, variance induced in network connectivity patterns plays a large role in the generation of STDP in the networks. Likewise, RSM networks rely on Connection Selection to generate synchronised synaptic excitations/inhibitions, which allow the network to generate appropriate output actions.

4.3 GA Control of Variable Synapses

The STDP responses of the variable RM synapses are governed by the self-adaptive parameter ι, which is initialised and self-adapted as with μ. Whereas memristive STDP can be viewed as a form of in-trial context-sensitive weight mutation, as shown in Fig. 4, RSM STDP is more akin to context-sensitive connection selection (Fig. 5).

4.3.1 Variable Memristor

To allow the memristive profiles to change, we self-adapt β, which is derived from the physical characteristics of the device (e.g., device thickness, mobility of ions, etc.). The self-adaptive memristor profiles are allowed to range from HP-like to PEO-PANI-like profiles, each of which are governed by different equations, outlined in Sects. 3.2.1, 3.2.2. Because of this, we augment each memristor with a *type*, which is set to either 0 and 1 on memristor initialisation, with $P = 0.5$ of each *type* being selected based on a uniform distribution. β is then initialised randomly uniformly in the range $[\beta_{\min}, \beta_{\max}]$. If $type = 0$, the refactored HP equations are

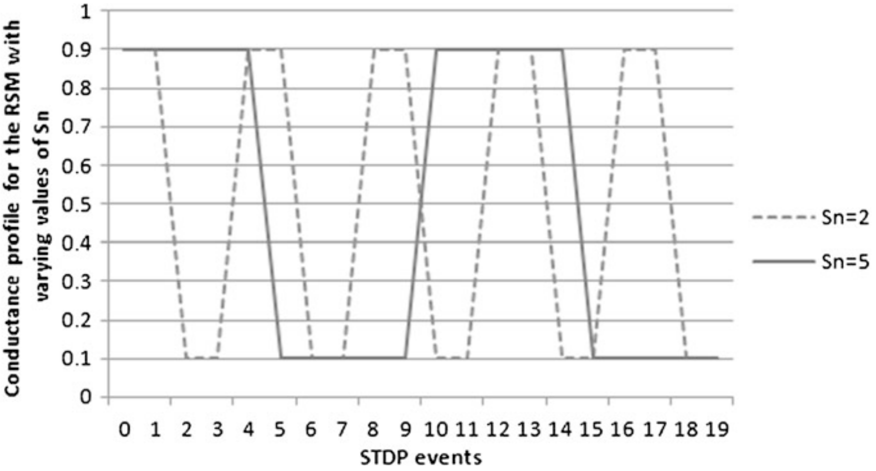


Fig. 5 Showing the resistance profile for an RSM with varying θLS when supplied with constant STDP for 20 time steps

used to calculate the profile of the device; otherwise the PEO-PANI equations are used. Satisfaction of ι during a GA application modifies β by ±0–10 % of the total range of β. If a memristor's new value of β surpasses a threshold $\beta_{\max}$, the *type* of the memristor is switched and a new β calculated as $\Delta\beta - \beta_{\max}$. In this way, a smooth transition between the different profile types is achieved.

4.3.2 Variable RSM

The variable S_n represents the voltage threshold of the device, that is the number of consecutive STDP events required to cause a switch in the RSM (see Fig. 3(b)). A lower S_n represents increased sensitivity to voltage spikes and enables more expedient switching dynamics—see Fig. 5. On synapse initialisation, the integer S_n is selected uniform-randomly in the range $[S_{n\mathrm{MIN}}, S_{n\mathrm{MAX}}]$, so that a minimum of $S_{n\mathrm{MIN}}$ consecutive STDP events are required to cause a switch. Satisfaction of ι during a GA cycle alters S_n by ± 1 of its current value, constrained to the range $[S_{n\mathrm{MIN}}, S_{n\mathrm{MAX}}]$ as before.

5 Experimentation

In the following experiment we gauge the impact of both variable RMs, comparing to benchmark systems comprised of static HP and PEO-PANI memristors, linear devices and constant connections (nonswitching RSMs). We refer to the various network types as follows: variable memristor = MEM, variable RSM = RSM, static HP memristor = HP, static PEO-PANI memristor = PEO, linear device = LIN,

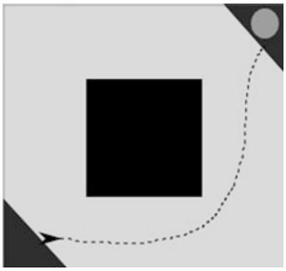


Fig. 6 The test environment. The agent begins in the *lower-left* and must reach a light source (*circle*) in the *upper-right*, circumnavigating the central obstacle. An example agent path is shown (*dotted line*)

constant-connection network = GA. The experiment was repeated 30 times, the statistics recorded were the averages of these 30 runs. All experiments had a population size of 100 networks and were evolved for 1000 generations, with a maximum of 4000 timesteps per trial. Every 20 trials, the current state of the system was stored and used to create the results that follow.

SNN parameters were initial hidden layer nodes = 9, $a = 0.3$, $b = 0.05$, $c = 0.0$, $c_{ini} = 0.5$, $\theta y = 1.0$, output window size = 21, $LS = 3$, $\theta LS = 4$. In memristive networks, all connections were memristive with $L = 1000$. Memristor parameters were $R_{\text{on}} = 0.01$, $R_{\text{off}} = 1$, static β=1, $\beta_{\text{min}} = 1$, $\beta_{\text{maxHP}} = 100$, $\beta_{\text{maxPEO-PANI}} = 100$ $q_{\text{min}} = 0.0098$. RSM parameters were $S_n\text{MIN} = 2$, $S_n\text{MAX} = 6$.

To facilitate useful comparisons, we defined a notion of "performance". As the start location was tightly constrained, we say the performance of the system is equal to the first trial in which the goal state is found, so that a lower value indicated higher performance. This measure allowed us to perform t-tests to compare the respective performances of the four systems. In the following tables, "Performance" was the average performance as outlined above. "High fitness" refers to the average fitness of the highest-fitness network in each run. "Neurons" were the average final connected neurons per network in the population and "Connectivity" was the average percentage of enabled connections in the population.

5.1 Test Environment

Our chosen robotics simulator was the Webots platform [32], a test bed that is popular amongst the research community. The environment was modelled as an enclosed (walled) arena with coordinates ranging from $[-1, 1]$ in both x and y directions; the boundary walls had height $z = 0.15$. A three-dimensional box was placed centrally, with vertices on "ground level" at ($x = -0.4$, $y = -0.4$), (-0.4, 0.4), (0.4, 0.4), and (0.4, -0.4), and raised to a height of $z = 0.15$. A light source, modelled on a 15 Watt bulb, was placed at the top-right hand corner of the arena ($x = 1$, $y = 1$). The agent initially faces North, and its initial start position was constrained to the range $x + y < -1.5$. The agent must traverse the environment and approach the light source to receive reward. The environment is shown in Fig. 6. When the agent reached the goal state (where $x + y > 1.6$), the responsible network received

Table 1 Detailing averages and standard deviations for common parameters of MEM, RSM, HP, PEO, LIN, and GA systems on the test problem

Synapse	Perf	High fit	Avg fit	Nodes	Conns (%)
MEM	38.4 (72.2)	11468 (345)	11171 (497)	16.71 (0.47)	50.52 (3.8)
RSM	2.9 (6.2)	11712 (364)	11280 (571)	16.67 (0.49)	48.53 (4.01)
HP	526.1 (992.4)	10660 (2280)	9477 (3333)	16.68 (1.74)	49.22 (9.61)
PEO	17 (34.6)	11581 (303)	11454 (319)	17.05 (0.1)	48.86 (4.63)
LIN	14.7 (32.5)	11363 (398)	11058 (728)	16.89 (0.54)	51.06 (4.06)
GA	77.6 (130.0)	11420 (423)	11402 (277)	17.10 (0.7)	48.24 (5.58)

a constant fitness bonus of 2500, which was added to the fitness function f outlined in (22). The denominator in the equation expresses the difference between the position of the goal state (1.6) and the current agent position (a_x and a_y), and st is the number of timesteps taken to solve. The minimum value of this function is capped so that $f > 0$. The fitness of an agent is calculated at the end of every timestep, with the highest attained value of f during the trial kept as the fitness value for that network. Optimal performance gives $f = 11800$, which corresponds to 700 timesteps from start to goal state with no collisions.

$$f = \big(1/\big(1.6 - (|a_x + a_y|)\big)\big) \times 1000 - st \tag{22}$$

5.2 Results

5.2.1 Performance

Averages and standard deviations for all network types are given in Table 1. T-tests, given in Tables 2 and 3, show a number of promising results. RSM networks had higher "performance" than all other network types with the exception of LIN ($p < 0.05$). MEM networks achieved higher "performance" than HP networks ($p < 0.05$) but trailed behind RSM networks. This indicates that variable RSMs are more forgiving in terms of network composition when inducing goal-finding behaviour in the networks. RSM networks also possessed significantly higher fitness solutions than HP, LIN, GA and MEM networks ($p < 0.05$), via the ability to rapidly vary synaptic weighting in addition to the three action generation mechanisms outlined in Sect. 5.2.2. All other network types relied on numerous repeated STDP events of a particular polarity to provide large increases in weight, whereas the RSM could switch back and forth multiple times in a short number of time steps e.g. more expedient binary switching allows for additional freedom in possible generated behaviours.

Figure 7(a) shows that MEM/RSM networks are at least competitive with (and usually superior to) all other network types in terms of performance and final fitness.

Table 2 Detailing performance t-test results (p values) for common parameters comparing MEM networks to all other network types in the experiment

vs.	Performance	High fit	Nodes	Conn.	ψ	ω	τ
RSM	**0.012**	**0.014**	0.7	**0.047**	0.949	**0.008**	**<0.001**
HP	**0.012**	0.016	0.919	0.462	0.939	0.115	0.08
PEO	0.173	0.165	**0.001**	0.189	0.112	0.094	0.461
LIN	0.119	0.301	0.295	0.566	**0.001**	**0.039**	0.882
GA	0.196	0.61	**0.009**	0.076	**0.004**	0.064	0.542

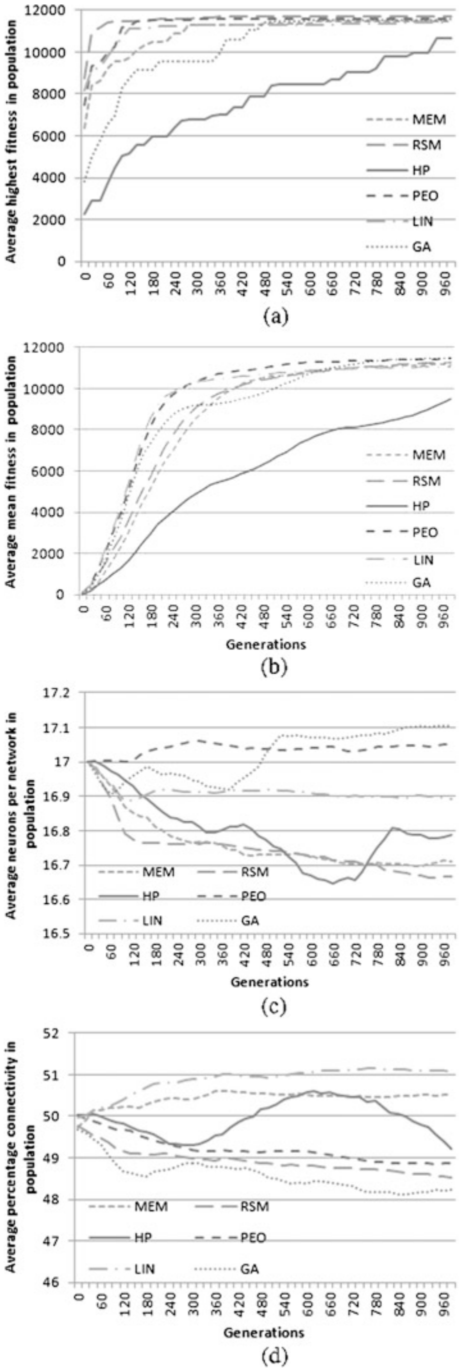
Table 3 Detailing performance t-test results (p values) for common parameters comparing RSM networks to all other network types in the experiment

vs.	Performance	High fit	Nodes	Conn.	ψ	ω	τ
HP	**0.007**	**0.02**	0.975	0.698	0.975	0.415	**<0.001**
PEO	**0.041**	0.118	**0.001**	0.761	0.187	0.55	**<0.001**
LIN	0.067	**0.002**	0.202	**0.019**	**0.003**	0.184	**<0.001**
GA	**0.004**	**0.014**	**0.01**	0.820	**0.023**	0.336	**<0.001**

This is an encouraging result considering the increased search/behaviour space the SA discovery component has to deal with, especially in the case of MEM networks and β, as it indicates that the variable RM induces no significant performance overhead. MEM networks comprised less nodes than the best static memristor (PEO) ($p < 0.05$), and RSM networks comprised less neurons than their static counterpart GA networks ($p < 0.05$), see Fig. 7(c). A possible explanation is that the higher degrees of freedom afforded to the variable synapse imparts increased computational power and allowing for a more compact representation. RSM networks were also had significantly sparser connectivity patterns than MEM networks—see Fig. 7(d). This provides some insight into how the functionality of the two devices differ: both provide a compact neural representation, however only RSM networks couple this with a compact synaptic representation. In the case of MEM networks, denser synaptic topologies are required as more of the computational power of the network being embodied in the synapse itself (a notion echoed in recent literature [1]). Comparing a single MEM synapse to a single RSM synapse, we note that the MEM device is capable of more complex, history-dependent temporal behaviour. However the RSM network as a whole overcomes this via arrangements of simpler devices and synchronised LRS/HRS switching in conjunction with excitatory/inhibitory neural spikes.

Considering possible hardware implementations, CMOS neurons are larger and more complex than the synapses that connect them. As neuron numbers are more likely to be a constraint, a reduction in neurons despite increased connectivity (e.g. in the case of MEM networks) can be said to be beneficial.

Fig. 7 Average (**a**) highest fitness (**b**) mean fitness (**c**) connected hidden layer nodes (**d**) enabled connections for the experiment



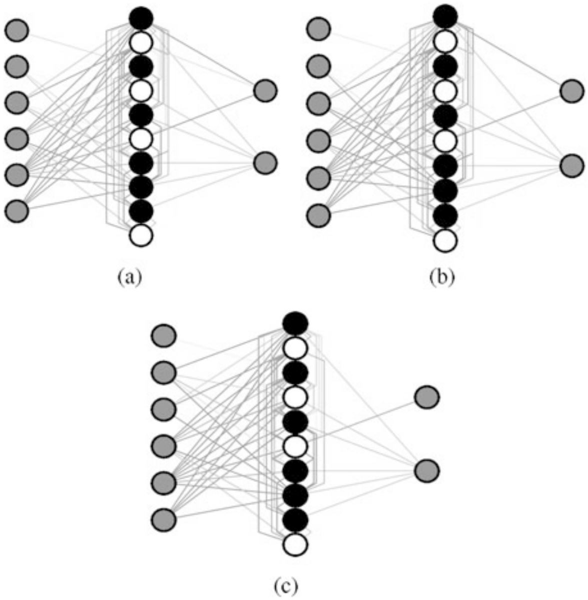


Fig. 8 Showing STDP activity in the best RSM network (**a**) before (**b**) during (**c**) after turn. In the hidden layer, light coloured neurons are excitatory and dark neurons are inhibitory

5.2.2 Behaviour

In solving the test problem, two general high-fitness strategies were employed by the MEM and RSM networks. The first involved a chain of "forwards" actions, a number of "turn right" actions as the agent circumvented the obstacle, and finished with successive "forwards" actions until the goal state was reached. The second strategy was a mirror of the first, but passing below the obstacle and turning left. In either case, STDP was harnessed to turn the agent. HP-governed MEM profiles were found to quickly reduce spike weights incoming to the left (right) motor, causing perturbation of calculated action during turn by bringing that motor below the "high activated" threshold. PEO-PANI-governed MEM profiles to the same motor were used to swiftly increase the level of spiking activity (usually in response to a light sensor surpassing some threshold) until a "forwards" action was calculated after the turn was completed.

RSM networks used the rapid-switching ability of the synapse in three main ways (i) to perform additional "connection selection" in-trial e.g. to switch a synapse to a given state and leave it there (ii) as (i) but varying the connectivity map of the network multiple times based on the sensory input (iii) in the creation of weight oscillators in the network, whereby the firing on the neurons and switching of the synapses synchronised through time to generate appropriate output actions from a subgroup of neurons. In the latter case, the input state was found to perturb both the firing pattern of the neurons and weight-switching pattern of the synapses to generate e.g. turning actions when required. Use of STDP by the best RSM network is shown in Fig. 8.

Table 4 Detailing averages and standard deviations for common mutation parameters of MEM, RSM, HP, PEO, LIN and GA systems on the test problem

	μ	ψ	ω	τ	ι
MEM	NA	0.072 (0.02)	0.085 (0.03)	0.011 (0.01)	0.023 (0.02)
RSM	NA	0.071 (0.03)	0.105 (0.02)	0.014 (0.01)	0.033 (0.02)
HP	NA	0.073 (0.04)	0.122 (0.11)	0.022 (0.03)	NA
PEO	NA	0.062 (0.02)	0.113 (0.07)	0.01 (0.01)	NA
LIN	NA	0.055 (0.02)	0.139 (0.11)	0.01 (0.01)	NA
GA	0.018 (0.01)	0.056 (0.03)	0.122 (0.09)	0.011 (0.01)	NA

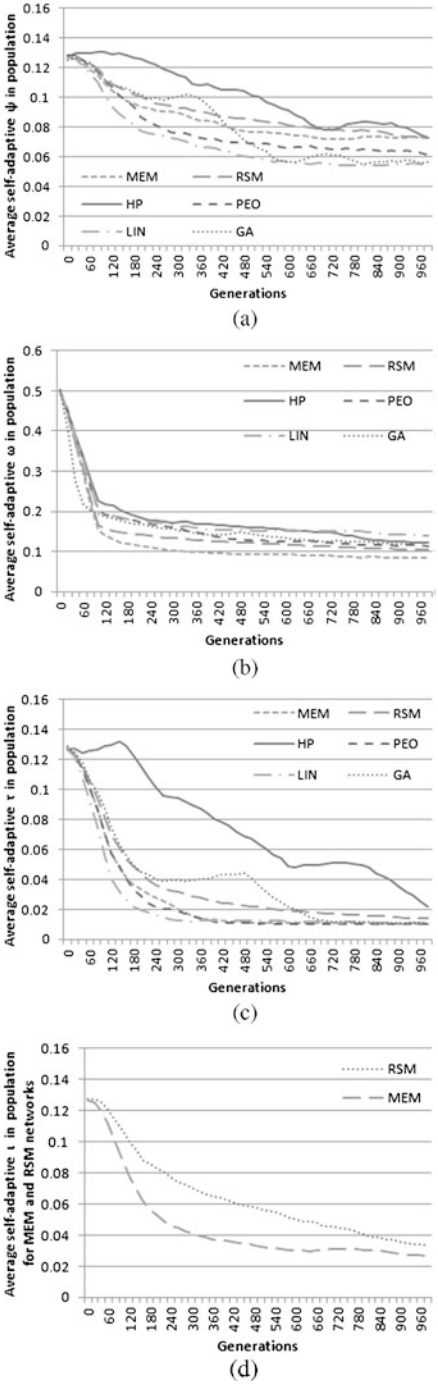
5.2.3 Self-adaptive Parameters

Self-adaptive parameter results can be seen in Table 4 and Figs. 9(a)–(d); μ was not compared as it was only used in GA networks. Tables 2 and 3 reveal that MEM and RSM networks had statistically higher ψ (see Fig. 9(a))—which governed the frequency of neuron addition/removal events—than LIN and GA networks ($p < 0.05$). Combined with the fact that both RSM and MEM networks had lower ω (see Fig. 9(b))—which governed the rate of neuron addition—than the other network types, this allowed the networks more opportunities (ψ) to remove neurons from their networks (ω) and accounts for RSM and MEM networks having few neurons per network. We also note that RSM networks possessed statistically different τ (rate of Connection Selection) than other network types ($p < 0.05$ in all cases). Between MEM and RSM networks, ι (the rate of altering the variable component of the synapse) was statistically higher for RSM networks ($p < 0.05$, see also Fig. 9(d)), highlighting how the two synapse types differ in functionality.

5.2.4 Evolution of β / S_n

MEM As β varied between 1–101 in the case of variable HP-governed profiles and 1–100 in the case of variable PEO-PANI-governed profiles, the total range of β was 199, where a value between 1–101 was considered a HP-governed profile and anything >101 a PEO-PANI-governed profile. Analysis of the MEM networks revealed that the connections to the motor on the side that made the turn evolved less linear profiles, allowing for quicker action switching behaviour. In addition, connections between the input and hidden layer had a lower average maximum (145.98 vs. 150.11) and higher average minimum (50.945 vs. 27.533) than those between the hidden and output layer. This suggests that connections to motors in general were evolutionarily preferred to have more nonlinear, strongly varying conductance profiles. Connections in both of these areas had higher average maximum and lower average minimum β values than connections within the hidden layer (116.16 and 84.1 respectively), suggesting that more steady behaviour was preferred there.

Fig. 9 Average (**a**) ψ (**b**) ω (**c**) τ (**d**) ι for the experiment



RSM The numbers of connections belonging to each S_n within the networks were approximately even (all $p > 0.05$). Some differences were observed regarding the specific placement of synapses—for example, more sensitive ($S_n = 2$ or 3) synapses were frequently found connected to IR sensors, with $S_n = 3$ or 4 the most common synapse connected to light sensors. This suggests that fast-switching synapses are required to immediately generate activity within the network. IR sensors have lower S_n than light sensors as they trigger only when near obstacles and so must be able to quickly switch to perturb network output and avoid the obstacle. Synapses were more frequently connected to light sensors than IR sensors, reinforcing the idea that light sensors were responsible for maintaining activity within the network, which was then modified (frequently by IR sensors, but also via light sensors exceeding some threshold) to alter behaviour as required. Due to the differing implementation of STDP for RSM networks, no significant differences were observed with respect to the type of neuron (excitatory or inhibitory) the synapses were presynaptic/postsynaptic to (p-values range from 0.11–0.76).

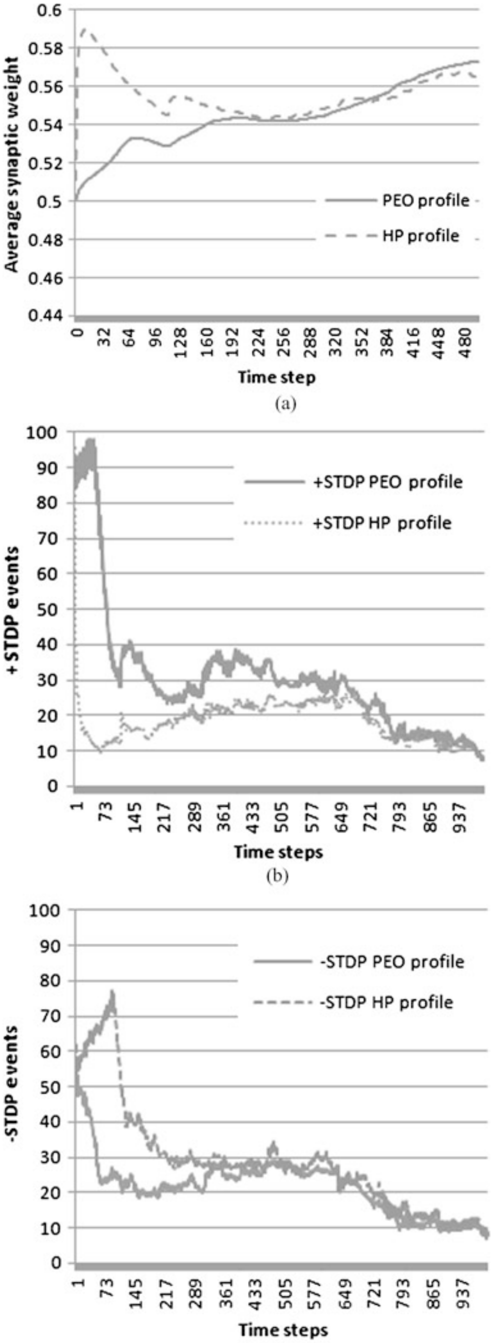
STDP Here we average the best network per run rather than average all networks. Synaptic weights in MEM networks are seen to generally rise (Fig. 10(a)). Higher MEM weights are eventually achieved by the facilitating PEO-PANI-governed profiles—see Fig. 10(a) after 400 time steps. RSM networks experience a gradual drop in weight due to being initialised in the LRS (Fig. 11(a)), the most pronounced of which coincides with an increase in switching activity. Both network types possess statistically similar ($p > 0.05$) average synaptic weight after 500 timesteps.

STDP is used in different ways by the two network types. In MEM networks, PEO-governed profiles experience statistically more positive STDP and statistically less negative STDP (both $p < 0.05$) than HP-governed profiles. These findings seem to concur with previous work [21] in which static PEO memristors were found to be more suited to facilitating (conducting) weight through the network (with the opposite being true for HP memristors). As a result, the evolutionary process harnessed the differing profiles by placing PEO memristors where they would receive the most positive STDP, etc.

In contrast to MEM networks, RSM networks experience a gradual increase in rate of STDP throughout the lifetime of the network. A distinct increase in STDP rate can be seen at time step 120 in Fig. 11(a), which corresponds to the time at which the network must make the agent turn around the obstacle. Figure 11(b) provides a more detailed view of the contribution to this jump in activity by each S_n. With the exception of $S_n = 5$–$S_n = 6$, all S_n are experience statistically different amounts of switching ($p < 0.05$). Lower S_n synapses are also present more variable STDP profiles. Overall, less total STDP events occur in RSM networks than MEM networks, presumably because (i) consecutive STDP events are more difficult to attain (ii) each switch can have a more dramatic effect on the activity of the network.

In summary, both MEM and RSM networks show evidence that the evolutionary mechanism differentiates between, and exploits, the differing resistance profiles and STDP behaviours of the synapses.

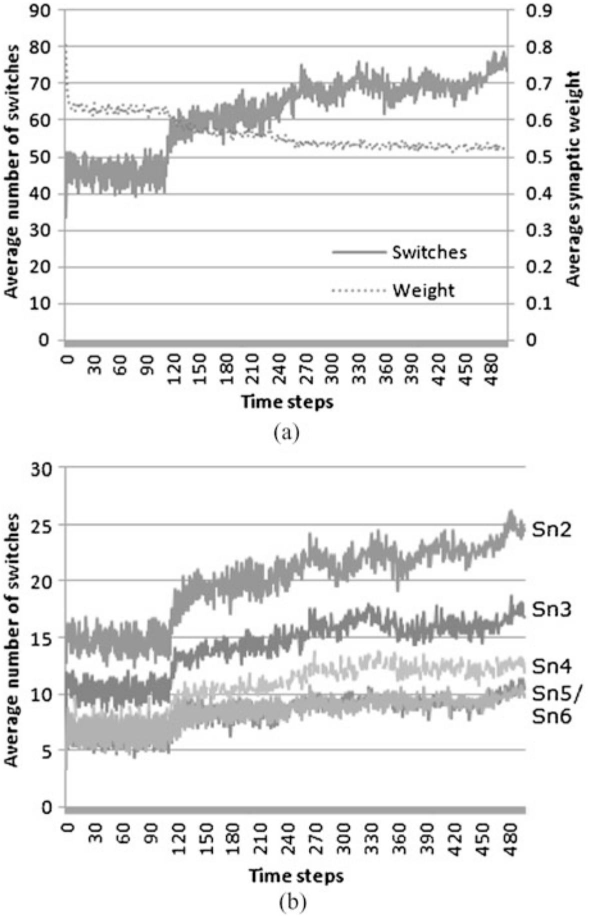
Fig. 10 (**a**) average synaptic weight (**b**) average positive STDP (**c**) average negative STDP per profile type in the MEM networks



(a)

(b)

(c)



(a)

(b)

Fig. 11 (**a**) average synaptic weight/average switch frequency (**b**) average switch frequency per S_n in the RSM networks

6 Conclusions

In this study we have analysed two types of variable RM synapse and compared to static RMs and benchmark connections in a simulated robotics environment. Our hypothesis was that the additional degrees of functional freedom afforded to the variable RMs allowed them to outperform these other synapses in key areas. Numerous findings supported this hypothesis, including (i) variable memristors having more expedient goal-finding behaviour than HP networks (ii) variable RSMs having significantly better "Performance" than PEO and GA networks, as well as better quality solutions than HP, LIN and GA networks (iii) both variable synapses providing a more compact network representation than PEO and GA networks.

These findings suggest that self-adaptation of the characteristic resistance profile of both variable RMs is harnessed by the evolutionary process to provide variable plastic networks with more implicit degrees of freedom than the other network types. Variable plasticity was harnessed via STDP to achieve more expedient goal-finding behaviour, higher quality solutions, and reduced topological complexity when compared to certain other network types. Importantly, the need to explore additional search space (especially in the case of β) was found to be non-disruptive (and in most cases beneficial) with respect to network performance, whilst providing a more flexible synaptic representation.

The inclusion of self-adaptive mutation parameters with a neuro-evolutionary approach is likely to be necessary for the autonomous emergence of NM processing units. This study presents a candidate implementation that allows for the formation of such (task-specific) neural groupings.

Results enforce the view that this kind of approach may be utilised to guide the synthesis requirements of functional memristor/RSM hardware systems. Trials on different task types may provide insights into the optimal composition of such systems on a per-task basis.

Possible future research directions include hardware and mixed-media implementations, provided the two RM types can be integrated into the same circuit architecture. The main benefit of memristive STDP over other STDP implementations lies in hardware realisation, as the history-dependent weight of the synapse is stored in the nonvolatile physical state of the device and thus does not require simulation. Similarly, RSM state is nonvolatile. Titanium dioxide additionally allows for memristive behaviour and binary switching to be elicited from the same material. As well as providing more functional degrees of freedom to the synapse, evolution could control switching between the behaviours to autonomously create task-optimal NM subarchitectures, as well as online synaptic transformations via targetted irradiation for self-repair or self-modification.

Acknowledgements This work was funded by EPSRC grant number EP/H014381/1.

References

1. Abbott, L., Regehr, W.: Synaptic computation. Nature **431**, 796–803 (2004)
2. Afifi, A., Ayatollahi, A., Raissi, F.: Stdp implementation using memristive nanodevice in cmos-nano neuromorphic networks. IEICE Electron. Express **6**(3), 148–153 (2009)
3. Akinaga, B.H., Shima, H.: Resistive random access memory (reram) based on metal oxides. Proc. IEEE **98**(12), 2237–2251 (2010)
4. Chua, L.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971). doi:10.1109/TCT.1971.1083337
5. Chua, L., Sbitnev, V., Kim, H.: Hodgkin Huxley axon is made of memristors. Int. J. Bifurc. Chaos **22**(03), 1230011 (2012)
6. Dolan, C.P., Dyer, M.G.: Toward the evolution of symbols. In: Grefenstette, J.J. (ed.) Genetic Algorithms and Their Applications (ICGA'87), pp. 123–131. Erlbaum, Hillsdale (1987)
7. Doolittle, W., Calley, W., Henderson, W.: Complementary oxide memristor technology facilitating both inhibitory and excitatory synapses for potential neuromorphic computing applications. In: Semiconductor Device Research Symposium, 2009. ISDRS'09. International, pp. 1–2 (2009)
8. Durr, P., Mattiussi, C., Soltoggio, A., Floreano, D.: Evolvability of neuromodulated learning for robots. In: Proceedings of the 2008 ECSIS Symposium on Learning and Adaptive Behavior in Robotic Systems, pp. 41–46. IEEE Computer Society, Los Alamitos (2008)

9. Erokhin, V.: On the learning of stochastic networks of organic memristive devices. Int. J. Unconv. Comput. **9**(3–4), 303–310 (2013)
10. Erokhin, V., Fontana, M.P.: Electrochemically controlled polymeric device: a memristor (and more) found two years ago (2008). arXiv:0807.0333
11. Erokhin, V., Howard, G.D., Adamatzky, A.: Organic memristor devices for logic elements with memory. Int. J. Bifurc. Chaos **22**(11) (2012)
12. Floreano, D., Dürr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. Evol. Intell. **1**, 47–62 (2008)
13. Gale, E.: The memory-conservation model of memristance. In: Heber, J., Schlom, D., Tokura, Y., Waser, R., Wutting, M. (eds.) Technical Digest of Frontiers in Electronic Materials. Nature Conferences, pp. 538–539. Wiley VCH, New York (2012)
14. Gale, E., de Lacy Costello, B., Adamatzky, A.: The effect of electrode size on memristor properties: an experimental and theoretical study. In: International Conference in Electronics Design, Systems and Applications (ICEDSA 2012) (2012, in press)
15. Gerstner, W., Kistler, W.: Spiking Neuron Models—Single Neurons, Populations, Plasticity. Cambridge University Press, Cambridge (2002)
16. Hebb, D.O.: The Organisation of Behavior. Wiley, New York (1949)
17. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
18. Hosoi, Y., Tamai, Y., Ohnishi, T., Ishihara, K., Shibuya, T., Inoue, Y., Yamazaki, S., Nakano, T., Ohnishi, S., Awaya, N., et al.: High speed unipolar switching resistance ram (rram) technology. Int. Electron Devices Meet. **1**, 1–4 (2006)
19. Howard, G.D.: Constructivist and spiking neural learning classifier systems. PhD thesis, University of the West of England (2010)
20. Howard, G.D., Bull, L., Adamatzky, A.: Cartesian genetic programming for memristive logic circuits. In: Moraglio, A., Silva, S., Krawiec, K., Machado, P., Cotta, C. (eds.) Proceedings of the 15th European Conference on Genetic Programming, EuroGP 2012. LNCS, vol. 7244, pp. 37–48. Springer, Malaga (2012)
21. Howard, G., Gale, E., Bull, L., de Lacy Costello, B., Adamatzky, A.: Evolution of plastic learning in spiking networks via memristive connections. IEEE Trans. Evol. Comput. **16**(5), 711–729 (2012)
22. Howard, G.D., Bull, L., Costello, B.D.L., Adamatzky, A., Erokhin, V.: Creating unorganised machines from memristors. Int. J. Appl. Math. Inf. Sci. (2013, in press)
23. Howard, G.D., Bull, L., Costello, B.D.L., Gale, E., Adamatzky, A.: Evolving spiking networks with variable resistive memories. Evol. Comput. (2013, in press)
24. Hurst, J., Bull, L.: A neural learning classifier system with self-adaptive constructivism for mobile robot control. Artif. Life **12**(3), 353–380 (2006)
25. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. **10**(4), 1297–1301 (2010)
26. Kistler, W.M.: Spike-timing dependent synaptic plasticity: a phenomenological framework. Biol. Cybern. **87**(5–6), 416–427 (2002)
27. Linares-Barranco, B., Serrano-Gotarredona, T.: Memristance can explain spike-time-dependent-plasticity in neural synapses. Nature Precedings (2009)
28. Liu, S., Wu, N., Ignatiev, A.: Electric-pulse-induced reversible resistance change effect in magnetoresistive films. Appl. Phys. Lett. **76**(19), 2749–2751 (2000)
29. Maass, W.: Networks of spiking neurons: the third generation of neural network models. Neural Netw. **10**, 1659–1671 (1996)
30. Maass, W., Zador, A.M.: Dynamic stochastic synapses as computational units. Neural Comput. **11**(4), 903–917 (1999)
31. Mead, C.: Neuromorphic electronic systems. Proc. IEEE **78**(10), 1629–1636 (1990)
32. Michel, O.: Webots™: professional mobile robot simulation. Int. J. Adv. Robot. Syst. **1**(1), 39–42 (2004)
33. Mouttet, B.: Memristor pattern recognition circuit architecture for robotics. In: Proceedings of the 2nd International Multi-Conference on Engineering and Technological Innovation II, pp.

65–70 (2009)

34. Nolfi, S., Floriano, D.: Evolutionary Robotics. MIT Press, Cambridge (2000)
35. Pershin, Y.V., La Fontaine, S., Di Ventra, M.: Memristive model of amoeba learning. Phys. Rev. A **80**(2), 021926 (2009). doi:10.1103/PhysRevE.80.021926
36. Querlioz, D., Bichler, O., Gamrat, C.: Simulation of a memristor-based spiking neural network immune to device variations. In: Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN), pp. 1775–1781. IEEE Press, San Jose (2011)
37. Rechenberg, I.: Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann-Holzboog (1973)
38. Rocha, M., Cortez, P., Neves, J.: Evolutionary neural network learning. In: Progress in Artificial Intelligence. Lecture Notes in Computer Science, vol. 2902, pp. 24–28. Springer, Berlin (2003)
39. Rumelhart, D., McClelland, J.: Parallel Distributed Processing, vols. 1 & 2. MIT Press, Cambridge (1986)
40. Saggie-Wexler, K., Keinan, A., Ruppin, E.: Neural processing of counting in evolved spiking and McCulloch-Pitts agents. Artif. Life **12**(1), 1–16 (2006)
41. Snider, G.: Computing with hysteretic resistor crossbars. Appl. Phys. A **80**, 1165–1172 (2005)
42. Snider, G.: Spike-timing-dependent learning in memristive nanodevices. In: IEEE International Symposium on Nanoscale Architectures, 2008. NANOARCH 2008, pp. 85–92 (2008). doi:10.1109/NANOARCH.2008.4585796
43. Soltoggio, A.: Neural plasticity and minimal topologies for reward-based learning. In: Proceedings of the 2008 8th International Conference on Hybrid Intelligent Systems, pp. 637–642. IEEE Computer Society, Washington (2008)
44. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
45. Strukov, D.B., Borghetti, J.L., Williams, R.S.: Coupled ionic and electronic transport model of thin-film semiconductor memristive behavior. Small **5**(9), 1058–1063 (2009)
46. Urzelai, J., Floreano, D.: Evolution of adaptive synapses: robots with fast adaptive behavior in new environments. Evol. Comput. **9**, 495–524 (2001)
47. Vujisic, M., Stankovic, K., Marianovic, N., Osmokrovic, P.: Simulated effects of proton and ion beam irradiation on titanium dioxide memristors. IEEE Trans. Nucl. Sci. **57**(4), 1798–1804 (2010)
48. Wang, X., Chen, Y.: Spintronic memristor devices and application. In: Proceedings of the Conference on Design, Automation and Test in Europe, European Design and Automation Association, pp. 667–672 (2010)
49. Waser, R., Aono, M.: Nanoionics-based resistive switching memories. Nat. Mater. **6**(11), 833–840 (2007)
50. Xia, G., Tang, Z., Li, Y., Wang, J.: A binary Hopfield neural network with hysteresis for large crossbar packet-switches. Neurocomputing **67**, 417–425 (2005)

Behavior of Multiple Memristor Circuits

Ram Kaji Budhathoki, Maheshwar P. Sah, Shyam Prasad Adhikari, Hyongsuk Kim, and Leon Chua

Abstract Memristor is a new circuit element defined by a state-dependent Ohm's law between the memristor voltage and current. It has recently been successfully built, however, its electrical characteristics are not fully known yet. Like other circuit elements R, L and C, there could have various configurations of multiple memristors including serial and parallel connections in a variety of applications. When input voltage/current is supplied to a circuit with multiple memristors, behavior of the device becomes complicated and is difficult to predict. In this chapter, composite characteristics of the serial and parallel connections of memristors are investigated using both linear and nonlinear models. Also, the behavior of individual memristor is formulated mathematically and a general computation method of composite memristance for multiple memristor circuits of diverse configurations is proposed.

1 Introduction

In 1971, Leon O. Chua postulated the memristor as the fourth basic circuit element [1], based on a nonlinear relationship between charge and flux. Chua and Kang later extended the idea of memristors to memristive systems and devices in 1976 [2].

R.K. Budhathoki · M.P. Sah · S.P. Adhikari · H. Kim (✉)
Intelligent Robots Research Center, Division of Electronics and Information Engineering, Chonbuk National University, Jeonju 561-756, Republic of Korea
e-mail: hskim@jbnu.ac.kr

R.K. Budhathoki
e-mail: ramkaji@gmail.com

M.P. Sah
e-mail: maheshwarsah@hotmail.com

S.P. Adhikari
e-mail: all.shyam@gmail.com

L. Chua
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA
e-mail: chua@eecs.berkeley.edu

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_15,

Memristor relates charge q and magnetic flux φ in a circuit. Memristance (resistance of memristor) M can be controlled by applying a voltage or a current signal across the memristor, where $R = M(q) = d\varphi/dq$. When a sinusoidal, or any bipolar periodic signal that assumes both positive and negative values, is applied to the memristor, it exhibits a hysteresis loop in the v–i plane which is pinched at the origin. This pinched hysteresis loop is considered as a fingerprint of the memristor [3, 4]. The area of the pinched hysteresis loop decreases with the frequency ω of the input signal, and tends to a straight line as $\omega \to \infty$, for all bipolar periodic signals and for all valid initial conditions.

An actual physical memristor was realized as a TiO_2 nano-component by researchers at HP in 2008 [5]. It consists of two layer thin film (size $D = 10$ nm) of TiO_2, sandwiched between platinum contacts. One of the TiO_2 layer is doped with oxygen vacancies and acts like a semiconductor, whereas another undoped layer acts as a resistor. The total memristance is a sum of the resistances of the doped and undoped regions.

Characteristics of memristor are reviewed, and a mathematical and a SPICE model are provided by several research groups ([6–9]). Its emulators and macro models have been proposed ([10–12]). Also, the behavior of memristor-capacitor, memristor-inductor and memristor-capacitor-inductor circuits is presented [13].

When multiple memristors are connected together, the behavior of the composite device becomes complicated, and is difficult to predict, due to the polarity dependent nonlinear variation of memristance of individual memristors. So it is vital to understand the characteristics and behavior, when memristors are connected in different topologies.

If two memristors with opposite polarities are combined together, the nonlinearity of memristance is reduced dramatically due to the complementary action of two memristors. Kim et al. presented an efficient weighting circuit for artificial neural networks by building a memristor-based bridge structure [14, 15]. M.P. Sah et al. implemented bridge synaptic circuit with the memristor emulator [16]. E. Linn et al. reported a fabrication result of complementary resistive switch (CRS) consisting of two back-to-back (antiserial) memristive elements for the construction of large passive crossbar arrays by solving the sneak path problem [17]. Later, the CRS architecture has been further investigated via an analytical approach [18].

T. Liu et al. also reported the i–v characteristics of antiparallel resistive switches (APRS) that strongly depend on the parameters of the individual switches [19].

In the previous works, the principles of the composite behaviors of the multiple memristor circuits are not fully explained. Aiming to explain such behavior analytically, the relationship among flux, charge and memristance of diverse composite memristors are investigated, using both linear and nonlinear models, and simulations of the memristors' characteristics are presented in this chapter. The voltage-current (v–i) graphs for different connections, and the variations of overall memristance, are also analyzed.

Rest of the chapter is organized as follows. In Sect. 2, the basic principle and the mathematical model of the titanium dioxide memristor is introduced considering both the linear and nonlinear model. In Sect. 3, the stable and transient composite memristance state of memristors is discussed. In Sect. 4, series configurations of memristors with two different polarities are discussed. Similarly in Sect. 5, parallel configurations of memristors with two different polarities are discussed. In Sect. 6, simulation results of the composite flux and composite memristance of the single, series and parallel connections of memristors, for both linear and nonlinear models, are presented. Concluding remarks are presented in Sect. 7, followed by References.

2 Single Memristor Circuit

The constitutive relation of a memristor is defined by a nonlinear algebraic relation between its charge q and its flux φ, namely,

$$\varphi = \hat{\varphi}(q) \tag{1}$$

where, charge is defined as the integral of current,

$$q(t) \triangleq \int_{-\infty}^{t} i(\tau)d\tau \tag{2}$$

and flux is defined as the integral of voltage,

$$\varphi(t) \triangleq \int_{-\infty}^{t} v(\tau)d\tau \tag{3}$$

Taking the derivative on both sides of Eq. (1), we obtain

$$\frac{d\varphi}{dt} = \frac{d\hat{\varphi}(q)}{dq} \cdot \frac{dq}{dt}. \tag{4}$$

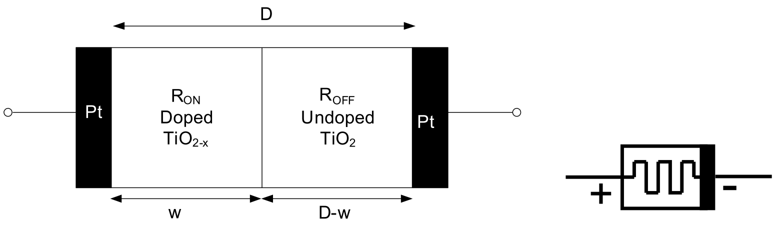
It leads to

$$v(t) = \frac{d\hat{\varphi}(q)}{dq} i(t) \equiv M(q)i(t) \tag{5}$$

where, $M(q)$ is a charge controlled memristance, defined as,

$$M(q) = \left.\frac{d\hat{\varphi}(q)}{dq}\right|_{q=q_Q}. \tag{6}$$

From Eq. (6), $M(q)$ can be interpreted as the slope at an operating point $q = q_Q$ at time t on the memristor φ–q curve. Since the memristance depends on the operating point $q = q_Q$ and $q = q_Q$ remains fixed when $v(t) = 0$ and $i(t) = 0$, the device can be used as nonvolatile memory. Thus, the resistance $M(q)$ is called the memristance, an acronym for memory resistance.



(a) (b)

Fig. 1 (**a**) TiO$_2$ Memristor of length D. (**b**) Memristor symbol

2.1 Linear Model

Figure 1(a) shows the structure of the TiO$_2$ memristor, and the symbol of a memristor is shown in Fig. 1(b), where the negative end is labeled black. In the TiO$_2$ memristor, a thin undoped titanium dioxide (TiO$_2$) layer and a thin oxygen-deficient doped titanium dioxide (TiO$_{2-x}$) layer are sandwiched between two platinum electrodes. When a voltage (or current) is applied to the device, the width of the TiO$_2$ and TiO$_{2-x}$ layer changes as a function of the applied voltage (or current). As a result, the resistance between the two electrodes is altered.

Let D and w denote the thickness of the sandwiched area and the doped area (oxygen deficient area) in the TiO$_2$ memristor, respectively, and let R_{ON} and R_{OFF} denote the resistances at high ($w/D = 0$) and low dopant ($w/D = 1$) concentration areas, respectively.

Then, the voltage current relationship is given as,

$$v(t) = \left(R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right) \right) i(t) \tag{7}$$

and the state variable w is defined by,

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{ON}}{D} i(t) \tag{8}$$

where, $\mu_v$is the dopant mobility. This model is called a linear drift model, as the drift of the state variable (w) is linearly proportional to the current. Integrating (8) on both sides, we obtain,

$$w(t) = \mu_v \frac{R_{ON}}{D} \int_0^t i(\tau) d\tau + w_0. \tag{9}$$

It follows from (7) that the memristance $M(t)$ can be expressed as,

$$M(t) = R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right). \tag{10}$$

Substituting (9) into (10), we get

$$M(t) = \left\{ R_{OFF}\left(1 - \mu_v \frac{R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)\int_0^t i(\tau)d\tau\right) + \frac{w_0 R_{OFF}}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right\}. \tag{11}$$

As memristance is defined in terms of flux and charge in (6), integrating both sides of (11) yields,

$$\varphi(t) = R_{OFF}\left[q(t)\left(1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right) - \mu_v \frac{R_{ON}}{2D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)q(t)^2\right] \tag{12}$$

where, $\varphi(t)$ and $q(t)$ denote the *flux* and *charge*, respectively, at time t. Thus, the memristance can be interpreted as the *slope* at the operating point $q = q_Q$ at time t on the memristor φ–q curve. If the φ–q curve is nonlinear, the memristance will vary with the operating point. From (12), we obtain

$$M = \frac{d\varphi}{dq} = R_{OFF}\left\{\left[1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right] - \mu_v \frac{R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)q(t)\right\} \tag{13}$$

From (13), we see that the memristance is a linear function of the charge $q(t)$, where the theoretical range of φ, q and M of the TiO$_2$ memristor are $\varphi = [0, 0.805]$ Weber, $q = [0, 1 \times 10^{-4}]$ Coulomb and $M = [16, 0.1]$ KΩ, respectively for the range $w = [0, D]$, when $\mu_v = 10^{-10}$ cm^2/V s and $D = 10$ nm. From (7) and (11), the relationship between the memristor voltage and current is given by,

$$v(t) = \left\{ R_{OFF}\left(1 - \mu_v \frac{R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)\int_0^t i(\tau)d\tau\right) + \frac{w_0 R_{OFF}}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right\} i(t) \tag{14}$$

2.2 Nonlinear Model

The linear-drift model describes many of the salient features of a memristor. However, it does not take into account the boundary effects. Qualitatively the boundary between the doped and undoped regions moves with speed μ_v in the bulk of the memristor, but that speed is strongly suppressed when it approaches either edge $w \sim 0$ or $w \sim D$. The nonlinear phenomena often appear at the boundaries of nano scale devices. With even the small voltage applied across nanometer devices, a large electric field is produced and therefore, the ion boundary moves in a nonlinear fashion in nanoscale devices [13]. This nonlinear phenomenon can be modeled by multiplying the right side of (8) by a window function $F_p(w)$, namely,

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{ON}}{D} i(t) F_p(w) \tag{15}$$

where p is a parameter whose value is a positive integer. As the integer p increases, the model tends to a linear model. This is called the nonlinear drift model (or nonlinear model). Several nonlinear memristor models with different window functions have been proposed. One reasonable window function is defined by [13],

$$F_p(w) = 1 - \left(2\frac{w}{D} - 1\right)^{2p} \tag{16}$$

It is difficult to find the solution satisfying both (15) and (16) analytically. However, $w(t)$ can be computed numerically as,

$$w(t + \Delta t) = \mu_v \frac{R_{ON}}{D}\left(1 - \left(2\frac{w}{D} - 1\right)^{2p}\right)\Delta q + w(t) \tag{17}$$

where Δq is the charge increment fed to the memristor during the time interval Δt and computed by integrating the input current as,

$$\Delta q = \int I(t)dt = I\Delta t \tag{18}$$

Substituting the value of w from (17) in (8), we get

$$M \approx \frac{R_{ON}}{D} w_0\left(1 - \frac{R_{OFF}}{R_{ON}}\right) + \frac{R_{ON}}{D} K\Delta q \times F_p(w)\left(1 - \frac{R_{OFF}}{R_{ON}}\right) \tag{19}$$

The current voltage relationship can be obtained as,

$$v(t) = \left\{\frac{R_{ON}}{D} w_0\left(1 - \frac{R_{OFF}}{R_{ON}}\right) + \frac{R_{ON}}{D} K\Delta q \times F_p(w)\left(1 - \frac{R_{OFF}}{R_{ON}}\right)\right\} i(t) \tag{20}$$

3 Transient and Stable State of Composite Memristance

The flux or charge in a memristor is accumulated only within a given range of flux or charge where the thickness w/D does not approach 0 and 1. If external voltage or current is applied additionally to a memristor when the state of flux or charge of a memristor is outside of this region, they overflow without being accumulated as the memristor state. Computing the composite behavior of a circuit with multiple memristors is not easy when the state of some memristors in a circuit is in the operational range while that of others is out of the range.

For the analytical explanation of the behavior as a function of the applied voltage or current, we propose a method in which the operation of a memristor at the outside of this boundary is explained with the shifting of the flux or charge function by the amount of overflowed flux or charge. Figure 2(a) describes a concept that the

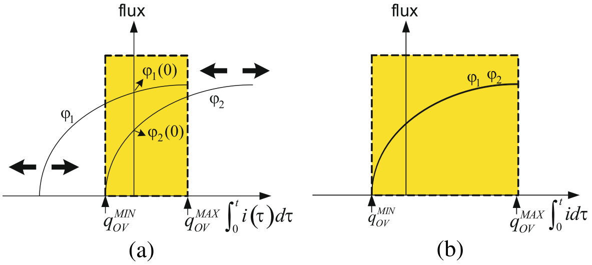


Fig. 2 Composite memristance of a multiple memristor circuit in (**a**) transient state, and (**b**) stable state of composite memristance

overflow phenomena can be formulated as the function shift when the accumulated flux curve is expressed as a function of the integration of externally applied current, where the curve is shifted to the left or right direction on the horizontal axis whenever the externally applied flux or charge exits from the max or the min threshold, respectively, as in Fig. 2(a).

Being associated with such shifting behavior, a multiple memristor circuit works in very complicated manner. Figure 2(a) shows flux curves of two memristors, whose initial fluxes $\varphi_1(0)$ and $\varphi_2(0)$ are not identical. In the figure, the flux φ_2 below q_{OV}^{MIN} is not accumulated, and similarly the flux φ_1 is not accumulated above q_{OV}^{MAX}. Therefore, the range in which the composite flux of this memristor connection can be computed is $[q_{OV}^{MIN}, q_{OV}^{MAX}]$, where the operation ranges of two memristors are overlapped. We call this range the *composite memristance range*, in which the multiple memristors act like a single memristor with a composite behavior.

When the accumulated charge $\int_0^t i(\tau)d\tau$ goes out of this range, the whole flux curve shifts along the charge axis. For example, when the externally applied charge exceeds q_{OV}^{MAX}, the whole curve φ_1 moves to the right direction of Fig. 2(a), by the amount of excess charge. On the other hand, when the externally applied charge goes below q_{OV}^{MIN}, the whole curve φ_2 moves to the left direction of Fig. 2(a), by the integration of the charge below q_{OV}^{MIN}. In any of these cases, the overlapped area of the two curves expands, and so does the composite memristance range. This is called the *transient state* of a composite memristor circuit.

With the repeated expanding of the composite memristance range, due to the repeated exiting of this boundary, two curves will eventually arrive at *a fully overlapped state*, as in Fig. 2(b). Once two curves come to have a fully overlapped memristive range, the curves move concurrently. As a result, the composite flux curve does not vary anymore. Henceforth, the memristor circuits acts as a single memristor. We call this *a stable state* of composite memristance.

Since any multiple memristor circuit will eventually reach this stable state of composite memristance, we assume that all memristor circuits in the following sections are at this stable state.

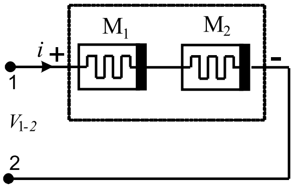


Fig. 3 A circuit with two memristors connected in series with same polarities

4 Composite Memristance of Serially Connected Memristors

4.1 Serial Memristor Circuit with the Same Polarities

When a voltage input is applied to two serially connected memristors with the polarity as shown in Fig. 3, the input voltage gets distributed across each memristor according to Kirchhoff's Voltage Law. In this type of serial connection, the memristive effect is retained, as the doped regions in both the memristors either shrink or expand simultaneously, depending upon the polarity of the applied input voltage. Applying KVL in Fig. 3,

$$V_{1-2} = iM_1 + iM_2 = V_1 + V_2 \tag{21}$$

where, V_1 and V_2 are voltages across M_1 and M_2 generated by the common current i. Integrating both sides of (21),

$$\int V_{1-2}dt = \int V_1 dt + \int V_2 dt. \tag{22}$$

From (3), (22) can be expressed as

$$\varphi_c(t) = \varphi_1(t) + \varphi_2(t) \tag{23}$$

where, $\varphi_c(t)$ is the total composite flux computed by integrating the voltage V_{1-2}, and $\varphi_1(t)$ and $\varphi_2(t)$ are the flux across M_1 and M_2 computed by integrating V_1 and V_2, respectively. From Eq. (23) we see that *the composite flux of a serially connected memristor with similar polarity is computed via a simple arithmetic sum of the individual fluxes*.

Note that the sign of the flux for each memristor is different, depending upon its polarity; when the polarity of a memristor is the same as that of the predefined reference polarity, i.e. polarity of the composite device, the sign of the individual flux is positive, and vice versa. Therefore, the signs of both flux and charge of M_1 and M_2 in Fig. 3 are identical to that of the reference.

Let us assume that the initial states of the two memristors M_1 and M_2 are $(M_1(0), \varphi_1(0), q_1(0))$ and $(M_2(0), \varphi_2(0), q_2(0))$, respectively, then the flux $\varphi_1(t)$

for the memristor M_1 can be written as a function of charge $q_1(t)$, as

$$\varphi_1(t) = R_{\mathit{off}}\left\{q_1(t)\left[1 + \frac{w_{01}}{D}\left(\frac{R_{on}}{R_{\mathit{off}}} - 1\right)\right] - \frac{\mu_v R_{on}}{2D^2}\left(1 - \frac{R_{on}}{R_{\mathit{off}}}\right)q_1(t)^2\right\} + \varphi_1(0)$$
$$-\,Q_{\max n1} \le q_1(t) \le Q_{\max p1} \tag{24}$$

where, $Q_{\max pi} = \frac{D^2}{\mu_v R_{on}}(1 - \frac{w_{0i}}{D})$ for positive polarity, and $Q_{\max ni} = \frac{D^2}{\mu_v R_{on}}\frac{w_{0i}}{D}$ for negative polarity. Similarly, the flux of M_2, $\varphi_2(t)$, can be written as a function of charge $q_2(t)$, as

$$\varphi_2(t) = R_{\mathit{off}}\left\{q_2(t)\left[1 + \frac{w_{02}}{D}\left(\frac{R_{on}}{R_{\mathit{off}}} - 1\right)\right] - \frac{\mu_v R_{on}}{2D^2}\left(1 - \frac{R_{on}}{R_{\mathit{off}}}\right)q_2(t)^2\right\} + \varphi_2(0).$$
$$-\,Q_{\max n2} \le q_2(t) \le Q_{\max p2} \tag{25}$$

Since the same amount of charge $q(t)$ is injected into each memristor in the serially connected memristor circuit, the composite flux is the simple sum of the individual fluxes. When the two memristors are assumed to be identical, and they are in the stable composite memristance state, as discussed in Sect. 3, the composite flux from (23) can be computed as,

$$\varphi_c(t) = 2R_{\mathit{off}}\left\{q(t)\left[1 + \frac{w_0}{D}\left(\frac{R_{on}}{R_{\mathit{off}}} - 1\right)\right] - \frac{\mu_v R_{on}}{2D^2}\left(1 - \frac{R_{on}}{R_{\mathit{off}}}\right)q(t)^2\right\} + 2\varphi_2(0) \tag{26}$$

where, $-\min\{Q_{\max n1}, Q_{\max p2}\} \le q(t) \le \min\{Q_{\max p1}, Q_{\max n2}\}$ and $w_{01} = w_{02} = w_0$, since $M_1 = M_2$.

Furthermore, the memristance of the composite memristor can be obtained by differentiating (26) with respect to $q(t)$, as

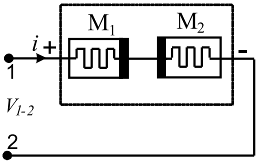
$$M_c = \frac{d\varphi_c(t)}{dq(t)} = 2R_{\mathit{off}}\left\{\left[1 + \frac{w_0}{D}\left(\frac{R_{on}}{R_{\mathit{off}}} - 1\right)\right] - \frac{\mu_v R_{on}}{D^2}\left(1 - \frac{R_{on}}{R_{\mathit{off}}}\right)q(t)\right\} \tag{27}$$
$$-\min\{Q_{\max n1}, Q_{\max p2}\} \le q(t) \le \min\{Q_{\max p1}, Q_{\max n2}\}$$

From (27), we see that the composite memristor exhibits memristance in the range of $-\min\{Q_{\max n1}, Q_{\max p2}\} \le q(t) \le \min\{Q_{\max p1}, Q_{\max n2}\}$ and *acts like a single memristor in that range.*

4.2 Serial Memristor Circuit with the Opposite Polarities

When a positive voltage (or current) signal is applied to a circuit with two memristors connected in series, but with opposite polarities, as shown in Fig. 4, then

Fig. 4 Memristor circuit with two memristors connected back to back in series



the memristance of M_1 decreases, whereas the memristance of M_2 increases. As a result, the composite memristance becomes constant, due to their complementary action. In this case also, the flux arithmetic of (23) can be applied, to obtain the flux for the composite device.

Let us assume that the polarity of M_1 is the same as that of the predefined reference polarity, and the polarity of M_2 is opposite to that of the reference. If charge $q(t)$ is injected into the positive terminal of the composite device, it acts as positive charge for M_1, whereas it acts as negative charge for M_2. Thus,

$$q_2(t) = -q(t) \tag{28}$$

Similarly, the sign of flux $\varphi_2(t)$ is opposite to the reference, i.e. $\varphi_{c2}(t) = -\varphi_2(t)$. The composite flux $\varphi_{c2}(t)$ can be written as a function of charge $q(t)$,

$$\varphi_{c2}(t) = R_{off}\left\{q(t)\left[1+\frac{w_{02}}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right]+\frac{\mu_v R_{on}}{2D^2}\left(1-\frac{R_{on}}{R_{off}}\right)q(t)^2\right\}-\varphi_2(0). \tag{29}$$

In contrast, in the case of M_1, the signs of both flux and charge are identical to that of the composite memristor. From (24), $\varphi_{c1}(t)$ is given as, $\varphi_{c1}(t) = \varphi_1(t)$. Therefore, $\varphi_{c1}(t)$ can be written as,

$$\varphi_{c1}(t) = R_{off}\left\{q(t)\left[1+\frac{w_{01}}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right]-\frac{\mu_v R_{on}}{2D^2}\left(1-\frac{R_{on}}{R_{off}}\right)q(t)^2\right\}+\varphi_1(0) \tag{30}$$

where, $q_1(t)$ in (24) is replaced by $q(t)$, and $\varphi_{c1}(t)$ is the converted flux of $\varphi_1(t)$ in terms of composite flux. The total flux $\varphi_c(t)$ is the sum of $\varphi_{c1}(t)$ and $\varphi_{c2}(t)$. When two memristors are assumed to be identical, and they are in the stable composite memristance state, as discussed in Sect. 3, the flux of the composite memristor becomes

$$\varphi_c(t) = 2R_{off}q(t)\left[1+\frac{w_0}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right] \tag{31}$$

where $-\min\{Q_{\max n1}, Q_{\max p2}\} \le q(t) \le \min\{Q_{\max p1}, Q_{\max n2}\}$ and $w_{01} = w_{02} = w_0$. Furthermore, the memristance of the composite memristor can be obtained by differentiating (31) with respect to $q(t)$, as

$$\begin{gathered} M_c = \frac{d\varphi_c(t)}{dq(t)} = 2\left\{R_{off}\left[1+\frac{w_0}{D}\left(\frac{R_{on}}{R_{off}}-1\right)\right]\right\} = 2M_0; \\ -\min\{Q_{\max n1}, Q_{\max p2}\} \le q(t) \le \min\{Q_{\max p1}, Q_{\max n2}\} \end{gathered} \tag{32}$$

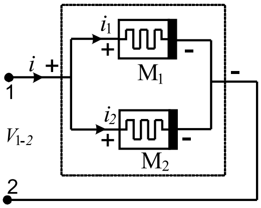


Fig. 5 Two memristors M_1 and M_2 connected in parallel with the same polarity

where, M_c is the composite memristance, and

$$M_0 = R_{off}\left[1 + \frac{w_0}{D}\left(\frac{R_{on}}{R_{off}} - 1\right)\right].$$

Note that M_c in (32) is constant, since all the related parameters for M_0 are constant. We can see that the composite memristor exhibits a resistive behavior, in the range of

$$\left[-\min\{Q_{\max n1}, Q_{\max p2}\} \leq q(t) \leq \min\{Q_{\max p1}, Q_{\max n2}\}\right].$$

5 Composite Memristance of Parallel Memristors

5.1 Parallel Memristor Circuit with Identical Polarities

When two memristors are connected in parallel with the same polarities as shown in Fig. 5, voltage across both the memristors are equal. According to Kirchhoff's Current Law, the total (composite) input current is the sum of the individual currents, i.e.

$$i = i_1 + i_2 \tag{33}$$

Integrating (33), we obtain

$$q_c(t) = q_{c1}(t) + q_{c2}(t) \tag{34}$$

where $q_c(t)$ is the total charge injected into the circuit, $q_{c1}(t)$ and $q_{c2}(t)$ are the converted expressions of $q_1(t)$ and $q_2(t)$, respectively, relative to the predefined reference.

The composite flux remains constant, as the memristors connected in parallel share a common voltage. Note that the composite flux is computed by the integration of the common voltage across the parallel memristor circuits. The expression for charge of M_1 and M_2 can be obtained by rearranging Eqs. (24) and (25),

$$q_{c1}(t) = \frac{Q_o R_{o1}}{\Delta R}\left(1 - \sqrt{1 - \frac{2\Delta R}{Q_o R_{o1}^2}(\varphi_{c1})}\right) + q_1(0) \quad \varphi_{\min} \leq \varphi_{c1} \leq \varphi_{\max} \tag{35}$$

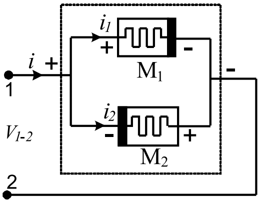


Fig. 6 Two memristors M_1 and M_2 connected in parallel with opposite polarity

$$q_{c2}(t) = \frac{Q_o R_{o2}}{\Delta R}\left(1 - \sqrt{1 - \frac{2\Delta R}{Q_o R_{o2}^2}(\varphi_{c2})}\right) + q_2(0) \quad \varphi_{\min} \le \varphi_{c2} \le \varphi_{\max} \tag{36}$$

where, $\Delta R = R_{OFF} - R_{ON}$ and $Q_o = D^2/\mu_v R_{on}$. The value of $\varphi_{\min}$ and $\varphi_{\max}$ are $[0, 0.805]$ and $[-0.805, 0]$ for positive and negative polarity, respectively.

When two memristors are assumed to be identical, and they are in the stable composite memristance state, as discussed in Sect. 3, the charge $q_c(t)$ of the composite device is computed from (34) as,

$$q_c(t) = \frac{2Q_o R_o}{\Delta R}\left(1 - \sqrt{\left(1 - \frac{2\Delta R\varphi(t)}{Q_o R_o^2}\right)}\right) + q_c(0). \tag{37}$$

Differentiating (37) with respect to φ, we obtain

$$\frac{dq_c(t)}{d\varphi(t)} = \frac{2}{R_o\sqrt{1 - \frac{2\Delta R(\varphi(t))}{Q_o R_o^2}}}. \tag{38}$$

Hence, the composite memristance is given by the

$$\begin{aligned} M_c(t) &= \frac{d\varphi(t)}{dq_c(t)} \\ &= \frac{R_o}{2}\sqrt{1 - \frac{2\Delta R(\varphi)}{Q_o R_o^2}} = \frac{M_1}{2} = \frac{M_2}{2} \end{aligned} \tag{39}$$

where, $M_1 = M_2 = R_o\sqrt{1 - \frac{2\Delta R(\varphi)}{Q_o R_o^2}}$.

We can conclude that the charge of the composite device is the sum of the individual charges of M_1 and M_2. In addition, the computation of the composite memristance can be performed in the same way, as that for ordinary parallel resistors.

5.2 Parallel Memristor Circuit with Opposite Polarities

When an input current (or voltage) signal is applied to a memristor circuit connected in parallel, but with opposite polarities, as shown in Fig. 6, the memristance of each memristor changes in opposite direction to that of the other, depending upon their respective polarities.

For the parallel memristor circuit shown in Fig. 6, the signs of charges are different from the circuit shown in Fig. 5. Assuming that the polarity of M_1 is same as that of the composite device, the polarity of M_2 is opposite. Then, from (35) and (36),

$$q_{c1}(t) = \frac{Q_o R_{o1}}{\Delta R}\left(1 - \sqrt{1 - \frac{2\Delta R}{Q_o R_{o1}^2}(\varphi_{c1})}\right) + q_1(0) \tag{40}$$

$$q_{c2}(t) = -\frac{Q_o R_{o2}}{\Delta R}\left(1 - \sqrt{1 + \frac{2\Delta R}{Q_o R_{o2}^2}(\varphi_{c2})}\right) - q_2(0) \tag{41}$$

Plugging (40) and (41) into Eq. (34), we obtain the composite charge of the circuit in Fig. 6, as

$$\begin{aligned} q_c(t) = {} & \frac{Q_o R_{o1}}{\Delta R}\left(1 - \sqrt{1 - \frac{2\Delta R}{Q_o R_{o1}^2}(\varphi)}\right) + q_1(0) \\ & - \frac{Q_o R_{o2}}{\Delta R}\left(1 - \sqrt{1 + \frac{2\Delta R}{Q_o R_{o2}^2}(\varphi)}\right) - q_2(0) \end{aligned} \tag{42}$$

Differentiating (42) with respect to φ, where $\varphi = \varphi_{c1} = \varphi_{c2}$

$$\begin{aligned} \frac{dq_c(t)}{d\varphi(t)} = \frac{1}{M_c(t)} &= \frac{1}{R_{o1}\sqrt{1 - \frac{2\Delta R(\varphi_{c1})}{Q_o R_{o1}^2}}} + \frac{1}{R_{o2}\sqrt{1 + \frac{2\Delta R(\varphi_{c2})}{Q_o R_{o2}^2}}} \\ &= \frac{1}{M1} + \frac{1}{M_2} \end{aligned} \tag{43}$$

6 Simulation Results

6.1 Linear Model

The simulations in this section are based on the $\mathrm{TiO_2}$ memristor model, where $R_{ON} = 100\ \Omega$, $R_{OFF} = 16\ \mathrm{K}\Omega$, $D = 10$ nm and $\mu_v = 10^{-14}\ \mathrm{m^2\,V^{-1}\,S^{-1}}$. When a sinusoidal input signal $v(t) = V_m \sin(\omega t)$, is applied to a single memristor with positive polarity, the memristance decreases from its initial value, and when the polarity of the memristor is reversed, memristance increases. The range of memristance depends on the value of R_{ON} and R_{OFF}. The memristance can be interpreted as the *slope* at the operating point $q = q_Q$ at time t on the flux curve of the memristor as shown in Fig. 7.

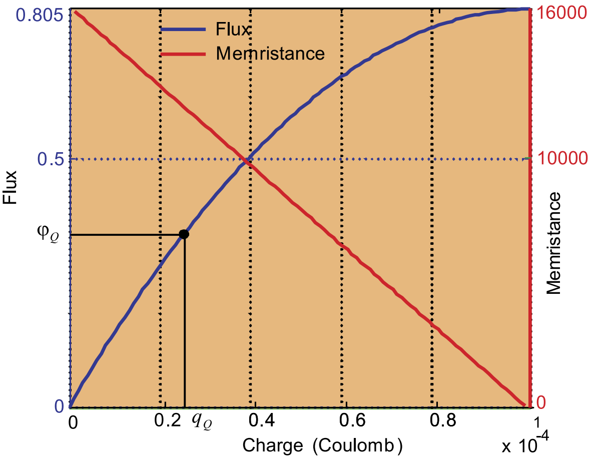


Fig. 7 Flux vs. charge curve (*Blue color*) and Memristance vs. charge curve (*Red color*) for a single decrement type memristor. The initial state of the memristor is ($M(0) = 15890\ \Omega$, $\varphi(0) = 0\ Wb$, $q(0) = 0$)

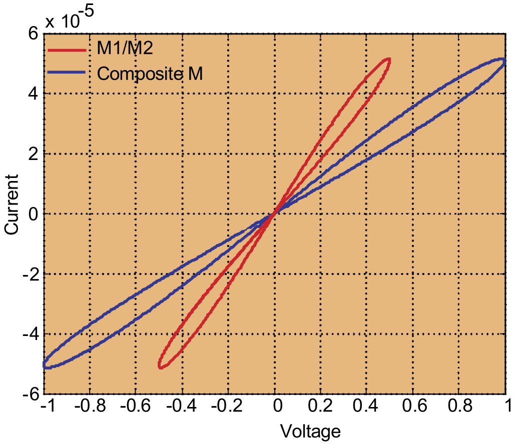
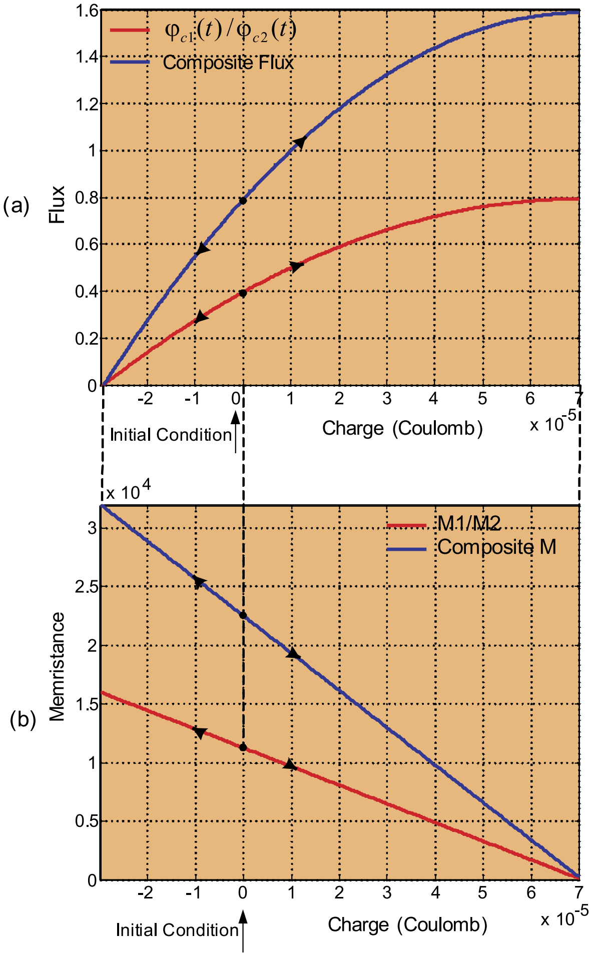


Fig. 8 Current vs. voltage graph for individual memristors, and that of the composite memristor at $\omega = 5$

Figures 8 and 9 show the simulation results for the serially connected memristors, with the polarities as shown in Fig. 3. It is assumed that the composite memristance is in stable state as discussed in the Sect. 3, which is the assumption applied to all cases of circuits afterwards. The initial states of the memristors are ($M_1(0) = 11250\ \Omega$, $\varphi_1(0) = 0.395$ Wb, $q_1(0) = 2.9 \times 10^{-5}$ C) and ($M_2(0) = 11250\ \Omega$, $\varphi_2(0) = 0.395$ Wb, $q_2(0) = 2.9 \times 10^{-5}$ C). When a sinusoidal voltage signal $v(t) = A\sin(\omega t)$ with $\omega = 5$ rad/s is applied to the input terminal, the applied voltage is distributed across each memristor equally at any instant of time.

As the memristors are connected in series, equal current flows through both memristors. In consequence, the behaviors of both memristors are identical, as shown in the red graph of Fig. 8. The composite memristance measured across the terminals 1 and 2 of Fig. 3 is the sum of the individual memristances. Therefore, the slope of the pinched loop of the composite memristor is half of that of the individual memristor, as shown in the blue pinched loop of Fig. 8. If the number of memristors in series

Fig. 9 (**a**) Flux versus charge graph, and (**b**) Memristance vs. charge for a composite device with two identical memristors in series, with the polarities shown in Fig. 3



is increased, the slope of the pinched hysteresis loop of the composite device will decrease.

The flux (φ) vs. charge (q) graph for this series connection is shown in Fig. 9(a), where the slope ($= d\varphi/dq$) at each point on the graph gives the memristance at that instant. Note that instead of the integration of externally applied current, the horizontal axis of the flux curve is drawn with the charge which is stored in a memristor, since flux can be defined in the stable memristance state as a fixed function of charge, regardless of the integration of the input charge. The slope of the φ–q curve of the composite memristor (blue graph) is two times steeper than that of the φ–q curve of individual memristors (red color).

The memristance vs. charge curve of the serially connected memristors with the same polarities is shown in Fig. 9(b), where the memristance of the composite memristor is twice as big as that of the individual memristors.

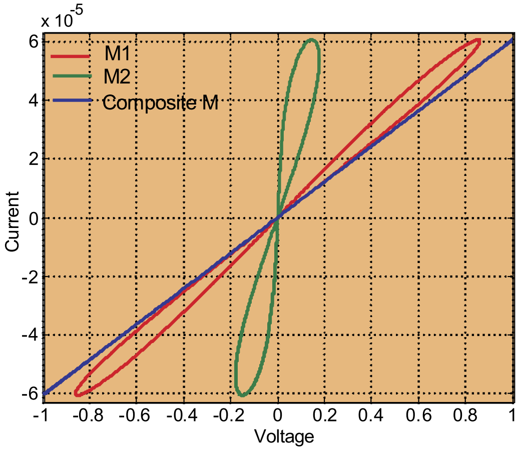


Fig. 10 Current vs. voltage graph of individual memristors and composite memristor, for the circuit in Fig. 4, at $\omega = 5$

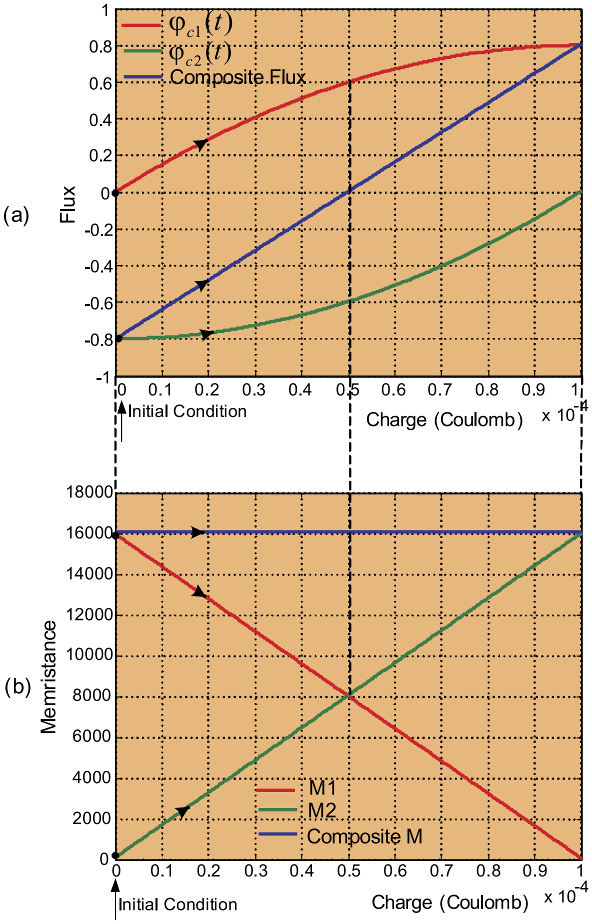
The resultant current-voltage curve for the circuit in Fig. 4 is shown in Fig. 10, where the initial state of the memristors M_1 and M_2 are ($M_1(0) = 15900\ \Omega$, $\varphi_1(0) = 0$ Wb, $q_1(0) = 0$ C) and ($M_2(0) = 116\ \Omega$, $\varphi_2(0) = -0.80$ Wb, $q_2(0) = -9.5 \times 10^{-5}$ C), respectively. In contrast to the identical polarity case in Fig. 8, the i–v curves of the two memristors M_1 and M_2 are composed of two separate loops, though their initial state is identical. This phenomenon is caused from the opposite polarities of the two back-to-back connected memristors; while memristance of M_1 increases, the memristrance of M_2 decreases, and vice versa.

Figure 11(a) shows the detailed flux $\varphi(t)$ and charge $q(t)$ relationships of the composite memristor. The upper and lower graphs of Fig. 11(a) are those of $\varphi_{c1}(t)$ in (30) and $\varphi_{c2}(t)$ in Eq. (29) respectively, where $\varphi_{c1}(0)$ and $\varphi_{c2}(0)$ are the initial fluxes of M_1 and M_2, respectively. Since the initial memristances of memristors M_1 and M_2 are 15,900 Ω and 116 Ω respectively, the corresponding initial flux $\varphi_{c1}(0)$ for M_1 is positive, whereas the initial flux $\varphi_{c2}(0)$ for M_2 is negative. Note that the negative sign of the initial flux comes from the conversion of $\varphi_{c2}(0)$, in terms of the composite memristance, as described in (29). When positive charge is applied to the circuit in Fig. 4, the magnitude of $\varphi_{c1}(t)$ increases, whereas that of $\varphi_{c2}(t)$decreases, as shown in Fig. 11(a). Once $\varphi_{c2}(t)$ reaches zero, the variation of $\varphi_{c2}(t)$stops, due to the physical limit of the memristor. Meanwhile, $\varphi_{c1}(t)$ increases further, until it reaches its physical limit. When the states of both the memristors are within the region of non zero flux, the two memristors act complementarily, and the composite flux is a linear function of charge, as shown in Fig. 11(a). The same phenomenon takes place when the negative voltage (or current) signal is applied to the circuit, as shown in Fig. 4, resulting in the linearly-varying composite flux as a function of charge, in the direction of negative charge.

Figure 11(b) shows the resultant memristance graph for the circuit of Fig. 4. Observe that the composite memristance corresponding to the linear region of Fig. 11(a) is constant.

Simulations conducted for the circuit in Fig. 5, with $M_1(0) = M_2(0) = 11,$ 250 Ω, are shown in Fig. 12. Since the value of voltage and current are the same

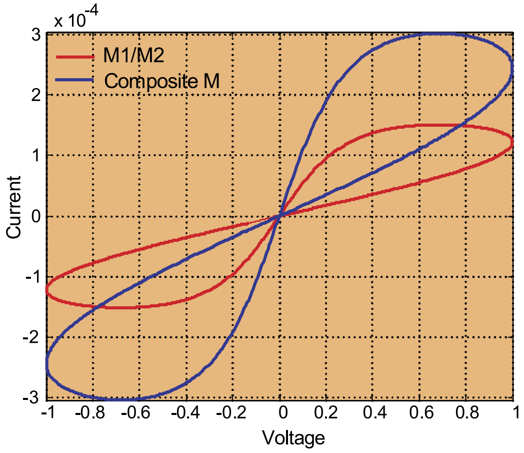
Fig. 11 (**a**) Flux versus charge graphs, and (**b**) Memristance vs. charge for a composite device with two identical memristors connected back-to-back in series



for each memristor, the pinched loops are identical for the two memristors. However, the pinched loop of the composite device is steeper than that of the individual memristors, as the composite memristance is smaller than that of the individual memristance. The flux axis is composite flux of the device and does not represent the flux of the individual memristors.

The flux vs. charge relationship of the composite device is shown in Fig. 13(a), where the initial states of the memristors are ($M_1(0) = 11250\ \Omega$, $\varphi_1(0) = 0.395$ Wb, $q_1(0) = 2.9 \times 10^{-5}$ C) and ($M_2(0) = 11250\ \Omega$, $\varphi_2(0) = 0.395$ Wb, $q_2(0) = 2.9 \times 10^{-5}$ C), respectively. The composite charge is calculated by adding the charge in individual memristors, whereas the composite flux remains the same as that of a single memristor. It follows that the variation of charge in parallel memristors with the same polarity increases as the number of parallel memristors is increased, while the flux remains constant. Figure 13(b) shows the memrisance vs. flux curve, where $M_1(0)$, $M_2(0)$ are the initial memristance of M_1 and M_2, respec-

Fig. 12 Current vs. voltage graph for memristors connected in parallel, with the same polarity at $\omega = 5$



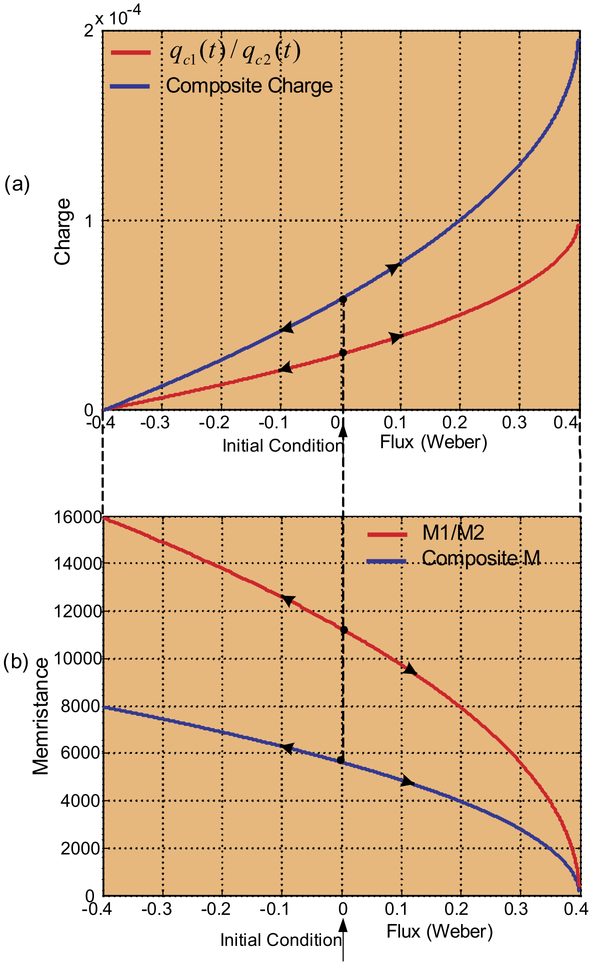
tively. Observe that the composite memristance is half of the individual memristance.

The current vs. voltage curve for the parallel memristor M_1 and M_2 with opposite polarities is shown in Fig. 14. Although the initial states of the two memristors are the same, they exhibit two different shapes of pinched hysteresis loops. While a positive signal is applied, the memristance of M_2 increases, so the current through the M_2 decreases, as shown in the graph in Fig. 14 (green graph), whereas the memristance of M_1 decreases, so the current through it increases, as shown in the red graph in Fig. 14. The pinched loop for the composite device is shown in blue in Fig. 14.

Figure 15(a) shows the flux vs. charge relationship of the composite device. The upper and lower graphs of Fig. 15(a) are those of $q_{c1}(t)$ and $q_{c2}(t)$, respectively, where $q_{c1}(0)$ and $q_{c2}(0)$ are the initial charges of M_1 and M_2 respectively. Since the initial memristance of both memristors are equal at 11,250 Ω the corresponding initial charge $q_{c1}(t)$ for M_1 is positive, while $q_{c2}(t)$ for M_2 is negative. Note that the sign of charge for a memristor depends upon its polarity; it is positive for a memristor with the same polarity as that of the predefined reference polarity, and vice versa. When a positive voltage is applied at the memristor circuit, as shown in Fig. 6, the positive charge of the memristor M_1 increases, while the negative charge of M_2 decreases, as shown in Fig. 15(a).

As a result, the composite charge, which is computed as the sum of individual charges, is small, since the signs of individual charges are opposite. The resultant charge vs. flux curve is shown in blue in Fig. 15(a). The resultant memristance graph for the circuit of Fig. 6 is shown in Fig. 15(b). Observe that the composite memristance varies gently in the middle, corresponding to the near linear region of Fig. 15(a).

Fig. 13 (**a**) Charge vs. flux curve, and (**b**) Memristance vs. flux curve for a composite device with two identical memristors, connected in parallel with the same polarity



6.2 Nonlinear Model

Simulations for the nonlinear model of the memristor were also carried out. The parameters used for these simulations were the same as those of the above linear model. Also, the value of the initial memristance is the same as that of the linear model in each case. Figure 16(a) shows the flux versus charge graphs of nonlinear model of a single memristor. We can see that the value of the flux increases as the value of p decreases for a fixed charge. Figure 16(b) shows the graphs of the memristance versus charge of the linear and nonlinear models. As the number p becomes smaller, the nonlinearity increases. At the middle regions of the curves, the increment rate of the memristance with respect to the charge, dM/dq, is constant. This fact shows that the memristance can be calculated linearly at the middle region regardless of the value of p (degree of nonlinearity).

Fig. 14 Current vs. Voltage for memristors connected in parallel, with opposite polarity at $\omega = 5$

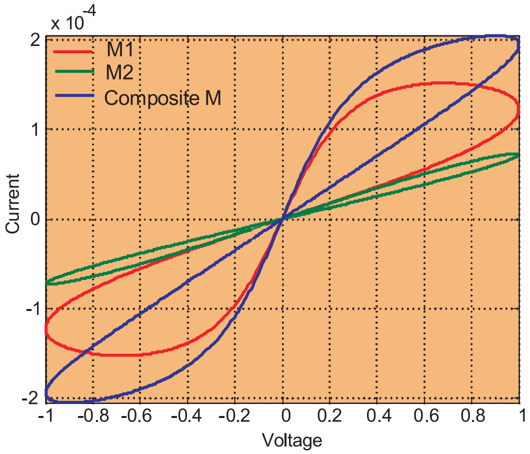


Fig. 15 (**a**) Charge vs. flux curve, and (**b**) Memristance vs. flux for a composite device with two memristors, connected in parallel with opposite polarity

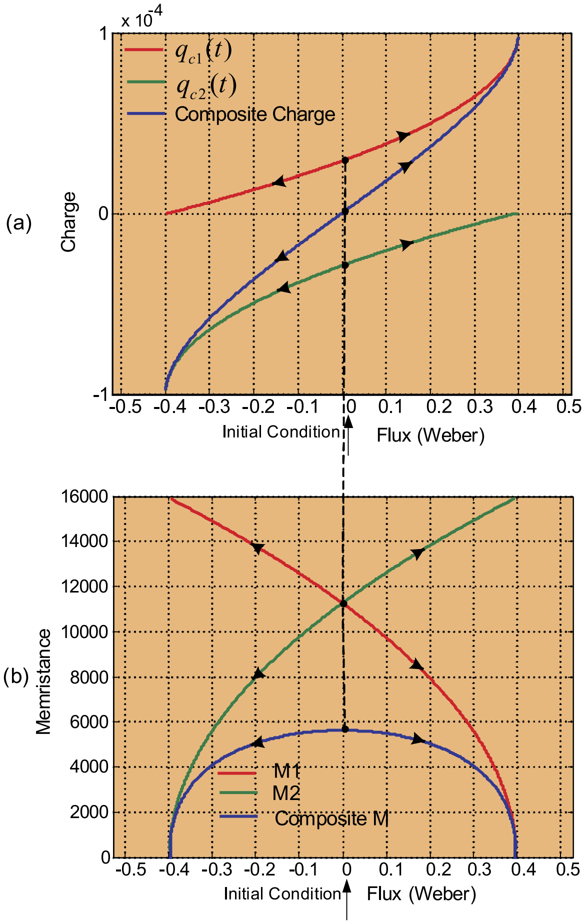


Fig. 16 (**a**) Flux vs. charge curve (**b**) Memristance vs. charge curve for the linear and nonlinear model of memristor. The initial state of the memristor is ($M(0) = 15890\ \Omega$, $\varphi(0) = 0$ Wb, $q(0) = 0$ C)

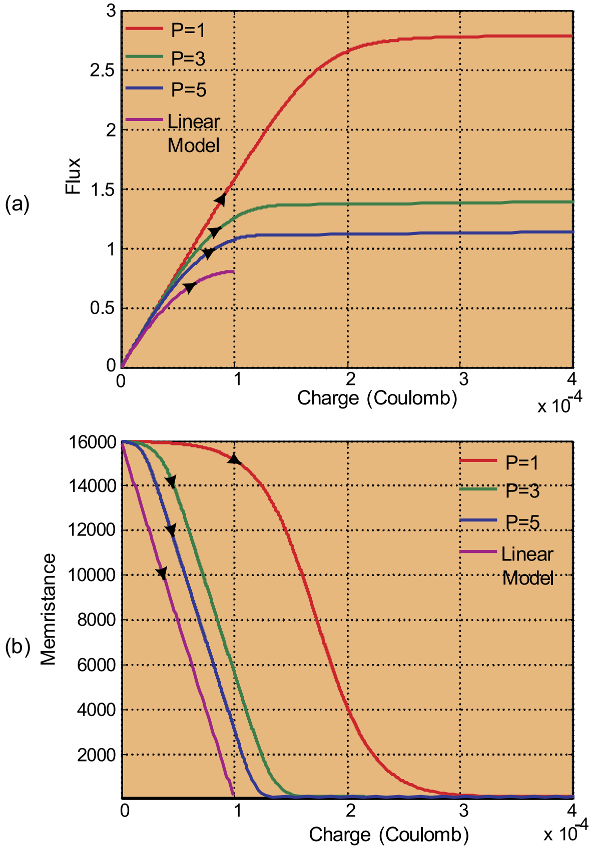


Figure 17(a) shows the flux vs. charge graph for the memristors connected in series with identical polarities using the nonlinear model of the memristor. We see that the behavior of both memristors is identical and the composite flux (blue color) is two times larger than that of the individual memristor (red color). The value of p for this simulation was 1 which is regarded as a highly non-linear model.

The memristance vs. charge curve of the two serially connected memristors with nonlinear model are shown in Fig. 17(b). In contrast to the linear model of the memristor, the memristance vs. charge curve is nonlinear. It is seen that the composite memristance (blue color) is two times of the individual memristance (red color).

The flux $\varphi(t)$and charge $q(t)$ relationships of the series back to back connected memristors with nonlinear model are shown in Fig. 18(a). The upper and lower graphs of Fig. 18(a) are those of $\varphi_{c1}(t)$ and $\varphi_{c2}(t)$ respectively, where $\varphi_{c1}(0)$ and $\varphi_{c2}(0)$ are the initial fluxes of M_1 and M_2, respectively. The varying rates of individual fluxes $\varphi_{c1}(t)$ and $\varphi_{c2}(t)$ are quite different as expected. However, their composite memristance is nearly linear. As a result, the variation of the composite memristance is less than 2.5 KΩ which is equivalent to only 16% of the variation of individual memristor as shown in Fig. 18(b).

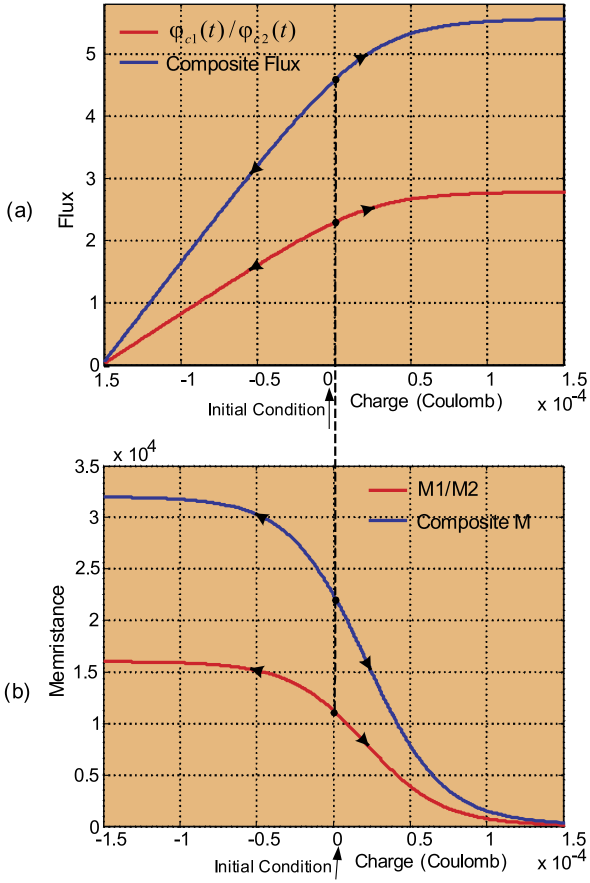
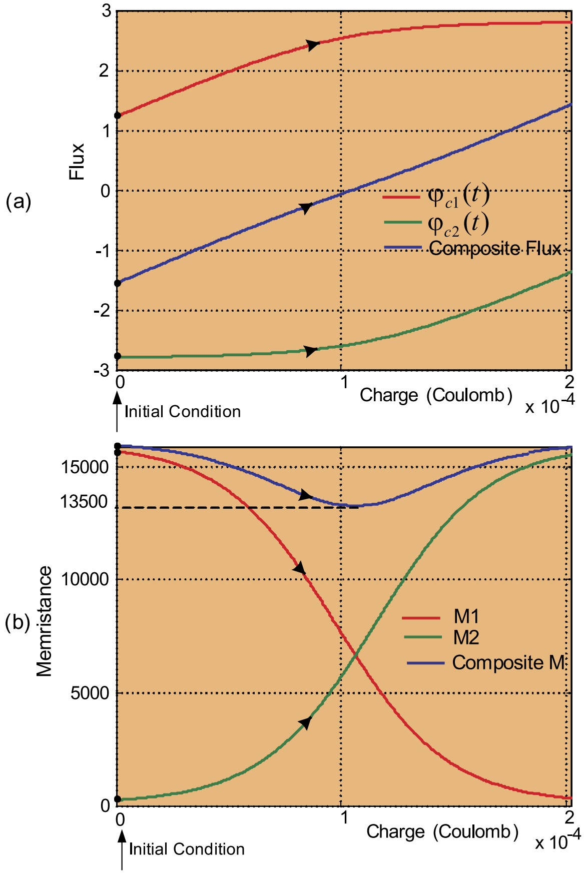


Fig. 17 (**a**) Flux versus charge graph, and (**b**) Memristance vs. charge for a series nonlinear memristor $(p = 1)$ connection with identical polarities

The simulations for parallel circuits of nonlinear memristors were also conducted. The first simulation was for parallel connection of nonlinear memristor $(p = 1)$ with identical polarities. The flux vs. charge relationship of the parallel memristor circuit with nonlinear model is shown in Fig. 19(a). The flux curves are relatively linear at low state of flux and charge values, but abruptly become highly nonlinear at high values of flux and charge. The memristance vs. flux curve is shown in Fig. 19(b), where $M_1(0)$ and $M_2(0)$ are the initial memristance of M_1 and M_2, respectively. Observe that the memristance variation is very small at its maximum value unlike that of linear models.

Figure 20(a) shows the flux vs. charge relationship of a parallel nonlinear memristor circuit with opposite polarities. The upper and lower graphs of Fig. 20(a) are those of $q_{c1}(t)$ and $q_{c2}(t)$, where $q_{c1}(0)$ and $q_{c2}(0)$ are the initial charges of M_1 and M_2, respectively. When a positive voltage is applied at the memristor circuit, as shown in Fig. 6, the positive charge of the memristor M_1 increases, while the magnitude of negative charge of M_2 decreases, as shown in Fig. 20(a). As a result, the composite charge, which is computed as the sum of individual charges,

Fig. 18 (**a**) Flux versus charge graph, and (**b**) Memristance vs. charge graph for a series nonlinear memristor ($p = 1$) connection with opposite polarities

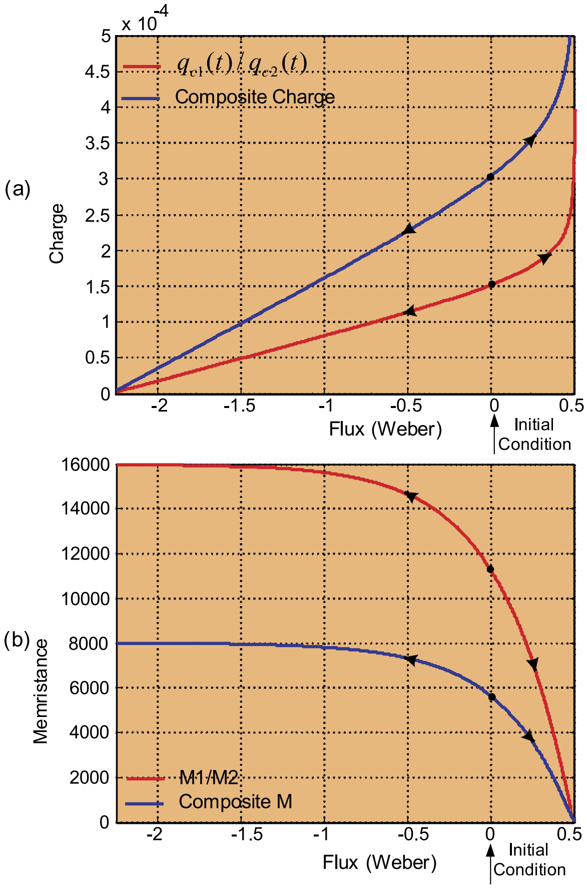


increases. The resultant memristance graph for the circuit of Fig. 6 is shown in Fig. 20(b), where the composite memristance varies gently in the middle. Observe that the memristance variation in the middle of the parallel nonlinear memristors with opposite memristors is smaller than that of the linear counterpart in Fig. 15(b).

6.3 Memristance Variance

The electrical characteristics of memristors can vary depending upon material characteristic and fabrication process. Especially, process variations may cause the deviation of the actual electrical behavior of memristors from the original design and result in the malfunction of the device. Process variations exist in three-dimension of the thin film memristor result in the cross section area and thickness variation [20]. Therefore, we have to distinguish the different impacts on the memristor brought by these process variations. However, the thickness variation is not as much as the

Fig. 19 (**a**) Charge versus flux graph, and (**b**) Memristance vs. flux graph for a parallel nonlinear memristor ($p = 1$) connection with identical polarities



cross section area variation, we investigated the memristance variation impact on the composite memristance with respect to area as in [20]. The relationship between net input flux and composite memristance of parallel memristors with opposite polarity are shown in Fig. 21. We can see that the negative variation of cross section area increases memristance while positive variation leads to the decrement of memristance, The change in memristance is almost same as that of area within the range in Fig. 21. Though the test was conducted only for the parallel memristors with opposite polarities, similar effect can be expected for all other configurations of connections.

7 Conclusion

The composite behavior of series and parallel memristor circuits are analyzed, and their computer simulations are carried out using the linear and the nonlinear model

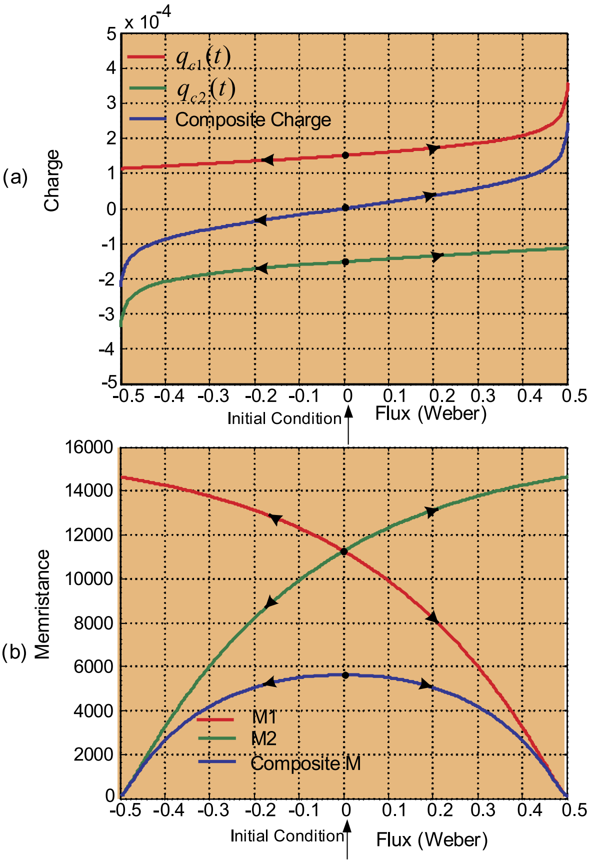


Fig. 20 (**a**) Charge versus flux graph, and (**b**) Memristance vs. flux graph for a parallel nonlinear memristor $(p = 1)$ connection with opposite polarities

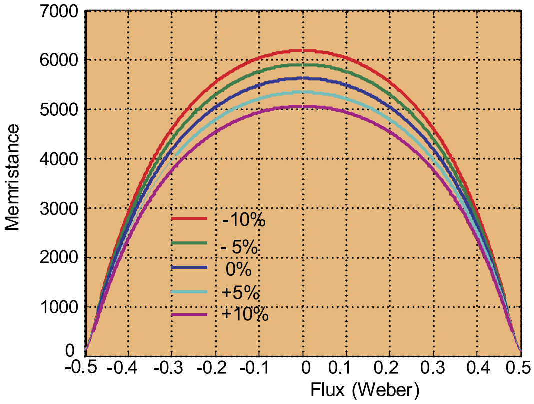


Fig. 21 Effect of cross section area variation on composite memristance of the two memristors connected in parallel with opposite polarity

of memristors. It is assumed that all memristor circuits operate at a stable composite memristance state, in which the composite flux curve does not vary and the memristor circuits act as a single memristive system, regardless of input current or voltage.

In serially-connected memristors, the composite flux is computed via a simple arithmetic sum of the individual memristor fluxes, where the sign of the individual flux is changed, depending upon the corresponding memristor polarity. The flux vs. charge curve of the serially connected memristors with the same polarities becomes more nonlinear, and the resultant composite memristance increases for the same amount of input charge.

On the contrary, the flux vs. charge curve of the serially connected memristors with opposite polarities becomes linear, due to the complementary action of the back-to-back series connected memristors, and the resultant composite memristance is constant as long as none of the memristors in the composite device is operating at the boundary.

In parallel memristors, the total charge of the multiple memristor circuit can be calculated by summing the charge in each memristor. The sign of each charge is given depending upon the polarity of the corresponding memristor. In the case of the parallel memristor circuit with the same polarities, the composite charge is the multiple of the single memristor charge, while the flux remains constant. In contrast, the charge vs. flux curve of parallel memristor circuit with the opposite polarities becomes more linear than that of a single memristor, and the resultant composite memristance changes gently.

The same simulations as the above series and parallel circuits have been conducted with nonlinear memristor models as well. The composite memristance curve with two memristors of identical polarities has simply the double and the half of the single memristor for series and parallel configurations, respectively. However, the composite memristances of memristors with opposite polarities are non-trivial. The memristance graph with series connection and opposite polarities is nearly constant even for the circuit with highly nonlinear memristor models. In case of parallel connection, memristance change in the middle of operation region is even gentler than that of linear memristor case.

The results of this study can be utilized in future studies, for analyzing the characteristics of combined memristor circuits.

References

1. Chua, L.O.: Memristor-the missing circuit element. IEEE Trans. Circuit Theory **CT-18**(5), 507–519 (1971)
2. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
3. Kim, H., Adhikari, S.P.: Memistor is not memristor. IEEE Circuits Syst. Mag. **12**(1), 75–78 (2012)
4. Wang, F.Y.: Memristor for introductory physics (2008). arXiv:0808.0286v1 [physics, Class-ph]
5. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**(7191), 80–83 (2008)
6. Kavehei, O., Eqbal, A., Kim, Y.S., Eshraghian, K., Al-sarawi, S.F., Abbott, D.: The fourth element: characteristics, modelling and electromagnic theory of the memristor. Proc. R. Soc. A, Math. Phys. Eng. Sci. **466**(2120), 2175–2202 (2010)

7. Rak, A., Cserey, G.: Macro modelling of memristors in SPICE. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **29**(4), 632–636 (2010)
8. Biolek, Z., Biolek, D., Biolková, V.: SPICE model of memristor with nonlinear dopant drift. Radioengineering **18**(2), 210–214 (2009)
9. Benderli, S., Wey, T.A.: On SPICE macromodelling of $\mathrm{TiO_2}$ memristors. Electron. Lett. **45**(7), 377–379 (2009)
10. Kim, H., Sah, M.P., Yang, C., Cho, S., Chua, L.O.: Memristor emulator for memristor circuit applications. IEEE Trans. Circuits Syst. I **59**(10), 2422–2431 (2012)
11. Pershin, Y.V., Ventra, M.D.: Practical approach to programmable analog circuits with memristors. IEEE Trans. Circuits Syst. I, Regul. Pap. **57**(8), 1857–1864 (2010)
12. Multu, R., Karakulak, E.: Emulator circuit of $\mathrm{TiO_2}$ memristor with linear dopant drift made using analog multiplier. In: 2010 National Conference on Electrical, Electronics and Computer Engineering (ELECO), pp. 380–384 (2010)
13. Joglekar, Y., Wolf, S.: The elusive memristor: properties of basic electrical circuits. Eur. J. Phys. **30**, 661–675 (2009)
14. Kim, H., Sah, M.P., Yang, C., Roska, T., Chua, L.O.: Neural synaptic weighting with a pulse-based memristor circuit. IEEE Trans. Circuits Syst. I **59**(1), 148–158 (2012)
15. Kim, H., Sah, M.P., Yang, C., Roska, T., Chua, L.O.: Memristor bridge synapses. Proc. IEEE **100**, 2061–2070 (2012)
16. Sah, M.P., Yang, C., Kim, H., Chua, L.: A voltage mode memristor bridge synaptic circuit with memristor emulators. Sensors **12**(3), 3587–3604 (2012)
17. Linn, E., Rosezin, R., Kügeler, C., Waser, R.: Complementary resistive switches for passive nanocrossbar memories. Nat. Mater. **9**(5), 403–406 (2010)
18. Kavehei, O., Al-Sarawi, S., Cho, K., Eshraghian, K., Abbott, D.: An analytical approach for memristive nanoarchitectures. IEEE Trans. Nanotechnol. **11**(2), 374–385 (2012)
19. Liu, T., Kang, Y., Verma, M., Orlowski, K.: Switching characteristics of antiparallel resistive switches. IEEE Electron Device Lett. **33**(3), 429–431 (2012)
20. Niu, D., Chen, Y., Xu, C., Xie, Y.: Impact of process variations on emerging memristor. In: 47th ACM/IEEE Design Automation Conference (DAC), pp. 877–882 (2010)

A Memristor-Based Chaotic System with Boundary Conditions

Xiaofang Hu, Guanrong Chen, Shukai Duan, and Gang Feng

Abstract This chapter proposes and studies a memristor-based chaotic system, which is constructed by incorporating a memristor into the canonical Chen oscillator with boundary conditions. Specifically, charge-controlled and flux-controlled memristor models with appropriate boundary conditions are introduced and the relation between the charge through and the flux across the memristor is derived. The rich and interesting dynamical behaviors of the memristive system are demonstrated. In particular, chaos in the system is verified by conventional means of, for instance, the Lyapunov exponent spectrum, observation of chaotic attractors, as well as basic bifurcation analysis. Finally, a basic analog circuit implementation of the memristive chaotic system based on PSPICE is presented.

1 Introduction

Memristor, theoretically postulated by Chua in 1971 [1] and physically developed by William and his team at HP labs in 2008 [2], has received increasing attention from both academic as well as industrial communities. In particular, its theoretical model has been studied extensively in the field of nonlinear science [3, 4] and its

X. Hu (✉) · G. Feng
Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Kowloon, Hong Kong SAR, China
e-mail: Huxiaofang26@163.com

G. Feng
e-mail: megfeng@cityu.edu.hk

G. Chen
Departments of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong SAR, China
e-mail: eegchen@cityu.edu.hk

S. Duan
School of Electronics and Information Engineering, Southwest University, Chongqing 400715, China
e-mail: duansk@swu.edu.cn

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_16,

physical implementation is now attracting more and more interests from material science [5, 6].

A memristor is defined by the relationship between the flux and the charge of a basic circuitry, featuring a pinched hysteresis loop in the current-voltage plane under a sine-like controlling current or voltage source. Today, memristor has been found to have promising potentials for many technological applications [7–13]. Because of the continuously variable resistance and zero-power message storage, memristor is believed to be a competitive candidate for the next-generation non-volatile semiconductor memory [7–9]. Furthermore, since the memristor-resistance is varied in response to the charge through the device, which exhibits a striking resemblance to biological synapse, memristor has been strongly recommended for artificial synapses in neural networks [10]. Today, novel implementation of a chaotic oscillator consisting of memristive elements has also become a new paradigm in nonlinear circuits [11–13].

The past three decades have witnessed a swift progress and development in theoretical research and physical realizations of various chaos generators [12–14]. However, due to the technical difficulty in physically implementing the chaos generators that are expected to exhibit more complicated chaotic behaviors, the application of chaotic oscillators has been limited in industry. Memristor, with nanoscale size and prominent nonlinear characteristics, can significantly contribute to this traditional research filed. Noticeably, Itoh, Muthuswamy and Chua proposed several Chua-circuit-based chaotic oscillators by replacing Chua's diode with a more general memristor having a similar odd-symmetric charge-flux characteristic, or a memristive system [11, 12]. Spurred by this seminal work, many research efforts have been devoted to various memristive chaotic systems, including system design, physical implementation and dynamics analysis [13, 15]. Recently, a few chaotic oscillators constituted by HP memristors have also been constructed [16].

This chapter proposes and investigates a new chaotic system based on the canonical Chen oscillator using the HP memristor. First, the theoretical models and the constructive charge-flux relationship of the memristor are derived. Then, a new chaotic system using the memristor is proposed, with its dynamical behaviors analyzed systematically. In the following, Sect. 2 presents the HP memristor, including the theoretical charge-controlled model, flux-controlled model and the charge-flux relation; Sect. 3 introduces the new memristive chaotic system, with calculations of Lyapunov exponent spectrum, Lyapunov dimension and power spectrum. Section 4 provides further analysis on its various chaotic attractors and bifurcations. A basic analog circuit implementation of the memristive chaotic system and SPICE simulations of the chaotic attractor are presented in Sect. 5. Finally, conclusions are drawn in Sect. 6.

2 The HP Memristor Model with Boundary Conditions

The HP memristor is made up of two titanium dioxide layers sandwiched between two platinum electrodes. One titanium dioxide (TiO_{2-X}) layer misses some oxygen molecules, exhibiting high conductivity. The other layer is pure TiO_2 with high resistivity. The two layers' sizes are w and $D - w$, respectively. The total resistance of the memristor is constituted by the two materials' resistance. When a current flows through the device from the conductive TiO_{2-X} region, the interface between the two materials shifts toward the insulating TiO_2 region, and as a result, the overall resistance of the memristor decreases. Oppositely, a current flowing out of the device near the TiO_{2-X} region causes an increase in the memristor's resistance. When the external excitation goes off, the interface between the two regions freezes. Thus the resistance retains its last value and this is why the device is called a memory resistor, i.e., memristor.

Memristance (short name for memristor-resistance) M is defined via the constitutive relationship of the flux versus the charge and obeys Ohm's law. So, based on the chain rule, one gets

$$v(t) = \frac{d\varphi}{dt} = \frac{d\varphi}{dq} \cdot \frac{dq}{dt} = M(t)i(t) \tag{1}$$

Memristance is given by

$$M(t) = R_{OFF} + (R_{ON} - R_{OFF})\frac{w(t)}{D}\ , \tag{2}$$

where R_{OFF} and R_{ON} are the limit memristances for $w(t) = 0$ and $w(t) = D$, respectively.

In the form of the normalized conductive layer size, $x(t)$ (i.e., $x(t) = w(t)/D$),the drift velocity satisfies

$$\frac{dx(t)}{dt} = \frac{\mu_v M_{on}}{D^2} i(t) f\big(x(t)\big)\ , \tag{3}$$

where μ_v is the average mobility per $m^2\,s^{-1}\,V^{-1}$, and $f(x)$ denotes a window function of the form

$$\begin{gathered} f\big(x(t)\big) = stp\big(x(t)\big) \cdot stp\big(1 - x(t)\big) \\ stp(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \end{gathered} \tag{4}$$

thus this window function considers the boundary conditions, i.e., $0 \leq w(t) \leq D$.

Integrating (3) yields

$$x(t) = x(0) + \frac{\mu_v M_{on}}{D^2} q(t) \tag{5}$$

Equation (2) can be rewritten as

$$M(t) = R_0 + kq(t) \tag{6}$$

where $R_0 = R_{OFF} + (R_{ON} - R_{OFF})x(0)$ is the initial memristance with constant $k = (R_{ON} - R_{OFF})\mu_v R_{ON}/D^2$.

Note that the memristance is governed by the charge (or flux) through the device, which works normally for $M \in [R_{ON}\ R_{OFF}]$; beyond this range its nonlinearity degenerates to be linear. Specifically,

$$\frac{R_{OFF} - R_0}{k} \leq q(t) \leq \frac{R_{ON} - R_0}{k} \tag{7}$$

Let $q_{low} = (R_{OFF} - R_0)/k$ and $q_{high} = (R_{ON} - R_0)/k$, a charge-controlled memristor model with appropriate boundary conditions is derived as follows

$$M(t) = \begin{cases} R_{OFF}, & q(t) \leq q_{low} \\ R_0 + kq(t), & q_{low} < q(t) < q_{high} \\ R_{ON}, & q(t) \geq q_{high} \end{cases} \tag{8}$$

Assuming that the external excitation is applied at time $t = 0$, since memristor is not an energy storage element, then $q(0) = 0$. Integrating (1), one can get the relation between the flux across and the charge through the device [13].

$$\varphi(t) = \begin{cases} R_{OFF}q(t) - \frac{(R_{OFF}-R_0)^2}{2k}, & q(t) \leq q_{low} \\ \frac{k}{2}q^2(t) + R_0 q(t), & q_{low} < q(t) < q_{high} \\ R_{ON}q(t) - \frac{(R_{ON}-R_0)^2}{2k}, & q(t) \geq q_{high} \end{cases} \tag{9}$$

Letting $c_1 = -\frac{(R_{OFF}-R_0)^2}{2k}$, $c_2 = -\frac{(R_{ON}-R_0)^2}{2k}$, and expressing the charge $q(t)$ with the flux $\varphi(t)$, one can obtain the constitutive relation of the memristor.

$$q(t) = \begin{cases} \frac{\varphi(t)-c_1}{R_{OFF}}, & \varphi(t) \leq \varphi_{low} \\ \frac{\sqrt{2k\varphi(t)+(R_0)^2}-R_0}{k}, & \varphi_{low} < \varphi(t) < \varphi_{high} \\ \frac{\varphi(t)-c_2}{R_{ON}}, & \varphi(t) \geq \varphi_{high} \end{cases} \tag{10}$$

In the above, $\varphi_{low} = \frac{R_{OFF}^2 - R_0^2}{2k}$ and $\varphi_{high} = \frac{R_{ON}^2 - R_0^2}{2k}$.

Substituting (10) into (8), we get a theoretical flux-controlled model of the memristor given by

$$M(t) = \begin{cases} R_{OFF}, & \varphi(t) \leq \varphi_{low} \\ \sqrt{2k\varphi(t) + (R_0)^2}, & \varphi_{low} < \varphi(t) < \varphi_{high} \\ R_{ON}, & \varphi(t) \geq \varphi_{high} \end{cases} \tag{11}$$

Several simulations have been carried out on software Matlab to observe the typical properties of the memristor. Figure 1(left) shows the proposed window function. Figure 1(right) denotes the pinched hysteresis current-voltage loop finger print of the memristor subject to a sinusoidal voltage bias.

Figure 2 depicts the nonlinear relation of the charge versus the flux. Observe that within a bigger range of the flux, the curve tends to be a piecewise-linear approximation of the global charge-flux characteristic (on the left), while the locally nonlinear

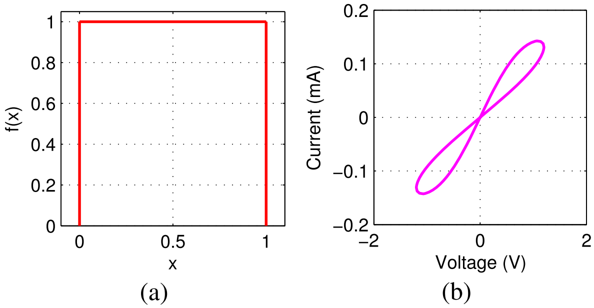


Fig. 1 Window function (*left*) and Hysteresis loop in I–V plane (*right*)

property is illustrated on the right. The memristor model parameters are chosen to be typical values: $R_{OFF} = 20\ \mathrm{k}\Omega$, $R_{ON} = 100\ \Omega$, $R_0 = 16\ \mathrm{k}\Omega$, $D = 10$ nm, and $\mu_\nu = 10^{-14}\ \mathrm{m}^2\,\mathrm{s}^{-1}\,\mathrm{V}^{-1}$, which will be used in all the following simulations.

Here, it is worth mentioning that there exist different scales between the parameters of the memristive element with a nanometer structure and the parameters determining the nonlinear circuit. Such scaling operations are often necessary [16], which is a common issue for almost all the hybrid canonical electric and nanometer memristive elements in circuit design, and will be further discussed in the next section.

3 A New Memristive Chaotic System

Applying the charge-flux characteristics of a memristor as a state feedback inserting into the canonical Chen system [14], we obtain a novel memristive chaotic system, as follows

$$\begin{cases} \dot{x} = a(y - x) + bv \\ \dot{y} = cx - xz + dy \\ \dot{z} = xy - z \\ \dot{v} = y - 10^3 \cdot 10 f(-|x|) \end{cases} \tag{12}$$

where x, y, z and v are the state variables of the system with initial condition set to [0, 0, 0, 1]; a, b, c and d are constant parameters; the term $f(-|x|)$ denotes the charge of the memristor given in (10), precisely expressed by

$$f\big(-|x|\big) = \begin{cases} \frac{x-c_1}{R_{OFF}}, & x \le \varphi_{low}, \\ \frac{\sqrt{2kx+R_0^2}-R_0}{k}, & \varphi_{low} < x < 0, \\ \frac{\sqrt{-2kx+R_0^2}-R_0}{k}, & 0 \le x < -\varphi_{low}, \\ \frac{-x-c_1}{R_{OFF}}, & x \ge -\varphi_{low} \end{cases} \tag{13}$$

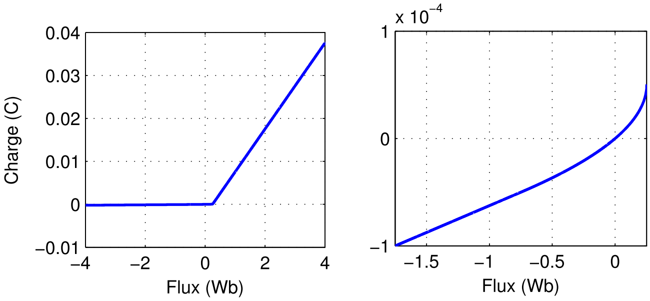


Fig. 2 Charge-flux characteristics

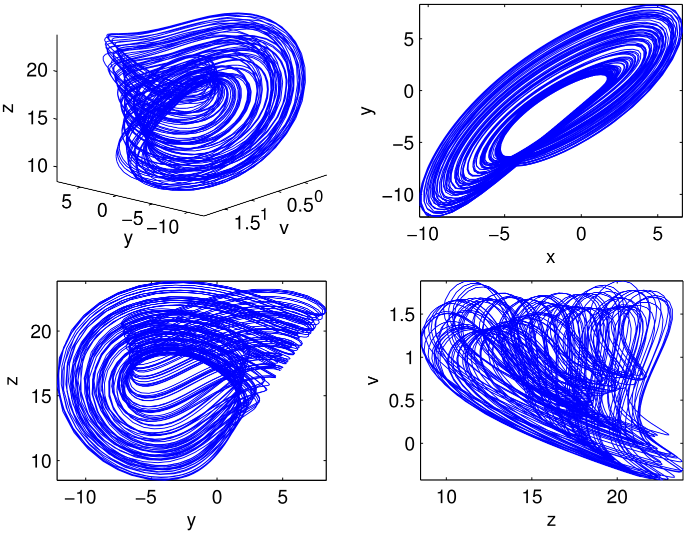


Fig. 3 Phase portrait of system (12)

Recall that for a dynamical system, if has a real solution, then this solution x_e is an equilibrium of the system [17]. For this memristive system codified by (12), there is only one equilibrium (0, 0, 0, 0) which is unstable.

One typical set of parameter values of the system with which the system can generate chaotic dynamics are: $a = 15$, $b = 7$, $c = 3$ and $d = 10.5$. Matlab ode45 solver was used to solve the system equations, with numerical simulations carried out from $t_{start} = 0$ s, $t_{start} = 1000$ s. To avoid transitory effects, the data used for illustration of the chaotic attractor refers to the time interval [900 s, 1000 s]. A three-dimension and three two-dimension projections of the chaotic attractor are shown in Fig. 3. Besides, the corresponding time-domain waveforms of the four dynamical states $x(t)$, $y(t)$, $z(t)$ and $v(t)$ are illustrated in Fig. 4. Figure 5 depicts Fast Fourier Transform (FFT), or power spectrum, of state $x(t)$, which is continuous.

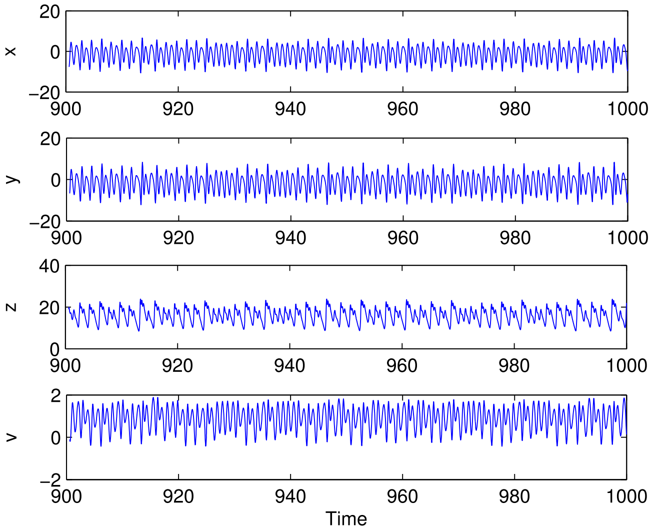
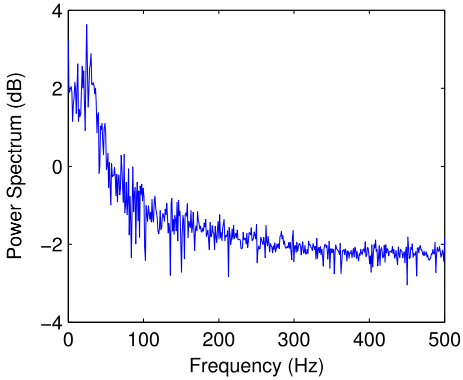


Fig. 4 Time evolution of states $x(t)$, $y(t)$, $z(t)$ and $v(t)$ of the memristive chaotic system

Fig. 5 Power spectrum of state $x(t)$



The standard QR method is used to compute the Lyapunov exponents of this system, again on Matlab, with $t = 20000$ s ($t_{final} = 2000$ and step $= 0.1$). The exponent spectrum is presented in Fig. 6, and part of the spectrum is enlarged in the inset. The finial values of the Lyapunov exponents are: $l_1 = 0.386$, $l_2 = 0.002$, $l_3 = -1.389$ and $l_4 = -4.450$.

Recall that the Lyapunov dimension is defined by

$$d_L = j + \frac{\sum_{i=1}^{i=j} l_i}{|l_{j+1}|} \qquad (14)$$

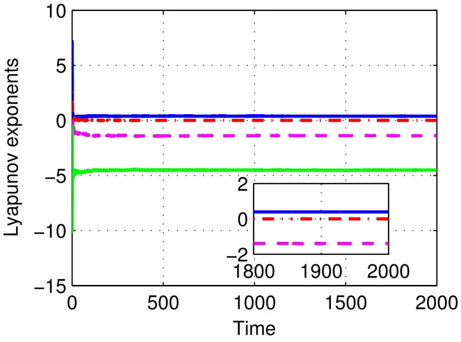


Fig. 6 Orbits of Lyapunov exponents of system (12)

where j is the maximum integer satisfying

$$\sum_{i=1}^{i=j} l_i \geq 0, \qquad \sum_{i=1}^{i=j+1} l_i \leq 0 \tag{15}$$

The Lyapunov dimension of the system (12) is calculated: $d_L = 2 + (l_1 + l_2)/|l_3| = 2.279$, which is fraction.

4 Chaotic Attractor and Its Bifurcation Analysis

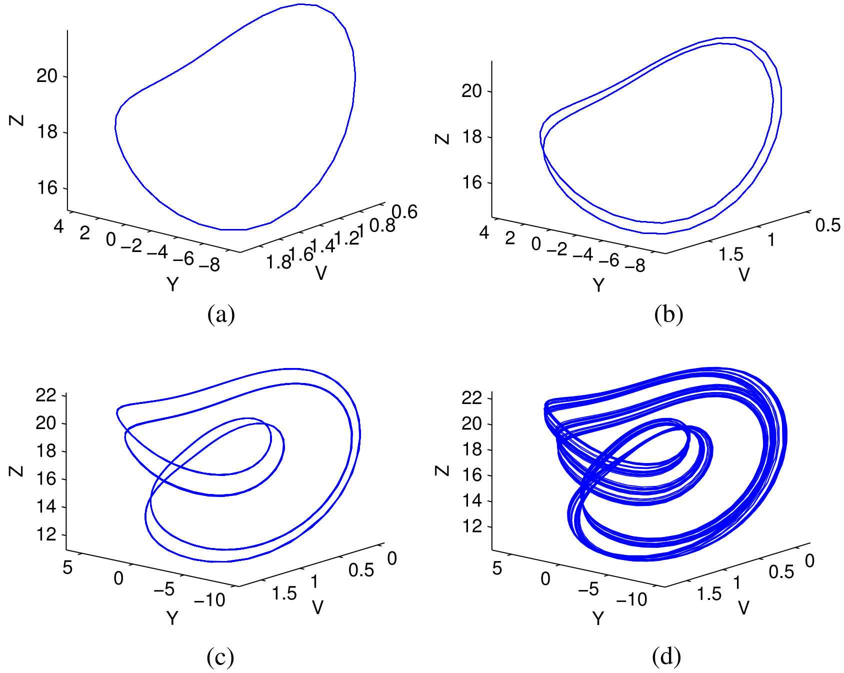
A large number of computer simulations have been carried out to verify and demonstrate that the proposed memristive chaotic system exhibits complex dynamical behaviors. Due to space limitation, only a few representative results are presented and discussed here.

Keep $a = 15$, $b = 7$ and $c = 3$ unchanged. For different values of d, Fig. 7(a) shows a period-1 orbit when $d = 11.50$, (b) a period-2 orbit when $d = 11.34$, (c) a multiple periodic orbit when $d = 10.86$, and (d) a chaotic attractor when $d = 10.74$, successively.

Next, further investigation on bifurcation versus parameter d of the new system is shown on top of Fig. 8, with the corresponding Lyapunov exponent spectrum shown on the bottom. Note that since the fourth Lyapunov exponent is always about $l_4 = -4.5$, which is kind of big as compared with the others, it is omitted from the figure for better view of the other exponents.

So far, several numerical calculations and simulations have demonstrated the basic and interesting dynamical behaviors of the memristive system. Nevertheless, more possible scenarios are worth being further investigated in the future.

In the next section, an analog circuit implementation of the memristive system is presented for the flux-controlled SPICE memristor model on software OrCAD.



(a)

(b)

(c)

(d)

Fig. 7 Projections of various attractors of system (12) for different values of d, in which (**a**), (**b**), (**c**) and (**d**) are period-1, period-2, multiple periodic orbit, and chaotic attractor when $d = 11.50$, $d = 11.34$, $d = 10.86$ and $d = 10.74$, successively

5 Analog Implementation and SPICE Simulations of the Chaotic Attractor

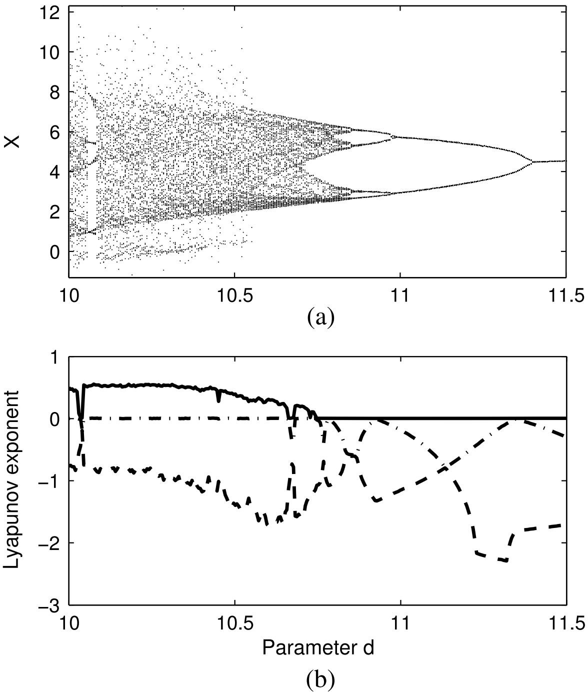
The new system may be implemented by means of an electronic circuit. An analog implementation, which realizes the memristive chaotic system, is shown in Fig. 9. It consists of four channels which perform the integrations of the four variables. Operational amplifiers, multipliers, diodes, linear capacitors and resistors are employed to realize the addition, reverse proportional amplification, multiplication, as well as integration and absolute operations. The operational amplifiers are all of LM675 type, which are powered with $V_{CC} = +30$ V and $V_{EE} = -30$ V. The multipliers are both MULT type. The voltages at the nodes labeled X, Y, Z and V correspond to the four states of the system formulated in (12).

The operational amplifiers U_1 and U_2 are employed to implement the following formulas

$$v_{U_1} = -\left(-\frac{R_4}{R_1}v_X + \frac{R_4}{R_2}v_V + \frac{R_4}{R_3}v_Y\right) \tag{16}$$

$$v_X = -\frac{1}{R_5C_1}\int v_{U_1}dt = \int\left(-\frac{R_4}{R_5C_1R_1}v_X + \frac{R_4}{R_5C_1R_2}v_V + \frac{R_4}{R_5C_1R_3}v_Y\right)dt \tag{17}$$

Fig. 8 Bifurcation diagram and Lyapunov exponent spectrum versus parameter d



Or equivalently,

$$\dot{v}_X = -\frac{R_4}{R_5 C_1 R_1} v_X + \frac{R_4}{R_5 C_1 R_2} v_V + \frac{R_4}{R_5 C_1 R_3} v_Y \tag{18}$$

Comparing with (12), by setting $R_1 = R_3 = 700\ \Omega$, $R_2 = 1.5\ \text{k}\Omega$, $R_4 = 10.5\ \text{k}\Omega$, $R_5 = 1\ \text{M}\Omega$ and $C_1 = 1$ uF, it leads to $\dot{x} = 15(y - x) + 7v$.

The operational amplifiers U_3 associated with $R_6 = R_7 = 1\ \text{k}\Omega$ realizes the reverse proportion operation, yielding $v_{U_3} = -v_X$. The multiplier MULT1 implements the term $v_X v_Z$. Similarly, the following formulas are realized by using the operational amplifiers U_4 and U_5

$$v_{U_4} = -\left(-\frac{R_{11}}{R_8} v_X + \frac{R_{11}}{R_9} v_X v_Z + \frac{R_{11}}{R_{10}} v_Y\right) \tag{19}$$

$$v_Y = -\frac{1}{R_{12} C_2} \int v_{U_4} dt = \frac{1}{R_{12} C_2} \int \left(-\frac{R_{11}}{R_8} v_X - \frac{R_{11}}{R_9} v_X v_Z + \frac{R_{11}}{R_{10}} v_Y\right) dt \tag{20}$$

Or equivalently,

$$\dot{v}_Y = \frac{1}{R_{12} C_2} \left(\frac{R_{11}}{R_8} v_X - \frac{R_{11}}{R_9} v_X v_Z + \frac{R_{11}}{R_{10}} v_Y\right) \tag{21}$$

Set $R_8 = 7\ \text{k}\Omega$, $R_9 = R_{11} = 21\ \text{k}\Omega$, $R_{10} = 2\ \text{k}\Omega$, $R_{12} = 1\ \text{M}\Omega$ and $C_2 = 1$ uF. Then, (21) is in line with $\dot{y} = 3x - xz + 10.5y$. The multiplier MULT12 is used to

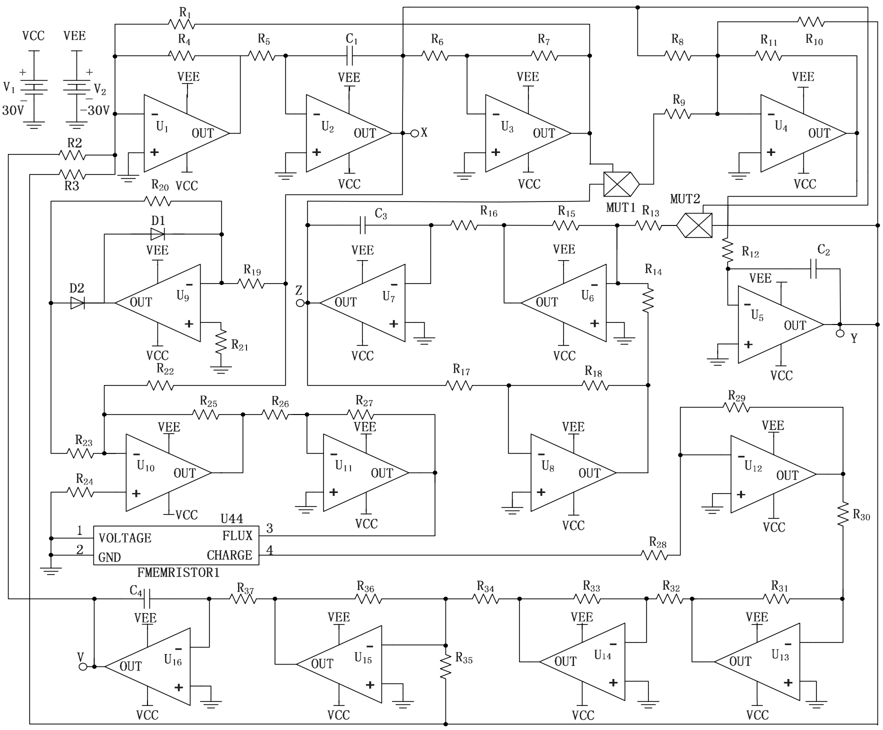


Fig. 9 Analog SPICE implementation of the memristive chaotic system

obtain $v_X v_Y$, and $U6$ implements

$$v_{U_6} = -\left(-\frac{R_{15}}{R_{14}} v_Z + \frac{R_{15}}{R_{13}} v_X v_Y\right) \tag{22}$$

where the term $-v_Z$ is the output of the operational amplifiers U_8, with $R_{17} = R_{18} = 1\ \text{k}\Omega$. The operational amplifier U_7 is an integrator which realizes

$$v_Z = -\frac{1}{R_{16}C_3}\int v_{U_6} dt = \frac{1}{R_{16}C_3}\int \left(-\frac{R_{15}}{R_{14}} v_Z + \frac{R_{15}}{R_{13}} v_X v_Y\right) dt \tag{23}$$

It follows that

$$\dot{v}_Z = \frac{1}{R_{16}C_3}\left(-\frac{R_{15}}{R_{14}} v_Z + \frac{R_{15}}{R_{13}} v_X v_Y\right) \tag{24}$$

where the related resistors and capacitor are set as:$R_{13} = R_{14} = R_{15} = 1\ \text{k}\Omega$, $R_{16} = 1\ \text{M}\Omega$ and $C_3 = 1$ uF.

The operational amplifiers U_9 and U_{10} are utilized to generate the absolute value of state $x(t)$, where two D120NQ045 diode models are used. The output voltage of

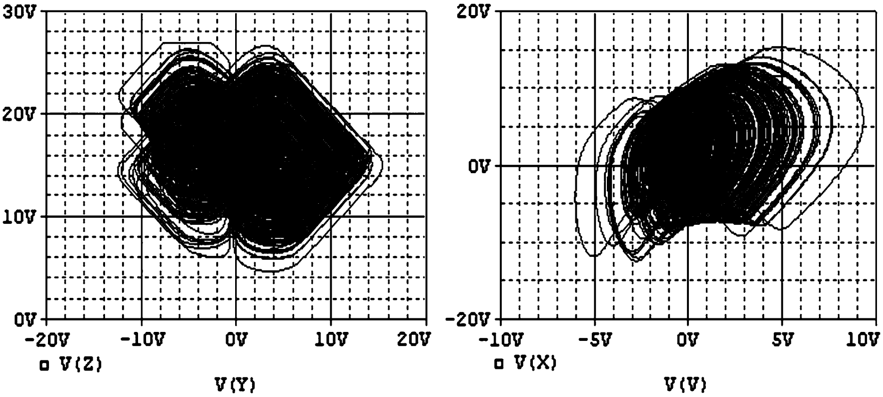


Fig. 10 Projections of the chaotic attractor generated from the analog SPICE chaotic circuit

U_{10} is given by

$$v_{U_{10}} = \begin{cases} (\frac{R_{20}R_{25}}{R_{19}R_{23}} - \frac{R_{20}}{R_{22}})v_X, & v_X \geq 0 \\ -\frac{R_{25}}{R_{22}}v_X, & v_X < 0 \end{cases} \tag{25}$$

The voltage $v_{U_{10}}|v_X|$ subject to $R_{19} = R_{20} = R_{22} = R_{25} = 1$ kΩ, $R_{23} = 500$ Ω, and the matched resistors are set to be $R_{21} = R_{19}//R_{20} = 500$ Ω and $R_{24} = R_{23}//R_{22}//R_{25} = 250$ Ω. Through U_{11}, it is possible to get $v_{U_{11}} = -\frac{R_{27}}{R_{26}}v_{U_{10}} = -|v_X|$ if $R_{26} = R_{27} = 1$ kΩ.

The voltage $v_{U_{11}}$ is input to the flux terminal of the structured SPICE model of the flux-controlled memristor. The charge of the memristor is then input to the U_{12}. The memristor can implement the function $f(\cdot)$ in (13) with the aforementioned parameter values.

When $R_{28} = R_{30} = 1$ kΩ and $R_{29} = R_{31} = 100$ kΩ, two cascading operational amplifiers, U_{12} and U_{13}, implement the gain 10,000 where each amplifies 100 times of the charge of the memristor. Then, the output of U_{13} satisfies $v_{U_{13}} = \frac{R_{31}R_{29}}{R_{30}R_{28}}f(-|v_X|) = 10000f(-|v_X|)$. U_{14} is a reverse proportional amplifier providing, $v_{U_{14}} = -10000f(-|v_X|)$ with $R_{32} = R_{33} = 1$ kΩ. Finally, the operational amplifiers U_{15} and U_{16} realize the following formulas

$$v_{U_{15}} = -\left(\frac{R_{36}}{R_{34}}v_{U_{14}} + \frac{R_{36}}{R_{35}}v_Y\right) = -\left(-\frac{R_{36}}{R_{34}}10000f(-|v_X|) + \frac{R_{36}}{R_{35}}v_Y\right) \tag{26}$$

$$v_V = -\frac{1}{R_{37}C_4}\int v_{U_{15}}dt = \frac{1}{R_{37}C_4}\int\left(-\frac{R_{36}}{R_{34}}10000f(-|v_X|) + \frac{R_{36}}{R_{35}}v_Y\right)dt \tag{27}$$

Setting $R_{34} = R_{35} = R_{36} = 1$ kΩ, $R_{37} = 1$ MΩ and $C_4 = 1$ uF, one gets $\dot{v}_V = -10000f(-|v_X|) + v_Y$.

The SPICE simulation time interval was ranged from 0 sec to 500 sec, and the maximum step size was set to 0.01 sec. Figure 10 shows the phase portraits of the analog realization of the memristive chaotic system. Noticeably, there exists a difference between the numerical simulation and the circuit simulation, which probably comes from the direction implementation of the two product terms xy and xz. Nevertheless, the simulation results have already demonstrated the chaotic behavior of the proposed memristive system, thus the constructed analog circuit can provide a reference for the circuit implementation of this kind of memristor-based chaotic system.

6 Conclusions

This chapter has reviewed the charge-controlled and flux-controlled HP memristor models, which take boundary conditions of operation into consideration. Then, based on the canonical Chen system and taking advantage of the nonlinear constructive relation between the charge and the flux of the HP memristor, a novel memristive chaotic oscillator has been constructed. A large number of computer simulations have been performed, including the Lyapunov exponent spectrum, Lyapunov dimension, power spectrum, and bifurcation diagram, demonstrating the complex chaotic dynamics of the new memristive system. In addition, an analog circuit realization of the memristive system and the corresponding SPICE simulations have been presented.

The finding of this study suggests that chaos can be generated by using the typical HP memristor rather than a general memristive system. It is believed [18] that memristors have promising potentials in constructing new chaotic circuits and systems towards more practical applications.

References

1. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**, 507–519 (1971)
2. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S., et al.: The missing memristor found. Nature **453**, 80–83 (2008)
3. Kavehei, O., Iqbal, A., Kim, Y.S., Eshraghian, K., Al-Sarawi, S.F., Abbott, D.: The fourth element: characteristics, modelling, and electromagnetic theory of the memristor. Proc. R. Soc. A **466**, 2175–2202 (2010)
4. Biolek, Z., Biolek, D., Biolková, V.: SPICE model of memristor with nonlinear dopant drift. Radioengineering **18**, 210–214 (2009)
5. Pershin, Y.V., Ventra, M.D.: Spin memristive systems: spin memory effects in semiconductor spintronics. Phys. Rev. B **78**, 3309–3312 (2008)
6. Jo, S.H., Kim, K.-H., Lu, W.: High-density crossbar arrays based on a Si memristive system. Nano Lett. **9**, 870–874 (2009)
7. Duan, S.K., Hu, X.F., Wang, L.D., Li, C.D., Mazumder, P.: Memristor-based RRAM with applications. Sci. China, Inf. Sci. **55**, 1446–1460 (2012)
8. Duan, S.K., Hu, X.F., Wang, L.D., Li, C.D.: Analog memristive memory with applications in audio signal storage. Sci. China, Inf. Sci. (2012). doi:10.1007/s11432-013-4864-z
9. Hu, X.F., Duan, S.K., Wang, L.D., Liao, X.F.: Memristive crossbar array with applications in image processing. Sci. China, Inf. Sci. **55**, 461–472 (2012)

10. Kim, H., Sah, M., Yang, C., Roska, T., Chua, L.O.: Neural synaptic weighting with a pulse-based memristor circuit. IEEE Trans. Circuits Syst. I **59**, 148–158 (2012)
11. Itoh, M., Chua, L.O.: Memristor oscillators. Int. J. Bifurc. Chaos **18**, 3183–3206 (2008)
12. Muthuswamy, B., Chua, L.O.: Simplest chaotic circuit. Int. J. Bifurc. Chaos **20**, 1567–1580 (2010)
13. Wang, L.D., Drakakis, E., Duan, S.K., He, P.F.: Memristor model and its application for chaos generation. Int. J. Bifurc. Chaos **22**, 1250205 (2012). 14 pp.
14. Ueta, T., Chen, G.: Bifurcation analysis of Chen's equation. Int. J. Bifurc. Chaos **10**, 1917–1931 (2000)
15. Zhong, Q.S., Yu, Y.B., Yu, J.B.: Fuzzy modeling and impulsive control of a memristor-based chaotic system. Chin. Phys. Lett. **27**, 020501 (2010)
16. Corinto, F., Ascoli, A., Gilli, M.: Memristor models for chaotic neural circuits. In: WCCI 2012 IEEE World Congress on Computational Intelligence, Brisbane, Australia, 10–15 June 2012
17. Sprott, J.C., Wang, X., Chen, G.: Coexistence of point, periodic and strange attractors. Int. J. Bifurc. Chaos **23**, 1350093 (2013). 5 pp.
18. Adamatzky, A., Chen, G. (eds.): Chaos, CNN, Memristors and Beyond. World Scientific, Singapore (2013)

Spiking in Memristor Networks

Ella Gale, Ben de Lacy Costello, and Andrew Adamatzky

Abstract Memristors have been suggested for the use as artificial synapses and have performed well in this role in simulations with artificial spiking neurons. We will show that real world memristors natively spike and describe the properties of these spikes. A network of purely memristors should not show any behaviour in addition to that expected from a single memristor. Networks of 2 and 3 memristor combinations were investigated. We demonstrate that, if the memristors are wired together with opposing polarity, oscillations and bursting spikes emerge. We compare two types of memristors, 'filamentary' and standard memristors (which are closer to Chua's theoretical memristors), and found that standard memristors do not exhibit these rich behaviours if they are wired with the same polarity. We propose that these oscillations and spikes may be similar phenomenon to brainwaves and neural spike trains and suggest that these behaviours can be used to perform brain-like computation.

1 Introduction

Memristors are non-linear resistors that possess a memory. They were first predicted to exist in 1971 [1] and were formally discovered in a device in 2008 [2] although other experimentally fabricated memristor devices had been made before, see for example [3–5]. After that first paper to relate the memristor theory with a real world device [2], excitement abounded over their use for computer memory (alongside ReRAM) and neuromorphic computing, as well as suggestions that they might be more energy efficient and resilient.

E. Gale (✉) · B. de Lacy Costello · A. Adamatzky
University of the West of England, Frenchay Campus, Coldharbour Lane, Bristol BS16 1QY, UK
e-mail: ella.gale@gmail.com

B. de Lacy Costello
e-mail: ben.delacycostello@uwe.ac.uk

A. Adamatzky
e-mail: andrew.adamatzky@uwe.ac.uk

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_17,

Memristor-based computer memory has not yet become popular, despite expectations. The reason for this may be because memristor memory is slower to switch that current generation computer memory. In our opinion, a more exciting use for the technology is in brain-like or neuromorphic computers. Neurons are believed to be both the seat of memory and the processor of the brain. Memristors, by combining memory with processing, offer a tantalising glimpse of devices which could do the same.

Neuromorphic computing is the concept of using computer components to mimic biological neural architectures, primarily the mammalian brain. Research has been done into using memristors as synapses [6], combining memristors and spiking neurons in simulations (see for example [7]) and even whether synapses are memristive [8] (not as unlikely as it seems as flowing blood [9], sweat ducts [10] and slime mould [11] have been shown to be memristive). Further investigations into neurons have revealed that the Hodgkin-Huxley circuit model of how neurons pass signals along their axons can be updated with memristors to offer a model more grounded in electronic engineering theory [12, 13].

In this chapter we will outline our research into spiking memristor networks. We have found that memristors spike in response to a change in voltage. These spikes bear a striking resemblance to neuronal spikes and thus could offer a route to neuromorphic computers. Furthermore, memristor spikes are faster than memristor switching and could offer a route to a memristor processor that could be more commercially viable than memristor memory. We shall first describe the qualities of the spikes and then investigate what spiking properties simple memristor networks possess.

2 Single Memristor Spiking Properties

We shall start by undertaking a thorough investigation of the current spikes seen in single memristors, before demonstrating how to deal with them theoretically.

2.1 Properties of Memristor Spikes

Memristors come in two flavours, charge-controlled (left) and flux-controlled (right) as shown below in Eq. (1) where q is the charge, φ, is the magnetic flux, M is the memristance and W is the memductance (inverse memristance) [1]

$$V(t) = M\big(q(t)\big)I(t), \qquad I(t) = W\big(\varphi(t)\big)V(t). \tag{1}$$

For a charge-controlled memristor we would input a current, I, and measure the voltage, V. Biological neurons may be described as charge-controlled because it is the movement of ions that causes the change in voltage giving rise to a voltage spike. Our memristors are flux-controlled and a change in voltage causes a spike

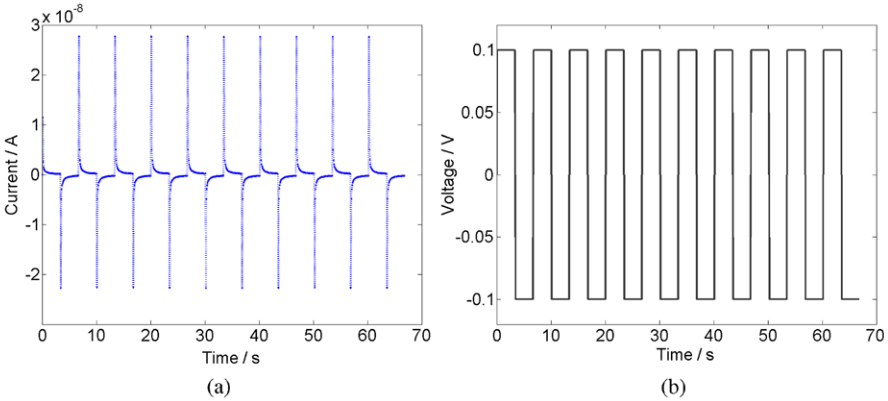


Fig. 1 (**a**) current spikes recorded from a memristor subjected to the voltage square wave in (**b**). The spike heights are highly repeaTab. and qualitatively resemble neuronal spikes

in the current. Thus, creation of a neuromorphic computer with memristors will be using the complimentary effect to the one utilized by nature, in that memristors have voltage-change-caused current spikes and neurons have current-change-caused voltage spikes. That both types of spikes have a similar form arises from the similarity in the underlying electromagnetics, in that circuits can considered as being constructed with either a voltage source or a current source.

Our memristors are flexible sol-gel titanium dioxide gel layers sandwiched between aluminium electrodes [14, 15] and they show a distinctive large spike that occurs when the voltage is changed. The experiments reported here were carried out with a Keithley 2400 sourcemeter sourcing voltage. There are no spikes in the voltage profiles, (see Fig. 1b) and no current spikes are seen when the same experiment was performed on a resistor. It has been suggested that these spikes are capacitance; however the time-scale is too long. The spikes have been reported by other groups in their memristors (see for example, [16]), however they are usually overlooked or attributed to artifacts arising from the experimental set-up or not reported at all (many researchers only report the I–V curves to demonstrate that they have a memristor). However, the current spike is an equilibrating process that is responsible for the frequency dependence of the I–V curves. In Fig. 1 each voltage step had 40 time-steps ($\approx$3.3 s) to allow the device to equilibrate. If the voltage is scanned quicker than this, the device has not equilibrated and the measured current is higher than the equilibration current. Thus, a faster switching time increases the size of the hysteresis of the I–V loop. This effect increases with frequency until it reaches the limit where the voltage frequency is too fast for the memristor to relax at all and the I–V curve just traces out the maximal spike currents for each voltage.

These current spikes can be seen whenever a voltage change occurs across the memristor. Unlike some neuronal spikes, the source voltage does not need to spike, a step will do. The current spikes are highly reproducible. For the experiment shown in Fig. 1 (10 pairs of positive to negative switches), the standard deviation was 0.0729 % of the mean for the negative voltages (where $n = 10$) and 0.1192 % of

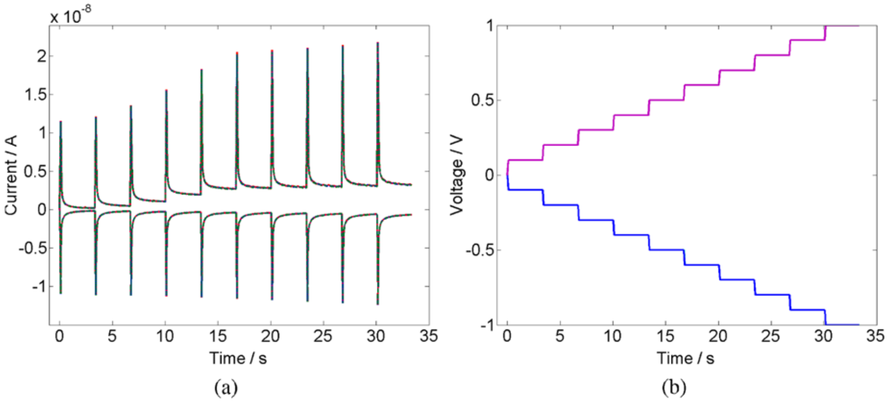


Fig. 2 (**a**) the current spike responses for 5 successive runs up and then down the voltage staircase shown in (**b**). The runs are coloured and overlap. The spikes are highly reproducible on successive runs

the positive voltages (where $n = 9$, due to incomplete recording of the first spike). For the repeated spikes in Fig. 2 (3 repeats each of both positive and negative ramps, as shown in Fig. 2) the largest difference between the spike current repeats was only 3.06×10^{-9} A and only 2.33×10^{-10} A for the equilibrated current—both taken from the positive side as it has a larger hysteresis than the negative side.

The direction of the current spikes is related to the change in voltage, not its sign, so a change from a positive voltage to zero (turning the voltage source off) gives a negative spike and vice versa for a negative voltage to zero. The spike current still flows for a short while after the voltage source has been turned off. This lag is a general thing and has been recorded in several different devices. In different devices the spikes are the same shape and seem to be following similar dynamics. The spike current is proportional to the equilibrated current. Intriguingly, spike shape closely resembles that of Bi and Poo's experimentally observed STDP function [17] and thus could be used to perform a similar function.

2.2 A Mathematical Description of Experimentally-Measured Spikes

Figure 4 shows the I–t response of a single spike to a voltage step like that shown in Fig. 3. The current spikes are roughly the same shape, and thus we can make some statements about the nature of the current spikes in memristors, which should also relate to the voltage spike in neurons. As shown in Fig. 4, there is a steady-state current, i_∞, a spike current i_0 and a transition between the two which is a time-dependent transient $i(t)$. We don't currently know if the $i(t)$ is dependent on i_0 or not. We do know that i_0 is related to i_∞. Until a thorough experimental study

Fig. 3 An example of a voltage step as applied to a memristor in an experiment (this is the same form but a different time response as the one applied to the memristor measured in Fig. 4)

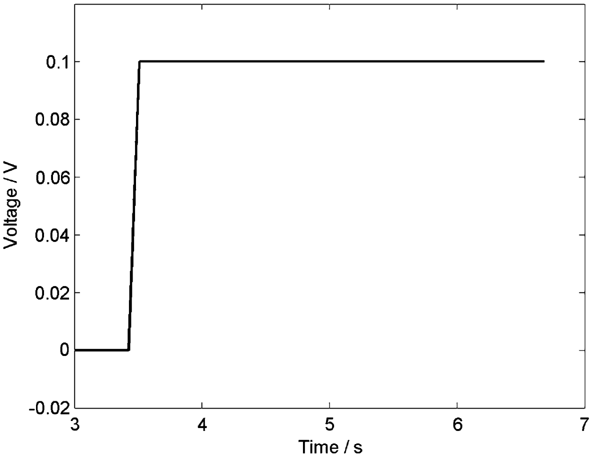
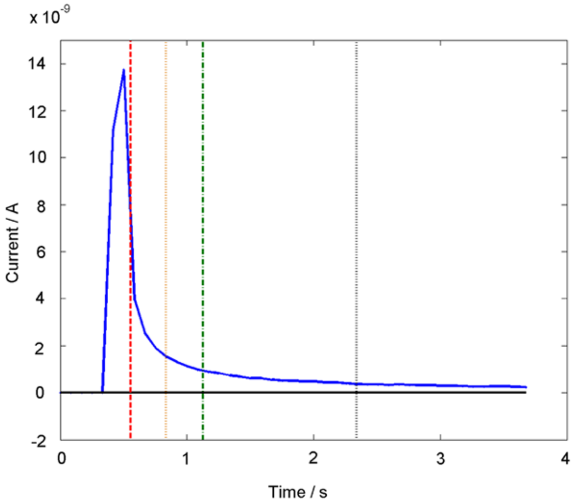


Fig. 4 An example response spike. *Red dashed line*: τ_{50}; *orange dotted line* τ_{90}; *green dot-dashed* τ_{95}; *blue dotted* τ_{99}. *Horizontal purple dot-dashed line* is i_∞ and the spike height is i_0



is undertaken, we shall assume that $i(t)$ is not dependent on i_0 as this is what the experimental evidence seems to suggest.

Thus, the time-dependent current response, $I(t)$, is assumed to be of the form:

$$I(t) = i_\infty + i(t) \tag{2}$$

where $i_0 < i(t) < 0$.

The current response to the voltage is thus:

$$\Delta I = \frac{V}{R(T)}. \tag{3}$$

The time taken to get to $i(t) = 0$ is the equilibration lifetime which we shall call τ, and this lifetime is the short-term memory of the memristor and relates to its dy-

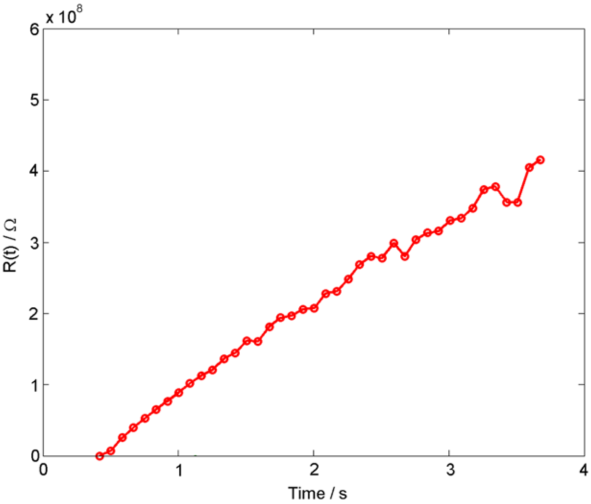


Fig. 5 The resistance profile for the memristor subjected to the voltage in Fig. 3 as calculated from the source voltage and measured current

namical properties; from longer time spike studies with our devices, we know that τ is approximately 3.3 s. We shall define the concept of the equilibration frequency as the ‘frequency’ associated with changing a discretized triangular voltage waveform such that each voltage step n lasts for τ seconds.

We know that

$$q_e = \int I(t)dt. \tag{4}$$

thus, the total measured charge in a memristor spike is

$$\Delta q_{\text{spike}} = \int_{t=0}^{\tau} i(t)dt + i_{\infty}\tau. \tag{5}$$

This number includes all the charge-carrying species in the system. Knowledge of this number may help us elucidate the mechanism of the spikes. For our example system shown in Fig. 4, we have an i_0 of 1.37×10^{-8} A, an i_{∞} of 2.40×10^{-10} A, with the τ_{50} of 0.56 s and an τ_{90} of 0.84 s, which shows how quick the fall off is (and τ_{95} of 1.13 s and τ_{99} of 2.34 s, as drawn in Fig. 4). The resistance profile for the memristor subject to a voltage step as shown in Fig. 3 is shown in Fig. 5. This is approximately a straight-line which is interesting as it is not required to be by memristor theory and tells us that the spike current response depends on a quantity in the system that is varying with linearly time.

2.3 Theoretical Model of Single Spikes

Here we demonstrate how the experimentally-measured memristor spikes can be described with our theoretical model.

2.3.1 The Memory-Conservation Model

The memory-conservation model of memristance [18] is a recently announced theoretical model that relates real world q and φ to Chua's constitutive equations and has been successful in modeling our memristors [19]. The memory-conservation theory has the concept of a memory property, the physical or chemical attribute of the device that holds the memory of the device. In titanium dioxide (and many others) it is related to the number of the oxygen vacancies. The presence of oxygen vacancies allows the creation of a doped form of titanium dioxide TiO_{2-x} which is more conducting than the un-doped (TiO_2) form. The memory-conservation theory requires that we calculate the memristance from the point of view of the memory property, i.e. the ions.

The memory-conservation model consists of sum of two components: the memory function, M_e, and conservation function, R_c, both of which are resistances dependent on the motion of the memory property, $q(t)$ and thus time-dependent. The memory function has a fitting parameter c_m within the model to account for the conversion between the material's resistance as for an oxygen vacancy to that for an electron. The conservation function has the fitting parameter c_c which accounts for the resistivity of the undoped material, ρ_{off}, which may not be the same as the bulk titanium dioxide. R_{on} is the final fitting parameter and relates to the resistivity of the doped material, which is the memristor in the equilibrated state and any resistance in the wires. The fitted equation is

$$I(t) = \frac{V}{R_{\text{on}}} - \frac{V}{c_c R_c(t) - c_m M_e(t)} . \tag{6}$$

2.3.2 Fitting the Memory-Conservation Model to the Spikes

As Figs. 6 and 7 show, the memory-conservation model fits these spikes quite well and much better than an exponential fit. For the positive spike, $c_M = 3.83 \times 10^6$, $c_c = 1.76 \times 10^6$ and $V/R_{\text{on}} = 2.97 \times 10^{-9}$, with a summed square of residuals of 1.61×10^{-17}. For the negative spike, $c_M = 1.06 \times 10^6$, $c_c = 1.86 \times 10^{-6}$ and $V/R_{\text{on}} = -3.16 \times 10^{-9}$, with a summed square of residuals of 1.63×10^{-17}. For the exponential fit, $I(t) = Ae^{\lambda t}$, and $A = 3.96$, $\lambda = -19.5$ with a summed square of residuals of 2.43×10^{-15}. The exponential fit could be fit to either the short time spike or the long time tail but not both, the short term spike fit goes erroneously to zero and the long-term spike fit grossly over-estimates the size of the spike. Furthermore, there is no experimental justification for using an exponential fit, unlike the memory-conservation fit. This model can be utilized to perform simulations of memristor spiking networks to test out possible neuromorphic architectures.

2.3.3 What Is the Mechanism?

The memory property of these memristors is the oxygen ions, specifically the vacancies caused by their absence which are usually viewed as positive holes in a

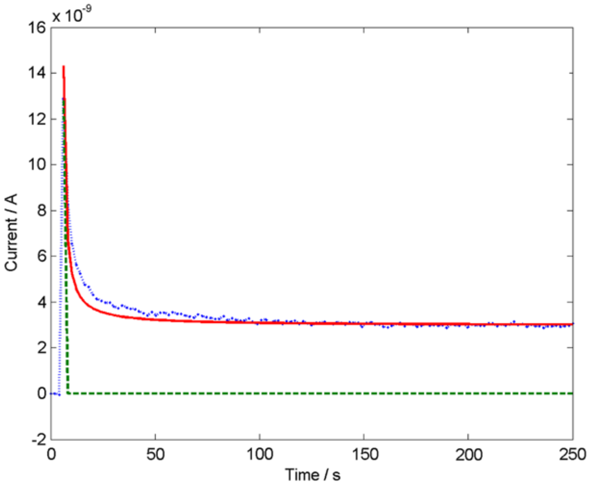


Fig. 6 A longer-term spike response fit by the memory-conservation theory. The memory-conservation theory fits the experimental data well, the best result fitting the data with an exponential is added as a comparison. *Blue dots*: experimental data, *red line*: memory-conservation fit, *green line*: exponential fit to the spike

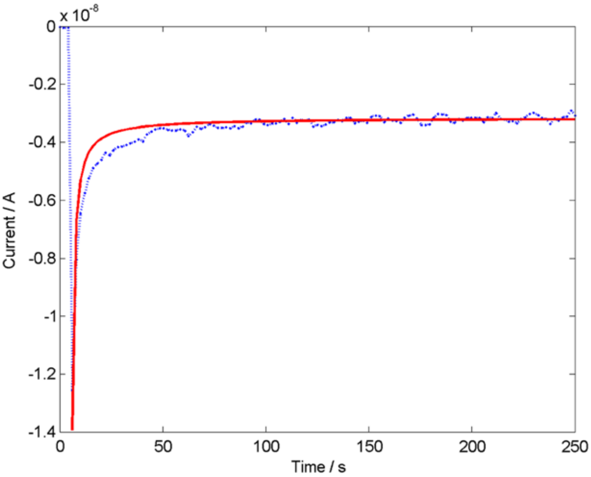


Fig. 7 A longer-term negative spike, demonstrating that the negative spikes are fit equally well by the memory-conservation theory. *Blue dots*: experimental data, *red line*: memory-conservation fit

semi-conducting material. We suspect that the motion of these ions is behind both the spikes and the memristance as we postulate that the two are the same phenomena. The current that flows at $t = 0$ s may be the ionic current, which would have a greater inertia, and thus takes longer to stop compared to the electrons, which may explain the cause of the memristor's hysteresis. This current flow can also explain the open-loop memristors (suggested by Pershin and di Ventra [20] to explain experimental results such as [21] which are similar to ones seen in our labs and others'). The spike shape would then be the result of the equilibrating of the ionic current to a change in voltage. We expect that the timescale and dynamics of the spikes will relate to the frequency effects seen in memristors. How-

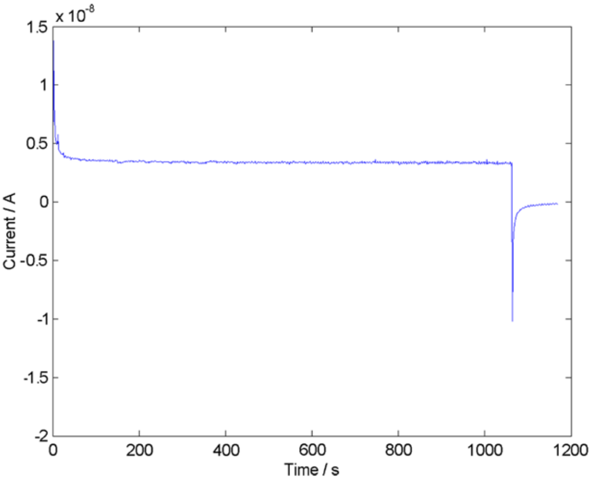


Fig. 8 A single memristor response. The voltage (+0.4 V) is turned on at $t = 0$ s, causing a positive current spike, and turned off at 1064 s, which causes a negative current spike. These current spikes are a feature of memristance

ever, there is much further experimental work to be done to prove this mechanism.

2.3.4 Single Memristor Circuits

The current response of a typical memristor to constant voltage is given in Figs. 6 and 7. For this experiment the memristor was taken to the test voltage at 0 seconds and the current recorded as the memristor 'equilibrated'. This was done for two voltages, +1 V and −1 V. There is asymmetry in the spike responses as the memristors are asymmetric. This asymmetry arises from the way the memristors are made, the bottom electrodes are deposited in vacuum and then the $Ti(OH)_4(sol) \rightarrow TiO_2(gel)$ reaction happens on top of them. The top electrodes are later deposited onto the reacted gel. It seems likely that the local environment around the Al-gel layer boundary is not exactly the same at both ends. For this reason, the devices are always tested the same way round (source to the bottom electrode) and a slightly larger hysteresis loop is seen in the positive voltage part of the curve compared to the negative voltage part.

When voltage is changed we expect a current spike, as this has been seen in all our tests [22] and this is the D.C. action of memristors (forthcoming paper). For the long time experiments shown later in this paper, the single memristor response is shown in Fig. 8. The spike from the original voltage switch occurs at the start, and then negative spike at 1000 timesteps that results from the change from +0.4 V → 0 V can be clearly seen. The signal from the middle section is blown up in Fig. 9 to show the quality of the noise; there are no oscillations or large spikes (the spike at the end is the switching spike of the voltage source being turned off).

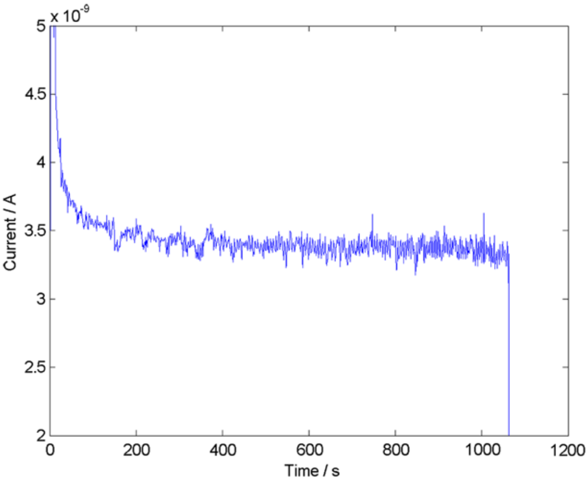


Fig. 9 A close up of the I–t plot in Fig. 8 to show the character of the noise from a single memristor

2.4 The Memory-Conservation Theory as Applied to Memristor Spikes

Theoretically, the voltage step is a discontinuous function and the voltage changes from voltage A, V_A, to voltage B, V_B, in an infinitesimal, i.e. $\Delta V = \frac{V_B \rightarrow V_A}{\delta t}$, $\delta t \rightarrow 0$. Experimentally this is not the case of course, but the response time-scale of the memristor is long enough that we needn't worry about this approximation.

Thus to elucidate what happens to the memristor during a current spike, and how the final current i_∞ is determined, we take differences of the memory-conservation theory. We shall assume our device is a TiO_2 memristor, with oxygen vacancies acting as the memory property [1].

As a reaction to the voltage step, we get a current spike, Δi, which can be expressed as a volume current within the device as ΔJ as given by:

$$\Delta \vec{J} = \left\{ \frac{\Delta q_v \mu_v L}{vol}, 0, 0 \right\} \tag{7}$$

for vacancies moving in the $+x$ direction where q_v is the charge in that volume due to the vacancies, μ_v is the ion mobility of vacancies and L is the average electric field causing the movement of the vacancies and vol is the volume full of moving ions.

The Chua memristance as experienced by the ions is:

$$\Delta M_q\big(\Delta q_v(t)\big) = U X \mu_v \Delta P_k\big(\Delta q_v(t)\big)\,, \tag{8}$$

where we have gathered up the constants and explicitly included P_k's dependence on q_v.

Equation (8) can be considered as three separate parts:

1. U, the universal constants: $\frac{\mu_0}{4\pi}$.

2. X, the experimental constants: DEL.
3. the material variable: $\mu_v P_k$, this includes the physical dimensions of the doped part of the device and the drift speed of the dopants.

Writing out the differences explicitly for Eq. (8) we end up with:

$$M(B) = M(A) + UX\mu_v\big[P_k(q_B) - P_k(q_A)\big], \tag{9}$$

which allows us to calculate how the final Chua memristance from knowledge of the peak and final currents. The Chua memristance is written for the vacancy charge, so to put it into the standard format for the electronic current we need to scale it thus:

$$R_M = C_M M, \tag{10}$$

where R_M is the electronic resistance of the doped part of the memristor and C_M is a fitting coefficient.

2.5 Conservation Function

The memory-conservation model describes a memristor by being the sum of the memory and conservation functions (both written for the electrons) and this then gives us the following expression for the change in time-varying resistance, $R(t)$, as measured after a change from $V_A \rightarrow V_B$ as:

$$\begin{aligned}\varDelta R(t) = c_m M(A) + R_{\text{con}}(A) + \frac{\rho_{\text{off}} D}{EF}\\ + c_M UX\mu_v\big[p_k\big(q_B(t)\big) - p_k\big(q_A(t_T)\big)\big]\\ - \frac{L\rho_{\text{off}}\mu_v[q_B(t) - q_A(t_T)]}{E^2F^2v_d},\end{aligned} \tag{11}$$

This equation has two parts:

1. S, the time-invariant part, which is: $c_m M(A) + R_{\text{con}}(A) + \frac{\rho_{\text{off}} D}{EF}$
2. Y, the time variant part: $c_M UX\mu_v[p_k(q_B(t)) - p_k(q_A(t_T))] - \frac{L\rho_{\text{off}}\mu_v[q_B(t) - q_A(t_T)]}{E^2F^2v_d}$,

the last two terms which are both dependent on q.

In the above Eq. (11) highlights a few subtleties of the model. p_k and q are time-dependent and thus change after the voltage step from $V_A \rightarrow V_B$. If we ask the question what the difference will be between the equilibrated current at V_A and that at V_B, $\varDelta R_{A_\infty \rightarrow B_\infty}$ Eq. (11) collapses to:

$$\begin{aligned}\varDelta R = c_m M(A) + R_{\text{con}}(A) + \frac{\rho_{\text{off}} D}{EF}\\ + c_M UX\mu_v\big[p_k\big(q_B(\tau)\big) - p_k\big(q_A(\tau)\big)\big]\end{aligned}$$

$$- \frac{L\rho_{\mathrm{off}}\mu_v[q_B(\tau) - q_A(\tau)]}{E^2F^2v_d},$$

which is time invariant and allows us to predict the value of the equilibrated current after a voltage step from the equilibrated current from the step before.

What if there was previous step in which the device did not equilibrate to i_∞? This would happen if the voltage was changed quicker than τ, i.e. T where $T < \tau$. The $q_A(t_T)$ is not $q_A(\tau)$ and thus needs to be shifted by its value as a proportion of τ. As an example, if we sped the voltage ramps up to 90 % of the equilibration frequency, q_A would be $q_A(\tau_{90})$ and the length of a time step would be τ_{90}. At first glance it might appear that this would merely modulate the starting point for $q_B(t)$, which, at times under $t < \tau$, this would be time dependent. But there is the interaction between $q_B(t)$ and $q_A(t_T)$, the memristor hasn't finished responding to V_A and that response should be mixed in with V_B, further complicating predictive efforts.

3 Constructionist Approach to Memristor Networks

The relation between memristors and neuromorphic computing dates back to 1976 when Chua and Kang expanded the idea of the memristor to a memristive system (which has two state variables rather than one as for a memristor) and suggested that the Hodgkin-Huxley model of the nerve axon could be improved by in incorporating memristors in place of the non-linear time dependent resistors [23]: an idea that wasn't theoretically demonstrated until 2012 [12, 13]. Meanwhile the scientific community has concentrated on the idea of using memristors as synapses rather than axons: simulations have shown that memristive connections could be used to reproduce spike-time dependent plasticity [8] (the process by which synapses adjust their connection weight to implement Hebbian learning [24]) and even implemented as synapses in evolved spiking networks simulations [7]. As discussed in the first half of this chapter, our memristors (and other group's) possess the ability to spike. Thus we decided to investigate whether memristors could replicate neuronal architecture and produce dynamics associated with neurons, such as brainwaves or spike trains.

When assembling multi-memristor systems in the laboratory, it is sensible to first ask which circuits are being designed by theorists and tested by simulationists for use with memristors, and overwhelmingly they investigate the Chua circuit. The original Chua circuit [25] was created to demonstrate that chaos was a real phenomena (not merely due to rounding errors in the computer simulations) and it has been suggested [13] that neurons are poised at the edge of chaos, so, it is worth investigating chaotic dynamics (and the related field of complexity) if we are interested in making circuits for neuromorphic computing.

There have been a plethora of different versions of and alterations to the Chua circuit, as summarized in [26], but the simplest version built [27] consists of one inductor, one resistor, two capacitors and a component called Chua's diode: a non-linear circuit element usually fabricated from several other circuit components including op amps. Itoh and Chua were the first to replace Chua's diode with a memristor [28]; they worked with the concept of an active memristor (a memristor is a passive device, but a circuit of a negative resistance and memristor can be viewed as an active memristor). There have been many papers since detailing the rich behaviour and chaotic properties of Chua circuits containing memristors (e.g. [29–31] and [32]). These contain simulations which use Chua's equations [1] for the perfect theoretical memristor and electronic experiments which replace the memristor with a circuit equivalent, presumably due to the difficulty in obtaining an actual memristor to use.

An important step forward in the direction of real world functionality was Buscarino's paper [33] where Chua's diode in Chua's circuit was modelled using Strukov et al.'s phenomenological model [2] which is based on real world measurables and relates to a real memristor. The resulting simulation demonstrated chaotic behaviour [33]. This paper used a pair of Strukov memristors connected in anti-parallel to give a symmetrical I–V curve as a replacement for Chua's diode. They then used a voltage frequency that took the memristor to its limits (i.e. maximum and minimum resistance) to introduce asymmetry and richer behaviour. However, from this data it is not known whether the chaotic behaviour they observed in their simulations arose from the memristors themselves or from the interactions of the errors in the model, which (even with windowing functions) is weakest at the edges of the memristor.

A recent experimental result of a possible neuromorphic building block was the Hewlett-Packard (HP) 'neuristor': a circuit consisting of two memristors and two capacitors (and a load resistor) which gives 'brainwave'-like dynamics from a constant voltage source [34]. This circuit also had the memristors in anti-parallel.

Another area of interest is how few components a chaotic circuit can be made with. A recent paper [35] suggested that the simplest circuit capable of producing chaos could be made with three components: a capacitor, an inductor and a memristor. Thus, circuits involving memristors, capacitors and inductors look likely to product interesting dynamics.

Furthermore, according to Chua [1] the linear combination of memristors in a circuit with only one input and one output to that circuit is indistinguishable from a memristor with a memristance value calculable by standard series and parallel resistor adding rules ('A 1-port containing only memristors is equivalent to a memristor' [1]), i.e. the memristors add up in series and in parallel similarly to resistors, which would suggest that a circuit made up of only memristors would be a trivial circuit with the same behaviour as a single memristor.

However, due to our observation of the memristor spikes, we decided to test whether circuits consisting of only real-world memristors would give rise to rich behaviour, rendering the complications of including capacitors or op amps unnecessary.

Therefore we investigated the effects of interacting memristors using real world memristors. By using real world memristors we are able to make use of the memristor's actual behaviour, whereas theoretical models of the memristor are less useful in this regard. We shall investigate how pairs of memristors interact and test the assertion that two memristors in series (or parallel) addressed only by their joint one port entry (i.e. there is one wire coming out and going in to that part of the circuit) are indistinguishable from a single memristor. We will look at several different three memristor circuits.

3.1 Methodology

Here we outline the types of memristors we manufacture, the constructed circuits and experimental details.

3.1.1 Memristor Types: Curved and Triangular Switching Behaviours

Our memristors show two characteristic behaviours: A: 'curved' pinched hysteresis curves and B: 'triangular' pinched hysteresis curves, as previously observed [36] and shown in Fig. 10. The type A memristors can be modelled by the standard memory-conservation theory and are thus close to Chua's theoretical memristor [1]. Type B memristors have an ohmic high conduction state as evidenced by a straight-line on the I–V graph. We suspect that the 'triangular' type B memristors switch via the formation and breaking of conducting filaments and the addition of a conducting filament to the memory-conservation theory of memristance [19] models this situation well. The memristors used in this work were classified based on observed I–V curves and all further results are experimental, not simulated.

3.1.2 Memristor Circuits Tested

The tested circuits are presented in Table 1. From the literature and our own intuition, we expected that two memristors in anti-parallel configuration would be the most likely 2-memristor circuit to exhibit rich dynamics, see experiment no. 4 in Table 1. For this reason, we decided to count the number of anti-parallel interactions in the 3-memristor circuits and also take note of memristors wired up with opposite polarity in a series circuit (anti-series), which is experiment 2 in Table 1.

3.1.3 Experimental Details

All experiments were performed with a Keithley 2400 Sourcemeter. For most the I–t curves, the memristor circuits were taken to +0.4 V for 1000 timesteps or 1.06 s, the voltage source was then switched to 0 V and data gathered for a further 100 timesteps. To investigate whether a slow changing voltage had an effect, a sinusoidal voltage of 1600 timesteps of 2 s was used. In all experiments voltages were kept very low to avoid the creation of filaments via Joule heating which would lead to filamentary memristors switching into lower resistance states.

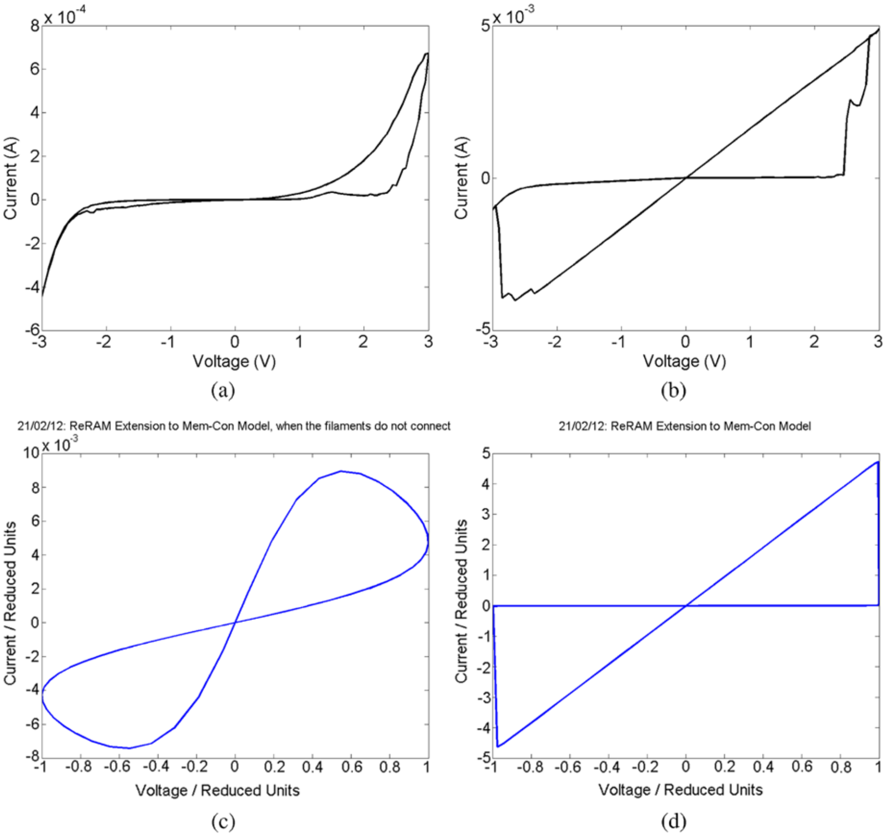


Fig. 10 Examples of the two different types of memristors: (**a**) experimental curved memristor, (**b**) experimental filamentary memristor, (**c**) theoretical curved type memristor, (**d**) theoretical filamentary memristor

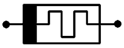
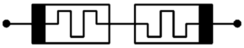
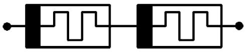
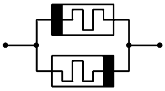
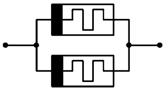
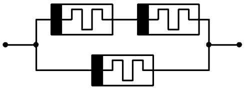
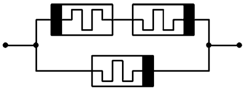
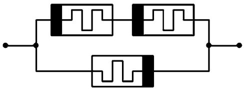
3.2 Two Memristor Circuit Results

The results for the 2-memristor circuits, numbers 2–5 in Table 1, are presented here.

3.2.1 Type A (Curved) Memristors in 2-Memristor Circuits

Using type A memristors in series (as in Fig. 13 gave an I–V profile similar to that for one memristor (compare with Fig. 8) other than an unexpected spike near the end. However, putting two memristors in ‘anti-series’ as in Table 1 gave the odd results of increased noise and several spiking events, as shown in Fig. 11. The effect of anti-polarity series interactions is this richer behaviour. Similarly, the 2-memristor parallel interactions, as shown Figs. 14 and 5 show more noise and spiking events. Note that only the two memristors in series show the expected spike at the start and

Table 1 Constructed experiments. Anti-parallel memristor interactions are represented by 'p', anti-polarity series memristor interactions are represented by 's'

Experiment Number	Wiring Diagram	Number and Type of anti-polarity interactions
1		0
2		1s
3		0
4		1p
5		0
6		0
7		1s, 1p
8		2p

when the voltage is switched off (as was seen for a single memristor), therefore only circuit 3 can be considered equivalent of a single memristor 1-port.

3.2.2 Type B (Triangular) Memristors in 2-Memristor Circuits

Figure 12 shows the results from the constructions of circuits 2, 3, 4 and 5 with filamentary memristors. These circuits show a richer behaviour with the emergence of oscillatory type behaviour in circuits, 2, 3 and 5. Filamentary memristors in a series (circuit 3) do not act like single memristors. Also, in some cases the filaments partially connect, as seen in Fig. 12c.

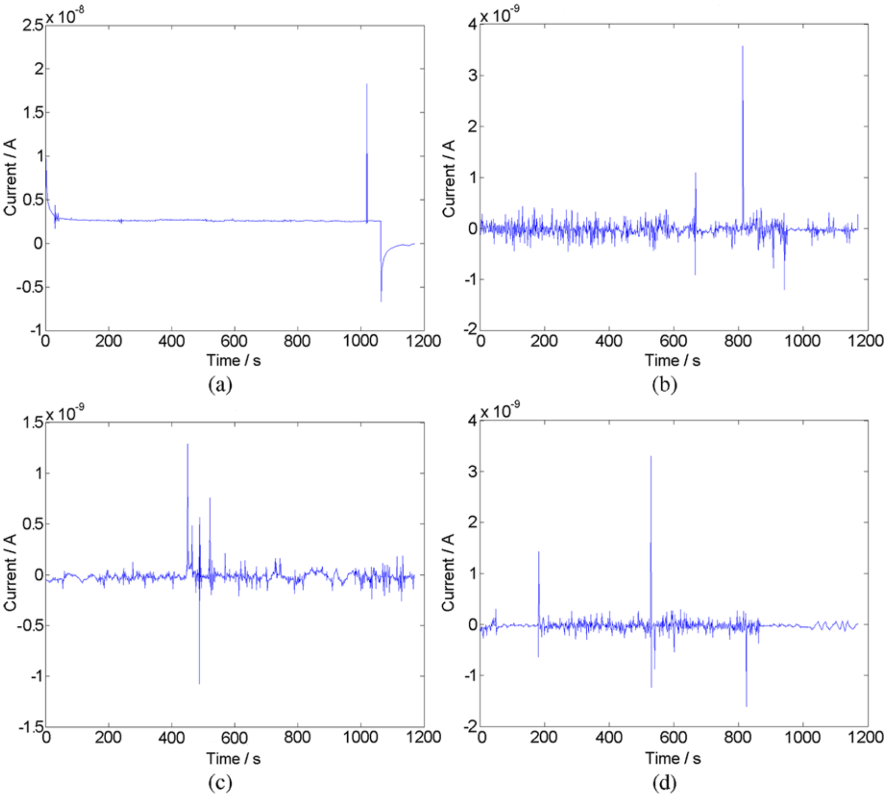


Fig. 11 Results for type A memristors: (**a**) two A memristors in series, opposite direction, circuit 2 in Table 1, (**b**) two type A memristors in series, same direction, circuit 3 in Table 1, (**c**) two type A memristors in anti-parallel, circuit 4 in Table 1 and (**d**) two type A memristors in parallel, circuit 5 in Table 1

3.3 *Three Memristor Circuit Results*

The most interesting results were obtained with circuit 7 (see Table 1) which has two memristors wired up in anti-polarity series and one memristor in parallel to this sub-circuit. This gives the circuit two anti-polarity interactions of two different types, one in series and one in parallel, and this seems to give the circuit a richer non-linear dynamics compared to circuit 6 (see Table 1) where the memristors are all wired up with the same polarity.

3.3.1 Type B (Triangular) 3-Memristor Circuits

A dynamical system can exhibit chaotic behaviour if it has at least three state variables, so we chose to create a circuit with three memristors, which gives us the

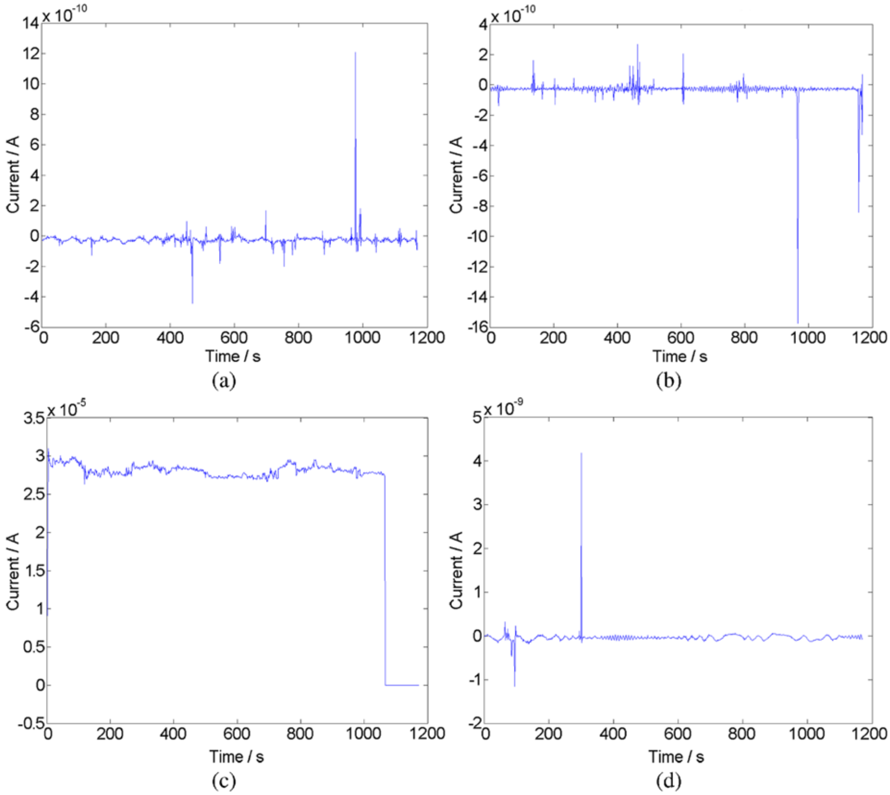


Fig. 12 Results for type B (triangular) memristors: (**a**) Two B memristors in series, opposite direction, circuit 2 in Table 1, (**b**). Two B memristors in series, same direction, circuit 3 in Table 1, (**c**). Two B memristors in anti-parallel, circuit 4 in Table 1 and (**d**). two B memristors in parallel, circuit 5 in Table 1

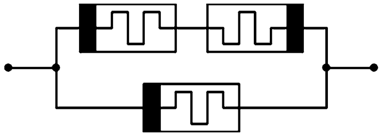


Fig. 13 A three memristor circuit with two opposite polarity interactions and one same polarity interaction

following three separate state variables, the current through the circuit, and the voltage across two of the memristors (the third being determined by the other two in a system kept at a constant voltage). In order to maximise the antiparallel interactions of the circuit, the memristors were wired up, two in series in reversed order, with one in parallel to the two in series as shown in Fig. 13. It was thought that the memristors would spike with the change of voltage and this would cause a change in resistance within a single memristor, which, with this circuit set-up would lead to a voltage change across the other memristors and thus further spikes. We used type B memristors for this circuit.

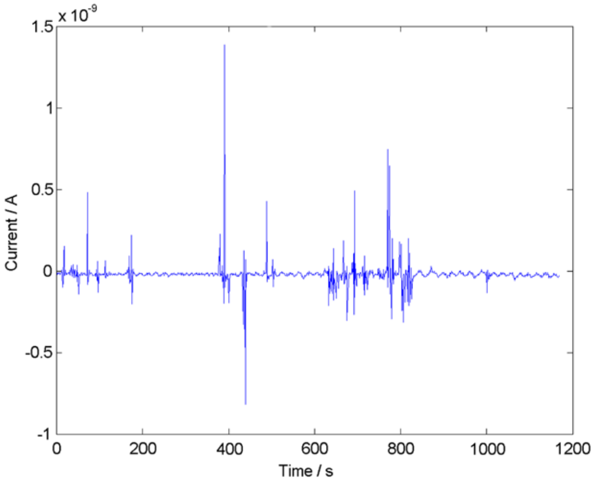


Fig. 14 The current response for the three memristor circuit shown in Fig. 13. There seems to be an oscillatory behaviour as well as periods of spiking that resembles spike-trains in neural networks

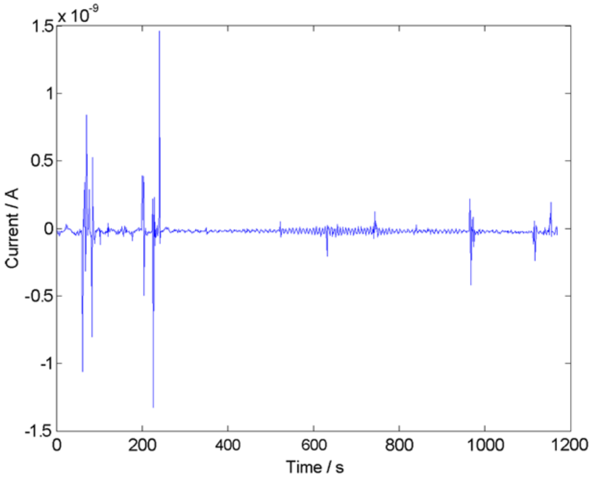


Fig. 15 Another typical I–t profile for the circuit shown in Fig. 13

Typical results for this circuit are given in Fig. 14. Comparing this with the expected curve in Fig. 8 for one memristor shown in Fig. 8, we can see differences. The large spike at the start has vanished, as has the one at the end. We see oscillations in the base line, with spontaneous spiking overlaid over the top. Figure 15 shows a later run where we see sections of oscillations of different frequency. Several runs of this circuit were done to see if there was a repetition in the spiking pattern and thus if the circuits were following long-term periodic dynamics, this was not the case.

We attempted to effect this oscillation by running a very slow I–V curve, the results of which are shown in Fig. 16 (the whole data is for one period). This does not show the expected response for a single memristor, or any change in the 'baseline' as a result of the changing voltage. The expected spikes from the one memristor

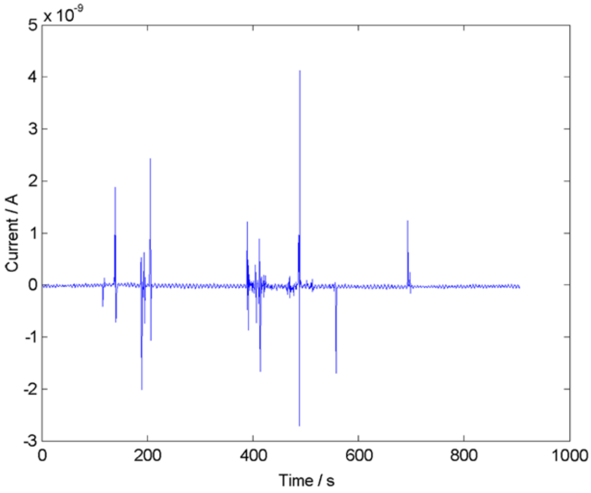


Fig. 16 An I–t curve for a very slow sinusoidally varying voltage

circuit have been suppressed or delayed. The frequency of the baseline oscillations seems more regular than for the constant voltage.

3.3.2 Type A 3-Memristor Circuits

Do these results mean that multi-memristors circuits do not combine as expected? Not necessarily. We decided to repeat the tests with three type A memristors which are closer to the theoretical perfect Chua memristors (type A from paper [37]). We specifically chose three memristors that had similar looking I–V curves that operated over a similar current range to try and decrease the compositional complexity of the circuit.

Figure 17a shows the same memristors wired up as circuit 7, and the output current looks like a single memristor. For the type A memristors, we found that three memristors in a circuit wired up with the same polarity (i.e. as in circuit 6) behave qualitatively just like one memristor. For this circuit, we see occasional switching events with a decay, see Fig. 17b, which strongly suggests that the more event rich behaviour seen in the other memristor systems are to do with interacting switching spikes.

4 Conclusion

Single memristors exhibit current spiking behaviour when subjected to voltage ramps. This behaviour is quicker than the switching measured over a pinched hysteresis curve and reproducible, and thus offers interesting technological applications. These spikes are well-modelled by the memory-conservation theory of memristance, showing further successes of that theory.

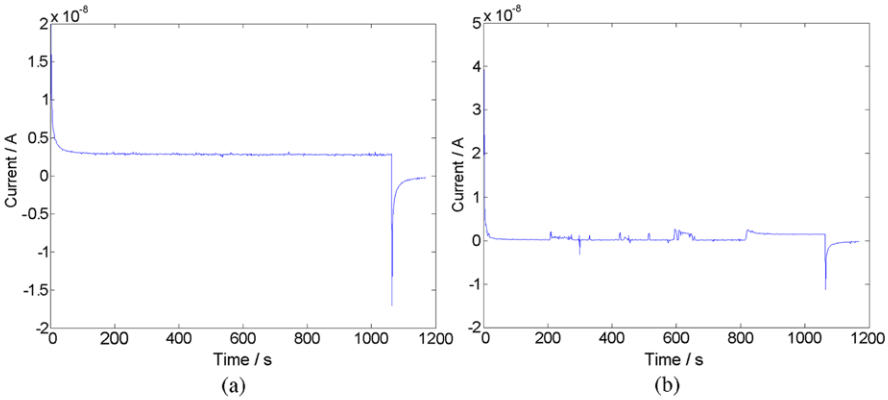


Fig. 17 Results for type A memristors: (**a**) three memristors in circuit 7, (**b**) three memristors in circuit 6

Using both type B (those thought to be filamentary) and type A (those closer to Chua's theory) memristors, 2 and 3 memristor networks have been investigated. If we use type A memristors, the assertion that circuits containing only memristors is equivalent to a single memristor seems to be true, which is sensible as they are the closest to the theoretical ideal. Intriguingly, this is not the case with type B memristors. This suggests that circuits of mixed types of memristors may be enough to make rich behaviour, possibly even a version of the Chua circuit using only memristors.

The emergence of oscillatory dynamics is interesting and shows that a capacitor is not necessarily needed to create such dynamics. The oscillations could be explained by time-delayed spike interactions within the circuit or interacting oscillators created by the movement of ions (the memory property) around the dynamic equilibrium point.

This work suggests that the chaotic dynamics seen in [33] and the oscillatory dynamics seen in [34] could be due to the sub-circuit of two anti-parallel memristors and suggests that their results are experimentally verifiable and not due to weaknesses in the model.

The next step is to quantify these oscillations and attempt to test whether the circuits shown here are exhibiting chaotic dynamics or not. We are also exploring the concepts of compositional complexity which were dealt with in a qualitative sense.

It is possible that the oscillations are an emergent network property, similar in causal nature to brain waves and thus could be used as models for spiking neuronal systems (including both neural nets and biological brains). The clusters of spikes may be stochastic or chaotic in origin: further work is needed to determine this. However they arise, it may be possible to use them as a control system.

References

1. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**, 507–519 (1971)
2. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
3. Prodromakis, C.T.T., Chua, L.: Two centuries of memristors. Nat. Mater. **11**, 478–481 (2012)
4. Erokhin, V., Berzina, T., Fontana, M.P.: Hybrid electronic device based on polyaniline-polyethylenoxide junction. J. Appl. Phys. **97**, 064501 (2005)
5. Askinaga, H., Shima, H.: Resistive random access memory (ReRAM) based on metal oxides. Proc. IEEE **98**, 2237–2251 (2010)
6. Erokhin, V., Berzina, T., Camorani, P., Smerieri, A., Vavoulis, D., Feng, J., Fontana, M.P.: Material memristive device circuits with synaptic plasticity: learning and memory. BioNanoScience (2011). doi:10.1007/s12668-011-0004-7
7. Linares-Barranco, B., Serrano-Gotarredona, T.: Exploiting memristance in adaptive asynchronous spiking neuromorphic nanotechnology systems. In: 9th IEEE Conference on Nanotechnology, pp. 601–609 (2009)
8. Zamarreno-Ramos, C., Carmu nas, L.A., Pérez-Carrasco, J.A., Masquelier, T., Serrano-Gotarredona, T., Linares-Barranco, B.: On spike-timing dependent plasticity, memristive devices and building a self-learning visual cortex. Front. Neuormorphic Eng. **5**, 26 (2011)
9. Kosta, S.P., Kosta, Y.P., Bhatele, M., Dubey, Y.M., Gaur, A., Kosta, S., Gupta, J., Patel, A., Patel, B.: Human blood liquid memristor. Int. J. Med. Eng. Inform. 16–29 (2011)
10. Lütken, C.A., Grimes, S., Martinsen, O.G.: Memristive properties of human sweat ducts. In: World Congress on Medical Physics and Biomedical Engineering, vol. 25/7, pp. 696–698 (2009)
11. Gale, A.A.E., de Lacy Costello, B.: Are slime moulds living memristors? (2013). arXiv:1306.3414v1
12. Chua, L., Sbitnev, V., Kim, H.: Hodgkin-Huxley axon is made of memristors. Int. J. Bifurc. Chaos **22**, 1230011 (2012)
13. Chua, L., Sbitnev, V., Kim, H.: Neurons are poised near the edge of chaos. Int. J. Bifurc. Chaos **11**, 1250098 (2012)
14. Gergel-Hackett, N., Hamadani, B., Dunlap, B., Suehle, J., Ricther, C., Hacker, C., Gundlach, D.: A flexible solution-processed memristor. IEEE Electron Device Lett. **30**, 706–708 (2009)
15. Gale, E., Pearson, D., Kitson, S., Adamatzky, A., de Lacy Costello, B.: Different behaviour seen in flexible titanium dioxide sol-gel memristors dependent on the choice of electrode material. In: Technical Digest of Frontiers in Electronic Materials, pp. 577–578. Wiley-VCH, Weinheim (2012)
16. Wu, J., Mobly, K., McCreery, R.L.: Electronic characteristics of fluorene/TiO$_2$ molecular heterojunctions. J. Chem. Phys. **126**, 024704 (2007)
17. Bi, G.-Q., Poo, M.-M.: Spike timing for LTP and LTD in. Cult. J. Neurosci. **18**, 10553–10565 (1998)
18. Gale, E.: The memory-conservation model of memristance (forthcoming)
19. Gale, E.M., de Lacy Costello, B., Adamatzky, A.: Filamentary extension of the Mem-Con theory of memristance and its application to titanium dioxide Sol-Gel memristors. In: 2012 IEEE International Conference on Electronics Design, Systems and Applications (ICEDSA 2012), Kuala Lumpur, Malaysia, November 2012
20. Troitsky, V.I., Berzina, T.S., Fontana, M.P.: Mater. Sci. Eng. **22**, 239 (2002)
21. Troitsky, V.I., Berzina, T.S., Fontana, M.P.: Synth. Met. **129**, 39 (2002)
22. Gale, E., de Lacy Costello, B., Adamatzky, A.: Observation, characterization and modeling of memristor current spikes. Appl. Math. Inform. Sci. **7**, 1395 (2013)
23. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**, 209–223 (1976)
24. Hebb, D.: The Organisation of Behaviour. Wiley, New York (1949)
25. Masumoto, T.: A chaotic attractor from Chua's circuit. IEEE Trans. Circuits Syst. **CAS-31**, 1055 (1984)

26. Chua, L.: Chua's circuit: ten years later. IEICE Trans. Fundam. E **77-A**, 1811–1821 (1994)
27. Madan, R.N.: Chua's Circuit: A Paradigm for Chaos. World Scientific, Singapore (1993)
28. Itoh, M., Chua, L.: Memristor oscillators. Int. J. Bifurc. Chaos **18**, 3183–3206 (2008)
29. Messias, M., Nespoli, C., Botta, V.A.: Hopf bifurcation from lines of equilibria without parameters in memristor oscillators. Int. J. Bifurc. Chaos **20**, 437–450 (2010)
30. Bao, B.C., Liu, Z., Xu, J.P.: Steady periodic memristor oscillator with transient chaotic behaviours. Electron. Lett. 237–238 (2010)
31. Bao, B.-C., Xu, J.-P., Liu, Z.: Initial state dependent dynamical behaviors in a memristor based chaotic circuit. Chin. Phys. Lett. **27**, 070504 (2010)
32. Bo-Cheng, B., Ping, X.J., Guo-Hua, Z., Zheng-Hua, M., Ling, Z.: Chaotic memristive circuit: equivalent circuit realization and dynamical analysis. Chin. Phys. B **20**, 120502 (2011)
33. Buscarino, A., Fortuna, L., Frasca, M., Gambuzza, L.V.: A chaotic circuit based on Hewlett-Packard memristor. Chaos **22**, 023136 (2012)
34. Medeiros-Ribeiro, G., Pickett, M.D., Williams, R.S.: A scalable neuristor built with Mott memristors. Nat. Mater. **16**, 114–117 (2012)
35. Muthuswamy, B.: Memristor based circuit chaos. IETE Tech. Rev. **26**, 1–15 (2009)
36. Gale, E., Pearson, D., Kitson, S., Adamatzky, A., de Lacy Costello, B.: Different behaviour seen in flexible titanium dioxide sol-gel memristors dependent on the choice of electrode material. In: Technical Digest of Frontiers in Electronic Materials. Nature Conference, pp. 577–578. Wiley-VCH, Weinheim (2012)
37. Gale, E., Pearson, D., Kitson, S., Adamatzky, A., de Lacy Costello, B.: Aluminium electrodes effect the operation of titanium oxide sol-gel memristors (2011). arXiv:1106.6293v1

Organic Memristive Devices and Neuromorphic Circuits

Victor Erokhin

Abstract Bio-inspired computational systems must be based on elements, involved, similarly to the brain, in both memorizing and processing of the information. This paper is dedicated to organic memristive devices—elements that were designed and constructed for mimicking the most important properties of synapses, responsible for Hebbian type of learning. We will consider the architecture of the device and its properties, as well as circuits and networks with adaptive features.

1 Introduction

There is a significant difference in the architecture of the computers and the brain: in the computer the memory and the processor are different devices. The information in this case plays a rather passive role—it can be recorded, accessed, cancelled, but it does not vary connections and properties of the processor. In the brain, instead, the same elements are used for both memorizing and processing of the information. Such architecture is responsible for the possibility of learning of the system at a hardware level. The information plays an active role in this case. It is not only memorized, but it varies connections within processor, what makes it more effective for the resolving of similar tasks in the future.

Nervous system and brain are composed from neurons. Each neuron has several dendrites bringing input signals to the neuron. In addition, for each neuron there is a single axon that provides a further propagation of the signal, when the sum of the inputs overcomes a certain threshold value. Synapse is a very important element of the nervous system. It is a contact point of an axon of one neuron with a dendrite of the other one. An important property of synapses is the possibility to vary the weight function of the signal transmission according to its previous functioning. Such property is a key feature for so-called Hebbian or synaptic learning. The Hebbian rule states [23]:

V. Erokhin (✉)
CNR-IMEM (National Council of the Researches—Institute of Materials for Electronic and Magnetism), Parco Area delle Scienze 37A, 43124 Parma, Italy
e-mail: victor.erokhin@fis.unipr.it

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_18,

> When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

Considering the electronic circuits, this rule can be considered in the following way. The system must be composed from nonlinear elements connected by a complicated system of wires. The contact points of these wires must increase their conductivity with the frequency and/or duration of their involvement into the formation of signal transfer pathways. In this case learning will mean the formation and the reinforcement of some possible signal pathways and the inhibition of the other ones.

Thus, if our purpose is the realization of a bio-mimicking computational system, we need to have special electronic elements with properties similar to those of synapses.

Even if the properties of neurons and synapses were reproduced with traditional electronic compounds, it seems very perspective to consider another element—memristor, that has recently attracted an explosive attention of numerous research groups. This element was theoretically predicted by Leon Chua in 1971 considering symmetry of electronic [8]. The very important property of the element is the dependence of its resistance on the time integral of the passed current. Later, the term "memristive devices" was introduced [9]. This property is rather similar to that of synapses, described by the Hebbian role: the conductivity depends on the history of the memristor previous functioning. The explosive growth of the activity in the field of memristors started in 2008 after the paper, stating the experimental realization of the memristor [31]. The device was realized from a thin TiO_2 film. The variation of its conductivity was attributed to a drift of oxygen vacancies in the applied external electric field, gradually shifting the relative contributions of the zones with higher and lover conductivities.

The most of the current works in the field of memristors are connected to the metal oxide materials. In this chapter we will not consider these devices. The review of such works can be found in [26].

The aim of this chapter is to describe an organic memristive device that was designed and realized for the artificial reproduction of essential properties of synapse. The device must serve as a key element in circuits allowing Hebbian type of learning [18]. As it was constructed for mimicking synapses, its properties are anisotropic with respect to the applied voltage and the direction of the current flow in a contrary to the "classic" memristors, described by L. Chua and observed in the most of inorganic devices [26] .

2 Architecture and Properties of Organic Memristive Devices

Polyaniline (PANI) is an essential material of the organic memristive devices. Its conductivity can be varied significantly when PANI is in reduced (insulating) and oxidized (conducting) states [25]. Figure 1 illustrates reactions occurring in PANI.

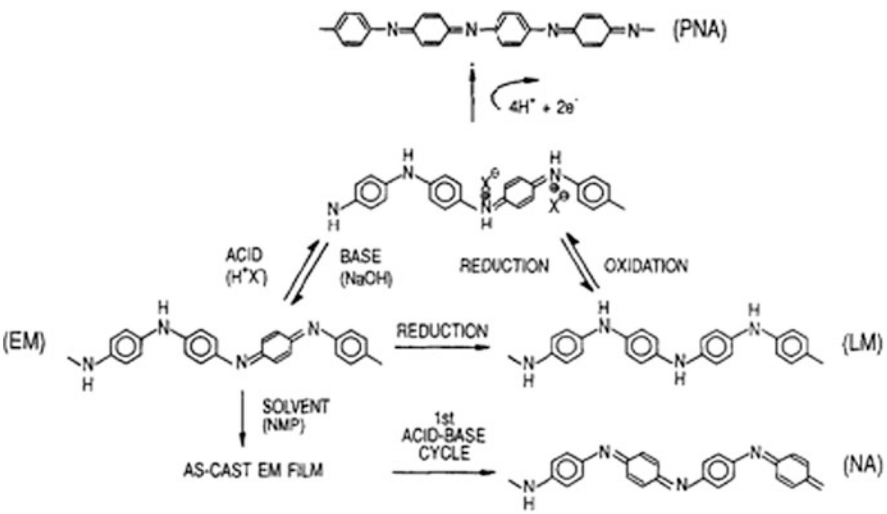


Fig. 1 Interconversions among the various intrinsic oxidation states and protonated/deprotonated states in polyaniline. Reprinted with permission from [25], E.T. Kang et al. in Progr. Polym. Sci. 23:277–324 (1998). Copyright 1998, Elsevier

Emeraldin base form of PANI is an insulator and it becomes conducting (emeraldin salt) after doping (usually by acid treatment). Being doped, PANI varies its conductivity in a reversible way according to the redox state, that can be controlled electrochemically by the application of the external voltage. Right part of the Fig. 1 illustrates these transformations.

The organic memristive devices are prepared in the following way. Thin film of PANI in the emeraldine base form is deposited onto solid insulating substrates with two evaporated metal electrodes by modified Langmuir-Blodgett (LB) technique [13]. LB method allows to form layers with nm resolution what is very important for the organic memristive devices as their working principle implies the diffusion of the metal ions.

The other important component of the device is the medium suitable for the redox reactions. For these reasons, a narrow line of solid electrolyte is deposited in the central part of PANI layer, as it is shown in Fig. 2.

Solid electrolyte line was formed from a polyethylene oxide (PEO) doped with lithium salt. The choice of lithium is determined by the necessity to have its diffusion in a solid state phase. Usually, LiClO4 was used for doping.

Two electrodes, connected to the PANI film are called “source” and “drain”, similarly to the field effect transistor. In order to have a reference potential, a silver wire was connected to the solid electrolyte. This electrode is called “gate” or “reference electrode”. For mimicking synapse properties, the element must have only two terminals. Therefore, the reference electrode is directly connected to the source electrode, that is usually maintained at a ground potential level.

It is possible to distinguish two types of current that give their contribution to the total current passing through the device. We can measure experimentally the ionic

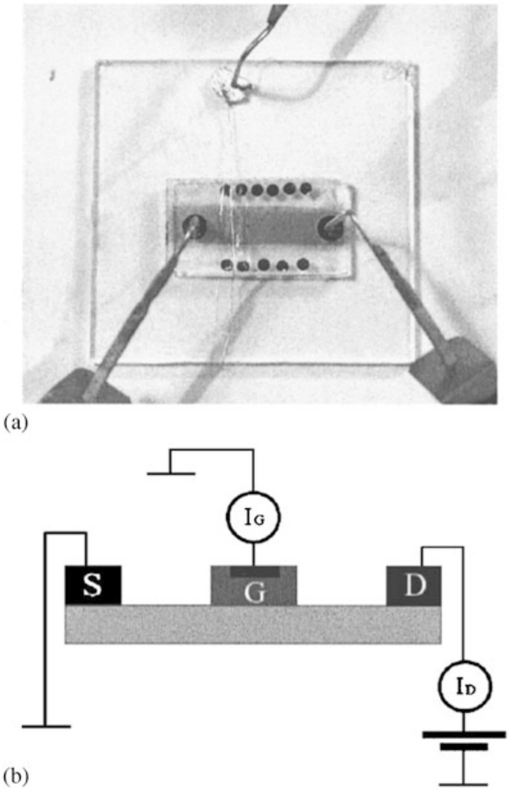


Fig. 2 Photo (**a**) and simplified scheme (**b**) of the organic memristive device. Reprinted with permission from [13], V. Erokhin et al. in J. Appl. Phys. 97:064501 (2005). Copyright 2005. American Institute of Physics

current in the circuit of reference electrode, and the total current, that is a sum of ionic current, mentioned above, and electronic current in PANI active layer, that is measured in the circuit of the drain electrode. However, in order to understand better the device working principle, it is more convenient to consider dependences of ionic and electronic (total minus ionic) currents on the cyclically applied voltage. Cyclic voltage-current characteristics for electronic (a) and ionic (b) currents are shown in Fig. 3. The measurements are usually performed in the following way. Measurements were started at 0 V applied voltage. Then, it is increased with a fixed step in the voltage (usually 0.1 V). After the application of the voltage, the system was equilibrated for fixed time interval (usually—one minute) before readout of the current value. Variation of this time interval results in the variation of the shape of the hysteresis loop [22], as it was also observed for inorganic memristors [26]. Maximum applied voltage must not exceed 1.5 V in order to prevent irreversible overoxidation of PANI.

As it was mentioned above, the measurements start from 0 V. Initial increase of the applied voltage result in the low current values—PANI is in the reduced insulating state. At about +0.5 V we can see a significant increase of the electronic conductivity, what is also accompanied by the appearance of the positive peak in the characteristics for the ionic current. PANI is transferred into oxidized conduct-

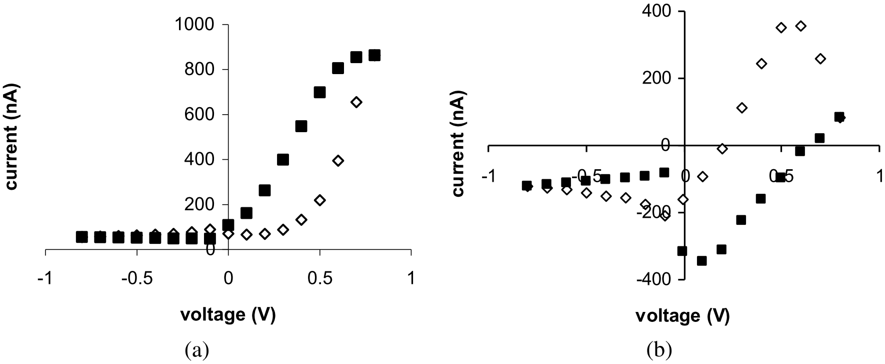


Fig. 3 Cyclic voltage-current characteristics for electronic (**a**) and ionic (**b**) currents of organic memristive device (*empty rhombuses*—increase of the voltage, *filled squares*—decrease of the voltage). Reprinted with permission from [13], V. Erokhin et al. in J. Appl. Phys. 97:064501 (2005). Copyright 2005. American Institute of Physics

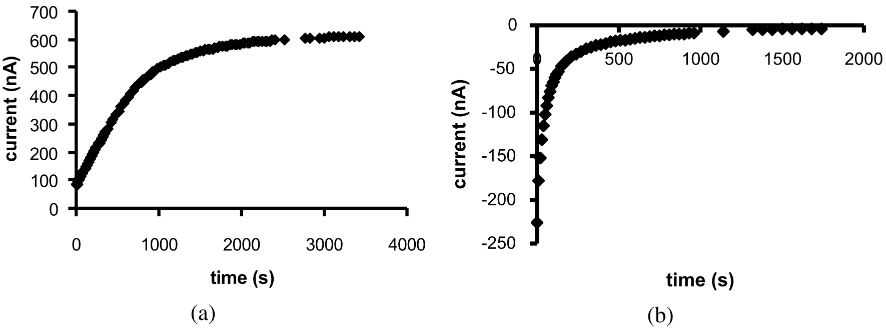


Fig. 4 Temporal variation of the current in organic memristive device at fixed applied voltage of +0.6 V (**a**) and −0.2 V (**b**). Reprinted with permission from [13], V. Erokhin et al. in J. Appl. Phys. 97:064501 (2005). Copyright 2005. American Institute of Physics

ing state. In the most of experiments the max applied positive voltage was +1.2 V in order to avoid the overoxidation, mentioned above. After reaching this value, the voltage was decreased with the same step. The device remains in a conducting state before the applied voltage is diminished till +0.1 V. At this value, PANI is transferred into reduced insulating state, what is also confirmed by the presence of the negative peak in the characteristics for the ionic current. The whole negative branch of the characteristics corresponds to the low conductivity of the device.

However, for the realization of bio-inspired systems capable to learning, it seems even more important the behaviour of the device at a constant bias voltage. These dependences for positive (a) and negative (b) voltages are shown in Fig. 4. In the case of positive bias, the applied potential must be higher than the oxidation one. Usually, it is significantly higher than the oxidizing potential, as the applied voltage is distributed on the whole PANI channel length, while the active zone (PANI-PEO contact) is in the centre of the organic memristor. In the case of the negative bias any potential will result in the diminishing of the conductivity as the reduction potential has a small positive value.

The curve in Fig. 4b makes directly a basis for the possibility to Hebbian type of unsupervised learning. Let us consider the system composed from such devices. Preferential signal pathways will be established by elements, the strength of which will increase with the frequency and/or the duration of their involvement into the signal transfer process. The dependence shown in Fig. 4b is also very important. On the one hand, if the system composed from organic memristive devices will operate at a positive bias for a rather long time, all components of the circuit will reach their saturated conducting state. No learning will be possible anymore. However, if we will provide a periodic short-term application of negative voltage between all input-output pairs of electrodes, we will be able to prevent the system from the saturation. On the other hand, this dependence establishes a basis for the possibility of so-called supervised learning. It implies the external action of a "teacher". In fact, if the system, during unsupervised learning, will establish some connections between inputs and outputs, that are a priori wrong, it will be enough just to apply negative voltage between chosen pairs of electrodes and the signal pathway between them will be suppressed.

Qualitatively, the observed difference in the kinetics for the conductivity variation for positive and negative applied voltages can be explained considering that in the case of the negative bias the whole active zone is under the reduction potential, while in the case on the positive bias only the part of the active zone, closer to the drain electrode, is at a oxidation potential. Thus, in a case of negative potential PANI in the active zone is reduced in the same time, while in a case of positive potential we have a gradual displacement of the conducting zone boundary in the direction from drain to source. Quantitatively, the experimental data were explained by the developed model, calculating temporal behaviour of the potential distribution profiles along the length of the active zone, where PANI is in a contact with solid electrolyte [30].

The conductivity variation in the organic memristive device can be described by the following formula:

$$\mathrm{PANI}^{+} : \mathrm{Cl}^{-} + \mathrm{Li}^{+} + e^{-} \leftrightarrow \mathrm{PANI} + \mathrm{LiCl}$$

When conducting, PANI chain is protonated (positive) what demands the presence of the counter-ion (Cl-) for the maintaining of the electrical neutrality of the molecule. When reduced, lithium enters the PANI and associates with chlorine. Direct detection of the Li ions motion between the PANI active layer and the solid electrolyte was first demonstrated using microRaman spectroscopy [2] and, then, confirmed by X-ray fluorescence measurements using synchrotron radiation for the excitation [3]. These measurements allowed to state that the conductivity of the organic memristive device is a function of the passed ionic charge (time integral of the ionic current).

From this point of view, the suggested device has also some similarities with a memistor—an element introduced by Widrow [32] for construction of adaptive

Fig. 5 Association of the object with the presence of two properties (shape and color)

circuits with memory. According to the definition: Like the transistor, the memistor is a 3-terminal element. *The conductance between two of the terminals is controlled by the time integral of the current in the third, rather than its instantaneous value as in the transistor.*

Finally, our device is also similar to a mnemotrix—an essential element of Valentino Braitenberg mental experiment, developed for the explanation of learning in the brain [7]

> ...we buy a role of special wire, called Mnemotrix, which has the following interesting property: its resistance is at first very high and stays high unless the two components that it connects are at the same time traversed by an electric current. When this happens, the resistance of Mnemotrix decreases and remains low for a while ...

3 Logic Elements with Memory

Brain does not use Boolean logic. The same must be done also for bio-inspired computational systems. The output of logic gates must depend not only on the actual configurations of input signals, but also on the history of their utilization.

For the illustration let us consider AND element with memory (MAND). Other logic gates with memory, such as MOR (OR with memory) and MNOT (NOT with memory) were also realized and their properties can be described similarly to the MAND gate [21]. In the case of living beings, the function of the AND element can be considered as the association of an object with the presence of two important properties. For example: the orange (fruit) can be associated with a color (orange) and a shape (spherical) as it is shown in Fig. 5.

However, for living beings the presence of these two stimuli will not immediately result in such association. The individual must learn that the association is correct. As more frequently these properties are present with the confirmation of correctness by the taste, for example, as the output signal value will be increased from 0 to 1 value. Moreover, if it will happen a wrong association, the value of the output signal will be decreased (if the system is equipped with adequate feedback, similar to taste in nature).

The architecture of the MAND element, based on the organic memristive device is shown in Fig. 6a.

The MAND element contains two inputs, one output and an organic memristive devices as a key basic element, allowing memory. External voltages are used as

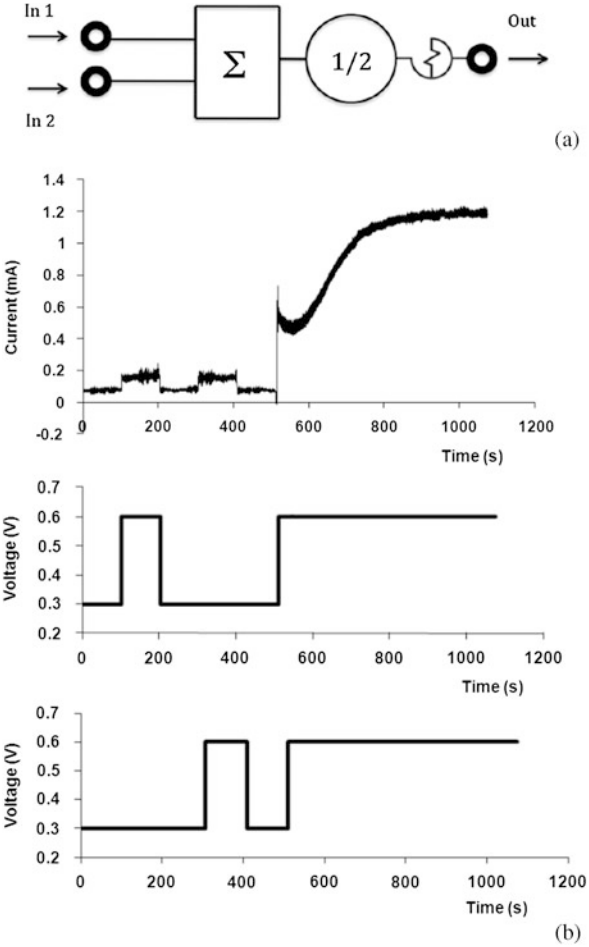


Fig. 6 Scheme of the MAND element (**a**). Temporal dependence of the output current of MAND element (*upper*) and voltages applied to the first and second inputs (**b**). Reprinted with permission from [21] V. Erokhin et al. in Int. J. Bifurc. Chaos 22:1250283 (2012). Copyright 2012, World Scientific Publishing Company

input signals, while the output is a current value. The other two elements in the circuit in Fig. 6a are: a summator, that provides a sum of the applied voltages; and a divider—this element divide the resultant voltage by a factor of 2. Thus, when only one input voltage is applied, after the division it will be not enough to vary the conductivity of the memristive device.

In order to have similar values of input and output signals, comparable also with those applicable to MOR and MNOT elements, the values of input voltages were chosen to be enough for transferring the organic memristive device into the conducting state (+0.6 V).

Experimental results of the variation of MAND conductivity in time together with dependences of the voltages, applied to the first and second input electrodes, are shown in Fig. 6b.

Figure 6b demonstrates that the application of individual signals to the first or the second input does not vary the state of the output current (small linear increase of

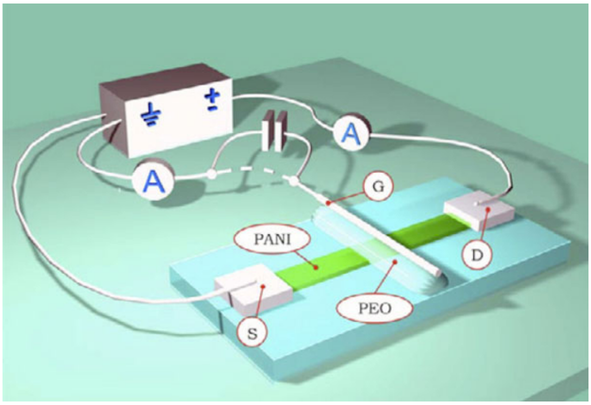


Fig. 7 Scheme of the modified organic memristor device for its use as an oscillator. Reprinted with permission from [15], V. Erokhin et al. in J. Phys. Condens. Matter 19:205111 (2007). Copyright 2007, IOP Publishing Ltd

the value). The application of both input signals results in the gradual increase of the memristive device conductivity. If necessary, the MAND element can be easily reset to the initial sate by the application of a negative potential to any input electrode.

4 Oscillating Element

An essential property of any living being is a capacity to produce rhythmic oscillations of signals even in not variable environmental conditions, as it was stated by E. Schrödinger: “Living matter evades the decay to equilibrium” [27].

Each living system contains in its neural system a neurone (or group of neurones) that, once been activated, produces rather long sequences of spikes.

For the bio-inspired computational systems we also need to foresee such kind of elements. These elements will act, for example, as clock generator analogs, providing a frequency references for all computations.

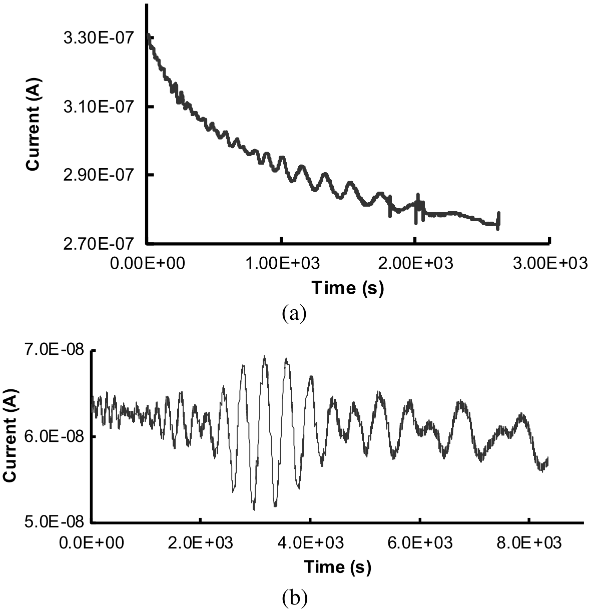
Memristors were considered as perspective candidates for the oscillator realization [24]. It is interesting to note that the organic memristive device can be easily transferred into an oscillator. For this reason it is enough just to insert an element, capable for the charge accumulation, into the circuit of the reference electrode, as it is schematically shown in Fig. 7 [15].

The simplest realization of such configuration can be done by attaching an external capacitor in the circuit of the reference electrode. If one wants to avoid the utilization of the external elements, it is possible to make a reference electrode from the material, capable to the charge accumulation. Experimentally observed temporal dependences of the output current at fixed applied voltages are shown in Fig. 8 [15].

It is to note that the phases of ionic and total currents in the organic memristive devices are shifted in phase.

If one does not want to connect additional external devices to the system, there is the other possibility to have current oscillations. In this case, the reference electrode must be realized from a special material, capable for charge accumulation

Fig. 8 Temporal dependences of the output current at fixed applied voltage for the organic memristive device, when the reference electrode was realized from highly oriented pyrolytic graphite (**a**) and where an external capacitor was connected into the reference electrode circuit (**b**). Reprinted with permission from [15], V. Erokhin et al. in J. Phys. Condens. Matter 19:205111 (2007). Copyright 2007, IOP Publishing Ltd



[16]. In our experiments, we have taken highly oriented pyrolytic graphite, as it is well known its capability to accumulate Li$^+$ ions due to the possibility of their intercalation between the planes of the crystal lattice of this material.

The observed characteristics were explained qualitatively using the already mentioned model, taking into account kinetics of all processes occurring in the device [30]. It is interesting to note that the observed phenomenon cab be also qualitatively explained using the approach, describing Belousov-Zhabotinsky reaction [33], mentioned also in [15].

5 Circuits with Adaptive and Neuromorphic Properties

Supervised [10, 16] and unsupervised learning [28] have been demonstrated in artificial systems by realizing several deterministic circuits, based on organic memristive devices. Even if we present here mainly the results obtained in the DC mode, similar results were obtained when the input signals were performed in a pulse mode [29].

The simplest neuromorphic circuit was realized with one organic memristive device [28], and it has demonstrated the possibility of unsupervised learning. The circuit for supervised learning was composed from 8 memristive devices [16]. Application of the appropriate training procedure resulted in the formation of electrical connections between pre-determined pairs of input-output electrodes. Let us consider now one example of memristive device-based neuromorphic circuits that demonstrates directly the possibility to mimic artificially synapse properties of living be-

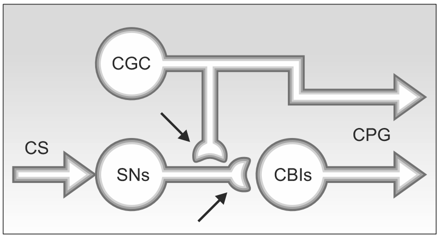


Fig. 9 Model of the part of the nervous system of pond snail Lymnaea stagnalis responsible for learning during feeding. Arrows indicate the position of synapses. Reprinted with permission from [19], V. Erokhin et al. in BioNanoScience 1:24 (2011). Copyright 2011, Springer Science+Business Media, LLC

ings. This example illustrates the electronic reproduction of a part of the nervous system of a simple animal, responsible for its learning.

The model of the part of the nervous system of the Pond snail *Lymnaea stagnalis*, responsible for learning of the animal during its feeding that was developed basing on the experimental data obtained with a system of implanted microelectrodes, was already available [1]. Therefore, it was taken as a biological benchmark for our experiments. Learning in this case means the association of an initially neutral stimulus with the presence of food (similarly to the famous Pavlov's dog; however, the snail is much easy to reproduce artificially (even at the level of the architecture), as the model already exists). In this case of learning of the snail, two stimuli must be applied to the system: initially neutral stimulus (mechanical touching of its lips) and the presence of food (sugar). Touching the lips with the sugar result in the fact that after the successive touching without sugar, the animal begins to open its mouth and start the digesting process.

The scheme of the part of the nervous system, responsible for such learning, is shown in Fig. 9 [19].

As it is clear from the Fig. 9, the architecture of the model is rather simple and it allows the direct reproduction using organic memristive devices in positions of synapses [19].

Therefore, the architecture, presented in Fig. 9 was taken as the starting point for the artificial reproduction with memristive devices. Scheme of the electronic circuit, reproducing the model and results of its experimental testing are shown in Fig. 10.

We have realized and tested two circuits. The first one was based on one organic memristive device, while the other one, similarly to the described model, included two memristive devices. Both circuits have two inputs: one of the inputs corresponds to the initially neutral touching stimulus and the other one corresponds to the stimulus, representing the presence of the food. The system has also one output electrode. If we consider that the learning procedure was successful when the system will be able to perform some execution function (supplying the power to the motor, representing the mouth opening, for example), the value of the output signal must be higher than a certain threshold level (the value of the threshold can be varied and

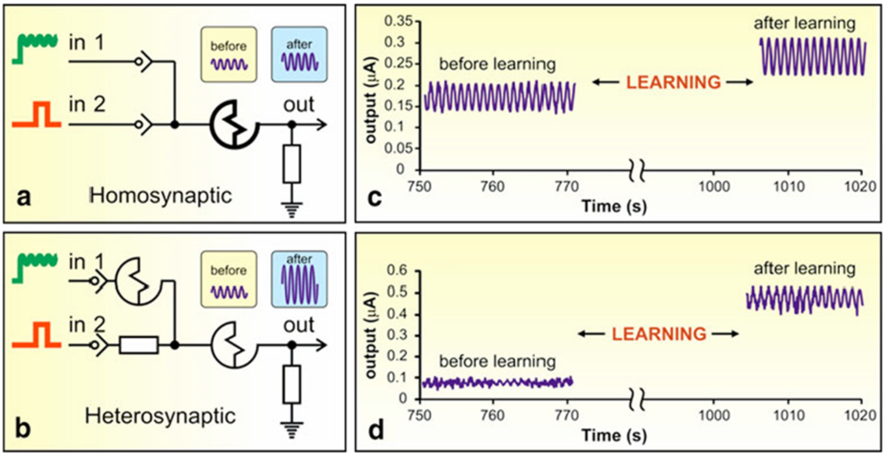


Fig. 10 Scheme of one- (**a**) and two- (**c**) memristor-based circuits, mimicking learning of the pond snail; (**c**) and (**d**) are experimental results, measured on these schemes. Reprinted with permission from [19], V. Erokhin et al. in BioNanoScience 1:24 (2011). Copyright 2011, Springer Science+Business Media, LLC

pre-defined). Thus, if the learning was successful, the mouth of the snail will be opened even in the presence of the touching-mimicking stimulus only.

Let us first consider a one-memristor device circuit shown in Fig. 10a. Touching-mimicking (neutral) signal is applied to the input 1. Initially, it results in the rather low value of the output current, shown in Fig. 10c. The signal, mimicking the presence of the food, is applied to the second input. During the "LEARNING" period (Fig. 10), both signals are applied to the circuit. We have chosen their values in such a way, that only their sum can result in the variation of the memristive device conductivity. After the end of the "LEARNING" phase, we have applied again only one input signal, corresponding to the touching (neutral) stimulus. As it is clear from Fig. 10c, the learning in this configuration was successful. After the simultaneous application of "touching" and "presence of food" stimuli, the resultant current for the successive application of the "touching" stimulus only was increase for about 50 %.

In the second phase of the experiment, we have realized a circuit based on two memristive devices (Fig. 10b) The architecture of this circuit was very similar to that of the model, represented in Fig. 9. Organic memristive devices are exactly in the position of synapses. Input and output signals are rather similar to the previous case, shown in Fig. 10a. However, the values of the input voltages (about +0.6 V) were chosen in such a way, that they are enough to transfer only one organic memristive device into a conducting state. Similarly to the previously described case of the one organic memristive device circuit, we have applied initially only a signal to a input 1, corresponding to a neutral touching-mimicking stimulus. In this case, the applied voltage is distributed between both memristive devices and, therefore, is not enough to transfer any of them into the oxidized conducting state. As a result, we can observe a rather low value of the output current (Fig. 10d). During the "LEARNING" phase, both signals ("touching" and "food presence") are applied. The situation in this case is very different: the input 2 voltage is applied to the one memristive device only. Thus, its value is enough now for the transferring it into a conducting oxidized state. When the transformation was done, the input 1 voltage is mainly distributed onto the second memristive device and can transfer it into the conducting oxidized state. After the finishing of the "LEARNING" phase, the successive application of an input 1 only (neutral "touching-mimicking" stimulus) results in the 5-times increase of the output current (both of DC off-set and AC amplitude), as it is shown in Fig. 10d.

The described experiments have successfully demonstrated that the organic memristive device can be really considered as an artificial electronic analog, mimicking main properties of biological synapses. Reproduction of the architecture of the part of the simple animal nervous system has resulted in the mimicking of learning capabilities at the level of hardware adaptations.

6 Stochastic Fibrillar and Self-assembled Networks

Even if, as it was shown above, the approach, connected to the fabrication of deterministic electrical circuits, based on organic memristive devices, allow to mimic some properties of elements of the nervous system [18], it does not permit high level of the integration of synapse analogs (about 10 in power of 14 in the brain). Therefore, further motion in the direction of mimicking of other simple brain properties will demand the consideration of alternative approaches. In particular, the brain has 3D organization with the existence of connections between rather distant neurones. Current planar technology cannot provide the possibilities for making it. Thus, it is necessary to develop other approaches based on bio-inspired bottom-up technologies, including self-assembling and phase separation.

The first attempt was done by performing the architectures, where the stochastic 3D networks were based on the statistically distributed connections of conducting and ionic elements using polymer fibers [14], forming free-standing networks.

In our case, the attempt to form fibers by electro-spinning turned out to be not very successful: it was possible to form PEO fibers, but not those of PANI. Therefore, the alternative approach was developed, using the capability of PEO to form fibrillar structures in vacuum chamber. Initially, fibers of polyethylene oxide were formed by vacuum treatment of its concentrated viscous solution. Then, these fiber structures were used as templates for the formation of PANI fibrillar systems over it, done by placing the solution of PANI over PEO fibers with the successive vacuum treatment. The optical microscopy image of the resultant structure is shown in Fig. 11.

As we already know from the consideration of the single discrete device, it is necessary to have a junction of the conducting polymer (PANI) and solid electrolyte (PEO) in order to realize the architecture of the organic memristive device.

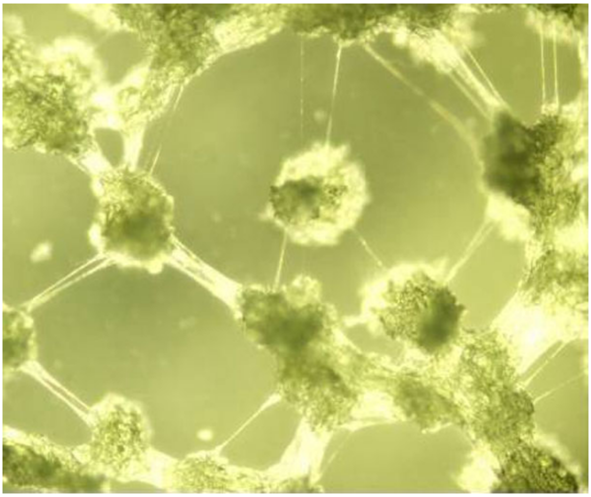


Fig. 11 Optical image of PEO-PANI fiber structure. Reprinted with permission from [14], V. Erokhin et al. in Soft Matter 2:870 (2006). Copyright 2006, The Royal Society of Chemistry

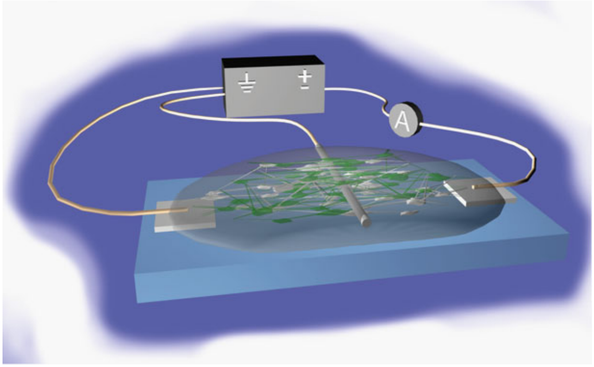


Fig. 12 Organic memristor based on statistically distributed structure of PANI and PEO fibers. Reprinted with permission from [12], V. Erokhin et al. in J. Comput. Theor. Nanosci. 8:313–330 (2011). Copyright 2011, American Scientific Publishers

Therefore, the main idea of the formation of the fibrillar structure was to organize a stochastic crossing between PEO and PANI fibers that form structures, similar to the architecture of the deterministically formed devices. In order to check whether this hypothesis works, the fibrillar structure was deposited between two planar metal electrodes, and a silver wire was placed in it before the vacuum treatment. Scheme of the realized structure and its electrical connection to the power supply and measuring devices is shown in Fig. 12.

Cyclic voltage-current characteristics, measured on these structures in the way, similar to that used for the deterministic stand-along organic memristive devices, revealed rectifying behaviour, that confirm the formation of desired crossing of fibers of two different materials.

Unfortunately, the properties of such structures were found to be very unstable. Only few cycles of voltage-current characteristics turned out to be possible to measure. Then, we have observed a significant decrease of the device conductivity. Finally, in about 40–60 minutes the device stopped working. However, this behaviour

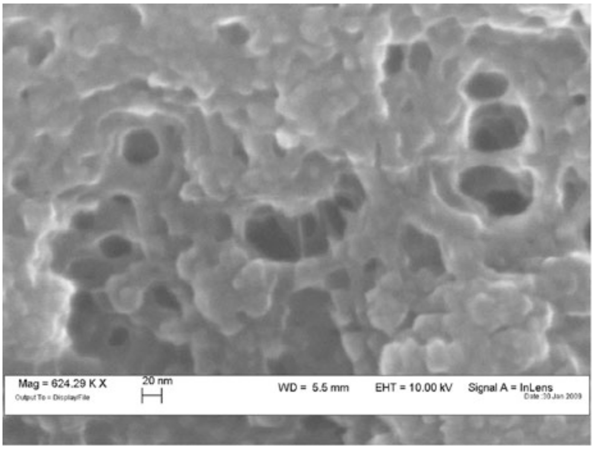


Fig. 13 SEM image of the polymer fibers formed on porous support SEM image of the polymer fibers formed on porous support. Reprinted with permission from [17], V. Erokhin et al. in Nano Commun. Netw. 1:108–117 (2010). Copyright 2010, Elsevier

is not strange for the sample of such type. The degradation of the sample was attributed to the free-standing nature of these structures. Passed current resulted in heating of fibers, their deformation and, finally, complete destruction. Several approaches were carried out for improving the stability. In particular, porous materials were used as supports for making a rigid "skeleton" for these soft structures [17]—the approach that is widely used in the nature. SEM image of the fibrillar structure, formed on porous matrix, is shown in the Fig. 13.

The mentioned approach turned out to be perspective and has allowed to improve the stability of the stochastic systems, based on the fibers of required polymers. However, our efforts were re-distributed because the other approach was found to be even better. This second approach was based on self-assembling of specially synthesized copolymers, allowing phase separation that were then used for the formation PANI layers with associated gold nanoparticle.

The mentioned approach had demanded a synthesis of several specially designed compounds and methods of their assembling into networks, having structure and properties similar to those of the fibrillar systems. In this case the active layer was more complicated with respect to the devices, reported above. Gold nanoparticles were added to the PANI films. The reason of the adding of these particles was to perform a threshold function, somehow similar to the function of the neuron body: to allow the entrance of the signal but to perform a certain barrier for its exit [10]. Such property was realized due to a significant difference of work functions of gold and PANI.

Several types of gold nanoparticles with different terminal-groups used for their stabilization were tested. The best results were obtained when gold nanoparticles were stabilized by 2-mercaptoethanesulfonic acid. The end-group not only stabilizes the structure of the particles, but it acts also as an additional doping agent, stabilizing the conductivity properties of PANI [5]. SEM images of networks of such gold nanoparticles wires are shown in Fig. 14.

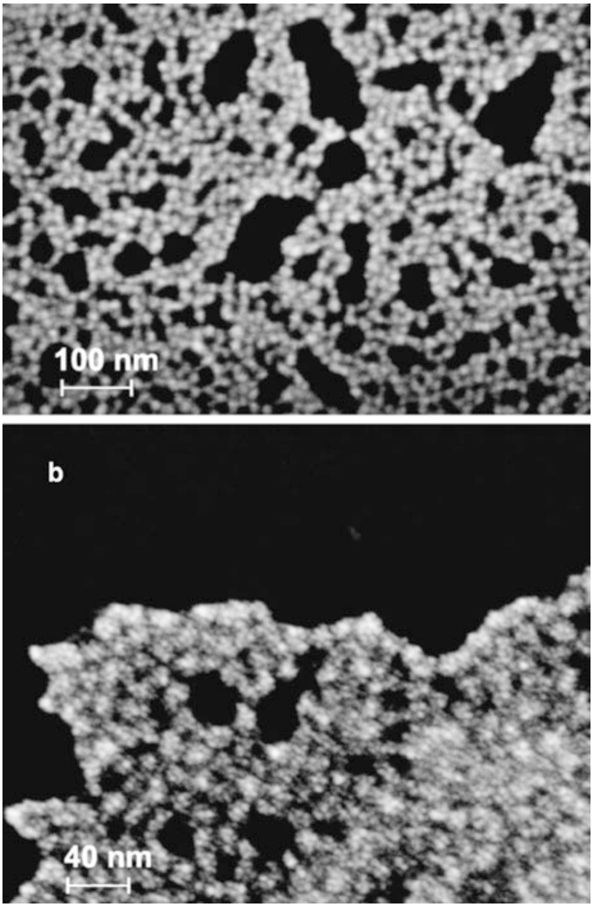


Fig. 14 SEM images of 2-mercaptoethanesulfonic acid stabilized gold nanoparticles. Reprinted with permission from [5], T. Berzina et al. in Synth. Met. 161:1408–1413 (2011). Copyright 2011, Elsevier

Deposition of the PANI/gold nanoparticles composite layers was done using modified Langmuir-Blodgett technique [4]. As it was already mentioned, LB films provide nm resolution in the thickness.

The other essential component of the 3D stochastic network is a block-copolymer, allowing phase separation during self-assembling. In our experiments we have synthetized and used a new block copolymer - poly(styrene sulfonic acid)-b-poly-(ethylene oxide)-b-poly(styrene sulfonic acid) (PSS-b-PEO-b-PSS) [20].

The schematic representation of the experimental sample is shown in Fig. 15a.

Four Cr electrodes were deposited onto a glass (or any other insulating material) support by thermal evaporation. Films of the composite material, containing alternating layers of PANI/gold nanoparticles composite and block copolymer, were deposited on this support and patterned in order to make connections between two pairs of diagonal electrodes in a crossed configuration. A ring made from adhesive Kapton layer (36 microns thick) was placed over the crossed area and PEO gel containing Li^+ and H^+ ions was deposited within the area restricted by the ring. Three

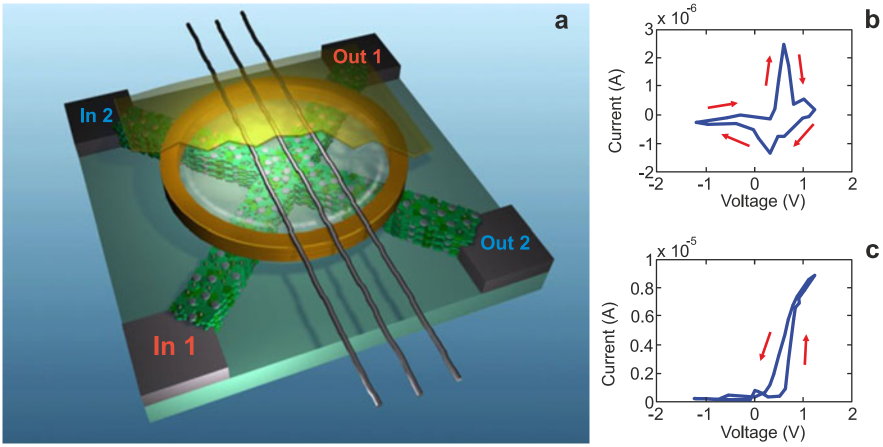


Fig. 15 Scheme of the system used for the learning experiments (**a**) and typical cyclic voltage–current characteristics for ionic (**b**) and electronic (**c**) conductivity measured between each input–output pair. Maximum (at about +0.5 V) and minimum (at about +0.1 V) of the ionic current correspond to the oxidation and reduction potentials of PANI, respectively. As a result, the increase or decrease of electronic conductivity is observed. The presence of hysteresis indicates the memory effect in the system. Reprinted with permission from [20], V. Erokhin et al. in J. Mater. Chem. 22:22881–22887 (2012). Copyright 2012, The Royal Society of Chemistry

silver wires, acting as reference electrodes, were placed in a contact with the PEO gel, whereas the latter in-turn isolated the wires from the active layer. The area was protected by Kapton film in order to prevent the system from the degradation.

The formation of the phase-separated structure was confirmed by optical microscopy and SEM imaging. The images are shown in Figs. 16a and 16b respectively.

Before studying the performance of the network as a whole, we have done the tests whether the conductivity variations between each pairs of input-output electrodes are similar to those, observed for deterministically fabricated organic memristive devices. The typical cyclic voltage-current characteristics for the ionic and electronic currents are shown in Figs. 15b and 15c respectively with the indication of the voltage variation direction.

Such phase-separated architecture of the realized network was expected to provide the formation of multiple possible signal pathways between both pairs of metal electrodes, realized by the stochastic connections of PANI areas with solid electrolyte zones that are also separated by the insulator areas. Conductivity variation will occur in the junctions of the contact of PANI chains with areas of solid electrolyte after the application of the appropriate voltage values. In particular, if PANI in the contact with electrolyte is in the conducting state and a negative potential is applied, it will be transferred into the reduced insulating state. Instead, if it was in the insulating state and a positive potential, higher than the oxidation potential is applied, we will observe a gradual transformation of PANI into the oxidized conducting state.

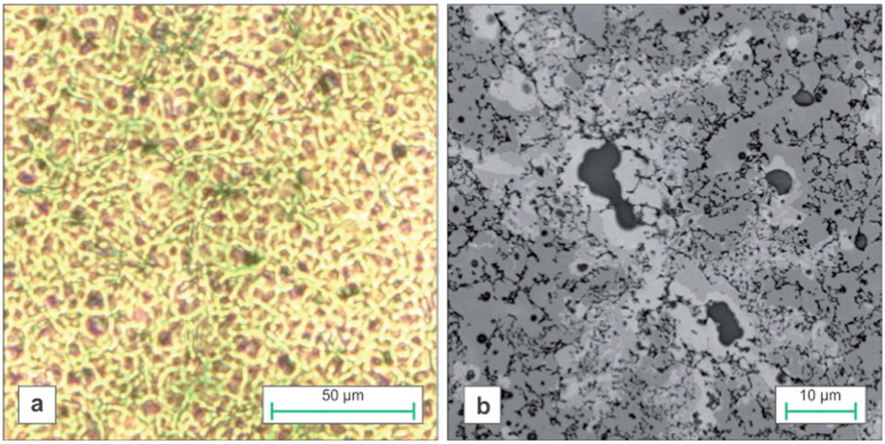


Fig. 16 Optical (**a**) and SEM (**b**) images of the cast film prepared from the composite material. Reprinted with permission from [20], V. Erokhin et al. in J. Mater. Chem. 22:22881–22887 (2012). Copyright 2012, The Royal Society of Chemistry

The aim of the performed experiments was to induce a high conductivity between one pair of input-output electrodes, on the one hand, and to suppress the conductivity between the other pair of electrodes, on the other hand.

We have applied two types of training algorithms to the system shown in Fig. 15a: simultaneous and sequential ones.

In the first case (simultaneous training of both pairs of electrodes), reinforcing and inhibiting voltages were applied simultaneously to the two different pairs of electrodes. After the training, the state of the conductivity between different pairs of electrodes was checked applying a positive voltage with the value that cannot vary the state of the conductivity of the induced signal pathways (+0.3 V). As a result, it was observed that the ratio of the conductivity between pairs of electrodes where the reinforcing and inhibiting potentials were applied was about one order of magnitude. During the second stage of the simultaneous training, the situation was inverted: the reinforcing potential was applied between the pair of electrodes with previously inhibited conductivity and the inhibiting potential was applied between the pair where the conductivity was reinforced. The test measurements, performed similarly to the measurements during the first stage, revealed that the second training was also successful. It was possible to invert the conductivity state between the electrode pairs. The final ratio of the conductivities was also about one order of magnitude.

In the case of the sequential training the procedure was the following one. Initially, the reinforcing voltage was applied between one pair of electrodes, while no voltage was applied between the other one. Then, the inhibiting voltage was applied between the second pair of electrodes, while no voltage was applied to the first one. Test measurements, performed as in the case of simultaneous training, revealed that the conductivity ratio between reinforced and inhibited signal pathways was about two orders of magnitude in this case.

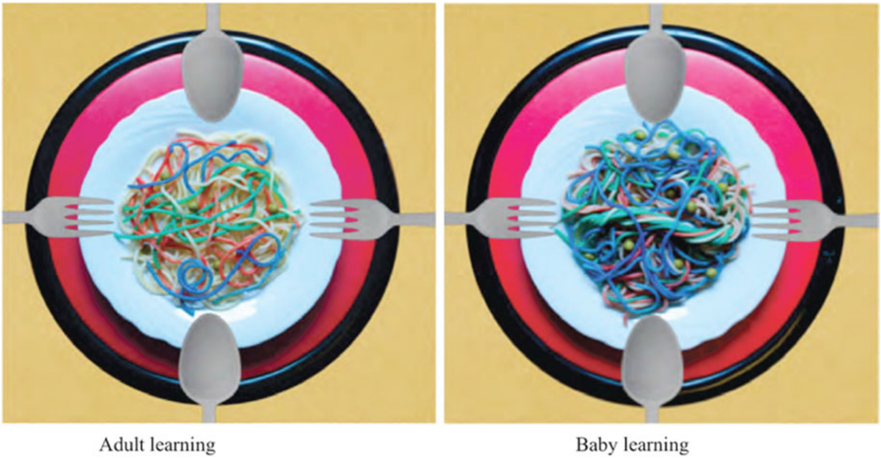


Fig. 17 Schematic representation of the stochastic network after "adult" (*left*) and "baby" (*right*) learning. Reprinted with permission from [11] V. Erokhin et al. in Int. J. Unconv. Comput. 22:1250283 (2013). Copyright 2013, Old Sity Publishing, Inc.

Similarly to the case of simultaneous training experiment, second sequential training was applied to the stochastic system, tending to invert the conductivity state of the already formed signal pathways. However, in this case it was found to be impossible: we have observed only small variation of the conductivity for both channels. Moreover, if the system was leaved without the application of the external potentials, it relaxed itself to the conductivity state reached after the application of the first training.

Making the comparison with the behaviour of living beings, we can conclude that the two observed situations were rather similar to the so-called "adult" and "baby" learning (the last one can be also called—imprinting). In the first case several external stimuli were simultaneously applied to the network. Such algorithm results in the dynamic equilibrium between the formation and the inhibition of signal pathways. There is a "cross-talk" of stochastically distributed memristive devices [6]. In the second case, instead, when single long-term stimulus is applied to a stochastic network, it results in the formation of stable configurations of signal pathways (channels) and potential distribution maps due to the charge trapping in gold nanoparticles. Such stable configuration is acting against successive action of external stimuli and is responsible for the long-term memory. Qualitative explanation of the observed behaviours can be found in [11]. The formation of the connections within the stochastic network was explained by the variation of the "spaghetti" color, representing the areas of PANI in a contact with solid electrolyte. The scheme, illustrating the difference between the states of the system after the "adult" and "baby" learning is shown in Fig. 17.

The reinforcing of the signal pathways in this case was done between two electrodes, shown as forks (only forks are used in Italy for eating spaghetti), while the signal pathways between "spoon" electrodes were inhibited.

"Adult" learning results in the formation of few single preferential pathways between "fork" electrodes, while there is a partial suppression of the conductivity near the "spoon" electrodes. Instead, in the case of "baby" learning (imprinting), stable channels, including multiple signal pathways, are formed between "fork" electrodes.

In other words, the "adult" learning can be compared with the behaviour of foreigners in Italy: looking around they begin to eat spaghetti with forks. However, if their stay was rather short, they will begin to eat spaghetti with spoons when they will come back to their countries. The formed connections within the brain are not stable and can be easily varied in the varied environmental conditions.

In the case of "baby" learning (imprinting) the situation is very different. Italian children in the childhood learn that spaghetti can be eaten with forks only and not with spoons. Application of the training algorithms to the fresh stochastic network without already formed connections result in the formation of stable configurations that are preserved for the practically whole life period and tending to relax to the pre-formed configuration of the connections even if the environmental stimuli are changed.

7 Conclusion

Bio-inspired computing systems must integrate processing and memorizing of the information within the same devices. It demands the utilization of new types of electronic compounds: their electrical properties must depend on the history of their involvement into the formation of signal transfer pathways.

We have considered in this chapter the organic memristive device. This element was designed and constructed for mimicking synapse properties for its successive utilization in circuits allowing Hebbian (or synaptic) type of learning. Some properties of the organic memristive devices, such as hysteresis loop and dependence of the conductivity on the ionic compound of the passed charge, are rather similar to the "classic" memristors. However, it has also some differences, such as anisotropy of the conductivity upon the polarity of the applied voltage. This last feature is very important for mimicking the synapse properties as there is a strong anisotropy of the signal propagation in the nervous system. On the other hand, the described device has properties of the mnemotrix of V. Braitenberg,—key element of the mental experiment, describing learning processes in the nervous system.

Several important characteristics of the device, presented in this chapter, have demonstrated the possibility of the realization of artificial electronic circuits, mimicking some functions of the parts of nervous system of real animals (pond snail). The possibility of the utilization of organic memristive as key elements for the realization of variable logic gates (fuzzy logic) has been also demonstrated. Such logic gates work in such a way that the output signal depends not only on the actual configuration of the input signal values but also on the "history" of its involvement into the information processing.

Just small variations of the device architecture are required for the possibility of the generation of auto-oscillations in the fixed environmental conditions (biased voltage). Such property is a characteristic feature of the living beings and it is very important for all types of the computational systems.

Very interesting properties were registered by the construction and study of the stochastic networks, fabricated by self-assembling of polymeric and nanoparticulate compounds. These features are somehow similar to the learning of the living beings. Application of different training algorithms can result in the long-term (imprinting) associations or in the short-term (every day learning) formation of the preferential signal pathways. In addition, we have observed a "cross-talk" of elements in the network even in the absence of the external stimuli [6, 20]. We can predict at least two possible applications of the deterministic and/or stochastic systems, described in this chapter.

On the one hand, they can constitute a key part of the "brain" of robotic systems: being adequately combined with traditional electronic devices, transferring the signals form sensors (optical, acoustic, mechanical, etc.) to adequate values for application to the network and providing necessary offset and feedback values, it will be possible to attribute "personal features" to each robot for working along or in the group. It will also allow "learning" and adaptations of properties as the reaction to the variable environmental conditions.

On the other hand, the stochastic network can be considered as a very useful tool for making model hardware experiments for the study of processes in the brain. In fact, it will be possible to study the reaction of the system to the variations of the external stimuli at different stages of the system learning. In this respect, it seems important the fact that the variation of the conductivity results also in the variation of color of signal pathways composed from PANI. This property will be used for studying not only the final state of the conductivity between input and output electrodes, but also reinforcement and inhibition of the signal pathways inside the network.

Acknowledgements We acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the Collaborative project PhyChip, grant agreement number 316366.

References

1. Benjamin, P.R., Staras, K., Kemenes, G.: A systems approach to the cellular analysis of associative learning in the pond snail Lymnaea. Learn. Mem. **7**, 124–131 (2000)
2. Berzina, T., Erokhin, V., Fontana, M.P.: Spectroscopic investigation of an electrochemically controlled conducting polymer-solid electrolyte junction. J. Appl. Phys. **101**, 024501 (2007)
3. Berzina, T., Erokhina, S., Camorani, P., Konovalov, O., Erokhin, V., Fontana, M.P.: Electrochemical control of the conductivity in an organic memristor: a time-resolved X-ray fluorescence study of ionic drift as a function of the applied voltage. ACS Appl. Mater. Interfaces **1**, 2115–2118 (2009)
4. Berzina, T., Gorshkov, K., Pucci, A., Ruggeri, G., Erokhin, V.: Langmuir-Schaefer films of a polyaniline—gold nanoparticle composite material for applications in organic memristive devices. RSC Adv. **1**, 1537–1541 (2011)

5. Berzina, T., Pucci, A., Ruggeri, G., Erokhin, V., Fontana, M.P.: Gold nanoparticles—polyaniline composite materials: synthesis, structure and electrical properties. Synth. Met. **161**, 1408–1413 (2011)
6. Berzina, T., Gorshkov, K., Erokhin, V.: Chains of organic memristive devices: cross-talk of elements. AIP Conf. Proc. **1479**, 1888–1891 (2012)
7. Braitenberg, V.: Vehicles. Experiments in Synthetic Psychology. MIT Press, Cambridge (1984)
8. Chua, L.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**, 507–519 (1971)
9. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**, 209–223 (1976)
10. Erokhin, V.: Polymer-based adaptive networks. In: Erokhin, V., Ram, M.K., Yavuz, O. (eds.) The New Frontiers of Organic and Composite Nanotechnologies, pp. 287–353. Elsevier, Oxford (2007)
11. Erokhin, V.: On the learning of stochastic networks of organic memristive devices. Int. J. Unconv. Comput. **9**, 303–310 (2013)
12. Erokhin, V., Fontana, M.P.: Thin film electrochemical memristive systems for bio-inspired computation. J. Comput. Theor. Nanosci. **8**, 313–330 (2011)
13. Erokhin, V., Berzina, T., Fontana, M.P.: Hybrid electronic device based on polyaniline-polyethylenoxide junction. J. Appl. Phys. **97**, 064501 (2005)
14. Erokhin, V., Berzina, T., Camorani, P., Fontana, M.P.: Conducting polymer—solid electrolyte fibrillar composite material for adaptive networks. Soft Matter **2**, 870–874 (2006)
15. Erokhin, V., Berzina, T., Camorani, P., Fontana, M.P.: Non-equilibrium electrical behaviour of polymeric electrochemical junctions. J. Phys. Condens. Matter **19**, 205111 (2007)
16. Erokhin, V., Berzina, T., Fontana, M.P.: Polymeric elements for adaptive networks. Crystallogr. Rep. **52**, 159–166 (2007)
17. Erokhin, V., Berzina, T., Smerieri, A., Camorani, P., Erokhina, S., Fontana, M.P.: Bio-inspired adaptive networks based on organic memristors. Nano Commun. Netw. **1**, 108–117 (2010)
18. Erokhin, V., Schüz, A., Fontana, M.P.: Organic memristor and bio-inspired information processing. Int. J. Unconv. Comput. **6**, 15–32 (2010)
19. Erokhin, V., Berzina, T., Camorani, P., Smerieri, A., Vavoulis, D., Feng, J., Fontana, M.P.: Material memristive device circuits with synaptic plasticity: learning and memory. BioNanoScience **1**, 24–30 (2011)
20. Erokhin, V., Berzina, T., Gorshkov, K., Camorani, P., Pucci, A., Ricci, L., Ruggeri, G., Sigala, R., Schuz, A.: Stochastic hybrid 3D matrix: learning and adaptation of electrical properties. J. Mater. Chem. **22**, 22881–22887 (2012)
21. Erokhin, V., Howard, G.D., Adamatzky, A.: Organic memristor devices for logic elements with memory. Int. J. Bifurc. Chaos **22**, 1250283 (2012)
22. Gorshkov, K., Berzina, T.: On the hysteresis loop of organic memristive device. BioNanoScience **1**, 198–201 (2011)
23. Hebb, D.O.: The Organization of Behavior. A Neurophychological Theory, 2nd edn. Wiley, New York (1961)
24. Itoh, M., Chua, L.O.: Memristor oscillators. Int. J. Bifurc. Chaos **18**, 3183–3206 (2008)
25. Kang, E.T., Neoh, K.G., Tan, K.L.: Polyaniline: a polymer with many interesting intrinsic redox states. Prog. Polym. Sci. **23**, 277–324 (1998)
26. Pershin, Y.V., Di Ventra, M.: Memory effects in complex materials and nanoscale systems. Adv. Phys. **60**, 145–227 (2011)
27. Schrödinger, E.: What Is Life? Physical Aspect of the Living Cell. Cambridge University Press, Cambridge (1944)
28. Smerieri, A., Berzina, T., Erokhin, V., Fontana, M.P.: A functional polymeric material based on hybrid electrochemically controlled junctions. Mater. Sci. Eng. C **28**, 18–22 (2008)
29. Smerieri, A., Berzina, T., Erokhin, V., Fontana, M.P.: Polymeric electrochemical element for adaptive networks: pulse mode. J. Appl. Phys. **104**, 114513 (2008)
30. Smerieri, A., Erokhin, V., Fontana, M.P.: Origin of current oscillations in a polymeric electrochemically controlled element. J. Appl. Phys. **103**, 094517 (2008)

31. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
32. Widrow, Pierce, W.H., Angell, J.B.: Birth, life, and death in microelectronic systems. Office of Naval Research Technical Report 1552-2/1851-1, 30 May 1961
33. Zaikin, A.N., Zhabotinsky, A.M.: Concentration wave propagation in two-dimensional liquid-phase self-oscillating system. Nature **225**, 535–537 (1970)

Memristive in Situ Computing

Omid Kavehei, Efstratios Skafidas, and Kamran Eshraghian

Abstract The missing link between a nonlinear circuit element that is able to self-adjust its conductance according to the history of applied voltage/current and physical realizations of two-terminal oxide-based resistive memory was discovered in early 2008, and has since been intensively studied. This class of memory elements is called memristive devices, which includes resistive random access memories (RRAM), phase change memories (PCM) and spin-transfer torque magnetoresistive memories (STT-MRAM). Memristive devices are mostly CMOS and fab friendly, and promise simpler architecture, higher scalability and stackability (3D), good selectivity, relatively low-power consumption, high endurance and retention, fast operation by utilizing parallelism, and the most important of all, the ability to merge logic and memory. A significantly wide range of resistive switching materials can be categorized under three main redox-related effects, electrochemical metalization effects (ECM), valency change memory effect (VCM) and thermochemical memory effects (TCM). Although, the behavior of these resistive memories can be modeled using high-level finite-state machines (FSMs), the underlying switching mechanisms is yet to be fully understood. Despite the lack of comprehensive understanding of the switching behavior, their application in memory and computing has been constantly improved. These devices can be programmed to exhibit multi-level cell (MLC) and binary cell behavior, thus analog and digital memories can be exists in one device depends on programming. In this chapter, we highlight some of the in situ computational capability of memristive devices.

O. Kavehei (✉) · E. Skafidas
Centre for Neural Engineering, Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Australia
e-mail: omid.kavehei@unimelb.edu.au

E. Skafidas
e-mail: sskaf@unimelb.edu.au

E. Skafidas
National ICT Australia (NICTA), Victoria Research Laboratory, Melbourne, VIC 3010, Australia

K. Eshraghian
iDataMap Corporation, Perth, Australia
e-mail: keshraghian@idatamap.com

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_19,

1 Introduction

The connection between an existing mathematical foundation for memristive devices, described in [7], and a nanoscale memory system, reported in [41], has created a lot of attention and accelerated activities toward the next disruptive innovation in computer architecture and computing [50]. The memristive devices category includes resistive switches as long as they meet the "memristive devices and systems" criteria that is outlined in [8]. A voltage-controlled memristive device with only one state variable (memristor), w, can be described using the following system of equations,

$$\begin{cases} I = G(h, V) \cdot V \\ \frac{dh}{dt} = f(h, V), \end{cases} \tag{1}$$

where dh/dt describes ionic motion, or filament elongation rate, using function $f(\cdot)$, for ECM (cation) and VCM (anion) systems. In both systems, h is bounded to two extreme values $0 < h < L$, where the simplest definition for L is the thickness of resistive switching material, or the active layer, namely between 3 nm and 30 nm. The $G(\cdot)$ function represents the device conductance, and I and V are current and voltage, respectively. For simplicity, the state variable, h, can be normalized, $0 < x(= h/L) < 1$.

Material system of ECM and VCM devices consists of an anode electrode that supports oxidation, a cathode electrode that supports reduction, and a resistive switching material. The conductive electrodes are used to facilitate reduction and oxidation (redox). Recent study in [44] shows that ideal memristor is still a hypothetical element and, interestingly, STT-MRAMs have the closest behavior to an ideal memristor. There are, therefore, some doubts about the HP's "ideal" memristor [41] from the electrochemistry point of view. The currently available memristive devices can be considered as 'voltaic cells' that, under equilibrium, derives electrical energy from spontaneous reduction and oxidation reaction taking place within the cell.

The deviation of ECM and VCM behavior from the criteria specifying memristive devices and systems, is caused by chemical potential gradients generating an electromotive force (emf), hence violating the current-voltage zero-crossing requirement of memristive systems. This creates non-equilibrium high conductance and low conductance states that may affect device operation and retention of resistances.

An extended modeling approach is required to compensate for such non-ideality. In [44], the device was considered as a memristive element with two state variables, h and c_{ion}, where c_{ion} is the ion concentration. The emf depends on the ion concentration and the current is a linear sum of the electronic current and the ionic current. Therefore, the nanobattery effects should be taken into account during the simulation. Our experiments show that such emf contribution is around $V_{\text{cell}} = 5$ mV, thus negligible, in a Ag/TiO$_x$/ITO system [17], and a more than 20 mV in a Pt/TiO$_x$/Pt system, as shown in Fig. 1b. All experiments are carried out using a Keithley Semiconductor Characterization System (4200-SCS).

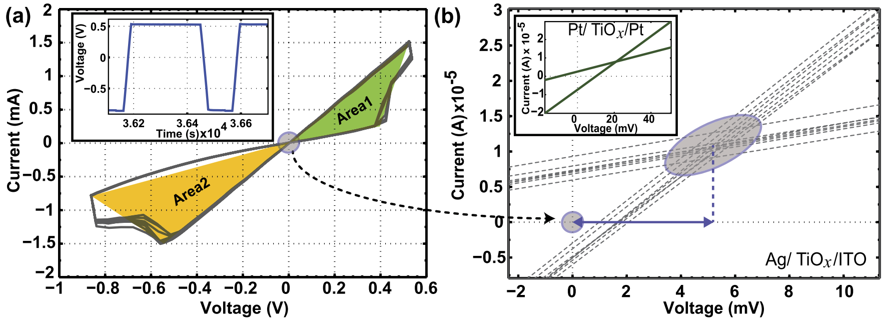


Fig. 1 (**a**) Experimental result of Ag/TiO$_x$/ITO memristive system using a pulse measurement unit instrument with asymmetric positive and negative voltage magnitudes and pulse-widths (inset). Each loop shows 25 experiments, electrode area is 1×10^{-4} cm^2 and rise/fall sweep rate is 46 $\mathrm{m\,V\,s^{-1}}$). (**b**) Shows the non-zero crossing of Ag/TiO$_x$/ITO (main) and Pt/TiO$_x$/Pt systems (inset). In a simplest form, Area1 and Area2 are highlighted as represent the electric power for the memristive device under test, $P = IV$

Given the definition and the characteristics of memristive devices, this chapter discusses potentials of these devices in computing and highlights their in situ computing capability. The aim is to highlight their alternative solution for beyond von Neumann model, such as cognitive computing. The classical von Neumann machine suffers from a large sequential processing load due to the existence of the separation between memory and logic [2]. In contrast, cognitive computing introduces a more efficient implementation but not necessarily low-power. Software techniques are also power hungry. A very first application of resistive memories for cognitive computing was proposed by S.R. Ovshinsky base on ovonic memories (chalcogenide memory materials), which are better known as phase change memories (PCM) [31]. Ovonic threshold switches are fast and relatively small [11, 30], however, resistance drift is a big problem for them to behave as a nonvolatile analog memory. This decay can also be seen, in a much limited form, in ECM and VCM systems [9, 15, 39].

It has been experimentally proven—in small scale—that memristive devices are able to carry out logic operations [4], hence a possible option for implementing a tighter coupling between memory and logic. Owing to the multi-stable state memory property and the relatively long term decay, they are also able to encode synaptic-like weights [14]. Other possibilities for cognitive computing domain have been reported in [33, 38, 39, 53].

1.1 Uncertainty Mitigation for Cycle-to-Cycle Switching

One of the pronounced differences between CMOS and resistive switching technologies is the cycle-to-cycle variation; something that was not a significant issue in CMOS technology. The amount of cycle-to-cycle resistance variation has been

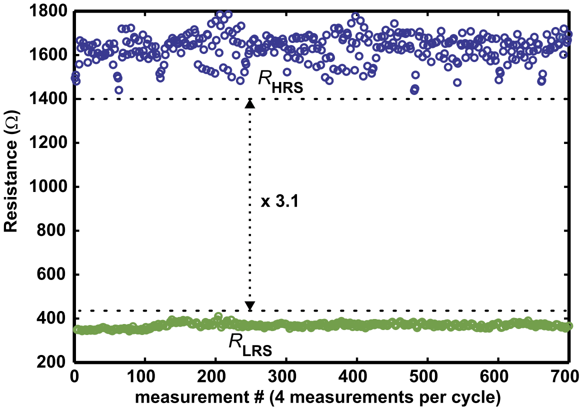


Fig. 2 Resistance measurements between 50 mV and 200 mV of the current-voltage curve in Fig. 1a. Four measurement points per cycle (two measurements per resistance state per cycle). Resistance ratio demonstrates a $R_{\mathrm{HRS}}/R_{\mathrm{LRS}} \approx 3$

constantly reported as significant [22]. Here we suggest that device (thin-film) characterization techniques can be used to mitigate such variation.

An interesting result of an appropriate device characterization is to determine voltage (or current) magnitudes as well as applied pulse-widths that result less variation in low and high state resistances (R_{LRS} and R_{HRS}) values. The result in this case is an asymmetric voltage pulse that is shown in Fig. 1a (inset) with −0.85 V and 0.53 V. A four-point resistance measurement between 50 mV and 200 mV per current-voltage loop (each loop represents a cycle) is shown in Fig. 2. As expected from the theory, the device behavior can be controlled using combination of voltage and time.

In this case, with this assumption that the barrier wall (filament tip) displacement for SET and RESET switching are the similar and the R_{LRS} is not hugely smaller than R_{HRS} (as shown in Fig. 2), one could argue that in the both scenarios an equal amount of 'work' should be put into the system to minimize the uncertainty. We adjusted the positive and negative pulse-widths (in Fig. 1a, inset) to compensate for the difference between Area1 and Area2 (in Fig. 1a), and balance the work for positive ($W_{\mathrm{pos}} = \mathrm{Area1} \times t_{\mathrm{pos,pulse}}$) and negative pulses ($W_{\mathrm{neg}} = \mathrm{Area2} \times t_{\mathrm{neg,pulse}}$). According to our characterization, the ratio of the positive pulse-width over the negative pulse-width should be in the vicinity of 2.3, which is about Area2/Area1.

2 Device Dynamics

To measure the function $f(\cdot)$ in Eq. (1), several time-domain experiments for I and V are required. According to our measurements, a sinh(·) like behavior explains dynamics of the device. The sinh(·) term defines the dependency of velocity, dh/dt, as a function of the effective applied electric field that has been described as an ionic crystal behavior in the presence of an external field [26, 27]. An additional term highlights the dependency of the current conductance, G_t, to the previous conductance state, G_{t-1}. This describes the relationship of long-term potentiation (LTP)

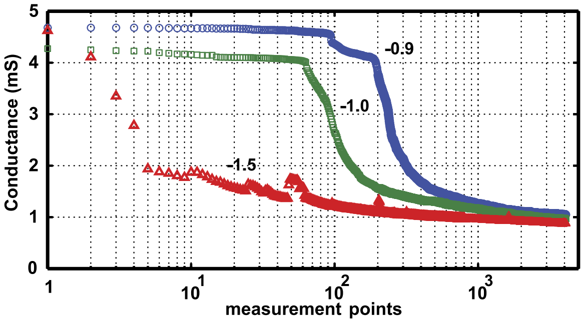


Fig. 3 Change in the conductance of a memristive device under three different DC voltages. We perform 4096 points measurements over 212 s

and long-term depression (LTD) in memristive devices as a function of their initial states.

The LTP and LTD are strong functions of the device's asymmetrical characteristics. Bipolar device's (e.g. ECM systems) current-voltage characteristic usually shows a certain asymmetry in the absolute value of the threshold voltages and filament growth/reduction rate (ΔG). This asymmetry can be engineered using different electrodes and/or during electroforming process [13, 51]. We perform 4096 points measurement over an execution time of 212 seconds, using a source-measurement unit instrument, at three different dc voltages to highlight the voltage-switching time relationship in the memristive systems.

We use a nonlinear form function, $y(h)$, to define dh/dt as a function of h based on Fig. 3 in [35]. The $y(h)$ function then should be multiplied by the $\sinh(\cdot)$. The conductance behavior as a function of h is also shown in Fig. 2 of [17]. Due to the asymmetric behavior of $h \to 0$ and $h \to L$ [35], we have used two different $y(h)$ to achieve a more accurate modeling [17, 18, 35]. The state variable equation then can be defined as

$$\frac{dh}{dt} = y(h)\sinh(\upsilon V) + D(h), \tag{2}$$

where υ is voltage coefficient and the function $D(h)$ represents the decay term. The decay term has a negative exponential form with a relatively long decay [6, 29]. The first term of Eq. (2), represents a voltage dependent and highly nonlinear part, which makes high-speed digital computing possible. This property originated from the fact that resistance modulation inside the metal-oxide occurs via electron-ion interactions. This term would create some problems for learning applications.

The nonlinear behavior produces a threshold-like region that voltages below that a certain point does not change the conductance significantly. Considering the fact that, device's conductance, G, can be tuned by a series of voltage pulses with appropriate pulse-width (pulse-rate modulation or pulse-width modulation). Applying a voltage around the threshold slightly changes h depends on the pulse-width (normally a few μs for low voltages). It is observed that such voltage cannot change the state if the duration is in the orders of ns. However, slightly increase in the applied voltage increases the speed by orders of magnitude, which makes nanosecond (digital) switching possible. Therefore, a series of few µs pulses with an appropriate pulse shape and voltage magnitude can be used to implement an analog memory device that is able to mimic individual ion channel of a synapse.

The analog memory behavior can be engineered either by applying appropriate voltage/current signals or device fabrication for specific type of desired behavior. Unfortunately, there is yet no systematic approach to manage and monitor the quality and performance of devices. Therefore, there is an international race to modularize, parameterize and ultimately uniformize of fabrication processes, modeling approaches and switching mechanisms.

3 Analog in Situ Computing

3.1 Muti-stable State

Using a source measurement unit instrument, another experiment was performed on the Ag/TiO$_x$ /ITO memristive system under 0.5 V to −0.5 V sweeps as shown in Fig. 4b. We intentionally removed the voltage sweep from −0.5 V to 0.5 V to create a net movement toward higher conductances. Repeating the experiments for more than 5000 measurement points causes a significant reduction in resistance. The resistance in Fig. 4b is calculated for all V and I points. The solid red line shows the slop of reduction in resistance. Figure 4a demonstrates the current-voltage curves of the system.

We consider that changes in the concentration gradient of majority charge carriers (electrons and anions/cations) are responsible for the switching phenomenon, having a perfect switching characteristics would require a perfect charge balance. In the absence of such balance, as a direct consequence of the discussion in Sect. 1.1, applying a positive and negative voltages for the same amount of time, to the device under test, causes a net movement toward either OFF state (R_{HRS}) or ON state (R_{LRS}). In this case, initial state of the device is close to R_{HRS}, therefore, if time of the positive voltage ramp, $t_{\mathrm{pos,ramp}}$, and time of the negative voltage ramp, $t_{\mathrm{neg,ramp}}$, are equal, $W_{\mathrm{neg}} < W_{\mathrm{pos}}$, because $I_{(V=-0.5)} < I_{(V=0.5)}$, and we observe a well controlled net movement toward higher conductance.

Considering Fig. 1a, other ways to achieve a controlled analog memory behavior apart from pulse-width and pulse-frequency modulation [19, 20], is to either increase the sweep rate of the applied triangular voltage or apply a current compliance with a maximum allowable current flow, which is much less than the required current for RESET (switch to R_{HRS}) or SET (switch to R_{LRS}) process. In circuit design terms, a current limiter can be an accurately biased series transistor.

3.2 Plasticity and Learning

Spike timing-dependent plasticity (STDP) is a form of Hebbian learning in which plasticity is induced by temporal correlations between pre- and post-synaptic spik-

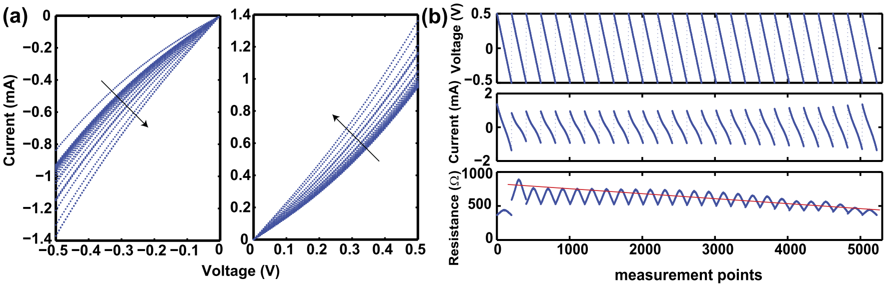


Fig. 4 Memristive analog behavior in Ag/TiO$_x$/ITO system. The experiments were performed at room temperature in dark. (**a**) sinh(·)-like current-voltage curves. (**b**) voltage, current and resistance of the device for more than 5000 measurement points

ing neurons [5, 10, 40]. Connections can be drawn between memristive devices and biological synaptic update rule, such as STDP [12, 14, 16, 24]. This can be achieved by collecting data from a memristive device based on the arrival time difference, Δt, between pre- and post-synaptic spikes. Mathematically, if $\Delta t = t_j^{\text{post}} - t_j^{\text{pre}}$ then the STDP weight modification rule can be defined as a linear summation of exponential functions,

$$w_{ij} = u(-\Delta t) A_+ e^{-\Delta t/\tau_+} - u(\Delta t) A_- e^{\Delta t/\tau_-}, \tag{3}$$

where $u(z) = 1$ if $z < 0$ and $u(z) = 0$ if $z > 0$, and A_+, A_-, τ_+, and τ_- are fitting parameters to model the experimentally determined STDP points. This can be interpreted as an specific form of exponential expression of Eq. (2). It is also shown that BCM (Bienenstock, Cooper, and Munro) [3] learning rule follows directly from STDP when uncorrelated or weakly correlated Poisson spike trains [10] are used. Therefore, it is expected that both STDP and SRDP (spike rate-dependent plasticity) can be implemented with memristive devices [23].

The results are shown in Fig. 5a, which shows how the device under test weight (resistance) changes as a function of Δt. The intermediate states vanish after a certain decay duration whereas a significantly higher potentiation ($x \to 1$) will be kept as a long term memory [6, 23, 29]. The results in Fig. 5, partially highlight that the memristive network follows the competitive learning behavior, reported in [40].

The collected information is then used as stimuli for a network of 1×1000 memristive devices are connected to one neuron being implemented and pre- and post-synaptic spikes shape is the same as [53], then this network implements the competitive Hebbian learning [40]. Initial states have been shown in Fig. 5c in red. Intentionally, a Gaussian distribution has been employed for the memristors' initial state values. After running the simulation for 35 minutes, the network results in a population distribution of weights similar to a previously published competitive Hebbian learning [40]. The additive, multiplicative and log (log-STDP) features of a memristive network strictly depends on the device and its nonlinearity parameters. Figure 5b demonstrates a Poissonian ISI distribution. There are several relatively large scale implementation of cognitive computing structures based on nanoelectronics, which can be found in [42, 52].

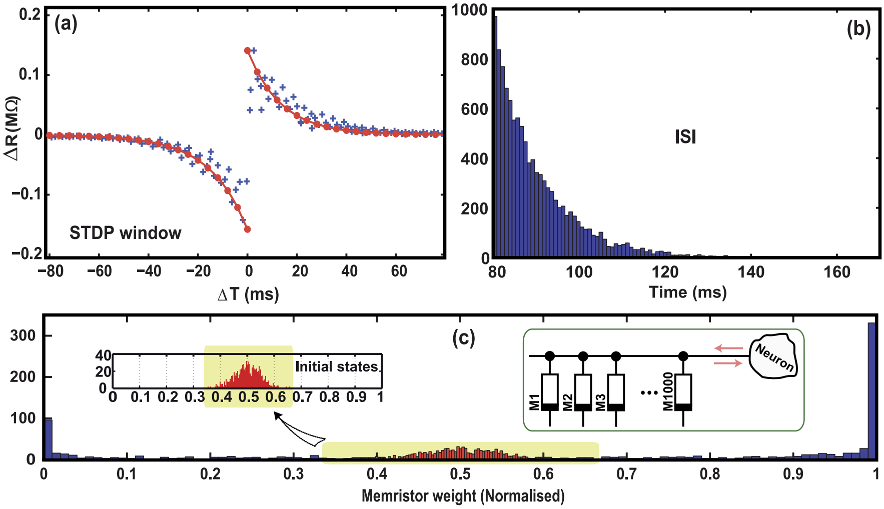


Fig. 5 Memristor, plasticity, and competitive learning. (**a**) *Dots* illustrate experimental data and the *solid* (*red*) *line* shows the fitting STDP rule. We exclude devices that reach lowest and highest conductances, because they add no extra information to our analysis. (**b**) A Poisson distributed inter-spike interval (ISI) for 1 × 1000 devices (synapse) connected to one neuron, inset in (**c**). (**c**) Illustrates simulation results of such network

It is important to note that unless appropriate design parameters, such as total required current for device programming, leakage currents, resistance decay, overhead programming and sensing circuitry, etc are taken into account, a hybrid memristive/CMOS cognitive computing system does not necessarily make an energy efficient system.

It worth to emphasize that the above mentioned evidence [44] regarding fundamental facts of electrochemistry of the memristive devices, allows a different point of view toward memristive devices. It indicates that it is plausible to expect that such devices are capable of mimicking galvanic effects in biology or vice versa.

3.3 Programmable Analog Circuits

Although plasticity plays an important role for adaptation and development, networks with fixed synaptic weight pattern should be studied as well. One of the challenges for the memristive emerging technologies is to integrate learning and unlearning hardware as part of a neural computational platform. Since memristive devices possess a threshold-like behavior, usually very low-voltage operations do not change their initial state. This fact helps developing programmable analog computing circuits [1, 28, 32].

Here, we introduce the use of a memristive array for implementing a multiplication of inputs and their internal state, h, which is a representation of the device

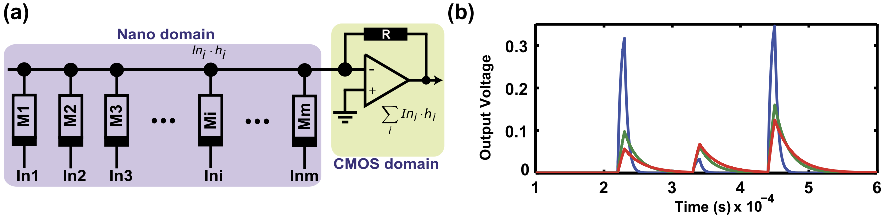


Fig. 6 Multiply-accumulate module. (**a**) Shows a single row of multiply elements (memristors), $\text{In}_i \cdot w_i$. (**b**) Demonstrates simulation results for two memristors, M1 and M2. In this simulation, memristor M2 programmed at $x = 0.5$, which is equivalent to $(R_{\text{HRS}} + R_{\text{LRS}})/2$. Then memristor M1 changes its resistance from R_{LRS} to R_{HRS} in three steps. Each step is a simulation that is shown with different colours. *Blue* for $R_{\text{M1}} = R_{\text{LRS}}$, *green* for R_{M1} close to $(R_{\text{HRS}} + R_{\text{LRS}})/2$, and *red* for R_{M1} close to R_{HRS}. The summing amplifier can be replaced by any threshold module for different applications

conductance. Figure 6a illustrates a single row of the array and Fig. 6b shows its simulation results for two elements, M1 and M2, connected to two inputs, In_1 and In_2. In this case, we first applied a voltage pulse to M1 to read its conductance, then a pulse to M2 for the same reason. When two voltage pulses are simultaneously applied to M1 and M2, accumulation operation can be clearly observed.

4 Digital in Situ Computing

The existence of the sharp switching threshold, functional uniformity, intermediate state initialization, and most importantly state decay creates several problems that can be eliminated or compensated for by using the device as a binary switch [34, 45]. Other than the possibilities that the switch creates for cognitive computing, such as stochastic learning based on binary weighted synapses, implementing Boolean operations are also very interesting part of the current mainstream research on ‘More than Moore’ technologies [46].

Memristive devices have shown in situ computing in a digital form as well. While such capability is described in [4], here our focus is on a novel type of memristive devices, complementary resistive switch (CRS).

4.1 Complementary Resistive Switch: Diodeless Nanocrossbars

Although, memristive devices have introduced new possibilities and they are well adapted in the crossbar architecture, the inherent parasitic current paths (sneak current paths) between neighboring cells of an addressed cell impose limitations on the array scalability [18, 21, 25]. A possible solution is to build a diode or a transistor in series with the device. Therefore, more current requirements for the RESET

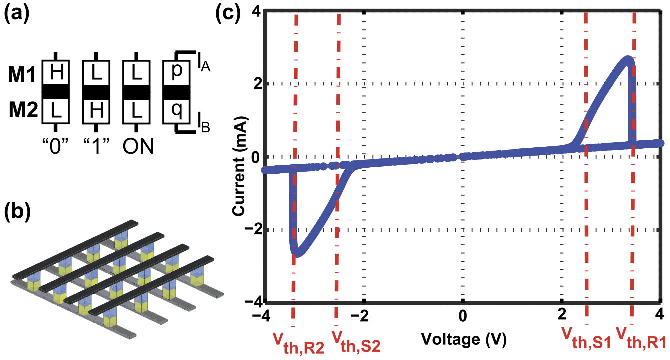


Fig. 7 CRS device structure and logical definition of each combination. (**a**) demonstrates all operational states, (**b**) illustrates the crossbar view, and (**c**) shows CRS functionality

and SET process means larger series diode or transistor. Using transistors also adds other scalability issues due to the fact that transistors are not very well stackable and the application of diodes imposes a high drive current limiting the use of such array in an ultra-low-power applications.

Linn et al. [25] introduced a new paradigm by exploiting two anti-serially connected bipolar memristive devices. The structure is similar to a *memistor* (note the missing "r") with a control gate as the middle electrode [43, 48, 49]. A (digital) CRS uses a combination of a HRS (H) and a LRS (L) to encode logic "0" and logic "1". Consequently, the overall resistance of such device is always around HRS, resulting in significant reduction in the parasitic current paths through neighboring devices. Figure 7a summaries the CRS states. If p and q indicate resistances of the memristors M1 and M2, respectively, four different states can be observed. For example, $p/q \leftarrow \text{L/H}$ indicates that LRS is written in p (device M1) and HRS in q (device M2). Combinations L/H and H/L for p and q represent logic "1" and logic "0", respectively. Note that the H/H state only occurs once in a fresh device. According to Fig. 7c any transition between the states occurs if the applied voltage exceed the SET thresholds, $V_{\text{th,S1}}$ or $V_{\text{th,S2}}$ and the device's initial state supports the transition. Possible state transitions are shown in Table 1, where p'/q' shows the next state, p/q illustrates the initial state, and output is a current pulse or spike. These outputs enable us to have two different read-out mechanisms, logic → ON or logic → logic.

The transitions in Table 1 can be defined using *material implication* logic [4, 37]. It has been proven that the implication operation and FALSE operation are a complete set for logical operations [47]. This logical operation results in change of q depending on the state of p (or vise versa), known as p IMP q, 'p implies q' or 'if p then q'. Therefore, if we represent p NIMP q it means 'p not implies q', Table 1 (i), for example, represents $q \leftarrow \text{H}$ and we say the conditions (initial p/q and ΔV) not implies q.

Table 1 State transitions in a CRS

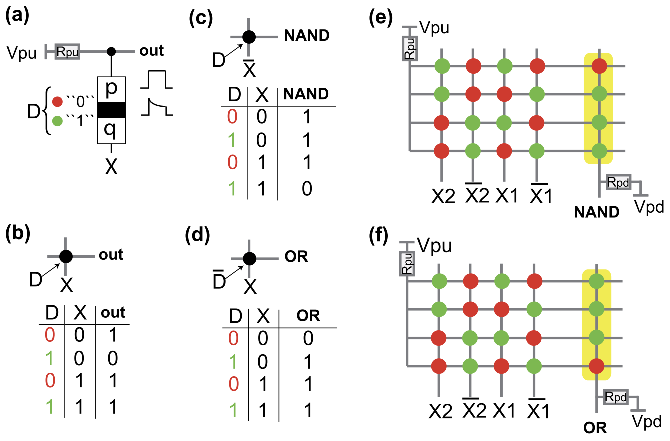
	p/q	$\Delta V = V_{\mathrm{I_A}} - V_{\mathrm{I_B}}$	p'/q'	Output
(i)	"1"	$V_{\mathrm{th,S1}} < \Delta V < V_{\mathrm{th,R1}}$	ON	Pulse
(ii)	"1"	$V_{\mathrm{th,R1}} < \Delta V$	"0"	Spike
iii)	"0"	$V_{\mathrm{th,R2}} < \Delta V < V_{\mathrm{th,S2}}$	ON	Pulse
(iv)	"0"	$\Delta V < V_{\mathrm{th,R2}}$	"1"	Spike
(v)	ON	$V_{\mathrm{th,R1}} < \Delta V$	"0"	–
(vi)	ON	$\Delta V < V_{\mathrm{th,R2}}$	"1"	–

4.2 CRS-Based Boolean Operations

In [16], we introduce CRS-based logical operation and PLA (programmable logic array) that works with the two transitions, logic → logic and logic → ON, but we only present it with the later transition. The idea is to charge a bit-line in a crossbar array, and applying inputs to its word lines. The inherent implication property of the device causes a change under certain conditions that we have already discussed. In [37], AND and NOR operations are proposed using the logic → logic transition and current spike read-out process. This method is very dependent on the current spike which occurs by a transient ON state between two logic states. In their implementation, two combinations have been evaluated out of two possible combinations for two CRS devices. Assume voltage, ΔV, is applied across a CRS device that is exceeded its RESET threshold, in this situation this device changes its stored logic, D, if D is a certain logic depends on the signature of ΔV. Furthermore, if two CRS devices are connected together, that intermediate point can be connected to either ground or power supply to generate NOR/AND gate. That is the reason that no more possible state can be assumed using such approach.

In Fig. 8a, the output is initially charged and it is discharged depends on D and In. In Fig. 8b complementary of signal X is applied to the device. Obviously, more operations can be done sequentially and they require one (or several) initialization but this is a drawback for all of the available Boolean logic operations reported in [4, 37]. Pull-up (charge) voltage is enough to push a device to its ON state and not writing a logic, $V_{\mathrm{th,S}} < V_{\mathrm{pu}} < V_{\mathrm{th,R}}$. NOT function can be also implemented using a single CRS if D stores (the data) A and $X = 0$, $F = \overline{A}$. Here we remove the outputs' complementary signals, AND and NOR. The yellow highlights show the OR-plane and the rest are in the AND-plane.

In [16], two comprehensive forms of building logical gates were introduced. The first form, allows storing one or more inputs as device state and the second method does not. Figure 8 illustrates how CRS works as an implementation of a not implication, NIMP, operation and how NAND and OR operations can be implemented using a single CRS device. Figure 8e and f follow similar phenomenon but in a form of a PLA. The idea is to have a logic → ON transition in the OR-plane whenever an output product term is addressed. From the NIMP operation, we know that if the applied inputs are part of the output product terms, that bit-line does not discharged



D	X	out
0	0	1
1	0	0
0	1	1
1	1	1

D	X	NAND
0	0	1
1	0	1
0	1	1
1	1	0

D	X	OR
0	0	0
1	0	1
0	1	1
1	1	1

Fig. 8 CRS-based logic gate structures. (**a**) D represents stored data, X is an input, and R_{pu} is pull-up resistor. (**b**) Shows how a not implication, NIMP, can be implemented. Here $q' \leftarrow D$ NIMP X. (**c**) Two inputs NAND gate is implemented by storing one input as device state and another one as an actual input. (**d**) Similar to NAND but complementary of D stored in the CRS and X is applied as an input. (**e**) and (**f**) are PLA implementation of the two logic gates

Table 2 CRS-based logic implementation with two inputs and two CRSs, $F = \overline{D_1} \cdot \overline{D_2} + \overline{D_1} \cdot X_2 + \overline{D_2} \cdot X_1 + X_1 \cdot X_2$

D_1	D_2	X_1	X_2	Function
$\overline{A}$	A	0	B	$A \cdot B$ (AND)
A	$\overline{A}$	0	$\overline{B}$	$\overline{A + B}$ (NOR)
A	$\overline{A}$	$\overline{B}$	B	$A \oplus B$ (XOR)
$\overline{A}$	A	$\overline{B}$	B	$A \odot B$ (XNOR)

so there will be enough voltage across the output CRS device with stored logic "1" (greens) to turn to ON and conduct significantly more current to charge the output signal load.

In the case of using differential voltage pairs, $V_{\mathrm{pu}} = -V_{\mathrm{pd}} = 1.4$ V was selected as 2.8 V is the READ voltage (in Fig. 7), where V_{pu} and V_{pd} are pull-up and pull-down voltages. Here we applied $V_{\mathrm{pu}} = 2.8$ V and $V_{\mathrm{pd}} = 0$ V, so we used 0.25 µm CMOS transistors in our CMOS domain. Therefore, equivalent input voltage for logics "1" is 2.8 V and for logics "0" is 0 V. The pull-up and pull-down resistors, R_{pu} and R_{pd}, are both equal to $R_{\mathrm{LRS}}\sqrt{2(r+1)}$, where $r = R_{\mathrm{HRS}}/R_{\mathrm{LRS}}$ [18]. The used peripheral CMOS circuitries can be found in [36]. The sense amplifier was designed for voltage sensitivity more than 100 mV.

Assuming we have two inputs, X_1 and X_2, and two CRS devices, D_1 and D_2, connected to these inputs and a charged bit-line. A number of functions can be implemented by writing $\overline{F} = D_1 \cdot \overline{X_1} + D_2 \cdot \overline{X_2}$, hence, $F = \overline{D_1} \cdot \overline{D_2} + \overline{D_1} \cdot X_2 + \overline{D_2} \cdot X_1 + X_1 \cdot X_2$. The first term, $\overline{D_1 D_2}$, indicates that if both CRSs store "0" TRUE ($F = 1$) is implemented. Some other function that is implemented using this configuration are shown in Table 2. Figure 9 (a) illustrates simulation of a two input NAND

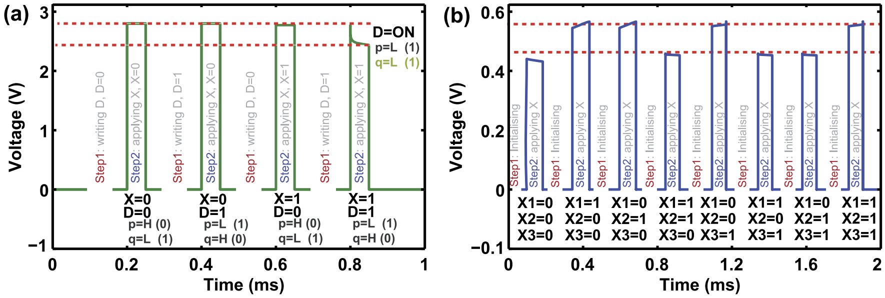


Fig. 9 CRS-based logic gate simulations. (**a**) A 2-input NAND gate (Fig. 8 (c)) simulation. In this style, we are allowed to store one input as the CRS state. (**b**) A 3-input XOR (SUM) function, implemented in a PLA structure. In both cases, (**a**) and (**b**), dashed red line show worst-case low and high output voltages that are sent to sense amplifiers. Due to limited space, complementary output, XNOR, is not shown. Initialization in (**b**) means, the array should be initialized before the next logical operation and this is the main reason that the first approach (in (**a**)) is a far more efficient implementation in terms of both hardware and number of steps. No initialization is required in (**a**), because 'writing *D*' effectively means writing one of the input's logic into the device

function. The most significant advantage of this method is that the initialization step (step 1) which is writing data into CRS arrays and not a simple refreshing cycle. While in a PLA structure, Fig. 9(b), the initialization is a refreshing cycle. Furthermore, in computer arithmetic operations signals arrive with relative delays, like SUM results and CARRY output, that can be used in parallel with the programming of CRS arrays. In [21], we demonstrate a CRS-based content addressable memory, which promises an entirely passive implementation of a content addressable memory array (excluding sensing and addressing circuitry).

Acknowledgement This work was supported by an Early Career Researcher grant from the Melbourne School of Engineering, The University of Melbourne.

References

1. Alibart, F., Zamanidoost, E., Strukov, D.B.: Pattern classification by memristive crossbar circuits using ex situ and in situ training. Nat. Commun. **4** (2013)
2. Backus, J.: Can programming be liberated from the von Neumann style?: A functional style and its algebra of programs. Commun. ACM **21**(8), 613–641 (1978)
3. Bienenstock, E.L., Cooper, L.N., Munro, P.W.: Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. J. Neurosci. **2**(1), 32–48 (1982)
4. Borghetti, J., Snider, G., Kuekes, P., Yang, J., Stewart, D., Williams, R.: 'Memristive' switches enable 'stateful' logic operations via material implication. Nature **464**(7290), 873–876 (2010)
5. Caporale, N., Dan, Y.: Spike timing-dependent plasticity: a Hebbian learning rule. Annu. Rev. Neurosci. **31**, 25–46 (2008)
6. Chang, T., Jo, S.-H., Lu, W.: Short-term memory to long-term memory transition in a nanoscale memristor. ACS Nano **5**(9), 7669–7676 (2011)

7. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
8. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
9. Hasegawa, T., Ohno, T., Terabe, K., Tsuruoka, T., Nakayama, T., Gimzewski, J., Aono, M.: Learning abilities achieved by a single solid-state atomic switch. Adv. Mater. **22**(16), 1831–1834 (2010)
10. Izhikevich, E.M., Desai, N.S.: Relating STDP to BCM. Neural Comput. **15**(7), 1511–1523 (2003)
11. Jackson, B.L., Rajendran, B., Corrado, G.S., Breitwisch, M., Burr, G.W., Cheek, R., Gopalakrishnan, K., Raoux, S., Rettner, C.T., Padilla, A., et al.: Nanoscale electronic synapses using phase change devices. ACM J. Emerg. Technol. Comput. Syst. **9**(2), 12 (2013)
12. Jeong, D.S., Kim, I., Ziegler, M., Kohlstedt, H.: Towards artificial neurons and synapses: a materials point of view. RSC Adv. **3**(10), 3169–3183 (2013)
13. Jiang, H., Xia, Q.: Effect of voltage polarity and amplitude on electroforming of TiO_2 based memristive devices. Nanoscale **5**(8), 3257–3261 (2013)
14. Jo, S., Chang, T., Ebong, I., Bhadviya, B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. **10**(4), 1297–1301 (2010)
15. Kavehei, O., Iqbal, A., Kim, Y., Eshraghian, K., Al-Sarawi, S., Abbott, D.: The fourth element: characteristics, modelling and electromagnetic theory of the memristor. Proc. R. Soc. A, Math. Phys. Eng. Sci. **466**(2120), 2175 (2010)
16. Kavehei, O., Al-Sarawi, S., Cho, K.-R., Iannella, N., Kim, S.-J., Eshraghian, K., Abbott, D.: Memristor-based synaptic networks and logical operations using in-situ computing. In: International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp. 137–142 (2011)
17. Kavehei, O., Cho, K., Lee, S., Kim, S., Al-Sarawi, S., Abbott, D., Eshraghian, K.: Fabrication and modeling of Ag/TiO_2/ITO memristor. In: 54th IEEE International Midwest Symposium on Circuits and Systems, pp. 1–4 (2011)
18. Kavehei, O., Al-Sarawi, S., Cho, K.-R., Eshraghian, K., Abbott, D.: An analytical approach for memristive nanoarchitectures. IEEE Trans. Nanotechnol. **11**(2), 374–385 (2012)
19. Kavehei, O., Cho, K.-R., Lee, S.-J., Al-Sarawi, S., Eshraghian, K., Abbott, D.: Integrated memristor-mos (M2) sensor for basic pattern matching applications. J. Nanosci. Nanotechnol. **13**(5), 3638–3640 (2013)
20. Kavehei, O., Lee, S.-J., Cho, K.-R., Al-Sarawi, S., Abbott, D.: A pulse-frequency modulation sensor using memristive-based inhibitory interconnections. J. Nanosci. Nanotechnol. **13**(5), 3505–3510 (2013)
21. Kavehei, O., Linn, E., Nielen, L., Tappertzhofen, S., Skafidas, E., Valov, I., Waser, R.: An associative capacitive network based on nanoscale complementary resistive switches for memory-intensive computing. Nanoscale **5**(11), 5119–5128 (2013)
22. Kim, K.-H., Gaba, S., Wheeler, D., Cruz-Albrecht, J.M., Hussain, T., Srinivasa, N., Lu, W.: A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. Nano Lett. **12**(1), 389–395 (2011)
23. Li, S., Zeng, F., Chen, C., Liu, H., Tang, G., Gao, S., Song, C., Lin, Y., Guo, D., et al.: Synaptic plasticity and learning behaviours mimicked through Ag interface movement in an Ag/conducting polymer/Ta memristive system. J. Mater. Chem. C **1**(34), 5292–5298 (2013)
24. Lim, H., Jang, H.-W., Lee, D.-K., Kim, I., Hwang, C.S., Jeong, D.S.: Elastic resistance change and action potential generation of non-faradaic Pt/TiO_2/Pt capacitors. Nanoscale **5**(14), 6363–6371 (2013)
25. Linn, E., Rosezin, R., Kügeler, C., Waser, R.: Complementary resistive switches for passive nanocrossbar memories. Nat. Mater. **9**(5), 403–406 (2010)
26. Menzel, S., Tappertzhofen, S., Waser, R., Valov, I.: Switching kinetics of electrochemical metallization memory cells. Phys. Chem. Chem. Phys. **15**(18), 6945–6952 (2013)
27. Mott, N., Gurney, R.: Electronic Processes in Ionic Crystals. Dover, New York (1964). Chapter 2

28. Mouttet, B.: Proposal for memristors in signal processing. Nano-Net 11–13 (2009)
29. Ohno, T., Hasegawa, T., Tsuruoka, T., Terabe, K., Gimzewski, J.K., Aono, M.: Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. Nat. Mater. **10**(8), 591–595 (2011)
30. Ovshinsky, S.R.: Reversible electrical switching phenomena in disordered structures. Phys. Rev. Lett. **21**, 1450–1453 (1968)
31. Ovshinsky, S.R.: The ovonic cognitive computer: a new paradigm. EPCOS Library (2004)
32. Pershin, Y., Di Ventra, M.: Practical approach to programmable analog circuits with memristors. IEEE Trans. Circuits Syst. I, Regul. Pap. **57**(8), 1857–1864 (2010)
33. Pershin, Y., Di Ventra, M.: Memory effects in complex materials and nanoscale systems. Adv. Phys. **60**(2), 145–227 (2011)
34. Pfeil, T., Potjans, T.C., Schrader, S., Potjans, W., Schemmel, J., Diesmann, M., Meier, K.: Is a 4-bit synaptic weight resolution enough?—constraints on enabling spike-timing dependent plasticity in neuromorphic hardware. Front. Neurosci. **6** (2012)
35. Pickett, M.D., Strukov, D.B., Borghetti, J.L., Yang, J.J., Snider, G.S., Stewart, D.R., Williams, R.S.: Switching dynamics in titanium dioxide memristive devices. J. Appl. Phys. **106**(7), 074508 (2009)
36. Qureshi, M.S., Pickett, M., Miao, F., Strachan, J.P.: CMOS interface circuits for reading and writing memristor crossbar array. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2954–2957 (2011)
37. Rosezin, R., Linn, E., Kügeler, C., Bruchhaus, R., Waser, R.: Crossbar logic using bipolar and complementary resistive switches. IEEE Electron Device Lett. **32**(6), 710–712 (2011)
38. Snider, G.S.: Cortical computing with memristive nanodevices. SciDAC Rev. **10**, 58–65 (2008)
39. Snider, G.S.: Instar and outstar learning with memristive nanodevices. Nanotechnology **22**, 015201 (2011)
40. Song, S., Miller, K., Abbott, L.: Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nat. Neurosci. **3**, 919–926 (2000)
41. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
42. Suri, M., Querlioz, D., Bichler, O., Palma, G., Vianello, E., Vuillaume, D., Gamrat, C., DeSalvo, B.: Bio-inspired stochastic computing using binary CBRAM synapses. IEEE Trans. Electron Devices **60**(7) (2013)
43. Thakoor, S., Moopenn, A., Daud, T., Thakoor, A.: Solid-state thin-film memistor for electronic neural networks. J. Appl. Phys. **67**(6), 3132–3135 (1990)
44. Valov, I., Linn, E., Tappertzhofen, S., Schmelzer, S., van den Hurk, J., Lentz, F., Waser, R.: Nanobatteries in redox-based resistive switches require extension of memristor theory. Nat. Commun. **4**, 1771 (2013)
45. Waser, R.: Nanoelectronics and Information Technology. Wiley-VCH, Weinheim (2012)
46. Waser, R., Aono, M.: Nanoionics-based resistive switching memories. Nat. Mater. **6**(11), 833–840 (2007)
47. Whitehead, A., Russell, B.: Principia Mathematica, vol. 2 (1912)
48. Widrow, B.: An adaptive 'ADALINE' neuron using chemical "memistors". Stanford Electronics Laboratories Technical Report, Tech. Rep. TR-1553-2, 23 Oct. 1960
49. Xia, Q., Pickett, M.D., Yang, J.J., Li, X., Wu, W., Medeiros-Ribeiro, G., Williams, R.S.: Two- and three-terminal resistive switches: nanometer-scale memristors and memistors. Adv. Funct. Mater. **21**(14), 2660–2665 (2011)
50. Yang, J.J., Strukov, D.B., Stewart, D.R.: Memristive devices for computing. Nat. Nanotechnol. **8**(1), 13–24 (2012)
51. Yang, J.J., Zhang, M.-X., Pickett, M.D., Miao, F., Strachan, J.P., Li, W.-D., Yi, W., Ohlberg, D.A., Choi, B.J., Wu, W., et al.: Engineering nonlinearity into memristors for passive crossbar applications. Appl. Phys. Lett. **100**, 113501 (2012)

52. Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., Wong, H.-S.P.: A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: experimental characterization and large-scale modeling. In: IEEE International Electron Devices Meeting, pp. 239–242 (2012)
53. Zamarreño-Ramos, C., Camuñas-Mesa, L., Pérez-Carrasco, J., Masquelier, T., Serrano-Gotarredona, T., Linares-Barranco, B.: On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. In: Frontiers in Neuroscience, vol. 5 (2011)

Memory Effects in Multi-terminal Solid State Devices and Their Applications

Davide Sacchetto, Pierre-Emmanuel Gaillardon, Yusuf Leblebici, and Giovanni De Micheli

Abstract We give a general overview on Silicon nanowire-based multi-terminal memristive devices. The functionality of the devices can be used for logic, memory and sensing applications. It is shown that three- and four- terminal memristive devices can be used for both logic and memory applications. In particular, Schottky-barrier silicon nanowire FETs are very interesting devices due to their CMOS-compatibility and ease of fabrication.

1 Introduction

The main source of the CMOS success lies in the exponential increase of device density that the silicon industry kept true for more than 40 years reducing the unit cost of integrated circuits. Recently, the pace of scaling has been slowing down due to approaching fundamental limits at the device level. While the paradigm of scaling is still alive, researchers are striving to follow Moore's law by focusing on new materials, new device structures and new state variables. The basic building block for circuits has always been the four-terminal planar transistor but new versions of the MOSFET structure such as the double-gate FET, the FinFET, the gate-all-around nanowire FET can be found in commercial products. Due to the natural limitations of materials, future nano-scale circuits will have to exploit more efficient ways for computation and memory storage. One possible scenario envisages an end of

D. Sacchetto (✉)
EPFL STI IEL LSM, ELD 337 (Bâtiment ELD), Station 11, 1015 Lausanne, Switzerland
e-mail: davide.sacchetto@epfl.ch

P.-E. Gaillardon
EPFL IC ISIM LSI1, INF 339 (Bâtiment INF), Station 14, 1015 Lausanne, Switzerland
e-mail: pierre-emmanuel.gaillardon@epfl.ch

Y. Leblebici
EPFL STI IEL LSM, ELD 340 (Bâtiment ELD), Station 11, 1015 Lausanne, Switzerland

G. De Micheli
EPFL IC ISIM LSI1, INF 341 (Bâtiment INF), Station 14, 1015 Lausanne, Switzerland

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_20,

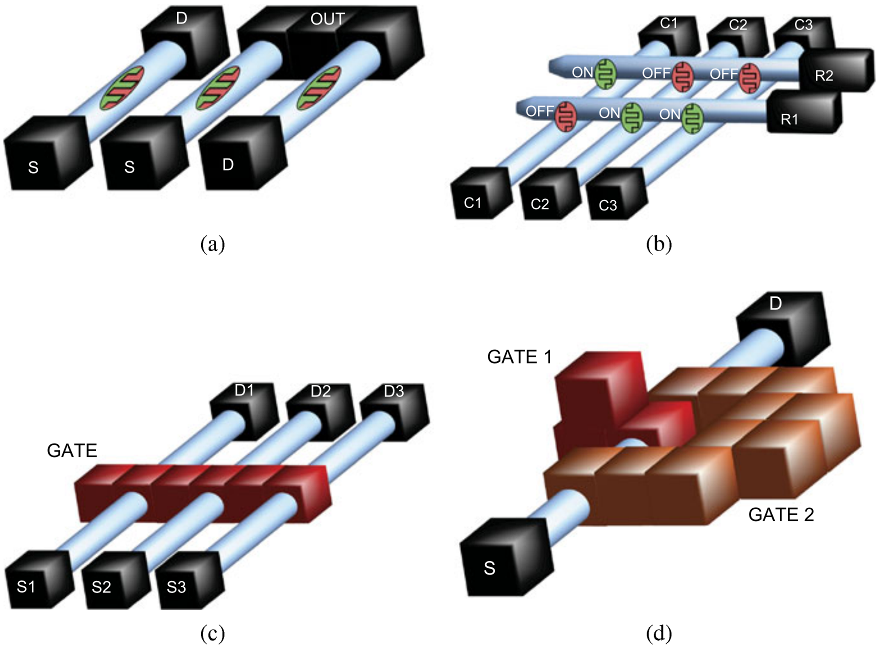


Fig. 1 (**a**) Parallel nanowire two-terminal memristive devices. (**b**) Crossbar array consisting of memristive cross-points (two-terminal). (**c**) Gate controlled three-terminal nanowire memristive device. (**d**) Double-gate four-terminal nanowire memristive devices

charge-based technologies, after which computation will rely on alternative, more power efficient state variable manipulation. A long list of fundamental state variables other than charge includes the spin, phase, multipole orientation, mechanical position, polarity, orbital symmetry, magnetic flux quanta, molecular configuration, and other quantum states [1]. Technologies using new state variables would have to be implemented within a completely new technological platform, and cannot be seen as CMOS-compatible alternatives.

The recent realization of devices by Stanley Williams and his team [2] gave new push to solid state research for logic and memory applications. For instance, ultra-dense crossbar memristive memory arrays can be made thanks to the compactness of the two terminal junction. Complementary logic based on two-terminal memristive devices (see Fig. 1a) or ultra-dense crossbar arrays with memristive cross-points (see Fig. 1b) can dramatically improve device density up to 10^{11} bits per square centimetre [3]. Moreover, the use of memristive effects as new state variables for computation can be exploited to build new types of functional devices with three- or four-terminals (see Figs. 1c and 1d, respectively).

The physical realization of the memristor, whose behavior was postulated by Leon Chua [4] and generalized by Chua and Kang [5] for memristive devices and systems, offers a completely new set of possibilities for logic operations [6]. It is worth noting that, a generalized model for memristive systems can be implemented under DC, small signal and sinusoidal excitation [5]. The implications of such modeling is linked with the observation of memristive functionalities over a broad range of technologies based on nanoelectronic and nanoionic behaviors.

The content of this chapter is organized as follows. First, a generalization of memristive devices is given (Sect. 2), including resistive RAMs, two-terminal devices with memory and multi-terminal memristive devices. Applications of resistive RAMs are discussed for application, such as standalone memory (Sect. 3.1), FPGAs (Sect. 3.2) and programmable TSVs (Sect. 3.3). Then, a discussion on the applications multi-terminal memristive devices is given in Sect. 4. Finally, we draw the conclusions.

2 Generalization of the Memristive Devices

In this section we survey the generalization of memristive theory, which is mainly a mathematical description of memory effects that can be experimentally observed in several electronic devices. The memristive functionality is not a unique property of two-terminal passive devices but mainly a memory effect related to internal state variable changes. For instance, the memristive functionality can also arise from a delayed switching response of a Zener diode [7]. As described by Di Ventra et al. [8], the memory effects in nanoscaled devices can be generalized for resistance, capacitance and inductance to their time dependent and non-linear responses. If $u(t)$ and $y(t)$ are two complementary constitutive circuit variables, denoting the input and the output of the system, respectively, and x is a n-order vector of internal state variables, the existance of a nth order u-controlled memory element as that defined by:

$$\begin{cases} y(t) = g(x, u, t)u(t) \\ \dot{x} = f(x, u, t) \end{cases} \tag{1}$$

where g is a generalized system response and f is a continuous n-dimensional vector function. As described in [8], the relation between current and voltage defines a *memristive system*, while the relation between charge and voltage defines a *memcapacitive system* and the flux-current relation defines a *meminductive system*. Equations (1) can be further generalized for cases where the internal state is described by a continuous function or functions instead of discrete variables as well as cases for which the internal state variables follow a *stochastic differential equation* rather than a deterministic one. In the next sections, we review some of the two-terminal passive devices that can be modeled as memristive. The memristive functionality is seen as arising from different physical mechanism for different device classes. Thus, three- and four-terminal devices that show memristive functionality can be used for electrical control signals (either voltage or current). It is important to remember that a device showing a “pure” flux-charge relationship has not been found yet. Conversely, internal state variables can provide a stimulus response that can be modeled as memristive, thus adding more degrees of freedom for circuit designers.

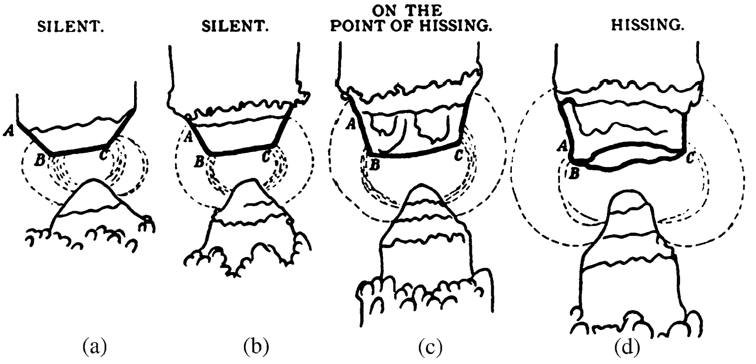


Fig. 2 The "hissing of the electric arc" as presented to the Institution of Electrical Engineers in 1899 by Hertha Ayrton. Reprinted from [10]

2.1 Resistive RAMs

A mathematical definition of two-terminal passive mem-resistive devices has been formulated by Chua and based on this interpretation a mem-resistive device would present three fingerprints:

- pinched hysteresis loop in the $i(t)$–$v(t)$ plane;
- passive: $i = 0$ when $v = 0$;
- frequency dependence with the input signal $u(t)$

However, many electronic devices that were previously classified as "*memristors*" or as "*memristive devices and systems*" have been recently re-classified as their actual behavior requires an extension of the memristive theory [9]. Nevertheless, by looking at these three fingerprints, it has been recently reported that mem-resistive behavior has been noticed in several electric devices in the past two centuries [10]. More specifically, a predominance of mem-resistive behavior seems to be linked with an electric discharge phenomenon, and as such it is believed that the first memristor was actually found in 1899 by Herta Ayrton (see Fig. 2). Later on the solid-state realization of devices by Stanley Williams and his team [2] which showed mem-resistive fingerprints gave new push to research on nano-scaled mem-resistive devices. It has been stated that the mem-resistive "fingerprints criteria" are satisfied by several resistive switching materials, and as such it becomes clear that the mathematical formulation of memristors and memristive devices does not necessarily link to a specific physical phenomenon. Considering resistive switching materials, a classification based on physical phenomena has been provided by Waser (see Fig. 3) [11].

Two-terminal memristive devices can be based on metal/oxide switches (one example is the Pt/TiO2/Pt sandwich shown in Fig. 4), such as for SiO_x [12], HfO_2 [13], CuO [14], NiO [15], ZnO [16], Al_2O_3 [17], VO_2 [18], $SrTiO_3$ [19]. These devices behave as solid-state electrochemical switches, whose resistance is defined

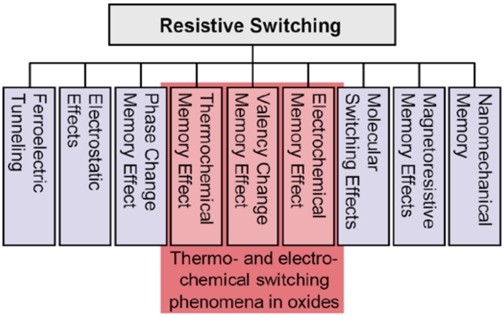

Resistive Switching
Ferroelectric Tunneling
Electrostatic Effects
Phase Change Memory Effect
Thermochemical Memory Effect
Valency Change Memory Effect
Electrochemical Memory Effect
Molecular Switching Effects
Magnetoresistive Memory Effects
Nanomechanical Memory
Thermo- and electro-chemical switching phenomena in oxides

Fig. 3 A classification of resistive switching devices depending on the underlying physical mechanisms as provided by Waser [11]

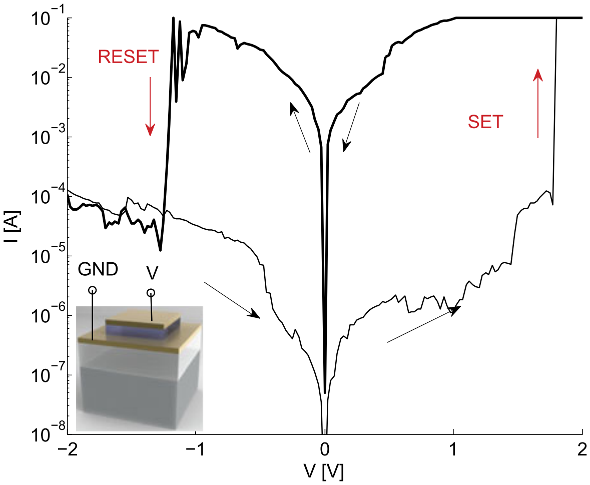


Fig. 4 Resistive switching through I–V sweeps for planar Pt/TiO$_2$/Pt [23]. The inset shows the Pt/TiO$_2$/Pt ReRAM sandwich layer on top of an insulated substate with the two terminal electrical contacts

by a metallic filament formation mechanism related to the solid-state redox reactions stimulated by the polarity of the applied electric field [20]. One example is the CuO-based ReRAM of Dong et al. [14] (see Fig. 5), that shows repeatable resistive switching at very low voltages.

The well known TiO$_2$-based ReRAM [21, 22] seems to be based on a different mechanism, which is attributed to the vacancy/dopant diffusion in the oxide layer. The re-distribution of oxygen vacancies into the TiO$_2$ depends on the polarity of the applied voltage, and it causes the switching between a semiconductor state into a metallic one. Typical ReRAM functionality of the TiO$_2$-based ReRAM is shown in Fig. 4 [23] and Fig. 6 [21], with the resistivity switching by 5 orders of magnitude.

Another type of two-terminal memristive device is the *phase change* (*PC*) RAM [24]. The main switching mechanism is based on phase transition between an amorphous and a crystalline type due to Joule heating effects controlled by a voltage pulse. For instance, Si nanowires can be engineered such that melting and solidification processes can be iterated, thus giving rise to alternate resistance states [25].

Another class is based on polymers [26]. Several memristive switches can be build by inter-posing a bio-molecule layer with properties ranging from molecule-

Fig. 5 Switching mechanism of a CuO-based ReRAM. Reprinted from [14]

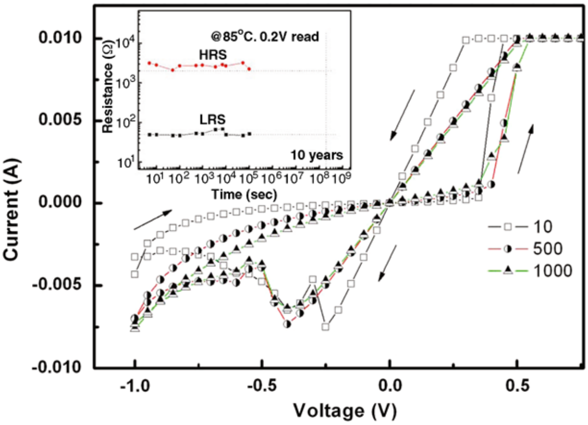
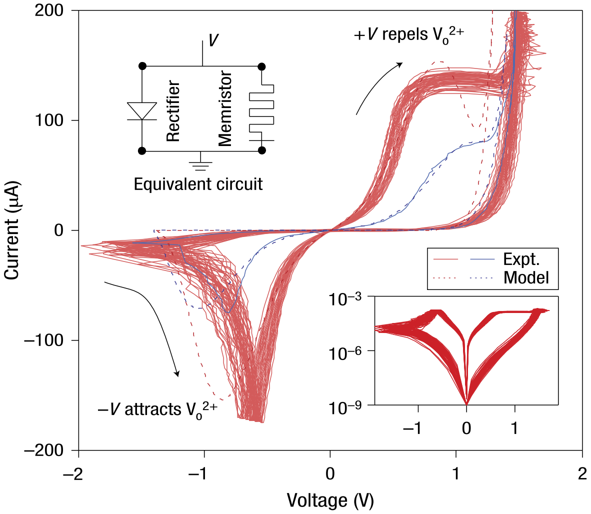


Fig. 6 Pinched hysteresis loop from a Pt/TiO_2/Pt memristive junction. Reprinted from [21]



dependent switching, such as rotaxanes [3], or more in general on inter-locked molecules [27] but also on molecule-independent switching, where a filament formation mechanism through the molecular layer is involved [28].

A fifth class belongs to spintronics [29]. The spin-transfer induced domain wall motion in a spin valve structure is by nature a memristive effect and is confirmed by some recently published results [30]. Moreover, Pershin et al. demonstrated that electron-spin polarization controlled by the external voltage applied to a spintronic device (see Fig. 7), acts as a state variable that can be modeled as memristance [31].

In all these devices the amplitude and frequency of the input signal contribute in the formation of a so called pinched hysteresis loop, whose salient feature is its zero crossing property [5], which is critical for ultra low power operation.

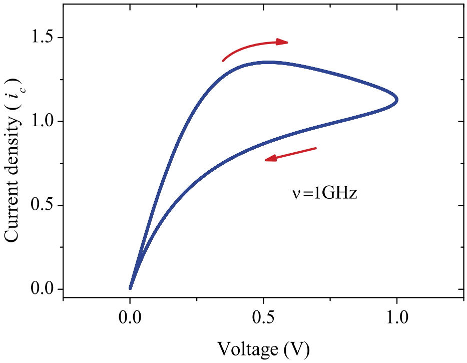


Fig. 7 Unipolar I–V simulation for a semiconductor spintronic system. Reprinted from [31]

2.2 *Mem-Capacitive Switching Devices*

As per two-terminal memristive devices, two-terminal mem-capacitive devices can be based on several mechanisms. A classification has been given by Pershin et al. [8], and it is divided mainly in two categories: geometrical and permittivity-related memcapacitors.

An example of mem-capacitive switching device based on geometrical effects consists of a capacitor with a movable plate, whose movement depends on the applied electric field [32]. One such device would be a capacitor having one of the electrodes being elastic. If a voltage pulse is applied across the capacitor, the elastic plate moves towards the other electrode, in response of the electric field, thus modifying its capacitance. One possible realization of the geometrical mem-capacitor can be obtained from a *Nano Electro Mechanical System* (*NEMS*) resonator configuration. Such devices can be, for instance, fabricated with a CMOS compatible process flow on SOI substrates. As depicted in Fig. 8, a NEMS device consisting of a suspended SiNW and Al metal electrodes can be utilized as a geometrical-effect memcapacitor. In this device, the SiNW in the middle is utilized as an elastic electrode [33]. It is worth mentioning that the NEMS resonator can also be utilized as resistive switching device, but this does not represents a case of two-terminal memristive device because its pinched hysteresis loop does not depend on the frequency of actuation, as it results on the absolute value of the DC signal (see Fig. 9).

A second example is a device whose permittivity modifies upon the application of a voltage bias. A memcapacitor proposed by Di Ventra et al [34], consists of a parallel plate capacitor with additional floating plates in between, separated by dielectrics. The internal charge can move between the different plates and its polarization can screen the electric field applied at the external plates. This way, based on the distribution of the charges in the polarized plates, a memcapacitive behavior is obtained [34].

Another memcapacitive device has been demonstrated with a III–V diode and high-κ dielectrics [35].

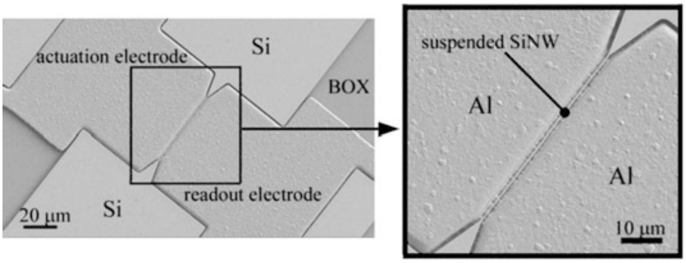
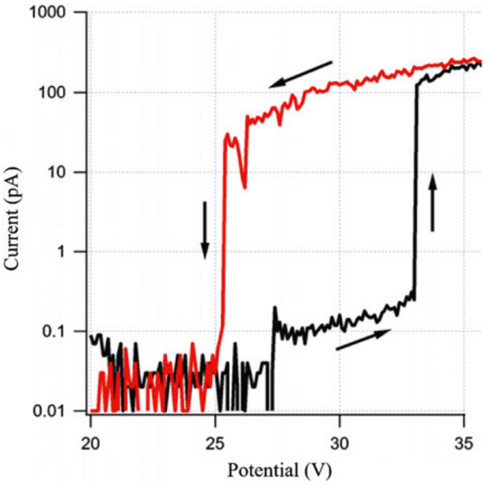


Fig. 8 NEMS resonator consisting of a SiNW and two Al side-electrodes. A two-terminal mem-capacitor configuration can be obtained by using the elastic SiNW and either one of the Al pads as the electrodes. Reprinted from [33]

Fig. 9 Resistive switching characteristic of SiNW NEMS [33]. The hysteresis arise from the var-der-Walls forces that retain the SiNW at pull-out. Since the hysteresis does not depends on the actuation frequency, this resistive switching mechanism cannot be considered as memristive



2.3 Mem-Inductive Switching Devices

Similarly to the mem-capactive devices, geometrical effects [36] and permeability-related effects [37] can lead to an inductive system showing memory, thus forming a mem-inductive device. Several properties, including mechanical, electrical, magnetic and thermal were discussed by Pershin et al. [8] as possible mechanisms that can be exploited in a mem-inductive system.

2.4 Three-Terminal Memristive Devices

The new concept of three-terminal memristive device is presented in terms of memristive functionality for logic and memory applications.

Examples of three-terminal memristive devices are the electrochemical organic memristor [39], the solid-electrolyte nanometer switch [40], the ferroelectric

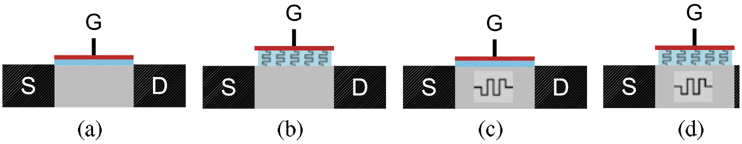


Fig. 10 Categorization of FETs with memristive functionalities: (**a**) Conventional FET. (**b**) FET with memristive gate dielectric. (**c**) Gated memristor (memristive channel). (**d**) Gated memristor with memristive dielectric

FET [41], non passivated transistor devices (here cite the non-passivated CNT and graphene transistors), Carbon Nanotube and/or graphene transistors and the ambipolar Si nanowire Schottky barrier FET [42, 43].

A classification of three-terminal memristive devices can be based from the general concept of the FET structure (see Fig. 10a) in which memristive functionality can be inserted either by engineering the gate dielectric or by gating a memristive channel. For instance, trap charging dielectric layers inserted between the channel and the gate fall into the category of FET with capacitive memory storage (Fig. 10b). One example in this category can be the flash memory for which the trap charging into the gate dielectric influences the transconductance state of the channel [44]. Thus a first category that exploits the operation of writing/erasing cycles into the gate dielectric will be a generalization of the flash memory concept, for which volatility of the charges that are injected into the trap charging layer can be tuned accordingly to a desired frequency response. Such a device has been recently demonstrated by Ziegler et al. [44] by using a FLASH transistor in a two-terminal configuration. As depicted in Fig. 11, the device can be operated such that a bipolar resistive switching behavior is measured and non-volatility is a consequence of trap-charging into the floating gate.

Another memristive functionality is linked with charging/decharging mechanisms into the gate dielectric. We measure I_{ds} for constant $V_{\mathrm{ds}} = 1$ V while sweeping V_{gs} back and forth between -5 V and $+5$ V (see Fig. 12d). The devices are not annealed, and the source is connected to the substrate. This measurement is repeated for different integration times; which is a parameter of the measurement set-up that can be set by the operator; and it represents the time over which the measurement is repeated and averaged. The I_{ds}–V_{gs} curves show an ambipolar behavior, meaning a large current conductance under either high and low gate bias. This is mainly due to the utilization of a silicide (NiSi) having a work function value that falls within the silicon bandgap, and to the utilization of a lightly doped silicon. On the other hand, the I_{ds}–V_{gs} curves have a hysteretic behavior that suggests the hypothesis of charge trapping at the semiconductor/oxide interface of the MOS capacitor, as well as the existence of interface states at the metal/semiconductor junction. Both ambipolarity and hysteresis depend on the integration time (see Fig. 12). When V_{gs} reaches $+5$ V in forward mode, the electrons experience a maximum probability of charges being trapped in the gate oxide, which is enhanced when the integration time is longer. Due to the electron trapping at the gate oxide, the channel operates in accumulation mode, with a lower conductance state for positive V_{gs}, than in the inversion mode

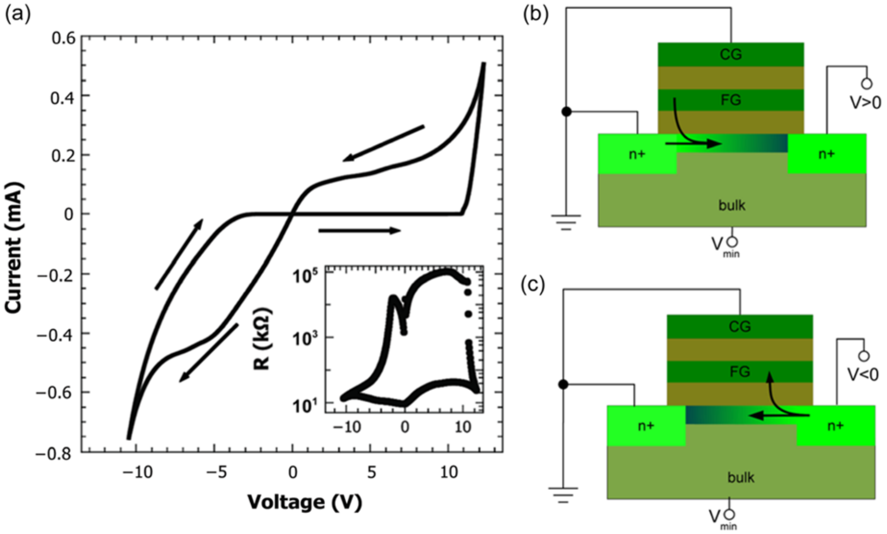


Fig. 11 Two-terminal memristive configuration of a FLASH memory cell. Depending on the polarity of the applied voltage, trap charging dynamics of electrons to and from the floating gate and dielectric is responsible of the hysteresis. Reprinted from [44]

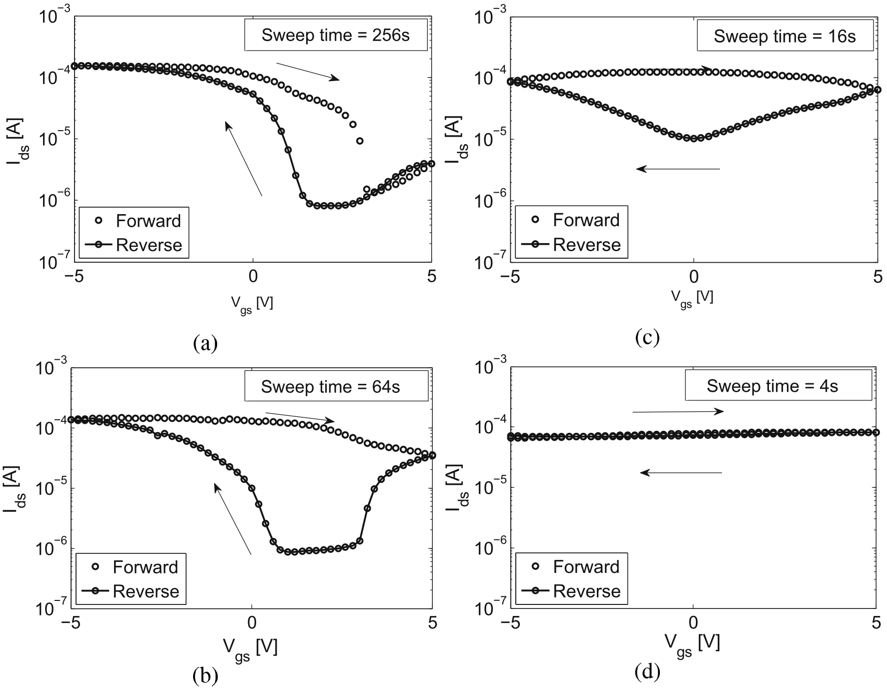
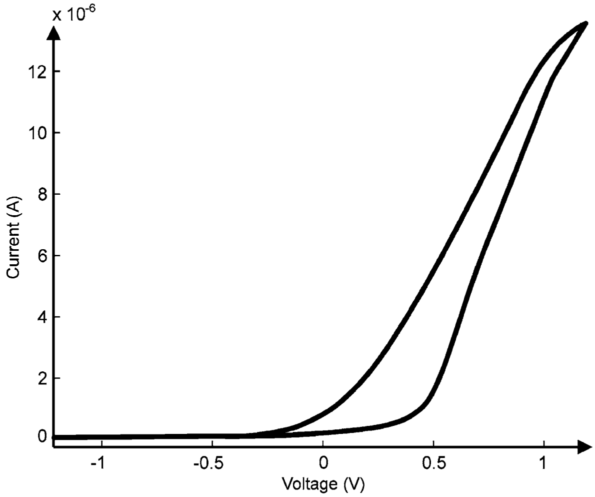


Fig. 12 Hysteretic dependence with the measurement (sweep time) duration, the hysteresis window closes by reducing the measurement time: (**a**) 256 sec (**b**) 64 sec (**c**) 16 sec (**d**) 4 sec

Fig. 13 Cyclic voltage measurement showing bistable memristive behavior. Reprinted from [39]

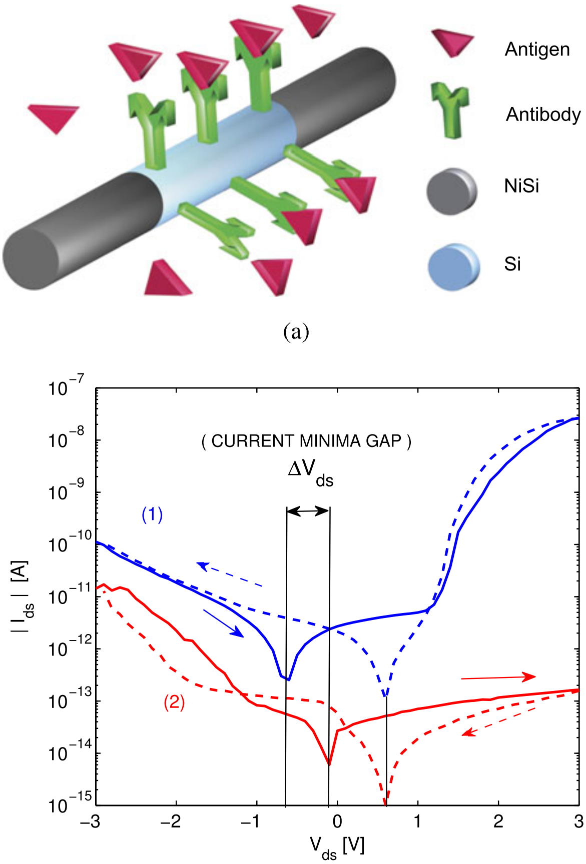


for negative V_{gs}. Sweeping V_{gs} back to negative values, reduces the amount of electrons trapped and the device operates in inversion mode, which restores the higher conductance state.

The threshold at which this high conductance state is reached depends on the integration time. The hysteresis window is larger when the integration time is shorter, because the charges have less time to be trapped and detrapped. In the case of a very short integration time the charges cannot be completely trapped and the lower conductance state is not reached (see Fig. 12).

Another example can be the *zero-capacitor RAM* (*ZRAM*) and the *twin-transistor RAM* (*TTRAM*), which rely on floating body effects in SOI transistor to provide a volatile hysteresis effect [45, 46].

A second category is the one of the gated memristor (Fig. 10c). A few examples are the electrochemical organic memristor [39] (see Fig. 13), the bio-memristive nanowire [38] and the solid-electrolyte nanometer switch [40]. In the electrochemical organic memristor the gate potential is represented by the potential of the bath, which is used to transfer positively charged Rb^{+} ions into a polyaniline (PANI) layer [47]. The conductivity change can be iterated by switching the polarity of the bath potential, thus giving rise to a unipolar I_{ds}–V_{ds} curve that can be modeled as a memristor. In this case the device can be set into either memristive or diode functionality. Similarly, a novel method for bio-sensing that has been recently proposed exploits the memristive effect to detect low concentration of bio-molecules [38]. The device consists of a NiSi/Si/NiSi nanowire structure coated with antibody layer (see Fig. 14a) shows memristive behavior. The hysteresis loop of this device has been demonstrated for detection of low concentrations of bio-molecules (antigen) in a dry environment (see Fig. 14b). Conversely, the three-terminal solid-state electrolyte nanometer switch shows a typical bistable resistance state but using 100 times less current than standard two terminal operation [48]. This device is based



(b)

Fig. 14 (**a**) Suspended functionalized Si nanowire with NiSi extremities. The functionalized layer is capable of trapping antigen molecules which in turn affects the memristive hysteresis behavior, giving a new method for bio-sensing [38]. (**b**) A typical memristive hysteresis is observed; the blue curve (*1*) is measured after drying the sample from de-ionized water. The *red curve* (*2*) is measured after dipping in 5 pM antigen solution and drying. The measured ΔV_{ds} is proportional to the concentration of antigen [38]

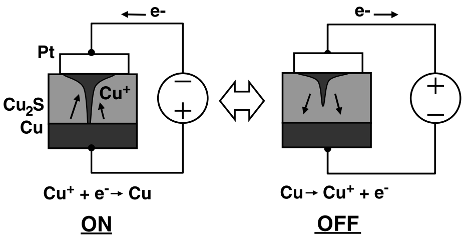


Fig. 15 Controlled filament formation of the three-terminal nanobridge device. Reprinted from [48]

on controlling the filament formation mechanism using the voltage of a gate terminal (see Fig. 15).

The ambipolar SB FET with SiNW channel reported in [49] (see Fig. 10d), falls in both categories of gated memristor and trap charging dielectric, as it shows dynamic trap charging mechanisms at the Schottky junctions and in the gate dielectric insulator. The result depicted in Fig. 16 shows a hysteretic behavior that is reminis-

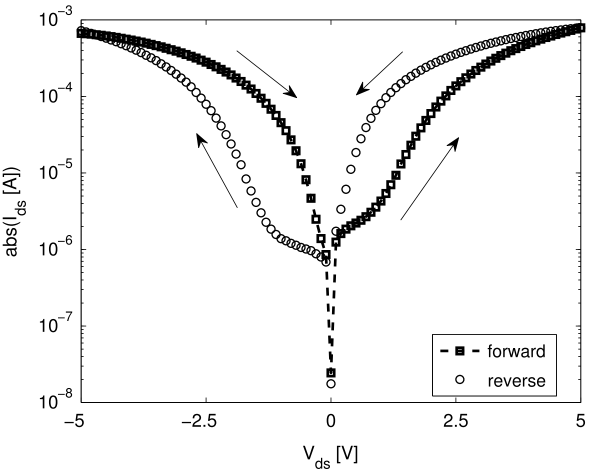


Fig. 16 I_{ds}–V_{ds} characteristic showing the trapping/detrapping of charges at the metal/semiconductor junction. The device channel consists of 10 SiNW in parallel. The forward sweep curve has a symmetrical correspondence with the reverse sweep curve, showing the respective Schottky barrier modulation. A current ratio of about 50 is found at either $V_{ds} = \pm 1$ V. This behavior is typical of two-terminal memristive devices for ReRAM applications

cent of a two-terminal monolithic memristive device [50]. The hysteresis reflects the fact that the I_{ds}–V_{ds} curve for forward V_{ds} sweep is not identical to the same curve for backwards V_{ds} sweep. It can be attributed to the presence of interface states at the metal/semiconductor junctions as reported in literature for Schottky diodes [51]. First, two-terminal measurements are performed. The drain-source current I_{ds} is measured vs. the drain-source voltage V_{ds} at constant $V_{gs} = 5\ V$. The device is equivalent to two back-to-back Schottky diodes. The two diodes operate in opposite regimes: for negative V_{ds}, the source-to-channel diode is reversely biased while the drain-to-channel diode is forward biased. For positive V_{ds} both diodes invert their respective bias conditions. In either case, I_{ds} is limited by the current flowing in the reverse-biased diode. The reverse current of a metal-insulator-semiconductor diode has been observed to be very sensitive to charge trapping at the metal/semiconductor interface [17]. The large current value in the range of mA is most likely due to the large parallel parasitic structure in the bulk. In an ideal Schottky diode, the current is given by:

$$I = I_{\mathrm{S}} \cdot \mathrm{e}^{-\phi_{\mathrm{B}} q/kT} \left(\mathrm{e}^{Vq/kT} - 1\right) \tag{2}$$

with I and V the diode current and voltage respectively, ϕ_{B} the Schottky barrier, k the Boltzmann constant, q the elementary charge and T the absolute temperature. From the measured hysteretic behavior, it seems that the diode curve is modified as follows:

$$I = I_{\mathrm{S}} \cdot \mathrm{e}^{-\phi_{\mathrm{B}} q/kT} \left(\mathrm{e}^{(V - V_0(V))q/kT} - 1\right) \tag{3}$$

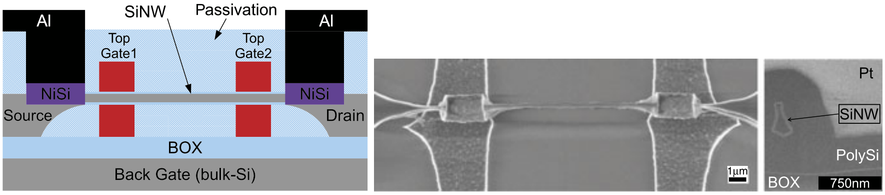


Fig. 17 (**a**) Schematic cross-section of a dual gate device with NiSi source and drain regions on SOI substrate. (**b**) A 20 µm long SiNW with 2 parallel GAA polysilicon gates having 4 µm gate lengths. (**c**) A FIB cut cross-section image showing the SiNW channel surrounded by a 500 nm polysilicon top gate

with V_0 a built-in voltage at the Schottky contact that is positive for a positive V sweep and negative for a negative V sweep.

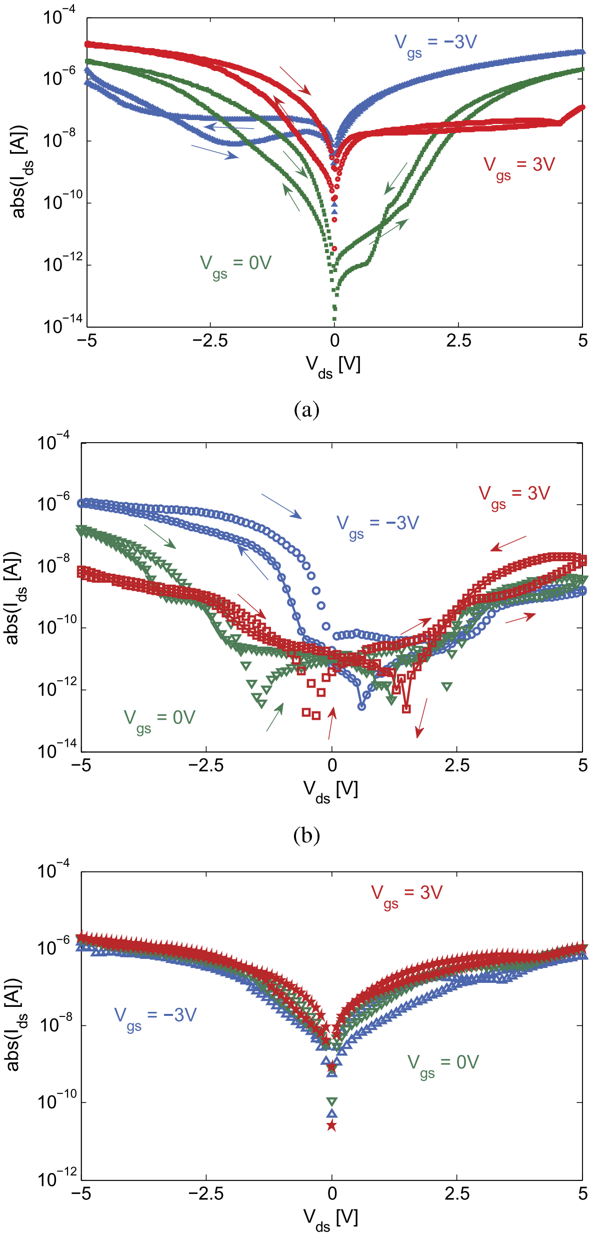
2.5 *Four-Terminal Memristive Devices*

Memristive functionality can be seen as state variable that can be used for more expressive logic gates [6]. For instance, the memristive behavior reported in Sect. 2.4 for the SB SiNW FETs can be tuned by operating on the polarity of the gate voltage. This type of behavior is linked with the double conductance, for holes and electrons. As described in [52] for SB *carbon-nanotube* (*CNT*) FETS, the ambipolarity can be controlled by using an additional control gate, such that it blocks one type of carrier conductance. Following this principle, four-terminal memristive SB SiNW FETs can be built (see Fig. 17). In the following, two modes of operation are reported, depending on the nature of the controlling signal applied at the Si nanowire channel.

2.5.1 Voltage-Controlled 4-T Memristive Device

A voltage-controlled four-terminal memristive Schottky barrier SiNW FET is obtained by using a dual gate configuration such that one of the two gate is controlling a portion of the channel that is between the source/drain contacts and the main gate. This configuration is exploited to control the ambipolarity imbalance, such as for CNT FETs [52]. Since the back-gate voltage modifies the ambipolar conductance, this fact can be used in ambipolar memristive devices to limit the current levels for one of the carriers. A fixed back-gate voltage $V_{bg} = +5$ V (see Fig. 18a) leads to imbalanced bistable hysteresis loops under different V_{gs} voltages. By applying a negative $V_{bg} = -5$ V this imbalance is toggled to the negative side of the characteristics, giving a complementary effect (Fig. 18b). Finally, a $V_{bg} = 0$ V (Fig. 18c) levels off the conductances of electrons and holes, giving a fairly symmetric hysteresis.

Fig. 18 (**a**) Controlled memristance for fixed top and back gate voltages $V_{\mathrm{bg}} = +5$ V. (**b**) Controlled memristance for fixed top and back gate voltages $V_{\mathrm{bg}} = -5$ V. (**c**) Controlled memristance for fixed top and back gate voltages $V_{\mathrm{bg}} = 0$ V



(a)

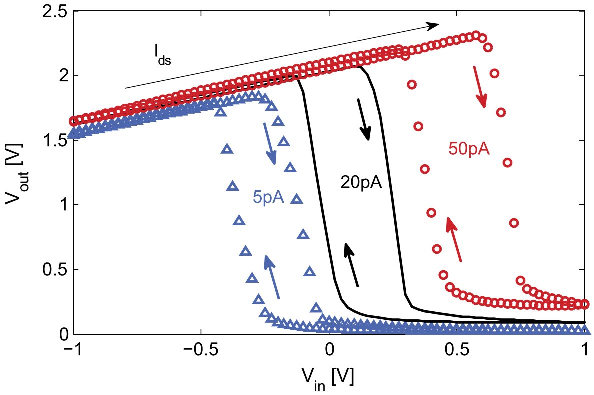
(b)

(c)

2.5.2 Current-Controlled 4-T Memristive Devices

A current-controlled version of the four-terminal memristive Schottky barrier SiNW FET is obtained by using a current I_{ds} bias instead of a V_{ds}. The output voltage is

Fig. 19 Current-controlled memristive Schottky barrier SiNW FET hysteresis loop



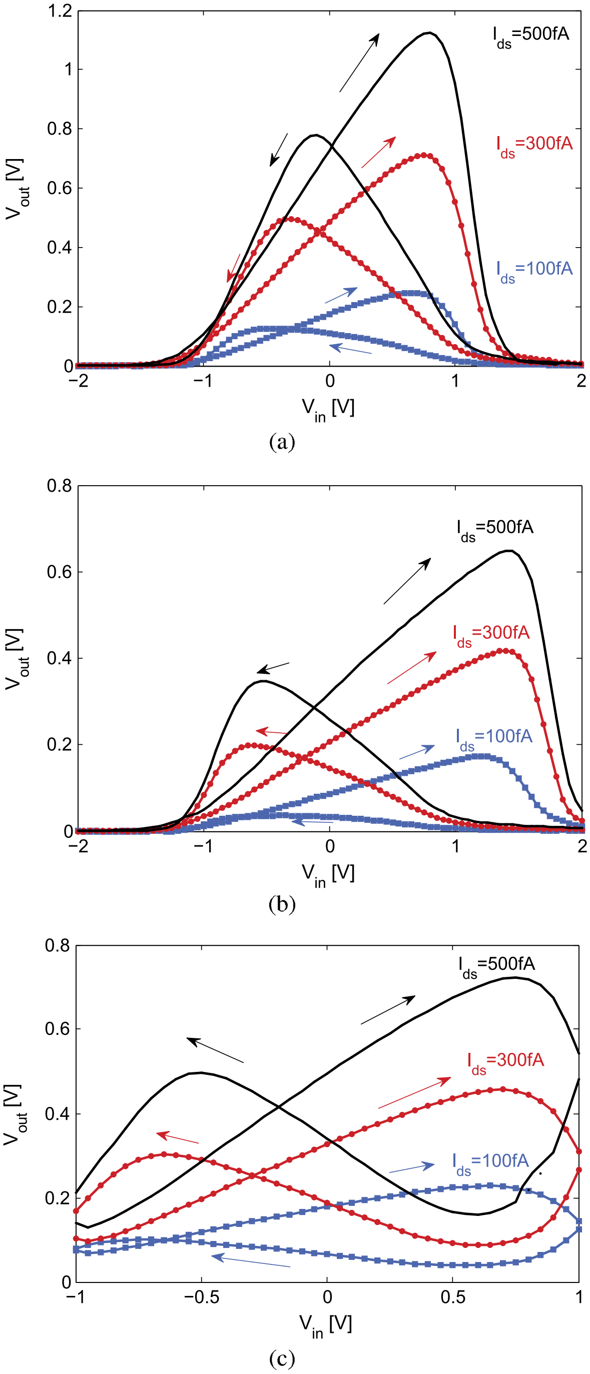
then compared with V_{gs} (Fig. 19). The obtained hysteresis can be used as a latch device, whose position in the V_{out}–V_{in} plane can be adjusted by using a different value of the current bias. A similar behavior has been exploited with three-terminal Schottky-barrier polysilicon nanowire FETs circuits to build a new logic family based on precharge and evaluation scheme [53]. Moreover, a similar scheme has been demonstrated for DRAM type of memory [54] and for pA current and temperature detection [55]. Similarly, SB Si nanowire transistors fabricated with a low termal budget process and biased in current-controlled mode shows a similar hysteresis. Moreover, polycrystalline SiNWs SB FETs can give an hysteretic transfer characteristic (Fig. 20) very similar to the one reported for crystalline SiNW SB FETs fabricated with a low thermal budget process [56]. As it is shown in Fig. 20 the maximum output voltage in the transfer characteristics increases with the I_{ds} bias current. From Figs. 20a–20c the sweep time is reduced. Similarly to what was discussed for the three terminal SB SiNW FET memristive device, the sweeping time impacts on the amount of charge that traps at the gate oxide/channel interface, thus influencing the conductance state of the nanowire channel. A faster sweeping time outbalances the charge trapping/detrapping mechanism, resulting in lower output voltages (see Figs. 20a, 20b compared with Fig. 20c).

3 Applications of Resistive RAMs

3.1 Standalone Memories

Future deeply scaled circuits will see their performances limited by the physical limitations of the materials. To keep pushing the performance of computation and the density of storage, the microelectronics industry envisages using more efficient state variable than the electronic charge. In this sense, the memristor is an attractive candidate for both computation and memory, thanks to its programmable resistive state. When considering the Resistive RAM (ReRAM) memories, which can be classified

Fig. 20 V_{out} voltages for increasing with I_{ds} current bias. In all the figures I_{ds} bias values are 100 fA, 300 fA and 500 fA. Notice the output voltage hysteresis narrows for increasing frequency sweep: (**a**) 24 sec (**b**) 6 sec (**c**) 0.5 sec



(a)

(b)

(c)

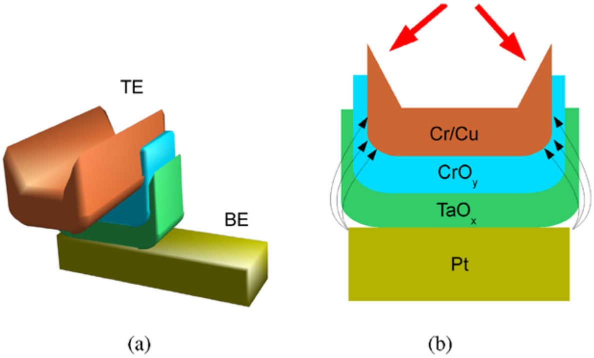


Fig. 21 (**a**) 3D cross-point ReRAM concept with fence-like Top Electrode (TE) shape. (**b**) Lateral cross-section view of $Pt/TaO_x/CrO_y/Cr/Cu$ cross-point ReRAM device. The fence-like TE enables better scalability of the cross-point device thanks to a reduction of the intense electric field distribution at the corners

as memristors, excellent scalability and programming time can be obtained if compared to traditional Flash, which make ReRAMs suitable candidates for standalone memory applications.

In this section, a ReRAM technology demonstrating a *Multi-Level* (*ML*) forming-free $Pt/TaO_x/CrO_y/Cr/Cu$ crossbars built with low thermal budget ≤200 °C is discussed. The devices show excellent scalability down to 2.5×10^9 bit/cm^2 with device half-pitch of 100 nm with projections of practical storage density of up to 10^{12} bit/cm^2 at the 10 nm technology node [57].

3.1.1 Fabrication

The basic device concept is a ReRAM consisting of 2 metal lines crossing orthogonally and a transition metal oxide stack in between (see Fig. 21a). Moreover, the shape of the *Top Electrode* (*TE*) has been fabricated such that corners are smoothed (Fig. 21b). This feature reduces the fringing field intensity at the wire ends, thus improving both reliability and scalability of the devices. Two different ReRAM ($Pt/TaO_x/CrO_y/Cr/Cu$ and $Al/TiO_2/Al$ with the Bottom Electrode/Transition Oxide/Top Electrode order) stacks have been built and compared in terms of performance. Both stacks have forming-free property and as such they do not require special forming steps to form the devices. This is a considerable advantage for actual implementation on chip. Several devices have been built in a passive crossbar array fashion as explained in the following text. Si bulk wafers are first isolated by depositing 100 nm thick Al_2O_3 with *Atomic Layer Deposition* (*ALD*) (Fig. 22a). Then, PMMA bi-layers are patterned with e-beam lithography as lift-off masks for 10 nm/80 nm Pt *Bottom Electrodes* (*BE*) deposition (Fig. 22b). In the next step, a second lift-off mask is defined and 15 nm TaO_x oxide layer (Fig. 22c) is deposited by sputtering from a Ta_2O_5 target with increasing RF power in Ar/O_2 atmosphere (Fig. 22e). Finally, 50 nm/100 nm thick Cr/Cu bi-layers TE are deposited by e-beam evaporation (Fig. 22d). For $Al/TiO_2/Al$ devices, the Al electrodes are deposited with e-beam evaporation while 10 nm thick TiO_2 is deposited by ALD. In Figs. 23a and 23b SEM images of 100 nm wide BE lines and 64 bit passive crossbar are shown, respectively. In Fig. 23c, a 3D reconstructed AFM profile is shown.

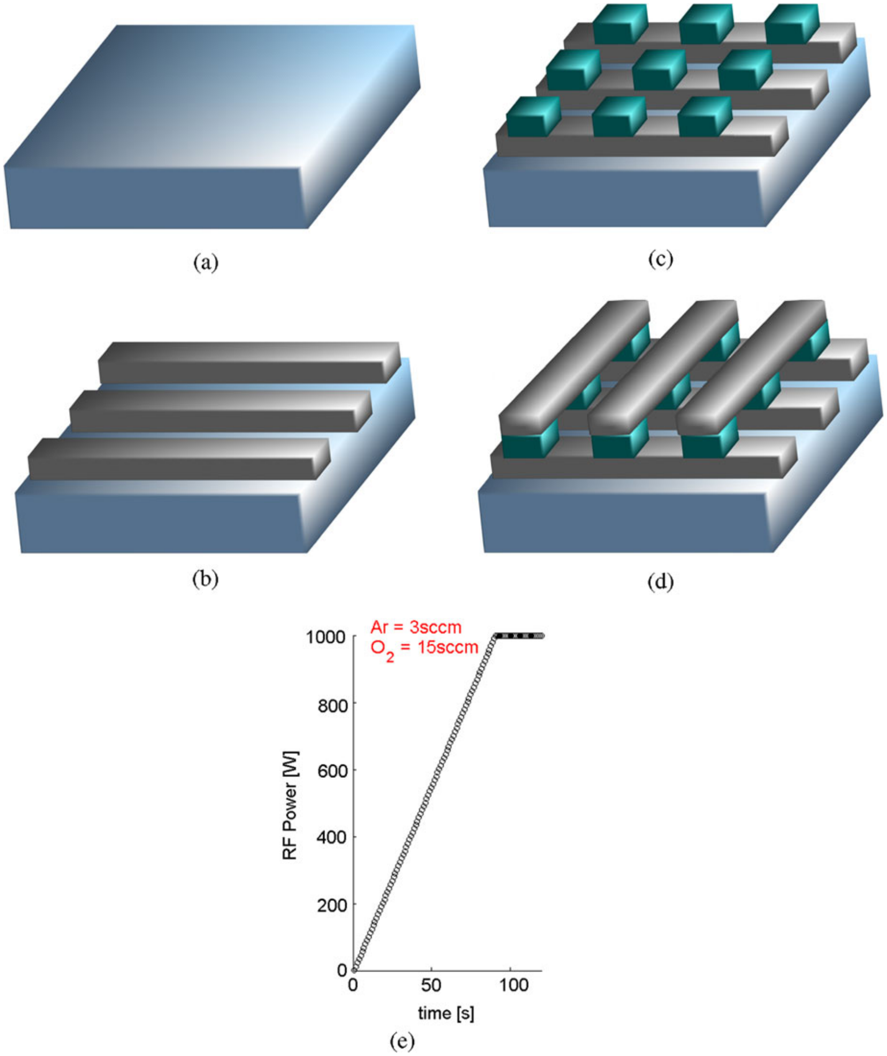


Fig. 22 Process Flow. (**a**) Insulated Si substrate; (**b**) Pt bottom electrode lift-off; (**c**) TaO_x sputtering deposition; (**d**) TE lift-off (**e**) Sputtering parameters of TaO_x, with increasing RF power and constant 3 sccm/15 sccm Ar/O_2 flows

About 250 individual $Pt/TaO_x/CrO_y/Cr$ cross-points for area sizes varying from 100 nm × 100 nm to 1 μm × 1 μm, and 64 bit crossbars with half-pitch varying from 100 nm × 100 nm to 500 nm × 500 nm are built on the same sample following the proposed fabrication steps. In Figs. 24a and 24b, a tilted SEM view, and a 3D reconstructed AFM profile of a cross-point device with fence-like TE are shown, respectively.

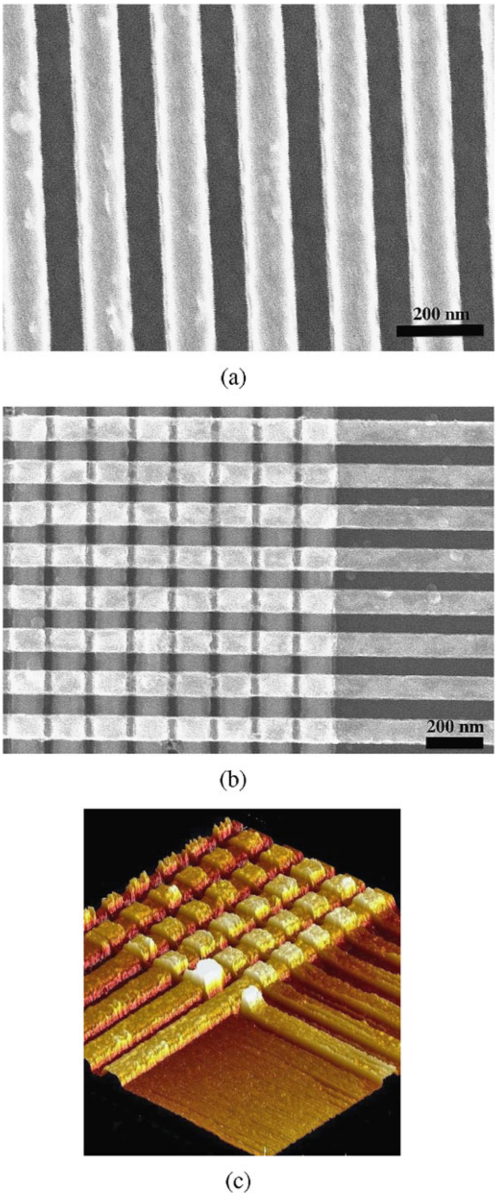


Fig. 23 *Top* SEM view of: (**a**) *bottom* electrode lines; (**b**) complete 64 bit crossbar array of devices with 100 nm half-pitch; (**c**) reconstructed 3D image from AFM profile

3.1.2 Electrical Characterization

Electrical measurements are carried out with an Agilent B1500 semiconductor device analyzer. Pulse mode sweeps with pulses of 500 µs demonstrate forming-free *Bipolar Resistive Switching* (*BRS*) for Pt/TaO_x/CrO_y/Cr/Cu (Fig. 25a). The BRS is obtained for a voltage range of less than 1 V with pristine ON state in

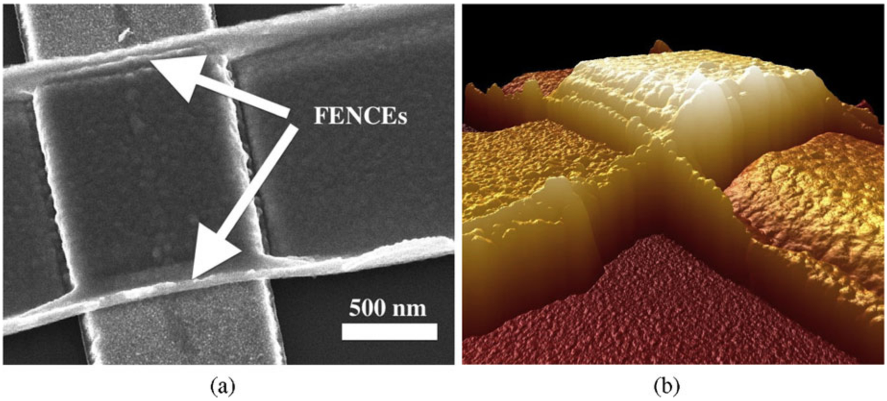


Fig. 24 Pt/TaO$_x$/CrO$_y$/Cr/Cu cross-point device with 900 nm × 900 nm cross-point area. (**a**) Tilted SEM image view. Notice the fence-like structures at the edges of the TE line. (**b**) Reconstructed 3D AFM image of the pristine cross-point device. Average roughness on TE is $\sigma \approx 42.6$ nm

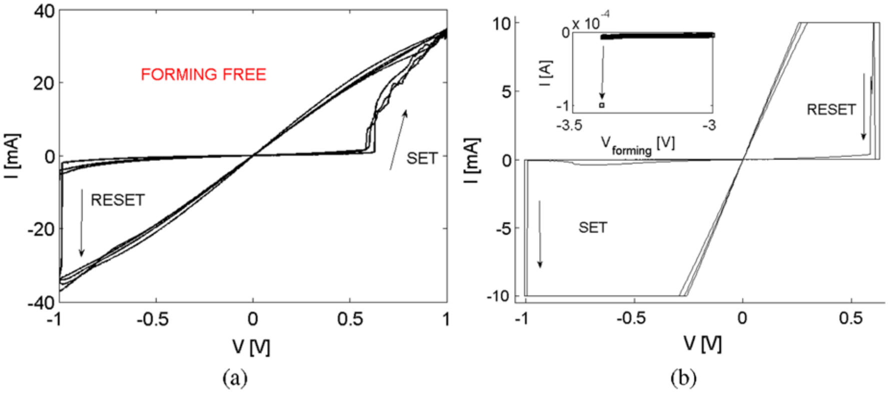
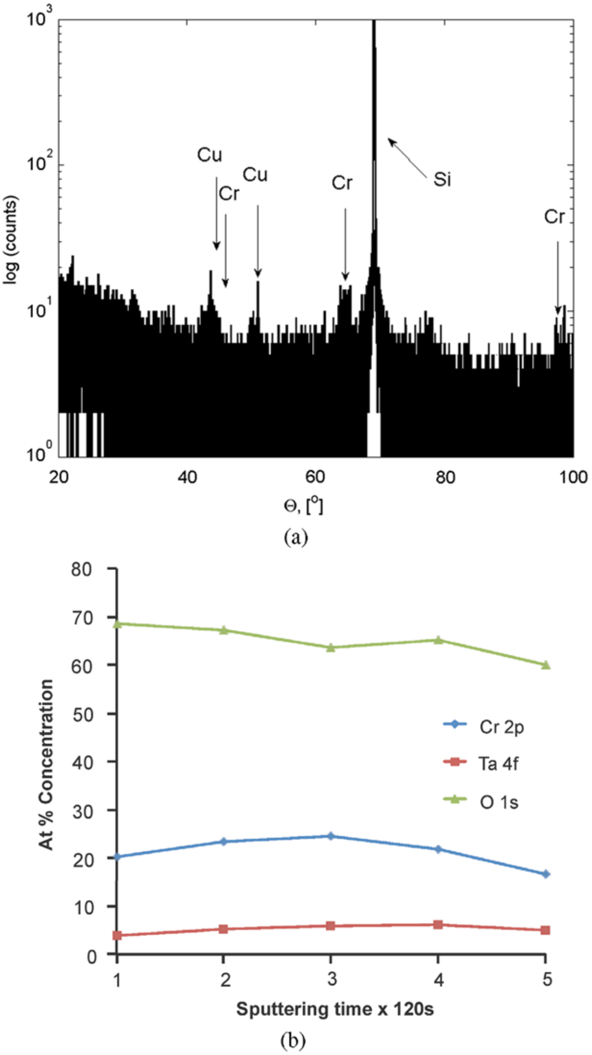


Fig. 25 (**a**) Typical I–V characteristic of the Pt/TaO$_x$/CrO$_y$/Cr/Cu ReRAM cell showing resistance ratio of 10^4. Notice very low $V_{\text{SET}} = +0.8$ V and $V_{\text{RESET}} = -1$ V. After fabrication the devices are forming-free and in the ON state. (**b**) Typical I–V characteristic of the Al/TiO$_2$/Al ReRAM cell after forming with $V_{\text{SET}} = -1$ V and $V_{\text{RESET}} = +0.8$ V. Inset shows $V_{\text{FORMING}} = -3.4$ V

the same range of the *Low Resistance State* (*LRS*). This is an important advantage compared with non forming-free ReRAM devices, because the forming operation requires higher voltages. A forming voltage of −3.4 V has indeed been necessary for the Al/TiO$_2$/Al (Fig. 25b), that then show similar performance as the Pt/TaO$_x$/CrO$_y$/Cr/Cu devices. In Table I structural and electrical parameters of the different cells are reported.

Fig. 26 (**a**) X-ray diffraction pattern on a Si wafer demonstrating amorphous TaO_x/CrO_y. The peaks are related to the presence of Cu and Cr metal layers on top of the TaO_x/CrO_y. The Si peak comes from the substrate and it has been utilized for the calibration of the X-ray diffractometer. (**b**) XPS depth profile analysis showing the presence of both Cr and Ta in oxidized states. Both $Ta_2O_{5-\delta}$ and$TaO_{2-\gamma}$ are present, with more conductive $TaO_{2-\gamma}$ close to the Pt BE and mixture of $TaO_{2-\gamma}$ and CrO_y at the Cr/Cu TE



(a)

(b)

3.1.3 Discussion

Material characterization has been carried out to understand the pristine ON state of the $Pt/TaO_x/CrO_y/Cr/Cu$ device. The X-ray diffraction pattern of Fig. 26a shows peaks from the TE and the Si substrate. The absence of any Ta_2O_5 or TaO_2 peaks indicates that the material is in amorphous state, due to the low deposition temperature. The visible peaks are related to the presence of Cu and Cr metal layers on top of the TaO_x/CrO_y. The Si peak comes from the substrate and it has been utilized for the calibration of the X-ray diffractometer. The pristine ON state excludes the conductive filament mechanism and observing the double logarithmic plot of the I–V curve (Fig. 27a), quasi-Ohmic regimes with slopes ≈1 are obtained for regions far

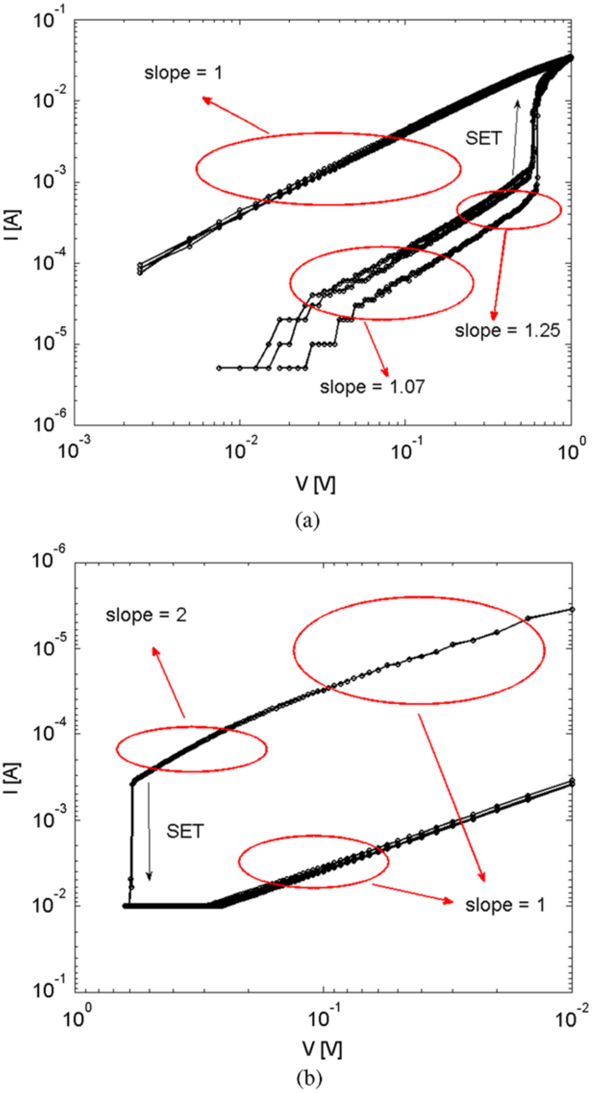
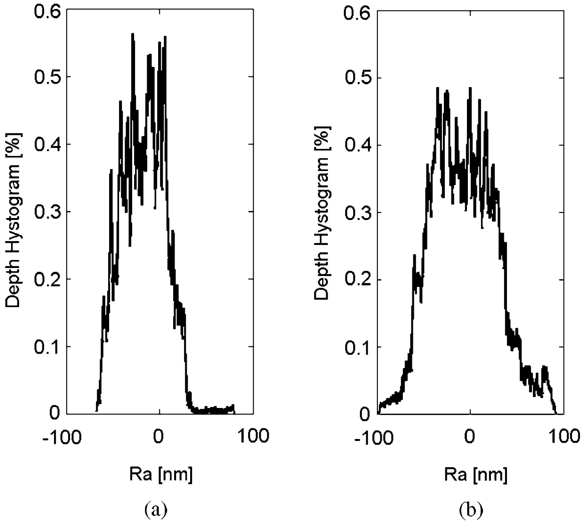


Fig. 27 (**a**) The log(I)–log(V) plot of Pt/TaO$_x$/CrO$_y$/Cr/Cu shows typical trap-controlled conduction of *Space-Charge-Limited-Conduction* (*SCLC*) before SET. In the SET region, the slope is about 25 and it is indication of gradual distribution of defects. (**b**) The log(I)–log(V) plot of the Al/TiO$_2$/Al shows typical trap-controlled conduction of SCLC with quadratic V dependence before SET condition, thus following Child's law dependence. In the SET region, the slope is about 50, and it is indication of an abrupt distribution of defects

from the SET condition. Typical trap-assisted *Space-Charge-Limited-Conduction* (*SCLC*) is observed close to the SET condition. The SCLC conduction is also observed for Al/TiO$_2$/Al devices (Fig. 27b) whose slopes indicate a more abrupt distribution of trap density, which can be related to the different deposition methods. In both devices, the resistive switching mechanism can be attributed to Redox reaction linked with the motion of oxygen-vacancies [58, 59]. Moreover, structural modification is observed from the roughness profile of Pt/TaO$_x$/CrO$_y$/Cr/Cu cross-point after 100 cycles. As shown in Figs. 28a and 28b, the Pt/TaO$_x$/CrO$_y$/Cr/Cu average roughness (Ra) measured above the TE broadens. The Ra changes from a pristine

Fig. 28 Average roughness profiles for TaO_x cross-points: (**a**) $\sigma \approx 42.6$ nm after fabrication; (**b**) $\sigma \approx 56.3$ nm after 100 cycles. The broadening is attributed to the structural change induced by the motion of oxygen-vacancies upon cycling



100 nm variation into a 200 nm broad window, indicating structural modification by the motion of oxygen vacancies upon switching. In addition, XPS-depth analysis (Fig. 26b) confirms that $Ta_2O_{5-\delta}$ and $TaO_{2-\gamma}$ phases are present, with more conductive $TaO_{2-\gamma}$ close to the Pt BE and mixture of $TaO_{2-\gamma}$ and CrO_y at the Cr/Cu TE that is consistent with the Redox switching mechanism.

3.1.4 High Density Multi-valued Crossbars

Several resistance levels of Pt/TaO_x/CrO_y/Cr/Cu devices can be programmed. As shown in Fig. 29a, four levels of resistance (encoding 2 bits) are found within a 4 orders of magnitude range. A larger resistance window of 1 bit is found for Al/TiO_2/Al devices, which show a LRS around 30 Ω and a *High Resistance State* (*HRS*) at 1 MΩ within 2 orders of magnitude variation (see Fig. 29b). The Al/TiO_2/Al show stable LRS and HRS in a large V_{READ} voltage range (Fig. 30a). The Pt/TaO_x/CrO_y/Cr/Cu devices demonstrate excellent scalability, as the HRS/LRS ratio improves for smaller device sizes (Fig. 30b). For instance, 2 bit can be written in a Pt/TaO_x/CrO_y/Cr/Cu by using shorter SET pulses in order to program the cell in one of the stable *Intermediate Resistance* (*IR*) states. An example of 2 bit storage using LRS, HRS and 2 IRs is demonstrated in Fig. 31, each level is separated of about one order of magnitude from each other for various V_{READ}. The devices could be easily assembled into dense 2.5×10^9 bit/cm^2 passive crossbar arrays whose storage density improves to 10^{10} bit/cm^2 thanks to ML capability of Pt/TaO_x/CrO_y/Cr ReRAMs.

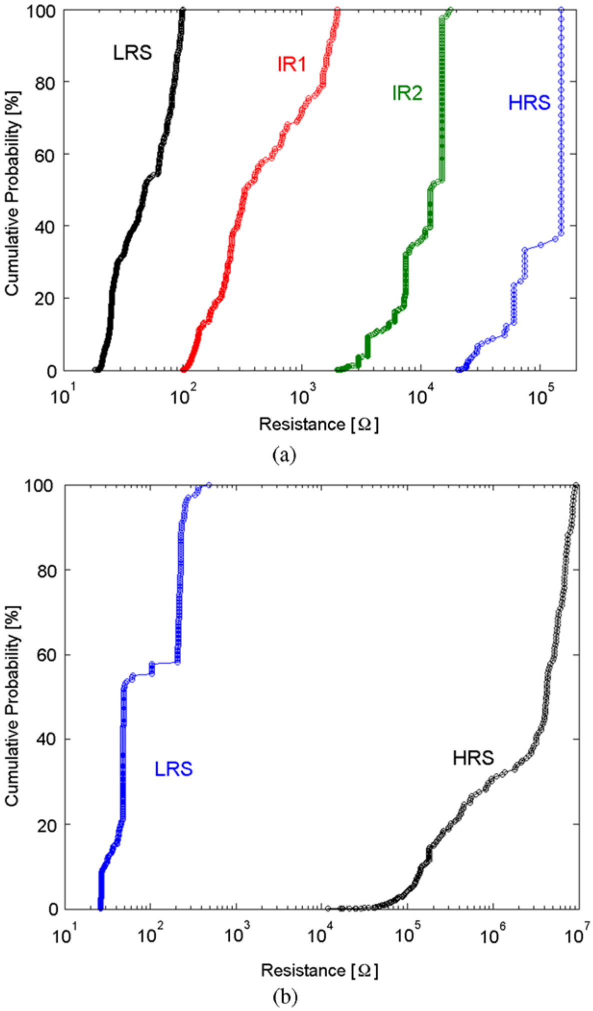


Fig. 29 (**a**) Cumulative probability of *Low Resistance State* (*LRS*), *Intermediate Resistance states 1* (*IR1*) *and 2* (*IR2*) and *High Resistance State* (*HRS*) for Pt/TaO_x/CrO_y/Cr devices are shown. The LRS, IR1 and IR2 are obtained by using SET pulses of 2 ms, 1 ms and 500 μs at 1 V, respectively. The HRS is obtained with a 500 μs RESET pulse at −1 V. (**b**) Cumulative probability of LRS and HRS for Al/TiO_2/Al devices are shown. The LRS and the HRS are obtained by using SET and RESET pulses of 500 μs at −1 V and +1 V, respectively

3.1.5 Summary

Bipolar Resistive Switching Pt/TaO_x/CrO_y/Cr/Cu and Al/TiO_2/Al devices built with thermal budget ≤200 °C have been fabricated and characterized. Very large storage density of TaO_x/CrO_y-based ReRAMs is demonstrated up to 10^{10} bit/cm^2 thanks to the excellent scalability of the fence-like top electrode lines.

3.2 Generic Memory Structure (GMS) for Non-volatile FPGAs

While a lot of research effort targets high density ReRAM-based standalone memories [60], the focus of this work is the usage of ReRAMs for FPGAs. The reason

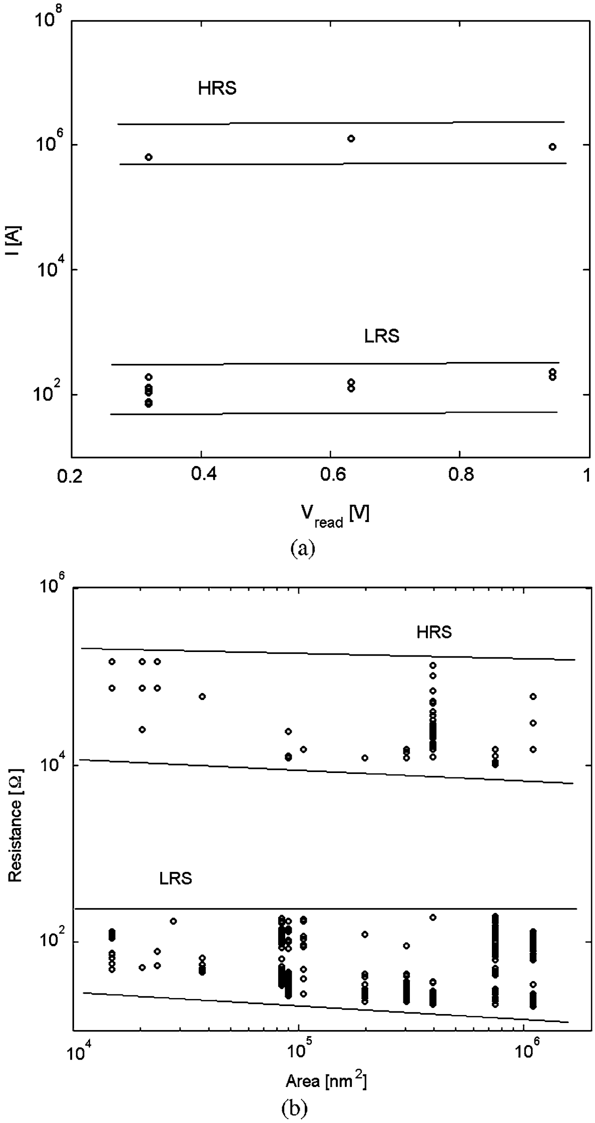


Fig. 30 (**a**) LRS and HRS resistance distributions for increasing V_{READ} of the Al/TiO$_2$/Al cell. The 10^4 resistance ratio is constant over a large range of reading voltage. (**b**) Measured HRS and LRS values for the Pt/TaO$_x$/CrO$_y$/Cr cell devices with different cross-point area demonstrating excellent scalability, indicating local switching at the nanoscale

behind this choice is that in reconfigurable logic, up to 40 % of the area is dedicated to the storage of configuration signals [61]. Traditionally, the configuration data is serially loaded in SRAM cells, distributed throughout the circuit [62]. As a consequence, power up of the circuit is limited by the slow serial configuration. To overpass SRAM volatility and loading time issue, Flash NVM have been proposed [63]. Nevertheless, the use of an hybrid CMOS-Flash technology results in high fabrication costs. Conversely, ReRAMs are fabricated within the *Back-End-of-the-Line* (*BEoL*) metal lines, moving the configuration memory to the top of the

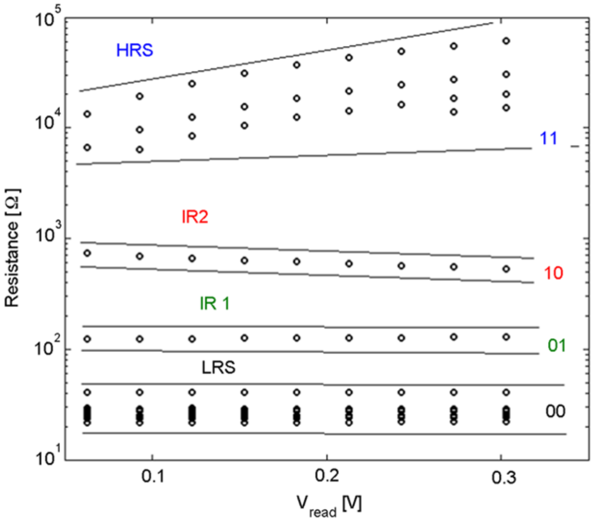


Fig. 31 Resistance distributions for multi-value storages vs. increasing V_{READ} of the Pt/TaO$_x$/CrO$_y$/Cr cell. The HRS, IR2, IR1, LRS are encoded in 2 bit

chip and reducing the area utilization [64]. Similarly, the ReRAM cells can be utilized in combination with TSVs, enabling 3-D stacked FPGA architectures [23]. In this section, we propose a complete proof of concept of an ReRAM-based *Generic Memristive Structure* (*GMS*) circuit for FPGAs from technology development to architectural evaluation. The main idea is to replace the pass-transistors in SRAM-based FPGAs by ReRAMs. Hence, the ReRAMs store the information in their resistive states and can be either used to route signals through low resistive paths, or to isolate them by means of high-resistive paths. Such a functionality is envisaged either to build routing *Multiplexers* (*MUXs*) or configuration nodes. In order to keep the programming complexity in the same range of SRAM-based FPGAs, we propose an efficient methodology based on the inherent GMS complementary programming. The proposed methodology, with GMS-based complementary programming has been validated by electrical measurements on a fabricated device.

3.2.1 BEoL Integration of ReRAMs

As per the previous section, ReRAMs can be fabricated within the BEoL processing. Hence, it is possible to fabricate them between two metal layers (e.g. in between Metal 1 and Metal 2). Because of the BRS of the ReRAMs of this study, depending on the forming polarity, either the Metal 1 or the Metal 2 terminal can be utilized as the positive electrode of the memory, giving two possible configurations (see Fig. 32).

3.2.2 Device Description

In the GMS, two ReRAMs are interconnected as shown in Fig. 33a. The positive terminal of the top memory is connected to the negative terminal of the bottom mem-

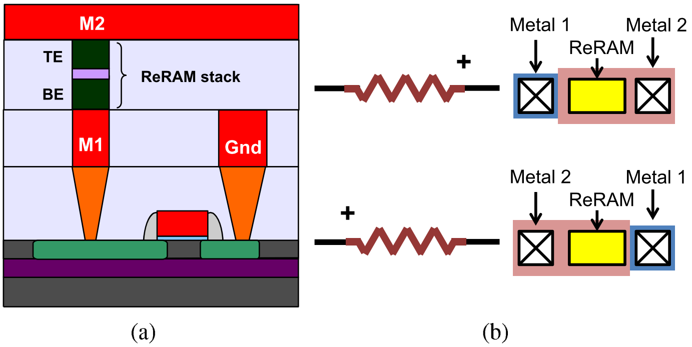


Fig. 32 Cross sectional schematic showing the integration of a ReRAM integrated between the M1 and M2 interconnection levels in the BEoL. (**a**) The *bottom electrode* is thus directly connected to a MOSFET selector (*bottom*) forming a 1-Transistor 1-Resistor (1T1R) memory node. (**b**) ReRAM polarity selection by physical design

ory. This arrangement enables complementary programming of the two ReRAMs composing a GMS. We call the concurrent programming of the GMS a complementary programming operation. A similar programming scheme was previously used for low power crossbars [65]. Figure 33b illustrates the programming of the top path (i.e. left to right arrow in the programming graph shown in Fig. 33d). R_1 and R_2 are switched simultaneously to R_{OFF} and to R_{ON} respectively. This operation is achieved by grounding the common right terminal and biasing the left terminal to Vth (which corresponds to the SET voltage $-V$th for R_1 and to the RESET voltage $+V$th for R_2). The programming of the bottom path (see Fig. 33c) is obtained by inverting the Vth and the GND (which corresponds to the RESET voltage for R_1 and to the SET voltage for R_2). In addition to speed up the programming operation, due to the complementary scheme, only two voltages are needed (GND and Vth). The complementary programming operation has been validated by electrical measurements, while the MUX performances have been extracted by electrical simulations. Figure 34 depicts the resistance values of R_1 and R_2 of an GMS-based MUX21. Resistances are read at $V_{\mathrm{READ}} = +0.1$ V. After a preliminary forming step, R_1 and R_2 are set to R_{ON}. The devices are then read for 10 cycles, showing a stable non-volatile resistance. Hence R_1 and R_2 are switched using the complementary programming operation. During the first writing operation SET and RESET events are induced on R_2 and R_1, respectively (see Fig. 33c) by applying a voltage pulse for 500 µs. After reading the resistance values for another 10 cycles, again validating the non-volatility of the resistance states, a second complementary switching operation is performed. Now the resistance states of R_1 and R_2 are complementary switching, as seen in the reading sequence of Fig. 34. As can be noticed the resistance values of R_1 and R_2 do not exactly match. This is due to the different ReRAM geometries and to the large variability of the cells utilized for the demonstrator. Nevertheless, improved variability of one order of magnitude has been demonstrated for ReRAM prototypes fabricated with industrial methods [66].

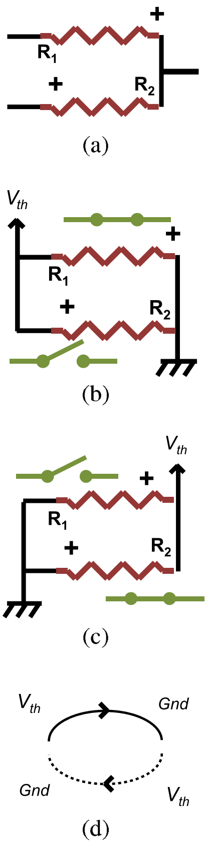


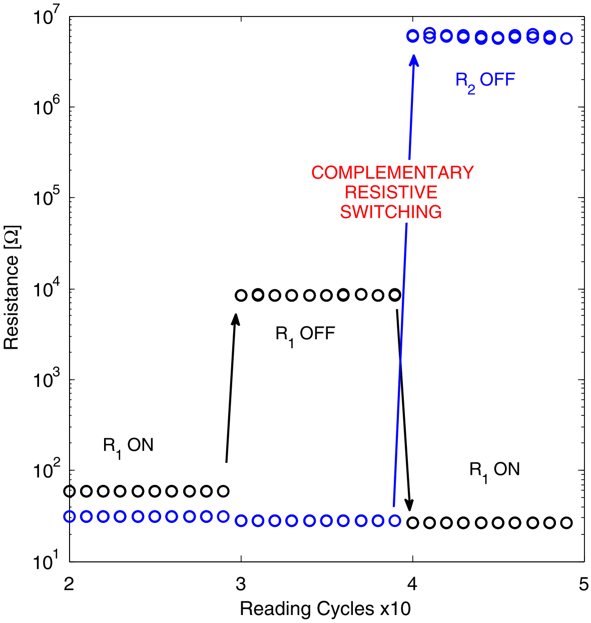
Fig. 33 GMS complementary programming. (**a**) Two BRS resistive RAMs are in series connected after forming. (**b**) When V_{th} is applied, R_1 switches ON, while R_2 switches OFF. (**c**) When $-V_{\text{th}}$ is applied, R_1 switches OFF, while R_2 switches ON. (**d**) Graph representation of the complementary switching operation

3.2.3 GMS Memory Node

In this section, we present an elementary circuit used to move most of the configuration part of reprogrammable circuits to the back-end, reducing their impact on front-end occupancy. Such a memory node is dedicated to drive LUT inputs. The memory node is based on a unique GMS node and provides intrinsically the retained information as a voltage level. Furthermore, it shall allow a layout efficient line sharing.

The basic memory node is presented in Fig. 35c. The circuit consists of 2 ReRAMs connected in a voltage divider configuration between 2 fixed voltage lines (L_A and L_B). The memories are used in a complementary manner, in order to improve reliability. Reliability is required as far as the output is not restored by an inverter for compactness purpose. The output is designed to place a fixed voltage on a conventional standard cell input. Read operations are intrinsic with the structure, while programming is an external operation to perform on the cell.

Fig. 34 Complementary switching operation for the ReRAM-based GMS



3.2.4 Read Operation

A voltage divider is used in this topology to intrinsically realize the conversion from a bit of data stored in the variable resistance to voltage level. Figure 35b presents a configuration example where the node stores a '1'. Voltage lines L_A and L_B are connected to V_{ss} and V_{dd}, respectively. For example, consider that the resistive memory R_1, connected to the V_{dd} line, is configured to the low resistivity state. The other memory R_2, connected to V_{ss}, is in the high resistivity state. As a consequence, a voltage divider is configured and the output node is charged close to the voltage of the branch with a high conductivity.

3.2.5 Write Operation

Figure 35c presents the programming phase of the node. First, the lines L_A and L_B are disconnected from the power lines and connected to the programming signals. The programming signals are chosen according to the GMS programming scheme. Figure 36 presents the programming circuits required to program an array of GMS-based configuration memories. To provide individual access, each GMS has its own selection transistor. Thus, the different lines can be shared in a standalone-memory-type architecture, yielding an efficient layout strategy. The different modes and programming signals are selected by line-driving multiplexers.

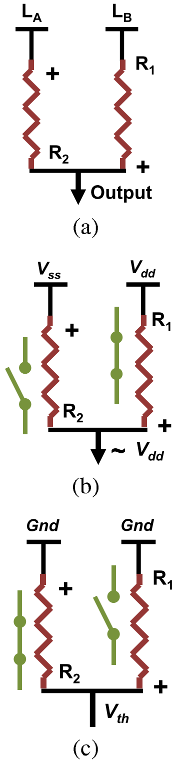


Fig. 35 GMS complementary programming: (**a**) Two in-series connected BRS RAMs forming a voltage divider between 2 metal lines, L_A, L_B. (**b**) Voltage distribution to program a logic 1 into the GMS cell. (**c**) Voltage distribution to program a logic 0 into the GMS cell

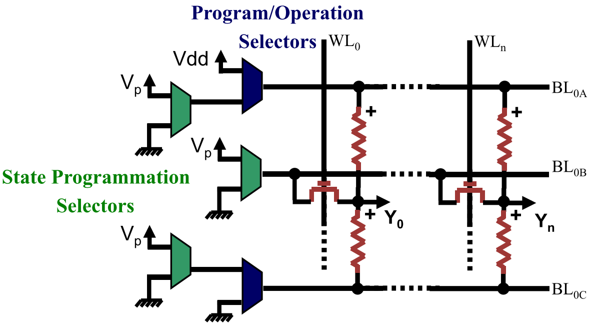


Fig. 36 Complementary switching operation for the ReRAM-based GMS

3.2.6 Summary

This section introduced a novel design, called GMS, based on resistive memories, designed to replace traditional routing resources in reconfigurable logic circuits. Resistive RAM memories combined into a complementary switching GMS cells were used to reduce the footprint and to improve the electrical performances of the data path. The GMS cell can also be used to replace standalone memories, leading to more compact LUTs and steering logic, due to the BEoL integration of ReRAMs.

Thanks to ReRAMs, the area and the delay are reduced by 7 % and 58 % respectively due to the compactness and the low on-resistance of ReRAMs.

3.3 Resistive Programmable TSVs

In this section, BEoL technology for 3D interconnects is addressed, as the interconnect delay is a limiting factor of semiconductor system integration. In this respect, there is a steadily increasing interest in *three-dimensional* (*3D*) wafer/chip stacking solutions utilizing *Through Silicon Vias* (*TSVs*) [1] as well as in reconfigurable interconnect fabrics. The discussed BEoL demonstrate the co-integration of TSVs with ReRAM stacks, offering a new path for re-programmable 3D chip routing. Moreover, the authors report on several device schemes that show different write/erase voltage windows, suggesting a new way for programmable 3D chip interconnects. the fabrication and characterization of titanium dioxide (TiO_2)-based *resistive RAM* (*ReRAM*) co-integration with 380 µm-height Cu *Through Silicon Via* (*TSV*) arrays for programmable 3D interconnects is discussed. Non-volatile resistive switching of $Pt/TiO_2/Pt$ thin films are first characterized with resistance ratio up to 5 orders of magnitude. Then co-integration of $Pt/TiO_2/Pt$ or Pt/TiO_2 memory cells on 140 µm and 60 µm diameter Cu TSV are fabricated. Repeatable non-volatile bipolar switching of the ReRAM cells are demonstrated for different structures.

3.3.1 Planar ReRAM Devices

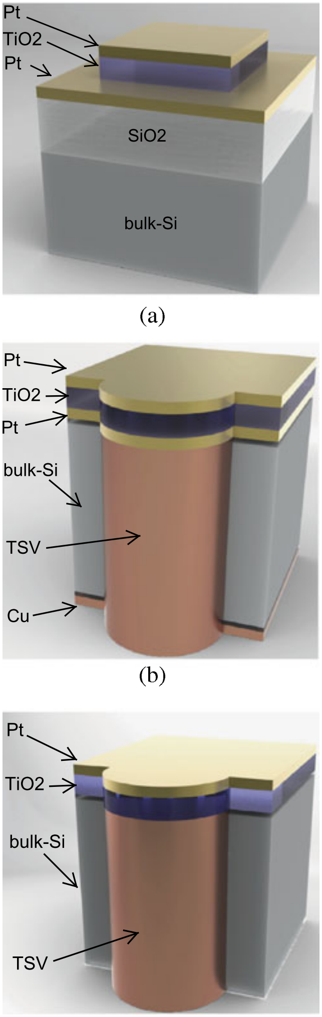
First, high resistivity p-type ($N_A \approx 10^{15}$ atoms/cm^2) bulk-Si wafers are prepared by 500 nm thermal oxidation in H_2O atmosphere. Then the deposition of the resistive switching materials is performed by sputtering of $Pt/TiO_2/Pt$ layers with 270 nm/80 nm/270 nm thicknesses. A conceptual picture is shown in Fig. 37a. The top electrode area of 100 µm by 100 µm squares were patterned by standard lithography and etched by ion milling technique. The etching step reveal the bottom Pt electrode, which can be now accessed for electrical measurements.

3.3.2 TSV Devices

The resistive switching materials were integrated with TSVs producing two different devices:

- 140 µm TSV diameter in 380 µm thick wafer, using $Pt/TiO_2/Pt$ memory stack
- 60 µm TSV diameter in 380 µm thick wafer, using $Cu/TiO_2/Pt$ memory stack. The relatively thin wafer is needed due to TSV aspect ratio limitation.

The TSVs are fabricated using the same process in both cases. A standard optical lithography is used to define the TSV openings. The lithographic step is followed by



(a)

(b)

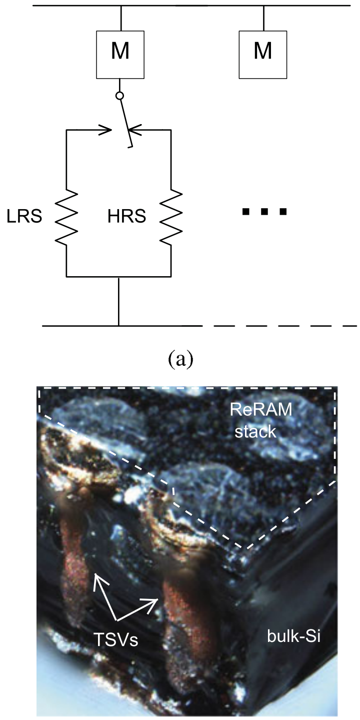
(c)

Fig. 37 (**a**) Concept image of planar ReRAM made of Pt/TiO$_2$/Pt stack. (**b**) Concept image of the ReRAM-TSV using Pt/TiO$_2$/Pt programmable fuse. (**c**) Concept image of the ReRAM-TSV using Cu/TiO$_2$/Pt programmable fuse

through wafer etch, RCA wafer cleaning and thermal oxidation in water atmosphere to grow a 3 μm thick oxide. A 750 nm thick Cu layer is sputtered on the backside of the wafer and the TSVs are filled using Cu electroplating. At this step the seed layer remains on the back of the wafer and the TSV create a positive topography on the front side.

3.3.3 TSV with Pt/TiO$_2$/Pt ReRAM

For the first type of devices, once the TSVs are fabricated the front side of the wafer is processed with *Chemical Mechanical Polishing* (*CMP*) technique



(a)

(b)

Fig. 38 (**a**) Equivalent electrical schematic of the TSV with ReRAM memory elements (denoted by the switch and the "ideal" memory element M). (**b**) Reconstructed 3D photograph of the TSV–Cu/TiO_2/Pt device stack. The die is cleaved to reveal the TSV and the ReRAM stack deposited on *top*

to form a flat surface. The Pt/TiO_2/Pt stack is sputtered with layer thicknesses 270 nm/80 nm/270 nm accordingly. A concept picture of the fabricated structure can be seen in Fig. 37b.

3.3.4 TSV with Cu/TiO_2/Pt ReRAM

For the second type of devices the wafer is polished using CMP on both sides to remove the seed layer and to planarize the surfaces. The Cu was then cleaned using a $NH_4 : H_2SO_4$ etching solution at room temperature for 10 minutes. Then the wafer was loaded into a vacuumed sputtering chamber and a TiO_2/Pt layer was deposited with thicknesses of 80 nm and 270 nm respectively. The Cu of the TSV is acting as the bottom electrode of the ReRAM (see Fig. 37c). Equivalent electrical schematics and the photograph of the devices in a cleaved substrate are shown in Figs. 38a and 38b, respectively.

3.3.5 Electrical Characterization

Electrical measurements were carried out with an HP4156A semiconductor parameter analyzer and cascade probe station in dark conditions. For electrical contacts, standard tungsten needles with 15 μm apex diameter were placed on the top electrode area very softly, since a dependence of the switching with needle pressure has been observed, similarly to the observation of local pressure modulated conductance with AFM tips [21]. Then double I–V DC sweeps have been used to investigate the resistive switching behavior. In all the cases, bipolar switching mechanism with different write/erase window and resistance states has been observed. The measured electrical parameters are summarized in Table 1.

3.3.6 Planar ReRAM Devices

First the planar Pt/TiO_2/Pt are characterized and it showed stable and repeatable bipolar switching behavior between 10 Ω and 1 MΩ read or measured at +1 V (see Fig. 39a). Originally the devices are in the high resistance state (HRS). By sweeping from negative to positive voltages the devices hold the HRS until a SET transition to a low resistance state (LRS) occurs at +1.8 V. After the SET event, the voltage sweep continues until +2 V and then move backward toward negative voltage region. When −1.3 V is reached, the device is RESET to the original HRS state. An HRS to LRS ratio of about 5 orders of magnitude is read at +0.5 V.

3.3.7 TSV–Pt/TiO_2/Pt Devices

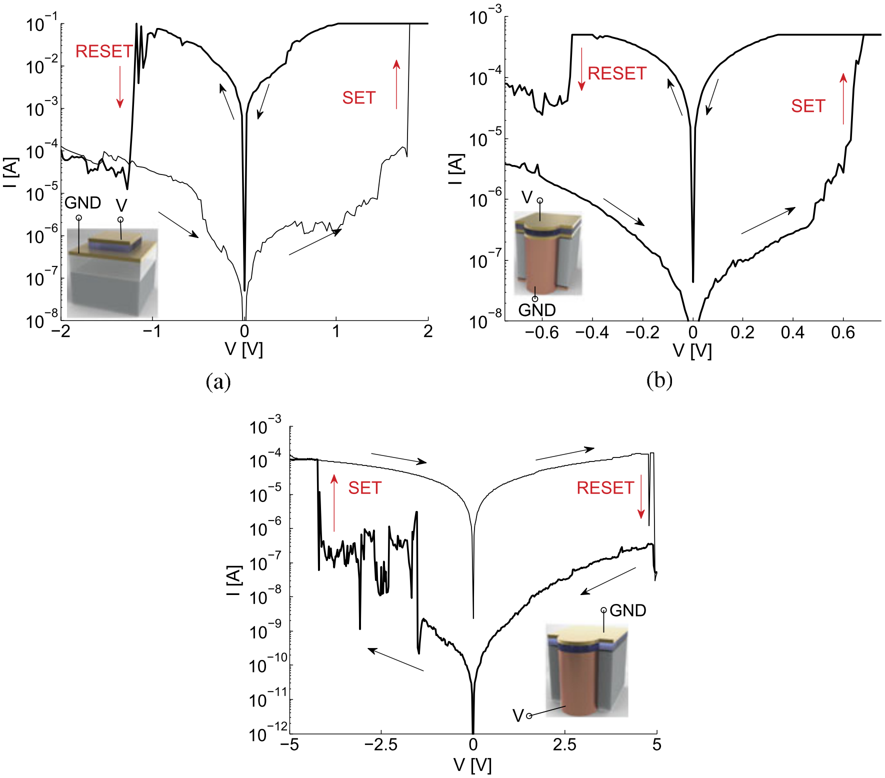
Next, TSV–Pt/TiO_2/Pt with the same layer thicknesses are measured, showing resistance switching below ± 1V (see Fig. 39b). This voltage reduction is attributed to a larger surface roughness of the films deposited on the TSVs, which would lead to a denser electric field at the hillocks as well as to surface states acting as dopants for the TiO_2 [67]. Similar to the planar ReRAM case, the devices are originally in the HRS, and bipolar resistive switching is obtained. Nevertheless, the SET condition is found to be only +0.6 V, while the RESET voltage is measured at −0.5 V. Using a reading voltage of +0.2 V, an HRS of 2 MΩ and a LRS of 666 Ω, with resistance ratio of 3000 are measured.

3.3.8 TSV–Cu/TiO_2/Pt Devices

Since the programming voltages also depend on the current density that can flow into the switching element, a different approach that limits the current flux is investigated. As the electrode material influences the Schottky barrier contact with the TiO_2 layer [68, 69], that is a n-type semiconductor, an alternative device is obtained by depositing TiO_2 and Pt directly on top of the Cu–TSV. Thus, thanks to a larger Schottky barrier height at the Cu–TiO_2 interface, a larger programming window is obtained (Fig. 39c). The SET and RESET voltage positions are now reversed with respect to the other devices, as the Cu has been used as the top electrode. A HRS of 500 MΩ and LRS of 5 kΩ are read at +1 V.

Table 1 Obtained ReRAM electrical parameters for the different devices

Device	V_{SET}	V_{RESET}	HRS	LRS	R_{RATIO}	V_{READ}
Planar Pt/TiO_2/Pt	+1.8 V	−1.3 V	10 MΩ	10 Ω	10^5	+1 V
TSV–Pt/TiO_2/Pt	+0.6 V	−0.5 V	2 MΩ	666 Ω	3003	+0.2 V
TSV–Cu/TiO_2/Pt	−4.2 V	+5 V	500 MΩ	5 Ω	10^5	+1 V



(a)

(b)

(c)

Fig. 39 (**a**) Resistive switching through I–V sweeps for planar Pt/TiO_2/Pt. (**b**) Resistive switching through I–V sweeps using TSV–Pt/TiO_2/Pt programmable fuse. (**c**) Resistive switching through I–V sweeps using TSV–Cu/TiO_2/Pt programmable fuse

3.3.9 Summary

In this study, Pt/TiO_2/Pt obtained by standard sputtering techniques on oxidized Si wafers showed stable bipolar resistive switching without the need of a forming step and with LRS to HRS resistance ratio up to 5 orders of magnitude. The device is successfully integrated on top of 140 μm and 60 μm TSV arrays either in the full Pt/TiO_2/Pt stack or using the Cu as the top electrode, demonstrating different write/erase voltage windows. The co-integration of ReRAM stacks with TSVs is envisaged as a new and compact solution for programmable/reconfigurable 3D chip interconnects.

4 Applications of Multi-terminal Memristive Devices

Multi-terminal memristive devices can be exploited by their additional functionality. For instance, the amplification of the filament formation in the atomic switch [40] is used to improve writing time and to reduce power consumption during switching phases. Recently, the authors demonstrated the use of three-terminal memristive Si nanowires for bio-molecule detection in dry environment [38]. More specifically, in [38] the third terminal is represented by an organic functionalization layer that wraps the Si nanowire all-around. Another example can be the use of a four-terminal GAA SB Si nanowire FETs for low current and temperature sensing, as demonstrated by the authors [55]. Regarding logic/memory applications, the integration of a three-terminal memristive device realized with Schottky-barrier polysilicon nanowire FETs demonstrated the concept of using this devices for new logic families and hybrid logic/memory gates [53]. For instance in [53], the three-terminal configuration can be used to compute basic digital functions, such as NAND, NOR and flip-flop by using a precharge-evaluation phase scheme. Another application for the three-terminal SB polysilicon nanowire transistors can be the design of a circuit cell reproducing an hysteretical negative differential resistance [54]. In thin-polysilicon grain SB FETs, the hysteresis can arise from the granularity of the channel. In Sect. 2.4, it is shown that a similar hysteresis can also be obtained independently on the phase state of the Si nanowire channel. Similarly, the same structure can exploit the functionality of an additional gate to tune the polarity of a SB SiNW FET, thus giving even more functionality (as discussed in Sect. 2.4). A very high expressive power architecture can be made of four-terminal memristive devices arranged in a crossbar implementation that exploits the high-density of the SiNW arrays.

4.1 Neuromorphic Circuits

The non-volatile property of the two-terminal memristive devices has a tremendous potential for neuromorphic circuits, in particular forming artificial synapses following the Hebbian rule of learning based on *Spike-Rate Dependent Plasticity* (*SRDP*) as well as new building elements for hybrid CMOS/memristor circuits. For instance, when considering the perceptron model of the neuron (see Fig. 40), the weighted connections of the inputs to the summation element can be modeled with the properties of non-volatile memristive devices.

With this respect, the Hudgkin-Huxley model, can be mathematically described by first order differential equations. More specifically, Chua and Kang demonstrated

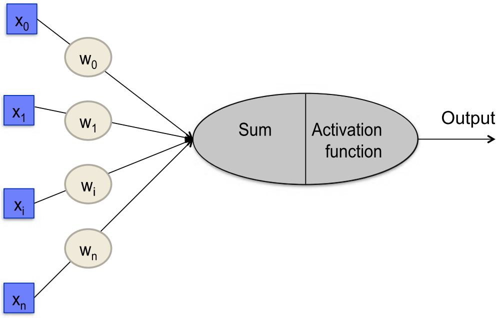


Fig. 40 Perceptron model of a neuron. Input signals x_i are weighted by programming each weight w_i and then summed. The output signal will be an activation function that depends on the summation

that the H–H model of the potassium channel can be identified as a first-order time invariant voltage-controlled memristive one-port and that the sodium channel can be described as a second-order time invariant voltage-controlled memristive one-port. Since this representation is compatible with the mathematical representation of memristive devices, it is noteworthy to notice that memristive devices based circuits can be built to emulate the behavior of biological systems, in this particular case emulating the potassium and sodium channels of the neurons. One example is an energy-efficient memristor-based integrate and fire neuron circuit which exploit the bistability of a ReRAM to model both the short time spike event and the refractory period [70]. Another example is the use of the analog programmability of the ReRAM devices that can be used to emulate the weighted connections of the perceptron model.

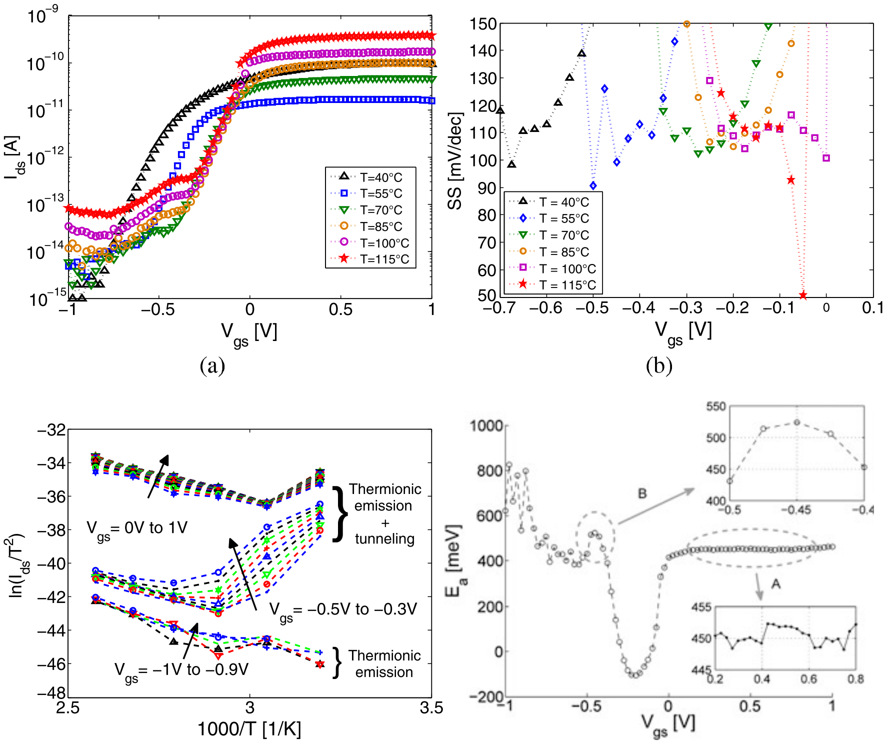
4.2 Current and Temperature Sensor

This section reports on the fabrication and characterization of a pA current and temperature sensing device with ultra-low power consumption based on a Schottky barrier silicon nanowire transistor. Thermionic and trap-assisted tunneling current conduction mechanisms are identified and discussed on the base of the device sensitivity upon current and temperature biasing. In particular, very low current sensing properties are confirmed also with previously reported polysilicon-channel nanowire Schottky barrier transistors. demonstrating that these devices are suitable for temperature and current sensing applications. Moreover, the process flow compatibility for both sensing and logic applications makes these devices suitable for heterogeneous integration. A range of device operation conditions are investigated, showing how an ambipolar device can be used for different applications, the only requirement being the biasing condition.

The I_{ds}–V_{gs} dependence with T is mainly attributed to the I_{th}, however T also influences the Itunnel since hotter carriers pass through a narrower Schottky barrier, leading to an increasing current level [51]. The I_{OFF} current is increasing exponentially with temperature and its main contribution is a thermionic emission component. A different behavior has been observed for the I_{ON} current. Increasing the temperature makes the I_{ON} current to decrease until the temperature reaches 55 °C and then it rise exponentially with linear increase of T. At lower temperatures tunneling and trap-assisted tunneling are more important than thermionic emission. Rising T up to 70 °C makes the charges trapped into the gate oxide to un-trap, reducing the I_{tunnel} component. A different behavior is observed for the I_{ON} currents for $70\ ^\circ\mathrm{C} \leq T \leq 115\ ^\circ\mathrm{C}$. In this range, the I_{ON} exponentially increases with T. This effect is evidence of two main current components, for which the I_{ON} changes from a tunneling to a thermionic emission dominated regime. A set of I_{ds}–V_{gs} curves (Fig. 41a) taken at different temperatures at constant $V_{\mathrm{ds}} = 100$ mV and $V_{\mathrm{bg}} = 5$ V are used to extrapolate the Arrhenius plot (Fig. 41c). The constant $V_{\mathrm{bg}} = 5$ V is used to set the device operation more favorable for electron conductance at low V_{gs}. Constant subthreshold swings ≈ 110 mV/dec are observed independently from the temperature (see Fig. 41b). Low negative V_{gs} voltages ranging from -1 V to 0 V show an almost linear slope with inverse of temperature and can be correlated to a thermionic-emission regime. However, for this V_{gs} range the current level is on the order of fAs, which is comparable to the background noise, and it cannot be used to extrapolate the Schottky barrier height. Another distinct regime is observed for$-0.3\ \mathrm{V} \leq V_{\mathrm{gs}} \leq -0.5$ V, for which the slopes are greatly affected by tunneling. This regime shows a dominant tunneling component for the two lowest temperatures. Finally, an exponential dependence with T is observed again for $V_{\mathrm{gs}} \geq 0$ V with the exception of the lower temperature. All these regimes demonstrate that the current in our device is mainly thermionic for $\geq$70 °C and that the tunneling contribution is trap assisted. The slopes from the Arrhenius plot are then used to extract the effective Schottky barrier height ϕ_{Beff} with the activation energy E_{a} method. As shown in inset A of Fig. 41d, an average effective barrier height $E_{\mathrm{a}} \approx 450 \pm 5$ meV is found over a large range of $V_{\mathrm{gs}} \geq 0.2$ V. However, these values cannot be taken as Schottky barrier height since in this regime the device has both tunneling and thermionic components. As suggested by Svensson et al. [71], a better evaluation of the Schottky barrier height can be taken at the maximum of Ea for low current levels. As shown in the inset B of Fig. 41d, this maximum corresponds to $V_{\mathrm{gs}} = -0.45$ V and gives a $\phi_{\mathrm{Beff}} = 525$ meV, confirming the mid-gap Schottky barrier height.

4.2.1 Current Sensing

Current biasing the devices with a constant Ids current makes the device to behave as a pseudo-inverter configuration with hysteretic transfer function. Thanks to the ambipolarity, the Vout-Vin curves shift linearly with the applied current bias. For instance in Fig. 42a, low pA current levels can be either read from the high-to-low or the low-to-high transition voltage with sensitivities of 17 mV/pA. A similar biasing scheme for polysilicon nanowires has been previously characterized by the authors show a similar trend. In Fig. 42b, forward and reverse threshold voltages for currents between 100 fA and 500 fA show a linear increase with current (adapted from [72]).



(a)

(b)

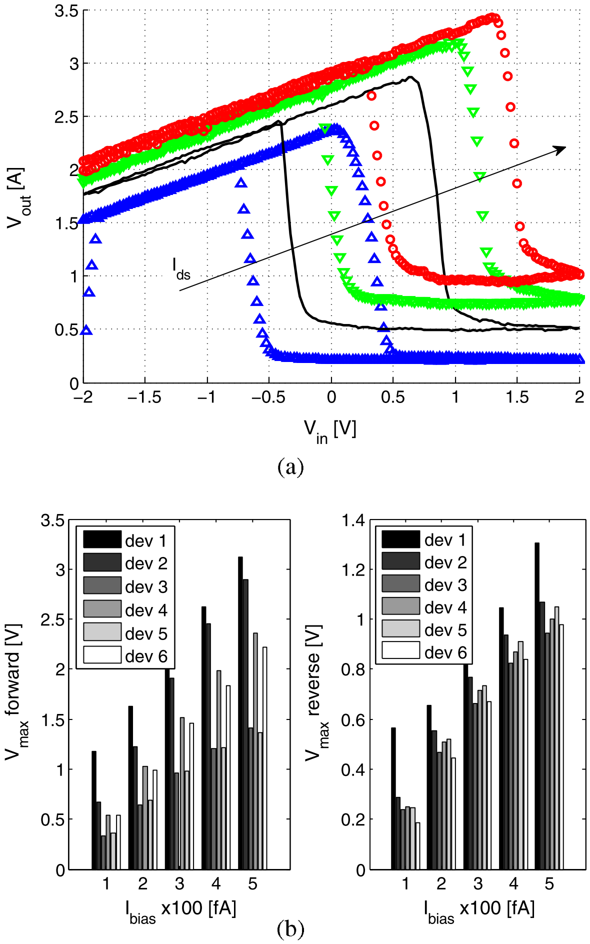
(c)

(d)

Fig. 41 (**a**) Effect of the temperature on the I_{ds}–V_{gs} at $V_{ds} = 100$ mV (**b**) Subthreshold swings associated with the I_{ds}–V_{gs}. Very low swing minima are measured at 100 °C and 115 °C close to threshold voltages. Notice the voltage shift with temperature increase and the extremely low minima of 40 mV/dec for the highest temperature. (**c**) Arrhenius plot for different V_{gs} values showing both thermionic emission and tunneling mechanisms. The linear decreasing slopes are associated with thermionic emission regimes. (**d**) Extracted E_a over a large range of V_{gs}. Inset A shows constant $E_a \approx 450 \pm 5$ meV. Inset B shows a maximum at 525 meV which is taken as the value of the effective Schottky barrier height

4.2.2 Temperature Sensing

Another application is temperature sensing. Upon application of increasing temperature of operation, the hysteresis window observed in pseudo-inverter biasing scheme shrinks. The crystalline Si nanowire Schottky barrier FET shows different sensitivities at different temperature regimes, depending on which mechanism dominates the conductance. Since the hysteresis is attributed to the storage of charges in either gate oxide and/or at the Schottky barrier junctions [49], an increased hysteresis window is expected for the lowest temperatures. The highest sensitivity of 40 mV/°C is found in the T range around 40 °C at which the trap tunneling mechanisms dominates. For temperatures higher than 55 °C the sensitivity tends to saturate according to the dominance of thermionic current contribution, leading to lower



(a)

(b)

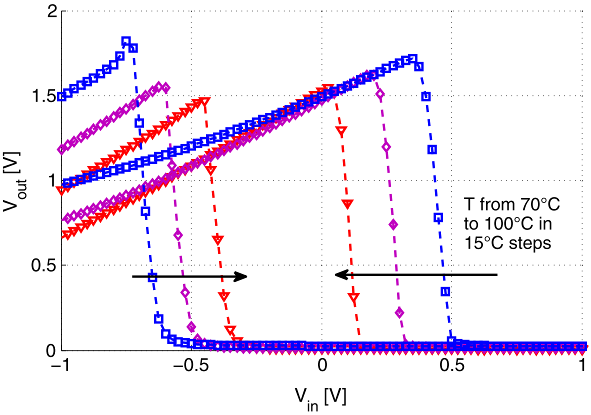
Fig. 42 (**a**) Measured input-output transfer characteristics of a hysteretic inverter based on a single Si nanowire FET with low current bias, showing current-dependent thresholds. (**b**) Forward and reverse threshold voltages for polysilicon Schottky barrier FETs under constant current biasing from 100 fA up to 500 fA (adapted from Ref. [72])

sensitivity of 10 mV/°C. In Fig. 43 the hysteresis window shrinks for increasing T when 70 °C ≤ T ≤ 100 °C.

5 Conclusions

A general overview on multi-terminal memristive devices is reported. The functionality of the devices can be used for logic, memory and sensing applications. Ultra-dense memristive ReRAMs crossbar arrays can be used for ultra-dense non-volatile memory storage. It was shown that three- and four-terminal memristive devices can be used for both logic and memory applications. In particular, Schottky-barrier silicon nanowire FETs are very interesting devices due to their CMOS-compatibility and ease of fabrication. Disruptive applications exploiting the high expressive power

Fig. 43 The hysteresis window shrinks with increasing temperature. Within this T range, the temperature sensitivity of 10 mV/°C is related with the thermionic current regime



of four-terminal memristive devices arranged in crossbar arrays are foreseen as a significant advance in the electronic computation.

Acknowledgements The authors would like to thank Dr. Michael Zervas as well as Dr. Sandro Carrara for useful discussions. Moreover, the authors thank the CMI staff of EPFL for help with the fabrication. This work has been partially supported by the Swiss NSF Grant No 200021-122168, Swiss NSF Grant No 200021-132539, Nano-Tera Grant No 20NA21-128841, Nano-Tera Grant No 20NA21-128840 and grant ERC-2009-AdG-246810.

References

1. http://www.itrs.net
2. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: Nature (2008)
3. Green, J.E., Wook Choi, J., Boukai, A., Bunimovich, Y., Johnston-Halperin, E., Deionno, E., Luo, Y., Sheriff, B.A., Xu, K., Shik Shin, Y., Tseng, H., Stoddart, J.F., Heath, J.R.: Nature **445**, 414 (2007)
4. Chua, L.: IEEE Trans. Circuit Theory **18**(5), 507 (1971)
5. Chua, L., Kang, S.M.: Proc. IEEE **64**(2), 209 (1976)
6. Borghetti, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R., Williams, R.S.: Nature **464**, 873 (2010)
7. Wang, F.Z., Helian, N., Wu, S., Lim, M.G., Guo, Y., Parker, M.A.: IEEE Electron Device Lett. **31**(7), 755 (2010)
8. Pershin, Y.V., Di Ventra, M.: Adv. Phys. **60**(2), 145 (2011)
9. Valov, I., Linn, E., Tappertzhofen, S., Schmelzer, S., van den Hurk, J., Lentz, F., Waser, R.: ArXiv e-prints (2013)
10. Prodromakis, T., Toumazou, C., Chua, L.: Nat. Mater. **11**(6), 478 (2012)
11. Waser, R.: In: Electron Devices Meeting, 2008. IEDM 2008. IEEE International, pp. 1–4. IEEE Press, New York (2008)
12. Yao, J., Sun, Z., Zhong, L., Natelson, D., Tour, J.M.: Nano Lett. **10**(10), 4105 (2010)
13. Kim, Y., Lee, J.: J. Appl. Phys. **104**(11), 114115 (2008)
14. Dong, R., Lee, D.S., Xiang, W.F., Oh, S.J., Seong, D.J., Heo, S.H., Choi, H.J., Kwon, M.J., Seo, S.N., Pyun, M.B., Hasan, M., Hwang, H.: Appl. Phys. Lett. **90**(4), 042107 (2007)
15. Tsunoda, K., Kinoshita, K., Noshiro, H., Yamazaki, Y., Iizuka, T., Ito, Y., Takahashi, A., Okano, A., Sato, Y., Fukano, T., Aoki, M., Sugiyama, Y.: In: Electron Devices Meeting, 2007. IEDM 2007. IEEE International, pp. 767–770 (2007)

16. Chang, W.Y., Lai, Y.C., Wu, T.B., Wang, S.F., Chen, F., Tsai, M.J.: Appl. Phys. Lett. **92**(2), 022110 (2008)
17. Zhu, W., Chen, T., Liu, Y., Yang, M., Zhang, S., Zhang, W., Fung, S.: IEEE Trans. Electron Devices **56**(9), 2060 (2009)
18. Driscoll, T., Kim, H.T., Chae, B.G., Ventra, M.D., Basov, D.N.: Appl. Phys. Lett. **95**(4), 043503 (2009)
19. Blanc, J., Staebler, D.L.: Phys. Rev. B **4**(10), 3548 (1971)
20. Waser, R., Aono, M.: Nat. Mater. **6**, 833 (2007)
21. Yang, J.J., Pickett, M.D., Li, X., Ohlberg, A.A., Stewart, D.R., Williams, R.S.: Nat. Nanotechnol. **3**(7), 429 (2008)
22. Kugeler, C., Nauenheim, C., Meier, M., Rudiger, A., Waser, R.: In: Non-volatile Memory Technology Symposium, 2008. NVMTS 2008. 9th Annual, pp. 1–6 (2008)
23. Sacchetto, D., Zervas, M., Temiz, Y., De Micheli, G., Leblebici, Y.: IEEE Trans. Nanotechnol. **PP**(99), 1 (2011)
24. Lankhorst, M.H.R., Ketelaars, B.W.S.M.M., Wolters, R.A.M.: Nat. Mater. **4**, 347 (2005)
25. Cywar, A., Bakan, G., Boztug, C., Silva, H., Gokirmak, A.: Appl. Phys. Lett. **94**(7), 072111 (2009)
26. Stewart, D.R., Ohlberg, D.A.A., Beck, P.A., Chen, Y., Williams, R.S., Jeppesen, J.O., Nielsen, K.A., Stoddart, J.F.: Nano Lett. **4**(1), 133 (2004)
27. Pease, A.R., Jeppesen, J.O., Stoddart, J.F., Luo, Y., Collier, C.P., Heath, J.R.: Acc. Chem. Res. **34**(6), 433 (2001)
28. Lau, C.N., Stewart, D.R., Williams, R.S., Bockrath, M.: Nano Lett. **4**(4), 569 (2004)
29. Wang, X., Chen, Y., Xi, H., Li, H., Dimitrov, D.: IEEE Electron Device Lett. **30**(3), 294 (2009)
30. Chanthbouala, A., Matsumoto, R., Grollier, J., Cros, V., Anane, A., Fert, A., Khvalkovskiy, A.V., Zvezdin, K.A., Nishimura, K., Nagamine, Y., Maehara, H., Tsunekawa, K., Fukushima, A., Yuasa, S.: Nat. Phys. **7**, 626 (2011)
31. Pershin, Y.V., Di Ventra, M.: Phys. Rev. B **78**(11), 113309 (2008)
32. Martinez-Rincon, J., Pershin, Y.: IEEE Trans. Electron Devices **58**(6), 1809 (2011)
33. Arkan, E.F., Sacchetto, D., Yildiz, I., Leblebici, Y., Alaca, B.E.: J. Micromech. Microeng. **21**(12), 125018 (2011)
34. Martinez-Rincon, J., Di Ventra, M., Pershin, Y.V.: Phys. Rev. B **81**, 195430 (2010)
35. Sun, J., Lind, E., Maximov, I., Xu, H.: IEEE Electron Device Lett. **32**(2), 131 (2011)
36. Chang, S., Sivoththaman, S.: IEEE Electron Device Lett. **27**(11), 905 (2006)
37. Lou, J., Reed, D., Liu, M., Sun, N.: Appl. Phys. Lett. **94**(11), 112508 (2009)
38. Sacchetto, D., Doucey, M.A., De Micheli, G., Leblebici, Y., Carrara, S.: BioNanoScience **1**, 1 (2011)
39. Berzina, T., Erokhina, S., Camorani, P., Konovalov, O., Erokhin, V., Fontana, M.: ACS Appl. Mater Interfaces (2009)
40. Sakamoto, T., Banno, N., Iguchi, N., Kawaura, H., Kaeriyama, S., Mizuno, M., Terabe, K., Hasegawa, T., Aono, M.: In: Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International, pp. 475–478 (2005)
41. Kaneko, Y., Tanaka, H., Ueda, M., Kato, Y., Fujii, E.: In: Device Research Conference (DRC), 2010, pp. 257–258 (2010)
42. Haykel, M., Jamaa, B., Carrara, S., Georgiou, J., Archontas, N., De Micheli, G.: In: 9th IEEE Conference on Nanotechnology, 2009. IEEE-NANO 2009, pp. 152–154 (2009)
43. Sacchetto, D., Ben-Jamaa, M., De Micheli, G., Leblebici, Y.: In: Solid State Device Research Conference, 2009. ESSDERC'09. Proceedings of the European, pp. 245–248 (2009)
44. Ziegler, M., Oberländer, M., Schroeder, D., Krautschneider, W.H., Kohlstedt, H.: Appl. Phys. Lett. **101**(26), 263504 (2012)
45. Bawedin, M., Cristoloveanu, S., Yun, J., Flandre, D.: Solid-State Electron. **49**(9), 1547 (2005)
46. Bawedin, M., Cristoloveanu, S., Flandre, D.: Solid-State Electron. **51**, 1252 (2007)
47. Berzina, T., Smerieri, A., Bernabo, M., Pucci, A., Ruggeri, G., Erokhin, V., Fontana, M.P.: J. Appl. Phys. **105**(12), 124515 (2009)

48. Kaeriyama, S., Sakamoto, T., Sunamura, H., Mizuno, M., Kawaura, H., Hasegawa, T., Terabe, K., Nakayama, T., Aono, M.: IEEE J. Solid-State Circuits **40**(1), 168 (2005)
49. Sacchetto, D., Ben-Jamaa, M., Carrara, S., De Micheli, G., Leblebici, Y.: In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 9–12 (2010)
50. Yang, J.J., Miao, F., Pickett, M.D., Ohlberg, D.A.A., Stewart, D.R., Lau, C.N., Williams, R.S.: Nanotechnology **20**(21), 215201 (2009). 9 pp.
51. Sze, S.M., Ng, K.K.: Physics of Semiconductor Devices, 3rd edn. Wiley, New York (2007)
52. Appenzeller, J., Knoch, J., Bjork, M., Riel, H., Schmid, H., Riess, W.: IEEE Trans. Electron Devices **55**(11), 2827 (2008)
53. Ecoffey, S., Mazza, M., Pott, V., Bouvet, D., Schmid, A., Leblebici, Y., Declereq, M., Ionescu, A.: In: Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International, pp. 269–272 (2005)
54. Ecoffey, S.: Ultra-thin nanograin polysilicon devices for hybrid CMOS-NANO integrated circuits
55. Sacchetto, D., De Micheli, G., Leblebici, Y.: In: The 16th International Conference on Solid-State Sensors, Actuators and Microsystems (2011)
56. Sacchetto, D., Savu, V., Micheli, G.D., Brugger, J., Leblebici, Y.: Microelectron. Eng. **88**(8), 2732 (2011)
57. Zhirnov, V., Cavin, R., Menzel, S., Linn, E., Schmelzer, S., Brauhaus, D., Schindler, C., Waser, R.: Proc. IEEE **98**(12), 2185 (2010)
58. Wei, Z., Kanzawa, Y., Arita, K., Katoh, Y., Kawai, K., Muraoka, S., Mitani, S., Fujii, S., Katayama, K., Iijima, M., Mikawa, T., Ninomiya, T., Miyanaga, R., Kawashima, Y., Tsuji, K., Himeno, A., Okada, T., Azuma, R., Shimakawa, K., Sugaya, H., Takagi, T., Yasuhara, R., Horiba, K., Kumigashira, H., Oshima, M.: In: Electron Devices Meeting, 2008. IEDM 2008. IEEE International, pp. 1–4 (2008)
59. Miao, F., Yi, W., Goldfarb, I., Yang, J.J., Zhang, M.X., Pickett, M.D., Strachan, J.P., Medeiros-Ribeiro, G., Williams, R.S.: ACS Nano **6**(3), 2312 (2012)
60. Sheu, S.S., Chang, M.F., Lin, K.F., Wu, C.W., Chen, Y.S., Chiu, P.F., Kuo, C.C., Yang, Y.S., Chiang, P.C., Lin, W.P., Lin, C.H., Lee, H.Y., Gu, P.Y., Wang, S.M., Chen, F., Su, K.L., Lien, C.H., Cheng, K.H., Wu, H.T., Ku, T.K., Kao, M.J., Tsai, M.J.: In: Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International, Feb., pp. 200–202 (2011)
61. Lin, M., El Gamal, A., Lu, Y.C., Wong, S.: In: Proceedings of the 2006 ACM/SIGDA 14th International Symposium on Field Programmable Gate Arrays, FPGA'06. pp. 113–122. ACM, New York (2006)
62. Betz, V., Rose, J., Marquardt, A. (eds.): Architecture and CAD for Deep-Submicron FPGAs. Kluwer Academic, Norwell (1999)
63. Han, K.J., Chan, N., Kim, S., Leung, B., Hecht, V., Cronquist, B., Shum, D., Tilke, A., Pescini, L., Stiftinger, M., Kakoschke, R.: In: Custom Integrated Circuits Conference, 2007. CICC'07. IEEE, Sept., pp. 89–91 (2007)
64. Liauw, Y.Y., Zhang, Z., Kim, W., Gamal, A., Wong, S.: In: Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International, Feb., pp. 406–408 (2012)
65. Linn, E., Rosezin, R., Kügeler, C., Waser, R.: Nat. Mater. **9**(5), 403 (2010)
66. Chen, Y., Lee, H., Chen, P., Tsai, C., Gu, P., Wu, T., Tsai, K., Sheu, S., Lin, W., Lin, C., et al.: In: Electron Devices Meeting (IEDM), IEEE International, 31, pp. 1–31 (2011)
67. Diebold, U.: Surf. Sci. Rep. **48**(5–8), 53 (2003)
68. Jeong, H.Y., Lee, J.Y., Choi, S.Y.: Adv. Funct. Mater. **20**(22), 3912 (2010)
69. Kim, W.G., Rhee, S.W.: Microelectron. Eng. **87**(2), 98 (2010)
70. Shin, S., Sacchetto, D., Leblebici, Y., Kang, S.M.: In: 13th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), 2012, pp. 1–4 (2012)
71. Svensson, J., Sourab, A.A., Tarakanov, Y., Lee, D.S., Park, S.J., Baek, S.J., Park, Y.W., Campbell, E.E.: Nanotechnology **20**(17), 175204 (2009)
72. Sacchetto, D., Savu, V., Micheli, G.D., Brugger, J., Leblebici, Y.: Microelectron. Eng. **88**(8), 2732 (2011)

Memristor-Based Addition and Multiplication

K'Andrea Bickerstaff and Earl E. Swartzlander Jr.

Abstract This chapter describes strategies for performing basic addition and multiplication in memristor-based structures. An overview of both analog and digital approaches for addition and multiplication is presented. Examples of memristor-based designs of ripple carry adders and array multipliers are shown.

1 Introduction

In 1971, Chua [3] presented the theoretical basis for a passive two-terminal circuit device that he called a "memristor," (a contraction of *memory* and *resistor*). By his models, the memristor would form a fourth basic circuit element, along with the resistor, the inductor, and the capacitor. At any given time, the memristor behaves like a conventional resistor, with the difference that its resistance depends on the past history of the current passing through it. The memristor, with memristance M, conforms to the fundamental relationship between magnetic flux, φ, and charge $q : \mathrm{d}\varphi = M(q)\mathrm{d}q$. Since M is a function of q, the memristor is a nonlinear circuit element with hysteretic current-voltage behavior.

In 2008, researchers at HP Labs realized the memristor in nanoscale titanium dioxide crosspoint switches [10]. Having shown the existence of memristors in the lab, the question becomes "What applications could memristors have?" Research is advancing quickly to develop memristors as non-volatile memory, called Resistive RAM (ReRAM), to potentially replace DRAM and Flash memory [6]. Jo, et al. show possible applications in artificial intelligence using memristors as synapses in neuromorphic circuits [4].

This chapter is an expanded version of [1], reprinted with permission.

K. Bickerstaff
KenQuest LLC, 5216 Crystal Water Drive, Austin, TX 78735, USA
e-mail: kbickerstaff@austin.rr.com

E.E. Swartzlander Jr. (✉)
University of Texas at Austin, 1 University Station C0803, Austin, TX 78712, USA
e-mail: eswartzla@aol.com

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_21,

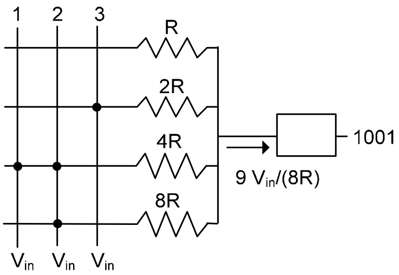


Fig. 1 Memristor crossbar array

This chapter examines the usage of memristors to perform arithmetic addition and multiplication. Section 2 gives an overview of previously proposed methods for performing arithmetic with memristors. Section 3 discusses the use of the "material implication" operation implemented with memristors to realize fundamental Boolean logic functions. In Sects. 4–7, the material implication technique is used to create a full adder, a ripple carry adder, a carry lookahead adder and an array multiplier. Finally Sect. 8 offers conclusions.

2 Previous Research

In the literature, arithmetic operations have been proposed using the memristor as (1) a switch, (2) an analog memory, and (3) interconnect. This section provides a brief summary.

2.1 Memristor Switch Logic

In the first approach, a memristor may be programmed to be either ON or OFF; that is, a low resistance, conducting state (ON) or a high resistance, non-conducting state (OFF). Arrays of memristor switches can be constructed by layering memristive material between vertical and horizontal nanowires.

Analog arithmetic processors can be constructed with memristor crossbar arrays and control and sensing logic [7]. Figure 1 shows a simplified example of addition using a memristor crossbar array. In the 3 by 4 array on the left, the switches that are conducting (ON) are indicated by a dot at the crosspoint. Each row wire includes a weighting resistor in order to assign the appropriate bit significance to the rows current contribution. In this example the most significant bit is in the top row and the least significant bit is in the bottom row.

For example, consider adding three binary numbers: 0010, 0011, and 0100. Columns 1, 2, and 3 of the crossbar are programmed with 0010, 0011, and 0100, respectively, with the 1's mapped to crosspoints. An input voltage, V_{in}, is applied to the columns of interest. In column 1, the one ON switch contributes

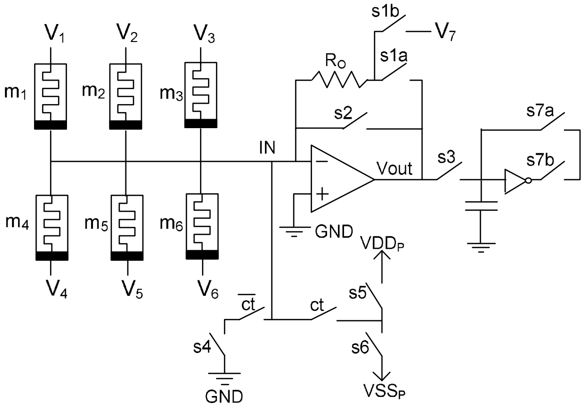


Fig. 2 Analog memory/computing unit (from [5])

$V_{in}/(4R)$ to the output current. The two conducting switches in column 2 contribute $(V_{in}/(8R) + V_{in}/(4R))$ to the output current. The one conducting switch in column 3 contributes $V_{in}/(2R)$ to the output current. The total output current of the crossbar array sums to $9V_{in}/(8R)$. The analog-to-digital converter with a resolution of $V_{in}/(8R)$ produces the final digital sum of 1001. Although the crosspoint array and analog summing can be fast, the need for an analog-to-digital converter (which is large, slow and consumes significant power) is a drawback.

2.2 Memristors as Analog Memory

Another approach to memristor-based analog arithmetic relies on the ability to program memristors to intermediate conducting states. Laiho and Lehtonen [5] present memristors as analog memories used for arithmetic computing. The schematic of an analog memory/computing unit is shown in Fig. 2 [5].

The analog memory/computing unit can be used to (1) copy the conductance of one memristor to another memristor, and (2) write the sum of multiple memory conductances into one memristor. These operations are done by the successive application of programming and monitoring cycles until the summing node of the operational-amplifier IN reaches the target level. When node IN reaches the target level, the voltage across the capacitor provides the control signal, *ct*, for the switches.

An "analog memory" is defined as a connected pair of memristors, as shown by m_1 and m_4 of Fig. 2. This memristor pairing allows for the representation of both positive and negative values. The conductance of the top memristor, g_{mt}, of the pair is always programmed to half the conductance range. The conductance of the bottom memristor, g_{mb}, may be programmed within the full conductance range. If $g_{mb} < g_{mt}$ then the value of the analog memory is negative. If $g_{mb} > g_{mt}$ then the value of the analog memory is positive. Using analog memories, addition as well as inversion (i.e., multiplication by -1) can be performed.

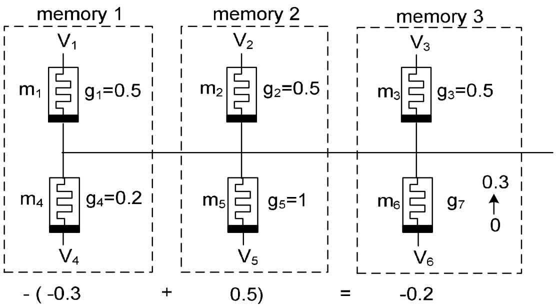
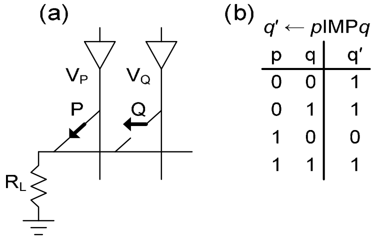


Fig. 3 Addition using three analog memories

Figure 3 shows the addition of the contents of two analog memories, with the sum stored in a third analog memory. Assuming relative conductance values ranging from 0 to 1, the conductances of m_1, m_2, and m_3 are all programmed to 0.5 and the conductances of m_4 and m_5 are 0.2 and 1 respectively. Analog memories 1 and 2 form the numbers -0.3 and 0.5. With cyclical programming and monitoring, the conductance of m_6 moves from an initial value of 0 to 0.3 and the third analog memory holds the value of -0.2. Each completed computation session using, analog memories, yields an output value that is inverted. Multiplication by -1 produces a final value of 0.2. To invert the value of memory 3 and store the result in memory 2, the conductance of m_5 is initialized to 0 and the voltage inputs to memory 1 would be set to high impedance. Once the programming and monitoring cycles complete, the conductance of m_5 would be 0.7 and the value of memory 2 is equal to 0.2. This technique is potentially fast, but it generates an analog result that is limited in precision. If a digital result is needed, an analog-to-digital converter would need to be added at a significant cost in speed, area, and power consumption.

2.3 Memristor Interconnect over CMOS

The third approach uses memristors as the programmable interconnect for CMOS logic. Xia, et al. report the implementation of the first memristor-CMOS hybrid circuits [11]. Memristor crossbar arrays are fabricated directly over CMOS logic devices. Using an array of vias, the memristor layer is electrically connected to the CMOS layer. Simulation results indicate that the FPGA-like hybrid CMOS integrated circuits benefit from significantly increased logic density without power or speed degradation [11]. This memistor-CMOS technology may be used to realize the same types of arithmetic circuits that are developed in CMOS/FPGA technology.



p	q	q′
0	0	1
0	1	1
1	0	0
1	1	1

Fig. 4 (**a**) Basic implication circuit and (**b**) truth table for IMP operation

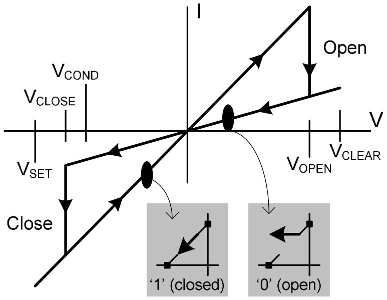


Fig. 5 Idealized memristive electrical characteristics (from [2])

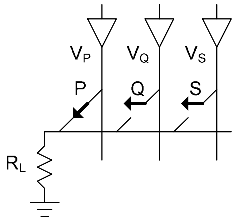
3 Logic Operations via Material Implication

Borghetti, et al. [2] use memristive switches to demonstrate the use of the material implication (IMP logic operation) to realize fundamental Boolean operations. The basic material implication circuit, shown in Fig. 4a, consists of two memristors, P and Q, which are connected using a horizontal nanowire to a load resistor, R_L. The P and Q devices are formed by vertical nanowires crossing over a layer of memristive switching material and the horizontal nanowire. Drivers are used to set each device to logic 1 or logic 0. By applying a negative voltage, V_{SET}, the memristive switch is placed in a low-resistance state (logic 1). Similarly, the application of a positive voltage, V_{CLEAR}, places the device in a high-resistance state (logic 0). An additional negative voltage, V_{COND}, is also needed for the implication operation, where $|V_{COND}| < |V_{SET}|$.

For operations, the notation $p \leftarrow x$ indicates that the state of switch P (the logic value p) is changed to x when P is pulsed by V_{SET}, V_{CLEAR}, or V_{COND}. Figure 5 from [2] shows the relationships among V_{SET}, V_{CLEAR}, and V_{COND} and the state of the memristive switch being considered 'open' (logic 0) and 'closed' (logic 1).

The IMP operation is given by $q \leftarrow p\,\mathrm{IMP}\,q$. The state of switch Q is changed by simultaneously applying a V_{SET} pulse to Q and a V_{COND} pulse to switch P. It is the interaction of the two pulses through P, Q, and R_L, that produce changes in the logical state of the switch Q depending on the existing values of p and q. Note that the IMP operation leaves p unchanged, but may change q. Figure 4b) shows the truth table for $q' \leftarrow p\,\mathrm{IMP}\,q$.

Fig. 6 Implication circuit for performing a NAND operation



$s \leftarrow p$NANDq

p	q	s
0	0	1
0	1	1
1	0	1
1	1	0

Fig. 7 Steps for performing a NAND operation via implication

Step 1

$V_S = V_{CLEAR}$
$s \leftarrow 0$

s
0
0
0
0

Step 2

$V_S = V_{SET}$
$V_P = V_{COND}$
$s' \leftarrow p$IMP$s$

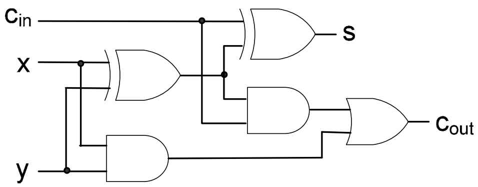
p	s	s'
0	0	1
0	0	1
1	0	0
1	0	0

Step 3

$V_S = V_{SET}$
$V_Q = V_{COND}$
$s'' \leftarrow q$IMPs'

q	s'	s''
0	1	1
1	1	1
0	0	1
1	0	0

Fig. 8 Full adder



Implication may be used to realize all fundamental logic functions [2]. Figure 6 shows the implication circuit with three memristive switches P, Q, and S to perform a NAND operation, $s \leftarrow p\,\text{NAND}\,q$. Figure 7 shows the three steps needed to execute the NAND. During Steps 2 and 3, the V_{SET} and V_{COND} pulses are applied simultaneously. The IMP operation changes the logic state of the S switch while leaving p and q unchanged.

4 A Memristor Full Adder

The full adder is the basic building block of most arithmetic circuits. It adds three binary inputs to generate a sum output and a carry output. Figure 8 shows the gate level schematic of a full adder. This full adder is implemented using two 2-input XOR gates, two AND gates, and an OR gate.

Table 1 Boolean operations implemented via material implication

Operation	Implementation	Devices (area)
$p\,\mathrm{NAND}\,q$	$= p\,\mathrm{IMP}\,(q\,\mathrm{IMP}\,0)$	3
$p\,\mathrm{AND}\,q$	$= (p\,\mathrm{IMP}\,(q\,\mathrm{IMP}\,0))\,\mathrm{IMP}\,0$	4
$p\,\mathrm{NOR}\,q$	$= ((p\,\mathrm{IMP}\,0)\,\mathrm{IMP}\,q)\,\mathrm{IMP}\,0$	4
$p\,\mathrm{OR}\,q$	$= (p\,\mathrm{IMP}\,0)\,\mathrm{IMP}\,q$	3
$p\,\mathrm{XOR}\,q$	$= (p\,\mathrm{IMP}\,q)\,\mathrm{IMP}\,((q\,\mathrm{IMP}\,p)\,\mathrm{IMP}\,0)$	3
$\mathrm{NOT}\,p$	$= p\,\mathrm{IMP}\,0$	2

Table 2 Counts of IMP operations and devices for Boolean logic

Operation	IMP Operations (latency)	Devices (area)
$s \leftarrow p\,\mathrm{NAND}\,q$	2	3
$s \leftarrow p\,\mathrm{AND}\,q$	3	4
$s \leftarrow p\,\mathrm{NOR}\,q$	5	6
$s \leftarrow p\,\mathrm{OR}\,q$	4	6
$s \leftarrow p\,\mathrm{XOR}\,q$	8	7
$s \leftarrow \mathrm{NOT}\,p$	1	2

4.1 Full Adder Realized via Material Implication

The supplementary information of [2] shows how the 16 binary operations of two variables are realized through the material implication operation. Table 1 shows several important logic functions used for forming basic arithmetic circuits such as half adders, full adders, etc., and the respective implication operations to implement them. The number of memristive devices is based on the assumption that the unconditional operation of $s \leftarrow 0$ is done in the first step of the process; that is, a device will not be cleared in subsequent steps.

Note that the Boolean logic operations listed in Table 1 are performed on two logic values p and q. For the XOR operation, the original logic states p and q are lost during the implication process. In fact, it is not clear that $(p\,\mathrm{IMP}\,q)$ and $(q\,\mathrm{IMP}\,p)$ could be performed either in parallel or serially without first copying the logic states p and q to additional memristive switches. Also, since the output of an IMP operation may be needed as an input to multiple subsequent IMP operations, it is probably best to retain a copy of the original logic value. A copy of a logic value of device P to device K can be made by executing two inversions, $k \leftarrow (p\,\mathrm{IMP}\,0)\,\mathrm{IMP}\,0$. Taking into account the possible need for copying one or both of the primary input logic values, the number of memristive devices (a measure of the circuit complexity) increases as shown in Table 2. This table also shows the number of sequential IMP operations, which is a measure of the latency.

Figure 9 shows the full adder function performed using material implication primitive operations. Initially, "input" memristors are set or cleared to the appropri-

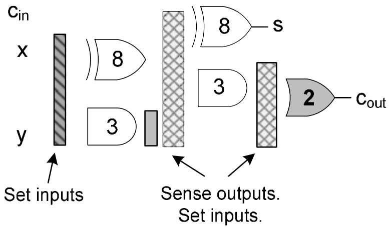


Fig. 9 Full adder via material implication

ate logic state to represent x and y. Also at this time, additional memristive devices are cleared to provide "0" logic values as needed. Using separate memristive circuit arrays, it is possible to compute the first XOR operation and the first AND operation in parallel. The first XOR operation takes 8 implication cycles. The first AND operation completes after 3 cycles. This early output of the first AND operation, relative to the first XOR operation, can be used to begin the processing of the OR operation. Two implication cycles of the OR operation can be performed in the same circuit array as the first AND. Thus, instead of taking 4 cycles, the OR operation is effectively reduced to 2 cycles.

Figure 10 details the implication cycles required to realize a full adder. By using separate memristor circuit arrays in parallel, time must be taken to sense the output logic states of the current stage and set the appropriate logic states of the input memristors of the next stage. This assumes that both the setting/clearing of memristive switches and the sensing of the logic states of memristive switches take 1 cycle each. Therefore, for the intermediate stages, sensing and setting logic states takes 2 cycles. Assuming that the inputs X, Y, and C_{in} are available at the same time, for a full adder via material implication, the sum, S, is produced in 19 cycles and the carry, C_{out}, in 18 cycles. Note that in cycles 6 and 7 a copy of the first AND's output state, k, is made to device G, indicated by the logic state g. This copy, g, is needed in order to maintain an original logic value of k, otherwise k would be clobbered at cycle 18 of the OR. Without a copy, device K would provide at different cycles the output logic state for both the first AND as well as the OR. Depending on the application, this reuse of device K may be an efficient implementation. Otherwise, copies in devices K and G may facilitate pipelined calculations.

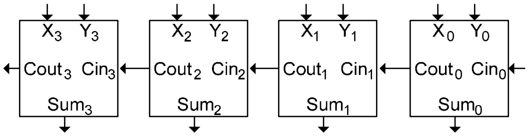
5 A Memristor Ripple Carry Adder

Figure 11 shows the block diagram of a traditional 4-input ripple carry adder. It is called a ripple carry adder because a carry generated at a given stage passes to the next more significant stage, where it may produce a carry that passes to the next stage, etc. It is a bit like dropping a pebble into a lake which produces waves that ripple outward.

1	*set inputs* j = 0, k = 0, f = 0, g = 0, w = 0		
2	j <- *Y* IMP j; **AND1**	s <- **X** IMP s; **XOR1**	
3	j <- *X* IMP j;	u <- s IMP u;	
4	k <- j IMP k;	t <- **Y** IMP t;	
5		v <- t IMP v;	
6	f <- k IMP f; *copy* **AND1**out	v <- **X** IMP v;	
7	g <- f IMP g; *into* g *for* ***OR***	u <- **Y** IMP u;	
8		w <- u IMP w;	
9		w <- v IMP w;	
10	*sense outputs* **W** = w		
11	*set inputs* i = 0, h = 0, m = 0, n = 0, p = 0, q = 0, r = 0		
12		i <- C_{in} IMP i; **AND2**	m <- C_{in} IMP m; **XOR2**
13		i <- **W** IMP i;	n <- m IMP n;
14		h <- i IMP h	p <- **W** IMP p;
15	*sense outputs* **H** = h		q <- p IMP q;
16	*set inputs* e = 0		q <- C_{in} IMP q;
17	e <- **H** IMP e; **OR** (AND2out)		n <- **W** IMP n;
18	g <- e IMP g; C_{out}		r <- n IMP r;
19			r <- q IMP r; ***S***

Fig. 10 Implication cycles for a full adder

Fig. 11 4-Input ripple carry adder



5.1 A Ripple Carry Adder Realized via Material Implication

The first and second full adders, producing Sum_0 and Sum_1 by material implication, are shown in Fig. 12. As indicated in Sect. 4, Sum_0 is generated in 19 cycles and Cout_0 in 18 cycles. Since the generation of the Cout_0 takes significantly longer than the 8 cycles of the first XOR operation, one implication cycle of the 2nd AND operation can be performed early. This reduces the 2nd AND operation from 3 cycles to 2 cycles.

Table 3 lists the number of implication cycles needed to generate each of the sums and carries for the 4-input ripple carry adder. The most significant sum, Sum_3, is created in 44 cycles and the most significant carry, Cout_3, in 42 cycles. In general, the delay to the most significant sum bit, Sum_{N-1}, of an N-bit ripple carry adder of this type is:

$$\text{Delay}(\text{Sum}_{N-1}) = 12 + 8 \times N \tag{1}$$

The delay to the most significant carry output Cout_{N-1} is:

$$\text{Delay}(\text{Cout}_{N-1}) = 10 + 8 \times N \tag{2}$$

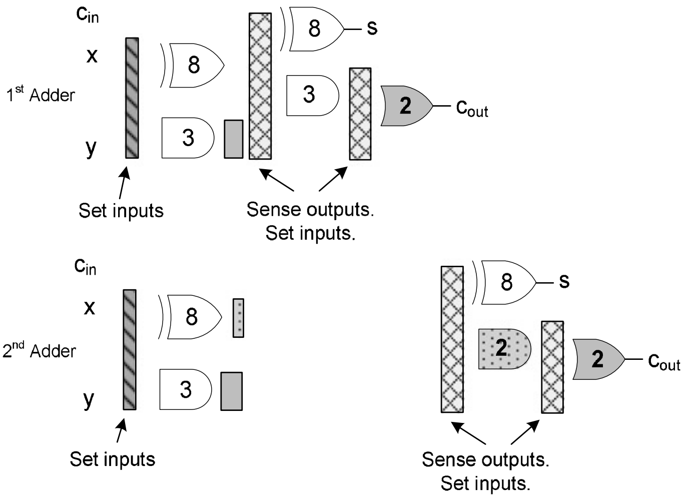


Fig. 12 First and second full adders of a ripple carry adder

Table 3 Count of IMP cycles to produce sum and carry outputs for a 4-bit ripple carry adder

Cout_3	Sum_3	Cout_2	Sum_2	Cout_1	Sum_1	Cout_0	Sum_0
42	44	34	36	26	28	18	19

6 A Memristor Carry Lookahead Adder

Shaltoot and Madian [9] produce two types of carry lookahead adders using the material implication logic listed in Table 1. The first is a conventional carry lookahead adder with outputs based on creating propagation and generation terms for inputs A and B. At the ith stage, the sum, S_i, and carry, C_{i+1}, are calculated based on propagate, P_i, and generate, G_i, signals:

$$\begin{aligned} P_i &= A_i \oplus B_i \\ G_i &= A_i B_i \\ S_i &= C_i \oplus P_i \\ C_{i+1} &= G_i + P_i C_i \end{aligned} \tag{3}$$

The second adder realization is based on a simplified carry lookahead adder design proposed by Pai and Chen [8]. Based on close examination of gate delays and simulation data, Pai and Chen propose gate level modifications that optimize critical path delays. In this design, the sum, S_i, and carry, C_{i+1}, are calculated based on

Table 4 Count of IMP cycles to produce sum and carry outputs for various 4-bit adders

	Cout_3	Sum_3	Cout_2	Sum_2	Cout_1	Sum_1	Cout_0	Sum_0
RCA	42	44	34	36	26	28	18	19
CLA	31	38	28	33	23	27	17	19
SimCLA	25	32	22	29	19	25	15	19

propagate, P_i, and negated generate, $\overline{G_i}$, signals:

$$\begin{aligned} P_i &= A_i \oplus B_i \\ \overline{G_i} &= \overline{A_i B_i} \\ S_i &= C_i \oplus P_i \\ C_{i+1} &= \overline{\overline{G_i}\ \overline{P_i C_i}} \end{aligned} \tag{4}$$

The simplified carry lookahead adder takes advantage of the faster NAND gate. Using the implication process, a NAND gate takes 2 IMP operations versus AND, OR, and NOR gates that require 3, 4, or 5 IMP operations, respectively. Therefore, opportunities to express circuits in terms of NAND gates with limited or no introduction of additional inverter stages will result in improved performance. Table 4 reports the number of implication cycles for Shaltoot and Madrian's 4-bit conventional and simplified carry lookahead adders, CLA and SimCLA, respectively. Also listed in Table 4 are the results for the ripple carry adder. Both carry lookahead adder implementations show improved performance over the ripple carry adder. The simplified carry lookahead adder outputs the most significant sum and carry bits 6 IMP operations faster than the conventional carry lookahead adder.

7 A Memristor Array Multiplier

The schematic for a 6 by 6 array multiplier is shown in Fig. 13. It has 7 rows and 6 columns of cells. The top row and left column consist of 2 input AND gates. The second row consists of half adders that are formed using one XOR operation and one AND operation. Most of the remaining cells have one AND operation and a full adder. The primary x and y inputs of the full adders in the third, fourth, fifth, and sixth rows arrive from the associated AND gate and the sum output of the adder directly above it in the previous row. The carry in, C_{in} arrives from an adder one column to the left and one row above. Assuming the inputs a and b are applied at the same time, the C_{in} signals to each full adder will be the earliest of each full adder input x, y, and C_{in}. The shortest delay path for each full adder will be C_{in} to Sum.

The bottom row of the array multiplier is a 5 input ripple carry adder. The first adder (on the right) is a half adder. The full adders forming the ripple carry adder, indicated in gray, have different input configurations in comparison to the upper

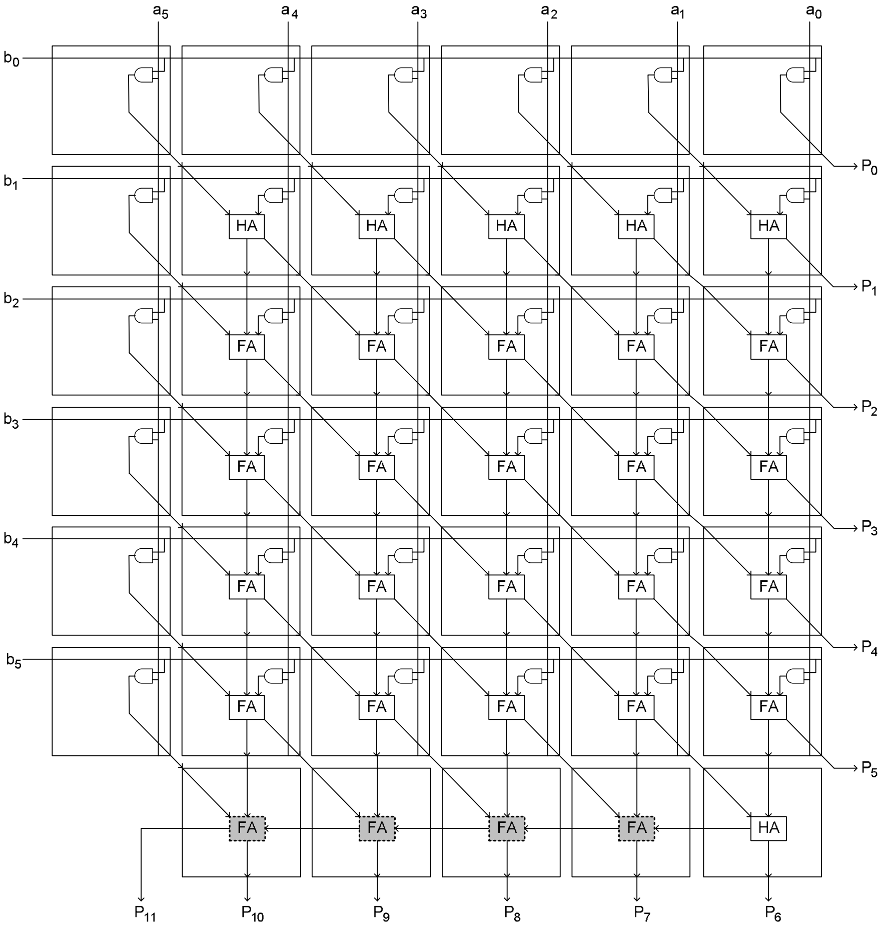


Fig. 13 6 by 6 array multiplier

array full adders. The primary *x* and *y* inputs of these full adders arrive from the sum output of the adder immediately above and the carry out from the adder immediately above and one column to the left. The C_{in} arrives from the adder to the immediate right of each adder. This provides the shortest delay path within the full adder to the slow carries that ripple from one full adder to the next.

Assuming inputs *a* and *b* are applied at the same time, the final product is produced by material implication in 127 cycles. Table 5 lists the number of cycles needed to generate each bit of the product of the 6 by 6 array multiplier.

The array multiplier can be expanded to accommodate inputs of any size. As the input word size is increased, additional rows (like the third, fourth, and fifth rows of Fig. 13) and additional columns (like the third, fourth, and fifth columns of Fig. 13) are added. As shown for a 6 by 6 multiplier in Table 5, each additional row increases

Table 5 Count of cycles to product bits for the 6 by 6 array multiplier

P_{11}	P_{10}	P_9	P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0
125	127	119	111	103	93	83	65	47	29	14	4

the delay by 18 and each column increases the delay by 8. The delay of an N by N multiplier for $N \geq 4$ is given by:

$$\text{Delay}(N \text{ by } N \text{ Multiplier}) = (26 \times N) - 29 \tag{5}$$

Thus, the delay of an array multiplier realized with memristive implication operations is proportional to the input word size. This is similar to the delay of CMOS array multipliers, although the proportionality constant is much larger.

8 Conclusions

This chapter has described techniques for performing arithmetic operations using memristors. Using memristors as switches in crossbar arrays is limited by (1) the interfacing and precision of the analog-to-digital converter and (2) the potential feedback paths within the array. Using memristors as an analog memory/computing unit depends on the accuracy of programming memristor conductance.

This chapter has shown that the material implication primitive operation can be used to perform addition and multiplication. Memristive implication operations were used to generate ripple carry adders, carry lookahead adders and array multipliers. These techniques could be used to create other types of adders and multipliers as wells as other arithmetic functions. The overhead both in terms of the operational delay (i.e., the number of clock cycles) and the programming complexity for the control, sensing and setting logic seems high. On the other hand memristors should achieve very high clock rates and logic density thus, offering an attractive alternative to CMOS.

Acknowledgement Earl Swartzlander is supported in part by a grant from AMD, Inc.

References

1. Bickerstaff, K., Swartzlander, E.E. Jr.: Memristor-based arithmetic. In: Forty-Fourth Asilomar Conference on Signals, Systems and Computers, November 2010, pp. 1173–1177 (2010)
2. Borghetti, J., et al.: 'Memristive' switches enable 'stateful' logic operations via material implication. Nature **464**, 873–876 (2010)
3. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**, 507–519 (1971)
4. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett., Am. Chem. Soc. **10**, 1297–1301 (2010)

5. Laiho, M., Lehtonen, E.: Arithmetic operations with memristor-based analog memory. In: Proceedings of the 12th International Workshop on Cellular Nanoscale Networks and Their Applications, February 2010
6. LaPedus, M.: HP, Hynix to commercialize the memristor. EE Times, 31 August 2010
7. Mouttet, B.: U.S. Patent 7,302,513, Programmable crossbar signal processor, 27 November 2007
8. Pai, Y.-T., Chen, Y.-K.: The fastest carry lookahead adder. In: Proceedings of the 2004 IEEE International Conference on Field-Programmable Technology, January 2004, pp. 434–436 (2004)
9. Shaltoot, A.H., Madian, A.H.: Memristor based carry lookahead adder architectures. In: Proceedings of the 2012 IEEE 55th International Midwest Symposium on Circuits and Systems, August 2012, pp. 298–301 (2012)
10. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
11. Xia, Q., et al.: Memristor-CMOS hybrid integrated circuits for reconfigurable logic. Nano Lett., Am. Chem. Soc. **9**, 3640–3645 (2009)

Memristor Emulators

Dalibor Biolek

Abstract There are three possible stages of exploring the memristor as the fourth fundamental circuit element: 1. Generation of the model, 2. Simulation of the element behavior with the aid of the model, and 3. Hardware emulation of the memristor. This Chapter deals with the third stage, describing circuit ideas of memristor emulators for practical laboratory experiments.

1 Memristor Emulator Requirements

Computer-aided modeling and simulation of memristors belong to important tools for researching general attributes of these fundamental elements. They also provide an important feedback for verifying the validity of theoretical theorems. The simulation can be invaluable for finding answers to questions such as "…what would happen if …" or, in other words, for studying the explored object via revealing its responses to changes in its parameters or in the structure of its model.

The laboratory experiments with memristors as truly existing objects is the next logical step. However, the unavailability of samples for common experiments has resulted in developing quite a number of hardware emulators of memristors and other mem-systems [1–15]. A more detailed analysis of such circuits leads to the knowledge that the purpose of some of them is to mimic the behavior of the TiO_2 memristor from [16]. Note that the simple model of the TiO_2 memristor is represented by the linear dependence of its memristance on the charge passing through or, in other words, by the quadratic dependence of the flux on the charge. Then the TiO_2 memristor is only one of many kinds of memristor whose constitutive relations (CRs) can be shaped almost in an arbitrary way. The above emulators cannot then be used for mimicking memristors with other types of the CR, the binary-state memristors being an example.

In addition, other required parameters and behavior of the developed emulator should be made clear prior to designing and producing it.

D. Biolek (✉)
Dept. of Electrical Engineering/Microelectronics, University of Defence/Brno University of Technology, Brno, Czech Republic
e-mail: dalibor.biolek@unob.cz

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_22,

As the first step, it should be considered if the purpose is to emulate the memristor as a hypothetic element or a generalized memristive system with its real properties (for example the TiO_2 memristive system with nonlinear boundary phenomena including the problem of "fixing the limiting states" [17], or the "threshold-voltage" memristive systems for CNN emulation [18]). Most of the published analog emulators serve for the demonstration of the basic fingerprints (particularly the pinched hysteresis loop, PHL) of the memristor as a fundamental circuit element [19]. Nevertheless, as noted above, these emulators mostly mimic the simple TiO_2 memristor [16]. For some emulators, the memristance/memductance dependence on the charge/flux is not defined clearly via the nonlinearity of the utilized principle of the simulation or it is not specified [2–4]. The memristor emulator in [5] exhibits the quadratic memductance/flux dependence. Attempts at the emulation of more complicated memristive systems, especially for neuromorphic applications, have been performed with digital emulators [6]. In contrast to analog emulators, they provide the software modification of the characteristics of the emulated system.

It is also important whether the grounded or the floating two-terminal device is the aim of the emulation. Most of the published circuits emulate the grounded memristors. It is quite sufficient if only the given element is emulated and excited by grounded signal source. However, such emulators are of no value if the emulated element were to be connected as floating into a more complicated circuit. The floating memristor can be emulated most conveniently via a floating two-terminal device with variable resistance, for example the digital potentiometer [6] or the light-dependent resistor (LDR) as a part of the optocoupler [3, 4]. However, a drawback of the optocoupler consists in the poorly defined and problematically adjustable dependence of resistance on the state variable. In addition, this resistance slowly fluctuates owing to light history. Another method of implementing the floating memristive port consists in the use of an independent, preferably battery-type power supply of the emulator [7, 8]. Some other methods, particularly those which utilize the bipolar outputs of the active elements [9] can cause unacceptable errors.

Since non-volatility is a key feature of the memristor, it is useful to ask how much the emulator should embody this attribute. Paradoxically, non-volatility is not very important for experiments and demonstrations whose lengths of duration are short in comparison with the time constant of the implemented memory (the integrator). The non-volatility can be accomplished via digital emulator.

As regards the analog emulator, the key role is played by the internal integrator, mostly implemented on the basis of operational amplifiers (OpAmps) [2], current conveyors [9–11], or classical current sources, charging the capacitors [12]. The input offset modifies intransigently the integrator output. On the other hand, the well-known method of offset reduction modifies the low-frequency behavior of the OpAmp-based integrators, i.e. at such frequencies where the memristor exhibits its well-known PHL. Quite a small deflection from the ideal 90-degree phase shift between the input and output signals of the integrator, caused by applying a big shunting resistor in parallel to the integrating capacitor, can cause an unpleasant modification of the PHL of the emulated memristor. This characteristic is also sensitive to parasitic voltage offsets. As a result, the PHL may not cross the v–i origin and it does not exhibit the odd-symmetry property. The internal integrators of the emulators must thus be designed carefully in order to minimize the offset and drift. Accurate operational and instrumentation amplifiers, appropriate for the construction of accurate integrators, are usually slow. However, it need not be a problem if the emulator is designed such that the desired memristor behavior (pinched hysteresis) is emulated just for low frequencies of the exciting signal. The possibility of adjusting the initial conditions, i.e. the initial state of the memristor memory, which can affect the emulator behavior is also an important feature.

Note that the offset and drift issue must be solved also for digital emulators, which contain the analog interface [13].

A concrete implementation of the emulator will also depend on whether or not the equipment will be used as an autonomous two-terminal device with in-built batteries [7, 8]. If yes, then the low-power low-voltage design [14] and the simplest possible circuitry are required.

2 Analog Emulators of the Memristor

The very first attempt at analogue emulation of the memristor comes from Chua's introductory paper [19]. The so-called M–R, M–L, and M–C mutators, described here, transform the non-linear resistor, capacitor, or inductor into the memristor. This transformation preserves the shape of the non-linear characteristic of the R, L, or C-type element, which is transformed into the corresponding charge-flux constitutive relation of the memristor. It is shown in [19] that each mutator can be of two types, depending on whether the independent/dependent variable in the original non-linear characteristic is transformed into independent/dependent variable of the constitutive relation of the memristor. Each type can be then implemented in two ways on the basis of various interconnections of controlled sources. The M–R mutator of type 1, realization 1, is then selected in [19] for a circuit implementation which consists of two operational amplifiers, 14 transistors, and a number of passive components (see Fig. 1).

The rather complicated schematic is given by the level of the IC industry in the 1970s. Otherwise such a concept of the emulator is very useful: The emulated device has all the memristor fingerprints since it is based on the definition of the memristor constitutive relation (CR). This CR is defined via the CR of a nonlinear resistor (or inductor or capacitor) connected to the corresponding port of the mutator. From this point of view, the emulator is easy to re-configure: The CR of the emulated memristor can be modified by replacing or "re-programming" the external two-terminal device. The disadvantage of the emulator from [19], i.e. the impossibility of a direct emulation of the floating memristor, can be overcome via the well-known circuit ideas (see Fig. 11 in Sect. 4 as an example).

The emulator published in [2] is based on a simple principle illustrated in Fig. 2. The voltage across the memristive port is sensed by the integrator, and its output voltage controls the conductance G_M, i.e. the memductance of the emulated memristor. It follows from the more detailed schematic in [2] that a JFET with a properly

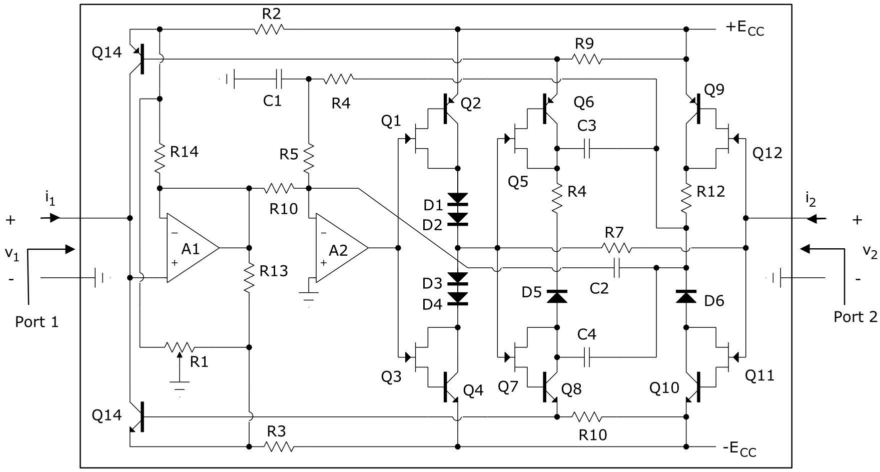
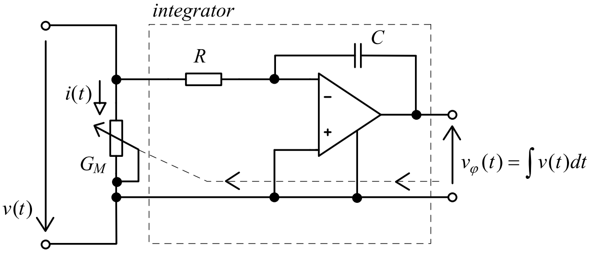


Fig. 1 Mutator which transforms nonlinear resistor connected to Port 2 into memristor emulated on Port 1. Adapted from [19]

Fig. 2 Simplified schematic of the memristor emulator [2]. Reprinted with permission from [2]. ©2011, John Wiley and Sons



adjusted operating point serves as this variable resistor. The emulator also contains the level converter at the input, the integrator, the circuit for setting the operating point of JFET, the transistor current-to-voltage converter, and the block for a final arrangement of the output voltage. The application is thus optimized for evaluating the pair of signals via a two-channel oscilloscope, namely the voltage which excites the memristor and the voltage which corresponds to the memristor current. Note that this emulator is not suitable for its integration into more complicated circuits as a two-terminal device. As a second drawback, the dependence of the memductance on the state variable is governed by the nonlinear characteristics of the transistor, and it can be modified only in a limited way via adjusting the operating point.

It is obvious from Fig. 2 that this emulator operates on the basis of the general model of the memristor in Fig. 3a: the excitation variable u, the voltage in this case, is integrated into the native state variable x. Then the memductance is determined via the block g, which represents the dependence of the memductance on this state variable. The memductance together with the voltage specify the output variable y, which is now the memristor current.

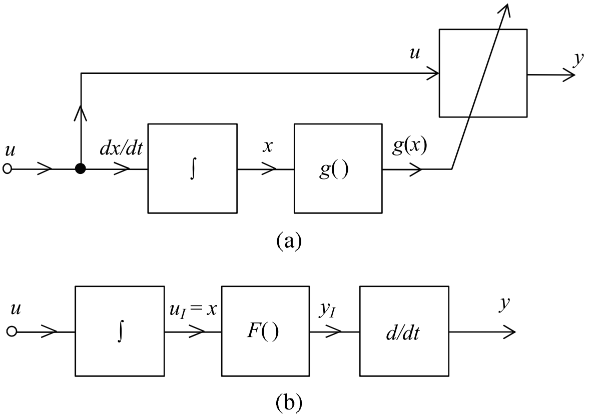


Fig. 3 Models of general memristor based on Parameter vs. State Map (PSM) $g(x)$ and state equation $dx/dt = u$ (**a**), and Constitutive Relation (CR) $y_I = F(u_I)$ (**b**), where $y = g(x)\ u$, u, y, and x are excitation variable, response, and state variable, y_I and u_I are time-domain integrals of the quantities y and u. For current-controlled memristor, u is current, y is voltage, and x is charge. For voltage-controlled memristor, u is voltage, y is current, and x is flux

The emulator in Fig. 4, published in [5], does not suffer from the above drawback. Its topology, containing the analog multipliers, ensures that the memductance of the emulated memristor is dependent on the flux φ, i.e. the integral of voltage v_1, according to the equation $G_M = (a + b\varphi^2)$, where $a < 0$, $b > 0$ are real coefficients, dependent on resistances R_1 to R_5 [5]. The emulation of other types of the characteristics is possible only after modifying the circuit. For example, omitting the multiplier U2 and interconnecting the output v_{15} of the integrator and the input y_1 of the multiplier U3 yield linear, not quadratic memductance-on-flux dependence. A similar emulator based on the analog multiplier was described in [15].

Another look at Fig. 4 reveals the following: the port voltage v_1 is integrated into the voltage v_{15}. The remaining circuitry transfers this voltage v_{15}, according to a concrete formula, into the current i_m, flowing through the emulated port. From this point of view, it is again a memristor model according to the general block diagram in Fig. 3a. In contrast to the emulator from Fig. 2, the current is not "computed" via a nonlinear resistor but is accomplished by the circuit containing multipliers and the amplifier U4. Using this circuit, we can adjust the prescribed dependence of the memductance on the integral of voltage.

The emulator in Fig. 5 was proposed in [3, 4]. The memristive port is emulated by the light-dependent resistor (LDR), which is a part of the optocoupler. The internal LED is excited from the integrator, which transfers the port voltage of the emulated memristor into the flux. The bias point of LED is set by an appropriate voltage V_{offset}. This circuit idea enables a simple emulation of the floating two-terminal device: the differential input of the integrator would be connected in parallel to the LDR. However, this possibility is not mentioned in [3, 4]. The drawback is the same as in the emulator from [2]: the impossibility of selecting a required type of the PSM (parameter vs. state map). This emulator probably suffers from another

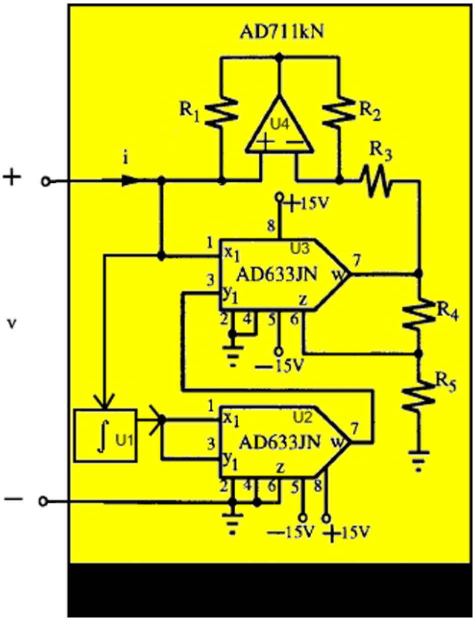


Fig. 4 Schematic of memristor emulator [5], containing integrator, analog multipliers, and v–i converter. Reprinted with permission from [5]. ©2010, World Scientific Publishing Co.

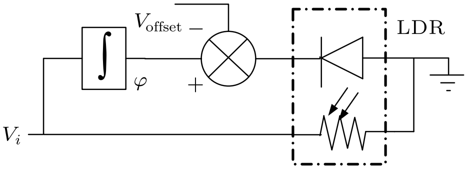


Fig. 5 Simplified schematic of memristor emulator (Adapted from [3] Wang et al., 2012), utilizing optocoupler

drawback: the resistivity of LDR depends not only on the instantaneous value of the LED current but it is governed by more complex inertial processes. That is why this circuit probably emulates not a memristor but a more complex higher-order element from Chua's table [20, 22].

The so-called incremental and decremental memristor emulators are proposed in [12]. The principle is illustrated in Figs. 6a and 6b. If the voltage v_x in Fig. 6a is derived from the voltage v_{in} or current i_{in}, then the input resistance of the circuit will be modified according to the principle of voltage bootstrap. If the voltage v_x is dependent on the time-domain integral of current i_{in}, i.e. on the charge, denoted q_C in Fig. 6a, then the "boosted" element R_S is seen from the viewpoint of input port as a charge-controlled resistor. The following formula is true for the circuit in Fig. 6a:

$$v_{in} = R_S i_{in} + v_x = \left(R_S + \frac{R_T}{C} q_C \right) i_{in}. \tag{1}$$

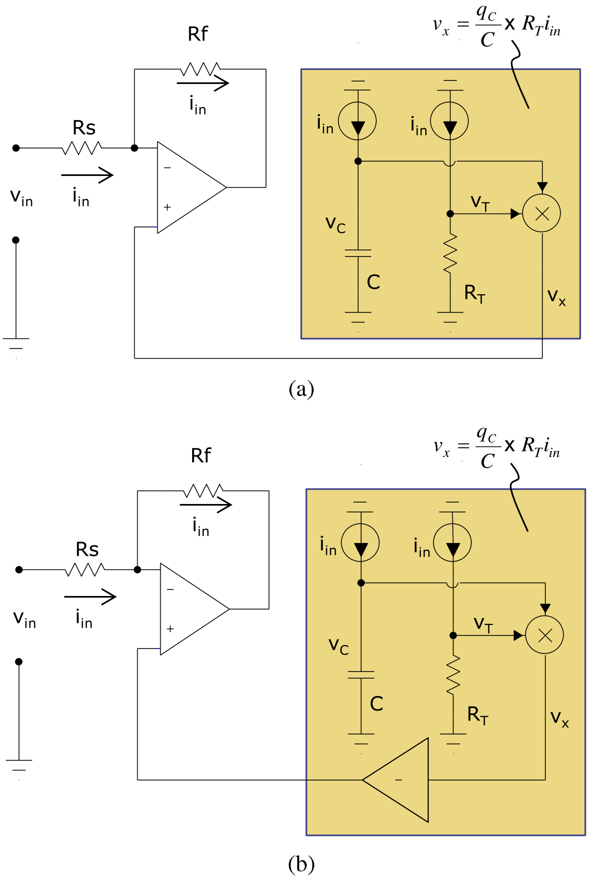


Fig. 6 Simplified schematic of memristor emulator of (**a**) incremental, (**b**) decremental type. Adapted from [12]

The inverter in the circuit in Fig. 6b causes a sign modification of the second term within the braces in (1):

$$v_{in} = R_S i_{in} + v_x = \left(R_S - \frac{R_T}{C} q_C \right) i_{in}. \tag{2}$$

The emulators in Figs. 6a and 6b are of the so-called incremental and decremental types. A comparison of Eq. (2) and the mathematical description of the TiO_2 memristor [16, 17] reveals that the decremental emulator (2) is suitable for the emulation of TiO_2 memristor with linear dopant drift, i.e. with the consideration of ideal rectangular window and on the assumption that the memristor will never pass into any of its boundary states. The pair of emulators in Fig. 6 can mimic an identical TiO_2 memristor but with mutually interconnected terminals.

In [12], the above emulators are generalized to the so-called expandable architecture, which enables, by means of a special interconnection of emulators, implementing serial, parallel, and hybrid connections of more memristors. Details are given in [12] and [21]. It is obvious that such a principle can be also used for the emulation of

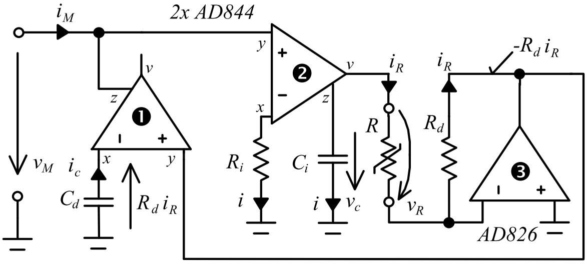


Fig. 7 Detailed schematic of memristor emulator from [10] (Biolek et al., 2011), operating on the principle of a mutator which transforms non-linear i–v characteristic of resistor R into a similar-in-shape charge-flux characteristic of emulated memristor

more complicated interconnections of two-terminal devices of various types, even if the authors do not mention it.

The circuit complexity and the sensitivity to the matching errors of internal controlled current sources appear as a certain drawback of such emulators. The final two-terminal device, emulated on the principle of expandable architecture, is always grounded. The application is limited to linear dependence of the memristance on the charge. On the other hand, Fig. 6 brings an interesting circuit idea, circumventing the problem of the emulation of floating components.

The emulator in Fig. 7 works on the principle of a mutator which transforms the non-linear characteristic $i_R = G(v_R)$ of resistor R, connected between the output of transimpedance OpAmp (TOA) No. 2 and the virtual ground of amplifier No. 3, into a similar-in-shape charge-flux characteristic of a memristor, emulated on the port v_M–i_M.

A more detailed analysis of the schematic in Fig. 7 results in the conclusion that it implements the mutator in Fig. 1 from [19] via present-day integrated circuits. The operation of such a mutator can be explained starting from the general model of the memristor in Fig. 3b: The exciting quantity u, the voltage in this case, is integrated and then transformed via a non-linear block F, which represents the constitutive relation of the memristor, and the output signal of this block is differentiated in order to get the output quantity y, which is the memristor current. Amplifier No. 2 together with R_i and C_i form the integrator of the voltage v_M with time constants R_iC_i. Its output voltage is also the voltage across the nonlinear resistor R. The current flowing through this resistor is converted to voltage via amplifier No. 3 with the feedback resistor R_d. This voltage is then converted to the current $i_C = i_M$ by the differentiating-type converter, consisting of amplifier No. 1 and capacitor C_d. In this way, the current through the memristive port is proportional to the time-derivative of the current through the nonlinear resistor with the proportionality coefficient R_dC_d. The integral of the current i_M—the charge of the emulated memristor—is directly proportional to the resistor current i_R with the proportionality constant R_dC_d, and the integral of memristor voltage—the flux—is proportional to the resistor voltage v_R with the proportionality coefficient R_iC_i. The circuit emulates a memristor with the memristance

$$R_{in} = \frac{d\varphi}{dq} = \frac{R_iC_i}{R_dC_d}\frac{dv_R}{di_R} = \frac{R_iC_i}{R_dC_d}R_{dif} \tag{3}$$

where R_{dif} is the differential resistance of nonlinear resistor R.

It follows from Eq. (3) that the ratio of the integration and differentiation time constants is equal to the ratio of the emulated memristance and the differential resistance of the resistor. It is useful to propose the integration constant with regard to the frequency range where we plan to study the hysteresis effects in the i–v characteristic of emulated memristor. The integrator gain is unity for the frequency of exciting voltage

$$f = \frac{1}{2\pi R_i C_i}. \tag{4}$$

Then the nonlinear resistor R is excited by a voltage with the same amplitude as v_M. Increasing this frequency will cause a decrease in the voltage swing on the resistor, and the nonlinear phenomena, including the pinched hysteresis loop, will gradually diminish.

For example, let us propose both time constants equal to ca 1.6 ms. With 100 Hz sinusoidal voltage, having the nominal swing $\Delta V = V_{max} - V_{min}$, the nonlinear resistor will be excited with the same voltage swing. Note that its v–i nonlinear characteristic is transformed into the φ–q characteristic of the emulated memristor. Then we can expect that the hysteresis effects will diminish for frequencies above 100 Hz, depending on the bias point of the resistor and on the shape of its v–i characteristic around this point.

The emulator in Fig. 7 represents one of several possible implementations of the classical M–R (memristor-resistor) mutators from the original paper [19] by means of up-to-date electronic components. It is simpler than the mutator from Fig. 1, while it embodies all its advantages. It does not provide the emulation of floating memristor but it can be accomplished, for example, via autonomous battery-type power supplies (see Fig. 11). Other versions of such mutators are described in [11, 14]. Various circuit implementations with up-to-date integrated circuits are possible, for example with transimpedance OpAmps (with the benefit of circuit simplicity) or with precise OpAmps or instrumentation amplifiers (offset and drift minimization, or low-power low-voltage design).

3 Digital and Hybrid Emulators of Memristors

The digital emulator of general memristive systems with a simplified schematic in Fig. 8 was introduced in [6]. The memristive port is represented by a digital potentiometer, whose resistance is set via a microcontroller. The port voltage is sensed using an analog-to-digital converter (ADC). The microcontroller provides the solution of state equation and computes the memristance from the state vector.

The digital potentiometer is a good choice for emulating floating two-terminal devices. This concept is useful for emulating not only memristors but also more general memristive systems. However, it cannot be used for mimicking other non-memristive devices. Such a disadvantage can be overcome by the emulator in

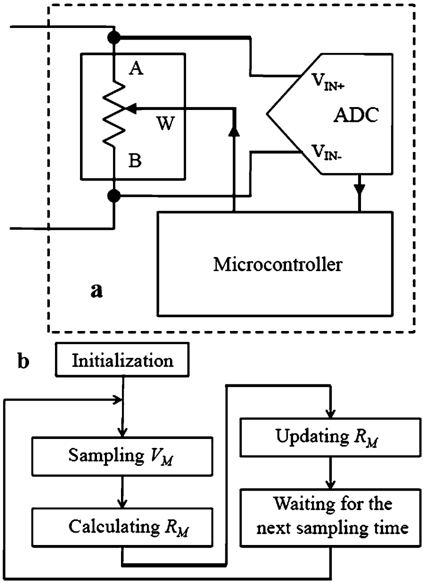


Fig. 8 Emulator employing digital potentiometer [6] (Pershin and Di Ventra, 2009)

Fig. 9, for the first time published in [13]. The authors called it "memulator" because of its ability to emulate general mem-systems including memristive, memcapacitive, and meminductive devices. It can be easily shown that, in principle, the memulator can mimic an arbitrary higher-order element from Chua's periodical table [22].

The memulator is based on the knowledge that, if the current i flowing through a two-terminal device can be determined from the device voltage v (via a device model), then such a device can be modeled by a source of current i which is controlled by the voltage v. Similarly, if the terminal voltage v of a two-terminal device can be determined from the device current i, then such a device can be modeled by a source of voltage v which is controlled by the current i.

This approach avoids the use of digital potentiometer in the emulator in Fig. 8. The type of the emulated device is then determined by a program. For the emulator in Fig. 9a, the terminal voltage v is sensed via the analog-to-digital converter (ADC), the current is computed in the microcontroller unit (MCU), and the result, after digital-to-analog conversion (DAC) serves to adjust the controlled current source. For the emulator in Fig. 9b, the current is sensed instead of the voltage, and then it is converted to input voltage for the ADC via auxiliary resistor R. After AD conversion, MCU computes the corresponding voltage, which is set via the DAC output as the terminal voltage of the controlled source.

Note that, in addition to the microcontroller and the ADC and DAC blocks, the memulator must be completed by auxiliary analog circuitries, namely by the controlled current source for version (a) and the controlled voltage source plus the circuit for current sensing and current-to-voltage conversion for version (b). That is why this emulator is called "hybrid". Details are given in [13].

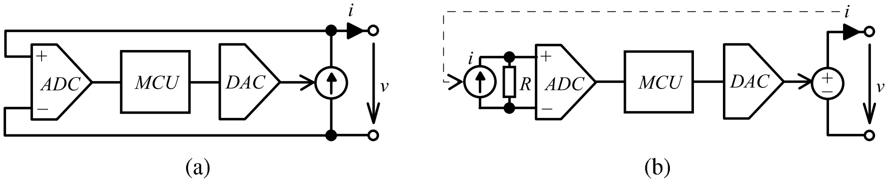


Fig. 9 Simplified schematics of hybrid emulator, employing controlled (**a**) current, (**b**) voltage source [13]. Reprinted with permission from [13]. ©2012, John Wiley and Sons

4 Future Possible Trends in Memristor Emulation

As far as the emulation of memristor—the fourth fundamental electrical element—is concerned, it is useful to develop such emulators that start from the memristor constitutive relation (CR) or the parameter vs. state map (PSM).

It was shown in Sect. 2 that most of the hitherto published emulators are based on the general block diagrams in Fig. 3. Those which enable a flexible configuration of the required CR or PMS are particularly appreciated. Emulators with a fixed-type characteristics cannot be utilized universally. This is the case of all analog emulators from Sect. 2, with the exception of mutators.

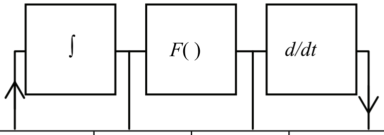
Universality is a key feature of digital and hybrid emulators. Emulators employing digital potentiometers (DP) are useful for mimicking the floating memristors. Certain problems can appear with the offset and drift of analog subcircuits as well as with a low number of bits of DP, which is the source of the quantization noise. It can be also problematic to make such emulators behave as autonomous two-terminal devices with internal batteries, since low-power design of digital emulators is not as easy as for the analog domain [6, 8, 14].

Let us analyze possible trends in designing universal analog emulators, based on the models in Figs. 3.

A general model of the memristor from Fig. 3b is shown at the head of Table 1. All possible combinations of the circuit quantities u, u_I, y_I, and y in terms of voltages (v) and currents (i) are drawn out below. For the voltage-controlled memristor (VCMR), the quantities u and y must be voltage and current. The current-controlled memristor (CCMR) is the opposite case. It is indicated in the column “schematics” that VCMR may be implemented via a controlled current source. The type of the control results from the sequence in which blocks of the type of integrator, nonlinear function F, and differentiator are arranged. The columns u_I and y_I contain all the possible combinations of voltages and currents by which these quantities can be represented within the emulator. Eight different methods of memristor implementation are behind this specification.

The green-coded lines in the Table 1 show cases when one of the circuit quantities in the pair u_I and y_I (input and output of the nonlinear block F) is represented by voltage and one by current. Then the block F can be easily implemented as a resistor with nonlinear v–i characteristic. Such an emulator can be regarded as a mutator which transforms this resistor into a memristor. Four ways of implementing such mutators follow from Table 1. For example, the first line in Table 1 for VCMR

Table 1 Possible ways of circuit implementation of memristor according to model from Fig. 3b



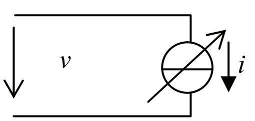
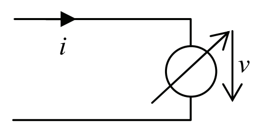
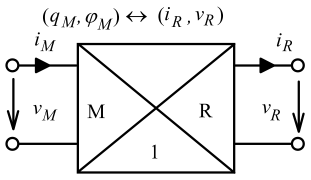
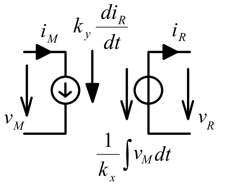
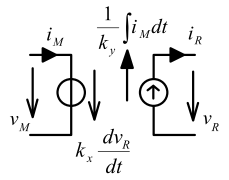
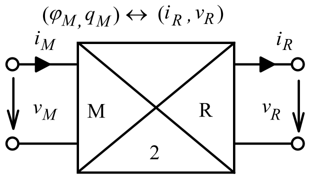
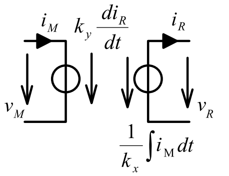
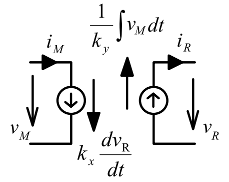
	u	u_I	y_I	y	note	schematics
VCMR	v	v	i	i	mutator	
	v	v	v	i		
	v	i	v	i	mutator	
	v	i	i	i		
CCMR	i	v	i	v	mutator	
	i	v	v	v		
	i	i	v	v	mutator	
	i	i	i	v		

Table 2 Two types and four basic realizations of M–R mutator [10]. Table style is adopted from [19]

type	symbol	implementation using controlled sources	
		realization 1	realization 2
1	 $(q_M, \varphi_M) \leftrightarrow (i_R, v_R)$ i_M i_R v_M M R 1 v_R	 i_M $k_y \frac{di_R}{dt}$ i_R v_M v_R $\frac{1}{k_x}\int v_M dt$	 $\frac{1}{k_y}\int i_M dt$ i_M i_R v_M v_R $k_x \frac{dv_R}{dt}$
2	 $(\varphi_M, q_M) \leftrightarrow (i_R, v_R)$ i_M i_R v_M M R 2 v_R	 i_M $k_y \frac{di_R}{dt}$ i_R v_M v_R $\frac{1}{k_x}\int i_M dt$	 $\frac{1}{k_y}\int v_M dt$ i_M i_R v_M v_R $k_x \frac{dv_R}{dt}$

describes an implementation where the voltage exciting the memristor is integrated in the form of a voltage (the quantity u_I), this voltage is converted via nonlinear resistor to current (the quantity y_I), and the current is transformed by the differentiating cell into the output quantity (the memristor current). In terms of the mutator concept from Chua's original paper [19], it is the M–R (memristor-resistor) mutator of type 1 realization 1 with its source-level schematic in Table 2. The coefficients k_x and k_y in the schematic diagrams denote time constants of the integrator and differentiator [10, 11]. The third line in Table 1 for VCMR corresponds to the M–R

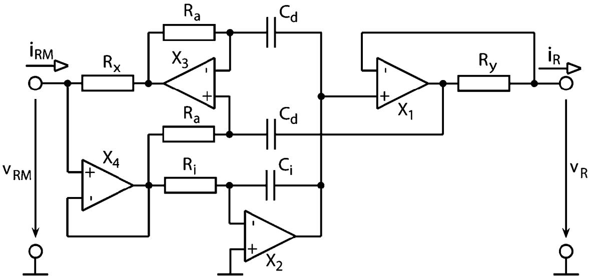


Fig. 10 Example of circuit implementation of Type 1 Realization 1 M–R mutator employing accurate OpAmps [8] (Biolek, 2012)

mutator of type 2 realization 2 in Table 2. The first and third lines in Table 1 are then presented in Table 2 by mutators of type 2 realization 1 and type 1 realization 2.

The controlled sources from Table 2 can be implemented in several ways. A concrete mutator (type and realization) should be selected with regard to an easy implementation of given controlled sources by concrete integrated circuits. For OpAmps, the integrating and differentiating cells are more easily implemented with voltage and not current outputs. Then the designer will probably prefer the mutator of type 2 realization 1. Such a mutator needs to sense the current i_M flowing through the memristive port for controlling the differentiator. The sensing of current can be accomplished with the help of current conveyors, which are parts of transimpedance OpAmps (AD844 being a typical commercial representative). However, these OpAmps are not manufactured as low-power low-offset ICs. The final choice of a proper mutator thus also depends on the designer's experience.

Analyzing again the mutator from Fig. 7, it is obvious that it is an M–R mutator of type 1 realization 1. It is shown in [11] that the transimpedance OpAmp (TOA) is the active block suitable for implementing all the versions of the controlled sources from Table 2. Concrete TOA-based schematics of all mutators from Table 2 are designed therein. Low-power accurate emulators can be made either by assembling TOAs from appropriate classical OpAmps [14] or by synthesizing the mutators from Table 2 directly from these OpAmps. The latter approach is illustrated in Fig. 10 [8]. The type 1 realization 1 M–R mutator is implemented via four accurate autozeroing low-offset OpAmps. The voltage v_{RM} of the memristive port is sensed by the buffer X4 and conveyed to the integrator consisting of the OpAmp X2, R_i and C_i. The integrator output voltage is copied by the buffer X1 into the resistive port. The current i_R through this port is converted to a voltage drop on resistor R_y, and this voltage is processed by a differential-input differentiating voltage-to-current converter, formed by X3, C_d, R_a and R_x.

The low power dissipation of the mutator in Fig. 10 and the low supply voltages for internal rail-rail OpAmps enable implementing the floating memristor emulator with in-built batteries according to Fig. 11. This circuit is interconnected with the remainder of the application circuit only via the memristive port. In this way, a full-value emulation of the floating component is assured (see bottom left image). The bottom-right image introduces another mutator implementation which is a part of the universal building kit of the so-called incremental mutators [8] for emulating arbitrary higher-order elements from Chua's periodical table [22].

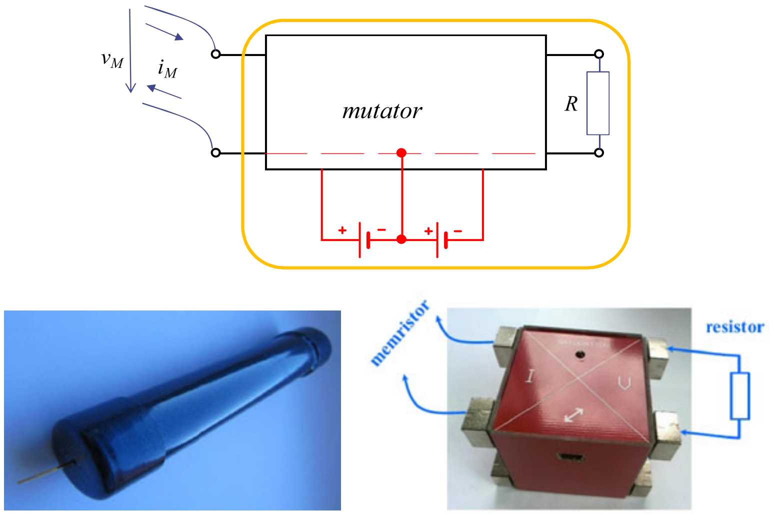


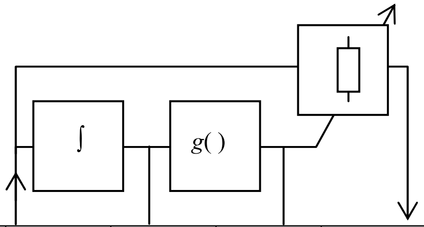
Fig. 11 Possible implementation of floating mutator employing independent power supply: Emulator as two-terminal equipment with in-built batteries (*bottom left*) [7]; universal mutator with internal power supply which is derived from USB powering (*bottom right*) [8]

It is obvious from Table 1 that, in addition to mutators, another four possible implementations of memristor emulators also exist, which are based on the general model in Fig. 3b. The transforming block F, which determines the shape of the CR of emulated memristor, can be made up using a nonlinear voltage-voltage or current-current converter. It should be carefully considered which is preferable for a concrete constitutive relation: the synthesis of one of these converters or of a nonlinear resistor.

The emulators from Table 1 require a differentiating cell which can be a source of potential problems and unstable behavior. The memristor model in Fig. 3a does not use this cell. The block which generates the output quantity y from the input quantity u and from the parameter g can be implemented either by a resistor with variable resistance/conductance (see Table 3) or by a source which is controlled from the output of analog multiplier (see Table 4).

For a successful implementation according to Table 3, it is crucial to design a resistor whose resistance is electronically controllable according to a prescribed algorithm. An interesting implementation of this kind can be the emulator with piece-wise-constant memristance vs. state map for memristor switching memory. The variable resistor can be made up via switching a set of fixed linear resistors. The particular switches can be controlled by simple comparator-based circuits which implement the block $g()$ and are driven from the integrator output. The emulator in Fig. 4 is an example of the VCMR emulators from Table 4. Most of the commercial analog multipliers, available on the market, have voltage inputs and, with a few exceptions, also voltage outputs. Then it is necessary to include the appropriate voltage-current converters into the emulator. More specifi-

Table 3 Possible variants of memristor circuit implementation based on controlled resistor, according to model in Fig. 3a



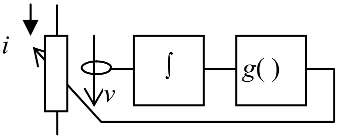
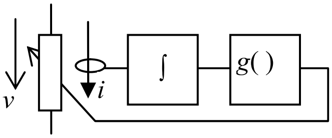
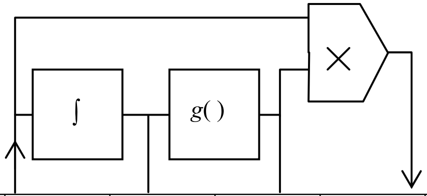
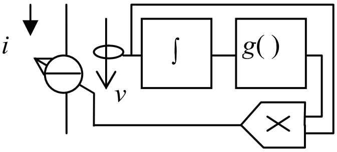
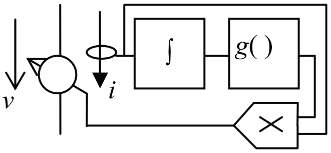
	u	u_I	g	y	schematics
VCMR	v	v	i	i	
	v	v	v	i	
	v	i	v	i	
	v	i	i	i	
CCMR	i	v	i	v	
	i	v	v	v	
	i	i	v	v	
	i	i	i	v	

Table 4 Possible variants of multiplier-based memristor circuit implementations according to model in Fig. 3a



	u	u_I	g	y	schematics
VCMR	v	v	i	i	
	v	v	v	i	
	v	i	v	i	
	v	i	i	i	
CCMR	i	v	i	v	
	i	v	v	v	
	i	i	v	v	
	i	i	i	v	

cally, the voltage-current converter should be included at the multiplier output for VCMR and the current-voltage converter at the bottom input of the multiplier for CCMR emulator.

5 Conclusions

This Chapter illustrates the usefulness of the modeling procedure from Fig. 3 for designing the memristor emulators. Several hitherto published circuits are analyzed. In addition to digital and hybrid emulators, simple analog emulators, enabling an easy modification of the constitutive relation (CR) or Parameter vs. State Map (PSM) of the emulated memristors, are found to be interesting and promising, particularly those based on mutators. Important challenges of future research are low-power low-voltage emulators of true floating memristors and higher-order elements from Chua's periodical table [22].

Acknowledgement This work was partially supported by the Czech Science Foundation under grant No P102/10/1614, and by the Project for development of K217 Dept., UD Brno, Czech Republic.

References

1. Sodhi, A., Gandhi, G.: Circuit mimicking TiO_2 memristor: a plug and play kit to understand the fourth passive element. Int. J. Bifurc. Chaos **20**(8), 2537–2545 (2010)
2. Valsa, J., Biolek, D., Biolek, Z.: An analogue model of the memristor. Int. J. Numer. Model. Electron Netw. Devices Fields **24**(4), 400–408 (2011)
3. Wang, X.Y., et al.: Implementation of an analogue model of a memristor based on a light-dependent resistor. Chin. Phys. B **21**(10), 108501 (2012)
4. Wang, X.Y., et al.: Design of a memcapacitor emulator based on a memristor. Phys. Lett. A **376**, 394–399 (2012)
5. Muthuswamy, B.: Implementing memristor based chaotic circuits. Int. J. Bifurc. Chaos **20**(5), 1335–1350 (2010)
6. Pershin, Y.V., Di Ventra, M.: Experimental demonstration of associative memory with memristive neural networks (2009). arXiv:0905.2935
7. Biolek, D.: Analog building blocks for emulating memristors, memcapacitors, and meminductors. In: Invited Lecture at 7th Int. Conf. on Electrical and Electronics Engineering (ELECO 2011), Bursa, Turkey (2011)
8. Biolek, D.: Modeling, simulation and analog emulation of memristors and higher-order elements. In: Invited Lecture at the 3rd Memristor and Memristive Symposium, Turin, Italy (2012)
9. Pershin, Y.V., Di Ventra, M.: Emulation of floating memcapacitors and meminductors using current conveyors. Electron. Lett. **47**(4), 243–244 (2011)
10. Biolek, D., et al.: Mutators for transforming nonlinear resistor into memristor. In: 20th European Conf. on Circuit Theory and Design (ECCTD), Linkoping, Sweden, pp. 488–491 (2011)
11. Biolkova, V., Biolek, D., Kolka, Z.: Unified approach to synthesis of mutators employing operational transimpedance amplifiers for memristor emulation. In: Proc. of the 11th Int. Conference on Instrumentation, Measurement, Circuits and Systems (IMCAS'12), Rovaniemi, Finland, pp. 110–115 (2012)

12. Kim, H., et al.: Memristor emulator for memristor circuit applications. IEEE Trans. Circuits Syst. I, Regul. Pap. **59**(10), 2422–2431 (2012)
13. Kolka, Z., Biolek, D., Biolkova, V.: Hybrid modeling and emulation of mem-systems. Int. J. Numer. Model. Electron Netw. Devices Fields **25**(3), 216–225 (2012)
14. Biolek, D., Biolkova, V., Kolka, Z.: Low-voltage-low-power current conveyor for battery supplied memristor emulator. In: Proc. of 5th International Conference on Circuits, Systems and Signals (CSS'11), Corfu, Greece, pp. 171–175 (2011)
15. Mutlu, R., Karakulak, E.: Emulator circuit of TiO_2 memristor with linear dopant drift made using analog multiplier. In: Proc. of 2010 National Conference on Electrical, Electronics and Computer Engineering (ELECO 2010), Bursa, Turkey, pp. 380–384 (2010)
16. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
17. Biolek, Z., Biolek, D., Biolkov, V.: Spice model of memristor with nonlinear dopant drift. Radioengineering **18**(2), 210–214 (2009)
18. Pershin, Y.V., Fontaine, S., Di Ventra, M.: Memristive model of amoeba's learning. Phys. Rev. E **80**, 021926 (2009)
19. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
20. Chua, L.O.: Device modeling via basic nonlinear circuit elements. IEEE Trans. Circuits Syst. **CAS-27**(11), 1014–1044 (1980)
21. Maheshwar, P.S., et al.: A voltage mode memristor bridge synaptic circuit with memristor emulators. Sensors **2012**(12), 3587–3604 (2012)
22. Chua, L.O.: Nonlinear circuit foundations for nanodevices, part I: the four-element torus. Proc. IEEE **91**(11), 1830–1859 (2003)

Modeling Memristor-Based Circuit Networks on Crossbar Architectures

Ioannis Vourkas and Georgios C. Sirakoulis

Abstract Over 30 years ago Leon Chua proposed the existence of a new class of passive circuit elements, which he called memristors and memristive devices. The unique electrical characteristics associated with them, along with the advantages of crossbar structures, have the potential to revolutionize computing architectures. Being associated with the totally nonlinear behavior of individual memristive elements, circuits of multiple memristors may work in very complicated way, quite difficult to predict, due to the polarity-dependent nonlinear variation in the memory resistance (memristance) of individual memristors. A well defined and effective memristor model for circuit design combined with a design paradigm which exploits the composite behavior of memristive elements, based on well understood underlying logic design principles, would certainly accelerate research on nanoscale circuits and systems. Towards this goal, we explore the dynamics of regular network geometries containing only memristive devices and present a memristor crossbar circuit design paradigm in which memristors are modeled using the quantum mechanical phenomenon of tunneling. We use this circuit model to test various logic circuit designs capable of universal computation, and finally, we develop and present a novel design paradigm for memristor-based crossbar circuits.

1 Introduction

Memristor was originally proposed in 1971 by the nonlinear circuit theorist Leon Chua [1]. The memristor is a fourth class of electrical circuit element, joining the resistor, the capacitor, and the inductor and exhibits its unique properties primarily at the nanoscale [2, 3].

Currently, the version of the titanium dioxide (TiO_2) substrate memristor by Hewlett Packard (HP) [3] is the most generally recognized memristor type. It is

I. Vourkas (✉) · G.C. Sirakoulis
Democritus University of Thrace, Xanthi, Greece
e-mail: ivourkas@ee.duth.gr

G.C. Sirakoulis
e-mail: gsirak@ee.duth.gr

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_23,

based on two thin-layer TiO_2 films. The bottom layer acts as an insulator whereas the top film layer acts as a conductor via oxygen vacancies in the titanium dioxide (TiO_2 changes its resistance in the presence of oxygen). Voltage increment moves the oxygen vacancies from the top layer towards the bottom layer, thus changing its resistance. As an ac voltage is applied to the terminals of the device, the current-voltage (I–V) curve shows a pinched hysteresis loop that passes through the origin. Around the origin the device acts as a traditional resistor (linear region). Nevertheless, until nowadays there has been no direct connection between a model and the memristor physical properties, except for a simple physical model proposed by Strukov et al. [3]. An appropriate descriptive model will not only lead to a better understanding of its behavior, but will also result to a better exploitation of its unique properties. Currently, there is a growing variety of systems that exhibit memristive behavior, as academia and industry keep on with their research and prototyping. However, while most of the research has focused on the properties of these single devices, very little is known about their response when they are organized into networks. When multiple memristors are connected to each other, the overall behavior of the devices becomes complicated and is difficult to predict.

To access the memristive properties, memristors are placed at the intersection of crossbar nanowires so that a charge can be passed through them [2]. Many proposed architectures for nanoscale electronics have focused on the crossbar architecture because of its fabrication simplicity and of the inherent redundancy which supports defect tolerance [3–5]. The crossbar architecture can be used to compute logic functions based on the placement of specific device switches at the wire junctions and on their state. Currently, known disadvantages of crossbar-based designs compared to CMOS technology are mainly due to the device choice used to implement the switches. Configurable nanoscale memristors in a crossbar [6–8], where memristors are simply used as two-state switches, would provide a more powerful foundation for nanoelectronic computation. It would also offer considerable flexibility to system architects and would enhance the toolkit of circuit designers.

In this chapter we focus on memristor device modeling and on the architectural perspectives that arise from circuits with configurable memristors, studying the composite characteristics of memristive elements connected in regular parallel and serial configurations. We analyze the characteristics of complex memristor circuits and conduct simulations in order to investigate the relationships among the single devices, using a proper memristor circuit model which is based on the effective tunneling distance modulation [9, 10]. Depending on their orientation (polarity), their initial condition and their internal properties, which are summarized in the actual values of the parameters of the model, their overall response turns up totally nontrivial. We also show how composite memristive systems can be efficiently built out of individual memristive devices, presenting different electrical characteristics from their structural elements for several considered input signals. Finally, we exploit the threshold-dependent nonlinear memristive behavior and elaborate the presented memristive networks to propose a novel CMOS-like [5] circuit design paradigm comprising memristive elements. Based on the proposed memristor model, we built a crossbar circuit simulator using the Java programming language [11], and used it to simulate the equivalent circuits of the universal digital logic gates set (NOT, NAND, NOR) as well as more complex circuits (AND-OR-INVERT function, Half Adder), which altogether make possible the design and implementation of any digital logic circuit.

2 Application Potential of Memristor Based Circuits

In this section, the potential of memristor-based crossbar circuits is discussed. A series of technical specifications regarding performance metrics found in the literature are summarized and presented. Such experimental information forms a ground basis for future establishment and practical use of emerging nanotechnologies, verifying their potentiality and also further motivating the study of new design paradigms and future applications.

In their work, Stan et al. [12] point out that molecular electronics have a lot of potential to enable electronic functionality to continue scaling beyond the end of CMOS. Specifically, in Table I of their work, potential moletronic circuit approaches are compared with predictions made in the ITRS [13] for the end of the CMOS roadmap, in terms of speed, power, and density. The table compares footprint, energy/transition, delay, power density, and compute density, taking into account only high-performance CMOS. It can be seen that crossbar-based circuits show clear potential for superiority in area, as well as in energy consumption, but not yet in performance. Furthermore, as long as the cost of building CMOS chips continues to follow an exponential law with time, it is pointed out that it is reasonable to expect that molecular chips will be less expensive to build, since chemical self-assembly is used to build the devices, rather than many, very precise lithography steps.

Moreover, Jo et al. [14] investigate two terminal amorphous-silicon (a-Si) based resistive switches. These devices are found to exhibit a number of desirable performance metrics in terms of speed ($<$50 ns programming time), and endurance ($>10^5$ cycles), which make them suitable for high-performance memory and logic applications based on conventional or emerging hybrid nano/CMOS architectures. Kim et al. in [15] demonstrate a high-density, fully-operational hybrid crossbar/CMOS system which utilizes a memristor-based crossbar array. The structure of the studied device consists of an a-Si layer acting as the switching medium. A 50 nm half pitch was achieved through electron beam lithography and yielded an equivalent data storage density of 10 Gbits/cm^2 when storing one bit per memory cell. In addition, Lu et al. in [16] review the recent progress on the development of two terminal resistive devices and report on a number of promising performance metrics shown by devices based on solid state electrolytes like a-Si. Specifically, resistance switching speed of $<$10 ns and endurance of $>10^8$ cycles are mentioned, whereas data retention of $>$10 years at 85 °C and nominal energy consumption per operation in the subpicojoule range have also been reported [17–20]. Some of the recent advances of binary metal-oxide resistive switching devices reported in the literature are summarized in Table 4 of [21].

Also, Ebong and Mazumder [22] analyze the feasibility of memristor memories and introduce an adaptive read, write, and erase method. The power metrics are compared to flash memory technology, and the memristor-based memory exhibits an energy per bit consumption about one tenth of that of flash when programming, comparable to flash when erasing, and about one fourth of flash when reading. The aforementioned results are summarized in Table I of [22]. Also, Eshraghian et al. in [23] provide a new approach towards the design and modeling of memristor-based content addressable memory (CAM). Emerging memory devices and technologies are discussed, and a range of performance parameters and salient features of characteristic emerging technologies for memories can be found in Table II of [23]. The memristor-based crossbar architecture is shown to be highly scalable [24] and promising for ultra-high density memories [25]. Notably, a memristor with minimum feature sizes of 10 and 3 nm yields 250 Gb/cm^2 and 2.5 Tb/cm^2, respectively.

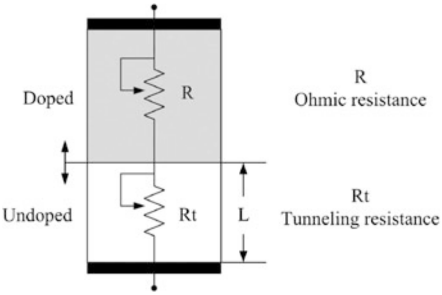
It is worth to mention that in 2010, almost two years after their first memristor announcement (their device comprised a 50 nm titanium dioxide film and exhibited ion mobility of 10^{-10} cm^2/(Vs)), HP Labs also declared that they had practical memristors working at 1 ns ($\approx$1 GHz) switching times and 3 nm by 3 nm sizes, with an impressive electron/hole mobility of 1 m/s [26]. These statistics forespeak well for the future of the technology and memristors could easily rival the current sub-25 nm flash memory technology.

3 Memristor Device Modeling

3.1 Related Work

The HP Laboratories group in their first memristor implementation announcement [3], along with experimental device examples, suggested a coupled variable-resistor model for memristors. Ever since, this model was improved by Joklegar and Wolf [27], whereas several papers by HP [28, 29] report on further developments of resistance switching theory for TiO_2 devices. Di Ventra et al. [30] suggested a simple threshold-type model of memristive systems [31] and employed it in programmable analog circuits [32, 33]. Liu et al. [34] proposed a material-oriented methodology to control resistance switching behavior of oxide-based resistive switches, based on a unified physical model [35] where formation of conducting filaments (CFs) is due to the generation of new oxygen vacancies by ionizing oxygen ions from the lattice under voltage bias. Furthermore, approximated SPICE memristor models have been proposed and tested with promising simulation results [36]. However, little work has been done towards memristor modeling, whereas various implementation paradigms are continually being proposed combining nano/CMOS [7], reconfigurable architectures and memristors [37, 38], resulting in hybrid implementations [39, 40], that could have a profound effect on integrated circuit performance.

Fig. 1 Equivalent circuit of the coupled ohmic-tunneling variable-resistor model; L is the width of undoped dioxide layer (tunnel barrier width)



3.2 A Novel Memristor Circuit Model

We describe here an alternative solution for modeling memristors, recently proposed by the present authors [10], explaining the devices memristive behavior by investigating the occurrence of quantum tunneling [9]. The equivalent circuit of the proposed model is depicted in Fig. 1. It is a threshold-type switching model of a two-terminal voltage-controlled electrical device that exhibits memristive behavior [31], and it is described by the following expressions:

$$I(t) = G(L,t)V_M(t), \tag{1}$$

$$\dot{L} = f(V_M,t), \tag{2}$$

L is the single state variable of the system which in our model is the tunnel barrier width (thickness of the free of oxygen vacancies dioxide layer), with the electrical current transport process being limited primarily by tunneling through it. Also, G is the device's conductance and V_M is the applied ac voltage. The time derivative of the state variable in (2) is interpreted as the speed of movement of the barrier between the two layers, due to the applied voltage bias.

We suggest the coupled ohmic-tunneling variable-resistor equivalent circuit of Fig. 1, where we consider an ohmic variable-resistor R and a tunneling variable-resistor Rt connected in series. R represents the resistance of the doped dioxide layer and Rt represents the tunneling resistance of the undoped layer of the device. The doped layer acts as a conductor, whereas the undoped layer is a pure insulator. There is therefore a significant difference between the actual values of their resistances, with $Rt \gg R$, which is the reason why our proposed model concentrates mainly on the Rt.

Tunneling resistance Rt is expected to be proportional to the tunnel barrier width L, given the fact that the larger the barrier width, the higher the resulting resistance should be. Also, its value is anticipated to change according to the movement of the boundary between the two materials because of the transport of oxygen deficiencies under positive or negative ac voltage. Thus, any mathematical formulation for Rt could include at least a fitting parameter which would bound the effect of the device's varying geometry on the actual concentration of the oxygen vacancies in either of the sides (doped/undoped) of the TiO_2 film. Furthermore, according to Schiff

[9], Rt is inversely proportional to the product of the voltage-dependent tunneling transmission coefficient, denoted here as T_0, and the electron effective density of states, defined here as N_{eff}, whereas it is exponentially proportional to the tunnel barrier width L. Therefore, its particular mathematical formulation is:

$$Rt(V_M) = \frac{1}{N_{eff}} \times \frac{e^{2\times k_{V_M} L}}{T_{0,V_M}} \tag{3}$$

The voltage dependence of (3), due to the presence of the voltage-dependent parameters T_0 and k, can be translated into a corresponding variation of the tunnel barrier width L; therefore it can be passed to a new voltage-dependent parameter $L_{V_M,t}$ with no significant error implication. In our model, we defined Rt to be described by the following equation, whose graphical representation is demonstrated in Fig. 2(d):

$$Rt(L_{VM}, t) = f_o \frac{e^{2\times L_{VM},t}}{L_{VM}, t} \tag{4}$$

Equation (4) gives the devices resistance (memristance) for a certain restricted range of the state variable L. All unknown material-specific and geometrical issues are contained into the model-fitting constant parameter f_0, whose value has been determined by comparison with experimental results [3]. The qualitative agreement of (3) and (4) verifies our assumption for the exponential dependence of the tunneling resistance on the tunnel barrier width.

In addition, the tunnel barrier width is expected to vary within a restricted valid range, based on the assumption that the switching rate of L is small (fast) below (above) a threshold voltage V_{th}. A heuristic equation $L(V_M, t)$ that qualitatively gives the expected response of the tunnel barrier width as a function of time t and applied voltage V_M is given below, whereas the corresponding graph is shown in Fig. 2(c):

$$L(V_M, t) = L_0\left(1 - \frac{m}{r(V_M, t)}\right) \tag{5}$$

L_0 is the maximum value that L can attain. The term in parenthesis of (5), which contains a voltage-dependent function $r(V_M, t)$ and a fitting constant parameter m, determines the boundaries of the barrier width. The function $r(V_M, t)$, incorporates the assumption for the expected different switching rate of L based on the applied voltage bias discussed above. Particularly, the time derivative of $r(V_M, t)$ is given by the following equation:

$$\begin{aligned} &\dot{r}(V_M, t) = a \times \frac{V_M + V_{th}}{c + |V_M + V_{th}|}, \quad V_M \in [-V_0, -V_{th}) \\ &\dot{r}(V_M, t) = b \times V_M, \quad V_M \in [-V_{th}, +V_{th}] \\ &\dot{r}(V_M, t) = a \times \frac{V_M - V_{th}}{c + |V_M - V_{th}|}, \quad V_M \in (+V_{th}, +V_0] \end{aligned} \tag{6}$$

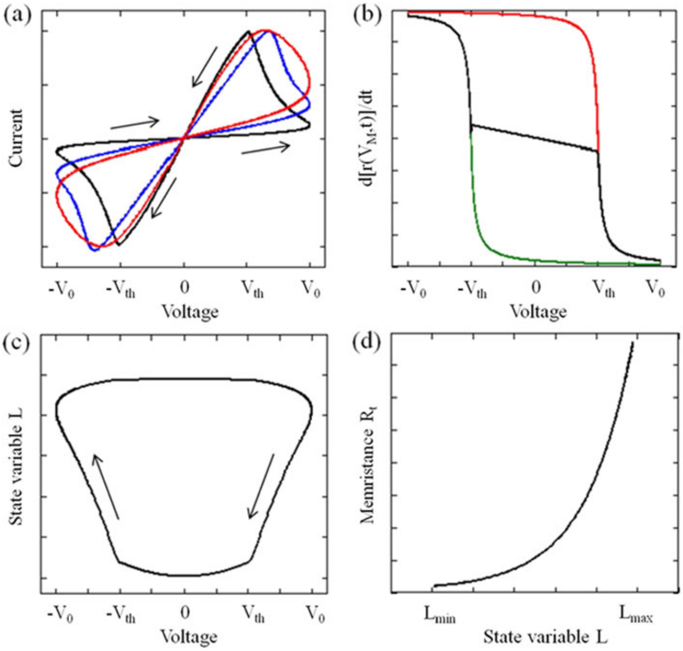


Fig. 2 (**a**) A comparison of the normalized I–V characteristics obtained from a memristor under ac voltage given by the equation $V(t) = V_0 \sin(\omega t)$, with $V_0 = 4$ V and threshold voltage $V_{th} = 2$ V. The models' results are from: the discussed here model (*black*), the model in [3] including a proposed modified window function from [27] (*blue*), and the model in [30] (*red*). (**b**) Graphical representation of (6). In the region $[-V_0, -V_{th})$ the graph (*black*) follows the *green* sigmoid function, whereas in the region $(V_{th}, V_0]$ it follows the red one. (**c**) Response of the state variable L according to (5), and (**d**) the device memristance R_t given by (4) for a restricted range of the state variable L

Several thresholds can be programmed by tuning the shaping parameters of $r(V_M, t)$, namely a, b, and c. Equation (6) comprises one-parameter sigmoid functions for the regions above V_{th} (first and last leg), whereas a linear relation of the applied voltage is used for the region below V_{th}. a, b, and c are fitting constants that define the slope and the magnitude of (6), with $a \gg b$ and $0 < c < 1$. A different set of values for the parameters a, b, c, m defines a different set of boundaries for the tunnel barrier width in (5). Setting b equal to zero imposes a hard switching behavior, i.e. there is no state change in the memristor unless a certain voltage threshold is exceeded. The graphical representation of (6) is shown in Fig. 2(b), where the two sigmoid functions were also included separately to facilitate visual correspondence. It is obvious that in the region $[-V_0, -V_{th})$ the graph (black line) follows the green sigmoid graph, whereas in the region $(V_{th}, V_0]$ follows the red graph. $r(V_M, t)$ is the most important parameter because it defines both the device dynamics and the corresponding state. Its value is monitored at each time step and maintained within a valid defined range; i.e. in cases when $r < r_{min}$ or $r > r_{max}$, it is set equal to r_{min} or r_{max}, respectively. As a consequence, the device memristance is correspondingly set to R_{ON} or R_{OFF}. Equations (5) and (6) bound the fundamental switching dynamics in TiO_2-based memristive devices, correlating the tunnel barrier width L with the applied voltage V_M. However, several memristor devices have been proposed using different material structures, so the resistance switching mechanism is not always due to the change in thickness of a specific material layer. Our model has the potential to describe memristive functionality in a more generalized way if the state variable is normalized between 0 and 1. This can be done by dividing $L(V_M, t)$ with L_0 and by multiplying with the same term the exponent and denominator of (4). Therefore, when $L(V_M, t) \approx 0$ the memristor device is in the most conductive state, and the least conductive state occurs when $L(V_M, t) \approx 1$. This change in state variable represents a generalization of the model so that it can represent more types of memristive devices.

3.3 Verification of the Proposed Model

Figure 2(a) demonstrates a comparison of the normalized I–V hysteretic curves obtained from a memristor under ac voltage bias, between our model and two published device models [27, 30]. The results of our model exhibit the expected "bow tie" shape, and apparently correspond qualitatively to the other models' simulation results, as well as to the experimental I–V curve shown in [3]. In order to illustrate the versatility of our model, we present in Fig. 3 the I–V and M–V (M-Memristance) characteristics as calculated by the presented here model and the model proposed by Joglekar and Wolf [27]. This model is an extension of the linear ionic drift as described by HP [3], where a particular window function is incorporated to illustrate nonlinearities in ionic transport. In order to obtain a fairer comparison, where it applies we use the same parameters for both models. In specific, we use an 8 V peak-to-peak triangular voltage pulse of period $T_1 = 2.6$ s and $T_2 = 5.5$ s to simulate memristors with total width $L_{0,1} = 3$ nm and $L_{0,2} = 5$ nm, respectively. We consider a R_{OFF}/R_{ON} ratio of 10, a dopant mobility of 3×10^{-8} m^2/(Vs) [23] and we set the exponent variable of the corresponding window function $p = 2$ [27]. Figure 3 summarizes the simulation results for both the first (a, b) and the second (c, d) memristor. In each simulation we set our model's parameters $\{a, b, c, f_0, m, |V_{th}|\}$ to the values {1000, 50, 0.1, 86.49, 56.06, 1.7 V} and {350, 20, 0.1, 2.67, 29.97, 1.5 V}, respectively. In both cases our model delivers satisfying quantitative results which coincide with the results from the published model. The small difference in the maximum observed currents is attributed to the slightly different moments when the maximum memristance is achieved, particularly shown in Fig. 3(b), (d).

Moreover, Pickett et al. in [41] report on experimental results from the application of a dynamical testing protocol applied to a set of TiO_2-based memristive devices. Through analysis of the switching dynamics that arise from ionic motion in the devices, it is concluded that electronic conduction in these devices is dominated by an effective tunneling barrier width that varies with time under the applied voltage. Thus, the switching effect is primarily attributed to an effective tunneling

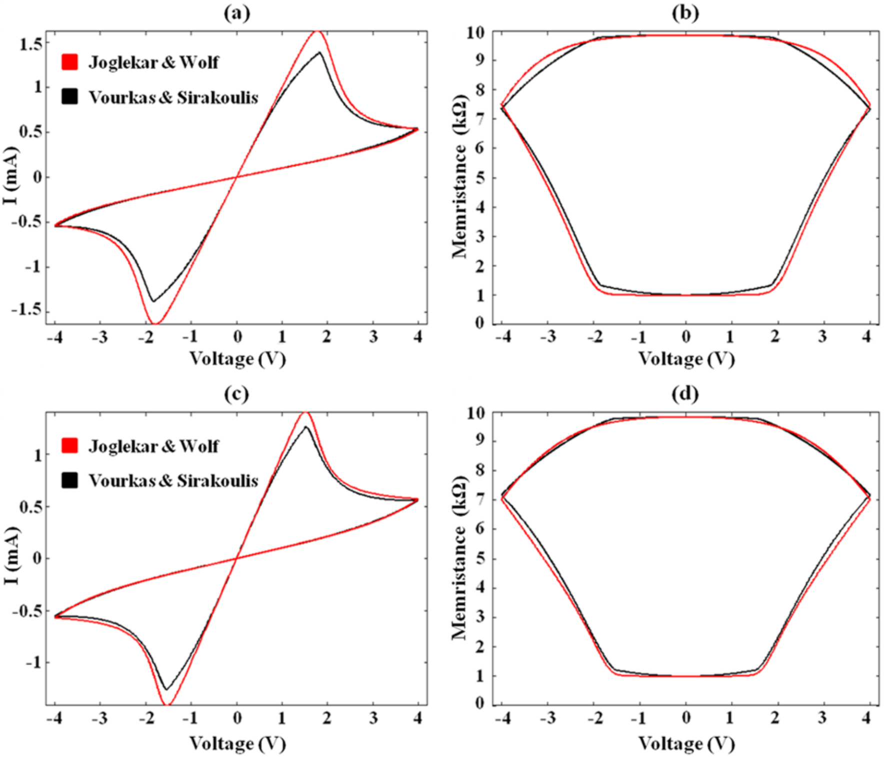


Fig. 3 Calculated I–V (**a**), (**c**) and M–V (**b**), (**d**) characteristic responses of memristors with $L_{0,1} = 3$ nm and $L_{0,2} = 5$ nm, for a 8 V peak-to-peak triangular voltage pulse of period $T_1 = 2.6$ s and $T_2 = 5.5$ s, respectively, according to the presented here model and the model of Joglekar & Wolf [27]. Our model successfully reproduces the characteristic responses by setting $\{a, b, c, f_0, m, |V_{th}|\}$ to the values $\{1000, 50, 0.1, 86.49, 56.06, 1.7\text{ V}\}$ and $\{350, 20, 0.1, 2.67, 29.97, 1.5\text{ V}\}$, correspondingly

distance modulation, which supports our initial assumptions. Therefore, although the switching behavior is definitely complex, it has been showed that it is well represented in our model. Compared to other published models, like the HP's model [3], our proposed model provides intuition into these strongly nonlinear dynamical systems, comprising simple and well understood equations and avoiding the use of restrictive material-specific parameters [3, 41]. Different value-sets for all fitting parameters, namely $\{a, b, c, f_0, m\}$, provide the capability of simulating TiO_2-based memristive devices with different physical structures and geometries. In addition, our model offers the option for different threshold voltages to be applied to the ON and OFF switching cases respectively, in order to simulate asymmetric dynamical behavior during each case. The programmable thresholds provide the possibility of having different thresholds based on the polarity of the applied voltage. This feature is required to provide a better fit to available characterization data, since it has been experimentally verified that both the state variable motion and the threshold voltages are not equivalent in both directions, i.e. when being forward or reversely biased. Although symmetric behavior is presented here as the default option, various tunneling distance change rates could be attributed to the interaction of the external applied field, the internal field of the concentrated vacancies, and the diffusion, all acting in the same or in the opposite directions according to the applied voltage bias [41].

4 Dynamics of Memristors in Regular Network Connections

The composite behavior of circuits comprising memristors connected in a serial or a parallel manner is analyzed in this section. Being associated with the totally nonlinear behavior of individual memristive elements, circuits of multiple memristors may work in very complicated way, quite difficult to predict, due to the polarity-dependent nonlinear variation in the memory resistance (memristance) of individual memristors. Here we explore the dynamics of regular network geometries containing only memristive devices. We particularly focus on one-dimensional memristive networks with all the devices connected in series or in parallel. Depending on their internal state, their polarity and the device-specific properties represented by the values of the parameters of the model, even such simple compositions of memristors can prove to respond in a much unexpected manner. We employ the model summarized earlier in Sect. 3 and conduct our circuit simulations, with all differential equations numerically solved using a 4th order Runge-Kutta integration method, as it is implemented in [11]. Figure 4 illustrates the response of a single memristor under ac voltage bias according to our model. Model parameter values are used as given in $\{a, b, c, m, f_o, L_o, |V_{th}|\} = \{5 \times 10^3, 0, 0.1, 82, 310, 5, 1\text{ V}\}$ and the resulting resistance ratio is $R_{OFF}/R_{ON} \approx 10^2$ with $R_{OFF} \approx 200$ KΩ and $R_{ON} \approx 2$ KΩ. We note here that, the equations of the model are written in such a way that when $\{a, b\} > 0$ then a positive (negative) voltage applied to the top terminal with respect to the bottom terminal, denoted by the black thick line (see Fig. 5 for the corresponding schematic), always tends to increase (decrease) the memristance. The characteristics demonstrated in Fig. 4 will serve as a reference when studying the composite memristive behavior of multiple devices.

4.1 *Memristors Connected in Series*

Considering a single memristive device as a structural element, we here analyze the behavior of circuit branches with more than one device connected in series. Starting from the smallest configuration, i.e. that which consists of two devices, Figs. 5 and 6(a) present the set of three different possible polarities which the memristors will likely have. More specifically, Fig. 5 presents the simulation results for the composite response of a pair of memristors with the same polarity connected in series, under ac triangular applied voltage. Three possible combinations

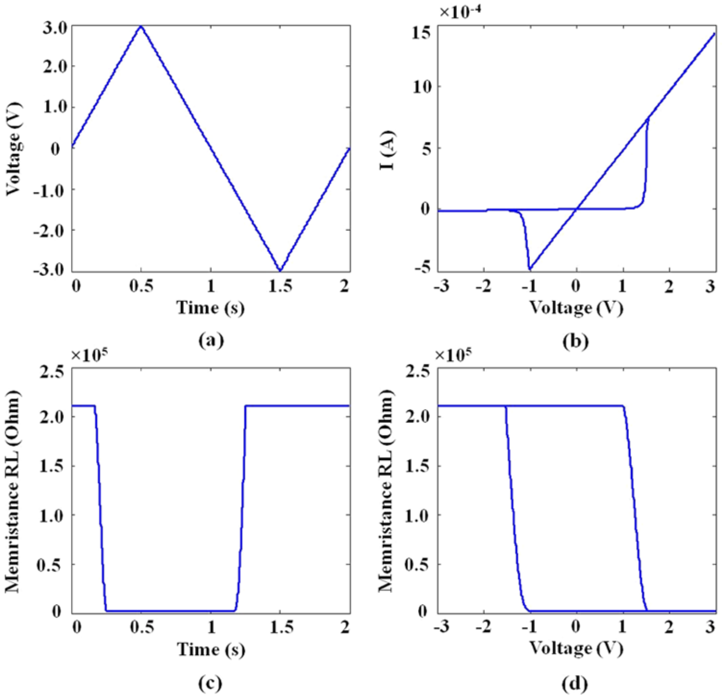


Fig. 4 Memristor model response to a 3 V and 0.5 Hz triangular applied voltage. Model parameters values are used as given in $\{a, b, c, m, fo, Lo, |V\text{th}|\} = \{5 \times 10^3, 0, 0.1, 82, 310, 5, 1\text{ V}\}$ and the resulting resistance ratio is $R_{OFF}/R_{ON} \approx 10^2$ with $R_{OFF} \approx 200\text{ K}\Omega$ and $R_{ON} \approx 2\text{ K}\Omega$. (**a**) The applied voltage as function of simulation time. (**b**) The current-voltage hysteretic characteristic of the memristor. (**c**) The device memristance as function of time, and (**d**) as function of the applied voltage at the terminals of the device

of their initial states are examined, namely the cases (using the following notation to define the placement of the devices as upper/lower) OFF/OFF, OFF/ON, and ON/ON, when both devices are forward polarized. For all initial state configurations we study the following characteristics: current-voltage (I_m–V_m), total resistance-voltage (R_{TOTAL}–V_m), current-time (I_m–t), and total resistance-time (R_{TOTAL}, t), where total resistance (i.e. memristance) is the sum of the individual memristances in the circuit branch.

From Fig. 5(a) one can understand that, when employing devices with identical properties (i.e. equal memristance ratios, switching rates and threshold voltages), if they are placed with the same individual polarity (here both forward polarized), then their overall behavior resembles that of a single memristor whose properties combine the properties of the individual elements. In each situation the connected elements form a voltage divider circuit; thus in the case of being initialized as OFF/OFF, since the devices are absolutely identical, the corresponding voltage drop at each device during the simulation is equal and hence both elements switch from OFF to ON simultaneously when the corresponding voltage at their

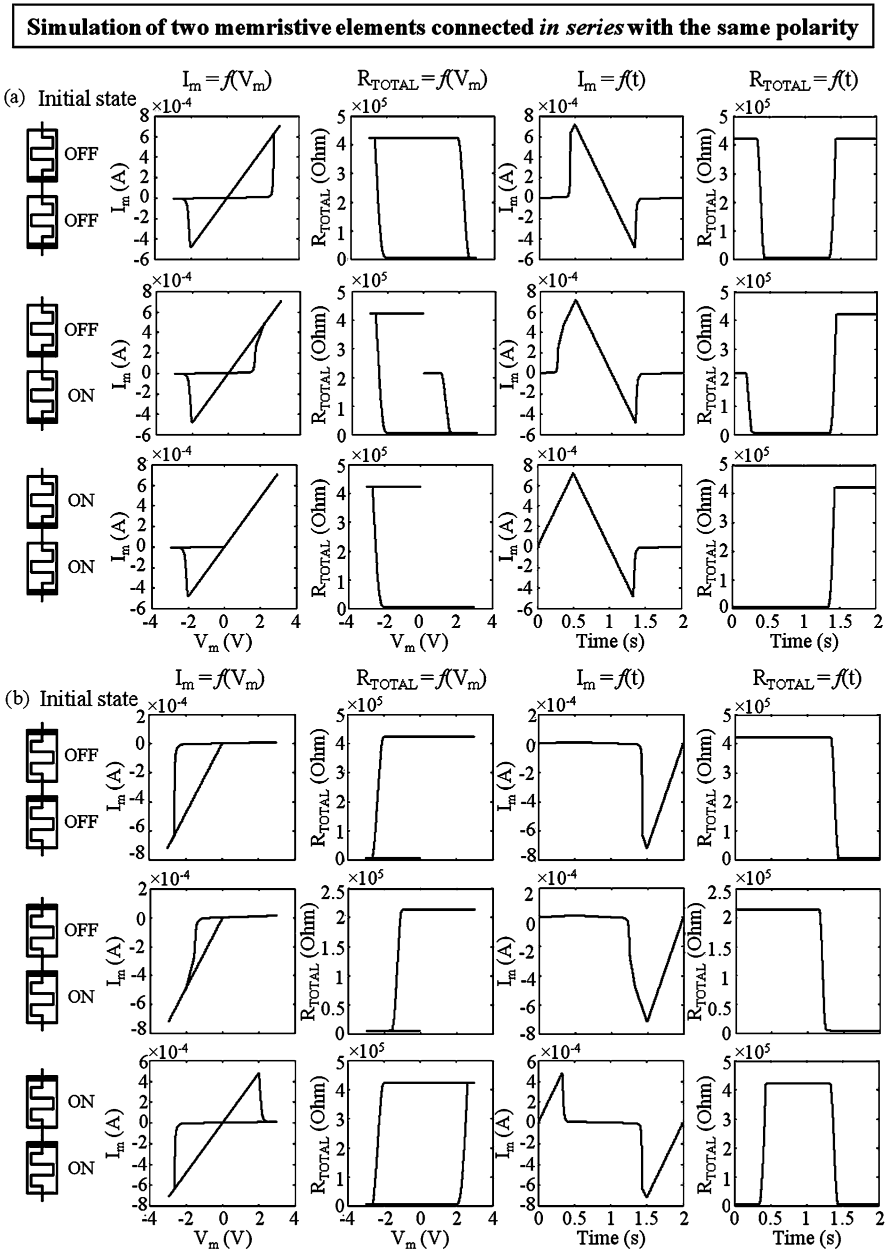


Fig. 5 Simulation results for two memristive elements connected in series with the same polarity when both (**a**) forward biased or (**b**) reversely biased

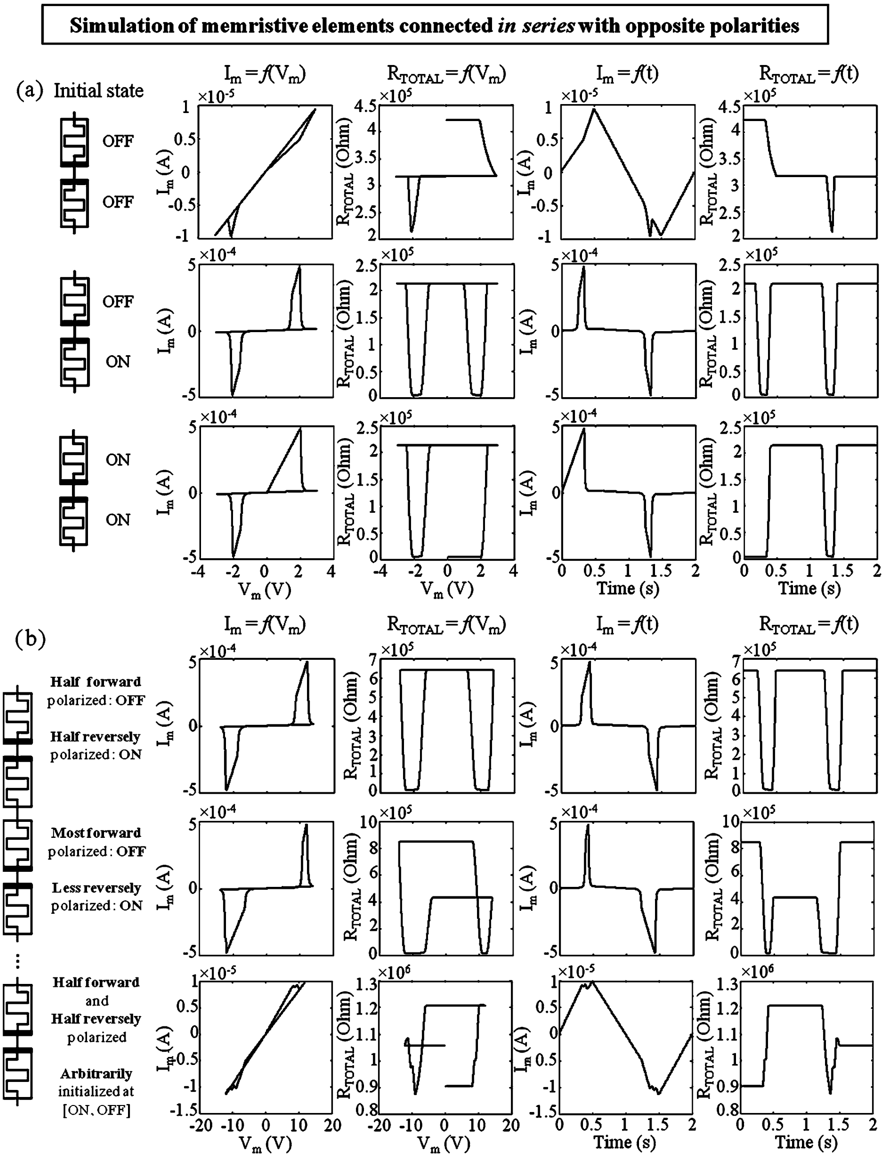


Fig. 6 Simulation results for (**a**) two or (**b**) more than two memristive elements connected in series with opposite polarities

terminals exceeds their threshold value. Therefore, the resulting threshold of the composite device is the sum of the individual thresholds and the composite memristance ranges in the interval [2 × R_{ON}, 2 × R_{OFF}]; both devices are toggled from OFF to ON and vice versa when the total applied voltage exceeds |2 V| and the lobes of the composite hysteretic I–V result smaller, mostly because of the doubled

R_{ON} value. However, depending each time on the different initial states of the individual elements, the applied voltage will either affect their state or it will leave the memristors unaffected, resulting in complicated and polarity-dependent nonlinear behavior. For example, in the case of being initialized as OFF/ON, during the first positive half of the voltage sweep the upper device normally changes its state but this time this happens much sooner and at a lower voltage compared to the previous example. This is due to the high difference between the extreme memristance values which has as a consequence almost the entire applied voltage to correspond to the element which is found at the OFF state; therefore it faster exceeds the threshold value. However, the lower device is not affected at all. It is already found in the ON state, thus under a positive voltage bias it is kept unaffected. Afterwards, during the negative part of the voltage sweep, both devices change from ON to OFF simultaneously. Finally, when initialized as ON/ON, both devices are unaffected by a positive bias (the I–V graph is linear, resembling that of a typical resistor), but normally change to OFF/OFF with a negative voltage of appropriate magnitude.

Figure 5(b) presents the simulation results when the devices are both reversely polarized. Depending on their initial state, individual memristors either change or remain unaffected during the applied voltage sweep. In the OFF/OFF case, both devices are unaffected during the positive half of the ac voltage sweep, consequently a very low current can be observed until the voltage exceeds −2 V when both elements are switched simultaneously. This behavior is similar to that of the last example of Fig. 5(a), only that this time the gradient of the graph is much smaller. In the OFF/ON case in the positive half, given that a reversed bias is applied to the lower device it should have switched from ON to OFF as expected. However, the high memristance ratio of two orders of magnitude results in a very small corresponding voltage drop at this device (consequence of the voltage divider) which never exceeds the threshold value, thus the total resistance is unaffected. Higher selected amplitudes for the applied voltage would have successfully switched this specific element. The final example, where the two devices are initialized in the ON/ON combination, evidently constitutes the opposite case of the first example of Fig. 5(a); thus the overall behavior is that of a composite and reversely polarized memristor.

Figure 6 summarizes the simulation results for two or more than two memristors, this time connected in series while having opposite polarities, studied for a triangular ac applied voltage. We will note that opposite polarities along with different initial states cause highly nontrivial composite responses to the applied voltages. Three possible combinations of the initial states of the individual elements are examined. When studying the behavior of circuit branches comprising only two elements in Fig. 6(a), the same voltage pulse with Fig. 5 is used and the devices are found at the following initial states: OFF/OFF, OFF/ON, and ON/ON. From the content of Figs. 4 and 5 it was figured out that single memristors with opposite polarities present a flipped I–V characteristic and generally demonstrate reversed behavior to the applied signals; in brief, a positive applied voltage tends to switch a forward polarized (reversely polarized) device from OFF to ON (from ON to OFF). Therefore, during a single period of the applied ac voltage, the devices will be likely changing their states in a reciprocal way. In the first case where the devices are initialized as OFF/OFF, the positive applied voltage aims to switch the upper device from OFF to ON, whereas the lower placed device will remain where it is. However, during the switching process, the lower the memristance of the upper device gets, the smaller the corresponding voltage drop in the formed voltage divider becomes, thus the switching is never completed and this can be noted in both the I–V and the R_{TOTAL}–t graphs. Next, the negative voltage sweep tends to change to lower device from OFF to ON and the upper device from its last intermediate state to the OFF state, where it was at the beginning of the simulation. The spike-like change of the total memristance is due to the fact that, first the lower device begins its switching because its higher resistance has as a consequence the major part of the total applied voltage amplitude to correspond to this device. Nevertheless, likewise happened with the positive voltage sweep, during the switching the resistance of the lower element falls enough so that the voltage drop that corresponds to the upper device exceeds the voltage threshold and, as a result, restores it to the OFF state. Thereupon, the voltage drop at the terminals of the lower device never reach again its threshold and thus the total memristance is maintained unaffected until the end of the voltage sweep.

As far as the second possible configuration is concerned, we have the individual elements connected in an antiserial manner and initialized in the states OFF/ON, making the well-known complementary resistive switch (CRS) [42] which was proposed as a possible solution to the current sneak-path problem of large passive crossbar memory arrays. In the CRS concept, a memory cell is formed by two bipolar memristive elements, vertically stacked in an antiserial configuration on top of each other. The first feature that appears from the CRS simulation is a perfectly symmetric I–V curve out of an asymmetric memristor I–V curve. We here illustrate a typical I–V behavior of a simulated CRS switch after preprogramming the individual devices into the aforementioned state prior to further processing. First a positive voltage is applied creating the necessary conditions to either change the state of the lower placed device to OFF or to change the state of the upper element to ON. As it can be seen in the corresponding graph, when voltage reaches a particular point, the state of the upper device changes first and the current rises to very high values until the other device finally switches to the OFF state. At this point, the initial state configuration OFF/ON of the CRS has been flipped to ON/OFF. Next, the flipped CRS exhibits an ohmic behavior until the voltage reaches a specific negative value, when the lower placed element first changes to the ON state. As the negative voltage sweep continues, the upper device is also flipped and the CRS continues exhibiting again ohmic behavior until the end of the voltage sweep. In the resulting I–V characteristic of the CRS the current is linear with the voltage except in two finite voltage intervals. In the final example the devices are both initialized in the low memristance state, i.e. ON/ON. Therefore, in the positive part of the sweep the upper device is unaffected and only the lower device is switched, whereas during the rest of the ac sweep the composite behavior is the same with the previously studied case, so at the end of simulation the devices are found at the OFF/ON combination.

Up to this point we have thoroughly examined the dynamic behavior of circuit branches comprising at most two devices in serial or antiserial configurations.

However, appropriate functionality of memristive elements when they are combined and/or introduced to larger circuits proves quite intriguing and it has been also explored here. In Fig. 6(b) we employ more devices (in particular we consider series of six or ten devices) while appropriately adjusting the voltage amplitude to finally examine their overall response when: (1) three out of six (i.e. half of them) devices are found forward polarized and in the OFF state and the other three are reversely polarized and found in the ON state, (2) four out of six devices are forward polarized in the OFF state and the other two are reversely polarized in ON state, and (3) five out of ten (i.e. half of them) devices are found forward polarized and the rest are reversely polarized, when all being arbitrarily initialized at intermediate states within the interval [ON, OFF]. The response of the first circuit resembles absolutely that of the CRS demonstrated before, with the only difference that the two high-conduction intervals are dragged horizontally in the V-axis. Here the transition begins when the applied voltage exceeds the resulting accumulated threshold value of the devices with the same polarity. Also, since the total R_{ON} of the branch is here three times larger, the highest measured current is found smaller but this difference is infinitesimal and can hardly be detected in the provided graphs because of the high R_{OFF}/R_{ON} ratio. This is a good example showing how groups of individual devices can be effectively combined to deliver composite structures that produce combinatorial complex behavior. Likewise, when having most of the devices forward polarized and initialized as OFF, we see that we can selectively widen and shorten specific lobes and dominate the duration of the high-conduction intervals at will, maintaining the same operation. It is worth noticing that the total resistance, after the voltage sweep is completed, always returns to its initial state, thus the examined device combinations guarantee stable function.

In the final case of our study concerning in-series connected memristors invokes the arbitrary initialization of the devices according to the uniform distribution, to any possible intermediate memristance within the interval $[R_{ON}, R_{OFF}]$. We employ ten devices to increase the total complexity and the variety of possible resistance switching events, with half of them forward and half reversely polarized, whereas we correspondingly increase the applied voltage amplitude to make sure that it will initiate various switching events. The actual initialization is not shown here but the amount of switching events can be easily noted from the corresponding memristance characteristics. The value of the total memristance at the end of the voltage sweep cannot be foreseen and can take any intermediate value between $n \times R_{ON}$ and $n \times R_{OFF}$, where n is the number of employed elements. Furthermore, after conducting multiple similar simulations, we noted that the resulting I–V is always very similar to the presented one, where the current graph can hardly deviate significantly, presenting always a predominating gradient.

4.2 Memristors Connected in Parallel

Once again, considering a single memristive device as a structural element, we analyze the behavior of circuit branches with more than one device connected, this

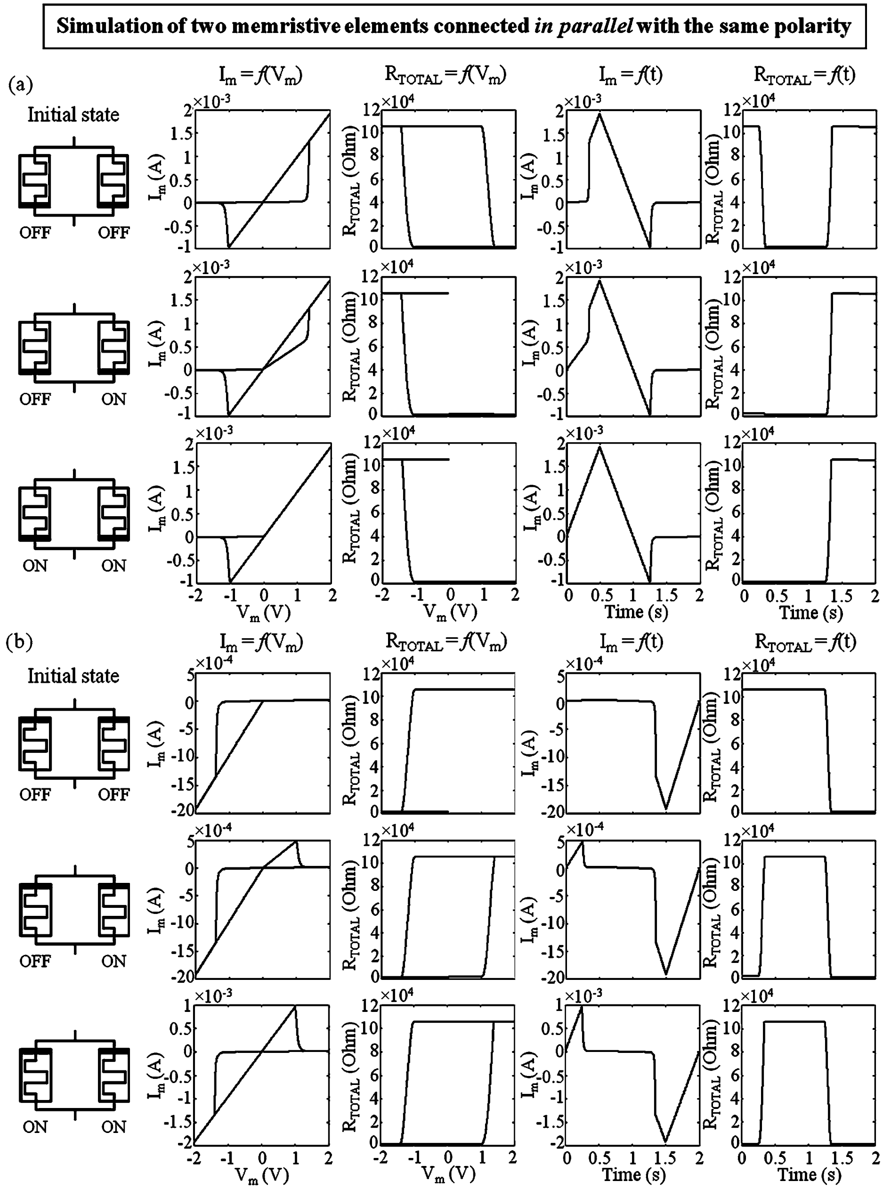


Fig. 7 Simulation results for two memristive elements connected in parallel with the same polarity when both (**a**) forward biased or (**b**) reversely biased

time in parallel. Starting from the smallest configuration which consists of two devices, Figs. 7 and 8(a) present the set of three different possible polarities which the memristors will likely have. More specifically, we are interested in the composite response of a pair of memristors with the same polarity, connected in parallel, under ac triangular applied voltage and examine the same three possible combinations of their initial states, as we did before for the in-series configuration. In this case, since the same voltage is simultaneously applied to all memristors, we do not expect to notice any shift in the threshold voltages that dominate the composite behavior of the memristive combinations, compared to the response of the individual elements. However, unlike the series connection, in the parallel connection the lower resistance values (R_{ON}) will dominate the total resistance of each branch.

In the first case of Fig. 7(a), both devices switch simultaneously from the OFF to the ON state and vice versa, and the resulting I–V resembles that of the single memristor of Fig. 4, only that this time the maximum current is doubled because the total memristance when both devices are found at the ON state is $R_{ON}/2$ ($R_{ON} \parallel R_{ON}$). Therefore, it can be concluded that by connecting identical memristors with the same polarity in parallel, we reproduce the individual memristive behavior and achieve higher total current values. Of course, the higher achieved value for the composite memristance is also lower, but the difference in the current can be hardly noticed because of the high enough selected resistance ratio. In the second case, the influence of having one of the two devices initialized at the ON state can be observed. The initial total memristance is very low and during the positive voltage sweep the device already found at the ON state is not affected, whereas the memristor initially set to the OFF state switches its state as soon as the applied voltage exceeds its threshold. Their composite response during the negative sweep is the same with the previous example of Fig. 7 with both devices switching their states simultaneously to the OFF state. In the last case of Fig. 7, both devices are unaffected by the positive part of the voltage sweep, whereas during the negative part they repeat the aforementioned behavior.

In Fig. 7(b) we examine the composite response of a pair of reversely polarized memristors for the same three scenarios of initial configuration and the simulation results illustrate a very similar composite behavior to the one described before. More specifically, the first case resembles the last case of Fig. 7(a), only that this time during the positive voltage sweep the devices remain unaffected at the OFF state and simultaneously switch to the ON state with a negative applied voltage. In the second case again only one device changes its state, resulting in a verisimilar I–V with the corresponding case of Fig. 7(a), whereas the last case is exactly the same with the first one of the forward polarized devices, with the only difference that the composite device functions in the opposite way.

Having already noticed that identical memristors (or groups of memristors) with opposite polarities can deliver symmetric individual (composite) behavior, it is of great interest to explore their composite response when such devices are connected together. Therefore we next examine the total response of groups of two or more than two devices with anti-parallel configurations. In Fig. 8(a) we investigate the behavior of the smallest configuration which consists of only two devices. Compared to Fig. 7, we notice a significant difference in the overall composite memristance switching; during the simulation and in all demonstrated cases the memristance is kept at low values except for certain intervals which are denoted by spike-like transitions. This is because, as we have concluded before, devices with opposite polarities

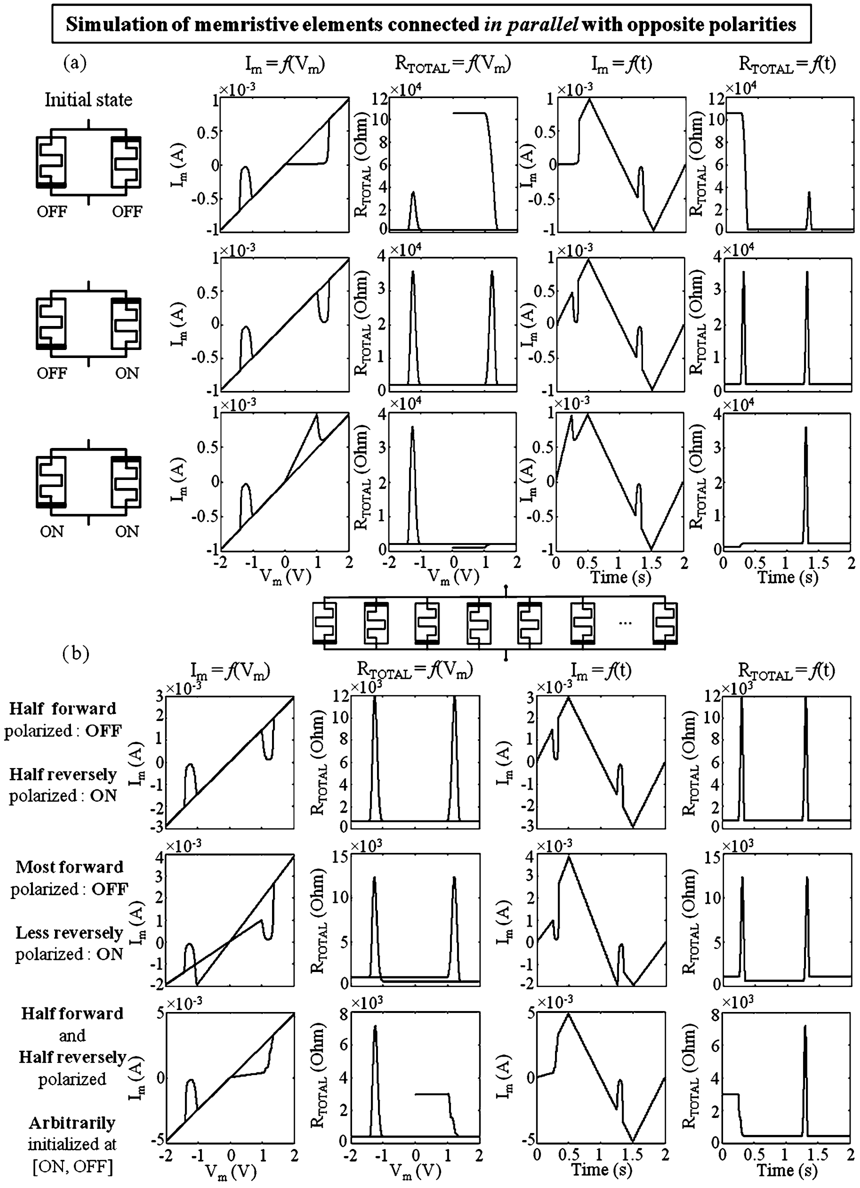


Fig. 8 Simulation results for (**a**) two or (**b**) more than two memristive elements connected in parallel with opposite polarities

have opposite switching characteristics; each time a voltage is applied one of the devices tends to switch to the OFF state and the other to the ON state, respectively. Hence, there will almost always be a device at the ON state, dominating this way the total memristance. It can be seen that, except for the first example where both devices are initially set to the OFF state, the OFF/OFF combination is then found again only as an intermediate state during the state transitions of the devices. Of course, proper selection of the threshold voltages will either broaden or shorten the period of duration of the OFF/OFF combination. The most characteristic case of the ones presented in this figure is the second one, where we have a forward polarized device initialized as OFF and a reversely polarized device initialized as ON, giving an anti-parallel resistive switch (ARS) [43]. The resulting I–V characteristic looks like a truncated Ohm's law; the current is linear with the voltage exclusive of two finite voltage intervals. This behavior is opposite to the I–V characteristic of the complementary resistive switch (CRS) which we described in Fig. 6(a).

Nevertheless, although only a pair of memristor devices was used to illustrate functionality of ARS combinations, the same principles apply for more than two simultaneously connected devices. Depending on their polarity, each time a voltage is applied, the memristors will change their state or they won't be affected at all. The only practical difference lies in the magnitude of the total current observed which depends on the instant combination of the memristances. In Fig. 8(b) we explore the composite response of larger groups of memristive elements connected in parallel. Likewise in Fig. 6(b), in the first case we again employ six devices where half of them are forward polarized at the OFF state and the rest are reversely polarized and initialized at the ON state. It can be observed that the resulting graphs resemble those of the aforementioned ARS combination consisting of two memristive elements. The only difference lies in the measured currents which in this case are larger, as a consequence of the submultiple limits of the composite memristance values. Therefore we can conclude that by introducing more devices to the ARS configuration while maintaining equal the number of forward and reversely polarized devices, we can reproduce the ARS function but with higher currents and smaller composite resistance ratio. In the second case of Fig. 8(b) we notice how we can create multiple dominating gradients in the I–V characteristic by choosing a different distribution between the forward and the reversely polarized elements. We again employ six devices in total with four of them being forward polarized in the OFF state and the other two reversely polarized in ON state. It can be seen that the finite voltage intervals when all devices are found at the OFF state are maintained. However, the initial and the final gradients of the current before and after the aforementioned intervals result different. In the last example of our case study, we employ ten devices with half of them forward polarized and the rest reversely polarized, but this time all being arbitrarily initialized at intermediate states within the range [ON, OFF]. Here we notice that during the first voltage sweep, although the initial state of each device is arbitrary, the circuit branch settles to a particular state close to the lowest possible memristance and then, during the negative voltage sweep, continues functioning normally as a ARS, resembling the previously described behavior but with even higher currents and smaller overall memristance ratio due to the higher number of employed devices.

Up to now we have highlighted the most important characteristics of the composite behavior of memristors when organized in simple one-dimensional networks. In the following section we discuss how the particular ARS operation of multiple

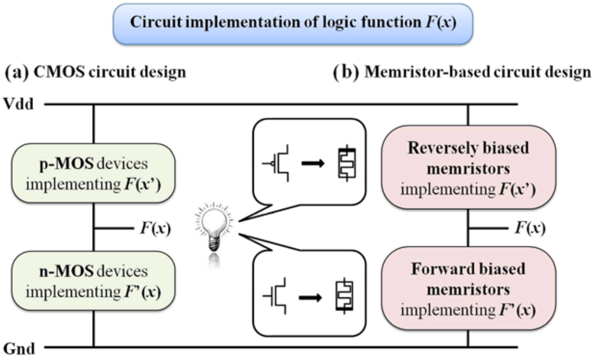

Circuit implementation of logic function F(x)
(a) CMOS circuit design
(b) Memristor-based circuit design
Vdd
p-MOS devices implementing F(x')
F(x)
n-MOS devices implementing F'(x)
Gnd
Reversely biased memristors implementing F(x')
F(x)
Forward biased memristors implementing F'(x)

Fig. 9 CMOS VLSI design method compared to CMOS-like nanoscale design with appropriately polarized memristors replacing FETs in the corresponding pMOS and nMOS areas

memristors can be effectively elaborated and incorporated in a CMOS-like circuit design paradigm for the creation of complementary logic in circuits with memristors, implemented in the nano-crossbar geometry.

5 Circuit Design Paradigm

5.1 Implementation of the Universal Digital Logic Gates

In this section we summarize a novel design paradigm for circuits with memristors, which is based on well-known logic design principles for the CMOS Very Large Scale Integration (VLSI) technology and invokes the anti-parallel resistance switching (ARS) notion.

We present the implementation of the equivalent circuit of the universal digital logic gate set. According to CMOS VLSI design theory, for every logic function $F(x)$ implementation there is a specific formation for the Field Effect Transistors (FETs) circuit. In particular, p-type FETs are located in the upper part of the design plane, implementing the $F(x')$, whereas the n-type FETs are located in the lower part, implementing the $F'(x)$, with the circuit output always taken from between these two parts. Figure 9 depicts the above description and also shows that appropriately polarized memristors can be used to replace existing FETs, maintaining the well understood CMOS-like design methodology [5]. This equivalence is mostly based on the fact that each memristor is considered to be a two-state switch, allowing or preventing the current flow in a circuit branch. It should be mentioned that in every circuit design presented throughout this work, the memristor devices are deliberately shown as three-terminal devices to facilitate comparison with CMOS circuits. In fact, memristors are two-terminal devices and each input signal in the circuit is applied to the particular set of crossbar nanowires which form the junction where the specific memristor is located. Thus, during programming state, applying an input signal simultaneously to all the devices found in a horizontal circuit line corresponds to the application of an appropriate voltage (positive or negative) to the crossbar nanowires which will affect the internal state of the memristor devices.

More specifically, we use forward polarized memristors (FPMs) in the n-MOS area and reversely polarized memristors (RPMs) in the p-MOS area. Also, for the input signals we use conventional notation; namely, we represent logic “1” with positive voltage bias and logic “0” with negative voltage bias on the memristor terminals. The RPMs, corresponding to p-type FETs, are initially found in the R_{OFF} state. Thus a negative voltage changes their state from R_{OFF} to R_{ON}, whereas a positive voltage either restores their state from R_{ON} to R_{OFF} or lets them unchanged in R_{OFF}. On the contrary, the FPMs which correspond to n-type FETs, are initially found in the R_{ON} state. So, a negative sinusoidal voltage changes their state from R_{ON} to R_{OFF}, though a positive sinusoidal voltage either restores their state from R_{OFF} to R_{ON} or lets them unchanged in R_{ON} correspondingly.

For every logic function, the final circuit consists of an equivalent ohmic resistance for the upper and another one for the lower part of the CMOS-like implementation respectively. Therefore the output voltage V_{OUT} is always a fraction of the V_{dd} (voltage divider), with voltage values close to V_{dd} corresponding to logic “1” and values close to zero (Gnd) corresponding to logic “0”. Figure 10 demonstrates the actual circuit implementation of the universal digital logic gates with memristors, whose functionality we explain taking for example the INVERTER (NOT gate). At its initial state, before any applied input, it is $V_{OUT} = V_{dd} \times R_{ON}/(R_{ON} + R_{OFF}) \ll V_{dd}$. After appropriate positive input voltage (logic “1”) the memristors maintain their state, thus we have the same V_{OUT}. However, with a negative input voltage (logic “0”) the previous state of the memristors is changed, so we have $V_{OUT} = V_{dd} \times R_{OFF}/(R_{ON} + R_{OFF}) \approx V_{dd}$. Afterwards, every time a positive input is applied to the gate following the application of a negative one, the state of the memristors is restored to their initial state. Consequently, V_{OUT} changes as expected for any input variation. Likewise, the proposed paradigm works correctly for the rest of the universal digital logic gates, making the design of every digital logic circuit possible.

5.2 *Crossbar Circuit Simulator*

In order to effectively simulate circuits with memristors on crossbar architectures [4, 5], we built a crossbar circuit simulator which utilizes the memristor device model presented earlier in this work, using the Java programming language [11]. The simulator has been properly constructed in order to facilitate the circuit design following the proposed design paradigm, which is based on the CMOS VLSI theory. Snider et al. in [5] presented an approach to building nanoscale computing elements on mosaics of crossbars of configurable FETs and switches, which comprise a set of complementary arrays, i.e. pairs of nFET and pFET arrays supporting a CMOS-like design methodology. Inspired from their particular work and their concept of a fundamental crossbar-based building block, we adopted the basic geometrical features of the proposed logic block, which we then extensively transformed, proposing here our approach to building nanoscale compute fabrics out of configurable memristor

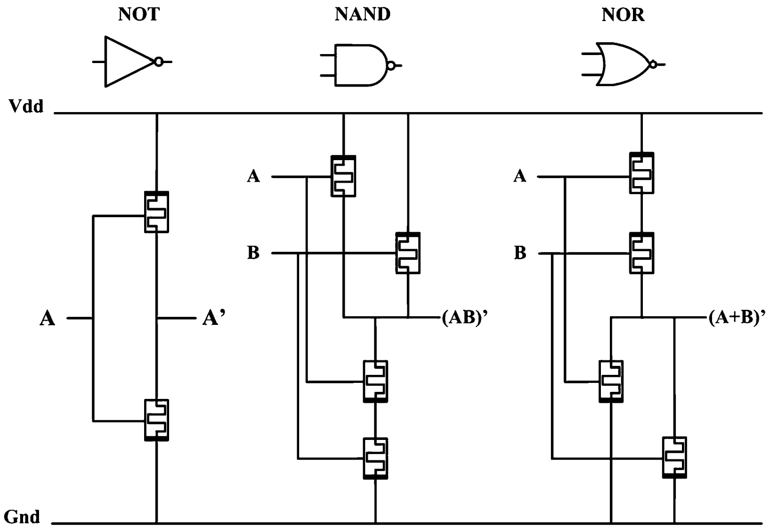


Fig. 10 CMOS-like nanoscale circuit design of the universal digital logic gate set utilizing FPMs and RPMs instead of FETs

switches. Particularly, we maintained the sharp geometry of the array-based architecture and considered appropriately polarized memristor devices to replace nFETs and pFETs.

More specifically, our simulator incorporates a practical user-friendly interface, which comprises a basic logic block (LB) that facilitates the creation of logic with the use of memristor arrays, shown in Fig. 11. The horizontal and vertical lines represent nanowires, with the horizontal lines in one plane and the vertical wires in another. The nanowires are divided into different quadrants of adjustable dimensions, each quadrant possessing different electrical properties due to the nature of the nanowire types in the quadrant and the chemical properties of the interlayer used in that region. The dark gray rectangles at the top represent structures that supply power and ground to the array. The small circles represent connections between the structures in the two different planes. The leftmost vertical wires are electrically connected to the V_{dd} power supply whereas the rightmost vertical wires are electrically connected to the ground. The quadrants at the bottom of the LB represent configurable routing switches; i.e. each junction is normally open, but can be electrically configured to be closed (a low impedance path). In the pink quadrant each junction can be configured to be an RPM, with the configuration input implemented with the horizontal wire and the poles of the device implemented with the vertical wire. In the same way, each of the blue quadrant junctions can be configured to be an FPM.

Every logic function can be implemented in a straightforward manner by decomposing it into two sets of minterms (one set for the RPMs and another for the FPMs), and by selective configuration of junctions in each of the quadrants. For sum-of-products function representations, each product term is implemented with a

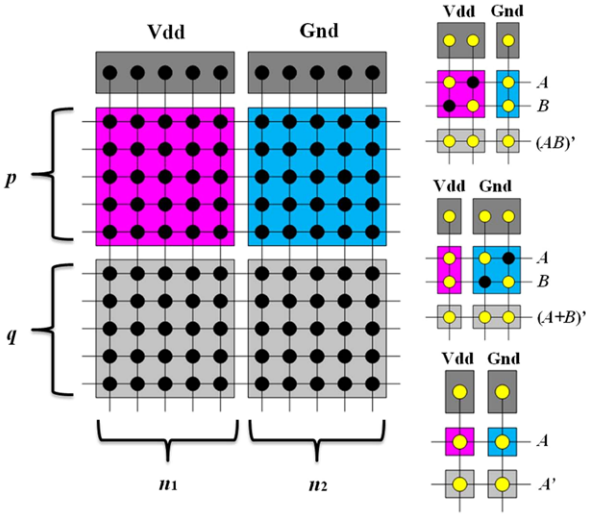


Fig. 11 Basic logic block of the crossbar simulator for circuits with memristors. It comprises two horizontal metal wires for power supply (V_{dd} and Gnd), $p(q)$ horizontal metal wires for inputs (outputs), and $n = n_1 + n_2$ vertical nanowires. On the *right*, the most compact implementation for the set of universal digital logic gates is presented, where *yellow dots* denote currently configured memristors and routing switches

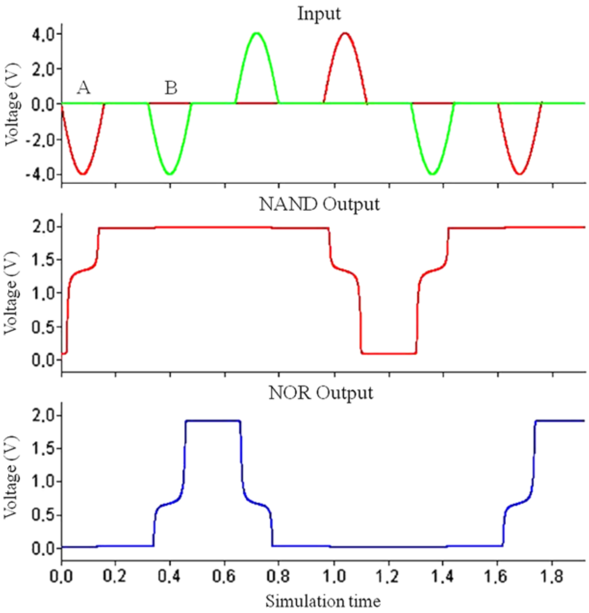
single vertical chain of memristors and the final sum is created by wired-ORing the existing products. A particular input signal may be brought in on any of the horizontal nanowires in the top main two quadrants and it applies to all of the configured memristors found in the same horizontal line. Correspondingly, an output signal may be driven out on any of the horizontal nanowires in the bottom quadrants, on either side of the array, limited only by the ability to allocate junctions within the array to implement the computation function.

5.3 Simulation of Memristor-Based Crossbar Circuits

In this section we demonstrate the results of the simulations conducted using our simulator environment, which utilizes the coupled ohmic-tunneling variable-resistor circuit model for memristors described earlier in Sect. 3. We have successfully simulated the universal digital logic gates, as well as a half adder (HA) and an AND-OR-INVERT function implementation, which are powerful enough to implement universal computation. In our simulations we defined the value margin of L between 1 and 4 nm, and also used a 4 V ac voltage amplitude, getting a satisfactory R_{OFF}/R_{ON} ratio of approximately two orders of magnitude. All differential equations are numerically solved using a 4th order Runge-Kutta integration method, as it is implemented in [11].

Figure 12 shows the output response of a NAND and a NOR logic gate for the corresponding input signals, both implemented with memristors under the CMOS-like design paradigm proposed earlier in Sect. 5. All possible input combinations are presented, starting with the pair $(A, B) = (0, 0)$ and finishing with the same signal values. This is done on purpose in order to demonstrate that the circuit always returns to its initial state. The output voltages should be considered after an input

Fig. 12 Output response of a NAND (*middle graph*) and a NOR (*lower graph*) digital logic gate implemented with memristors, for all possible input variations (*upper graph*) of signals A (*red*) and B (*green*), with $V_{dd} = 2$ V



signal completes its transition; i.e. in the deliberately left wide time gap between consecutive input signal variations.

Figure 13 shows the LB configuration and the circuit design diagram of a HA, under the proposed design paradigm. On the LB figure, red color is used to distinguish the circuit part which corresponds to the *Carry* and green for the part of the *Sum* circuit respectively, with the inverted signals driven from the outside. Also, the output response for *Sum* and *Carry*, which proves the correct functionality of the circuit, is also presented. The simulation begins with the combination $(A, B) = (0, 0)$ and finishes also with the same signal values. This time attention should be paid to the output graph after each inverted signal completes its transition. Furthermore, Fig. 14 demonstrates the circuit diagram for the implementation of an AND-OR-INVERT function, namely $F = (AB + CD)'$, along with the output response for all of the sixteen different input signal combinations. The output graph is successfully confirmed by the function's truth table, also given in Fig. 14.

5.4 Performance Evaluation of Memristor-Based Circuits

Nanowire crossbar is considered one of the most promising circuit solutions for nanoelectronics having many favorable properties, including its periodic geometry and the very compact definition of devices and interconnects, facilitating large scale fabrication and ultra high device density. We next analyze how logic circuits implemented in memristor-based crossbars will perform especially in comparison with MOSFET logic circuits. Evaluation is done with three basic circuit properties,

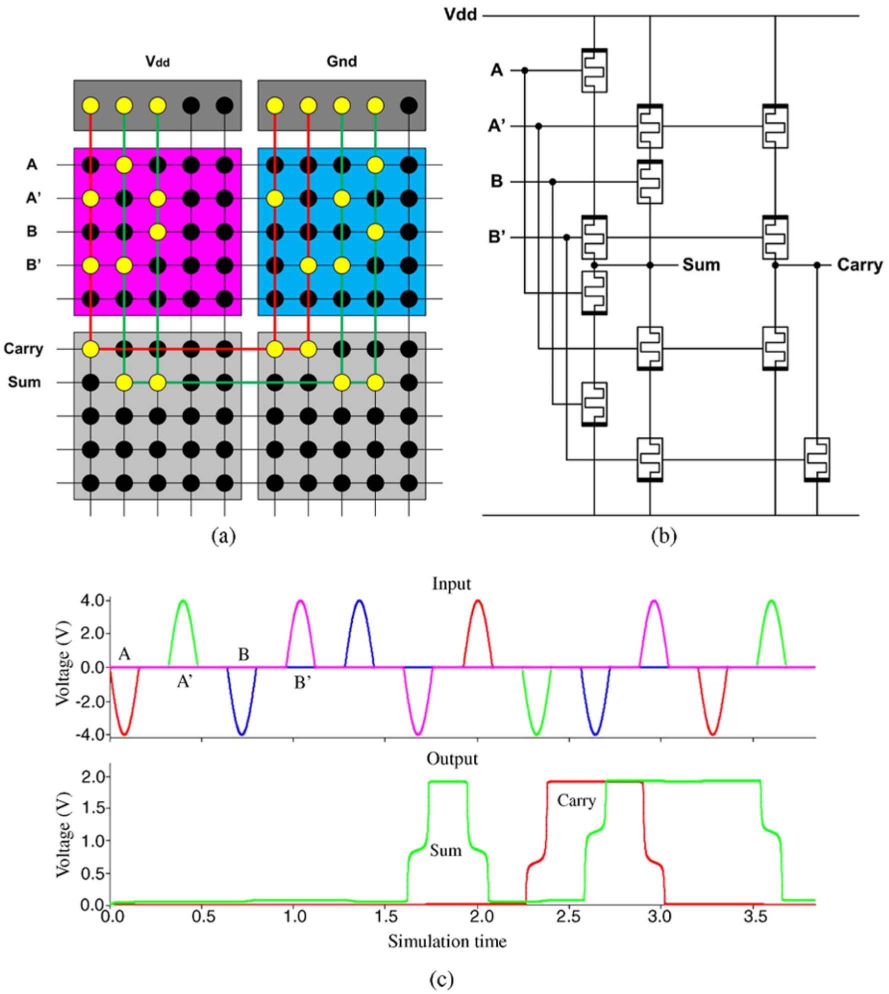
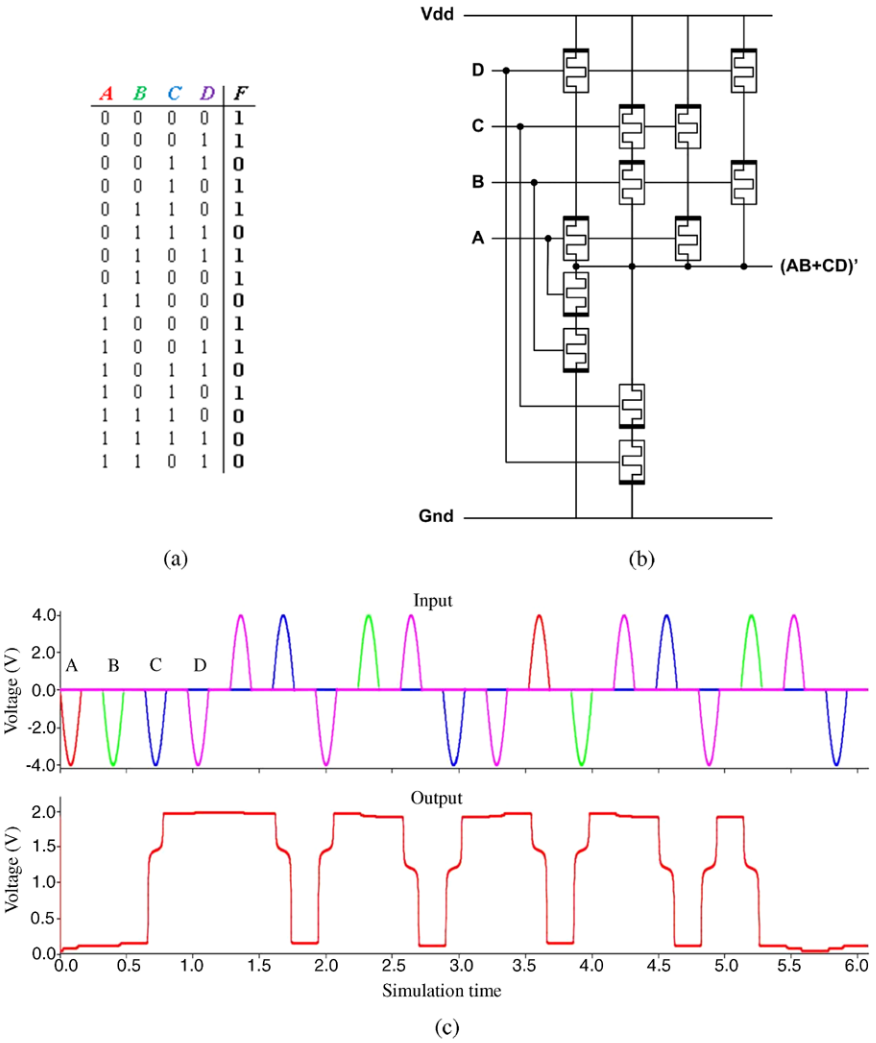


Fig. 13 (**a**) The LB configuration and (**b**) the circuit diagram for the design of a half adder (HA). (**c**) Output response of *Sum* (*green*) and *Carry* (*red*) for all possible input variations of signals *A* (*red*) and *B* (*blue*)

namely area, operation frequency (delay) and energy. Ultra high cross-point density is the major advantage of nanowire crossbar. The generic memristor-based crossbar considered here consists of two horizontal metal wires for power supply (V_{dd} and Gnd), $p(q)$ horizontal metal wires for inputs (outputs), and $n = n_1 + n_2$ vertical nanowires, where n_1 and n_2 is the number of nanowires in each of the orthogonal arrays where RPMs and FPMs are located. If f is the half pitch (minimum feature size) of the process technology, the *area* of a memristor at a cross-point is estimated as small as $4f^2$ [44]. Considering all pitches equal for simplicity reasons, i.e. horizontal power line pitch, input/output line pitch and nanowire pitch, the *area* of any



A	*B*	*C*	*D*	*F*
0	0	0	0	**1**
0	0	0	1	**1**
0	0	1	1	**0**
0	0	1	0	**1**
0	1	1	0	**1**
0	1	1	1	**0**
0	1	0	1	**1**
0	1	0	0	**1**
1	1	0	0	**0**
1	0	0	0	**1**
1	0	0	1	**1**
1	0	1	1	**0**
1	0	1	0	**1**
1	1	1	0	**0**
1	1	1	1	**0**
1	1	0	1	**0**

(a)

(b)

(c)

Fig. 14 (**a**) Truth table of the logic function $F = (AB + CD)'$ and (**b**) the circuit diagram for its implementation. (**c**) Output response for all input variations of signals A (*red*), B (*green*), C (*blue*) and D (*magenta*)

circuit is calculated as follows:

$$Area(p, q, n_1, n_2) = (p + q + 1) \times (n_1 + n_2) \times 4f^2 \tag{7}$$

For example, the set of parameters $\{p, q, n_1, n_2\}$ for the crossbar implementation of NAND and NOR universal logic gates shown in Fig. 11 is $\{2, 1, 2, 1\}$ and $\{2, 1, 1, 2\}$, respectively. Thus, the corresponding area for both of the logic gates is $48f^2$, which is only the 16 % of a four-transistor MOSFET gate's area of $300f^2$ [45].

In order to estimate the operating frequency of the presented here memristor-based circuits, we need the time required to change the state of a memristor (write) from 0 to 1 and 1 to 0, which is directly connected to a voltage source. This time (T_w) was previously derived in [46] and is given by the following equation:

$$T_w = \frac{L_0^2 \times \beta}{2 \times \mu_V \times V_m} \tag{8}$$

β is the achieved resistance ratio of R_{OFF}/R_{ON}, L_0 is the thickness of a memristor, V_m is the magnitude of the applied voltage and μ_v is the mobility of oxygen vacancy dopants. According to this equation, the write time of a memristor is a function of physical parameters of the device and increases with increase in L_0 and β and decreases with increase in the applied voltage. The time required to read the state of a single memristor is not needed for our logic circuits. Introducing in (8) the set of parameters' values considered in our simulations for the presented circuits, utilizing a dopant mobility value also used in [23], namely the set $\{L_0, \beta, V_m, \mu_v\} = \{5\text{ nm}, 100, 4\text{ V}, 3 \times 10^{-8}\text{ m}^2/(\text{Vs})\}$, results in a memristor access time T_w of 10.4 ns. Such a delay is relevant with measured values for switching speed reported in the literature, although higher (lower) dopant mobility can lead to lower (higher) delays respectively. The instantaneous current of the memristor $i(t)$ while writing one or zero depends on the memristor resistance at that instance of time (R_L) and on the applied voltage (V_m). The energy dissipated during access time in each cross-point device can therefore be calculated as:

$$E_w = \int_0^{T_w} V_m \times i(t)d(t) \tag{9}$$

According to data reported in recent literature listed earlier in Sect. 2, measured values for switching speed and energy dissipation per operation for memristors can reach <10 ns and <6 pJ, respectively. Thus, in terms of area and energy consumption, memristor-based circuits are very competitive, delivering also comparable speed with conventional MOSFET-based circuits.

6 Conclusions

This chapter underlines the architectural perspectives that arise from networks of configurable memristors implemented in nano-crossbar platforms. More specifically, a novel circuit model for memristors, which explains the devices memristive behavior by investigating the occurrence of quantum tunneling, and a CMOS-like circuit design paradigm for circuits with memristors, which emerged from the composite behavior of networks of multiple memristors, were presented. We proved the fine application of both by demonstrating the results of various simulations conducted with a properly built crossbar circuit simulator. This work applies well to nowadays technology trends on nanoscale circuits and systems and motivates for further experimental investigation on issues concerning power consumption, operating speed and circuit reliability. Detailed study related to application of this work to many emerging applications of memristors, concerning non-conventional computational and logic circuits, nonvolatile memories, neuromorphic computing and reconfigurable analog/digital circuits will be also part of our future investigation.

References

1. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**, 507–519 (1971)
2. Williams, R.: How we found the missing memristor. IEEE Spectr. **45**(2), 28–35 (2008)
3. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
4. Heath, J.R., Kuekes, P.J., Snider, G.S., Williams, R.S.: A defect-tolerant computer architecture: opportunities for nanotechnology. Science **280**, 1716–1721 (1998)
5. Snider, G.S., Kuekes, P.J., Williams, R.S.: CMOS-like logic in defective, nanoscale crossbars. Nanotechnology **15**, 881–891 (2004)
6. Snider, G.S., Williams, R.S.: Nano/CMOS architectures using a field-programmable nanowire interconnect. Nanotechnology **18**, 035204 (2007)
7. Strukov, D.B., Likharev, K.K.: CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices. Nanotechnology **16**, 888–900 (2005)
8. Yang, J.J., Borghetti, J., Murphy, D., Stewart, D.R., Williams, R.S.: A family of electronically reconfigurable nanodevices. Adv. Mater. **21**, 3754–3758 (2009)
9. Schiff, L.I.: Quantum Mechanics, 3rd edn. Int. Series in Pure. and Appl. Physics, pp. 100–104. McGraw-Hill, New York (1968)
10. Vourkas, I., Sirakoulis, G.Ch.: A novel design and modeling paradigm for memristor-based crossbar circuits. IEEE Trans. Nanotechnol. **11**(6), 1151–1159 (2012)
11. Easy Java Simulations. http://fem.um.es/Ejs/, Cited 15 June 2013
12. Stan, M.R., Franzon, P.D., Goldstein, S.C., Lach, J.C., Ziegler, M.M.: Molecular electronics: from devices and interconnect to circuits and architecture. Proc. IEEE **91**, 1940–1957 (2003)
13. SIA/Sematech: International Technology Roadmap for Semiconductors (ITRS). http://www.itrs.net, Cited 15 June 2013 (2000)
14. Jo, S.H., Kim, K.-H., Lu, W.: Programmable resistance switching in nanoscale two-terminal devices. Nano Lett. **9**(1), 496–500 (2009)
15. Kim, K.-H., et al.: A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. Nano Lett. **12**, 389–395 (2012)
16. Lu, W., Kim, K.-H., Chang, T., Gaba, S.: Two-terminal resistive switches (Memristors) for memory and logic applications. In: 16th Asia and South Pacific Design Automation Conf. (ASP-DAC 2011), pp. 217–223 (2011)
17. Rahaman, S.Z., et al.: Excellent resistive switching memory: influence of GeOx in WOx mixture. In: VLSI Technology Int. Symp. Syst. and Applications (VLSI-TSA 2012), pp. 1–2 (2012)
18. Mondal, S., Her, J.-L., Chen, F.-H., Shih, S.-J., Pan, T.-M.: Improved resistance switching characteristics in Ti-doped Yb2O3 for resistive nonvolatile memory devices. IEEE Electron Device Lett. **33**(6), 1–3 (2012)

19. Yu, S., Wu, Y., Jeyasingh, R., Kuzum, D., Philip Wong, H.-S.: An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. IEEE Trans. Electron Devices **58**(8), 2729–2737 (2011)
20. Kim, H.-D., An, H.-M., Lee, E.B., Kim, T.G.: Stable bipolar resistive switching characteristics and resistive switching mechanisms observed in aluminum nitride-based ReRAM devices. IEEE Trans. Electron Devices **58**(10), 3566–3573 (2011)
21. Wong, H.-S.P., et al.: Metal-oxide RRAM. Proc. IEEE **100**(6), 1951–1970 (2012)
22. Ebong, I.E., Mazumder, P.: Self-controlled writing and erasing in a memristor crossbar memory. IEEE Trans. Nanotechnol. **10**(6), 1454–1462 (2011)
23. Eshraghian, K., et al.: Memristor MOS content addressable memory (MCAM): hybrid architecture for future high performance search engines. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **19**(8), 1407–1416 (2011)
24. Strukov, D.B., Williams, R.S.: Four-dimensional address topology for circuits with stacked multilayer crossbar arrays. Proc. Natl. Acad. Sci. **106**(48), 20155–20158 (2009)
25. Vontobel, P.O., et al.: Writing to and reading from a nano-scale crossbar memory based on memristors. Nanotechnology **20**(42), 425204 (2009)
26. Williams, R.S.: Finding the missing memristor. Keynote talk at UC San Diego CNS Winter 2010 Research Review. http://cns.ucsd.edu/files_2010/january_2010/agenda2010winterreivew.pdf, Cited 15 June 2013
27. Joklekar, Y.N., Wolf, S.J.: The elusive memristor: properties of basic electrical circuits. Eur. J. Phys. **30**, 661–675 (2009)
28. Strukov, D.B., Borghetti, J.L., Williams, R.S.: Coupled ionic and electronic transport model of thin-film semiconductor memristive behavior. Small **5**(9), 1058–1063 (2009)
29. Strukov, D.B., Williams, R.S.: Exponential ionic drift: fast switching and low volatility of thin-film memristors. Appl. Phys. A, Mater. Sci. Process. **94**, 515–519 (2009)
30. Di Ventra, M., Pershin, Yu.V., Chua, L.O.: Circuit elements with memory: memristors, memcapacitors and meminductors. Proc. IEEE **97**(10), 1717–1724 (2009)
31. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**, 209–223 (1976)
32. Pershin, Yu.V., Di Ventra, M.: Practical approach to programmable analog circuits with memristors. IEEE Trans. Circuits Syst. I, Regul. Pap. **57**(8), 1857–1864 (2010)
33. Shin, S., Kim, K., Kang, S.: Memristor applications for programmable analog ICs. IEEE Trans. Nanotechnol. **10**(2), 266–274 (2011)
34. Liu, L., et al.: Engineering oxide resistive switching materials for memristive device application. Appl. Phys. A, Mater. Sci. Process. **102**(4), 991–996 (2011)
35. Xu, N., et al.: Characteristics and mechanism of conduction/set process in TiN/ZnO/Pt resistance switching random-access memories. Appl. Phys. Lett. **92**, 232112 (2008)
36. Yakopcic, C., Taha, T.M., Subramanyam, G., Pino, R.E.: Memristor SPICE modeling. In: Advances in Neuromorphic Memristor Science and Applications. Springer Series in Cognitive and Neural Systems vol. 4, pp. 211–244 (2012)
37. Ziegler, M.M., Stan, M.R.: CMOS/nano co-design for crossbar-based molecular electronic systems. IEEE Trans. Nanotechnol. **2**, 217–230 (2003)
38. Yan, H., et al.: Programmable nanowire circuits for nanoprocessors. Nat. Lett. **470**, 240–244 (2011)
39. Xia, Q.F., et al.: Memristor-CMOS hybrid integrated circuits for reconfigurable logic. Nano Lett. **9**, 3640–3645 (2009)
40. Borghetti, J., et al.: A hybrid nanomemristor/transistor logic circuit capable of self-programming. Proc. Natl. Acad. Sci. **106**(6), 1699–1703 (2009)
41. Pickett, M.D., et al.: Switching dynamics in titanium dioxide memristive devices. J. Appl. Phys. **106**, 074508 (2009)
42. Linn, E., Rosezin, R., Kugeler, C., Waser, R.: Complementary resistive switches for passive nancrossbar memories. Nat. Mater. **9**(5), 403–406 (2010)
43. Liu, T., Kang, Y., Verma, M., Orlowski, M.K.: Witching characteristics of antiparallel resistive switches. IEEE Trans. Electron Device Lett. **33**(3), 429–431 (2012)

44. Kavehei, O., Al-Sarawi, S., Cho, K.-R., Eshraghian, K., Abbott, D.: An analytical approach for memristive nanoarchitectures. IEEE Trans. Nanotechnol. **11**(2) (2012)
45. Dong, M., Zhong, L.: Nanowire crossbar logic and standard cell-based integration. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **17**(8), 997–1007 (2009)
46. Ho, Y., Huang, G.M., Li, P.: Dynamical properties and design analysis for nonvolatile memristor memories. IEEE Trans. Circuits Syst. I, Regul. Pap. **58**(4), 724–736 (2011)

Computing Shortest Paths in 2D and 3D Memristive Networks

Zhanyou Ye, Shi Hong Marcus Wu, and Themistoklis Prodromakis

Abstract Global optimisation problems in networks often require shortest path length computations to determine the most efficient route. The simplest and most common problem with a shortest path solution is perhaps that of a traditional labyrinth or maze with a single entrance and exit. Many techniques and algorithms have been derived to solve mazes, which often tend to be computationally demanding, especially as the size of maze and number of paths increase. In addition, they are not suitable for performing multiple shortest path computations in mazes with multiple entrance and exit points. Mazes have been proposed to be solved using memristive networks and in this paper we extend the idea to show how networks of memristive elements can be utilised to solve multiple shortest paths in a single network. We also show simulations using memristive circuit elements that demonstrate shortest path computations in both 2D and 3D networks, which could have potential applications in various fields.

1 Introduction

Many combinatorial optimisation problems in graph theory [17], such as the Travelling Salesman Problem, involve deriving the shortest path within networks [29]. Applications of such computations include optimising routing protocols [8], trans-

Z. Ye (✉)
Centre for Bio-Inspired Technology, Department of Electrical and Electronic Engineering,
Imperial College London, London SW7 2AZ, UK
e-mail: zhanyou.ye09@imperial.ac.uk

S.H.M. Wu
Systems Engineering Advancement Research Initiative (SEAri), Massachusetts Institute of
Technology, 77 Massachusetts Avenue, E38-576 Cambridge, MA 02139-4307, USA
e-mail: marcuswu@mit.edu

T. Prodromakis
School of Electronics and Computer Science, University of Southampton, Southampton
SO17 1BJ, UK
e-mail: t.prodromakis@soton.ac.uk

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_24,

portation models [21] and recurrent neural networks [11]. Perhaps the simplest shortest path problem is a traditional maze, where one has to determine the path to the exit of a labyrinth whilst only given the entrance point. However, when there are a larger number of pathways in a maze, this increases the number of solutions. Out of these possible pathways, finding the shortest or least-cost one may not necessarily be straightforward. Many mathematical algorithms have been proposed to solve mazes, such as random mouse or mathematical search algorithms [9, 18]. Such algorithms derive solutions in a sequential fashion, thus solution times can increase exponentially in complex networks.

There are also many innovative methods prescribed to solve mazes using biological and chemical systems, such as amoeboid organisms [19, 20], chemotaxis [28] and chemotactic droplets [16]. However, such methods also suffer from increased time complexity when maze sizes increase. The problem is further exacerbated with the introduction of multiple users to a network, such as a traffic optimisation problem where multiple cars would like to find the shortest travelling path in order to avoid congestion in the network. In this paper, we propose using networks of memristive elements to perform multiple shortest path computations in a given network.

The memristor, short for memory resistor, is a passive two-terminal circuit element capable of altering its resistance based on the input and remember its past dynamics [2]. After the device was postulated by L. Chua, a generalised concept of the memristor was further proposed by Chua and Kang [3], defined as

$$v = R(x)i \tag{1}$$

$$\frac{dx}{dt} = f(x, i) \tag{2}$$

where v represents the voltage, i represents the current and $R(x)$ denotes the instantaneous resistance of the device that changes based on its internal state variable, x [24]. Memristance signatures are also observed in various dissipative systems that support discharge phenomena, such as discharge lamps and biological ion channels [22, 26]. Since its implementation by Strukov et al. [32], the solid-state memristor has been proposed to be of use in various applications such as memory storage [15] and neuromorphic implementations [13, 31].

Memristor networks—several memristors connected in the form of an array—have been postulated to be able to perform complex cortical computing functions [31]. Pershin and Di Ventra have also demonstrated that abstract mazes can be solved in a parallel fashion using memristive networks, a termed coined as *analog parallelism*. They have also shown that all solutions in the maze can be determined and the results are separated in order of path length [23].

In the rest of the paper, we exploit the plasticity of 2D and 3D memristive networks for extrapolating various shortest path solutions via simulations in PSPICE. Our study initiates by deciphering the fundamentals of the network to derive shortest path solutions to a given maze in 2D. We then expand this concept to exhibit how a memristive grid can be used to perform multiple shortest path computations in a network involving several users; the example here is London's Tube Network. Lastly, we show how multiple shortest path problems can be solved using 3D memristive networks.

2 Methodology

The maze or network must first be mapped onto a regular memristive grid. The representative memristive grid is then implemented in MATLAB and the corresponding circuit simulations are performed using PSPICE. Entrance/Exit or Start/End nodes for the circuit simulation are represented by a 1 V DC Voltage source and Ground respectively.

2.1 Memristive Components

In 1971, the memristor was predicted theoretically by L. Chua in his seminal paper [2] but it remained a theoretical abstraction until researchers at Hewlett Packard (HP) Laboratories discovered similar properties while fabricating crossbar-type nano-devices in 2008 [32, 33]. The memristor was postulated based on a mathematical relationship between charge q and magnetic flux $\varphi : d\varphi = Mdq$, where M denotes the memristance, which has the same units as resistance (Ω) and is defined as the resistance across the memristor. By taking the time integrals of q and φ, the non-linear relationship between voltage and current across the memristor is established:

$$v(t) = M(q) * i(t) \tag{3}$$

In Eq. (3), $v(t)$ is the applied bias, $i(t)$ is the current flowing through the memristor and $M(q)$ is the charge-dependent memristance. The simplest abstraction of the memristor is that of a time-dependent resistor [14]:

$$M(t) = \frac{W(t)}{D} * R_{ON} + \left(1 - \frac{W(t)}{D}\right) * R_{OFF} \tag{4}$$

2.2 Memristive Fuse

However, when devices are used in memristive networks for shortest path computations, polarity dependence is not desirable since the direction of current flow cannot always be determined.

It was previously proposed that by connecting two memristors with opposing polarities [12], the non-linear relationships between time integrals of voltage and current can be preserved without any polarity dependence. This new combination of

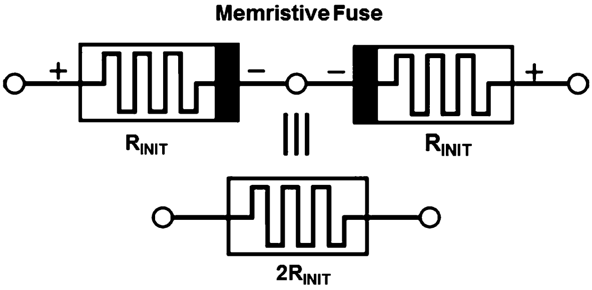


Fig. 1 Two memristors connected with opposite polarities to form a Memristive Fuse [10]

devices is termed the memristive fuse [10], shown in Fig. 1, and is used as the primary memristive device in all the networks in the following simulations. Since the fuse is made of two ideal memristors connected in series, the total initial resistance of the overall device is twice the initial resistance of a single memristor ($2R_{INIT}$).

2.3 SPICE Simulations

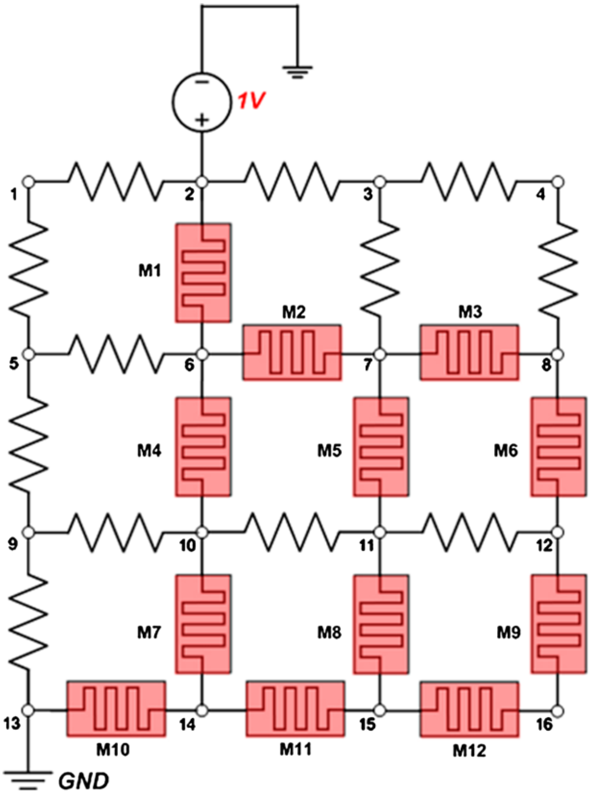
All the memristive networks are simulated using Biolek's Memristor SPICE Model [1] with Prodromakis' non-linear kinetics dopant model [25]. Details about the models can be found in the corresponding references. Throughout this study, the memristor parameters within the SPICE model were defined as follows: Initial Width $W_0 = 5 \times 10^{-9}$ m, Active-Core Thickness $D = 10 \times 10^{-9}$ m, ON Resistance $R_{ON} = 100\ \Omega$, OFF Resistance $R_{OFF} = 16$ kΩ, Net Resistance at $t = 0$ $R_{INIT} = 1000\ \Omega$, Mobility $\mu = 1 \times 10^{-14}$ m^2 s^{-1} V^{-1}.

3 Shortest Path Solution of Mazes Using 2D Memristive Networks

The first simulation shows the varying conductance paths in a memristive network, which correspond to the various solutions to a simple shortest path computation. A simple memristive network is first constructed from a combination of memristive fuses and resistors. Throughout this work we refer to the points where devices are connected together as nodes, while we refer to a branch in the case it comprises one or more devices between two nodes. In Fig. 2, a simple maze is illustrated using a 4×4 memristive network. The paths of the maze are simulated using 12 memristive fuses (labelled M1–M12) and 2 MΩ resistors are used to represent the blocked conductance paths. A 1 V DC Voltage Source and Ground are placed at the nodes corresponding to the entrance and exit of the maze respectively.

The memristive network is simulated for 35 s and the results are shown in Figs. 3 and 4. For each device, the change in memristance ΔM is determined by taking

Fig. 2 4 × 4 Memristive Network consisting of Memristive Fuses and Resistors



the difference between the ‘resistance’ across each branch and the initial resistance ($2R_{INIT}$), which is calculated via:

$$\Delta M_k = \left| \frac{V_X - V_Y}{I_{XY}} \right| - 2R_{INIT} \tag{5}$$

x and y are the nodes connecting each branch and k is the device number. Figure 3 illustrates the transient response of ΔM for devices M4, M5 and M6 against time. This change shows how the devices in different paths respond to the input voltage due to the variance in current amplitudes flowing through them. Figure 4 shows the temporal evolution of memristance of all 12 devices in the network for time instances 1 s, 5 s, 10 s and 30 s. For better visualisation of the change in memristance, ΔM for each device was translated to a linear colour scale of 0–64, where 0 corresponds to zero ΔM and 64 represents the maximum ΔM observed throughout the duration of the simulation.

We first analyse the memristance change of the devices between three branches: Branch 1 (nodes 6 and 14), Branch 2 (nodes 7 and 15) and Branch 3 (nodes 8

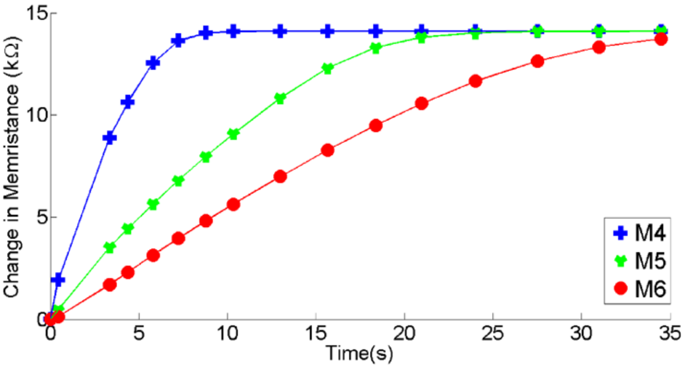


Fig. 3 Plot of Memristance Change (Devices M4, M5 and M6) for time period 0–35 s

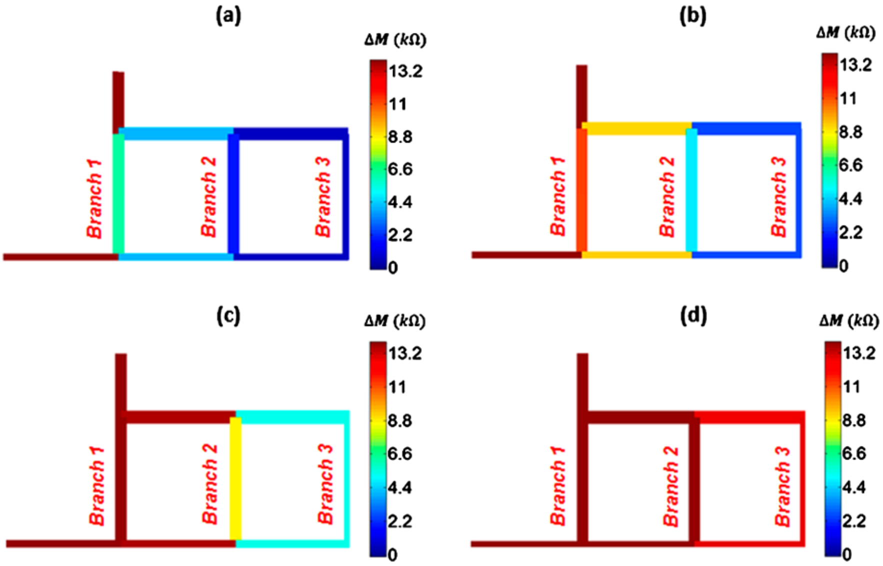


Fig. 4 Spatiotemporal plot depicting ΔM of all Memristive Elements at varying times: (**a**) 1 s, (**b**) 5 s, (**c**) 10 s, (**d**) 30 s. The colour bar on the right show the corresponding ΔMvalues

and 16). Kirchoff's Current Law states:

$$I_{in} = \sum_{i=1}^{n} I_i \tag{6}$$

n refers to the number of branches at the particular node. Applying the formula at nodes 14 and 15, the following relationship regarding the overall current flow across all three branches can be deduced: $I_{B3} < I_{B2} < I_{B1}$.

A larger current flow across a memristor will result in higher rate of change of memristance of the device. Assuming that very little current flows through the 2 MΩ resistors in the grid network, this implies that the ΔM of the devices in all three branches after a short time period will have a similar relationship to that of the total initial current flow across the branches: refer to the change in memristance across devices M4, M5 and M6, shown in Fig. 3 respectively. An increase in the memristance across the devices in Branch 1 will in turn channel more current to Branches 2 and 3. After a stipulated simulation time, all the memristive devices in the network will reach the high resistive state (R_{OFF}).

It can be observed that the shortest path in a maze will exhibit a larger change in memristance over a single period. For the simple maze in Fig. 2, Branch 1 is clearly the shortest path, and the corresponding length of the other two paths (Branch 2 followed by 3) can be identified by comparing ΔM of all devices. We also note the limit of the simulation where all devices reach the high resistive state (R_{OFF}) and the paths are no longer distinguishable by measuring ΔM. Based on the argument that discharge-phenomena support memristive signatures [26], we have reviewed numerous reports on unconventional computation via discharge mechanisms; the most prominent one being [27]. The maze, shown in Fig. 5a, is reproduced by Reyes et al. [27], where the solution is determined using an analog computation method via glow discharge in microfluidic chips (Fig. 5b). The maze is first mapped onto a 15 × 15 memristive grid and the red and blue lines of the grid overlapped onto the maze represent memristive fuses and 2 MΩ resistors respectively, as shown in Fig. 5c. A 1 V DC Voltage Source and Ground are placed at the entrance and exit nodes respectively. Spatiotemporal representation of the change in memristance ΔM of the memristive devices are shown in Figs. 5d, 5e and 5f for the times 2 s, 6 s and 10 s respectively, and the shortest path solution is shown to be identical when compared with the solution derived using microfluidic chips.

This case verifies that the solution to a maze can indeed be determined by mapping it to a memristive grid and placing the source and ground at the entrance/exit nodes. At the same time, this example proves the concomitantly argument presented in [26]: discharge phenomena manifest memristive signatures. By exploiting the analog computations facilitated by Kirchoff's Current Law and that current follows the shortest path to ground, the shortest conductance path will exhibit the largest ΔM. In addition, the altered devices will stay at their given resistive states even after the source and ground nodes have been removed.

In this example, the memristive network converged to a possible solution to the maze after a simulation time of approximately 6–10 s. Nonetheless, this approach is clearly amenable to the use of larger biasing potentials that will in turn speed up the solution. It is interesting to compare this to other mathematical search algorithms performed by a micro-mouse robot; a robot searching for the shortest path in a 16 × 16 unit square maze using either Dijkstra's [6] or Flood-Fill algorithms typically requires 100 s of seconds to accomplish similar tasks [18, 30]. Although actual memristive hardware implementations may yield different solution times from software simulation results, this comparison gives us a scale of the improvements in time complexity by utilising such memristive networks.

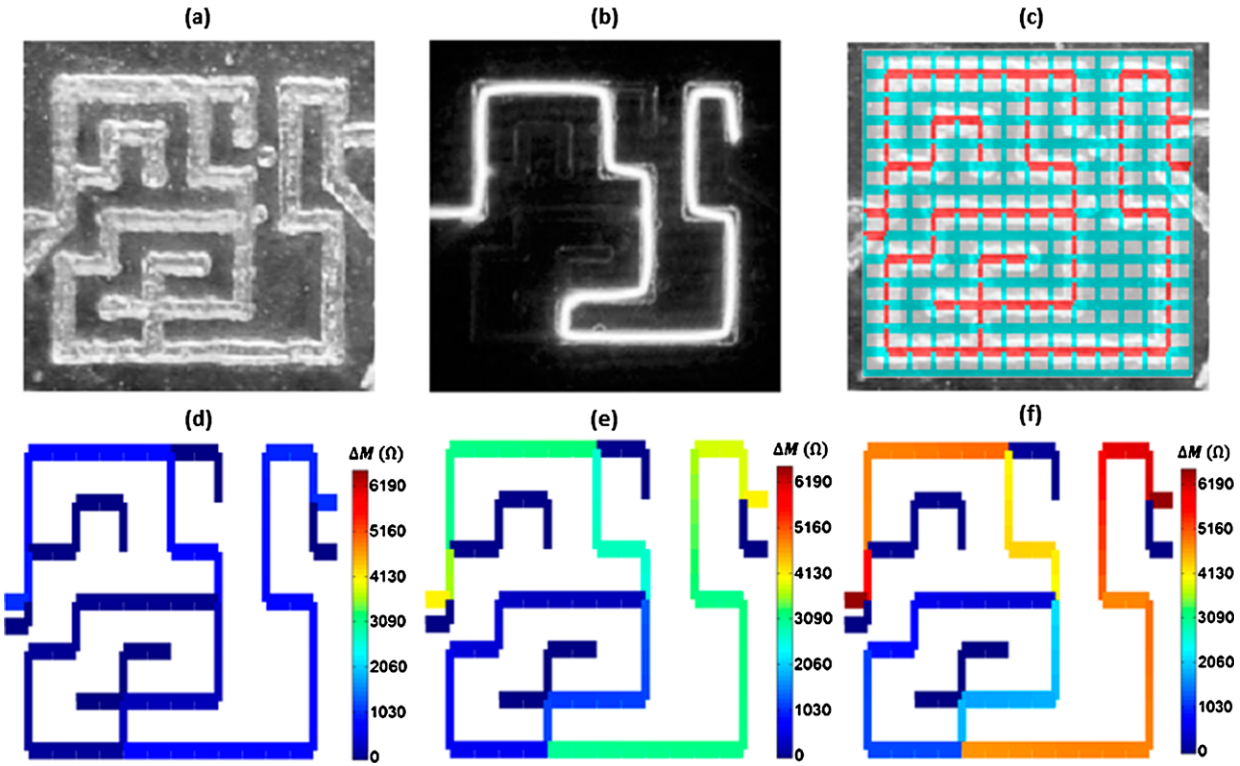


Fig. 5 Maze (**a**) and Solution (**b**) shown by Reyes et al. [27]. (**c**) Mapping of Maze to a 15 × 15 Memristive Grid. Solutions to maze shown by simulations after 2 s (**d**), 6 s (**e**) and 10 s (**f**)

4 Multiple Shortest Path Computations Using 2D Memristive Networks

So far, we have seen the computation of shortest paths for mazes with fixed entrance and exit points. In this section, we further elaborate on the possibility of concurrently solving multiple shortest paths within a same network via an example of travellers determining the shortest path on London's Tube Network. Zone 1 of London's Tube Map, shown in Fig. 6a, is first mapped onto an 18×20 memristive grid, as illustrated in Fig. 6b. Similarly, the red and blue lines on the grid represent memristive fuses and 2 MΩ resistors respectively. The mapping is an approximation of the actual distances and time taken between the tube stations and solely for the demonstration of shortest path computations between stations.

One of the limitations of performing simulations using 2D memristive grids is that each centre node and corner node can accommodate a maximum of four and two paths passing through them respectively. While investigating the Tube Network application, this limitation in paths per node is insufficient for representing some stations such as Green Park, which has six lines going in and out of the station. Hence, in order to increase the number of possible paths through each node without increasing the dimensional space of the network, a 1 Ω resistor is hereby used to link two neighbouring nodes to increase the node size. These extensions are shown as black lines in Fig. 6b and they signify that the two nodes are now effectively the same station. In this scenario, the voltage drop across the resistor is assumed to be negligible since the corresponding resistance is three orders of magnitude smaller than the initial resistance R_{INIT} of the memristors used in the circuit. All starting and destination nodes in the memristive network are simulated using 1 V DC Voltage Sources and Ground.

We first show the shortest path computation in the Tube network for a single traveller wishing to get from Gloucester Road Station to Warren Street Station. Figure 7 shows the corresponding results of the network for the times 3 s, 7 s and 10 s and the shortest path is accurately determined by observing the spatiotemporal plot of ΔM for all the memristive devices in the network. Moreover, we demonstrate how the memristive network computes shortest paths for three travellers, namely Travellers A, B and C in the tube network concurrently. In the first scenario, all three travellers wish to get to the same destination, Holborn from their respective starting stations: A—Paddington, B—Gloucester Road and C—London Bridge. At the circuit level implementation of the memristive network, this translates to three 1 V DC sources at the starting nodes and a single ground placed at the node representing Holborn station. The shortest paths of the three travellers will be termed P_A, P_B and P_C respectively and are shown in the memristive network in Fig. 8.

By comparing the relative ΔM on the spatiotemporal plot shown in Fig. 8, we note that there are two possible shortest paths solutions, P_{A1} and P_{A2} . The number of memristive elements (N) for the two solutions are $N_{A1} = 19$ and $N_{A2} = 20$, as shown in Fig. 8. As the memristive network size increases, the average N increases as well. If the difference in path lengths, $(N_1 - N_2) \ll N_x$ (where $x = 1$ or 2), it will be increasingly difficult to distinguish between two shortest paths using ΔM of the

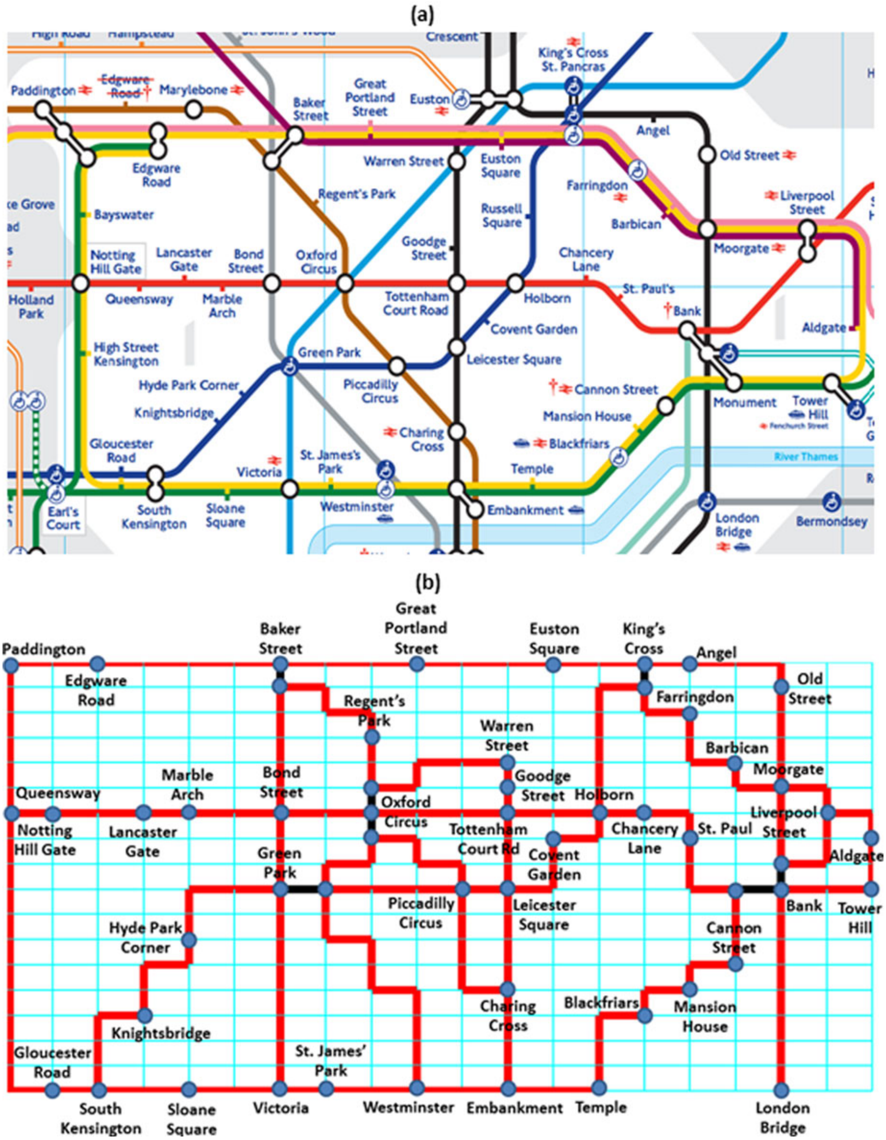


Fig. 6 Zone 1 of London's Tube Network (**a**) is mapped onto an 18 × 20 Memristive Grid (**b**)

devices as $\langle \Delta M_{P1} \rangle \approx \langle \Delta M_{P2} \rangle$, where $\langle \Delta M_{P1} \rangle$ and $\langle \Delta M_{P2} \rangle$ are the average ΔM of the memristive devices in paths 1 and 2 respectively. In the second scenario however, shown in Fig. 9, Travellers A, B and C all have different start and end stations: Traveller A wishes to get from Gloucester Road to Paddington, Traveller B from Hyde Park Corner to Holborn and Traveller C from London Bridge to Old Street. It

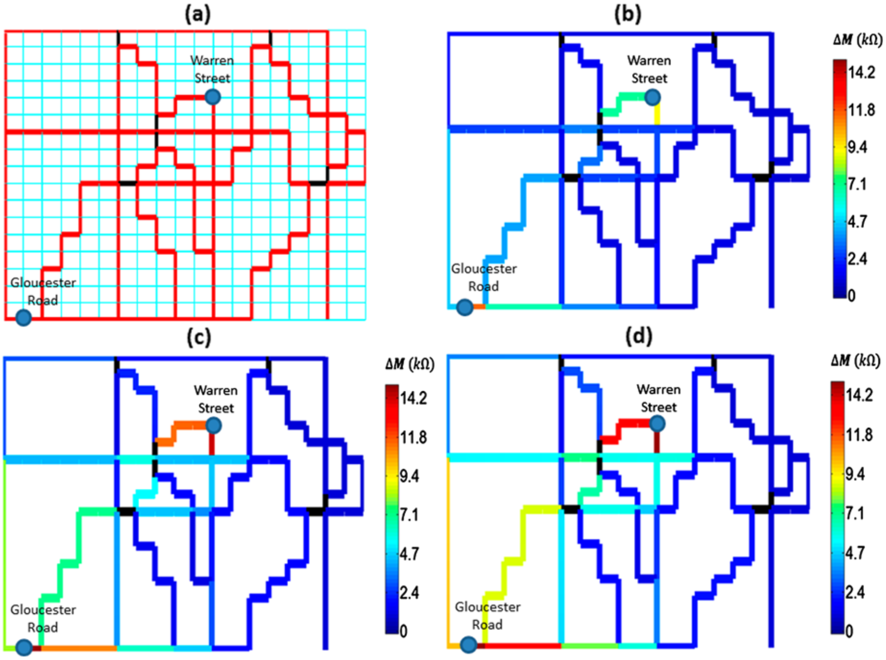


Fig. 7 Traveller A heading from Gloucester Road to Warren Street Station. The respective positions of nodes are shown on the network in (**a**). Solution to the network shown by simulations after 3 s (**b**), 7 s (**c**) and 10 s (**d**)

is noted that all computed paths are unique solutions; there are no overlapping paths between the travellers.

Due to the use of voltage sources at the starting node of the route, the shortest path computed for one traveller will not pass through the starting point of another. This is shown in another example, when Travellers A and B travel from Gloucester Road and Notting Hill Gate to Paddington respectively. As seen from the shortest path computations presented in Fig. 10, the path computed by the memristive network for Traveller A does not pass through Notting Hill Gate station although that path has a lower N value. In the circuit implementation, both station nodes are at high voltage potential, hence resulting in a negligible amount of current flow between them. Even after 10 s, the measured ΔM of the devices between the two nodes is approximately only 2 Ω. The shortest path for Traveller A will essentially be the next alternative path, as marked out by the red arrow in Fig. 10.

This series of cases exhibit that multiple shortest path computations can be performed based on the overall change in memristance due to the current flows in a single 2D memristive network. This has been demonstrated using London's Tube Network, where the shortest paths of three travellers are determined concurrently using a single network. If other known shortest path algorithms such as Dijkstra's [6] were used in this example, routes for the three travellers will have to be determined independently, which increases the time complexity of computation by an

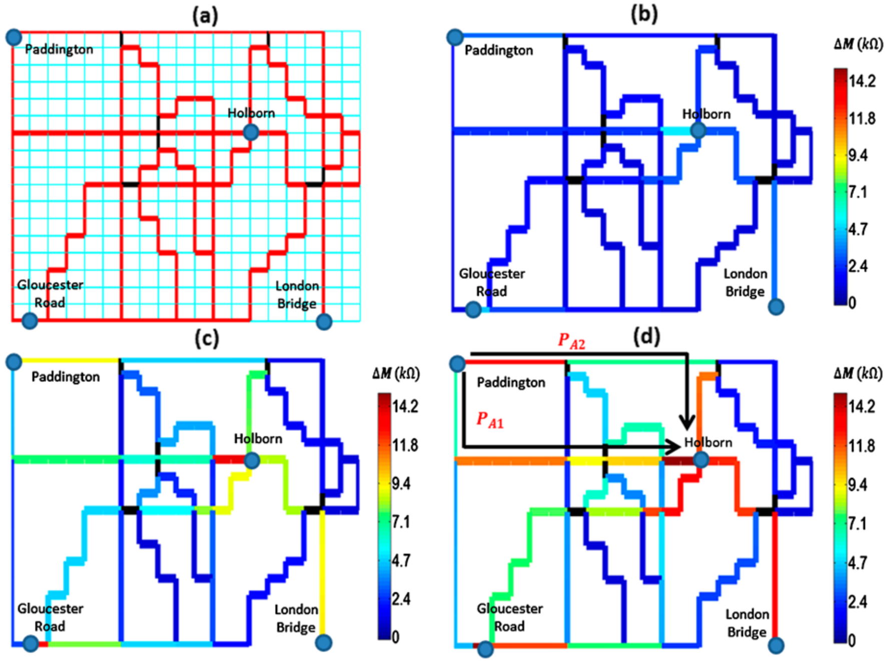


Fig. 8 Travellers A, B and C heading from Paddington, Gloucester Road and London Bridge Stations to Holborn Station. The respective positions of nodes are shown on the network in (**a**). Solution to the network shown by simulations after 1 s (**b**), 5 s (**c**) and 10 s (**d**). *Black arrows* shown in (**d**) indicate the two paths for Traveller A (Paddington to Holborn)

order of the number of travellers there are in a network. By computing the shortest paths in a parallel manner by solving a series of Kirchoff's Current Law equations, the memristive grid is able to compute all shortest paths in a single step. In addition, all the solutions are shown over a fixed time period regardless of the number of travellers in the network.

5 Shortest Path Computations Using 3D Memristive Networks

The limitations using 2D networks are fewer input and output paths per node, in addition to the relatively low spatial resolution that can be achieved. For example, if all the lines in London's Tube Network (Zones 1–5) were to be mapped onto a single memristive network, it will be more accurately performed in 3D, where an additional layer can accommodate overlapping lines in the Tube network. In this section, we describe the computation of shortest paths by employing 3D grids, using a simple maze constructed in a 4 × 4 × 3 3D memristive network with two entrances and a single exit. The paths for the maze, represented using memristive fuses in the corresponding circuit are shown in Fig. 11a as light blue lines, while all static

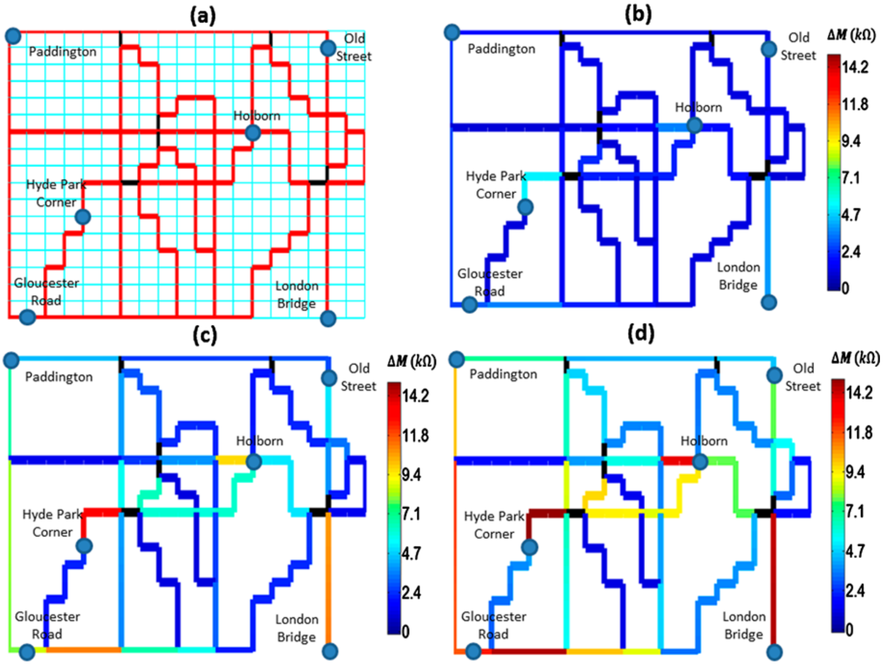


Fig. 9 Travellers A, B and C on three routes: Gloucester Road to Paddington, Hyde Park Corner to Holborn and London Bridge to Warren Street. The respective positions of nodes are shown on the network in (**a**). Solution to the network shown by simulations after 1 s (**b**), 5 s (**c**) and 10 s (**d**)

resistive elements are represented by thin black lines. The corresponding circuit is exploited in a similar manner to the pre-discussed scenarios. The two shortest path solutions of the maze, shown in Figs. 11b, 11c and 11d, are clearly depicted by monitoring the ΔM of all memristive devices.

A 3D network can also be viewed as several 2D arrays stacked onto each other, with the addition of linking elements between the layers. We compare the time complexity of solving a 3D maze if any random mouse algorithm is used [18]. Assuming that the number of vertices and paths in each layer remain the same, the total time complexity for the random mouse method will increase by an order of the number of 2D arrays, including the number of interconnecting paths.

This computation method via 3D memristive networks has been proven to be simple to execute and does not require long computation times. Current shortest path algorithms such as Dijkstra's [6] and Floyd-Warshall's [7] have time complexities of $O(V^2)$ and $O(V^3)$ respectively where V is the number of vertices (nodes) [5]. In comparison to the employed memristive network, the best theoretical estimate for a linear system is the Coppersmith Winograd algorithm [4, 23] which is described as $O(n^{2.376})$ where n is the number of edges in a network. However, it is noted that a memristive network implemented in hardware only has a single overall computation step in order to determine the shortest paths in the network [23]. This makes it

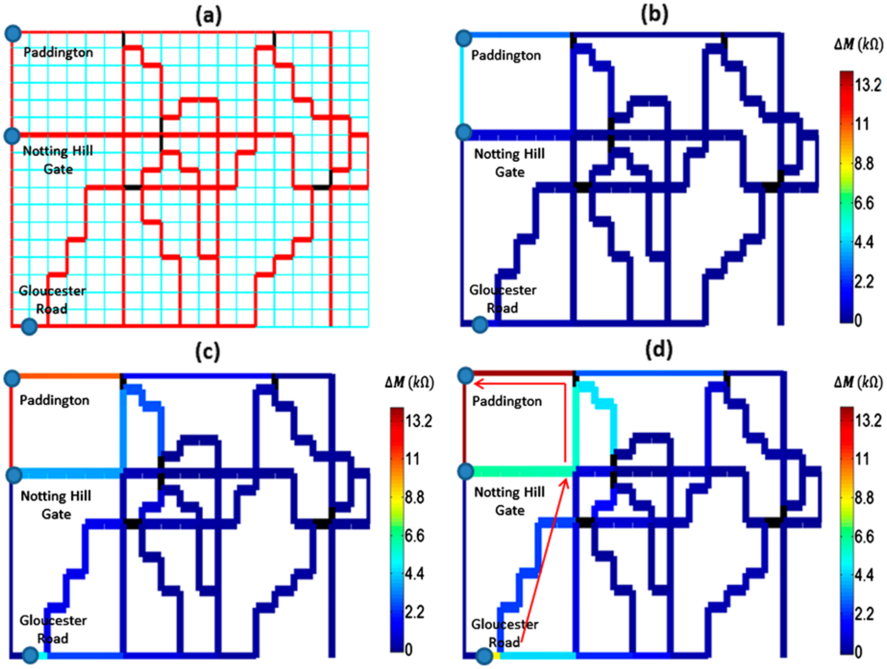


Fig. 10 Travellers A and B heading from Gloucester Road and Notting Hill Gate to Paddington Station. The respective positions of nodes are shown on the network in (**a**). Solution to the network shown by simulations after 1 s (**b**), 5 s (**c**) and 10 s (**d**). The *red arrow* shown in (**d**) indicates the alternate path for Traveller A

more efficient than the algorithms listed above, before even considering multiple computations in a single network.

6 Conclusion

Paths of an existing network can be mapped on a memristive network using a series of memristive devices and resistors. By exploiting the analog computations performed by solving Kirchoff's Current Laws in a parallel manner [23], memristive networks have been shown to be capable of computing shortest paths in a given maze, leveraging on the dynamic adjustment of their intrinsic conductance. This computation method has also been extended to show how multiple computations can be performed. Furthermore, this concurrent solution method can also be exploited to include 3D spaces, where shortest paths through stacks of 2D arrays can be efficiently determined by performing a single step via employing distinct voltage sources and ground terminals to the entrances and exits of the network. Such networks, if implemented in hardware, have great application prospects and can be used to solve many optimisation problems in various fields.

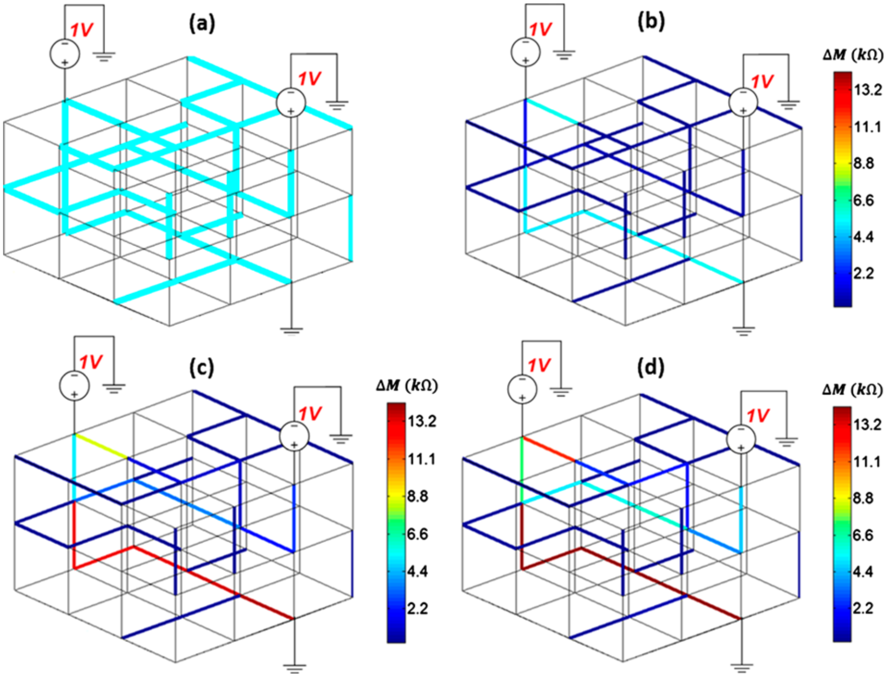


Fig. 11 Maze shown in a 4 × 4 × 3 Memristive Network. Paths of the maze highlighted in *light blue* (**a**), with entrances and exits indicated by 1 V and Ground respectively. Solution to the network shown by simulations after 1 s (**b**), 5 s (**c**) and 10 s (**d**)

Acknowledgements The authors wish to acknowledge the financial support of the CHIST-ERA ERAnet EPSRC EP/J00801X/1 and EP/K017829/1.

References

1. Biolek, Z., Biolek, D., Biolkova, V.: Spice model of memristor with nonlinear dopant drift. Radioengineering **18**, 210–214 (2009)
2. Chua, L.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
3. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
4. Coppersmith, D., Winograd, S.: Matrix multiplication via arithmetic progressions. J. Symb. Comput. **9**(3), 251–280 (1990)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms, 1st edn. MIT Press/McGraw-Hill, Cambridge/New York (1990)
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959)
7. Floyd, R.W.: Algorithm 97: shortest path. Commun. ACM **5**(6), 345 (1962)
8. Fortz, B., Rexford, J., Thorup, M.: Traffic engineering with traditional ip routing protocols. IEEE Commun. Mag. **40**(10), 118–124 (2002)
9. Gallo, G., Pallottino, S.: Shortest path methods: a unifying approach. Netflow Pisa 38–64 (1986)

10. Gelencser, A., Prodromakis, T., Toumazou, C., Roska, T.: Biomimetic model of the outer plexiform layer by incorporating memristive devices. Phys. Rev. E **85**(4), 041918 (2012)
11. Hopfield, J.J., Tank, D.W.: "Neural" computation of decisions in optimization problems. Biol. Cybern. **52**(3), 141–152 (1985)
12. Jiang, F., Shi, B.E.: The memristive grid outperforms the resistive grid for edge preserving smoothing. In: Circuit Theory and Design. European Conference on Circuit Theory and Design, 2009. ECCTD 2009, pp. 181–184. IEEE Press, New York (2009)
13. Jo, S.H., Chang, T., Ebong, I., Bhadviya, B.B., Mazumder, P., Lu, W.: Nanoscale memristor device as synapse in neuromorphic systems. Nano Lett. **10**(4), 1297–1301 (2010)
14. Joglekar, Y.N., Stephen, J.W.: The elusive memristor: properties of basic electrical circuits. Eur. J. Phys. **30**(4), 661 (2009)
15. Kim, H., Sah, M.P., Yang, C., Chua, L.O.: Memristor-based multilevel memory. In: 12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), 2010, pp. 1–6 (2010)
16. Lagzi, I., Soh, S., Wesson, P.J., Browne, K.P., Grzybowski, B.A.: Maze solving by chemotactic droplets. J. Am. Chem. Soc. **132**(4), 1198–1199 (2010)
17. Mani Chandy, K., Misra, J.: Distributed computation on graphs: shortest path algorithms. Commun. ACM **25**(11), 833–837 (1982)
18. Mishra, S., Bande, P.: Maze solving algorithms for micro mouse. In: IEEE International Conference on Signal Image Technology and Internet Based Systems, 2008. SITIS'08, pp. 86–93 (2008)
19. Nakagaki, T., Yamada, H., Toth, A.: Intelligence: maze-solving by an amoeboid organism. Nature (2000)
20. Nakagaki, T., Yamada, H., Hara, M.: Smart network solutions in an amoeboid organism. Biophys. Chem. **107**(1), 1–5 (2004)
21. Pallottino, S., Scutella, M.G.: Shortest path algorithms in transportation models: classical and innovative aspects. Equilib. Adv. Transp. Model. **245**, 281 (1998)
22. Pershin, Y.V., Di Ventra, M.: Memory effects in complex materials and nanoscale systems. Adv. Phys. **60**(2), 145–227 (2011)
23. Pershin, Y.V., Di Ventra, M.: Solving mazes with memristors: a massively parallel approach. Phys. Rev. E **84**(4), 046703 (2011)
24. Prodromakis, T., Toumazou, C.: A review on memristive devices and applications. In: 2010 17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 934–937. IEEE Press, New York (2010)
25. Prodromakis, T., Pin Peh, B., Papavassiliou, C., Toumazou, C.: A versatile memristor model with nonlinear dopant kinetics. IEEE Trans. Electron Devices **58**(9), 3099–3105 (2011)
26. Prodromakis, T., Toumazou, C., Chua, L.: Two centuries of memristors. Nat. Mater. **11**(6), 478 (2012)
27. Reyes, D.R., Ghanem, M.M., Whitesides, G.M., Manz, A.: Glow discharge in microfluidic chips for visible analog computing. Lab Chip **2**(2), 113–116 (2002)
28. Reynolds, A.M.: Maze-solving by chemotaxis. Phys. Rev. E **81**(6), 062901 (2010)
29. Schrijver, A.: On the History of Combinatorial Optimization (till 1960). Handbook of Discrete Optimization, pp. 1–68 (2005)
30. Sharma, M., Robeonics, K.: Algorithms for micro-mouse. In: International Conference on Future Computer and Communication, 2009. ICFCC 2009, pp. 581–585 (2009)
31. Snider, G.: Self-organized computation with unreliable, memristive nanodevices. IOPSci. Nanotechnol. **8**(36), 365202 (2007)
32. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**(7191), 80–83 (2008)
33. Williams, R.: How we found the missing memristor. IEEE Spectr. **45**(12), 28–35 (2008)

Computing Image and Motion with 3-D Memristive Grids

Chuan Kai Kenneth Lim, A. Gelencser, and T. Prodromakis

Abstract In this paper, we first present a biorealistic model for the first part of early vision processing by incorporating memristive nanodevices. The architecture of the proposed network is based on the organisation and functioning of the Outer Plexiform Layer (OPL) and Inner Plexiform Layer (IPL) in the vertebrate retina. The non-linear and adaptive response of memristive devices make them excellent building blocks for realizing complex synaptic- like architectures that are common in the human retina. We particularly show how that hexagonal memristive grids can be employed for faithfully emulating the smoothing effect occurring in the OPL to enhance the dynamic range of the system. A memristor-based thresholding scheme is employed for detecting the edges of grayscale images, while evaluating the proposed system's adaptability to different lighting conditions and fault tolerance capacity. We then extend our work to computing relative motion of objects, which is an important navigation task that vertebrates routinely perform by relying on inherently unreliable biological cells in the retina. Here, a novel memristive thresholding scheme that facilitates the detection of moving edges is introduced. In addition, a double-layered 3-D memristive network is employed for modeling the motion computations that take place in both the OPL and IPL that enables the detection of on-center and off-center transient responses. Applying the transient detection results, it is shown that it is possible to generate an estimation of the speed and direction a moving object.

C.K.K. Lim
Centre for Bio-inspired Technology, Department of Electrical and Electronic Engineering,
Imperial College London, London SW7 2AZ, UK
e-mail: chuan.lim08@imperial.ac.uk

A. Gelencser
Interdisciplinary Technical Sciences Doctoral School, Pázmány Péter Catholic University, 1088
Budapest, Hungary
e-mail: gelencser.andras@itk.ppke.hu

T. Prodromakis (✉)
School of Electronics and Computer Science, University of Southampton, Southampton
SO17 1BJ, UK
e-mail: t.prodromakis@soton.ac.uk

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_25,

1 Introduction

Over the past years, the performance and efficiency of biological systems has inspired many researchers and engineers, giving birth to the emerging fields of biomimetics [49] and bio-inspiration [62].The human retina is anything but a simple passive relay station, as it pre-processes and compresses all sensed information through an immensely complex neuronal network that on average contains 4.6 million cones, 92 million rods [21] and 1 million ganglion cells [20]. If we take into account the remaining bipolar, horizontal and amacrine cells and the fact that they are highly interconnected we get a parallel network of grand complexity. This complexity is further elucidated in recent studies where evidence is provided that at least ten parallel signals arise from a single visual point [53] and we get a much more accurate view from the functioning of the vertebrate retina. In fact it is nowadays believed that although biological systems are based on relatively primitive elements, it is this naturally occurring interconnection complexity that facilitates higher order functioning.

In our work, we demonstrate the potential of emerging nanoscale elements as synapse emulators for mimicking complex biological functions. We specifically focus on the connection of the sensory and consecutive system of the retina, which is the first and common step in the visual information flow. These connections take place on the OPL in the vertebrate retina; they form a highly dynamic system that enables the smoothing of optical inputs and thus catalyzes the enhancement of the retina's dynamic range, while the different parallel channels emerge after this point. Our approach alleviates these issues by employing the latest biological knowledge [53], and an emerging nanoscale device that is used as a more adequate synapse emulator [2], the memristor. This device exhibits a highly non-linear dynamic behaviour, which along its infinitesimal dimensions serves as an excellent building block for facilitating practical realisations of the highly complex synaptic networks constituting the OPL. We further expand this approach by utilising a memristive-based thresholding scheme for performing edge-detection. Finally, we demonstrate that this platform exhibits similar attributes to naturally occurring systems such as noise resilience, self-adaptation and fault tolerance.

Computing motion is essential for performing many daily tasks that also find applications in machine vision [13, 22], industrial automation [37, 43, 55, 58] and robotics [15, 29]. Traffic engineers and transport authorities require motion computation and object tracking in order to implement effective intelligent transportation systems [44, 59]. For the defense industry, the ability to track mobile enemy targets [11] and control unmanned vehicles [56] is critical. The tracking of cell movements is also recently shown to play an important role in medical diagnostic procedures [12, 51]. While in the field of communications, combining motion estimation and tracking with predictive coding can allow for significant reduction in transmission bandwidth [48].

By utilizing parallel networks and hierarchical structures, biological systems are able to perform motion computation very efficiently [64]. In comparison, conventional motion computation systems typically utilize charged-coupled device (CCD) cameras and digital processors that result in time-sequential computations. As a result, large processing power is required to ensure that these serial computations can be performed in real time. In recent years, the need for physically smaller and more power efficient devices has driven research into neuromorphic [45, 63]and biomimetic systems [7], which are largely inspired by the parallelization naturally found in biological systems.

One of the first biomimetic motion computation systems was implemented by a team lead by Carver Mead in Caltech [14, 30]. By modeling the optical flow via a network of analog resistive devices, this approach was shown to be able to compute an estimate of the optical flow. Over the past two decades, various other biomimetic approaches to motion computation have been implemented in CMOS technology [1, 4, 9, 10, 16, 23, 26, 30, 33, 34, 38, 46, 47, 68].

Perhaps, the most prominent implementation is a 128 × 128 asynchronous temporal contrast silicon retina developed by Delbruck [38]. This system makes use of address-event representation (AER) [41]; inspired by the way neurons utilise spiking events for communication. Essentially, the AER approach allows the transmission of only the local changes caused by movement in a scene, instead of transmitting at a fixed frame rate. This increases the computational efficiency of the chip, enabling it to perform over an impressive dynamic range of more than 120 dB with a low power consumption of 23 mW.

Another notable implementation is Visio1 [9], a retinomorphic chip with parallel pathways that mimics the OPL and IPL. Subthreshold current-mode circuits are used to model the autofeedback characteristics of horizontal cells (for spatial filtering) and the loop-gain modulation of amacrine cells (for adapting temporal filtering to motion). Detection of edges moving in one direction or the other is made possible by aVLSI implementation of ganglion cells that respond to motion in a quadrature sequence.

The recent discovery of practical memristive devices [67] has opened the option of improving on these CMOS motion sensors. It has been shown in [25] that memristor-MOS technology (MMOST) is capable of outperforming CMOS implementations in terms of power and size. Furthermore, the non-linear and adaptive response of memristors allow them to serve as excellent building blocks for the practical realisation of complex synaptic connections [5, 57]. Our group has previously demonstrated that memristive grids have a great potential for modeling certain aspects of early vision processing, which takes place in the OPL [27].

Here, we describe a novel approach, distinct from the current implementations, that allows for motion computation based on memristive networks, by extending our results in [27]. A novel memristive thresholding scheme is utilized for enabling the tracking of moving objects within a scene. Also, a biomimetic model of both the OPL and IPL are described that facilitates the detection of ON and OFF center transient responses. In turn, we investigate the directional and speed computation capacity of our approach. Finally, we demonstrate that this platform can operate in a fault tolerant manner and can adapt to distinct lighting conditions, similarly to biological counterparts.

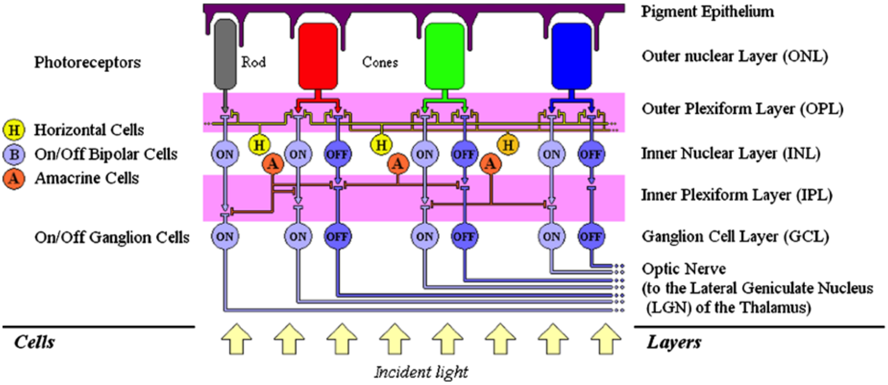


Fig. 1 Conceptual representation of the mammalian retina. To avoid confusion, only five major cell types are depicted with the main interconnection between them organised across the seven different layers. The visual information flows through the retina via bipolar cells and travels to the thalamus via the axons of ganglion cells. Lateral interconnections also exist between the bipolar and horizontal cells in the outer plexiform layer (OPL) and the bipolar and amacrine cells in the inner plexiform layer (IPL)

2 Background

2.1 *Vertebrate's Retina: Architecture and Cells*

In order to perform any form of biomimicry, it is necessary to comprehend the architectural and functioning features of the leveraging biological system. This section serves as a brief unqualified introduction of the retina and is not meant to be a thorough review. More in depth information can be found in [24, 52, 65]. A simplified drawing showing the different layers of the human retina is shown in Fig. 1.

There are 5 main types of retina cells (photoreceptor cells, bipolar cells, horizontal cells, amacrine cells and ganglion cells) laid along two major layers: the Outer Plexiform layer (OPL) and the Inner Plexiform layer (IPL). The photoreceptors (rods and cones) receive visual stimuli from the outside world. The outputs of the photoreceptor cells form synaptic connections with bipolar cells and horizontal cells. The horizontal cells are interneurons that form lateral local connections. The bipolar cells connect the OPL, comprising of horizontal and photoreceptor cells, to the IPL containing the amacrine cells and ganglion cells. Similar to horizontal cells, amacrine cells mediate synaptic lateral interactions between bipolar cells and ganglion cells in the IPL.

The photoreceptor cells essentially act as transducing elements, transforming the incident light stimuli into electrical signals. The magnitude of voltage change in the cells membrane is proportional to the logarithm of the intensity of light [26]. Horizontal cells form synaptic connections between photoreceptors and bipolar cells. Via the horizontal cells, lateral connections with neighbouring groups of photoreceptors and bipolar cells are made. These can be conceived as being a resistive layer spanning the OPL, with the architecture allowing dynamic range adjustments.

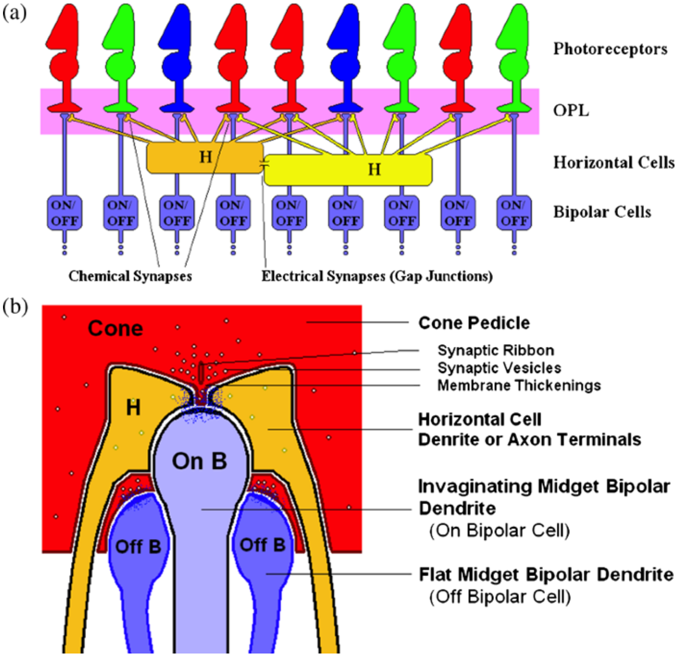

(a)
Photoreceptors
OPL
Horizontal Cells
Bipolar Cells
H
H
ON/ OFF
Chemical Synapses
Electrical Synapses (Gap Junctions)
(b)
Cone
H
On B
Off B
Off B
Cone Pedicle
Synaptic Ribbon
Synaptic Vesicles
Membrane Thickenings
Horizontal Cell Denrite or Axon Terminals
Invaginating Midget Bipolar Dendrite (On Bipolar Cell)
Flat Midget Bipolar Dendrite (Off Bipolar Cell)

Fig. 2 The OPL architecture is illustrated in (**a**) with ON and OFF bipolar cells contracted for simplicity. In the OPL the photoreceptor, horizontal and bipolar cells are interconnected through so called triads. The horizontal cells are interlinked through gap junctions, forming an extended lateral network. (**b**) A triad from the OPL is formed by cone pedicle, ON and OFF bipolar cell dendrites and horizontal cell dendrites or axons

The output of a bipolar cell depends on the difference between the voltage at the photoreceptor it is connected to and the voltage at neighbouring photoreceptors, through the indirect connection via the horizontal cells. There are 2 kinds of bipolar cells. The ON-cells that respond to the onset of light and the OFF-cells that respond to the cessation of light. The photoreceptor, horizontal and bipolar cells do not generate action potential, instead, computation is performed via the flow of depolarization potentials. On the contrary, amacrine and ganglion cells in the IPL do generate spikes. Amacrine cells give transient light responses to either ON stimulus, OFF stimulus or both. Amacrine cells can be classified into sustained and transient cells. Sustained amacrine cells obtain inputs from the bipolar cells that are not inhibited by other amacrine cells. The response of sustained amacrine cells then inhibits bipolar terminals in the narrow field region, causing those terminals to respond transiently. Essentially, this network of amacrine cells forms a high pass filter of the bipolar signal that facilitates the detection of moving edges [36].

Ganglion cells carry the output of the retina to the optical nerve, which leads to the visual cortex in the brain. Of the different types of ganglion cells, the directionally selective (DS) ganglion cells play a huge role in motion detection. There are three identified types of DS ganglion cells, namely ON/OFF DS ganglion cells, ON

DS ganglion cells and OFF DS ganglion cells. ON/OFF DS ganglion cells perform the function of local motion detectors.

2.2 Mathematical Model of Memristors

The derivation of the mathematical memristor model employed here is largely guided by [66]. A memristive system was defined by Chua and Kang [18], following the original 1971 definition of the ideal memristor [17], as:

$$v = R(x)i \tag{1}$$

$$\frac{dx}{dt} = f(x, i) \tag{2}$$

This implies that the resistance $R(x)$ depends on an internal state x of the device while the time derivative of the internal state x is a function of x and i. The first physical model was conceived of a solid-state implementation by Strukov et al. [61], where x is proportional to charge q flowing through it. In the device, the magnitude of R can be modified reversibly between a highly conductive state R_{on} and a highly resistive state R_{off}, by modulating x:

$$R(x) = x(t)R_{on} + \big(1 - x(t)\big)R_{off} \tag{3}$$

$x(t)$ is restricted to the interval [0, 1] and the time derivative of $x(t)$ is proportional to the current, as shown:

$$\frac{dx}{dt} = \frac{R_{on}}{\beta} i(t) \tag{4}$$

Substituting Eqs. (3) and (4) into (1) we obtain:

$$v(t) = \beta\big\{x(t) + r\big[1 - x(t)\big]\big\}\frac{dx(t)}{dt} \tag{5}$$

where $r = R_{off}/R_{on}$ is the resistance ratio and β has a dimension of magnetic flux. Moreover, since $\varphi = \int v dt$, and by using $x\frac{dx}{dt} = \frac{1}{2}\frac{d}{dt}x^2$, it is possible to integrate both sides of Eq. (5) to obtain:

$$\varphi = \beta\left[-\frac{r-1}{2}x^2 + rx + c\right] \tag{6}$$

where c is a constant of integration determined by the initial conditions of x. This equation shows that flux is a quadratic function of the charge as x is proportional to q, hence, the non-linear relationship [54, 67].

Solid-state TiO_2 memristors implementations typically comprise two metal contacts that encompass a TiO_2 active core of thickness D. An external bias across the device causes charge to flow through the device. This causes a drifting of dopants resulting in the movement of the boundary between the two regions. By defining $w(t)$ as the coordinates of this boundary and μ_v as the average ion mobility, Eq. (4) becomes:

$$\frac{1}{D}\frac{dw(t)}{dt} = \mu_v \frac{R_{on}}{D^2} i(t) \tag{7}$$

and by integrating both sides, we obtain:

$$\frac{dw(t)}{D} = \mu_v \frac{R_{on}}{D^2} q(t) \tag{8}$$

It is observed that $x(t) = w(t)/D$ corresponds to the normalized width of the R_{on} region while $\beta = D^2/\mu_v$. Setting c to zero in Eq. (6) and substituting Eq. (8):

$$\varphi = -\frac{R_{on}\mu_v}{2D^2}\left(\frac{R_{on}}{R_{off}} - 1\right)q^2 + R_{off}q \tag{9}$$

Assuming that $R_{off} \gg R_{on}$, the memristance of the device is obtained as:

$$M(q) = \frac{d\varphi}{dt} = R_{off}\left(1 - \frac{R_{on}\mu_v}{D^2}q\right) \tag{10}$$

Furthermore, as mentioned in [61], there are significant non-linearities in ionic transport, especially in the thin film edges at the boundary, which can be modeled by a window function,$f(x)$, on the right hand side of Eq. (7), giving:

$$\frac{dx(t)}{dt} = ki(t)f(x), k = \mu_v \frac{R_{on}}{D^2}, x(t) = \frac{w(t)}{D} \tag{11}$$

2.3 Optical Flow

The estimation of the direction and speed of a pixel in a sequence of image can be obtained from the optical flow; the approximation of local motion in an image by computing local spatial and temporal derivatives in a given sequence of frames. For a 2D picture, the optical flow indicates the speed and direction in which each pixel in an image moves between frames. The computation of optical flow is based on spatio-temporal intensity variations in brightness patterns. In this work we assume that all intensity variations are due to the motion of an object. In general, the optical flow and the velocity of an object are different. As explained in [6], a perfectly featureless rotating sphere has a non-zero velocity component, but it will not give rise to any optical flow. Whereas, a shadow moving across the same sphere, which is now stationary, will produce an optical flow that is non-zero, although its velocity is zero. Nonetheless, apart from these situations, the computed optical flow is a good indication of the velocity of the moving object, provided a strong enough gradient between the moving object and its background exists. This is generally true for most natural scenes and it is assumed to be true for all input stimuli employed in this work.

2D Motion Constrain Equation Assuming $I(x, y, t)$ is the intensity of a brightness pattern at specific pixel located at coordinates (x, y) in the image at time t, and if this brightness pattern moves by $(\delta x, \delta y)$ within a timeframe δt, the intensity at the corresponding pixel will be $I(x + \delta x, y + \delta y, t + \delta t)$. Since this is essentially the same brightness pattern, it can be modeled as:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{12}$$

And by employing a Taylor expansion about the point (x, y, t), this reads:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + H.O.T. \tag{13}$$

By comparing Eqs. (12) and (13), it can be derived that $\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$ and $\frac{\partial I}{\partial x}\frac{\delta I}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t}\frac{\delta t}{\delta t} = 0$ and eventually,

$$\frac{\partial I}{\partial x}v + \frac{\partial I}{\partial y}u + \frac{\partial I}{\partial t} = 0 \tag{14}$$

This can be written more compactly as

$$(I_x, I_y) \cdot (v, u) = -I_t = \nabla I \cdot \overrightarrow{v} \tag{15}$$

∇I refers to the spatial intensity gradient and $\overrightarrow{v}$ refers to the optical flow at pixel (x, y) at time t.

Equation (15) has 2 unknowns, which as explained in [3, 6] stem from the aperture problem. Insufficient information is available to measure the full image velocity, and only the component perpendicular to the edge or along the spatial gradient can be measured. In other words, a point in an image sequence only provides one independent image measurement whereas the velocity field has two components, forming an ill-posed problem.

Smoothness Assumption In order to solve this ill-posed problem, a second constrain is therefore required. Horn and Schunck [6] combined the 2D motion constrain together with a global smoothness term, such the final velocity field $\overrightarrow{v}$ is one which minimizes:

$$\begin{aligned} E(u, v) = \iint & (I_x + I_y + I_t)^2 \\ & + \lambda\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] dxdy \end{aligned} \tag{16}$$

The first term implies that the solution should be as close as possible to the measured data while the second term imposes a smoothness constrain on the solution.

The smoothness constrain derives from the fact that, apart from at specific discontinuities, adjacent points on an object have similar velocity and the brightness pattern varies smoothly almost everywhere. The magnitude of λ determines the importance of minimizing the smoothness term. With high Signal to Noise Ratio (SNR), the importance of the first term increases and λ will be small. In contrast, if the data is unreliable, λ will be larger, emphasizing the importance of the smoothness term.

Equation (16) shows that $E(u, v)$ is quadratic in the unknowns u and v. From standard variation calculus it can be shown that the corresponding Euler-Lagrange equations are linear in u and v:

$$\begin{aligned} I_x^2 u + I_x I_y v - \lambda \nabla^2 u + I_x I_t = 0 \\ I_x I_y u + I_y^2 v - \lambda \nabla^2 v + I_y I_t = 0 \end{aligned} \tag{17}$$

This results into two linear equations describing every point, which essentially capture the necessary components for calculating optical flow.

3 Biomimetic Outer Plexiform Layer

Instead of explicitly modeling in the neurons in the OPL, our work focus on the synaptic interconnections of such neurons and how these adapt to different stimuli for inherently performing smoothing and edge detection. Any visual input is captured via photoreceptors cells in the retina. As this is out of the scope of our work, we opted not to emulate the exact functioning of these receptors but, rather, the effect in translating any light stimuli into an appropriate current bias. We this represent the cells' signaling with equivalent voltage sources to bias the underlying memristive network. Clearly, this approach is only applicable for colourless image inputs.

As illustrated in Fig. 3, every voltage source connects via a resistor to a discrete node of an hexagonal memristive network. The value of this resistance is comparable to the initial memristive states (1 kΩ), since the smaller this resistance is the larger the current in the underlying node will be. It therefore serves as a control parameter of the time evolution of our system. These nodes are essentially representations of the triads comprising the OPL [35], with the memristive fuses employed for emulating the synaptic interconnects of the horizontal cell network. In a similar manner to Fig. 2(b), single memristors were used to emulate the bipolar cells' dendrites, while the biasing provided by the voltage sources and their series resistors resemble the activity occurring at the cone pedicles.

Here we have ascertained an hexagonal topology for complying with the biological counterpart system. Every node within the grid is linked with six neighbouring nodes through memristive fuses. This approach is essentially similar to the one employed by Carver Mead in [30] with our system being distinct in that we employed non-linear memristive devices instead of linear resistors. This option facilitates the proposed model with an inherent local Gaussian filtering functioning to any input image.

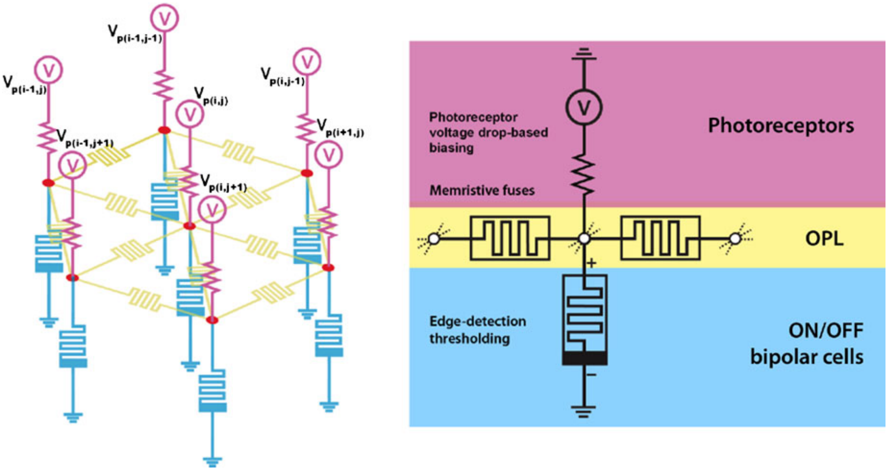
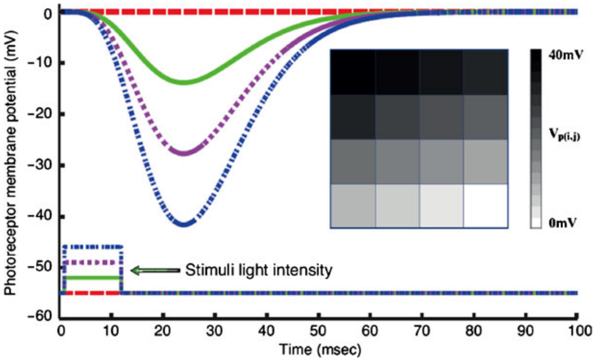


Fig. 3 Close view of a single node of the proposed grid. Voltage sources and serial resistors are representations of the output signals of photoreceptors, when subjected to an optical stimulus. Every node establishes a triad through hexagonally interconnected memristive fuses for imitating the dendrites or axons of horizontal cells, one memristor that represents the dendrite of bipolar cells, while the bias sources and resistors at the input stage denote the cone pedicle

Fig. 4 Simulated photoreceptor depolarisation due to distinct grayscale levels



Typical resting potential of neuron cells is −65 mV, with the photoreceptors resting membrane potential being about −40 mV. When some light stimulus is present, these cells use the absorbed photons energy to hyperpolarise their inherent membrane potential, i.e. light intensity of an optical stimulus will cause photoreceptor cells to hyperpolarise in relation to the light intensity. On the other hand a dark intensity will have an opposing effect, causing the cells to depolarise, as illustrated in Fig. 4.

In this approach the stimulating potential resulting from photoreceptor cells being either hyperpolarised or depolarised was modulated arbitrarily through the employed voltage sources $V_p(i, j)$. Every pixel of an input image was represented with a corresponding voltage level, as depicted in the inset of Fig. 4. The pixel intensity of all stimuli was set on a 16-level grayscale, where value 0 represent the highest possible light intensity and 15 corresponds to a dark current. Thus, all optical inputs were transcribed into a 2D vector of biasing voltages that through a serial resistor establish the OPL's stimulating currents. In a similar manner, the recorded potentials of the memristive OPL nodes are monitored as the network is adjusting the corresponding memristive weights and is transcribed back to matching pixel intensities, for obtaining some meaningful data. We should note here that as the employed depolarisation potentials are in the mV range, these will provoke a relatively slow state modulation of the memristive network. In consequence, the system evolvement is better observed in the time period of a few seconds rather in milliseconds, which however can be adjusted by modifying the stimuli amplitudes.

3.1 Smoothing and Local Gaussian Filtering

Most of the existing edge detection schemes demonstrate various issues when the optical input is distorted [42]. To address these, Gaussian filtering was proposed for suppressing noise through the smoothing (blurring) of the image. A one dimensional Gaussian function can be described as:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{18}$$

where x and y are the distance from the origin and σ is the standard deviation of the Gaussian distribution, which could also be expanded into two-dimensions through:

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{19}$$

Nevertheless, a uniform Gaussian blur across the whole image can cause the displacement of edges, the vanishing of less intense edges along with the creation of edge artefacts [19, 40]. The occurrence of this effect however is diminishable, in the case of local Gaussian filtering [28]. In our approach, memristive dynamics are employed for achieving this performance intrinsically. The filtering variance is dynamically adapting to the local variance of the image and the smoothing alleviates any non-uniformities. The amplitude of the current flowing through any memristive fuse in the OPL depends on the weighted sum of the current flow through the neighbouring cells. If the potential difference between adjacent nodes is high, then the current flow gets higher; if the difference is however low, then the current flow is significantly less. In other words, a clear edge on the input image causes a big intensity gradient and consequently a faster memristance modulation. Such memristive fuses drift towards a higher resistive state faster than the neighbouring cells and the lateral current flow from the high pixel gradient essentially diminishes. Consequently, the edge is preserved while the smoothing effect is vigorously decreased. On the other hand, if the contrast between adjacent pixels is low, the corresponding

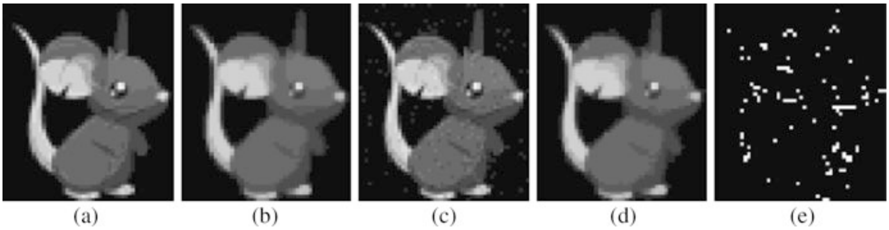


Fig. 5 Demonstration of local Gaussian filtering with our proposed OPL model. (**a**) is the original input image, (**b**) is the extrapolated smoothed version of (**a**) after $t = 30$ sec. (**c**) is a distorted version of (**a**) and the corresponding smoothed output after t=30sec is shown in (**d**). (**e**) is the accentuated intensity mismatch between the smoothed outputs with and without distortion

memristance change will be significantly slower. In this case, this particular memristive fuse will allow a larger lateral current flow and thus the resulting image will be more homogeneous over this area. Hence, small intensity variations on an image tend to smooth out, meaning that added noise is also annihilated, proving the feasibility of the proposed memristive-based local Gaussian filtering.

This is particularly demonstrated through the examples depicted in Figs. 5(a) and 5(c), with Fig. 5(a) being the original input image and Fig. 5(c) a distorted version of the original image with additive white noise with a Gaussian distribution ($\mu = 0$ and $\sigma = 0.3$). In both cases the smoothing caused by this network after $t = 30$ seconds is illustrated in Figs. 5(b) and 5(d) respectively. These figures depict the static voltages measured at the OPL's nodes after being transcribed back to the corresponding grayscale intensities in accordance with the scale shown in the inset of Fig. 4. By observation, the two figures do not show any considerable difference and this was quantified in terms of grade intensity mismatch to be approximately 3 %, as shown in Fig. 5(e). In both cases, the smoothed versions preserved the main edges, while wherever there was an insignificant intensity gradient or small intensity variation caused by the addition of noise, the system smoothed this out.

3.2 *Edge Detection*

The intensity contrast between adjacent pixels imposes the biasing of the underlying memristive fuses with corresponding potential differences and as such an edge can easily be detected by monitoring the outgoing current flow at the OPL nodes. In the counterpart biological system, this information is conveyed through the dendrites of ON/OFF bipolar cells to the IPL. It is important to mention that this edge detection phenomenon is a part of the early vision processing, while the main output of the edge information occurs later in the retina. Here, we utilize two approaches: (1) we employ single memristors in the output stage of the OPL to facilitate a resistive thresholding scheme and (2) we monitor the state variance of the OPL memristive fuses.

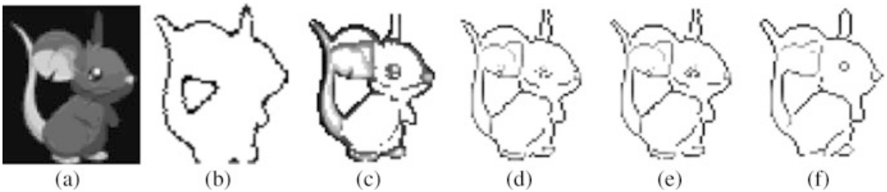


Fig. 6 Detection of the edges of the utilized input image shown in (**a**). Edge-detection was achieved via: (**b**) a bipolar threshold scheme at the output of the OPL nodes with 600 Ω $\le M_T \le$ 2 kΩ and (**c**) a thresholding scheme at the memristive fuses with $M_T = 1.6$ kΩ. These results are compared against conventional edge-detection algorithms: (**d**) Prewitt, (**e**) Sobeland (**f**) Canny

At any given time, the relative memristance change, both in the memristive fuses and the single devices, is a measure of the current flowing through these devices. As the system evolves, the devices associated with nodes that are exposed to large potentials, i.e. in neighbouring pixels at an edge, will drift towards lower conductive states at a rate set by the overlying intensity contrast. By monitoring the transient memristance change of the devices in the OPL output nodes, we associate appropriate thresholding values for defining clear edges. All devices falling between these thresholds will thus indicate the existence of an edge. In the case where Fig. 6(a) is the source image and the threshold is bounded within 600 Ω $\le M_T \le$ 2 kΩ, the detected edges will correspond to what is shown in Fig. 6(b). Clearly, these thresholds could be manually adjusted for attaining more or less edge details.

In the second approach, a more elaborate thresholding scheme involves the monitoring of the states of all memristive fuses associated with a node. If at least three devices are exceeding a preset threshold, this particular node is then denoted as an edge pixel. Figure 6(c) illustrates the edges as detected through this approach for a $M_T = 1.6$ kΩ. Figures 5(a) and 5(c) are respectively used to bias the memristive network. Additionally, as a figure of merit, Figs. 6(d), 6(e) and 6(f) illustrate the edges detected by employing conventional algorithms, specifically: Prewitt, Sobel and Canny.

Similarly, this method was also exploited with a rubic cube image, as shown in Fig. 7(a), with the memristive threshold being set at $M_T = 3$ kΩ. In this example, the smoothing process has caused some distortion on the inhibited pattern of the front sides of the cube, as this is erroneously considered as noise due to the small contrast difference existing between these single-pixel lines and their background. As a result this pattern tends to be smoothed out, as illustrated in Fig. 7(b). Nevertheless, our model manages to distinguish the main edges of the cube, as well as the finer edges that are inhibited on the top side of the cube. This is clearly illustrated in Fig. 7(c), where it appears to attain clearer edges when compared against the conventional edge detectors Prewitt, Sobel and Canny, which results are correspondingly shown in Figs. 7(d), 7(e) and 7(f).

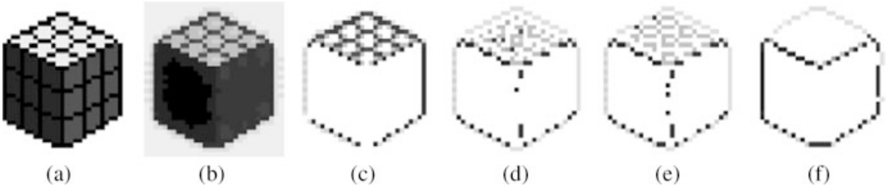


Fig. 7 Detection of the edges of the utilised input image shown in (**a**) and the corresponding smoothed output after $t = 30$ sec is shown in (**b**). Edge-detection was achieved by applying a thresholding scheme with $M_T = 3$ kΩ, shown in (**c**). This result is also compared against conventional edge-detection algorithms: (**d**) Prewitt, (**e**) Sobel and (**f**) Canny

3.3 Adaptation to Light Conditions

The vertebrate retina is capable of self-adapting to maintain the retinal response to visual objects approximately the same when the level of illumination changes. Here we demonstrate that our model behaves in a similar manner to its biological counterpart when subjected with distinct light conditions. Figures 8(b) and 8(c) demonstrate that the proposed memristive network is capable of detecting the edges inhibited in the original image despite the two-times light variance in the original figure. In these examples we have manually adjusted the memristive threshold for achieving similar edge detection to the original system. When the image is brighter, the difference between a contour's pixels will be relatively higher, meaning that there will be more current flowing through the memristive devices that correspond to this edge, thus their state will be altered in a faster manner. In this case, matching the edge-detection performance of our system, as shown previously in Fig. 7(c), necessitates the use of a higher memristance threshold of $M_{TL} = 6.35$ kΩ.

On the other hand, when the image has a darker tone, the contrast between the pixels defining an edge will be less significant. In consequence, smaller potentials are established across the corresponding memristive fuses and their state will change at a slower manner. Likewise, a lower memristance threshold of $M_{TD} = 1.2$ kΩ is required to achieve a similar performance to Fig. 7(c). Clearly, a lower threshold implies that more memristive fuses will exceed this threshold at any given time, justifying the detection of thicker edges as illustrated in Fig. 8(e). Supplementary videos S.6a and S.6b illustrate the transients of Figs. 8(b) and 8(e) respectively.

Since the relative contrast in the pixels of a "light" and "dark" tone image will be rather similar with that in the original image, if one maintains the same threshold ($M_T = 3$ kΩ) for both light conditions, the proposed system can in principle detect the same edges as previously showed in Fig. 7(c) ($t = 30$ sec). However, in the case of a lighter environment the system will converge to a similar solution after $t = 21.8$ sec, as shown in Fig. 8(c), while in the case of a darker environment the system will require double the time ($t = 44.7$ sec) to converge into a similar solution (Fig. 8(f)).

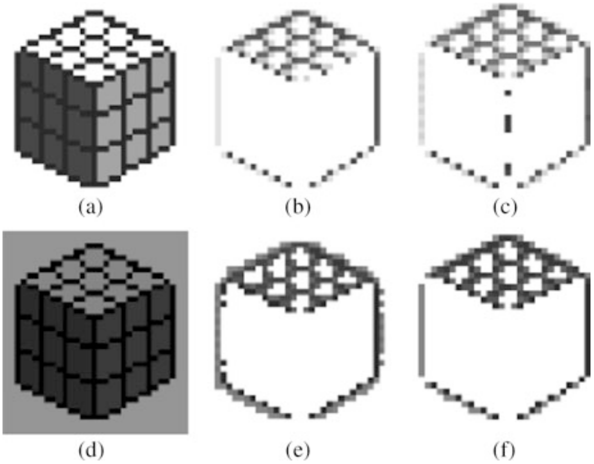


Fig. 8 Evaluation of the memristive platform against distinct light conditions. (**a**) and (**d**) are representations of the employed input image (Fig. 7(a)) in brighter and lighter tones. The detected edges for both conditions are illustrated in (**b**) and (**e**) when corresponding memristive thresholds are utilized $M_{TL} = 6.35$ kΩ and $M_{TD} = 1.2$ kΩ. While (**c**) and (**f**) are the corresponding results when $M_{TL} = 3$ kΩ is the same for both cases, after the system has evolved for $t_L = 21.8$ sec and $t_D = 44.7$ sec

3.4 Fault Tolerance

Biological systems depend on rather primitive elements whose properties often vary randomly. Yet, nature is capable of performing highly complex functions in a very reliable manner by employing redundancy. Similarly, solid-state devices and particularly memristive devices of deep submicron dimensions demonstrate a very poor yield. Given the fact that memristors are a disruptive technology, reliability and robustness of the devices becomes a significant burden. In this view, we extend our investigation on the effect defective devices could potentially have in our model.

We consider that for a 100 % yield, all memristive fuses are reliably set with $R_{ON} = 100\ \Omega$, $R_{OFF} = 16$ kΩ and $R_{init} = 200\ \Omega$. In order to test the robustness of our system we model different yields, by assigning erratic initial states to a number of randomly selected memristive elements. This means that the conductance of these memristive elements differs from the normal one. When a device is considered as faulty, its RON could vary from 50 % to 400 % compared to the ideal scenario. Similarly, R_{OFF} may be varied from 62.5 % to 125 % and R_{init} could take any value from 50 % up to 4000 % when compared with the ideal values. Figure 9 shows two circuit maps, where 25 % (Fig. 9(a)) and 50 % (Fig. 9(b)) of the total memristors in the network were randomly affected. The employed colour mapping corresponds to the randomly distributed initial states M_{init} with green hexagons marking the unaffected devices and red and blue the affected ones.

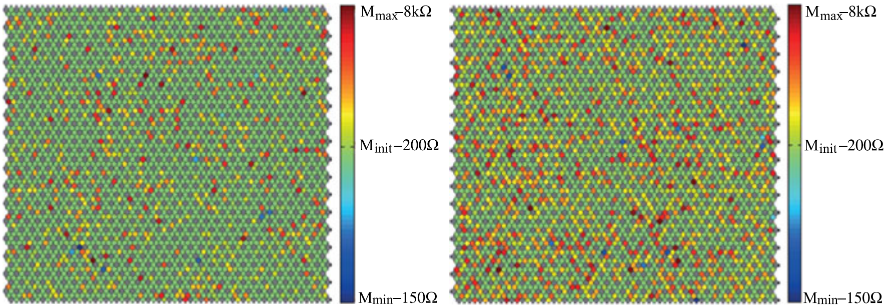


Fig. 9 Test conditions for evaluating the fault tolerance capacity of this model. Shown is the distribution of memristive elements whose initial state (Minit) was altered by (**a**) 25 % and (**b**) 50 % of the total memristors in the network

When the same optical conditions are applied as in Fig. 7(a), the OPL will produce a smoothed equivalent as shown in Fig. 10(a). When however the network's yield is set to 75 % and 50 %, the memristive grid acquires an uneven initial weight distribution that produces the smoothed versions shown in Figs. 10(b) and 10(c) respectively. The relative difference between the flawless and the affected memristive fuses are shown on Figs. 10(d) and 10(e) respectively. We can observe that the smoothing of the image will decrease, because of the high number of defective memristors. Yet, our edge detection method still holds and is capable of detecting most of the correct edges. Supplementary videos S.7a and S.7b demonstrate the time evolution of the network that causes the smoothing, while S.7c and S.7d illustrates dynamically the edges as detected for a yield of 75 % and 50 % respectively.

Regardless the low yield values we tested for, the proposed hexagonal memristive network appears to be proficient in detecting the inhibited edges effectively. However, when the same conditions are employed in a rectangular memristive architecture, the results are not as encouraging as in the hexagonal topology. Besides the geometrical advantage that allows more unit cells to be tessellated per unit area, the hexagonal topology bares two extra interconnections per node, enhancing the system's redundancy. Therefore, the local averaging occurs with two more spatial partners, accounting for the introduced faults in the devices' characteristics.

Although we have demonstrated the potential of using memristive networks for mimicking the retina dynamics, physical implementations can be rather challenging, requiring the employment of complex interconnection schemes [39, 60] that could impose challenging processing (eg.: the use of 3D CMOS). In turn, we are certain that this demanding interfacing scheme could also hinder the devices reliability, which is why we are investigating the effect of low yields on the overall system performance.

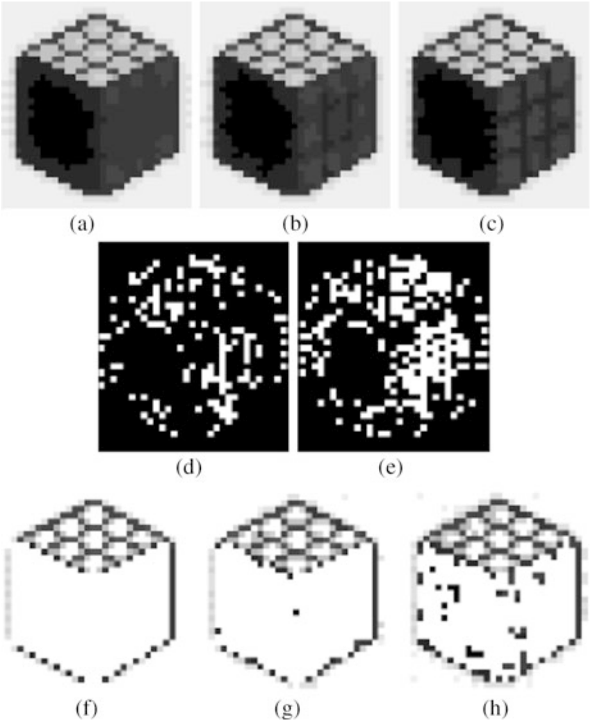


Fig. 10 Smoothing and edge detection performance for varying memristor yield. (**a**) Ideal scenario, (**b**) 75 % yield and (**c**) 50 % yield. (**d**) and (**e**) show the accentuated difference between the ideal and randomly affected networks. The corresponding detected edges are illustrated in (**f**), (**g**) and (**h**) for yields of 100 %, 75 % and 50 %

4 Detecting Moving Edges with Memristive Grids

We have previously presented a retinomorphic OPL, which modeled the synaptic interconnections between horizontal, bipolar and photoreceptor cells. Any light stimulus is translated into an appropriate voltage source to bias the underlying hexagonal memristive network via a series resistor. It was shown that the network was able to perform a local Gaussian filtering function similar to the biological counterpart, as well as edge detection via a memristor-based thresholding scheme. Nodes corresponding to edge pixels have large potential difference between neighbouring nodes due to the overlying intensity contrast at these pixels. From Eq. (10), the change in the memristance of the memristive fuse is proportional to the current flowing from that edge node to a neighbouring node. Therefore, by monitoring the state variance of memristive fuses associated with a node, edge detection can be performed. Here, we extend our work [27] by performing detection of edges of moving objects from a video, investigating how a memristive network can be used to detect moving edges from a sequence of frames.

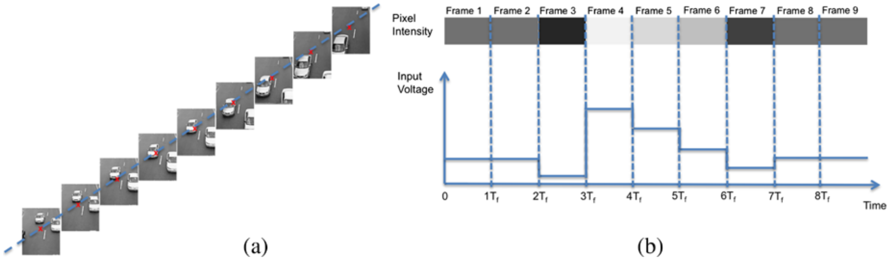


Fig. 11 (**a**) The sequence of the input frames is shown. A specific pixel in the video, marked by a Red X is indicated in each of the frames. The time between consecutive frames, T_f, is the inverse of the framerate. (**b**) The pixel intensity of each frame is shown and this corresponds to an input voltage stimulus, which is applied to the respective node in the memristive grid

4.1 Simulation Methods

The employed video input stimulus is a 80 × 70 pixel, 0.6 s segment of the video "traffic.mju", which can be found in the Matlab Image Processing Toolbox [31]. At a frame rate of 15 fps, this corresponds to 9 frames, shown as a montage in Fig. 11a. Changes only occur at interval of the frame period (1/15 s), and within this period the voltage stimulus at each pixel is a constant. This is modeled with a piecewise-linear voltage source at each node corresponding to a pixel.

Figure 11(b) describes how a single pixel in the video is converted into a voltage stimulus. In each gray scale frame, the intensity of each pixel is uniformly quantized into 256 levels, where 0 represents the absolute black and 255 represents white. Each pixel is therefore represented by an integer from 0–255. Since the maximum voltage input into each of the voltage input node in the memristive grid is set to 40 mV, if the quantised intensity of a pixel is represented by X, the stimulus to its respective voltage input node is X*40/255 mV. This operation is performed for each of the pixel in each of the consecutive frames. The input voltage stimulus into a specific node corresponding to a pixel is then set to the piece-wise-linear combination of this quantised voltage level.

MATLAB was employed to implement the netlist for the hexagonal memristive network and to convert the input frames into equivalent biasing voltage stimulus. The generated netlist is then imported into HSPICE to perform the necessary circuit simulations on the constructed system. Memristive devices were simulated with Biolek's model [8], with the following parameter settings: $R_{on} = 100\,\Omega$ $R_{off} = 16\,\mathrm{k}\Omega$ $R_{init} = 200\ \Omega$. In addition, the Prodromakis's windowing function [50] was employed to take into account the non-linear dopant kinetics of the memristive elements. The results from the HSPICE simulations were then imported back to MATLAB for further analysis.

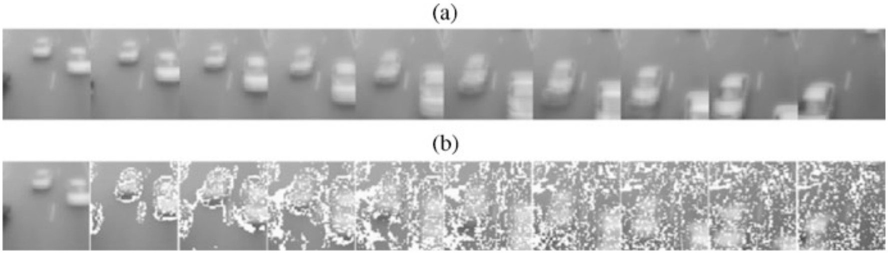


Fig. 12 (**a**) Evolution of the smoothed input stimulus. (**b**) Simulation of a sequence of images with the edge detection algorithm from [27]. Notice that edges were detected successfully only for the first 2 frames

4.2 Detection of Moving Edges via Memristive State Thresholding

The circuit was simulated for the input stimulus shown in Fig. 11(a). Edge detection was performed for each time step of the simulation, using the algorithm described in [27]. The results are shown in Fig. 12(b) where the edges are marked by a white pixel.

We notice that the memristance based thresholding scheme described in [27] was able to pick up the edges in the first 2 frames. However, for the rest of the frames, it seems like that edges were being picked up randomly. This is caused by the fact that the memristors used for the simulations are non-volatile and they do not reset or change before the next input frame is being simulated. Hence, even when the next frame is being simulated, the past memristance changes are still present, causing the results to be corrupted. In the following section, we address this issue by incorporating a novel memristive thresholding scheme that is applicable for detecting moving edges.

4.3 Edge Detection Based on Monitoring the Memristance Modulation Rate

To overcome the problem presented previously, we need to be able to separate the change in memristance corresponding to moving edges in the next frame from the changes in memristance corresponding to edges in the current frame. The proposed solution exploits the tracking of the temporal derivative of the memristance of the six memristive fuses connected to each node.

Following Eq. (10), the change in memristance is proportional to the current flowing through the device, which is dependent on the potential difference across the device. Since each memristive fuse links adjacent nodes, the potential difference across it will be proportional to the intensity difference between the two nodes. If a node is an edge, there will be a strong intensity gradient with its neighbouring nodes. This results in a large rate of change of memristance for the associated memristive

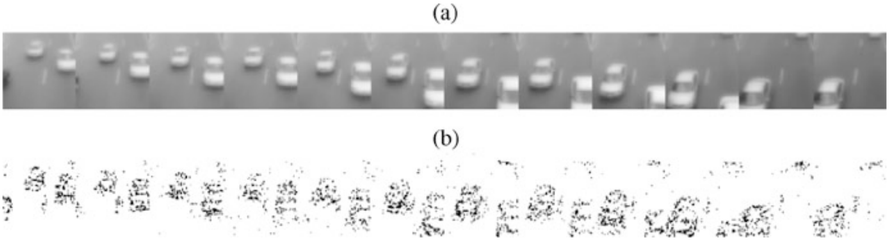


Fig. 13 (**a**) Evolution of the smoothed input stimulus. (**b**) Moving edge detection by tracking the temporal derivative of the memristance of the fuses associated to each node

fuses. Hence, by monitoring this rate of change, we are able to determine if a node corresponds to an edge pixel.

In a sequence of moving images, once the edge of the object moves away from a node, that particular node will cease to have a strong intensity gradient with its neighbouring nodes. Consequently, the temporal derivative of the memristance of the associated memristive nodes will no longer be high and hence the node will not be denoted as an edge. Figure 13 presents results that demonstrate our approach. A node is denoted as an edge pixel if the rate of memristance change of two or more memristive fuse associated with that node exceeds a predetermined threshold. Each edge pixel is denoted by a black pixel, superimposed on a white background. It is observed that the moving edges can now be successfully detected for the entire sequence of images.

4.4 Adaptation to Varying Lighting Conditions

Similar to [27], we evaluated the proposed algorithm against distinct luminance levels. Figure 14 demonstrates that even when the brightness level was halved, the moving edges were still detected successfully. The moving edge detection results under the two different lighting conditions were compared pixel by pixel and the average mismatch between these two cases was found to be 4.22 %. This is illustrated in Fig. 14(c), where each white pixel corresponds to a mismatch.

4.5 Memristance Variability Tolerance

The memristive network was tested to determine how tolerant it was to potential defects in the network. To simulate this, defect memristors were artificially introduced. Instead of simulating a circuit with all uniformly initialised memristors (i.e. $R_{on} = 100\ \Omega$ $R_{off} = 16\ \text{k}\Omega$ $R_{init} = 200\ \Omega$), we now initialise a percentage of the memristors randomly. These "defected" memristors will be initialised randomly

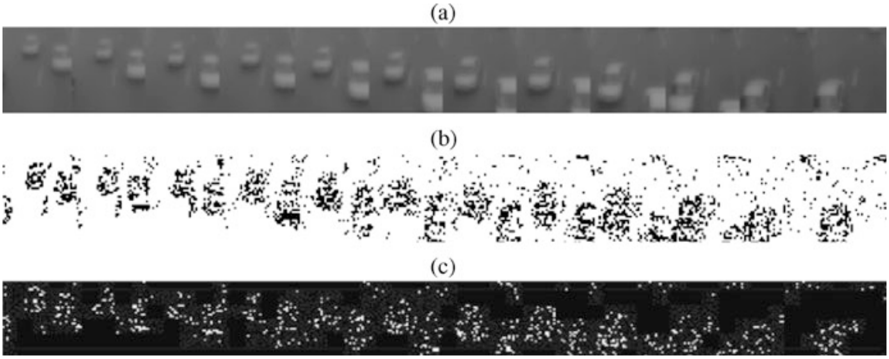


Fig. 14 (**a**) Evolution of the smoothed input stimulus. (**b**) Moving edge detection with input stimulus of half the original brightness. (**c**) Illustrates the accentuated mismatch between moving edge detection under normal lighting condition and under half the original brightness

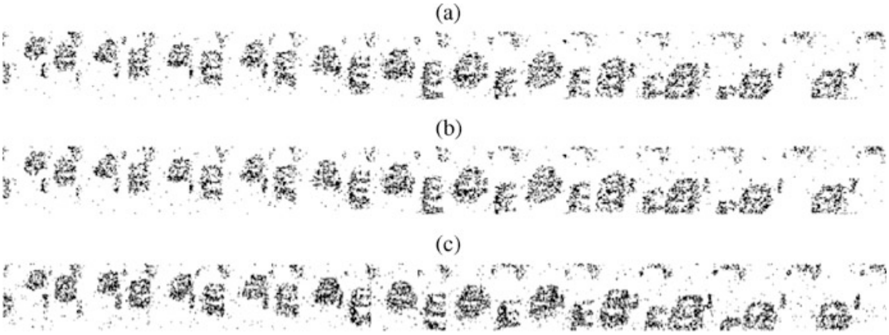


Fig. 15 (**a**) Evolution of the smoothed input stimulus. (**b**) Moving edge detection with artificially simulated faults of 50 %. (**c**) Moving edge detection with artificially simulated faults of 80 %

with $R_{on} = 50\ \Omega$–$100\ \Omega$ in increments of $10\ \Omega$, $R_{off} = 10\ \text{k}\Omega$–$20\ \text{k}\Omega$ in increments of $0.5\ \text{k}\Omega$, R_{init} as a multiple (2–20) of the new R_{on}. The simulation results for 50 % defects and 80 % defects are respectively shown in Figs. 15(b) and 15(c).

The memristive network appears to be rather robust to faults, even when 80 % of the devices were initialised randomly. The results presented in Fig. 15 demonstrates that the proposed system was still able to perform the detection of moving edges reliably. This is largely attributed to the fact that there are inherent redundancies in the network, as each node is interconnected to 6 neighbouring nodes. This is particularly important as the fabrication yield of memristive devices is relatively poor given that the memristors' technological readiness is rather low. The scalability of memristive technology indeed opens the possibility of achieving reliability through redundancy.

5 Double Layered Memristive Network

In the previous section, it was demonstrated that by employing the proposed thresholding scheme, memristive networks were able to perform moving edge detection, even with artificially simulated faults and at varying luminance levels. At the same time, this implementation is able to achieve local Gaussian filtering with a lower complexity as compared to other systems [27] and has been shown to perform better than resistive grids for edge preserving smoothing [32]. However, with this implementation, full optical flow computation is still not quite possible, as we have not yet obtained enough information from the sequence of moving images. The derivation of optical flow (in Sect. 2.3) shows that Eq. (17) has to be solved in order to obtain the required optical flow. This requires both the spatial derivatives and the temporal derivatives at each pixel but our current implementation only provides us the spatial derivative (i.e. through the rate of change of memristance of memristive fuses). In this section, we present a 3D memristive grid that facilitates the computation of the temporal derivative.

5.1 Biomimetic OPL and IPL

From the results in the previous section, we note that a second memristive layer is required to model the amacrine cells within the IPL of the retina that are responsible for generating the transient responses. Our model does not explicitly emulate the functioning of the neurons in the IPL, but rather the synaptic interconnections between them. In order to generate the transient responses, a delay of the input stimulus is required, and this is introduced by the delayed version of the input in the second layer.

The 3D network, part of which is illustrated in Fig. 16, consists of 2 hexagonal memristive layers interconnected with memristive fuses at each node. The current signal is applied to layer 1, which is responsible for computing the edges from the sequence of frames. Whereas, in layer 2, a delayed version of each frame is used, enabling the computation of the temporal derivative at each node via the memristance changes in the interconnecting memristive fuses.

Each connecting memristive fuse is made up of two individual memristors. We label the two memristor in each memristor fuse as Connecting Memristor 1 (cM1) and Connecting Memristor 2 (cM2), where cM1 is the memristor closer to layer 1 and cM2 is the one closer to layer 2.

When an object in the input stimulus is completely stationary, it implies that the pixels corresponding to that object remains the same for the current frame n and the delayed frame $(n-1)$. Consequently, the spatial voltage input (representing this object) to each layer will be identical and both ends of all connecting memristors are at equipotential. No current would thus flow and there will be no observable change in memristance.

Conversely, when an object in the input stimulus moves, the pixels corresponding to that object will move. This means that the current frame n is slightly different

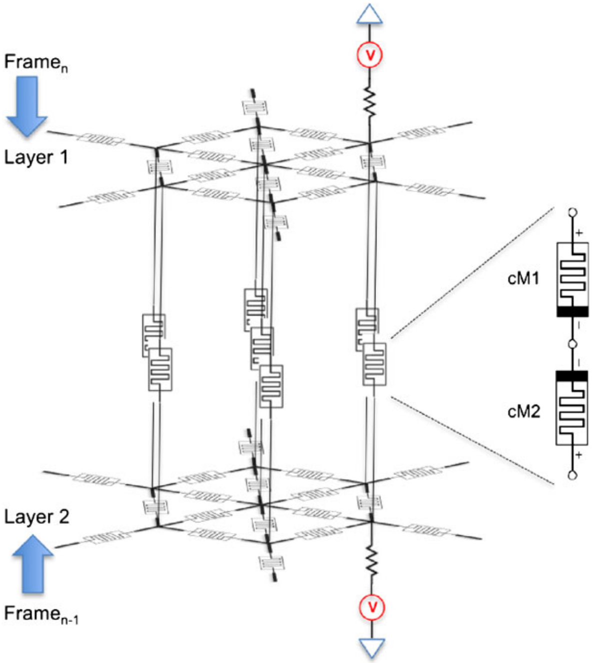


Fig. 16 Illustration of a section of the proposed 3D hexagonal memristive network model. Each node corresponding to a specific pixel is connected in a hexagonal manner to 6 neighbouring nodes via memristive fuses. The network consists of two interconnected layers. The current signal is applied to Layer 1 while a delayed signal of 1 frame is applied to layer 2. Each pixel in the input frame is converted into a piecewise-linear voltage value, which is then applied through a resistor to the memristive grid. For clarity, only two of these biasing connections are shown

from the delayed frame $(n-1)$, reflecting this movement. Consider the case where a black box moves to the right by 2 pixels on a white background, as shown in Fig. 17(a). It can be observed that the rightmost and leftmost black edges of the box will be moving by 2 pixels to the right. Comparing with frame $(n-1)$, the black box in frame n will be a shifted version by 2 pixels to the right. Hence, there will be a potential difference between the nodes corresponding to the new rightmost black edge (in frame n) and the old leftmost black edge (in frame $(n-1)$). At the new rightmost black edge, there will be a higher potential at layer 2 compared to layer 1. This causes a current flow through the connecting memristor fuse in from layer 2 to layer 1. Therefore, it is expected that the memristance cM2 will be increasing while the memristance of cM1 will be decreasing. Whereas for the old leftmost black edge, there will be a higher potential at layer 1 compared to layer 2. A current will flow from layer 1 to layer 2 causing the memristance of cM1 to increase and cM2 to decrease.

Therefore, by monitoring the memristance changes in each memristor of each connecting memristor fuse, it is possible to determine the movement of the edges. For a dark object moving on a light background, decreasing cM1 and increasing cM2 corresponds to an appearing edge while increasing cM1 and an decrease of cM2 corresponds to a disappearing edge. This is similar to the on-center response in the retina. On the other hand, for a light object moving on a dark background, increasing cM1 and decreasing cM2 corresponds to an appearing edge while decreasing cM1 and increasing cM2 corresponds to a disappearing edge. This is

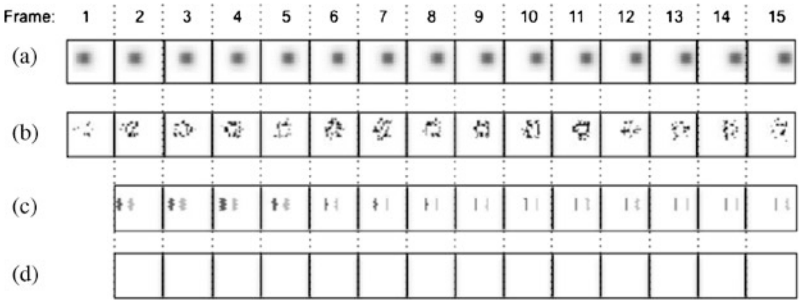


Fig. 17 (**a**) Evolution of input stimulus: A *black box* moving to the right at a rate of 2 pixels per frame from frame 1 to frame 5, and 1 pixels per frame from frame 6 to frame 15. (**b**) Simulation result of moving edge detection. (**c**) OFF-center transient detection results from monitoring the connecting memristive fuses. (**d**) ON-center transient detection results from monitoring the connecting memristive fuses

similar to the off-center response of the retina. Essentially, the response of these connecting memristors captures the response of the transient amacrine cells in the retina.

With a delayed version of the input into the second layer of memristive network, it is possible to derive the temporal derivative between consecutive frames. The temporal derivative between corresponding pixels in consecutive frames is inherently computed as changes in memristance of the memristors within the connecting memristive fuse. By combining the temporal derivative information with the spatial derivative information obtained from the hexagonal memristive network, it will be possible to separate moving objects from stationary objects. Essentially the hexagonal grid is performing computation of the sustained response while the connecting memristors are responsible for computing the transient response.

5.2 Emulating Transient Detection

Simple Inputs As an initial proof of how the above-mentioned motion computation can be done with the double-layered memristive network, two simple simulations were performed. Each of the simulation input stimulus consists of a 1 s, 15 fps, 17 × 30 pixels video clip.

The first simulation, illustrated in Fig. 17, involves a black box moving across a white background. From frame 1 to frame 5, the box is moving to the right at a rate of 2 pixels per frame and from frame 6 to frame 15, the box continues moving to the right at a slower rate of 1 pixel per frame. Since the moving object is dark compared to the surrounding, the on-center transient response (Fig. 17(d)) is zero, which is expected. In the off-center transient response (Fig. 17(c)), the light grey pixel corresponds to the appearing edge, while the dark pixel corresponds to disappearing edges. We see that the model is capable of monitoring the movement of the edges, while the speed of this movement can be derived from the thickness of the transient edges.

Our second example is similar to the first except for the fact that instead of a black box moving across a white background, a white box is now moving across a black background. In this case, the off-center transient response is zero and all the transient response is due to the on-center transient response. Here, the dark grey pixels correspond to an appearing edge while the light grey pixels correspond to a disappearing edge.

Complex Inputs The simulation was extended to a more complex input, which involves the first two frames of the traffic video stimulus shown in Fig. 11(a). Only twp frames were simulated as the simulation of more frames are rather computationally intensive. The frame size used in this simulation was 80 × 70 pixel. This results in a circuit with 22400 nodes, 38602 memristive fuses, 11200 resistors and 11200 varying voltage sources. Even when running on a HP Z820 workstation with two 8-core Intel Xeon 2 Ghz processor and 64 GB of RAM, the simulation environment (HSPICE) is incompetent in solving more extended cases.

The input stimulus, shown in Fig. 18(a) consists of 2 frames. Frame 2 is overlaid on Frame 1, and for clarity, is shifted slightly to the right and it has purple colour saturation. From Frame 1 to Frame 2, it can be observed that the 2 white cars have moved slightly forward down the road, while the black car has almost moved off the image. The simulation was performed for 1 frame period of 1/15 s. By monitoring the memristance changes of the memristors within the connecting fuse, the results are shown on Figs. 18(c) and 18(d) are obtained. From Fig. 18(b), we do not consider the effects of the on-center or off-center response. The lighter grey dots correspond to decreasing cM1 and increasing cM2 while the darker grey dots corresponds to the increasing cM1 and decreasing cM2. This result shows the computation of transient response, where only moving objects were picked up.

In order to accurately compute the direction and speed of the object, it is necessary to separate the movement of dark edges from the movement of light edges. Essentially, the on-center response and the off-center response have to be distinguished. This can be done by splitting the result in acquired in Fig. 18(b) based on the intensity of the input stimulus to each pixel. The off-center response is shown in Fig. 18(c), where only the transient response due to a moving dark object/edge is shown. The darker dots correspond to a disappearing dark edge while the lighter dots correspond to an appearing dark edge. Similarly, the on-center response, showing the transient response due to a moving light object/edge is shown in Fig. 18(d). The darker dots now correspond to an appearing light edge while the lighter dots correspond to a disappearing light edge.

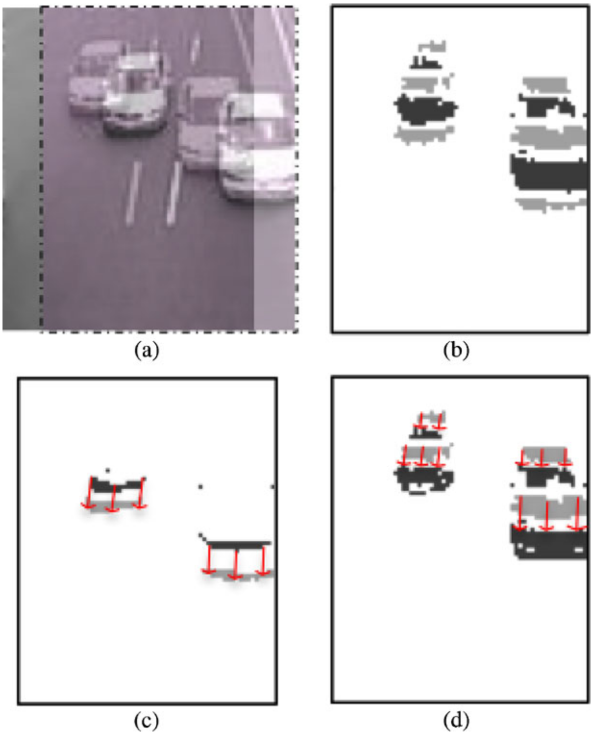
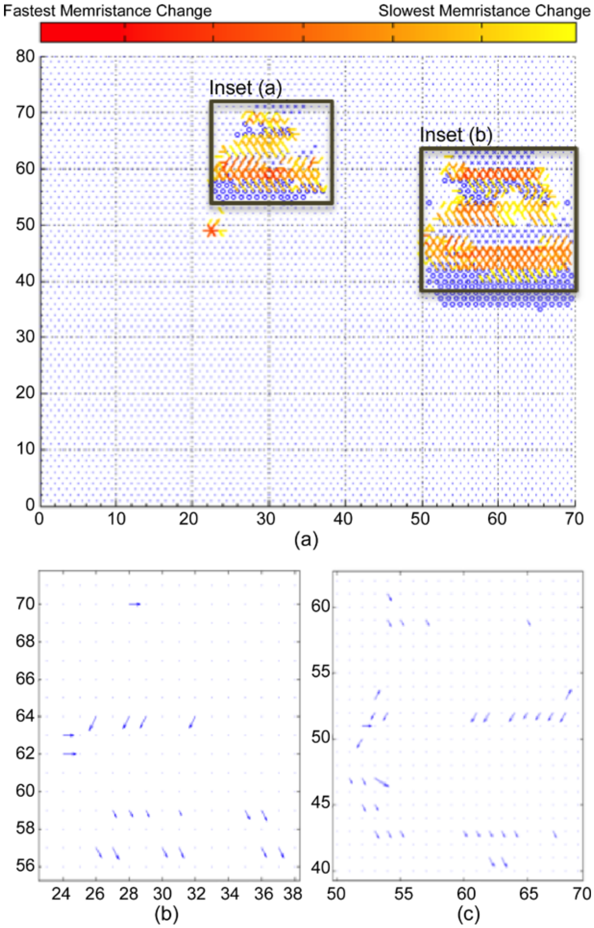


Fig. 18 Simulation of 2 frames from a video of traffic on a highway. (**a**) Input stimulus consists of 2 frames. Frame 2 is overlaid on Frame 1, and for clarity, it is shifted slightly to the right and it has purple colour saturation (**b**) Transient detection (**c**) OFF-Center response (**d**) ON-center response

5.3 Directional and Speed Detection

Given the output from the transient detection, it is possible to use a 1-layer hexagonal memristive grid to compute the direction and speed of the moving object. The results from the transient detection shown in Figs. 18(c) and 18(d) can be used as the input stimuli. To demonstrate this, the memristive grid was biased based on the off-center response from Fig. 18(d). The darker pixels will correspond to a positive voltage source into the memristive network while the lighter pixels will correspond to a ground. Figure 19(a) illustrates the 80 × 70 pixel hexagonal memristive grid where the voltage sources and ground are marked by "×" and "○" respectively. The rate of memristance changes of all interconnecting fuses are monitored and they are drawn in Fig. 19(a) with a value dependent colour variation. Since it is expected that the current will take the shortest path from the source to the ground, the rate of memristance changes of the fuses will provide an indication of the direction and flow of the current. At each of the moving edge pixel, an estimate of the optical flow can be computed by the vectorial sum of the rate of memristance change of the 6 interconnecting fuses. Figures 19(b) and 19(c) respectively show the results of this computation for Inset (a) and Inset (b) of Fig. 19(a). Our method provides local optical flow estimates at the brightness edges, and as illustrated in Figs. 19(b) and 19(c), was able to correctly determine that both cars were moving downwards.

Fig. 19 (**a**) Results illustrating the rate of memristance change of fuses within the 80 × 70 pixel network. Each memristive fuse is plotted with a color corresponding to the rate at which it's memristance changes. (**b**) Illustrates the optical flow estimate from Inset (**a**). (**c**) Illustrates the optical flow estimate from inset (**b**)



6 Conclusion

We first presented a biorealistic model of the OPL of the vertebrate retina which mimics early vision processing and then extended our proposed system for motion computation. Our proposed system is based on an hexagonal memristive grid implementation and it leverages on the scalability and non-linearity of memristors for emulating both the OPL and IPL of a vertebrate's retina. This implementation assists in minimising the overall complexity of other previously reported systems, while at the same time it achieves a local Gaussian filtering that facilitates an adaptive smoothing of both distorted and undistorted optical stimuli. Moreover, it was demonstrated that edge detection can be achieved by means of a simple memristor thresholding scheme implemented at the OPL's nodes outputs as well as through the collective evaluation of states of the memristive elements per node. Both smoothing and edge detection were assessed against distinct light conditions and it was shown that the proposed platform behaves in a similar manner as its biological counterpart. Finally, with yield being considered a very important aspect in deep sub-micron technologies and particularly practical memristive implementations, we have demonstrated that the proposed bio-inspired OPL model can effectively manage a relatively large variation in the devices' properties. Next, a novel memristive thresholding scheme was introduced to allow the detection of moving edges. The parallel processing nature of the proposed memristive network offers great potential in minimizing the computation and power requirements of conventional motion detection systems. In addition, a double-layered memristive network was constructed to enable to detection of ON-center and OFF-center transient responses. Applying the transient detection results into a 1-D memristive network, it was then shown that it is possible to generate an estimate of the speed and direction of the moving object. This study serves as a good proof of concept that memristive networks have indeed great potential in performing motion computation effectively and efficiently.

Acknowledgements The authors wish to acknowledge the financial support of the CHIST-ERA ERAnet EPSRC EP/J00801X/1 and EP/K017829/1.

References

1. Baccus, S.A., Olveczky, B.P., Manu, M., Meister, M.: A retinal circuit that computes object motion. J. Neurosci. **28**(27), 6807–6817 (2008)
2. Barranco, B.L., Gotarredona, T.S., Mesa, L.A.C., Carrasco, J.A.P., Ramos, C.Z., Masquelier, T.: On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. Front. Neurosci. **5** (2011)
3. Barron, J.L., Thacker, N.A.: Tutorial: computing 2d and 3d optical flow. In: Tina Memo (2005)
4. Benson, R.G., Delbruck, T.: Direction selective silicon retina that uses null inhibition. In: Advances in Neural Information Processing Systems, vol. 4, pp. 756–763. Morgan Kaufmann, San Mateo (1991)
5. Bernabe Linares-Barranco, Serrano-Gotarredona, T., Camuñas-Mesa, L.A., Perez-Carrasco, J.A., Zamarreño-Ramos, C., Masquelier, T.: On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. Front. Neurosci. **5**(26) (2011)
6. Horn, B.K.P., Schunck, B.G.: In: Determining Optical Flow (1981)
7. Bhushan, B.: Biomimetics: lessons from nature—an overview. Philos. Trans. R. Soc. A, Math. Phys. Eng. Sci. **367**(1893), 1445–1486 (2009)
8. Biolek, Z., Biolek, D., Biolkova, V.: Spice model of memristor with nonlinear dopant drift. Radioengineering **18**, 210–214 (2009)
9. Boahen, K.: A retinomorphic chip with parallel pathways: encoding INCREASING, ON, DECREASING, and OFF visual signals. Analog Integr. Circuits Signal Process. **30**(2), 121–135 (2002)
10. Borst, A.: Fly vision: moving into the motion detection circuit. Curr. Biol. **21**(24), R990–R992 (2011). CI: Copyright (c) 2011; JID: 9107782; CON: Curr Biol. 2011 Dec 20; 21(24), 2077–2084. PMID: 22137471; ppublish
11. Canals, R., Roussel, A., Famechon, J.L., Treuillet, S.: A biprocessor-oriented vision-based target tracking system. IEEE Trans. Ind. Electron. **49**(2), 500–506 (2002)
12. Capson, D.W., Maludzinski, R.A., Feuerstein, I.A.: Microcomputer-based interactive tracking of blood cells at biomaterial surfaces. IEEE Trans. Biomed. Eng. **36**(8), 860–864 (1989)
13. Carsten Steger, Ulrich, M., Wiedemann, C.: Machine Vision Algorithms and Applications (2008)

14. Carver, M.: Analog VLSI and Neural Systems. Addison-Wesley, Boston (1989)
15. chii Liu, S., Usseglio-Viretta, A.: Visuo-motor fly-like responses of a robot using avlsi motion-sensitive chips. In: 2nd Int. ICSC Symp. Neural Comput, pp. 23–26 (2000)
16. Chiu, C.-F., Wu, C.-Y.: Design of cmos elementary-motion-flow-selective image detector using the bjt-based silicon retina. In: Proceedings of 1997 IEEE International Symposium on Circuits and Systems, 1997. ISCAS'97, vol. 1, pp. 717–720 (1997)
17. Chua, L.: Memristor-the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
18. Chua, L., Mo Kang, S.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
19. Clark, J.J.: Authenticating edges produced by zero-crossing algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **11**(1) (1989)
20. Curcio, C.A., Allen, K.A.: Topography of ganglion cells in human retina. J. Comp. Neurol. **300**(1), 5–25 (1990)
21. Curcio, C.A., Sloan, K.R., Kalina, R.E., Hendrickson, A.E.: Human photoreceptor topography. J. Comp. Neurol. **292**(4), 497–523 (1990)
22. Davies, E.R.: Machine Vision: Theory, Algorithms, Practicalities. Morgan Kaufmann, San Francisco (2004)
23. Delbruck, T.: Silicon retina with correlation-based, velocity-tuned pixels. IEEE Trans. Neural Netw. **4**(3), 529–541 (1993)
24. Dowling, J.E.: The Retina—An Approachable Part of the Brain. The Belknap Press of Harvard University Press, Cambridge (1987)
25. Ebong, I.E., Mazumder, P.: Cmos and memristor-based neural network design for position detection. Proc. IEEE **100**(6), 2050–2060 (2012)
26. Eeckman, F.H., Colvin, M.E., Axelrod, T.S.: A retina-like model for motion detection. In: International Joint Conference on Neural Networks, 1989. IJCNN, vol. 2, pp. 247–249 (1989)
27. Gelencsér, A., Prodromakis, T., Toumazou, C., Roska, T.: Biomimetic model of the outer plexiform layer by incorporating memristive devices. Phys. Rev. E **85**, 041918 (2012)
28. Gomez, G.: Local smoothness in terms of variance: the adaptive Gaussian filter, pp. 815–824 (2000)
29. Huber, S.A., Franz, M.O., Bülthoff, H.H.: On robots and flies: modeling the visual orientation behavior of flies. Robot. Auton. Syst. **29**, 227–242 (1999)
30. Hutchinson, J., Koch, C., Luo, J., Mead, C.: Computing motion using analog and binary resistive networks. Computer **21**(3), 52–63 (1988)
31. Inc The MathWorks. MATLAB Image Processing Toolbox—User's Guide R2012b. The MathWorks, Inc, 3 Apple Hill Drive, Natick, MA 01760-2098 (2012)
32. Jiang, F., Shi, B.E.: The memristive grid outperforms the resistive grid for edge preserving smoothing. In: European Conference on Circuit Theory and Design, 2009. ECCTD 2009, pp. 181–184 (2009)
33. Kameda, S., Yagi, T.: Calculating direction of motion with sustained and transient responses of silicon retina. In: SICE 2002. Proceedings of the 41st SICE Annual Conference, vol. 3, pp. 1853–1858 (2002)
34. Kameda, S., Yagi, T.: A silicon retina system that calculates direction of motion. In: Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03, vol. 4, pp. IV-792–IV-795 (2003)
35. Kolb, H.: Organization of the outer plexiform layer of the primate retina: electron microscopy of Golgi impregnated cells. Philos. Trans. R. Soc. Lond. B, Biol. Sci. **258**(823), 261–283 (1970)
36. Lee, J.W., Chae, S.P., Kim, M.N., Kim, S.Y., Cho, J.H.: A moving detectable retina model considering the mechanism of an amacrine cell for vision. In: IEEE International Symposium on Industrial Electronics, 2001. Proceedings. ISIE 2001, vol. 1, pp. 106–109 (2001)
37. Li, H., Chung Lin, J.: Using fuzzy logic to detect dimple defects of polished wafer surfaces. IEEE Trans. Ind. Appl. **30**(2), 317–323 (1994)
38. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128 times 128 120 db 15 s latency asynchronous temporal contrast vision sensor. IEEE J. Solid-State Circuits **43**(2), 566–576 (2008)

39. Likharev, K.K., Mayr, A., Muckra, I., Türel, O.: Crossnets: high-performance neuromorphic architectures for cmol circuits. Ann. N.Y. Acad. Sci. **1006**(1), 146–163 (2003)
40. Lu, Y., Jain, R.C.: Behavior of edges in scale space. IEEE Trans. Pattern Anal. Mach. Intell. **11**(4), 337–356 (1989)
41. Mahowald, M.: An Analog VLSI System for Stereoscopic Vision. Kluwer Academic, Norwell (1994)
42. Maini, R., Aggarwal, Dr.H.: Study and comparison of various image edge detection techniques. Int. J. Image Process. **3**(1), 1–12 (2009)
43. Malamas, E.N., Petrakis, E.G.M., Zervakis, M., Petit, L., didier Legat, J.: A survey on industrial vision systems, applications and tools. Image Vis. Comput. **21**, 171–188 (2003)
44. Masaki, I.: Machine-vision systems for intelligent transportation systems. IEEE Intell. Syst. Appl. **13**(6), 24–31 (1998)
45. Mead, C.: Neuromorphic electronic systems. Proc. IEEE **78**(10), 1629–1636 (1990)
46. Meitzler, R.C., Strohbehn, K., Andreou, A.G.: A silicon retina for 2-d position and motion computation. In: EEE International Symposium on Circuits and Systems, 1995. ISCAS'95. vol. 3, pp. 2096–2099 (1995)
47. Mhani, A., Bouvier, G., Herault, J.: A contrast and motion-sensitive silicon retina. In: Solid-State Circuits Conference, 1995. ESSCIRC'95. Twenty-First European, pp. 326–329 (1995)
48. Netravali, A.N.: Motion: computation, synthesis and applications. In: The Sixteenth Conference of Electrical and Electronics Engineers in Israel, pp. 1–18 (1989)
49. Prodromakis, T., Bhushan, B.: Encyclopedia of Nanotechnology: Biomimetic Products. Springer, Berlin (2012)
50. Prodromakis, T., Pin Peh, B., Papavassiliou, C., Toumazou, C.: A versatile memristor model with nonlinear dopant kinetics. IEEE Trans. Electron Devices **58**(9), 3099–3105 (2011)
51. Ray, N., Acton, S.T., Ley, K.: Tracking leukocytes in vivo with shape and size constrained active contours. IEEE Trans. Med. Imaging **21**(10), 1222–1235 (2002)
52. Regan, D.: Human Perception of Objects. Sinauer, Sunderland (2000)
53. Roska, B., Werblin, F.: Vertical interactions across ten parallel, stacked representations in the mammalian retina. Nature **410**, 583–587 (2001)
54. Salaoru, I., Prodromakis, T., Khiat, A., Toumazou, C.: Resistive switching of oxygen enhanced tio[sub 2] thin-film devices. Appl. Phys. Lett. **102**(1), 013506 (2013)
55. Sanz, J.L.C., Petkovic, D.: Machine vision algorithms for automated inspection thin-film disk heads. IEEE Trans. Pattern Anal. Mach. Intell. **10**(6), 830–848 (1988)
56. Saripalli, S., Montgomery, J.F., Sukhatme, G.S.: Visually guided landing of an unmanned aerial vehicle. IEEE Trans. Robot. Autom. **19**(3), 371–380 (2003)
57. Serrano-Gotarredona Timothe Masquelier, T., Prodromakis, T., Indiveri, G., Linares-Barranco, B.: Stdp and stdp variations with memristors for spiking neuromorphic learning systems. Front. Neurosci. **7**(2) (2013)
58. Shafarenko, L., Petrou, M., Kittler, J.: Automatic watershed segmentation of randomly textured color images. Trans. Img. Proc. **6**(11), 1530–1544 (1997)
59. Sotelo, M.A., Rodriguez, F.J., Magdalena, L.: Virtuous: vision-based road transportation for unmanned operation on urban-like scenarios. IEEE Trans. Intell. Transp. Syst. **5**(2), 69–83 (2004)
60. Strukov, D.B., Likharev, K.K.: Cmol fpga: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices. Nanotechnology **16**(6), 888 (2005)
61. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**(7191), 80–83 (2008)
62. Toumazou, C.: Bio-inspired technologies. Ingenia **37** (2008)
63. Turel, Ö., Lee, J.H., Ma, X., Likharev, K.K.: Neuromorphic architectures for nanoelectronic circuits: research articles. Int. J. Circuit Theory Appl. **32**(5), 277–302 (2004)
64. Ullman, S.: Analysis of visual motion by biological and computer systems. Computer **14**(8), 57–69 (1981)
65. Wandell, B.A.: In: Foundation of Vision. Sinauer, Sunderland (1955)

66. Wang, F.Y.: Memristor for introductory physics. arXiv:0808.0286; Provided by the SAO/NASA Astrophysics Data System
67. Williams, R.: How we found the missing memristor. IEEE Spectr. **45**(12), 28–35 (2008)
68. Yang, W.-C., Lin, L.-J., Chiueh, H., Wu, C.-Y.: A cmos bio-inspired 2-d motion direction sensor based on a direction computation method derived from the directionally selective ganglion cells in the retina. IEEE Sens. J. **11**(12), 3341–3351 (2011)

Solid-State Memcapacitors and Their Applications

Jacek Flak and Jonne K. Poikonen

Abstract This chapter introduces the concept of a memcapacitor, and reviews different approaches to its physical realization. Also, practical constraints for their usage are assessed. Because of their compatibility with traditional circuit integration technologies, two approaches are particularly interesting: the ferroelectric capacitor and the memcapacitor constructed by appending metal-insulator-metal (MIM) capacitor with a memristive switching layer. Ferroelectric capacitors have already been in use for many years so the properties of this technology are relatively well researched. The MIM-memristor hybrid structure can take advantage of the vital research on memristive memories. With sufficiently large ratio of the OFF and ON resistances of a memristive material, the compound structure behaves as a memcapacitive system. Finally, the potential of memcapacitors for memory and logic applications as well as for artificial neural networks are discussed.

1 Introduction

After the memristor was proposed as the fourth basic element [5], the research was continued resulting in a systematic arrangement of circuit elements (including higher order nonlinear components) in form of a concise periodic table and a four-element torus [6]. Naturally, the characteristics of other possible components have been investigated and the concept of two new memory elements, i.e., memcapacitor (abbreviated from memory capacitor) and meminductor (abbreviated from memory inductor) appeared in 2008 [7]. Memristors, memcapacitors and meminductors have been categorized as sub-classes of the memristive, memcapacitive and meminductive systems, respectively [9]. Figure 1 presents the system of fundamental passive circuit elements with and without memory, which extends the four elements

J. Flak (✉)
VTT Technical Research Centre of Finland, Espoo, Finland
e-mail: jacek.flak@vtt.fi

J.K. Poikonen
BID Technology, University of Turku, Turku, Finland
e-mail: jokapo@utu.fi

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_26,

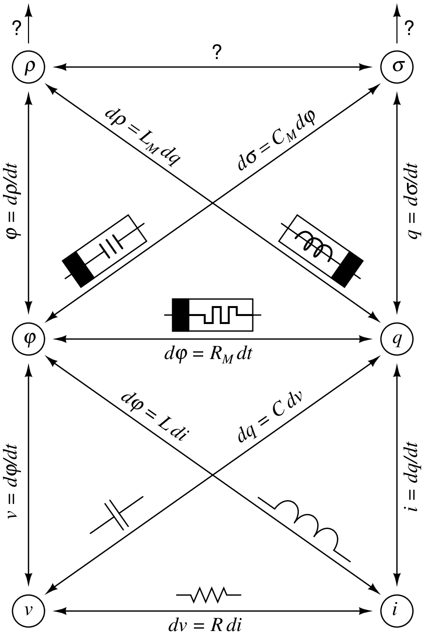


Fig. 1 Fundamental passive elements with and without memory

arrangement presented in [24] by adding the memcapacitor and meminductor. Memcapacitance C_M correlates the magnetic flux φ with the time integral of electronic charge σ, while meminductance L_M provides a relationship between the charge q and the time integral of flux ρ. Circuit elements with even higher order dynamics are yet to be defined, and whether their solid-state implementations are possible remains to be seen.

The memcapacitive systems can be divided into voltage- and charge-controlled. The voltage-controlled memcapacitive system is defined by

$$q(t) = C_M(x, V_C, t)\, V_C(t) \tag{1}$$

$$\dot{x} = f(x, V_C, t) \tag{2}$$

where $q(t)$ is the charge stored on the memcapacitor at the time t, $V_C(t)$ is the corresponding voltage, and C_M denotes the memcapacitance dependent on the state of the system x. For the n-th order system, x becomes a vector of n state variables ($x = [x_1, x_2, \dots, x_n]$). Equation (2) describes the dependence of the state of memcapacitive system x on its own history. Similarly, the charge-controlled memcapacitive system is defined by

$$V_C(t) = C_M^{-1}(x, q, t)\, q(t) \tag{3}$$

$$\dot{x} = f(x, q, t) \tag{4}$$

where C_M^{-1} stands for the inverse memcapacitance.

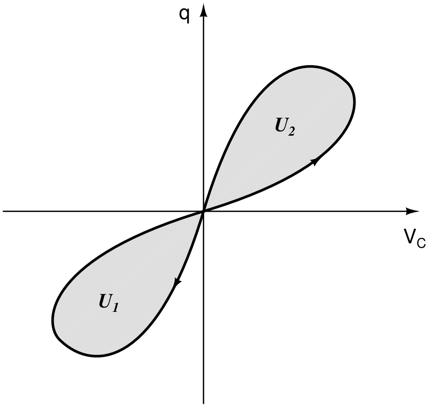


Fig. 2 Pinched hysteresis loop of a memcapacitive system [9]. The area of each shade region corresponds to the amount of energy added/removed in that half-period. For the marked loop direction, U_1 is negative and U_2 is positive

From the above system definitions a subclass of memcapacitor devices can be derived. For the case of voltage-controlled memcapacitors, Eqs. (1) and (2) can be written as

$$q(t) = C_M\left[\int_{t_0}^{t} V_C(\tau)d\tau\right]V_C(t) \tag{5}$$

and for the charge-controlled memcapacitors, Eqs. (3) and (4) yield

$$V_C(t) = C_M^{-1}\left[\int_{t_0}^{t} q(\tau)d\tau\right]q(t) \tag{6}$$

Analogically to I–V characteristics of memristors, memcapacitors exhibit a pinched hysteresis loop in the relationship between the control and output parameter. For voltage-controlled devices, the $q(t) = f[V_C(t)]$ characteristic may look as sketched in Fig. 2.

2 Physical Realizations of a Memcapacitor

The theoretical circuit elements can have multiple alternative physical realizations (vide ceramic, electrolytic, and MEMS capacitors). For instance, all types of resistance switching memories (exhibiting the pinched hysteresis loop) have recently been classified as memristors [8].

According to the theory, memcapacitor is a lossless component because it has no resistance [7]. Therefore, it could theoretically be used for a zero-power nonvolatile memory. However, just as in the case of all circuit elements, the physical realizations will certainly have some nonidealities leading to energy losses. Not many physical device concepts have been proposed yet, although the nonvolatile capacitance changes have been observed in, e.g., ferroelectric materials, nanopores [13] and certain diode structures [25].

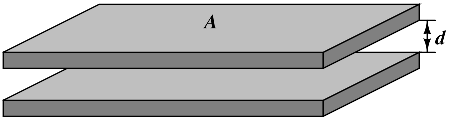


Fig. 3 Basic structure of a parallel-plate capacitor with electrode plates overlap area A and distance between the plates d

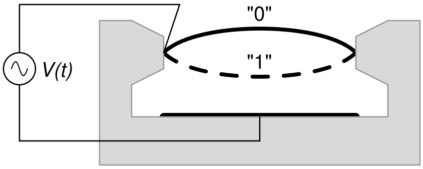


Fig. 4 Memcapacitor with elastic membrane

For simplicity, a parallel-plate capacitor shown in Fig. 3 will be considered as a basis for the following hardware realizations. The capacitance value of such a structure is given as:

$$C = \frac{A\varepsilon_0\varepsilon_r}{d} \tag{7}$$

where C is the capacitance value, A is the overlap area of the electrode plates, ε_0 is the vacuum permittivity (physical constant, $\varepsilon_0 \approx 8.854 \times 10^{-12}$ F/m), ε_r is the relative permittivity of a dielectric material (dielectric constant), and d is the distance between the electrodes. In this case, the change of capacitance value can be induced either by a modification of the device geometry and/or dimensions (A, d) or by the modulation of the material permittivity (ε_r).

2.1 Capacitor with Elastic Membrane Electrode

One concept of a memcapacitive device relies on replacing one electrode of a parallel-plate capacitor with a strained elastic membrane [16] as shown in Fig. 4. Such a system has two equilibrium positions corresponding to stable high and low capacitance configurations. Actually, this structure could be classified as a microelectromechanical system (MEMS) rather than a classical solid-state device. Nevertheless, it could possibly be integrated with electronics on the same chip or as a two-chip stack.

As a result of a strained membrane forming one of the electrodes, and thus a nonuniform distance between the plates across the structure, this approach may suffer from poorly defined capacitance. For instance, creating a model of this device that would account for the capacitance scaling with respect to the plate area may be challenging. However, the structure can be modified, so that a stiff top electrode is used in combination with flexible anchors as shown in Fig. 5.

Fig. 5 Modified structure of a memcapacitor with moving electrode plate

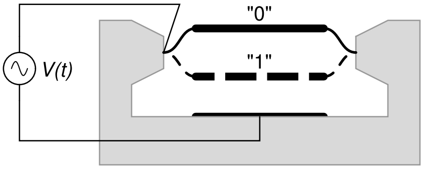
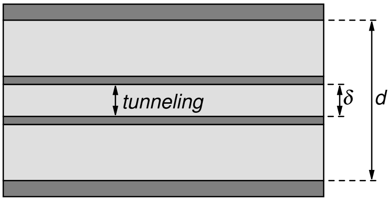


Fig. 6 Solid-state memcapacitor with multilayered structure



2.2 Capacitor with Multiple Metal and Insulator Layers

Another approach to the hardware realization of memcapacitor relies on a solid-state structure formed by multiple metal and dielectric layers embedded within a regular parallel-plate capacitor [17]. The simplest version with only two inner metal layers is shown in Fig. 6. The capacitance changes result from charge tunneling between the inner metal layers (distance $\delta \ll d$). The dependence of the tunneling current on the applied voltage is highly nonlinear, therefore providing a natural threshold between programming (high-voltage pulses) and sensing (low voltages) the state of the device. Within certain ranges of the electric field, such a structure exhibits hysteretic charge-voltage and capacitance-voltage characteristics. This structure can easily be fabricated using traditional methods for deposition of consecutive layers with lithographically defined lateral dimensions.

2.3 Ferroelectric Capacitor

Crystalline dielectric materials can exhibit spontaneous electrical polarization, caused by the displacement of positively and negatively charged ions with respect to each other within the material. In ferroelectric materials the direction of this spontaneous polarization can be reoriented by applying an external electric field. Figure 7 shows the hysteretic relationship between the voltage applied to a ferroelectric material and the polarization, P. Because the programmable polarization leads to a nonvolatile capacitive memory effect, ferroelectric materials exhibit memcapacitive behavior.

Typically, the ferroelectric capacitor has a structure of metal-insulator-metal (MIM) capacitor, in which the insulating layer is formed by ferroelectric material (e.g., $Pb(Zr_{1-x}Ti_x)O_3$ or $BaTiO_3$), as illustrated in Fig. 8. When a voltage pulse is

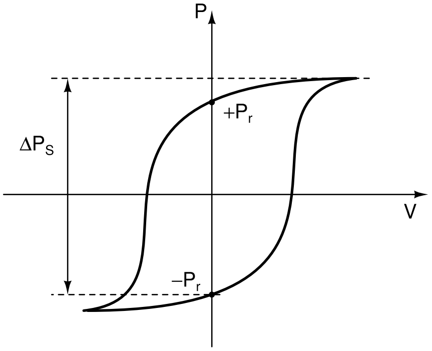


Fig. 7 Hysteretic relationship between the polarization in ferroelectric materials and the applied voltage. The remanent polarization states ($+P_r$ and $-P_r$) represent the stored "0" and "1". ΔP_S denotes the reversal of polarization detected as an additional switching charge ΔQ_S

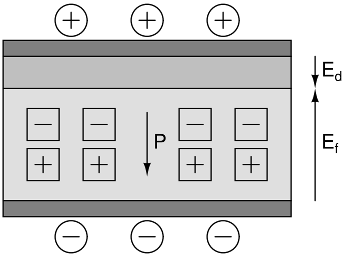


Fig. 8 Structure of a ferroelectric capacitor. E_d denotes the electric field of an antiferroelectric interface layer, which is a result from defects and interface effects in the ferroelectric material

applied to such a capacitor, a current flow occurs, which depends on the initial polarization of the device (i.e., the previously stored binary state). If the applied pulse has the same polarity, the current flow simply corresponds to charging the capacitor. If an opposite polarity voltage pulse is applied, an additional charge ΔQ_S is required to reverse the direction of the polarization, leading to a larger current flow through the device. The difference between these two signals is used to determine the state stored on the device. This procedure (conditional reversal of the polarization) naturally makes the readout process destructive [3].

2.4 Capacitor Appended with Memristive Layer

A solid-state memcapacitive device can also be composed by stacking a traditional MIM capacitor and a memristor as shown in Fig. 9(a). In the high resistance state, the memristive material such as transition metal oxide (e.g., TiO_2 or HfO_2) can be regarded as a dielectric. On the other hand, when programmed to a low resistance state, it behaves as an ohmic contact with finite resistance. Such a structure could benefit from fabrication techniques being developed for memristive arrays. An analogous memcapacitor structure has also been proposed by HP Labs [4, 19]. Moreover, it may be possible to omit the inner metal electrode and assume the conduction to stop at the edge of the stable dielectric, as shown in Fig. 9(b). Such a

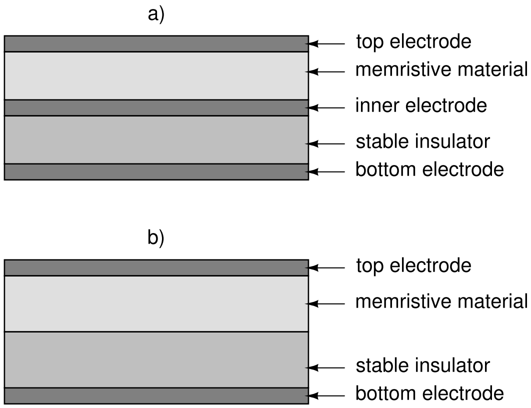


Fig. 9 Physical structures of a solid-state memcapacitor based on capacitor appended by memristive layer: (**a**) basic stack of a memristor and MIM capacitor, and (**b**) structure without the inner electrode

structure has been considered by Micron Technology [18]. In this way, the fabrication process can be simplified (reduced number of masks). However, without the inner metal layer, the high capacitance value may not be well defined. Unless the contact interface between the oxides is highly conductive, the area of the inner electrode will depend on the uniformity of the resistance changes across the device. It may turn out to be a significant problem in case of devices with switching dynamics based on the formation and rupture of local conducting filaments [21]. The structure containing the inner metal electrode is inherently immune to this issue.

This type of a memcapacitor in either high or low capacitance state can be represented by the corresponding simple circuit equivalents, shown in Fig. 10. When the memristor's off-resistance $R_{M\_OFF} \to \infty$, the capacitive component dominates its impedance in the high resistance state, so that the R_{M_OFF} could be neglected. In such a case, the memcapacitor structure can be treated as a serial connection of two capacitors with the total capacitance value $C_M = C_{MR} \cdot C_C/(C_{MR} + C_C)$. Analogically, when the memristor is programmed to low resistance state and its $R_{M\_ON} \to 0$, the memcapacitance value can be approximated as $C_M = C_C$. In practice, however, a designer should account for the resistive components, especially when targeting analog applications. For instance, in a dense crossbar architecture, the capacitance of a memristive layer programmed to high resistance state may well be on order of femtofarads. Assuming the R_{M_OFF} in the range of megohms, the C_{MR} would be discharged with time constant $\tau_{MR} = C_{MR} \cdot R_{M\_OFF}$, i.e., within nanoseconds. Therefore, the material to be used for the memristive layer of a memcapacitor should allow for maximal resistance ratio (preferably, $\gg 10^6$), so that the resistive components could be neglected. At the same time, it should exhibit as high as possible dielectric constant to achieve large capacitance within a small area. Meeting these two requirements simultaneously may prove difficult. For example, TiO_2 may reach $\varepsilon_r = 150$ but typically, the TiO_2-based memristors show a resistance ratio limited to 10^4 [12]. On the other hand, a GeO_x-based devices with $R_{M\_OFF}/R_{M\_ON} = 10^9$ have been reported [20], but the germanium oxides have significantly lower dielectric constant ($\varepsilon_r \approx 10$).

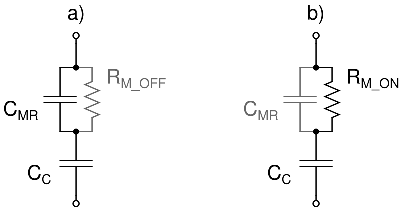


Fig. 10 Circuit equivalents of the memristor-based memcapacitor programmed to (**a**) low and (**b**) high capacitance values, respectively

2.4.1 Programming of the Memristor-Capacitor Hybrid

If the inner metal electrode is extending outside the device (resulting in a 3-terminal structure), the programming voltage can be applied between the top and inner electrode (see Fig. 9(a)). However, the access to the inner electrode would limit the possible scaling. Moreover, in case of dense nanowire fabrics, the overall capacitance of such a device could significantly be altered by the stray capacitance of the additional programming connection, which should be considered in the circuit design and layout.

Alternatively, the programming pulses can be applied between the top and bottom electrodes. Changing the capacitance value from low to high (i.e., switching the memristive layer from R_{M_OFF} to R_{M_ON}) is still relatively straightforward. With sufficiently large amplitude of the programming voltage V_P, the voltage across the memristive layer (neglecting the influence of R_{M_OFF})

$$V_{MR} \approx \frac{V_P \cdot C_{MR}}{C_{MR} + C_C} \tag{8}$$

will exceed the memristor switching threshold. On the other hand, changing the capacitance value from high to low is more complicated, because the current flow through the device (C_C and R_{M_ON}) needs to be large enough to cause the reverse switching of the memristor resistance from R_{M_ON} to R_{M_OFF}. Since a capacitor in series inherently imposes a high-pass filtering, it presents a smaller impedance for rapidly rising edge of a voltage pulse. Therefore, the programming signals with shapes as shown in Fig. 11 are proposed. Generating such waveforms would require a dedicated complex control electronics, which could be shared by many (or even all) memcapacitive devices in the system. This scheme could possibly be applied to the both memcapacitor structures of Fig. 9.

Figure 12 presents the AC simulation waveforms of the memcapacitor equivalent circuit with $C_C = C_{MR} = 5$ fF and R_M varying from 100 GΩ to 1 Ω. When the voltage at the inner electrode drops to 1.5 V, the resistive component becomes negligible and the structure behaves as a capacitive voltage divider. Interestingly, the minimum value of R_{M_ON} should not be below 1 kΩ, because an extremely high frequency would be required for the programming pulses to switch the memcapacitor to low capacitance state (i.e., switch the memristive layer from R_{M_ON} to R_{M_OFF}).

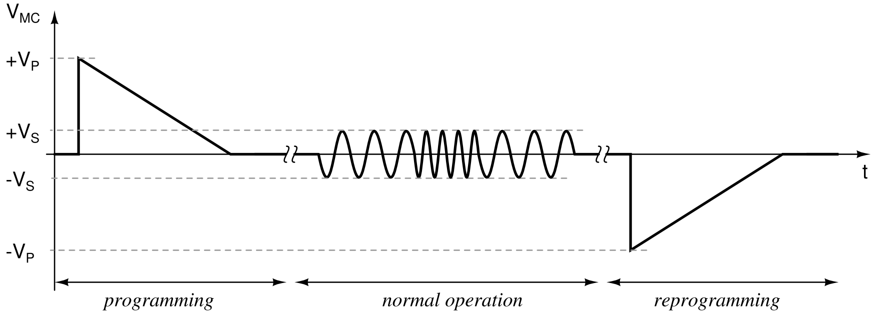


Fig. 11 Example of programming voltage waveforms for memcapacitor made of MIM capacitor and memristor stack (assuming bipolar-switching memristor). The amplitude of programming pulses V_P is significantly higher than the amplitude of processed signals V_S

3 Applications of Memcapacitors

3.1 Device Models

The concept of a memcapacitor is relatively new, and thus its modeling is not very advanced yet. Usually, derivatives of the state equation defining the element are used to qualitatively assess its behavior or certain properties. However, the models proposed in [1] and [2] already allow for simulating electronic circuits with memcapacitors using standard CAD software. These models are generic and can further be extended or parametrized to more realistically reflect a specific implementation and its technology parameters. Moreover, these models can be utilized in continuous-time simulations, in which the intermediate processing results are supposed to affect the value of a memcapacitance—a case not possible with the simple (static) circuit equivalents of Fig. 10.

3.2 Memory

Memristors are often projected to replace the flash-type devices in future generations of nonvolatile memories and memcapacitors could serve this purpose as well. Actually, the ferroelectric memcapacitors have already been on the market since 1990's. Due to the nonvolatile polarization and the lack of any DC current for operation, ferroelectric capacitors lend themselves naturally to low-power embedded memory applications. The technology development has resulted in a very long retention time ($>$10 years), unlimited endurance and an access time under 50 ns [23]. However, scaling of the ferroelectric random access memory (FeRAM) imposes a challenge as the typical one-transistor and one-capacitor (1T1C) configuration (shown in Fig. 13) has not been able to keep up with the rapid downscaling of CMOS technology, due to the area required by the separate Fe-capacitor. A denser memcapacitor-based

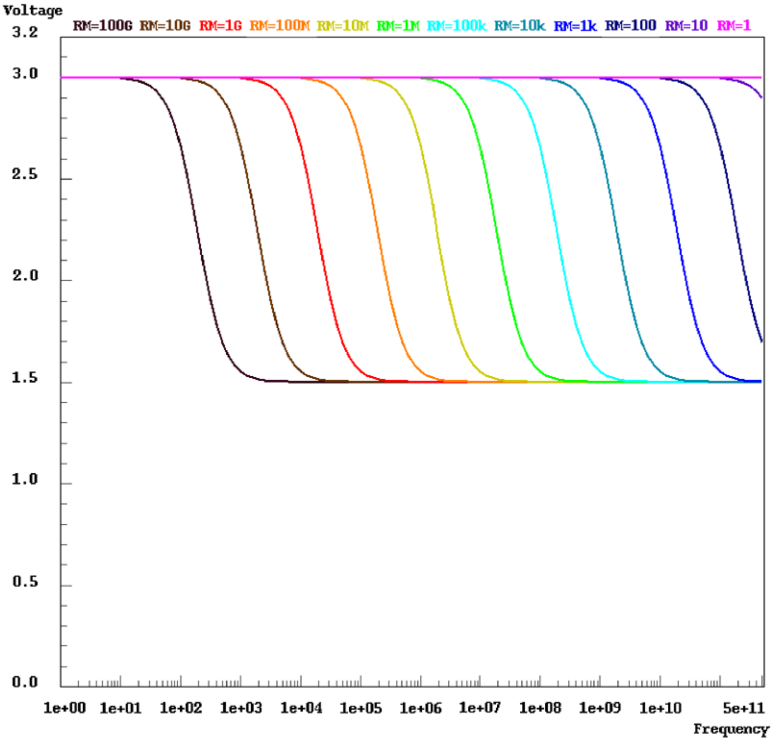


Fig. 12 AC simulations of the MIM-memristor hybrid memcapacitor showing the frequency dependence of the inner electrode potential on the memristor resistance value, when a 3 V signal is applied between the top and bottom electrodes (see Figs. 9(a) and 10)

memory could be obtained with a 1T-configuration, in which the memcapacitor is replacing the gate dielectric of a MOS transistor. Such a structure has already been proposed in form of ferroelectric field-effect transistor (FeFET), shown in Fig. 14. However, the interface between the ferroelectric material in the gate stack and the underlying Si device leads to disturbances in transistor parameters and decreased retention time [23]. To overcome this problem, one can apply an insulating buffer layer between the Fe-material and silicon, creating a Metal-Ferroelectric-Insulator-Silicon (MFIS) capacitor. This improves the performance but naturally also leads to a thicker gate stack, possibly requiring more area to achieve the same capacitance. Recently, a new FeFET device, based on a Si-doped HfO_2 gate dielectric with a thickness of <10 nm was reported [15]. HfO_2, due to its higher relative permittivity, is often replacing SiO_2 for gate dielectric in modern deep sub-micron CMOS technologies. This is an important observation because the process and material compatibility with downscaling CMOS structures would significantly increase the desirability of memcapacitors as nonvolatile memory contender. Also, the memcapacitors based on memristive materials have been proposed for the gate dielectric of a MOSFET [18], but its experimental verification has not yet been reported.

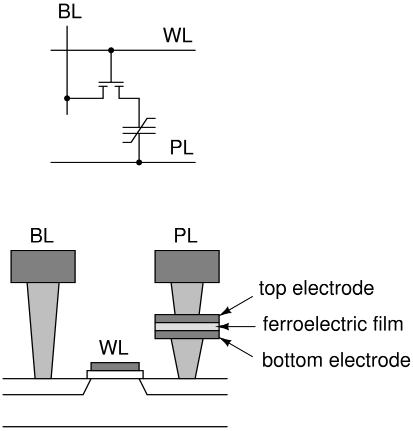


Fig. 13 1T1C FeRAM cell schematics and the cross section of its structure with marked bit line (BL), word line (WL), and plate line (PL)

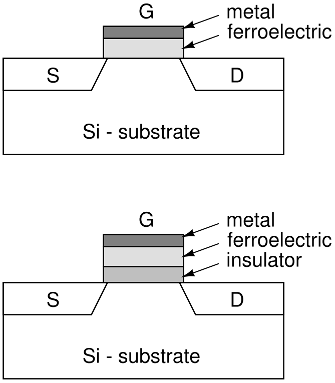


Fig. 14 Different structures of the ferroelectric field-effect transistor (FeFET)

Memcapacitors can be used for memory applications in two ways. First, only the memcapacitance value is used to store the data and thus a single bit of information is stored in each device. This mode of operation resembles the use of memristors for nonvolatile memory arrays. However, the achievable density of a memcapacitive matrices (in either 1T1C or even 1T configuration) will not likely be as high as the memristive nanowire crossbar nets. The second approach is to use the charge stored on a memcapacitor for information coding. In such a case, a single memcapacitor can have three distinguished states. Namely, discharged capacitance ($Q_{MC} = 0$), and charged capacitance programmed to either “small” ($Q_{MC} = Q_S$) or “large” ($Q_{MC} = Q_L$) value. Each of these cases can be assigned a separate logic level, e.g., LO, MID, and HI, respectively. In this way, the array of memcapacitors can effectively store more data than when relying on capacitance value only. Table 1 shows how two memcapacitors can be used to represent nine different values. However, in this method the memory is rather volatile. Moreover, with downscaling the memcapacitor plate dimensions, the requirements imposed on the sensing amplifier

Table 1 Data storage density of two triple-valued symbols achievable with two memcapacitors

Q_{MC1}	Q_{MC2}
LO	LO
LO	MID
LO	HI
MID	LO
MID	MID
MID	HI
HI	LO
HI	MID
HI	HI

may limit the application to local cache-type memories (moderate capacity, short retention times).

The ability to store three distinctive logic values on a single device also indicates a potential for memcapacitors to be used in multiple-valued logic circuits. Multiple-valued logic repeatedly attracts researchers' attention because of its potential to provide a high function density with lower connectivity requirements and lower power consumption than classical binary logic. Alternatively, memcapacitors could also replace the standard capacitors in capacitive threshold logic (CTL) gates providing an additional degree of freedom in circuit design. A generic design style based on CTL gates for synthesis of multiple-valued functions has already been demonstrated [22].

3.3 Tunable Analog and Neuromorphic Circuits

A device with programmable capacitance is also very attractive for tunable analog circuits and system calibration schemes. For instance, memcapacitors could replace varactors and capacitor banks or complement them for increased tuning accuracy. On the other hand, the frequency-dependent behavior of the MIM-memristor hybrid (indicated in Fig. 12) may be utilized for filters and/or oscillator applications. However, its resistive components will naturally limit the achievable quality factors.

Another opportunity to exploit the potential of a memcapacitor is to embed it into the gate stack of a MOSFET in the same way it is done for FeFET, leading to a transistor with threshold voltage programmable in non-volatile manner. Such a device may prove useful for a wide range of analog circuits, including amplifiers, comparators, etc. The inherent tunability may also be utilized for implementing learning algorithms in neuromorphic circuits. Recently, a spike-timing-dependent synaptic plasticity (STDP) mechanism has been demonstrated with a device resembling FeFET structure, in which the conductivity of a ZnO-based transistor channel is controlled through a ferroelectric gate [26]. The advantage of this approach over

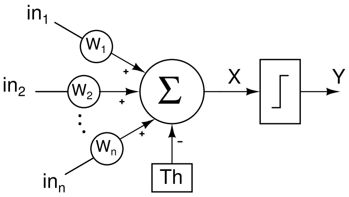


Fig. 15 McCulloch and Pitts neuron model. The output (Y) is activated whenever the sum of W_i-weighted input signals (in_i) exceeds the applied threshold (Th)

the standard memristor implementations of STDP is that the learning can be performed simultaneously with signal processing (and not in separate phases), which is the case in biological nervous systems, and thus may bring us a step closer to a brain-like processor.

3.4 CNN Cell with Memcapacitors

Apart from memory for data and/or program storage, memcapacitors can be used for synaptic connections between the cells of cellular neural/nanoscale network (CNN) as well as for setting the neuron activation threshold [10]. The proposed cell implements the McCulloch-Pitts neuron model, shown in Fig. 15, with activation function taking form of a step function:

$$V_Y = f(V_X) = \begin{cases} V_{DD}, & V_X \geq V_{DD}/2, \\ 0, & V_X < V_{DD}/2, \end{cases} \tag{9}$$

where voltages V_Y and V_X represent the neuron output and state, respectively. The proposed cell structure, presented in Fig. 16, has been derived from the floating-gate neuron structure [11]. For full functionality, the cell needs to include also local memories and the transient mask circuitry (not shown for clarity). The single-bit weight programmability [14] makes it versatile enough to perform all binary image processing either directly or through multi-step algorithms, and the application of memcapacitors to the synaptic connections further extends the cell capabilities.

Each synapse consists of a single memcapacitance C_M and three switches. The switch controlled by the template coefficient $AB_{i,j}$ is used to enable the given synapse to influence the cell state V_X, while the $start$ signal initiates the processing. Switches $pr\_s_{i,j}$ are used for connecting the memcapacitor to the programming voltages V_{pr1} and V_{pr2}. Since higher amplitude pulses are usually required for programming, it is necessary to separate the memcapacitors from the rest of circuitry during the programming phase. The $AB_{i,j}$ and the *start*-driven switches can serve this purpose. Since the output voltage V_Y is provided by the cell output buffer (a relatively strong driver) it can be used directly for the neighborhood inputs. This enables the synapse to have either a positive ($V_Y = V_{DD}$) or a negative ($V_Y = 0$) contribution to the cell state. The logical contribution to the cell state from each synapse is:

$$S_{i,j} = AB_{i,j} \cdot (C_M/C_u) \cdot (Vy_{i,j}/V_{DD}) \tag{10}$$

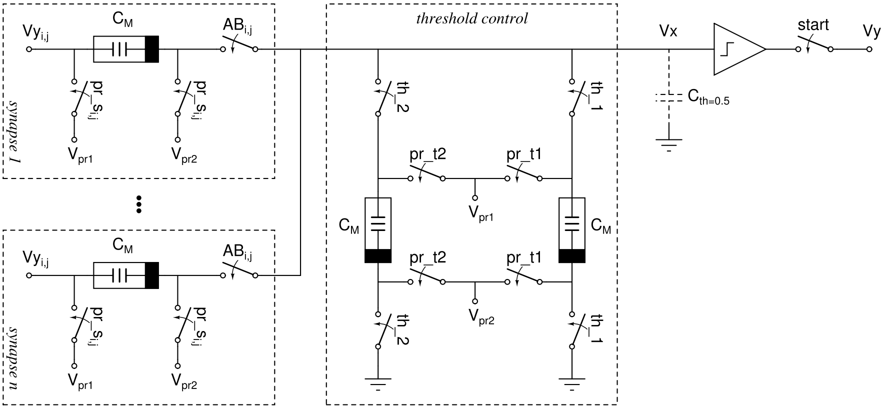


Fig. 16 CNN cell utilizing memcapacitors for synapses and activation threshold

Table 2 Truth table of the binary-programmable memcapacitive synapses

$AB_{i,j}$	C_M	$Vy_{i,j}$	$S_{i,j}$
0	$\times$[a]	$\times$[a]	0
1	C_u	0	-1
1	C_u	V_{DD}	$+1$
1	$2C_u$	0	-2
1	$2C_u$	V_{DD}	$+2$

[a] $\times$ = do not care

where C_u denotes an unity capacitance. Assuming that in the low capacitance state $C_M = C_u$ and in the high capacitance state $C_M = 2C_u$, $S_{i,j} \in [-2, -1, 0, +1, +2]$. The possible combinations are collected into a truth table of a memcapacitive synapse (Table 2).

The threshold control structure consists of two pull-down paths, each containing a memcapacitor and four switches. Additionally, the gate capacitance of the buffer's first stage is designed so that it forms a capacitive path corresponding to a threshold of 0.5 ($C_{th=0.5} = 0.5C_u$) shown with a dashed line in Fig. 16. Control signals *th*_1 and *th*_2 are used to enable the corresponding path, and thus adding to the threshold value. The switches driven by signals pr_t1 and pr_t2 are used for programming the memcapacitance values. Assuming again that C_M may take value of either C_u or $2C_u$ (as in the case of synapses), the overall threshold can be set to 0.5, 1.5, 2.5, 3.5, or 4.5, although, a threshold larger than 3.5 is not required in the binary programming scheme of the 8-connected network [14]. Nevertheless, it may be useful for processing some more complex templates.

The proposed CNN cell structure having nine synapses (for a fully connected first order neighborhood, the indices $i, j \in [-1, 0, +1]$ and $S_{0,0}$ represents self-feedback) has been designed for 0.35 μm CMOS technology and simulated using the circuit equivalents of Fig. 10 for modeling the memcapacitors (configurations

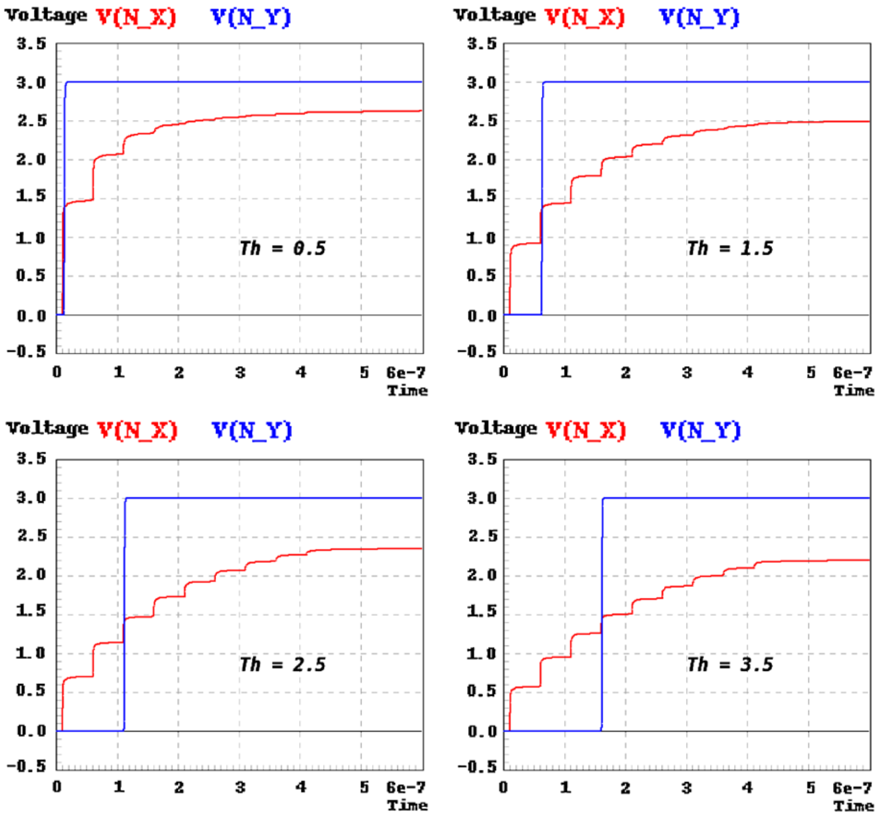


Fig. 17 Simulation waveforms of the state (*red lines*) and output (*blue lines*) voltages with threshold set to 0.5, 1.5, 2.5, and 3.5, respectively

kept constant during each simulation run). The basic perceptron operation has been verified through transient simulations with $R_{M\_OFF} = 100$ MΩ, $R_{M\_ON} = 1$ Ω, $C_C = C_{MR} = 80$ fF ($C_u = 40$ fF), $V_{DD} = 3$ V and an arbitrary time step of 1 ns, as shown in Fig. 17. For each threshold value (0.5, 1.5, 2.5, and 3.5), the nine synapses (with $S_{i,j} = +1$) are being consecutively switched on. The neuron output is activated each time the sum of inputs exceeds the programmed threshold value.

The transient simulation results presented in Fig. 18 show the ability of the proposed cell to deal with the negative synapse contributions. Namely, the synapse activated as the second one has $Vy_{i,j} = 0$ and $C_M = 2C_u$, and thus its contribution to the cell state $S_2 = -2$. Additionally, in the case (b) shown on the right hand side, the fourth activated synapse has $S_4 = +2$ and the fifth synapse contributes $S_5 = -1$. Therefore, this waveform proves the proper operation of the cell with the synapse contributions of all possible values.

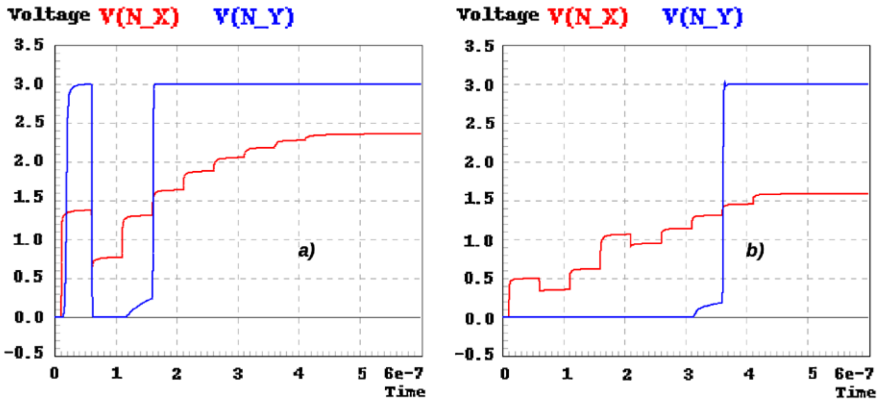


Fig. 18 The cell state (*red lines*) and output (*blue lines*) voltages with (**a**) $Th = 0.5$ and $S_2 = -2$, and (**b**) $Th = 3.5$, $S_2 = -2$, $S_4 = +2$, $S_5 = -1$. The contribution of all the other synapses are $S_i = +1$

4 Conclusions

Solid-state memcapacitor is a very promising emerging nanodevice, for which a broad application range from nonvolatile memories to tunable analog circuits can be envisioned and neural hardware implementations have already been proposed. When applied to capacitively coupled cellular nanoscale networks for synaptic connections and threshold control, the cell capabilities extend beyond the functionality provided by the binary-programming scheme. Moreover, embedding the memcapacitor into gate stack of a MOSFET enables implementions of learning algorithms in the way that allows for learning and processing to be conducted concurrently.

Different technologies and structures can be used to realize memcapacitive devices. Especially promising approaches are the ferroelectric capacitors and the memristor-based components, because of their compatibility with CMOS integration technologies.

It can be expected that the research on physical realizations of memcapacitors as well as circuit designs benefiting from those devices will soon gain a momentum, in similar way as it happened for memristors.

Acknowledgements This work was funded by the Academy of Finland under grants 140108 and 140290.

References

1. Biolek, D., Biolek, Z., Biolkova, V.: SPICE modelling of memcapacitor. Electron. Lett. **46**(7), 520–522 (2010). doi:10.1049/el.2010.0358
2. Biolek, D., Biolek, Z., Biolkova, V.: Behavioral modeling of memcapacitor. Radioengineering **20**(1), 228–233 (2011)

3. Böttger, U., Summerfelt, S.R.: Ferroelectric random access memories. In: Waser, R. (ed.) Nanoelectronics and Information Technology, 2nd edn. Wiley/VCH, New York/Weinheim (2005)
4. Bratkovski, A.M., Williams, R.S.: Memcapacitor. Patent application available on-line: http://www.freepatentsonline.com/20120039114.pdf (2010)
5. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
6. Chua, L.O.: Nonlinear circuit foundations for nanodevices, part I: the four-element torus. Proc. IEEE **91**(11), 1830–1859 (2003)
7. Chua, L.O.: In: Invited Speech at the Memristor and Memristive Systems Symposium, Berkeley, CA (2008)
8. Chua, L.O.: Resistance switching memories are memristors. Appl. Phys. A **102**(4), 765–783 (2011)
9. Di Ventra, M., Pershin, Y.V., Chua, L.O.: Circuit elements with memory: memristors, memcapacitors and meminductors. Proc. IEEE **97**(10), 1717–1724 (2009)
10. Flak, J.: CNN cell with memcapacitive synapses and threshold control circuit. In: Proc. of the 13th International Workshop on Cellular Nanoscale Networks and Their Applications, CNNA 2012 (2012). doi:10.1109/CNNA.2012.6331414
11. Flak, J., Laiho, M., Halonen, K.: FG-MOS neuron for binary CNN. In: Proc. of SPIE, vol. 5839 (2005). doi:10.1117/12.608059
12. Ha, S.D., Ramanathan, S.: Adaptive oxide electronics: a review. J. Appl. Phys. **110**, 071101 (2011). doi:10.1063/1.3640806
13. Krems, M., Pershin, Y.V., Di Ventra, M.: Ionic memcapacitive effects in nanopores. Nano Lett. **10**(7), 2674–2678 (2010)
14. Laiho, M., Paasio, A., Flak, J., Halonen, K.: Template design for cellular nonlinear networks with 1-bit weights. IEEE Trans. Circuits Syst. I **55**(3), 904–913 (2008)
15. Martin, D., et al.: Downscaling ferroelectric field effect transistors by using ferroelectric Si-doped HfO_2. In: 13th International Conference on Ultimate Integration on Silicon (ULIS), pp. 195–198 (2012)
16. Martinez, J., Pershin, Y.V.: Bistable nonvolatile elastic-membrane memcapacitor exhibiting a chaotic behavior. IEEE Trans. Electron Devices **58**(6) (2011)
17. Martinez, J., Di Ventra, M., Pershin, Y.V.: Solid-state memcapacitor, pp. 1–7 (2009). arXiv:0912.4921v2 [cond-mat.mes-hall]
18. Meade, R.E., Sandhu, G.S.: Memcapacitor device, field effect transistor devices, non-volatile memory arrays, and methods of programming. Patent application available on-line: http://www.freepatentsonline.com/20110199815.pdf (2011)
19. Pickett, M.D., Borghetti, J., Yang, J.: Two terminal memcapacitive device. Patent application available on-line: http://www.freepatentsonline.com/2012014184.pdf (2011)
20. Rahaman, S.Z., et al.: Record resistance ratio and bipolar/unipolar resistive switching scenario using novel Cu/GeOx/W memory device. In: Ext Abstracts of the 2011 International Conference on Solid-State Devices and Materials, pp. 1021–1022 (2011)
21. Sawa, A.: Resistive switching in transition metal oxides. Mater. Today **11**(6), 28–36 (2008)
22. Schmid, A., Leblebici, Y.: Realisation of multiple-valued functions using the capacitive threshold logic gate. IEE Proc., Comput. Digit. Tech. **151**(6), 435–447 (2004)
23. Setter, N., et al.: Ferroelectric thin films: review of materials, properties, and applications. J. Appl. Phys. **100**, 051606 (2006)
24. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
25. Sun, J., Lund, E., Maximov, I., Xu, H.Q.: Memristive and memcapacitive characteristics of a Au/Ti-HfO_2-InP/GaAs diode. IEEE Electron Device Lett. **32**(2), 131–133 (2011)
26. Ueda, M., et al.: Biologically-inspired learning device using three-terminal ferroelectric memristor. In: 70th Annual Device Research Conference (DRC), pp. 275–276 (2012)

Memristive Stateful Logic

Eero Lehtonen, Jussi H. Poikonen, and Mika Laiho

Abstract Memristive stateful logic refers to a form of computational logic in which memristors both store logic values and perform logical operations on these values. We present a generalized form of memristive stateful logic and define the logic operations realizable in this form. We also consider the CMOS circuitry required for reliable implementation of memristive stateful logic. Furthermore, synthesis of arbitrary Boolean functions and the parallelization of stateful logic to memristive crossbars is presented.

1 Introduction

Stateful logic refers to a form of computational logic in which the same devices both store logic values and perform logical operations. *Implication logic* was proposed as a specific realization of stateful logic for memristors by Phil Kuekes in a talk presented at the first Memristor and Memristive Systems Symposium at University of California, Berkeley in 2008 [5]. In this form of logic, values of Boolean variables are represented as memristances, and the stateful logic operation is realized simply by programming the state of a memristor conditionally to the state of another memristor. The programming of memristors is realized by driving the memristors with *conditional* and *programming* voltages, for example by using auxiliary CMOS circuitry.

In 2010, Borghetti et al. [1] published measurements of material implication operations performed using memristive switches, and demonstrated the computation of the logical NAND operation as a sequence of two implication operations between

E. Lehtonen (✉) · M. Laiho
BID Technology, University of Turku, Turku, Finland
e-mail: elleht@utu.fi

M. Laiho
e-mail: mlaiho@utu.fi

J.H. Poikonen
Department of Communications and Networking, Aalto University, Espoo, Finland
e-mail: jussi.poikonen@aalto.fi

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_27,

two input memristors and an output memristor. In [10] we proved that any Boolean function with N inputs can be synthesized using $N+2$ memristors—N memristors are needed to store the values of the inputs, and two auxiliary memristors are used for computation.

Material implication is not the only stateful logic operation which can be realized using conditional programming of memristors. In [4] it was shown how logical AND can be computed directly as a stateful logic operation. In [12] we considered a stateful logic operation called *converse nonimplication*, and showed that if rectifying memristors are used, this operation can be implemented with very simple CMOS circuitry even when the computing is performed within a memristive crossbar array. The fourth possible stateful logic operation that uses the computing scheme proposed by Kuekes is the logical OR. All of these operations will be discussed in more detail in the rest of this chapter.

Memristive stateful logic is sequential, and requires multiple elementary operations to synthesize a given Boolean function. Therefore minimization of the lengths of computational sequences is a significant task, and is considered for example in [12, 17]. Furthermore, parallelization of stateful logic computations within a memristive crossbar is discussed in [4, 12].

Using memristors for logic processing is an active area of research. Proposed methods that are closely related to the stateful logic considered in this chapter include for example the memristor programming scheme presented in [15, 19], where the memristors are directly driven with voltages representing logical values instead of using input memristors to store these values. Another approach closely related to stateful logic, called *memristor ratioed logic*, is presented in [7]. Briefly, the idea there is to use antiserially connected memristors to dynamically switch into a state that corresponds to a given logical function of the input voltages. In literature there exists also a variety of different logic computation approaches that use memristors as synaptic elements. Such approaches include the *wired-OR logic* in [21], and the *memristive threshold logic* in [18].

In Sect. 2 we present the different memristive stateful logic operations attained from a generalization of the circuit proposed in [5]. We describe various ways of synthesizing Boolean functions with memristive stateful logic in Sect. 3. Parallel stateful logic in memristive crossbars is considered in Sect. 4. We present concluding remarks and our views on the future development of memristive stateful logic in Sect. 5.

We define the memristor as a voltage controlled two-terminal bistable linear device having the on-resistance R_{ON} and the off-resistance R_{OFF}, where R_{OFF} is assumed to be much larger than R_{ON}. Correspondingly we say that a memristor is either in a conducting (1) or a nonconducting (0) state. A memristor is programmed to state 1 by applying across it a voltage larger than a threshold voltage V_{TH}; correspondingly applying a voltage more negative than $-V_{\mathrm{TH}}$ programs the memristor to state 0. We do not explicitly consider the duration of a programming operation, and assume that programming voltages are held

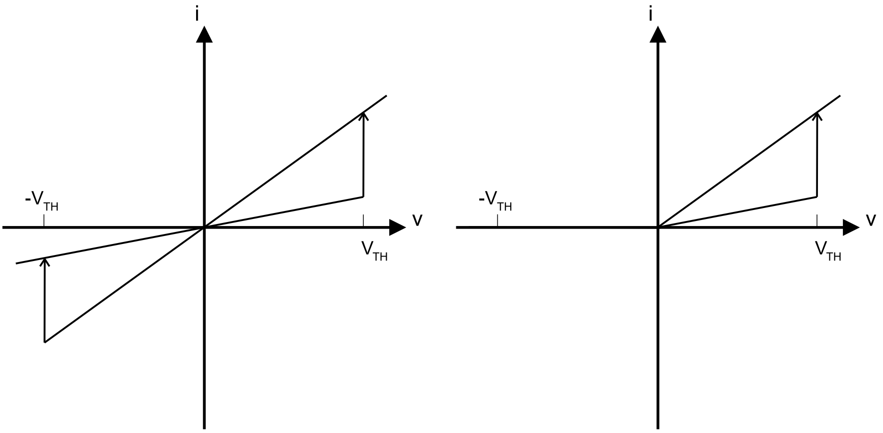


Fig. 1 Hysteresis curves of the memristor models used in this chapter. *Left*: Linear bipolar model. *Right*: Linear rectifying model

across memristors long enough to ensure full switching between states. When the voltage across a memristor is between $-V_{\mathrm{TH}}$ and V_{TH}, the state of the device is assumed to remain unchanged. In the left inset of Fig. 1 the hysteresis loop corresponding to this memristor model is depicted.

We also consider the use of rectifying memristors [3] for stateful logic. A simplified linear model of the rectifying memristor is depicted in the right inset of Fig. 1. This device is programmed to state 0 identically to the nonrectifying model, that is, by applying a voltage smaller than $-V_{\mathrm{TH}}$ across it, although negative current through the device is negligible. A physical explanation to why the rectification of current takes place at the same voltage polarity as switching to state 0 is given in [3]. In short, applying a small negative voltage partially retracts the ions forming the conducting channel within a memristor thus increasing its resistance substantially; this effect is undone by applying a small positive voltage across the device. The fact that the rectification takes place at the same voltage polarity as switching to nonconducting state affects the availability of some stateful logic operations, as is discussed in Sect. 2.1.

Note that the assumption of linear devices is made here to simplify analysis; effects of nonlinear I-V characteristics on the operation of memristive stateful logic are discussed for example in [1, 20].

2 Basic Memristive Stateful Logic Operations

Kuekes [5] proposed in 2008 that the circuit depicted in Fig. 2 would enable memristive stateful logic. The value of the series resistance R_0 is chosen between R_{ON} and

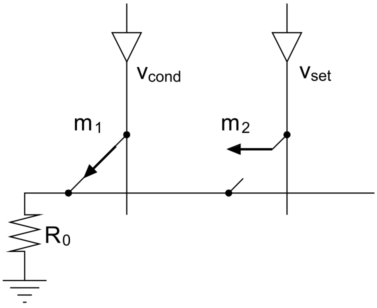


Fig. 2 Stateful memristor logic. The resulting state of m_2 depends on its previous state and the state of m_1. The state of m_1 remains unchanged during the operation

R_{OFF}; in the following we assume $R_0 = \sqrt{R_{\mathrm{ON}}R_{\mathrm{OFF}}}$, so that $R_0/R_{\mathrm{ON}} = R_{\mathrm{OFF}}/R_0$. The reason for using the series resistor R_0 is to realize a voltage division across the memristors and R_0 so that the voltage on the horizontal wire depends on the conductance states of the memristors. The optimal choice of R_0 depends for example on whether or not the considered memristors are linear or rectifying, and on the number of inputs per stateful operation; more detailed discussion about this topic can be found in [4, 6].

This stateful logic circuit is operated as follows. The memristor m_1 is driven with the conditional voltage v_{cond} whose magnitude is smaller than the switching threshold voltage of the memristor, $|v_{\mathrm{cond}}| < V_{\mathrm{TH}}$. The memristor m_2 is driven with a programming voltage v_{set}, whose value depends on the chosen stateful logic operation. Since the voltage across m_2 depends on the state of m_1 and on the voltage v_{cond}, it follows that the programming of m_2 is performed conditionally to the state of m_1. Furthermore, it is assumed that during the stateful logic operation the voltage across m_1 stays small enough to keep its state unchanged.

In the following subsection we generalize this concept of memristive stateful logic to multiple inputs and multiple outputs, and continue in the rest of the section to consider all possible combinations of polarities for the conditional and programming voltages, and the resulting logical operations.

2.1 Generalized Stateful Logic

Let us consider the memristive circuit depicted in Fig. 3. This circuit, used here for demonstrating generalized stateful logic, consists of two input memristors, m_1 and m_2, and two output memristors, m_4 and m_5; generally the number of allowed inputs and outputs can be larger and depends on the physical parameters of the circuit, for example on the resistance ratio $R_{\mathrm{OFF}}/R_{\mathrm{ON}}$. Memristor m_3 does not participate in the depicted operation, which is realized by setting its driver to high impedance state HZ. The stateful logic operation S performed by this circuit is determined by the polarities and magnitudes of the voltages v_{cond} and v_{set}. For circuit topologies of the form depicted in Fig. 3 we define the generalized stateful logic operation on input memristors $m_{i1}, \ldots, m_{ik}$ and output memristors $m_{o1}, \ldots, m_{ol}$ as one which fulfills the following conditions:

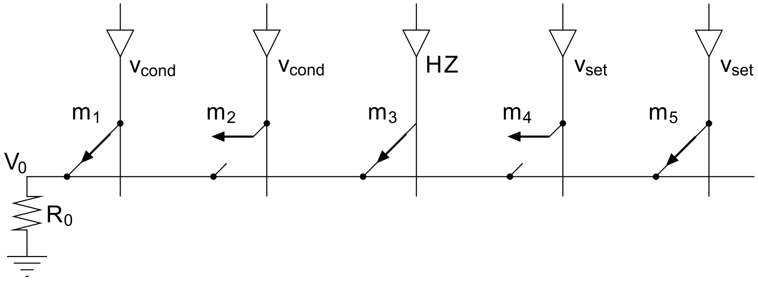


Fig. 3 A generalized stateful logic operation S yielding $m_4 = S(\mathrm{OR}(m_1, m_2), m_4)$ and $m_5 = S(\mathrm{OR}(m_1, m_2), m_5)$. Memristor m_3 does not participate into this stateful logic operation, and therefore its driver is set to a high impedance state. V_0 refers to the stabilized voltage on the *horizontal wire*, which depends on the states of the input memristors, here m_1 and m_2

1. The states of input memristors do not change during the operation.
2. Voltage V_0 on the horizontal wire is close to ground if all the input memristors are in state 0. Otherwise V_0 is close to v_{cond}. In particular, V_0 depends only on the states of the input memristors and not on the states of the output memristors.
3. The states of output memristors may change depending on their initial states and the voltage difference between v_{set} and V_0.

Assuming these conditions are fulfilled, the states of output memristors after the stateful logic operation S satisfy

$$m_{oj} := S\big(\mathrm{OR}(m_{i1}, \ldots, m_{ik}), m_{oj}\big) \quad \forall j = 1, \ldots, l. \tag{1}$$

Note that the maximum number of inputs depends on the resistance ratio $R_{\mathrm{OFF}}/R_{\mathrm{ON}}$, since the sum of currents through input memristors in state 0 should be significantly smaller than the current through a single input memristor in state 1.

In practice, fulfilling condition 2 above requires special attention. Care must be taken that the resistance ratios R_{OFF}/R_0 and R_0/R_{ON} are large enough so that the voltage on the horizontal wire essentially represents the OR-function of the states of the input memristors. It should also be noted that current flowing through the output memristors affects voltage V_0. The general solution to this problem is to measure the effect of the input currents on V_0, and then maintain the resulting voltage during the programming of the output memristors. This can be simply achieved by ensuring that there is sufficient capacitance on the horizontal wire, which is precharged by the currents flowing through the input memristors. In this case it is necessary that the programming time of the memristors is shorter than the RC time constant associated with the horizontal wire. Another solution is to apply a dedicated CMOS *keeper circuit* which maintains voltage V_0 during the programming of the output memristors. Keeper circuits are discussed in detail in Sect. 2.2.

Both of the outlined solutions to maintaining voltage V_0 have drawbacks. A substantial capacitance slows the operation of the circuit, while application of keeper circuits increases the CMOS area overhead. Moreover, relatively large currents may

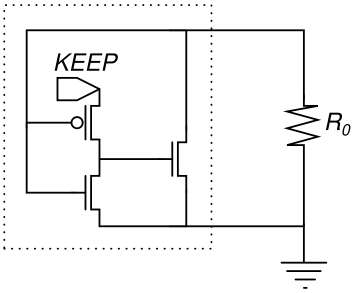


Fig. 4 A keeper circuit may be necessary to ensure the correct operation of stateful logic. This subcircuit should be used to replace the series resistance R_0 in Figs. 2 and 3

flow through the output memristors that are in state 1, which increases power consumption.

With the converse nonimplication operation described in Sect. 2.3.2, a third solution to maintaining voltage V_0 exists. Namely, by assuming rectifying memristors as described in Sect. 1, currents flowing through the output memristors can be suppressed. Since in this operation $v_{\text{cond}} > 0$ and $v_{\text{set}} < 0$, rectifying memristors allow current to be passed only through input memristors. Therefore converse nonimplication realized with rectifying memristors does not require a large capacitance on the horizontal wire, or additional CMOS keeper circuitry. However, a limitation with rectifying memristors is that the stateful AND and stateful OR operations specified in Sects. 2.3.3 and 2.3.4 are not directly available, since for these $v_{\text{cond}} < 0$. Moreover, multi-output operation is available only for converse nonimplication if no substantial capacitance exists on the horizontal wire and no keeper circuitry is used. A keeper circuit allowing the realization of all of the stateful logic operations with rectifying memristors is presented in the following subsection.

2.2 *Keeper Circuits*

As described above, auxiliary CMOS subcircuits can be used to maintain the voltage on the horizontal wire during the programming of the output memristors. These circuits are used to replace the series resistance R_0 in Fig. 3. When keeper circuits are used, the stateful logic operation is divided into two phases. In the first phase only the input memristors are driven, and the logical OR of their states is measured and stored in the keeper circuit. During the second phase the output memristors are driven with voltage v_{set}, and the keeper circuit is used to maintain the voltage on the horizontal wire at a value determined by the states of the input memristors. The use of keeper circuits brings memristive stateful logic closer to threshold logic described in [18]. The main difference is that here memristors are not only used as synapses, but also programmed conditionally according to the result of the threshold operation.

Originally presented in [8], the keeper circuit depicted in Fig. 4 is designed for implementing the material implication operation. In the first phase the input memristors are driven with v_{cond}, and the voltage across R_0 yields the logical OR of the

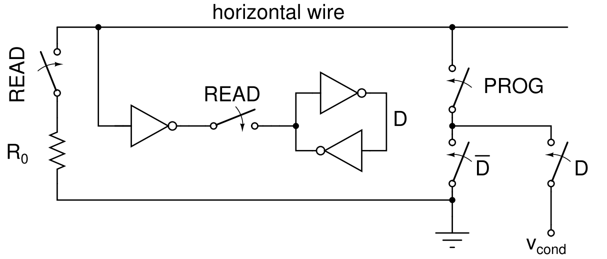


Fig. 5 A keeper circuit that enables all stateful logic operations also with rectifying memristors. In the first phase, the input memristors are driven with positive voltage $|v_{\text{cond}}|$ regardless of the chosen stateful logic operation, and in the second phase the *horizontal wire* is driven to either ground or to v_{cond} (which can be also negative) depending on the result obtained in the first phase

states of the input memristors. In the second phase the signal KEEP is used to activate the inverter circuit. If each input memristor is in state 0, the inverter drives the gate of the NMOS transistor high, providing a pull-down path to ground. Otherwise the voltage on the horizontal wire is at least v_{cond}, which in the case of the material implication operation prevents any programming of output memristors. Notice that when using this keeper circuit, the input memristors are driven in both phases.

The more complex keeper circuit depicted in Fig. 5 allows the realization of all four stateful logic operations described in Sect. 2.3. The main functional difference between this circuit and the one depicted in Fig. 4 is that here the horizontal wire is actively maintained either at ground or at v_{cond} depending on the states of the input memristors. In the first phase, the control signals READ and PROG are driven high and low, respectively, and the input memristors are driven with a positive voltage $|v_{\text{cond}}|$. The result of the OR of the input memristors is then stored in latch D. In the second phase the control signals READ and PROG are driven low and high, respectively, the output memristors are driven with v_{set}, and the drivers of the input memristors are set to high impedance. Depending on the result obtained in the first phase, the horizontal wire is driven either to ground or to v_{cond}, whose polarity depends on the chosen stateful logic operation. This keeper circuit can be used to enable all stateful logic operations with rectifying memristors, as the input memristors are driven always with a positive voltage.

Parallelization of stateful logic operations in a memristive crossbar topology may require specific keeper circuits to prevent the formation of undesired current paths. This topic is discussed in more detail in Sect. 4, where a keeper circuit for parallel stateful logic is depicted in Fig. 8.

2.3 *Stateful Logic Operations*

In the following we list the stateful logic operations resulting from different choices of the voltages v_{cond} and v_{set}.

Table 1 The truth tables of the material implication and converse nonimplication operations. Note that $p \rightarrow q \equiv \mathrm{OR}(\neg p, q)$ and $p \nleftarrow q \equiv \mathrm{AND}(\neg p, q)$

p	q	$p \rightarrow q$	$p \nleftarrow q$
0	0	1	0
0	1	1	1
1	0	0	0
1	1	1	0

2.3.1 Material Implication

Let the conditional voltage $0 < v_{\mathrm{cond}} < V_{\mathrm{TH}}$ and the programming voltage $v_{\mathrm{set}} > V_{\mathrm{TH}}$ be chosen such that

$$v_{\mathrm{set}} - v_{\mathrm{cond}} < V_{\mathrm{TH}}. \tag{2}$$

Then output memristor m_{oj} is programmed from state 0 to state 1 if and only if $\mathrm{OR}(m_{i1}, \ldots, m_{ik}) = 0$, resulting in

$$\begin{aligned} m_{oj} &:= \mathrm{OR}(m_{i1}, \ldots, m_{ik}) \rightarrow m_{oj} \\ &= \mathrm{OR}\big(\neg\mathrm{OR}(m_{i1}, \ldots, m_{ik}), m_{oj}\big) \\ &= \mathrm{NAND}\big(\mathrm{OR}(m_{i1}, \ldots, m_{ik}), \neg m_{oj}\big), \end{aligned} \tag{3}$$

where → denotes the logical operation called *material implication* whose truth table is presented in Table 1. In the case of a single input and a single output memristor, this is the stateful logic operation presented by Kuekes in [5].

2.3.2 Converse Nonimplication

Let $0 < v_{\mathrm{cond}} < V_{\mathrm{TH}}$, $-V_{\mathrm{TH}} < v_{\mathrm{set}} < 0$, and $v_{\mathrm{set}} - v_{\mathrm{cond}} < -V_{\mathrm{TH}}$. Output memristor m_{oj} is programmed from state 1 to state 0 if and only if $\mathrm{OR}(m_{i1}, \ldots, m_{ik}) = 1$. The truth table of this operation, called *converse nonimplication* and denoted by the symbol $\nleftarrow$, is also given in Table 1. The resulting state of m_{oj} equals

$$\begin{aligned} m_{oj} &:= \mathrm{OR}(m_{i1}, \ldots, m_{ik}) \nleftarrow m_{oj} \\ &= \mathrm{AND}\big(\neg\mathrm{OR}(m_{i1}, \ldots, m_{ik}), m_{oj}\big) \\ &= \mathrm{AND}(\neg m_{i1}, \ldots, \neg m_{ik}, m_{oj}). \end{aligned} \tag{4}$$

As noted in the previous subsection and originally proposed in [12], converse nonimplication with multiple inputs and multiple outputs can be realized without substantial capacitance on the horizontal wire or additional CMOS keeper circuitry, if rectifying memristors are used.

2.3.3 Stateful AND

Let the programming and conditional voltages both be negative,

$$-V_{\mathrm{TH}} < v_{\mathrm{cond}} < 0,$$
$$v_{\mathrm{set}} < -V_{\mathrm{TH}}, \quad \text{and} \quad v_{\mathrm{set}} - v_{\mathrm{cond}} > -V_{\mathrm{TH}}.$$

Now m_{oj} is programmed from state 1 to state 0 if and only if $\mathrm{OR}(m_{i1}, \dots, m_{ik}) = 0$ and thus the resulting state of m_{oj} is

$$m_{oj} := \mathrm{AND}\big(\mathrm{OR}(m_{i1}, \dots, m_{ik}), m_{oj}\big). \tag{5}$$

This stateful logic operation was first proposed for memristors in [4].

2.3.4 Stateful OR

The fourth possible choice of amplitudes and polarities of v_{cond} and v_{set} for a stateful logic operation satisfies $-V_{\mathrm{TH}} < v_{\mathrm{cond}} < 0$ and $0 < v_{\mathrm{set}} < V_{\mathrm{TH}}$ with the constraint $v_{\mathrm{set}} - v_{\mathrm{cond}} > V_{\mathrm{TH}}$. In this case m_{oj} is programmed from state 0 to state 1 if and only if $\mathrm{OR}(m_{i1}, \dots, m_{ik}) = 1$ and thus the resulting state of m_{oj} is

$$m_{oj} := \mathrm{OR}(m_{i1}, \dots, m_{ik}, m_{oj}). \tag{6}$$

2.4 Remarks

Changing the polarities, or rotating by 180 degrees, the memristors in Fig. 3 does not yield any new stateful logic operations. Indeed, the result of such a change is the same as if the voltage polarities of v_{cond} and v_{set} were reversed. For example, if the polarities of all memristors are reversed, the voltages specified above for material implication yield the stateful AND operation.

For the logical synthesis methods discussed in the following section it is useful to note that $\{\rightarrow, 0\}$ and $\{\nleftarrow, 1\}$ are functionally complete sets of Boolean operators, meaning that if memristors can be programmed to state 0 or 1 at will, both implication and converse nonimplication are alone sufficient to compute any Boolean function. This is not the case for stateful AND or stateful OR, which also require the negation operation to be functionally complete. Negation can be achieved for example by performing the material implication to 0, or converse nonimplication to 1, as $\neg p \equiv p \rightarrow 0$ and $\neg p \equiv p \nleftarrow 1$.

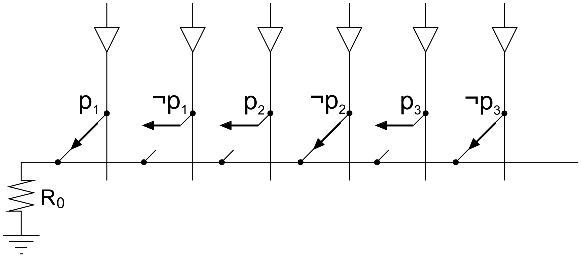


Fig. 6 Complementary representation of input variables. This figure is adapted from [11]

3 Synthesis of Boolean Functions

In this section we consider synthesis of Boolean functions using memristive stateful logic operations. By synthesis we mean determining a fixed sequence of stateful logic operations that yields the correct output for the target Boolean function for all values of input variables. In addition to the presented stateful logic operations, we assume that memristors can be programmed to state 0 or state 1 and that programming to a given state can be performed on any number of memristors in parallel.

3.1 Definitions

Let us consider the type of circuit topology depicted in Fig. 3. To synthesize a Boolean function $f : \{0, 1\}^N \to \{0, 1\}$, we divide the memristors in the circuit to three sets which we denote as *input memristors* p_i, *auxiliary memristors* a_i, and *result memristors* r_i. The states of the input memristors correspond to the values of the variables of the function and are not allowed to change during computation. The auxiliary memristors are used for the computation and for storing intermediate results. After the sequence of stateful operations used for synthesizing the function is performed, the result memristors should hold the value of the function as their states. Note that auxiliary and result memristors correspond to the output memristors discussed in the previous section. Either one or two result memristors are required per function, depending on whether or not a *complementary representation* of variables is used.

In complementary representation two memristors are reserved for each input and output variable: one which holds the value of the variable and another which holds its negation. We denote the memristors corresponding to the negated variables by $\neg p_i$, resulting in the input memristor set $\{p_1, \neg p_1, \ldots, p_N, \neg p_N\}$, as is illustrated in Fig. 6. The set of result memristors becomes $\{r_1, r_2\}$, and when computing Boolean function f, the result memristors will be programmed to $r_1 = f(p_1 \ldots, p_N)$ and $r_2 = \neg r_1$. Complementary representation allows for a direct way to synthesize a Boolean function based on logical normal forms, as will be seen in the following.

To simplify the notations in this chapter, the symbol p_i is used to denote an input memristor and its state, and a Boolean input variable and its value. Similarly a_i and r_i denote auxiliary and result memristors, and their states.

3.2 Synthesis Using the Conjunctive Normal Form

It is well known that any Boolean function $f : \{0, 1\}^N \rightarrow \{0, 1\}$ can be written in the *conjunctive normal form*

$$f \equiv \text{AND}(o_1, \ldots, o_m), \tag{7}$$

where each o_i is an OR-clause containing input variables or their negations.

The most straightforward way of synthesizing f is by using complementary representation and the stateful AND operation, by performing the following sequence of stateful operations:

$$\begin{aligned} &r_1, r_2 := 1 \\ &\quad r_1 := \text{AND}(o_1, r_1) \\ &\quad\quad \vdots \\ &\quad r_1 := \text{AND}(o_m, r_1) \\ &\quad r_2 := r_1 \nleftarrow r_2. \end{aligned}$$

In the first step, r_1 and r_2 are simultaneously programmed to conducting state. In the subsequent m steps, the stateful AND operation given in (5) is used to aggregate the OR-clauses o_i to r_1. Note that complementary representation is here necessary to ensure direct availability of the OR-clauses, which may contain also negated variables. The final step above is necessary to maintain complementary representation.

It is also possible to synthesize f based on the conjunctive normal form by using the stateful material implication operation (3). In this case, we write f in the form

$$f \equiv \text{NAND}(o_1, \ldots, o_h), \tag{8}$$

where o_i are again OR-clauses. Note however that these clauses are not the same as those contained in the conjunctive normal form (7) for f, as the form (8) is obtained from the conjunctive normal form of $\neg f$. Now f is synthesized as

$$\begin{aligned} &r_1, r_2 := 0 \\ &\quad r_1 := o_1 \rightarrow r_1 \\ &\quad\quad \vdots \\ &\quad r_1 := o_h \rightarrow r_1 \end{aligned}$$

$$r_2 := r_1 \to r_2.$$

In this case r_1 and r_2 are initialized to nonconducting state, the following h steps aggregate the OR-clauses to the NAND expression maintained in r_1, and finally $\neg r_1$ is stored to r_2.

Since there are 2^N values in the truth table of f, the conjunctive normal form of f or the conjunctive normal form of $\neg f$ contains at most 2^{N-1} OR-clauses. In complementary representation it is immaterial whether f or $\neg f$ is synthesized, since both are represented in the output memristors. Therefore with both of the two synthesis methods discussed above, computation of f requires at most $2^{N-1} + 2$ operations. Naturally, standard optimization methods for the conjunctive normal form can be used to shorten the sequence of stateful operations needed to synthesize a given function.

A third way of synthesizing f based on the conjunctive normal form uses the stateful converse nonimplication operation (4). In this case it is necessary to use auxiliary memristors a_i, or to perform several computation steps for each OR-clause in the conjunctive normal form, as is demonstrated in the following. When m auxiliary memristors are used, the computation of f—given again in the conjunctive normal form (7)—proceeds as follows:

$$\begin{aligned}
r_1, r_2, a_1, \ldots, a_m &:= 1 \\
a_1 &:= o_1 \nleftarrow a_1 \\
&\vdots \\
a_m &:= o_m \nleftarrow a_m \\
r_1 &:= \mathrm{OR}(a_1, \ldots, a_m) \nleftarrow r_1 \\
r_2 &:= r_1 \nleftarrow r_2.
\end{aligned}$$

Without auxiliary memristors the computational sequence is

$$\begin{aligned}
& r_1, r_2 := 1 \\
& \quad r_2 := o_1 \nleftarrow r_2, \qquad r_1 := r_2 \nleftarrow r_1, \qquad r_2 := 1 \\
& \quad \vdots \\
& \quad r_2 := o_m \nleftarrow r_2, \qquad r_1 := r_2 \nleftarrow r_1, \qquad r_2 := 1 \\
& \quad r_2 := r_1 \nleftarrow r_2.
\end{aligned}$$

As can be seen, in this case each OR-clause o_i requires two converse nonimplication operations and one unconditional programming of memristor r_2 to state 1.

3.3 Synthesis Without Complementary Representation of Variables

As described above, in complementary representation each Boolean variable requires two memristors. In practice such a redundant representation may be undesired, and therefore it is important to consider synthesis methods which use only a single memristor per Boolean variable. A trivial method is to use a set of auxiliary memristors, wherein the negated values of the input variables are stored before applying one of the above described synthesis methods. This strategy might be useful when the number of memristors per horizontal wire is large, but only a small subset of them is used in any given Boolean function.

Another solution, which uses only one auxiliary memristor and one result memristor, is obtained using the recursive form presented in [10]. This form contains only OR-clauses of positive, or non-negated, Boolean variables. More precisely, it can be shown that any Boolean function f can be written in the form

$$f \equiv \left(\sigma_1 \to \left(\sigma_2 \to \dots \left(\sigma_{m-1} \to \sigma_m^{\alpha_m}\right)^{\alpha_{m-1}} \dots\right)^{\alpha_2}\right)^{\alpha_1}, \tag{9}$$

where $\alpha_i \in \{0, 1\}$ for all $i = 1, \dots, m$, the notation $p^1 = p,\ p^0 = \neg p$ is assumed, and the terms σ_i are OR–clauses of positive input variables of the form

$$\sigma_i = \mathrm{OR}(p_{i1}, p_{i2}, \dots, p_{ik}).$$

The number of OR-clauses m in the above is at most 2^N, where N is the number of input variables of f. Details of synthesizing a given Boolean function using this recursive form and optimization methods for reducing m are presented in [17]. Notice that (9) can be realized using only stateful material implication operations, since negation is obtained as $p \to 0 \equiv \neg p$.

3.4 Remarks

Originally, synthesis methods for stateful logic were proposed only for single-input single-output material implication operation, for example in [1, 9]. These methods are inefficient and require large numbers of steps to synthesize a given function. For example, in [9] the three-input parity function was synthesized by a sequence of approximately 50 stateful logic and unconditional programming operations. In contrast to this, by using the conjunctive normal form method described above for material implication, the same function was synthesized using only five stateful logic operations in [11].

As an example of memristive stateful logic, let us consider a circuit similar to the one depicted in Fig. 3, with 9 memristors p_1, p_2, r_1, r_2, …, r_7. We assume that the result memristors r_i are all initially in state 0. Our goal is to perform a sequence of stateful logic steps after which the result memristors' states equal

$$r_1 = \neg p_1, \qquad r_2 = \neg p_2, \qquad r_3 = \mathrm{XOR}(p_1, p_2), \qquad r_4 = \neg r_3,$$

$$r_5 = \mathrm{AND}(p_1, p_2), \qquad r_6 = \neg r_5, \qquad r_7 = \mathrm{NOR}(p_1, p_2).$$

These functions are synthesized according to (8) by performing the following six material implication operations:

$$\begin{aligned} r_1 &:= p_1 \to r_1, \\ r_2 &:= p_2 \to r_2, \\ r_4, r_6, r_7 &:= \mathrm{OR}(p_1, p_2) \to r_4, r_6, r_7 \\ r_3, r_6 &:= \mathrm{OR}(p_1, r_2) \to r_3, r_6 \\ r_3, r_6 &:= \mathrm{OR}(p_2, r_1) \to r_3, r_6 \\ r_4, r_5 &:= \mathrm{OR}(r_1, r_2) \to r_4, r_5. \end{aligned}$$

In the sequence above, the first two steps simply copy the negations of p_1 and p_2 to r_1 and r_2, respectively, and the final four steps are generalized implication operations with multiple inputs and multiple outputs used to synthesize the functions related to r_i, where $i = 3, \ldots, 7$. Notice that in total seven functions are synthesized in six steps, as the synthesis is performed on multiple functions simultaneously.

4 Stateful Logic Within a Memristive Crossbar

A possible application of memristive stateful logic is to realize computation within memristive random-access memory. A possible circuit topology for such a memory is the memristive crossbar, which consists of perpendicular nanowires, where memristors are located at each crossing of the wires. In a *CMOL*-type memory architecture [14] the nanowires are driven by CMOS cells located beneath the crossbar. It has been proposed for example in [1, 4, 16]—and more thoroughly analyzed in the case of a crossbar consisting of rectifying memristors in [12]—that this topology allows also the realization of vectorized stateful logic operations when the CMOS cells are modified appropriately.

An example of a memristive crossbar is depicted in Fig. 7. A potential problem with such a crossbar structure is that undesired sneak current paths may form when the drivers of some rows or columns are set to high impedance state HZ. As mentioned in Sect. 2.1, this state is used to select which memristors do not participate in a given stateful operation. Since memristive stateful logic operations are based essentially on measuring the current flowing through the input memristors, the correct operation of stateful logic is disturbed by sneak current paths. In the following, memristors located in the crossings of a vertical wire and the horizontal wires are said to form a column. Likewise, a row of memristors refers to memristors located between a horizontal wire and the vertical wires in a crossbar.

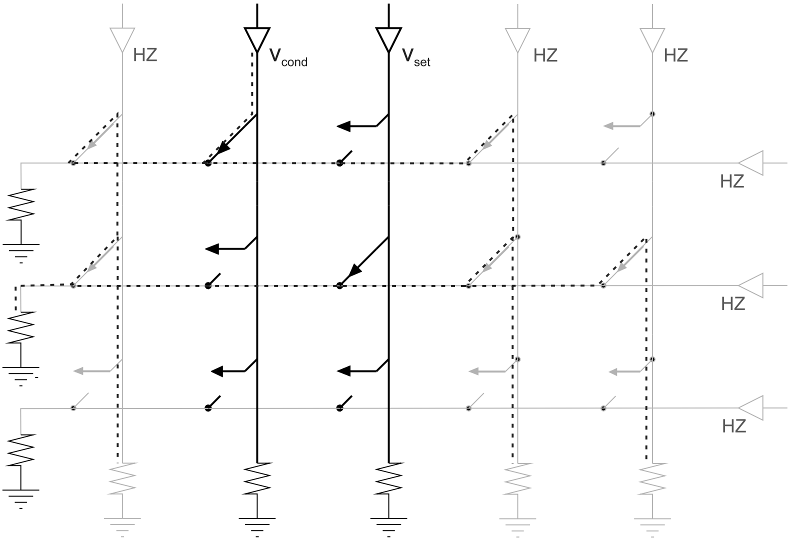


Fig. 7 Parallel stateful memristor logic within a crossbar. The depicted configuration performs a stateful logic operation in parallel on *all rows* over the *second* and *third memristors* from the *left*. This may cause an undesired current path, indicated here by a *dotted line*, through the wires whose drivers are set to high impedance state. Note that drivers of the *horizontal wires* must be set to high impedance state to allow stateful operations to be performed row-wise

4.1 Preventing Sneak Current Paths

There are several ways of preventing sneak current paths from forming in a memristive crossbar, such as driving all wires in the crossbar, that is, not allowing high impedance states in drivers [13], and the use of rectifying memristors [3]. In the following we consider these solutions and discuss how they can be used to facilitate stateful logic operations within a crossbar.

4.1.1 Non-rectifying Memristors

Let us consider a memristive crossbar consisting of non-rectifying memristors. We first concentrate on the realization of a column-wise stateful operation, that is, a bitwise vector operation between columns of memristors in this crossbar. For this, vertical wires participating in the stateful logic operation are driven with voltages v_{cond} and v_{set}. The goal is to have the stateful operation be performed in parallel on all the rows participating in this operation, and to leave the states of the memristors on non-participating rows unchanged.

To prevent sneak current paths from forming, an advanced keeper circuit such as the one depicted in Fig. 8 can be used. As described in Sect. 2.2, the stateful logic operation is divided into two phases. In the first phase, signals READ and

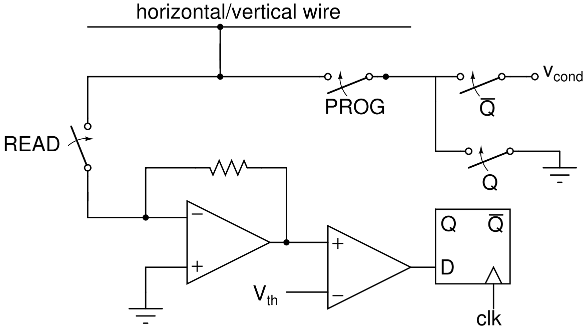


Fig. 8 An active keeper subcircuit that can be used for crossbars with non-rectifying memristors, replacing the series resistor R_0. To reduce CMOS area overhead, it may be beneficial to add switches to this circuitry so that a single keeper can be shared by a *horizontal* and a *vertical wire*

PROG are set high and low respectively. The currents flowing through the input memristors are measured by the inverting operational amplifier and the comparator, while the voltage on the horizontal wire is kept at virtual ground. The incoming currents are compared to a threshold voltage V_{th} in order to determine if any of the input memristors are in state 1. As with the keeper depicted in Fig. 5, the input memristors can be driven with a positive voltage $|v_{cond}|$, regardless of the chosen stateful logic operation. The result of the current measurement is written to the D-flip-flop. Since all the horizontal wires are kept at ground potential, the vertical wires not participating in this phase can also be driven to ground, thus preventing any sneak currents from forming.

In the second phase READ and PROG are set low and high respectively, and the horizontal wires participating in the stateful operation are driven either to ground or to v_{cond} depending on the result of the first phase. If a horizontal wire does not participate in this operation, it should be driven unconditionally either to ground or to v_{cond}, depending on the stateful logic operation, in order to prevent the memristors connected to it from being programmed. For simplicity, the logic required to achieve this is omitted from the circuit depicted in Fig. 8. The wires corresponding to the output memristor columns are driven with v_{set}.

Row-wise operations are performed similarly, the only difference being that the keeper circuits should be connected to vertical wires, while horizontal wires are driven to v_{cond} and v_{set}, or to ground in the case of non-participating wires. One keeper circuit is sufficient to control a pair of horizontal and vertical wires, since a stateful logic operation is performed either on a row or on a column of memristors. The main drawback of using the considered keeper circuit is the increased CMOS area overhead due to the required analog circuitry.

4.1.2 Rectifying Memristors

A more area-efficient solution to avoid undesired sneak current paths is to use rectifying memristors to allow currents to flow only in one direction in the memristive crossbar. As mentioned in Sect. 2.1, the drawback with this scheme is that only positive (negative) conditional voltages can be used for column-wise (row-wise) operations. This makes it impossible to realize the stateful AND and OR operations without using a keeper circuit such as the one depicted in Fig. 5. Notice that this keeper circuit is comprised of digital components, and therefore consumes significantly less CMOS area than the analog keeper circuit depicted in Fig. 8. However, as noted in Sect. 2.3.2, if only the converse nonimplication operation is used, then no keeper circuit is required.

Using rectifying memristors, column-wise operations can be performed as described in Sect. 2.1. As noted in [12], in row-wise operations it is necessary to reverse the polarities of the conditional and programming voltages, assuming that the ground voltage equals 0 V. When rectifying memristors are used, the input wires not participating in a stateful operation can be left floating, while the not participating output wires should be driven unconditionally either to v_{cond} or to ground, depending on the chosen stateful logic operation.

4.2 Example on Stateful Logic Within a Memristive Crossbar

In [12] it was demonstrated that when using rectifying memristors and generalized implication and converse nonimplication operations, any Boolean function $f : \{0, 1\}^N \to \{0, 1\}$ can be synthesized within a $2^N \times N$ memristive crossbar as a sequence of approximately $2N$ stateful operations. This is achieved by a computation that emulates the operation of a programmable logic array. In the following we show how the addition of two N-bit binary numbers can be performed using generalized stateful logic within a memristive crossbar.

Suppose that the binary numbers to be added together are represented by the states of the memristors on the first two rows of the crossbar, and that initially all other memristors are in state 0. For example, when adding together numbers 101 and 1110, the memristors in the crossbar should be initialized according to the first array in (10).

$$
\begin{array}{|c|c|}
\hline
p_1 & 0\,0\,1\,0\,1 \\ \hline
p_2 & 0\,1\,1\,1\,0 \\ \hline
r_1 & 0\,0\,0\,0\,0 \\ \hline
r_2 & 0\,0\,0\,0\,0 \\ \hline
r_3 & 0\,0\,0\,0\,0 \\ \hline
r_4 & 0\,0\,0\,0\,0 \\ \hline
r_5 & 0\,0\,0\,0\,0 \\ \hline
r_6 & 0\,0\,0\,0\,0 \\ \hline
r_7 & 0\,0\,0\,0\,0 \\ \hline
r_8 & 0\,0\,0\,0\,0 \\ \hline
\end{array}
\overset{6}{\mapsto}
\begin{array}{|l|c|}
\hline
p_1 & 0\,0\,1\,0\,1 \\ \hline
p_2 & 0\,1\,1\,1\,0 \\ \hline
r_1 = \neg p_1 & 1\,1\,0\,1\,0 \\ \hline
r_2 = \neg p_2 & 1\,0\,0\,0\,1 \\ \hline
r_3 = \mathrm{XOR}(p_1, p_2) & 0\,1\,0\,1\,1 \\ \hline
r_4 = \neg r_3 & 1\,0\,1\,0\,0 \\ \hline
r_5 = \mathrm{AND}(p_1, p_2) & 0\,0\,1\,0\,0 \\ \hline
r_6 = \neg r_5 & 1\,1\,0\,1\,1 \\ \hline
r_7 = \mathrm{NOR}(p_1, p_2) & 1\,0\,0\,0\,0 \\ \hline
r_8 & 0\,0\,0\,0\,0 \\ \hline
\end{array}
\overset{10}{\mapsto}
\begin{array}{|l|c|}
\hline
p_1 & 0\,0\,1\,0\,1 \\ \hline
p_2 & 0\,1\,1\,1\,0 \\ \hline
r_1 & 1\,1\,0\,1\,0 \\ \hline
r_2 & 1\,0\,0\,0\,1 \\ \hline
r_3 & 0\,1\,0\,1\,1 \\ \hline
r_4 & 1\,0\,1\,0\,0 \\ \hline
r_5 & 0\,0\,1\,0\,0 \\ \hline
r_6 & 1\,1\,0\,1\,1 \\ \hline
r_7 = \overline{c_i} & 1\,0\,0\,1\,1 \\ \hline
r_8 & 0\,0\,0\,0\,0 \\ \hline
\end{array}
\qquad (10)
$$

Let us then perform the six implication operations described in the example of Sect. 2.4 to the rows r_1 to r_7. As a result, the memristors' states should correspond to the second array in (10).

Next, the negation of the carry bits corresponding to the sum of rows p_1 and p_2 will be computed on the row r_7. For this we note that the negation of the ith carry bit $\overline{c_i}$ satisfies

$$\overline{c_i} = \overline{(p_1 \wedge p_2)_i} \wedge \left(\overline{c_{i-1}} \vee \overline{(p_1 \vee p_2)_i}\right), \tag{11}$$

where $\vee$ and $\wedge$ denote the logical OR and AND operations, respectively, and $\overline{x}$ denotes the negation of x. Since $c_0 = 0$, it follows that $\overline{c_1} = \overline{(p_1 \wedge p_2)}$. The state of the last memristor on row r_6 should be copied as the state of the last memristor on row r_7; this can be achieved by resetting the memristor on row r_7 and performing the stateful OR operation. In this example the first negated carry bit equals $\overline{c_1} = 1$.

Using (11), the second negated carry bit is then computed as a sequence of stateful logic operations. Assuming that the stateful OR is available, this is achieved simply by an OR operation between the last and second-to-last memristor of row r_7 followed by the converse nonimplication operation between the second-to-last memristors on rows r_5 and r_7. Continuing this process iteratively all the negated carry bits can be computed; the resulting state of the memristive crossbar used in this example is given in the third array of (10). In total 10 stateful operations are required to reach this state from the state represented in the second array of (10).

However, the carry bits must be shifted by one step to the left so that they can be used in the binary addition. This is why their negations were computed; to shift the carry bits one must perform a sequence of resetting the memristors' states to 0 and then performing the implication operation, which negates the states. For example, to shift the value of the second memristor on row r_7 one step to the left, one first resets the first memristor on row r_7 to state 0, and then performs the implication operation from the second memristor to the first one. This process is iterated for all the memristors on row r_7, and takes 9 operations to reach the configuration represented in the first array of (12).

The final steps of the computation consist of copying the negation of row r_7 to r_8 in a single material implication operation, of resetting the row r_1, and of computing the XOR of rows r_3 and r_7 using complementary representation. In the end, the sum of p_1 and p_2 is stored as the states of memristors on row r_1. These steps are depicted in (12).

$$\overset{9}{\mapsto}
\begin{array}{|l|l|}\hline
p_1 & 0\,0\,1\,0\,1 \\ \hline
p_2 & 0\,1\,1\,1\,0 \\ \hline
r_1 & 1\,1\,0\,1\,0 \\ \hline
r_2 & 1\,0\,0\,0\,1 \\ \hline
r_3 & 0\,1\,0\,1\,1 \\ \hline
r_4 & 1\,0\,1\,0\,0 \\ \hline
r_5 & 0\,0\,1\,0\,0 \\ \hline
r_6 & 1\,1\,0\,1\,1 \\ \hline
r_7 = c_{i-1} & 1\,1\,0\,0\,0 \\ \hline
r_8 & 0\,0\,0\,0\,0 \\ \hline
\end{array}
\overset{2}{\mapsto}
\begin{array}{|l|l|}\hline
p_1 & 0\,0\,1\,0\,1 \\ \hline
p_2 & 0\,1\,1\,1\,0 \\ \hline
r_1 & 0\,0\,0\,0\,0 \\ \hline
r_2 & 1\,0\,0\,0\,1 \\ \hline
r_3 & 0\,1\,0\,1\,1 \\ \hline
r_4 & 1\,0\,1\,0\,0 \\ \hline
r_5 & 0\,0\,1\,0\,0 \\ \hline
r_6 & 1\,1\,0\,1\,1 \\ \hline
r_7 = c_{i-1} & 1\,1\,0\,0\,0 \\ \hline
r_8 = \neg r_7 & 0\,0\,1\,1\,1 \\ \hline
\end{array}
\overset{2}{\mapsto}
\begin{array}{|l|l|}\hline
p_1 & 0\,0\,1\,0\,1 \\ \hline
p_2 & 0\,1\,1\,1\,0 \\ \hline
r_1 = p_1 + p_2 & 1\,0\,0\,1\,1 \\ \hline
r_2 & 1\,0\,0\,0\,1 \\ \hline
r_3 & 0\,1\,0\,1\,1 \\ \hline
r_4 & 1\,0\,1\,0\,0 \\ \hline
r_5 & 0\,0\,1\,0\,0 \\ \hline
r_6 & 1\,1\,0\,1\,1 \\ \hline
r_7 = c_{i-1} & 1\,1\,0\,0\,0 \\ \hline
r_8 = \neg r_7 & 0\,0\,1\,1\,1 \\ \hline
\end{array}
\tag{12}$$

5 Concluding Remarks

The previous example is representative of the advantages and disadvantages of memristive stateful logic. On one hand, bitwise vector operations are readily available and can be performed simultaneously on many rows or columns of memristors in the crossbar. On the other hand, the time complexity of a simple shift operation is linear in the number of memristors on a given row.

Many vector operations would become easier and faster to implement if the wires in the crossbar could be divided at will into independent segments, on which stateful operations could be performed simultaneously. For example, the left shift of an arbitrarily long vector could be performed as a sequence of eight converse nonimplication operations and a reset operation as follows. Suppose that part of the memristive crossbar is configured as in the first array of (13).

$$
\begin{gathered}
\begin{array}{|ccccccc|}
\hline
\cdots & p_1 & p_2 & p_3 & p_4 & p_5 & \cdots \\ \hline
\cdots & 1 & 1 & 1 & 1 & 1 & \cdots \\ \hline
\cdots & 1 & 1 & 1 & 1 & 1 & \cdots \\ \hline
\cdots & 1 & 1 & 1 & 1 & 1 & \cdots \\ \hline
\end{array}
\overset{2}{\mapsto}
\begin{array}{|ccccccc|}
\hline
\cdots & p_1 & p_2 & p_3 & p_4 & p_5 & \cdots \\ \hline
\cdots & \neg p_1 & 1 & \neg p_3 & 1 & \neg p_5 & \cdots \\ \hline
\cdots & 1 & \neg p_2 & 1 & \neg p_4 & 1 & \cdots \\ \hline
\cdots & 1 & 1 & 1 & 1 & 1 & \cdots \\ \hline
\end{array}
\\
\mapsto
\begin{array}{|cc|cc|cc|c|}
\hline
\cdots & p_1 & p_2 & p_3 & p_4 & p_5 & \cdots \\ \hline
\cdots & \neg p_1 & p_3 & \neg p_3 & p_5 & \neg p_5 & \cdots \\ \hline
\cdots & 1 & \neg p_2 & 1 & \neg p_4 & 1 & \cdots \\ \hline
\cdots & 1 & 1 & 1 & 1 & 1 & \cdots \\ \hline
\end{array}
\mapsto
\begin{array}{|c|cc|cc|cc|}
\hline
\cdots & p_1 & p_2 & p_3 & p_4 & p_5 & \cdots \\ \hline
\cdots & \neg p_1 & p_3 & \neg p_3 & p_5 & \neg p_5 & \cdots \\ \hline
\cdots & p_2 & \neg p_2 & p_4 & \neg p_4 & p_6 & \cdots \\ \hline
\cdots & 1 & 1 & 1 & 1 & 1 & \cdots \\ \hline
\end{array}
\\
\overset{2}{\mapsto}
\begin{array}{|ccccccc|}
\hline
\cdots & p_1 & p_2 & p_3 & p_4 & p_5 & \cdots \\ \hline
\cdots & \neg p_1 & p_3 & \neg p_3 & p_5 & \neg p_5 & \cdots \\ \hline
\cdots & p_2 & \neg p_2 & p_4 & \neg p_4 & p_6 & \cdots \\ \hline
\cdots & \neg p_2 & \neg p_3 & \neg p_4 & \neg p_5 & \neg p_6 & \cdots \\ \hline
\end{array}
\overset{2}{\mapsto}
\begin{array}{|ccccccc|}
\hline
\cdots & p_2 & p_3 & p_4 & p_5 & p_6 & \cdots \\ \hline
\cdots & \neg p_1 & p_3 & \neg p_3 & p_5 & \neg p_5 & \cdots \\ \hline
\cdots & p_2 & \neg p_2 & p_4 & \neg p_4 & p_6 & \cdots \\ \hline
\cdots & \neg p_2 & \neg p_3 & \neg p_4 & \neg p_5 & \neg p_6 & \cdots \\ \hline
\end{array}
\end{gathered}
\tag{13}
$$

First, two converse nonimplication operations are used to copy every second negated value on row 1 to rows 2 and 3. In the next two steps the array is vertically divided into segments of two memristors, and within each segment a converse nonimplication operation is performed to the left. This division is depicted in (13) by vertical lines. During this division it is important to be able to choose which rows participate in the converse nonimplication operation, so that the shift operation is only performed on desired rows.

After the operations requiring division of the array, two converse nonimplication operations are used to copy the negated and shifted variables onto row 4. Finally, all memristors on row 1 are set to state 1, and a converse nonimplication operation is performed from row 4 to row 1. As a result, row 1 has been shifted by one step to the left.

The above demonstrates the benefit of being able to divide the wires of a memristive crossbar into independent segments. In [4] such division is achieved by adding a CMOS switch between every crossing of horizontal and vertical wires, but such a solution unavoidably decreases the available density of the crossbar. A better solution in terms of density might be to use nanowire FETs such as those described in [2]. To preserve as much density as possible, no keeper circuits should be used for the individual segments. Instead, the crossbar should contain rectifying memristors, and converse nonimplication should be used as the stateful operation.

Acknowledgements This work was funded by the Academy of Finland (253596, 258831, 264914).

References

1. Borghetti, J., Snider, G.S., Kuekes, P.J., Yang, J.J., Stewart, D.R., Williams, R.S.: Memristive switches enable stateful logic operations via material implication. Nature **464**, 873–876 (2010)
2. Cui, Y., Zhong, Z., Wang, D., Wang, W.U., Lieber, C.M.: High performance silicon nanowire field effect transistors. Nano Lett. **3**(2), 149–152 (2003)
3. Kim, K.-H., Hyun Jo, S., Gaba, S., Lu, W.: Nanoscale resistive memory with intrinsic diode characteristics and long endurance. Appl. Phys. Lett. **96**(5), 053106 (2010)
4. Kim, K., Shin, S., Kang, S.-M.: Field programmable stateful logic array. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **30**(12), 1800–1813 (2011)
5. Kuekes, P.: Material implication: digital logic with memristors. In: A Presentation the Memristor and Memristive Systems Symposium at UC Berkeley (2008)
6. Kvatinsky, S., Kolodny, A., Weiser, U.C., Friedman, E.G.: Memristor-based imply logic design procedure. In: 2011 IEEE 29th International Conference on Computer Design (ICCD), pp. 142–147 (2011)
7. Kvatinsky, S., Wald, N., Satat, G., Kolodny, A., Weiser, U.C., Friedman, E.G.: MRL—memristor ratioed logic. In: 2012 13th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), pp. 1–6 (2012)
8. Laiho, M., Lehtonen, E.: Cellular nanoscale network cell with memristors for local implication logic and synapses. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2051–2054 (2010)
9. Lehtonen, E., Laiho, M.: Stateful implication logic with memristors. In: Proceedings of the 2009 IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2009, pp. 33–36 (2009)
10. Lehtonen, E., Poikonen, J.H., Laiho, M.: Two memristors suffice to compute all Boolean functions. Electron. Lett. **3**, 239 (2010)
11. Lehtonen, E., Poikonen, J.H., Laiho, M.: Implication logic synthesis methods for memristors In: Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS 2012 (2012)
12. Lehtonen, E., Poikonen, J.H., Laiho, M.: Applications and limitations of memristive implication logic. In: 2012 13th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), pp. 1–6 (2012)
13. Liang, J., Wong, H.-S.P.: Cross-point memory array without cell selectors—device characteristics and data storage pattern dependencies. IEEE Trans. Electron Devices **57**(10), 2531–2538 (2010)
14. Likharev, K.K., Strukov, D.B.: CMOL: devices, circuits, and architectures. In: Cuniberti, G., Fagas, G., Richter, K. (eds.) Introducing Molecular Electronics, pp. 447–478. Springer, Berlin (2005)
15. Linn, E., Rosezin, R., Tappertzhofen, S., Böttger, U., Waser, R.: Beyond von Neumann—logic operations in passive crossbar arrays alongside memory operations. Nanotechnology **23**(30) (2012)

16. Pershin, Y.V., Di Ventra, M.: Neuromorphic, digital, and quantum computation with memory circuit elements. Proc. IEEE **100**(6), 2071–2080 (2012)
17. Poikonen, J.H., Lehtonen, E., Laiho, M.: On synthesis of Boolean expressions for memristive devices using sequential implication logic. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. (2012)
18. Rose, G.S., Rajendran, J., Manem, H., Karri, R., Pino, R.E.: Leveraging memristive systems in the construction of digital logic circuits. Proc. IEEE **100**(6), 2033–2049 (2012)
19. Rosezin, R., Linn, E., Kugeler, C., Bruchhaus, R., Waser, R.: Crossbar logic using bipolar and complementary resistive switches. IEEE Electron Device Lett. **32**(6), 710–712 (2011)
20. Shin, S., Kim, K., Kang, S.-M.S.: Reconfigurable stateful NOR gate for large-scale logic-array integrations. IEEE Trans. Circuits Syst. II, Express Briefs **58**(7), 442–446 (2011)
21. Snider, G.S., Williams, R.S.: Nano/CMOS architectures using a field-programmable nanowire interconnect. Nanotechnology **18**(3) (2007)

Reaction-Diffusion Media with Excitable Oregonators Coupled by Memristors

Tetsuya Asai

Abstract This chapter presents dynamic behaviors of a new reaction-diffusion-type excitable medium, where the diffusion coefficient is represented by *memristive* dynamics. The medium consists of an array of excitable Oregonators, and each Oregonator is locally coupled with other Oregonators via *memristors*, which were claimed to be the fourth circuit element exhibiting a relationship between flux ϕ and charge q. By using the medium, this chapter exhibits that (i) the memristor conductances are modulated by the excitable waves and controlled the velocity of the waves, depending on the memristor's polarity, and (ii) nonuniform spatial patterns are generated depending on the initial condition of Oregonator's state, memristor polarity and stimulation.

1 Introduction

Semiconductor reaction-diffusion (RD) large-scale integrated circuits (LSIs) implementing RD dynamics have been proposed in the literature [1]. These LSIs, so-called reaction-diffusion chips, were mostly designed by digital, analog, or mixed-signal complementary-metal-oxide-semiconductor (CMOS) circuits of cellular neural networks (CNNs) or cellular automata (CA). Electrical cell circuits, which mimic *reactions* of chemical substances, were designed to implement several CA and CNN models of RD systems [2–6], as well as fundamental RD equations [7–10]. Each cell is arranged on a two-dimensional (2-D) square or a hexagonal grid and is connected to adjacent cells through coupling devices that mimic 2-D spatial *diffusion* of chemical substances, and indeed transmit the cell's state to its neighboring cells, as in conventional CAs. For instance, an analog-digital hybrid RD chip [3] was designed for emulating a conventional CA model for Belousov-Zhabotinsky (BZ) reactions [11]. A full-digital RD processor [4] was also designed on the basis of a multiple-valued CA model, called *excitable lattices* [12]. An analog cell circuit was also designed to be equivalent to spatial-discrete Turing RD systems [8]. A full-analog

T. Asai (✉)
Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan
e-mail: asai@ist.hokudai.ac.jp

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_28,

RD chip that emulates BZ reactions has also been designed and fabricated [7]. Furthermore, blueprints of non-CMOS RD chips have been designed, for example, a single-electron RD device [13]. The author and the colleagues previously proposed an RD device based on minority-carrier transport in semiconductor devices [14]. The point of the idea was to simulate chemical diffusion with minority-carrier diffusion in semiconductors and autocatalytic chemical reactions with carrier multiplication in *p–n–p–n* negative resistance diodes. Using CMOS and non-CMOS RD circuits enables us to simulate a variety of autocatalytic reactions and open up a variety of application fields for RD devices.

In this chapter, a new RD-based excitable medium is presented, keeping in mind its hardware implementation. Recently, the so-called "memristors," originally introduced by Leon Chua in 1971 [15] and claimed to be the fourth circuit element exhibiting a relationship between flux ϕ and charge q, have again been spotlighted since Strukov et al. presented equivalent physical examples [16]. Although the presented device was a bipolar resistive RAM that did not "directly" exhibit a relationship between ϕ and q, the device behaved as a non-volatile resistor whose resistance was continuously controlled by the amount of the charge flow (current). Here, the following question arises: "What happens if one replaces resistors for diffusion in analog RD LSIs with memristors?" This is the primary purpose of the investigation presented in this chapter. Through extensive numerical simulations, this chapter exhibits that (i) the memristor's conductances are modulated by excitable waves propagating on the memristor, depending on the memristor's polarity; (ii) velocity of the excitable wave propagation is thus modulated by the change of memristor conductance, and the degree of the modulation is inversely proportional to the time constant of the memristor's model, and (iii) nonuniform spatial patterns are generated depending on the initial condition of Oregonator's state, memristor polarity and stimulation.

In the following sections, an excitable RD model with memristors are introduced, and the spatiotemporal behaviors of 1-D and 2-D RD models are shown through extensive numerical simulations.

2 The Model

A general model of memristors is explained in terms of *memristance* $M(q)$ [15]. In this chapter, a simplified (and very comprehensive) memristor model is used. The dynamics are represented by

$$i = g(w)v, \qquad \frac{dw}{dt} = i \tag{1}$$

where v represents the voltage across the memristor; i, the current of the memristor; w, the nominal internal state of the memristor and corresponds to the charge flow of the memristor, and $g(w)$, the monotonically increasing function with increasing w [16]. This model implies that positive (or negative) i (current flow) increases (or

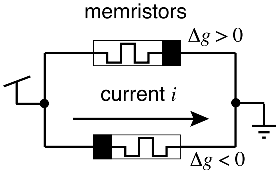


Fig. 1 Memristor symbols and polarity definition where Δg represents temporal differences of memristor's conductance

decreases) w, which results in an increase (or decrease) in the memristor conductance $g(w)$. Figure 1 illustrates these aspects of memristors, where Δg corresponds to dw/dt and hence $dg(w)/dt$. In the followings, we integrate these dynamics into a general RD model.

A general 1-D reaction-diffusion system is described by

$$\begin{aligned}\frac{\partial u(x)}{\partial t} &= g_u \nabla^2 u(x) + f_u\bigl[u(x), v(x)\bigr]\\ \frac{\partial v(x)}{\partial t} &= g_v \nabla^2 v(x) + f_v\bigl[u(x), v(x)\bigr]\end{aligned} \tag{2}$$

where $u(x)$ and $v(x)$ denote the concentrations of two different chemical species at spatial position x; $g_{u,v}$, the diffusion coefficients; and $f_{u,v}(\cdot)$, the reaction model. Here, we employ Oregonators [17] for the reaction model; i.e.,

$$\begin{aligned}f_u\bigl[u(x), v(x)\bigr] &= u(x)\bigl[1 - u(x)\bigr] - av(x)\frac{u(x) - b}{b + u(x)}\\ f_v\bigl[u(x), v(x)\bigr] &= u(x) - v(x)\end{aligned}$$

where a and b denote the reaction parameters. Depending on the reaction parameters, the Oregonator exhibits limit-cycle oscillations and excitatory behaviors. In this chapter, we consider the excitable properties only ($g_v = 0$ only), which means the Oregonator is stable as long as an external stimulus is not applied. In the model, three types of reaction states are defined at one Oregonator, namely, inactive, active, and refractory states. When the Oregonator is inactive, it is easily activated by an external stimulus, following which it changes to the refractory state. During the refractory state, the Oregonator cannot be activated even if an external stimulus is applied. Although g_u is constant in general RD models, one may be more interested in a system where g_u is locally modified by the potential gradient of $u(x)$.

When $u(x)$ and $v(x)$ are represented by voltages on the RD hardware, the gradient (diffusion terms in the RD model) is represented by linear resistors [7]. For example, if one discretizes Eq. (2) spatially as

$$\frac{du_i}{dt} = \frac{g_u \cdot (u_{i-1} - u_i) + g_u \cdot (u_{i+1} - u_i)}{\Delta x^2} + f_u(\cdot)$$

where i is the spatial index, and Δx the discrete step in space, terms $g_u \cdot (u_{i-1} - u_i)$ and $g_u \cdot (u_{i+1} - u_i)$ represent current flowing into the ith node from the $(i - 1)$th

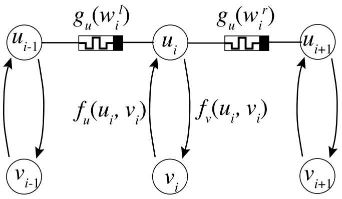


Fig. 2 Electrical representation of RD system whose diffusive resistors are replaced with memristors. Memristor's polarity is defined by coefficient η_1 in Eq. (4); e.g., polarity coefficient of memristor between u_{i-1} and u_i is set at $\eta_1 = 1$ in this figure

and $(i+1)$th nodes via two resistors whose conductance is represented by g_u. The spatial Laplacian ∇^2 in Eq. (2) can be approximated as

$$\nabla^2 u(x) = \frac{u_{i-1} + u_{i+1} - 2u_i}{\Delta x^2}.$$

Here, let us introduce the memristor model described by Eq. (1); in this model, the resistors are replaced with memristors. The resulting point dynamics are given as

$$\frac{du_i}{dt} = \frac{g_u(w_i^l)(u_{i-1} - u_i) + g_u(w_i^r)(u_{i+1} - u_i)}{\Delta x^2} + f_u(\cdot)$$
$$\frac{dv_i}{dt} = f_v(\cdot)$$

where $g_u(\cdot)$ denotes the monotonically increasing function defined by

$$g_u\left(w_i^{l,r}\right) = g_{\min} + (g_{\max} - g_{\min}) \cdot \frac{1}{1 + e^{-\beta w_i^{l,r}}} \tag{3}$$

where β denotes the gain; $g_{\min}$ and $g_{\max}$ denote the minimum and maximum coupling strengths, respectively, and $w_i^{l,r}$ denote the variables for determining the coupling strength of the ith Oregonator (l: leftward, r: rightward). Finally, the following memristive dynamics for $w_i^{l,r}$ are introduced as

$$\tau \frac{dw_i^{l,r}}{dt} = g_u\left(w_i^{l,r}\right) \cdot \eta_1 \cdot (u_{i-1,i+1} - u_i) \tag{4}$$

where the right-hand side represents the current of the memristors in Eq. (1), and η_1 denotes the polarity coefficient ($\eta_1 = +1 : w_i^l$, $\eta_1 = -1 : w_i^r$). Now one may notice that the model above corresponds to an electrical RD system consisting of Oregonators whose diffusive resistors are replaced with memristors (Fig. 2).

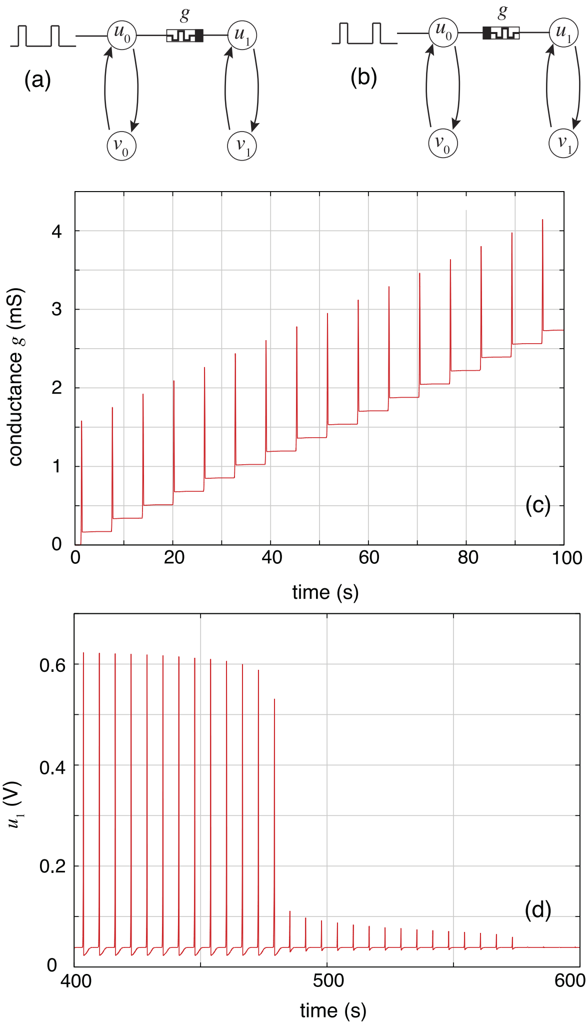


Fig. 3 Dynamics of memristor's conductances when two Oregonators are coupled by memristors with $\eta_1 = 1$ (**a**) and $\eta_1 = -1$ (**b**); (**c**) increased memristor's conductance as time with setups of (**a**) $\eta_1 = 1$; (**d**) dynamics of u_1 in (**b**) where u_1 is initially activated by u_0, however, since memristor's conductance is decreased as time with this setups (**b**) ($\eta_1 = -1$), u_1 becomes inactive as time

3 Dynamic Behaviors of Memristor RD Model

This section exhibits dynamic behaviors of memristor RD model. Memristor's common parameters in Eq. (3) and their values used in this section are: $\beta = 1$, $g_{\min} = 10^{-4}$, and $g_{\max} = 10^{-1}$.

3.1 1-D Reaction-Diffusion Medium with Memristors

First, let us see behaviors of a basic model shown in Fig. 3(a). One side of the boundary was stimulated by a periodic pulse sequence, and the conductance of the

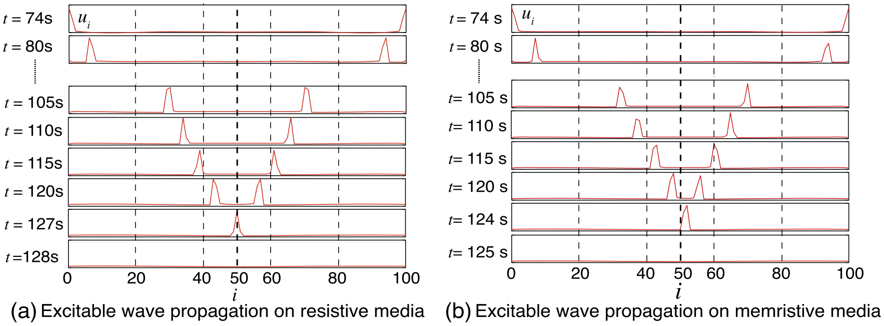


Fig. 4 Spatio-temporal behaviors of 1D RD model consisting of 100 Oregonators where each Oregonator was coupled by resistors (**a**) and memristors (**b**). Without memristive effects, excited waves emerged from both side collided at center (**a**), whereas waves collided at non center position when memristive couplings were introduced (**b**)

memristor was measured. The initial conductance of the memristor was set at zero. Figure 3(c) shows the simulated results. The conductance was increased considerably during the onset of the input pulse, which resulted in a small increase in the conductance. One may roughly estimate Δg per single pulse as 0.17 mS/pulse. Figure 3(b) shows the opposite simulation setup. In this simulation, the polarity of the memristor was inverted; therefore, one can expect the conductance to be decreased by the input pulses. The initial conductance was chosen such that stimulations to u_0 could cause chain excitation on u_1 via the memristor. Figure 3(d) shows the temporal responses of u_1. The stimulus was initially applied (u_1 was excited), but was terminated because of the decrease in the conductance. It should be noted that in both models of Figs. 3(a) and (b), the boundary condition is Neumann boundary condition.

Figure 4(a) shows the simulation results of a 1-D medium with 100 Oregonators without memristive effects. Excitable wave propagation on the medium is apparent. Both boundaries were simultaneously stimulated, and the waves collided at the center position (following which they disappeared). When the memristive effects were introduced, given that the coupling strength $g_u(w_i^{l,r})$ is modified by the direction of wave propagation, the results were different from those shown in Fig. 4(a). Figure 4(b) shows the simulation results of a 1-D medium consisting of 100 Oregonators with memristive effects, where the velocity of each excitable wave was different depending on the direction of wave propagation, which resulted in wave collision at a position other than the center (following which the waves disappeared). Excitable waves moving rightward (in the figure) increased $w_i^{l,r}$ of the memristors under the wave, whereas the leftward waves decreased $w_i^{l,r}$ under the wave, as a result of the polarity of memristors shown in Fig. 1.

A very interesting nonlinear behavior, emergence of stationary spiral patterns, has been found on the 1-D memristive medium. In the simulations, 100 Oregonators were connected by memristors with a cyclic boundary condition. All the memristors had the same polarity. After stimulating one node (the 58th node in simulation

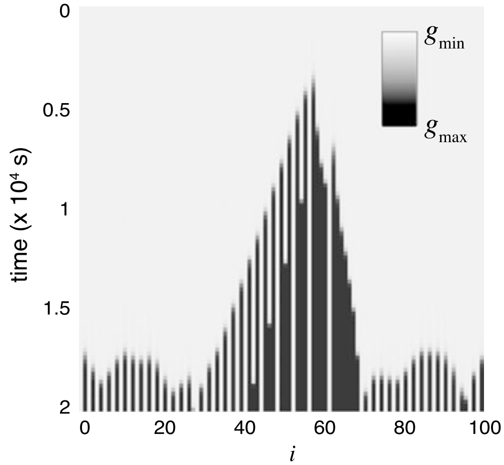


Fig. 5 Spatial pattern formation on 1-D excitable media with memristors under cyclic boundary condition

results shown in Fig. 5), an excitable wave propagated on the medium in a cyclic-looping manner. In the initial stimulation, wave propagation was controlled to be unidirectional by setting appropriate refractory states of the Oregonators. The initial conductance of the memristors was set at g_{min}. Figure 5 shows the time courses of all the nodes (u_i) where the magnitudes are represented by gray-scale tones. Spatial (nonuniform) patterns developed over time. Surprisingly, the developed patterns were periodic, like Turing patterns, and they reached equilibrium at around 2×10^4 s.

3.2 2-D Reaction-Diffusion Medium with Memristors

Now let us see dynamic behaviors of 2-D memristive medium. In the following simulations, a 2-D memristive medium with 100 × 100 Oregonators was introduced with a cyclic boundary condition.

For the 2-D RD medium, an additional memristive dynamics for $w_i^{up,d}$ for upward and downward connections are introduced:

$$\tau \frac{dw_i^{up,d}}{dt} = g_u\left(w_i^{up,d}\right) \cdot \eta_2 \cdot (u_{i-1,i+1} - u_i), \tag{5}$$

where $w_i^{up,d}$ denotes the variables for determining the coupling strength of the ith Oregonator (up: upward, d: downward); and η_2, the polarity coefficient ($\eta_2 = +1 : w_i^{up}$, $\eta_2 = -1 : w_i^d$). In the following simulations, one may assume both the polarity coefficients η_1 in Eq. (4) and η_2 to be -1.

The initial state of all the Oregonators was set to be inactive state. After stimulating the center node, the excitable waves propagated outwards, resulting in the generation of patterns of ocean surface waves. Figure 6 shows the time courses of all

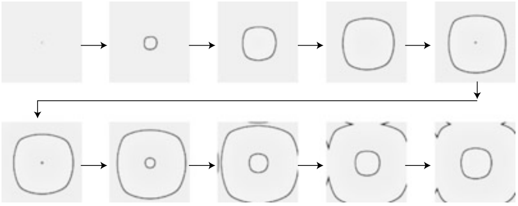


Fig. 6 Ocean-surface-wave patterns on 2-D media with memristors. Although the center was stimulated, wave propagated asymmetrically due to memristive effect with same polarity over the medium

the nodes $(u_{i,j})$ $(u_{i,j} = 0$: white, $u_{i,j} = 1$: black). The velocity of waves propagation was deferred depending on the direction of the wave propagation. According to the polarity of the memristors $(\eta_1, \eta_2 = -1)$, the conductance of the memristors shown on the right-hand side and below would increase whereas the conductance of those shown on the left-hand side or above would decrease; therefore, one can expect the waves to collide at the left-hand side and upward. Figure 7 shows the conductance of all the nodes, where the magnitudes are represented by gray-scale tones. These results were plotted when the conductance condition was stable given that the conductance changes considerably when the wave is propagating. Even after the waves propagated 10 times, the change in conductance was still small, after the waves propagated over that 20 times at the position where the waves generated at the beginning and collided, the memristor conductance changed considerably (Fig. 7(a)). Figure 7(b) shows the conductance of all the nodes over a sufficient time period.

Then, instead of extraneous stimulus, the initial stimulation was changed by controlling the states of the Oregonators. In Fig. 8, the values of $u_{i,j}$ are represented on a gray scale $(u_{i,j} = 0$: white, $u_{i,j} = 1$: black). Several Oregonators next to the inactive Oregonators were initially set in a refractory state (down side of the black bar in the top-left snapshot in Fig. 8). The inactive Oregonators next to the black bar were suppressed by the adjacent Oregonators in the refractory state (Oregonators in black bar). When the inactive Oregonators were in an active or inactive state, the wave rotated inwards, which resulted in the generation of clockwise and counterclockwise spiral patterns. Depending on the direction of wave propagation, the velocity of the rightward and downward waves was faster than that of the leftward and upward waves, given that η_1 in Eq. (4) and $\eta_2 = -1$ in Eq. (5) are both -1. Over time, the initial position of the generated waves, is moved to the lower right. Figure 9 shows the conductance of all the nodes, where the magnitudes are represented by gray-scale tones. It should be noted that these results were plotted when then conductance of some memristors was still unstable. Figure 9(b) shows the conductance of all the nodes over a sufficient time period, indicating global history of non-random movements of wave cores.

4 Conclusion

A new reaction-diffusion-based excitable medium that employed *memristors* to represent diffusion coupling was introduced. Through extensive numerical simulations, the following things were demonstrated:

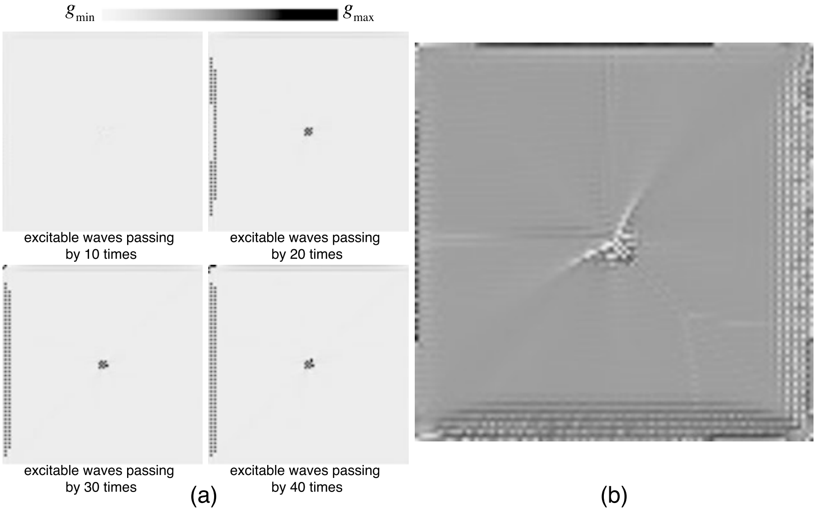
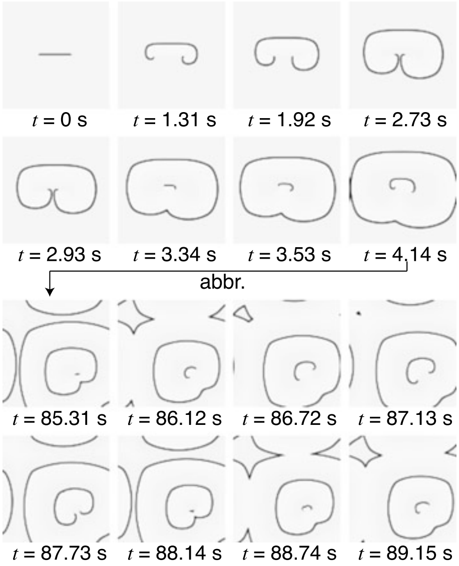


Fig. 7 Conductance map of memristors for 2-D media for ocean-surface-wave patterns; (**a**) early and (**b**) late maps

Fig. 8 Clockwise and counterclockwise spiral patterns on 2-D media with memristors



- The memristor's conductances are modulated by excitable waves propagating on the memristor, depending on the memristor's polarity.
- Velocity of the excitable wave propagation is thus modulated by the change of memristor conductance, and the degree of the modulation is inversely proportional to the time constant of the memristor's model.

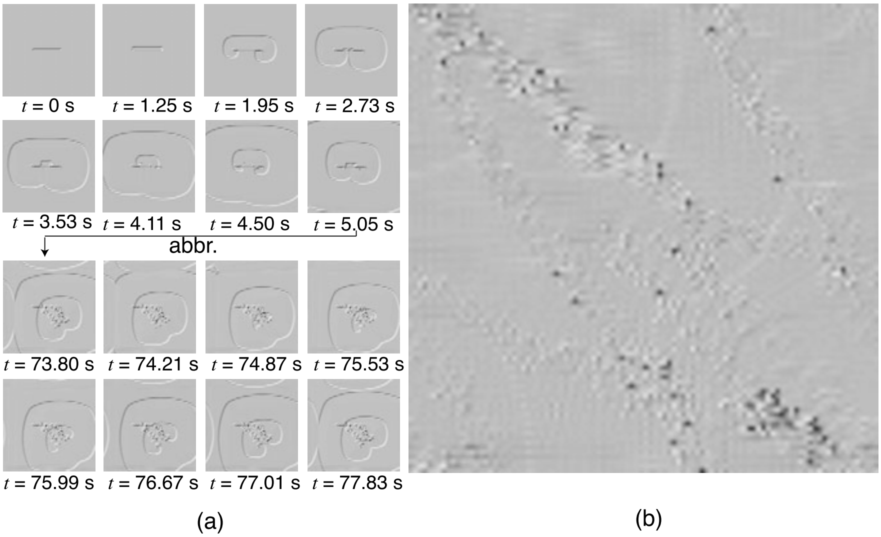


Fig. 9 Conductance maps of memristors for 2-D media for clockwise and counterclockwise spiral patterns; (**a**) early, and (**b**) late maps

- Nonuniform spatial patterns are generated depending on the initial condition of Oregonator's state, memristor polarity and stimulation.

Among the demonstrated behaviors, the nonuniform spatial-pattern generation would be applied to investigation on detecting global motion of excitable waves because the memristors accumulated the directions of waves on the media, which resulted in detecting majority of wave directions at every spatial point. Therefore if one has proper (non-memristive) RD media and a 2-D array of memristors without any reaction circuit, and the point dynamics of the RD media are given to the memristor array, one may detect the global motions of the RD media. This application is not limited in the analysis of RD systems, but the idea can be transferred to analyzing much more complex systems like brain networks, social networks, and so on.

Acknowledgements This study was supported by a Grant-in-Aid for Scientific Research on Innovative Areas [20111004] from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan.

References

1. Adamatzky, A., De Lacy Costello, B., Asai, T.: Reaction-Diffusion Computers. Elsevier, London (2005)
2. Adamatzky, A., Arena, P., Basile, A., Carmona-Galán, R., De Lacy Costello, B., Fortuna, L., Frasca, M., Rodríguez-Vázquez, A.: Reaction-diffusion navigation robot control: from chemical to VLSI analogic processors. IEEE Trans. Circuits Syst. I **51**(5), 926–938 (2004)

3. Asai, T., Nishimiya, Y., Amemiya, Y.: A CMOS reaction-diffusion circuit based on cellular-automaton processing emulating the Belousov-Zhabotinsky reaction. IEICE Trans. Fundam. E **85-A**(9), 2093–2096 (2002)
4. Matsubara, Y., Asai, T., Hirose, T., Amemiya, Y.: Reaction-diffusion chip implementing excitable lattices with multiple-valued cellular automata. IEICE Electron. Express **1**(9), 248–252 (2004)
5. Rekeczky, C., Roska, T., Carmona, R., Jiménez-Garrido, F., Rodríguez-Vázquez, A.: Exploration of spatial-temporal dynamic phenomena in a 32 × 32-cell stored program two-layer CNN universal machine chip prototype. J. Circuits Syst. Comput. **12**(6), 691–710 (2003)
6. Shi, B.E., Luo, B.T.: Spatial pattern formation via reaction-diffusion dynamics in 32 × 32 × 4 CNN chip. IEEE Trans. Circuits Syst. I **51**(5), 939–947 (2004)
7. Asai, T., Kanazawa, Y., Hirose, T., Amemiya, Y.: Analog reaction-diffusion chip imitating the Belousov-Zhabotinsky reaction with hardware Oregonator model. Int. J. Unconv. Comput. **1**(2), 123–147 (2005)
8. Daikoku, T., Asai, T., Amemiya, Y.: An analog CMOS circuit implementing Turing's reaction-diffusion model. In: Proc. Int. Symp. Nonlinear Theory and Its Applications, pp. 809–812 (2002)
9. Karahaliloglu, K., Balkir, S.: Bio-inspired compact cell circuit for reaction-diffusion systems. IEEE Trans. Circuits Syst. II **52**(9), 558–562 (2005)
10. Serrano-Gotarredona, T., Linares-Barranco, B.: Log-domain implementation of complex dynamics reaction-diffusion neural networks. IEEE Trans. Neural Netw. **14**(5), 1337–1355 (2003)
11. Gerhardt, M., Schuster, H., Tyson, J.J.: A cellular automaton model of excitable media. Physica D **46**, 392–415 (1990)
12. Adamatzky, A.: Computing in Nonlinear Media and Automata Collectives. IoP Publishing, Bristol (2001)
13. Oya, T., Asai, T., Fukui, T., Amemiya, Y.: Reaction-diffusion systems consisting of single-electron circuits. Int. J. Unconv. Comput. **1**(2), 177–194 (2005)
14. Asai, T., Adamatzky, A., Amemiya, Y.: Towards reaction-diffusion computing devices based on minority-carrier transport in semiconductors. Chaos Solitons Fractals **20**(4), 863–876 (2004)
15. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**(5), 507–519 (1971)
16. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **452**(1), 80–83 (2008)
17. http://www.scholarpedia.org/article/Oregonator

Autowaves in a Lattice of Memristor-Based Cells

Viet-Thanh Pham, Arturo Buscarino, Mattia Frasca, and Luigi Fortuna

Abstract In this Chapter, a Cellular Neural/Nonlinear Network (CNN) made of memristor-based cells is introduced. The Memristive CNN consists of identical cells, each containing a memristor, an inductor and a capacitor. We show how the Memristive CNN is able to generate autowaves. We investigate then an FPGA-based implementation of it and related experimental results confirming the capabilities of the system to generate autowaves. This shows that memristor can be used not only as a fundamental block of new chaotic circuits, but also to build complex systems made of interacting memristor-based elementary cells. In such systems many complex phenomena may take place like the autowave propagation discussed in this Chapter and thus feasible hardware emulators allowing an experimental investigation of systems based on electrical analogues of memristor devices are required. In this Chapter such a platform has been realized by using an FPGA-based approach which also has the further advantage of being flexible and easily adaptable to the study of other memristor-based complex systems.

1 Introduction

The discovery of a nanoscale memristor [1, 2] in 2008, that is, a device that was only theoretically postulated in 1971 [3], paved the way to an almost totally new research field, focused to the search of components whose behavior can be well represented by memristors or memristive systems, to the design of new devices with memristive characteristics and to the investigation of possible applications, ranging from biological models [4] to adaptive filters [5] and programmable analog integrated circuits [6, 7]. The potential applications of memristive systems lead to one series of questions, to which this book tries to give an answer. How complex systems made

V.-T. Pham · A. Buscarino · M. Frasca (✉) · L. Fortuna
Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania, 95125 Catania, Italy
e-mail: mfrasca@diees.unict.it

Present address:
V.-T. Pham
Hanoi University of Science and Technology, Hanoi, Viet Nam

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_29,

by interacting memristor-based circuits can be constructed? What kind of complex behavior do exhibit networks made of memristors? What can be the added value of using this component in a network of many interacting units?

In this Chapter, we discuss a specific aspect of such questions, that is, the possibility of observing autowave propagation phenomena in a lattice of memristive cells. To do this, we use the formalism of Cellular Neural/Nonlinear Networks (CNNs), and we study *memristive CNNs* (*MCNNs*), namely a CNN made of cells including memristors. We show that in fact autowaves can be generated with the proposed model. We discuss numerical results and experimental ones, based on an implementation of the model with FPGA.

The choice of using the CNN architecture as underlying formalism is motivated by the generality of the CNN approach, which has been successfully demonstrated as a paradigm for complexity, from image [8] and real-time signal processing [9] to implementation of chaotic circuits [10, 11], emulation of partial differential equations, and investigation of complex phenomena in the space such as autowaves [12, 13], spiral waves [14] or Turing patterns [15].

The classical CNN configuration includes a number of cells which consist of linear capacitors, linear resistors, linear and nonlinear voltage-controlled current sources and independent sources. In this Chapter we introduce the fourth element in CNN cells and we refer to this architecture as MCNN. In previous works [16], a cellular automaton and a discrete-time CNN (DTCNN) using nonlinear passive memristors were designed: one noticeable feature of memristive DTCNN was the multitasking, since memristive DTCNN were shown to be able to perform more than one functions of the memristor cellular automaton at the same time. Another work combining memristors and CNNs refers to the use of standard CNNs to implement memristive analogue circuits which then can be utilized as basic cells to realize chaotic circuits [17]. In another work [18], the role of memristors in implementing programmable connections of the cells was investigated.

The MCNN, discussed in this Chapter, follows a process of generalization which starts from the invention/discovery of the memristor leading to a new paradigm in which four components are at the basis of any electrical circuit, continues with the use of this novel component to design new dynamical, eventually chaotic, circuits and then goes in the direction of using these novel dynamical circuits based on memristors to define new CNN architectures with general features.

This Chapter is organized as follows. We present the model of a continuous-time CNN with memristive cells in Sect. 2. Section 3 illustrates through simulation results that the MCNN generates waves satisfying the classical properties of autowaves. Section 4 discusses the FPGA based implementation and experimental results. Conclusions are given in the last Section.

2 MCNN Model

Although, at the beginning, CNNs were made of first-order cells [19], their generalization to arrays of more complex cells (the cell itself can be a complex circuit, e.g.,

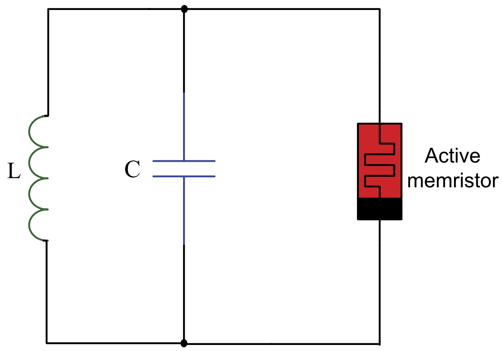


Fig. 1 The memristive cell

a Chua's circuit) allowed difficult computational problems to be reformulated naturally by CNN. These CNN arrays have been examined for the generation of Turing patterns and various autowaves [20], where CNNs were used to approximate partial differential equations, especially reaction-diffusion equations. In our case, we adopt this formalism to explore complexity through memristive CNN. Therefore, we first describe the memristive cell used and then the overall configuration of the MCNN investigated.

In their work, Itoh and Chua introduced a gallery of different memristive oscillators [21]. Some of these oscillators are also able to display chaotic dynamics. Among the oscillators introduced in [21], to construct the basic cell used in our investigation, we choose a simple one, the so-called memristor-based Chua oscillator with a flux controlled memristor. The circuit of the cell is illustrated in Fig. 1, where it can be observed that the cell includes three elements: an inductor, a capacitor, and an active memristor. The active memristor consists of a negative conductance ($-G$) and a passive memristor in parallel.

The dynamical equations of the cell are derived by applying Kirchhoff's circuit laws:

$$\begin{cases} C\dot{v} = -i - W(\varphi)v + Gv, \\ L\dot{i} = v, \\ \dot{\varphi} = v, \end{cases} \tag{1}$$

where v, i, and φ are voltage of the capacitor, current over inductor, and flux, respectively. The memristor is characterized by the memductance

$$W(\varphi) = \frac{dq(\varphi)}{d\varphi} = \begin{cases} a & |\varphi| < 1, \\ b & |\varphi| > 1, \end{cases} \tag{2}$$

where $q(\varphi)$ is a piecewise-linear function described by

$$q(\varphi) = b\varphi + 0.5(a-b)\big(|\varphi+1| - |\varphi-1|\big). \tag{3}$$

By introducing the new variables $x = v$, $y = i$, $z = \varphi$ and parameters $\alpha = 1/C$, $\beta = 1/L$, and $\gamma = G$, Eqs. (1) can be transformed into the following dimensionless

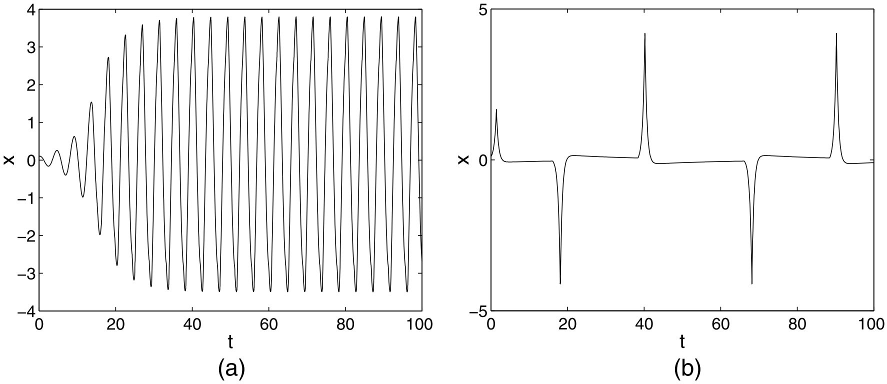


Fig. 2 Waveform of signal x when the parameters are chosen as (**a**) $\alpha = 2$, $\gamma = 0.3$, $\beta = 1$, $a = 0.1$, and $b = 0.5$, (**b**) $\alpha = 10$, $\gamma = 0.3$, $\beta = 0.01$, $a = 0.1$, and $b = 0.5$

equations:

$$\begin{cases} \dot{x} = \alpha(-y - W(z)x + \gamma x), \\ \dot{y} = \beta x, \\ \dot{z} = x, \end{cases} \tag{4}$$

When the parameters are chosen such as $\alpha = 2$, $\gamma = 0.3$, $\beta = 1$, $a = 0.1$, and $b = 0.5$, Eqs. (4) exhibit the periodical signal shown in Fig. 2(a). However, in order to get the autowaves, Eqs. (4) should be characterized by a slow-fast dynamics [14], [22]. In the slow regime, the state of the limit cycle remains at a constant value for a considerably long period of time τ_{st}. After this long period, the state returns rapidly in a significantly short period of time τ_{ex}, where $\tau_{ex} \ll \tau_{st}$. By choosing appropriate parameters e.g., $\alpha = 10$, $\gamma = 0.3$, $\beta = 0.01$, $a = 0.1$, and $b = 0.5$, our memristive cell satisfies this requirement. The waveform of the signal (variable $x(t)$) is shown in Fig. 2(b), in which the slow-fast dynamics is clearly evident.

Starting from the basic cell in Eqs. (1), the MCNN is constructed as shown in Fig. 3, by connecting each cell to four identical cells, called neighbours, through four linear resistors. As the result, a reaction-diffusion process is emulated.

From Fig. 3 the dynamical (dimensionless) equations of the MCNN are derived:

$$\begin{cases} \dot{x}_{i,j} = \alpha(-y_{i,j} - W(z_{i,j})x_{i,j} + \gamma x_{i,j}) \\ \qquad\quad + D(x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1} - 4x_{i,j}), \\ \dot{y}_{i,j} = \beta x_{i,j}, \\ \dot{z}_{i,j} = x_{i,j}, \end{cases} \tag{5}$$

where the diffusion coefficient D is constant.

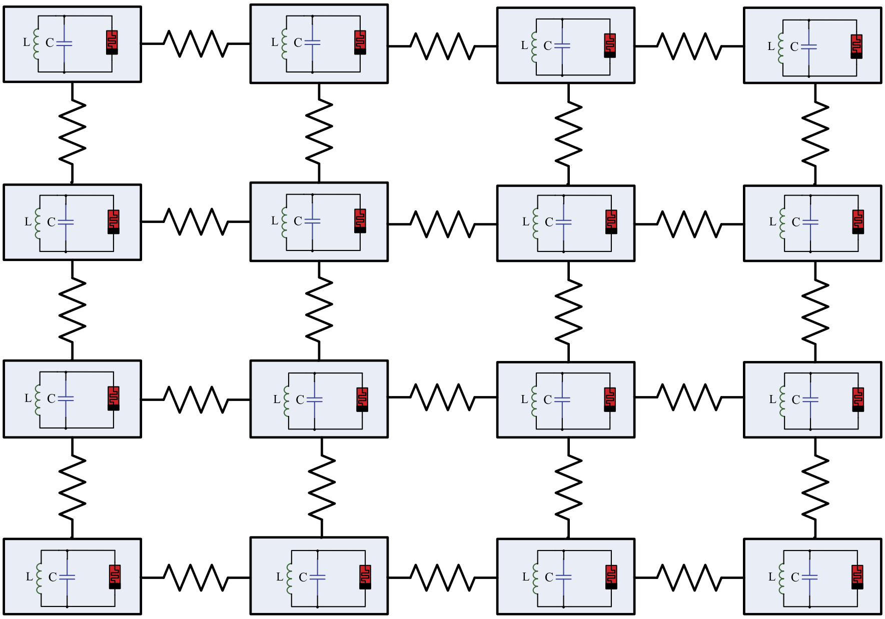


Fig. 3 Schematic illustration of the MCNN

3 Simulation Results

According to [12], the autowaves have some noticeable properties: the amplitude and shape of the autowave do not change during propagation; autowaves do not exhibit reflection or interference; annihilation occurs when two waves collides. In this Section, simulation results are reported to illustrate the principal features of autowaves in the MCNN.

A 50 × 50 MCNN with zero-flux boundary conditions has been simulated. Here, the diffusion coefficient was fixed to $D = 0.51$. We used the following initial conditions: $x_{i,j}(0) = y_{i,j}(0) = z_{i,j}(0) = 0$, where $1 \leq i, j \leq 50$, except $x_{i,2}(0) = 1.5$ in Fig. 4, $x_{25,25}(0) = 1.5$ in Fig. 5, and $x_{i,2}(0) = x_{i,49}(0) = 1.5$ in Fig. 6.

An autowave has been observed moving with the same shape, from the left-hand side to the right-hand side of the MCNN, as shown in Fig. 4. Unlike classical waves whose amplitude attenuates rapidly with the distance, the shape of autowave remains unchanged. In Fig. 5 an example of an autowave propagating from the centre of the lattice is shown.

Figure 6 represents two autowaves propagating from the left-hand side and the right-hand side as an effect of the initial conditions considered. The two waves move with the same velocity, finally annihilate each other when the two wavefronts collide. The major characteristics of autowaves were therefore observed in the MCNN, thus showing the effectiveness of the introduced structure to generate autowaves. It is worth to notice that autowaves are generated with a very simple basic cell con-

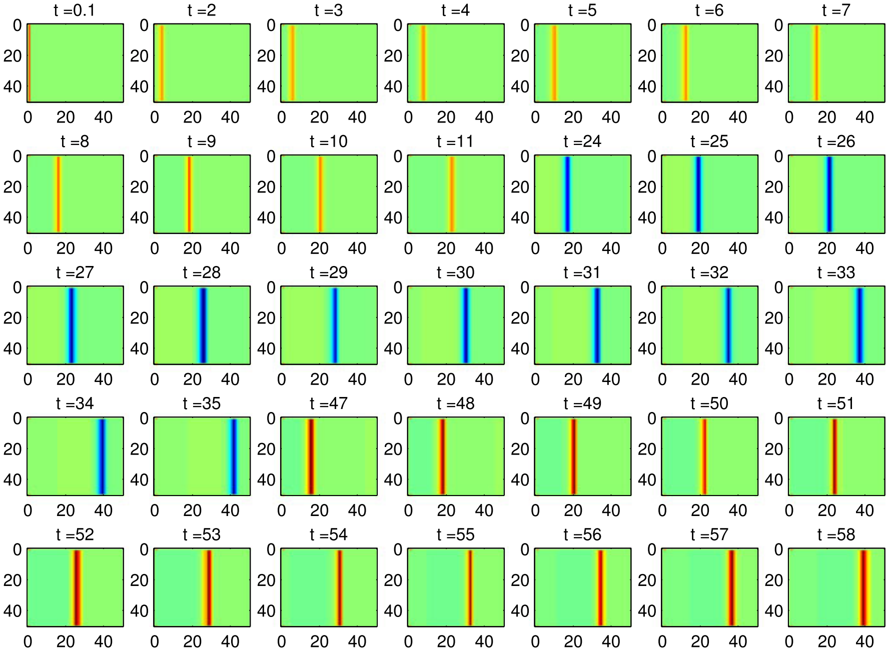
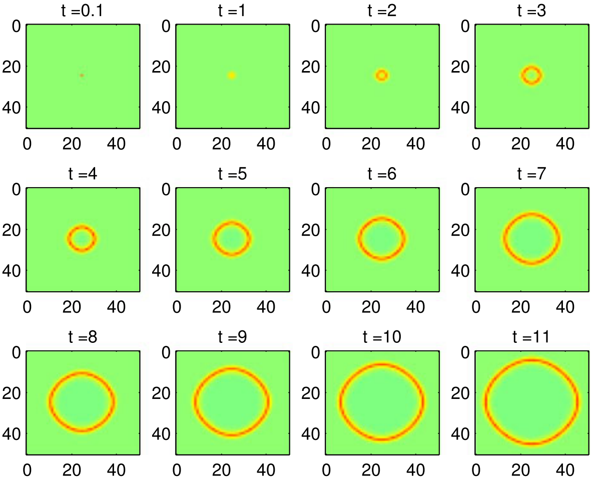


Fig. 4 An autowave propagating from the *left-hand side* to the *right-hand side* of a MCNN

Fig. 5 Formation of an autowave from the *centre* of the MCNN



sisting of three components and, in view of the considerations in [23], in principle, even a simpler cell (consisting of either a capacitor or an inductor and an active memristor) can be used.

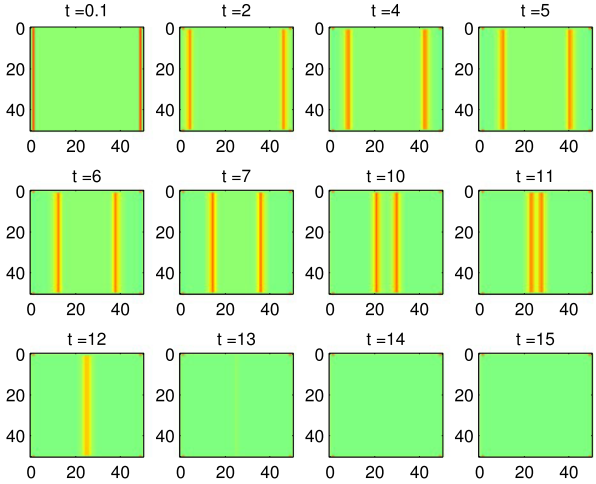


Fig. 6 Annihilation of two colliding autowaves, travelling from the *left-hand side* and the *right-hand side*

4 FPGA-Based Implementation of the MCNN

In this Section, we discuss how to implement the MCNN by using Field Programmable Gate Arrays (FPGA). Our system consists of a FPGA development board connected to a monitor and allows us to emulate autowave propagation in an efficient way. Experimental results showing the feasibility of FPGA-based approach to implement MCNN are also discussed.

The first step for the FPGA-based implementation is to obtain a discrete model of the MCNN. A discrete-time version of Eqs. (5) is obtained by applying the fourth-order Runge-Kutta integration method [24] as follows:

$$\begin{cases} x_{i,j}(k+1) = x_{i,j}(k) \\ \qquad + \frac{h}{6}(k_{1x;i,j} + 2k_{2x;i,j} + 2k_{3x;i,j} + k_{4x;i,j}), \\ y_{i,j}(k+1) = y_{i,j}(k) \\ \qquad + \frac{h}{6}(k_{1y;i,j} + 2k_{2y;i,j} + 2k_{3y;i,j} + k_{4y;i,j}), \\ z_{i,j}(k+1) = z_{i,j}(k) \\ \qquad + \frac{h}{6}(k_{1z;i,j} + 2k_{2z;i,j} + 2k_{3z;i,j} + k_{4z;i,j}), \end{cases} \tag{6}$$

where $k_{nx;i,j}$, $k_{ny;i,j}$ and $k_{nz;i,j}$ $(n = 1, 2, 3)$ are evaluated as

$$\begin{cases} k_{1x;i,j} = \alpha(-y_{i,j}(k) + \gamma x_{i,j}(k) - W(z_{i,j}(k))x_{i,j}(k)) \\ \qquad + D(x_{i-1,j}(k) + x_{i+1,j}(k) + x_{i,j-1}(k) \\ \qquad + x_{i,j+1}(k) - 4x_{i,j}(k)), \\ k_{1y;i,j} = \beta x_{i,j}(k), \\ k_{1z;i,j} = x_{i,j}(k), \end{cases}$$

$$\begin{cases} k_{2x;i,j} = \alpha[-y_{i,j}(k) - \frac{h}{2}k_{1y;i,j} + \gamma(x_{i,j}(k) + \frac{h}{2}k_{1x;i,j}) \\ \qquad - W(z_{i,j}(k) + \frac{h}{2}k_{1z;i,j})(x_{i,j}(k) + \frac{h}{2}k_{1x;i,j})] \\ \qquad + D[x_{i-1,j}(k) + \frac{h}{2}k_{1x;i-1,j} + x_{i+1,j}(k) \\ \qquad + \frac{h}{2}k_{1x;i+1,j} + x_{i,j-1}(k) + \frac{h}{2}k_{1x;i,j-1} + x_{i,j+1}(k) \\ \qquad + \frac{h}{2}k_{1x;i,j+1} - 4(x_{i,j}(k) + \frac{h}{2}k_{1x;i,j})], \\ k_{2y;i,j} = \beta(x_{i,j}(k) + \frac{h}{2}k_{1x;i,j}), \\ k_{2z;i,j} = x_{i,j}(k) + \frac{h}{2}k_{1x;i,j}, \end{cases}$$

$$\begin{cases} k_{3x;i,j} = \alpha[-y_{i,j}(k) - \frac{h}{2}k_{2y;i,j} + \gamma(x_{i,j}(k) + \frac{h}{2}k_{2x;i,j}) \\ \qquad - W(z_{i,j}(k) + \frac{h}{2}k_{2z;i,j})(x_{i,j}(k) + \frac{h}{2}k_{2x;i,j})] \\ \qquad + D[x_{i-1,j}(k) + \frac{h}{2}k_{2x;i-1,j} + x_{i+1,j}(k) \\ \qquad + \frac{h}{2}k_{2x;i+1,j} + x_{i,j-1}(k) + \frac{h}{2}k_{2x;i,j-1} + x_{i,j+1}(k) \\ \qquad + \frac{h}{2}k_{2x;i,j+1} - 4(x_{i,j}(k) + \frac{h}{2}k_{2x;i,j})], \\ k_{3y;i,j} = \beta(x_{i,j}(k) + \frac{h}{2}k_{2x;i,j}), \\ k_{3z;i,j} = x_{i,j}(k) + \frac{h}{2}k_{2x;i,j}, \end{cases}$$

$$\begin{cases} k_{4x;i,j} = \alpha[-y_{i,j}(k) - hk_{3y;i,j} + \gamma(x_{i,j}(k) + hk_{3x;i,j}) \\ \qquad - W(z_{i,j}(k) + hk_{3z;i,j})(x_{i,j}(k) + hk_{3x;i,j})] \\ \qquad + D[x_{i-1,j}(k) + hk_{3x;i-1,j} + x_{i+1,j}(k) \\ \qquad + hk_{3x;i+1,j} + x_{i,j-1}(k) + hk_{3x;i,j-1} + x_{i,j+1}(k) \\ \qquad + hk_{3x;i,j+1} - 4(x_{i,j}(k) + hk_{3x;i,j})], \\ k_{4y;i,j} = \beta(x_{i,j}(k) + hk_{3x;i,j}), \\ k_{4z;i,j} = x_{i,j}(k) + hk_{3x;i,j}, \end{cases}$$

and $h = 0.004$.

Using the model in Eqs. (6), the MCNN can be implemented in an FPGA-based system with the block diagram shown in Fig. 7. The core of system is the "calculation and control block" which computes the integration of Eqs. (6), displays the results on a monitor and controls the interface with the RAM memory. The RAM memory is an IS42S16400 high-speed synchronous dynamic RAM. It is an external memory provided on the development board and used to store the state variables of the memristive cells.

The whole system consists of an Altera DE2 development board and a monitor. The DE2 board includes a Cyclone II EP2C35F672C6 FPGA chip in a 672-pin package and provides other hardware resources to connect to other external devices. The DE2 board is connected to the monitor through a VGA DAC (10-bit high-speed triple DACs) with VGA-out connector. The calculation and control block, RAM driver block, and monitor driver block are programmed on FPGA chip by using VHDL. Firstly, the calculation and control block based on Altera's Nios II processor emulates the MCNN by integrating Eqs. (6) for a time step. After that, the

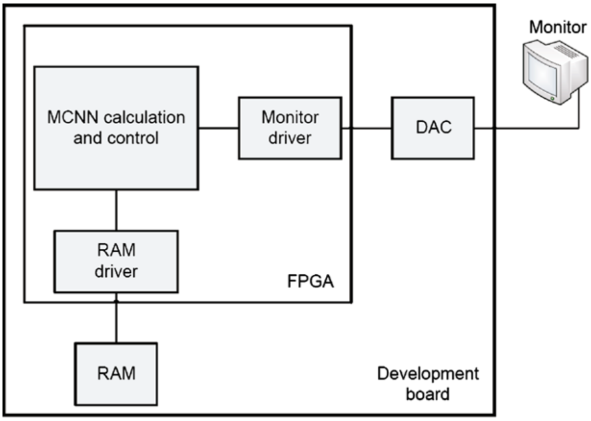


Fig. 7 Block diagram of the whole system based on an FPGA

result is stored in the RAM and displayed on the monitor. The outputs of the FPGA system as shown in the monitor are then captured by a camera. Some examples of the autowaves generated by this system are shown in Figs. 8, 9, and 10. These figures are intended to demonstrate that the generated waves have the main properties of the autowaves.

In Fig. 8 an autowave front moving from the left-hand side to the right-hand side of the MCNN can be observed. When one point at the centre of the MCNN is active, an autowave propagating in the medium forming a spatio-temporal pattern with a circular shape appears as in Fig. 9.

Figure 10 shows two autowaves propagating from the left-hand side and the right-hand side. The two waves move with the same velocity, and annihilate each other when the two wavefronts collide.

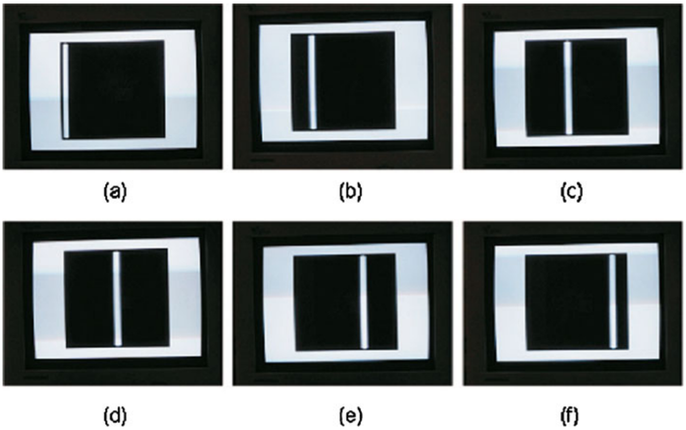


Fig. 8 Autowave propagation from the *left-hand side* to the *right-hand side* of the MCNN

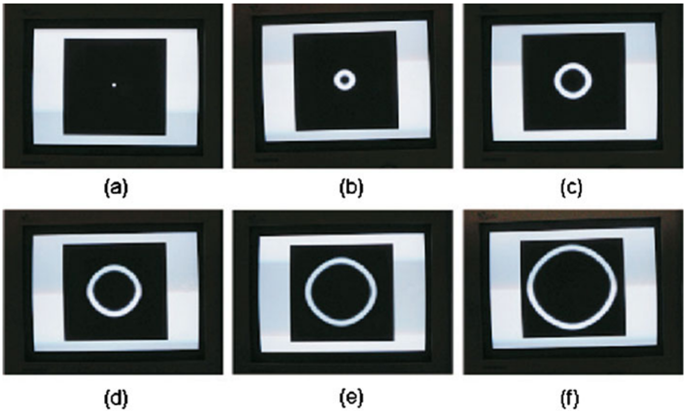


Fig. 9 Autowave formation starts from the *centre* of MCNN

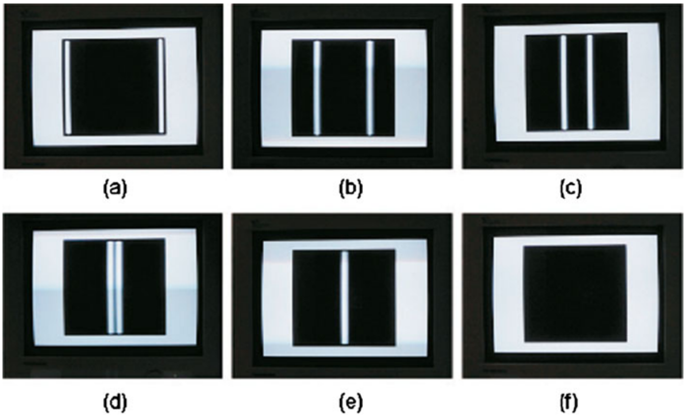


Fig. 10 Annihilation of two colliding autowaves

References

1. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
2. Tour, J.M., He, T.: The fourth element. Nature **453**, 42–43 (2008)
3. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **18**, 507–519 (1971)
4. Pershin, Y.V., La Fontaine, S., Di Ventra, M.: Memristive model of amoeba learning. Phys. Rev. E **80**, 021926 (2009)
5. Driscoll, T., Quinn, J., Klein, S., Kim, H.T., Kim, B.J., Pershin, Y.V., Di Ventra, M., Basov, D.N.: Memristive adaptive filters. Appl. Phys. Lett. **97**, 093502 (2010)
6. Shin, S., Kim, K., Kang Kang, S.M.: Memristor applications for programmable analog ICs. IEEE Trans. Nanotechnol. **410**, 266–274 (2011)
7. Chua, L.O.: Resistance switching memories are memristors. Appl. Phys. A **102**, 765–783 (2011)

8. Chua, L.O., Yang, L.: Cellular neural networks: applications. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **35**, 1273–1290 (1988)
9. Chua, L.O., Yang, L.: Cellular neural networks: theory. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **35**, 1257–1272 (1988)
10. Arena, P., Baglio, S., Fortuna, L., Manganaro, G.: Chua's circuit can be generated by CNN cells. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **42**, 123–125 (1995)
11. Arena, P., Baglio, S., Fortuna, L., Manganaro, G.: How state controlled CNNs generate the dynamics of the Colpitts-like oscillator. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **43**, 602–605 (1996)
12. Perez-Munuzuri, V., Perez-Villar, V., Chua, L.O.: Autowaves for image processing on a two-dimensional CNN array of excitable nonlinear circuits: flat and wrinkled labyrinths. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **40**, 174–181 (1993)
13. Arena, P., Baglio, S., Fortuna, L., Manganaro, G.: Self-organization in a two-layer CNN. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **45**, 157–162 (1998)
14. Perez-Munuzuri, V., Perez-Munuzuri, V., Perez-Villar, V., Chua, L.O.: Spiral waves on a 2-D array of nonlinear circuits. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **40**, 872–877 (1993)
15. Goras, L., Chua, L.O., Leenaerts, D.M.W.: Turing patterns in CNNs—Part I: once over lightly. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **42**, 602–611 (1995)
16. Itoh, M., Chua, L.O.: Memristor cellular automata and memristor discrete time cellular neural networks. Int. J. Bifurc. Chaos **19**, 3605–3656 (2009)
17. Buscarino, A., Fortuna, L., Frasca, M., Gambuzza, L.V., Sciuto, G.: Memristive chaotic circuits based on cellular nonlinear networks. Int. J. Bifurc. Chaos **22**(22), 1250070–1–13 (2012)
18. Lehtonen, E., Laiho, M.: CNN using memristors for neighborhood connections. In: Proc. IEEE Int. Workshop Cellular Nanoscale Networks Appl. (CNNA'2010), pp. 1–4 (2010)
19. Chua, L.O., Roska, T.: Cellular Neural Networks and Visual Computing. Cambridge University Press, Cambridge (2002)
20. Chua, L.O., Hasler, M., Moschytz, G.S., Neirynck, J.: Autonomous cellular neural networks: a unified paradigm for pattern formation and active wave propagation. IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. **42**, 559–577 (1995)
21. Itoh, M., Chua, L.O.: Memristor oscillator. Int. J. Bifurc. Chaos **18**, 3183–3206 (2008)
22. Arena, P., Caponetto, R., Fortuna, L., Manganaro, G.: Cellular neural networks to explore complexity. Soft Comput. **1**, 120–136 (1997)
23. Muthuswamy, B., Chua, L.O.: Simplest chaotic circuit. Int. J. Bifurc. Chaos **20**, 1567–1580 (2010)
24. Kilts, S.: Advanced FPGA Design: Architecture, Implementation, and Optimization, 1st edn. Wiley/IEEE Press, New Jersey (2007)

Memristor Cellular Automata and Memristor Discrete-Time Cellular Neural Networks

Makoto Itoh and Leon Chua

Abstract In this paper, we design a cellular automaton and a discrete-time cellular neural network (DTCNN) using nonlinear passive memristors. They can perform a number of applications, such as logical operations, image processing operations, complex behaviors, higher brain functions, etc. By modifying the characteristics of nonlinear memristors, the memristor DTCNN can perform almost all functions of memristor cellular automaton. Furthermore, it can perform more than one function at the same time, that is, it allows multitasking.

1 Introduction

Recently, a team led by R. Stanley Williams from the Hewlett-Packard Company announced the fabrication of a nanometer-size *memristor*, a contraction for *memory resistor* [15], which was postulated in [1, 4]. This passive electronic device has generated unprecedented worldwide interests because of its potential applications [12, 16] in next generation computers and powerful brain-like "neural" computers. One immediate application offers an enabling low-cost technology for *nonvolatile* memories where future computers would turn on instantly without the usual 'booting time", currently required in all personal computers. Another important application is the construction of *artificial neural networks* [14].

Cellular automaton [17] is a modeling and simulation tool in computer science, mathematics, physics, chemistry, biology, etc. It consists of a grid of cells, where each cell has a finite number of states. Every cell has the same rule for updating, based on the states of neighboring cells. Cellular automata can exhibit many *emergent* phenomena; such as fractals, chaos, randomness, auto waves, etc. Many

M. Itoh (✉)
1-19-20-203, Arae, Jonan-ku, Fukuoka 814-0101, Japan
e-mail: itoh-makoto@jcom.home.ne.jp

L. Chua
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA
e-mail: chua@eecs.berkeley.edu

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5_30,

one-dimensional cellular automata can be classified empirically into one of the four qualitative classes: homogeneous state, stable or periodic structures, chaotic pattern, and complex localized structures, sometimes long-lived [17]. Thus, cellular automata can explain how simple rules can generate complex results [3, 17].

Recently, Wolfman's fundamental research was further developed and extended by Leon O. Chua from the perspective of *neural networks* [3]. In this cellular automaton, the dynamics of the *universal neuron* is defined by a nonlinear *difference equation* (or a differential equation), having one or two *nested absolute-value* functions, and eight adjustable parameters. The universal neuron can generate all possible rules of one-dimensional cellular automata by adjusting these parameters.

Cellular Neural Network (CNN) [2, 5] is a dynamic nonlinear system defined by coupling only identical simple dynamical systems, called *cells*, located within a prescribed sphere of influence, such as nearest neighbors. Because of its simplicity, and ease for chip (hardware) implementation, CNN has found numerous applications in *Image and Video Signal Processing, Robotic and Biological Visions, and Higher Brain Functions*. It is a well-known fact that for many brainlike computations, the CNN universal chip [2, 5] is far superior to any equivalent DSP implementation by at least three orders of magnitude in either speed, power, or area. The CNN has the ability to mimic high level brain functions. Many well-known visual illusions have been simulated by CNN image processing [2, 5, 7, 8, 10].

In this paper, we design a cellular automaton and a cellular neural network using *nonlinear passive* memristors. We first propose a *basic memristor cell*, which can perform logical operations: "AND", "OR", "XOR", and "XNOR". We next design a *memristor cellular automaton* by adding some circuit elements to the basic cell, and we show that the memristor cellular automaton can exhibit some complex behaviors, and also perform image processing applications. We next design a memristor discrete-time cellular neural network (memristor DTCNN), which can perform many image processing operations and higher brain functions. We also show that by modifying the characteristics of memristors, the memristor DTCNN can perform almost all functions of memristor cellular automaton. Furthermore, it can perform more than one functions at the same time. Thus it allows *multitasking*.

2 Cellular Automata

A one-dimensional cellular automaton consists of a row of cells and a set of rules. Each cell can be in one of two states—black or white. Each cell has three neighbors (counting itself), so there are $2^3 = 8$ possible patterns for a neighborhood, and there are $2^8 = 256$ possible rules. Consider the rules defining the following cellular automaton:

1. if all three of these cells are white, then the new state of the cell will be white.
2. if all three of the cells are black, then the new state of the cell will also be white.
3. in any other case, the new state of the cell will be black.

Below is the table defining this rule.

Rule 126

current pattern	111	110	101	100	011	010	001	000
new state for center cell	0	1	1	1	1	1	1	0

(1)

Here, the symbols “1” and “0” denote “black” and “white”, respectively. For example, if three adjacent cells currently have the pattern 100 (left cell is black, middle and right cells are white), then the middle cell will become 1 (black) on the next time step. Since 126 is written “01111110” in binary representation, the above rule is referred to as “rule 126”.

Let us rewrite the truth table of rule 126 in terms of its equivalent *decimal* representations.

Rule 126

$(x_{i-1}(t), x_i(t), x_{i+1}(t))$	111	110	101	100	011	010	001	000
$w = x_{i-1}(t) \bullet 2^2 + x_i(t) \bullet 2^1 + x_{i+1}(t) \bullet 2^0$	7	6	5	4	3	2	1	0
$x_i(t+1)$	0	1	1	1	1	1	1	0

(2)

where $x_i(t)$ denotes a state variable of cell “i” at the discrete-time t. Note that each cell “i” is coupled only to its left neighbor cell $(i-1)$, and right neighbor cell $(i+1)$. From this table, we obtain the relation

$$x_i(t+1) = \begin{cases} 0 & \text{if } w = 0, 7, \\ 1 & \text{if } w = 1, 2, 3, 4, 5, 6, \end{cases} \tag{3}$$

where

$$w \overset{\triangle}{=} x_{i-1}(t) \bullet 2^2 + x_i(t) \bullet 2^1 + x_{i+1}(t) \bullet 2^0. \tag{4}$$

Observe that Rule 126 can be generated by the *difference equation*

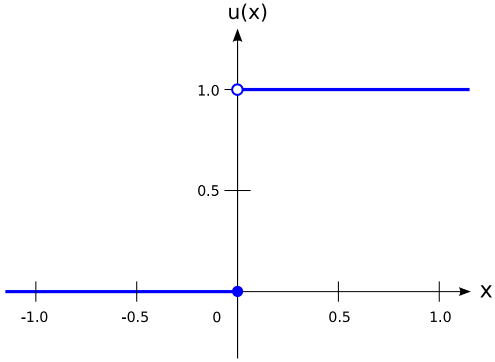
$$x_i(t+1) = u\big(\big|w(w-7)\big|\big) = \begin{cases} 0 & \text{if } w = 0, 7, \\ 1 & \text{if } w = 1, 2, 3, 4, 5, 6, \end{cases} \tag{5}$$

where $u(x)$ is a unit step function defined by

$$u(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ 0, & x < 0. \end{cases} \tag{6}$$

In this paper, $u(x)$ is defined as a piecewise constant function satisfying $u(0) = 0$ (see Fig. 1).

Fig. 1 The unit step function $u(x)$ satisfying $u(0) = 0$



Rule 126 can also be generated by the distinct but equivalent *difference equation*

$$x_i(t+1) = u\big(|w|\big) - u\big(|w| - 6\big), \tag{7}$$

or

$$x_i(t+1) = u\big(\mathit{floor}\big(|w|\big)\big) - u\big(\mathit{floor}\big(|w|\big) - 6\big), \tag{8}$$

where *floor*(x) indicates the largest integer not greater than x (see Fig. 2). We use the floor function to design memristor cellular automata in the latter section.

3 Cellular Neural Networks

Cellular Neural Network (CNN) [2, 5] is a dynamic nonlinear system defined by coupling only identical simple dynamical systems, called cells, located within a prescribed sphere of influence, such as nearest neighbors. The dynamics of a standard cellular neural network with a neighborhood of radius r are governed by a system of $n = MN$ differential equations

$$\frac{dx_{ij}}{dt} = -\gamma x_{ij} + \sum_{k,l \in N_{ij}} (a_{k,l} y_{kl} + b_{k,l} u_{kl}) + z_{ij},$$

$$(i, j) \in \{1, \ldots, M\} \times \{1, \ldots, N\} \tag{9}$$

where N_{ij} denotes the r-neighborhood of cell C_{ij}, and a_{kl}, b_{kl}, and z_{ij} denote the feedback, control and threshold template parameters, respectively. The matrices $A = [a_{kl}]$ and $B = [b_{kl}]$ are referred to as the feedback template A and the feedforward (input) template B, respectively. The output y_{ij} and the state x_{ij} of each cell are usually related via the piecewise-linear saturation function

$$y_{ij} = f(x_{ij}) = \frac{1}{2}\big(|x_{ij} + 1| - |x_{ij} - 1|\big). \tag{10}$$

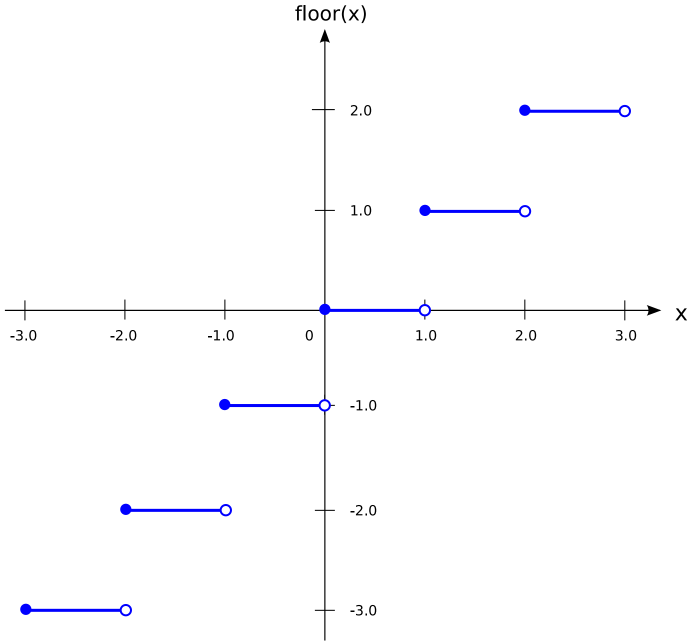


Fig. 2 The floor function *floor*(x)

If we restrict the neighborhood radius of every cell to 1, assume that z_{ij} is the same for the whole network, and set $\gamma = 1$ for the sake of simplicity, the template $\{A, B, z\}$ is fully specified by 19 parameters, which are the elements of two 3×3 matrices A and B, namely

$$A = \begin{array}{|c|c|c|} \hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|} \hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array}, \tag{11}$$

and a real number z.

Approximating the time derivate of Eq. (9) by the formula

$$\frac{dx_{ij}}{dt} = \frac{x_{ij}(t + \Delta t) - x_{ij}(t)}{\Delta t}, \tag{12}$$

we obtain

$$\frac{x_{ij}(t + \Delta t) - x_{ij}(t)}{\Delta t} = -\gamma x_{ij}(t) + \sum_{k,l \in N_{ij}} (a_{k,l} y_{kl}(t) + b_{k,l} u_{kl}) + z_{ij},$$

$$(i, j) \in \{1, \ldots, M\} \times \{1, \ldots, N\} \tag{13}$$

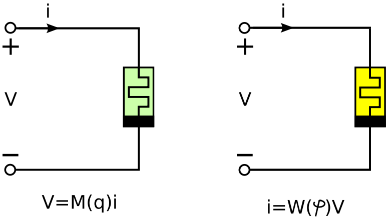


Fig. 3 A charge-controlled memristor (*left*). A flux-controlled memristor (*right*)

If we set $\Delta t = 1$, we would obtain

$$x_{ij}(t+1) = cx_{ij}(t) + \sum_{k,l\in N_{ij}} \big(a_{k,l}y_{kl}(t) + b_{k,l}u_{kl}\big) + z_{ij}, \tag{14}$$

where $c \triangleq 1 - \gamma$. By setting $c = 0$ (that is, $\gamma = 1$), we obtain the discrete-time recursive equation

$$x_{ij}(t+1) = \sum_{k,l\in N_{ij}} \big(a_{k,l}y_{kl}(t) + b_{k,l}u_{kl}\big) + z_{ij}, \tag{15}$$

where $t = 0, 1, 2, \ldots$ and

$$y_{ij}(t) = f\big(x_{ij}(t)\big) = \frac{1}{2}\big(\big|x_{ij}(t)+1\big| - \big|x_{ij}(t)-1\big|\big). \tag{16}$$

Equation (15) is called the *discrete-time CNN* (DTCNN) [5], and it can be transformed into the form

$$y_{ij}(t+1) = f\Bigg(\sum_{k,l\in N_{ij}} \big(a_{k,l}y_{kl}(t) + b_{k,l}u_{kl}\big) + z_{ij}\Bigg). \tag{17}$$

4 Memristor

Memristor [1, 4] in Fig. 3 is a passive 2-terminal electronic device which is described by a nonlinear constitutive relation

$$v = M(q)i \quad \text{or} \quad i = W(\varphi)v, \tag{18}$$

between the device terminal voltage v and the terminal current i. The two nonlinear functions $M(q)$ and $W(\varphi)$, called the *memristance* and *memductance*, respectively, are defined by

$$M(q) \triangleq \frac{d\varphi(q)}{dq} \quad \text{and} \quad W(\varphi) \triangleq \frac{dq(\varphi)}{d\varphi}, \tag{19}$$

representing the *slope* of scalar functions

$$\varphi = \varphi(q) \quad \text{and} \quad q = q(\varphi), \tag{20}$$

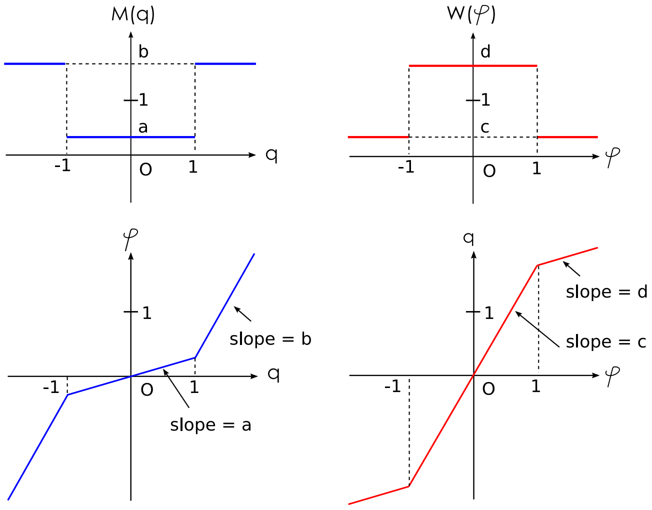


Fig. 4 The constitutive relation of a monotone-increasing piecewise-linear memristor: (*left*) charge-controlled memristor, (*right*) flux-controlled memristor

respectively, called the *memristor constitutive relation* (see Fig. 4).

The fundamental relation between q and i (resp. φ and v) is given by

$$i = \frac{dq}{dt} \qquad \left(\text{resp. } v = \frac{d\varphi}{dt}\right) \tag{21}$$

A memristor characterized by a differentiable q–φ (resp. φ–q) characteristic curve is *passive* if, and only if, its small-signal memristance $M(q)$ (resp. small-signal memductance $W(\varphi)$) is non-negative; i.e.,

$$M(q) = \frac{d\varphi(q)}{dq} \geq 0 \qquad \left(\text{resp.} W(\varphi) = \frac{dq(\varphi)}{d\varphi} \geq 0\right) \tag{22}$$

(see [1]). In this paper, we call them "memristance" and "memductance" for short. Since the instantaneous power dissipated by the above memristor is given by

$$p(t) = M\big(q(t)\big)i(t)^2 \geq 0, \tag{23}$$

or

$$p(t) = W\big(\varphi(t)\big)v(t)^2 \geq 0, \tag{24}$$

the energy flow into the memristor from time t_0 to t satisfies

$$\int_{t_0}^{t} p(\tau)d\tau \geq 0, \tag{25}$$

for all $t \geq t_0$. Thus, the memristor is *passive*.

Note that the memristor was generalized to memristive systems in [4]. These systems are being developed for application in nanoelectronic memories, computer logic, neuromorphic computer architectures, and chaotic oscillators [9, 14].

5 Memristor Cell

The *memristor cellular automaton* we propose in this paper consists of an $M \times N$ rectangular array of memristor cells $C(i, j)$ with Cartesian coordinate (i, j), $i = 1, 2, \dots, M$, $j = 1, 2, \dots, N$. All cells and their interconnections are assumed to be identical.

5.1 Basic Cell

A basic memristor cell $C(i, j)$ consists of a passive memristor and a switch as shown in Fig. 5. A current pulse generator provides a *pulse wave* I_p to memristors and switches in *all memristor cells*. A pulse wave I_p consists of positive-negative *paired* current pulses. A switch is turned on when (and only when) a positive current pulse is applied to the cell as shown in Fig. 5. A switching frequency f is given by

$$f = \frac{1}{T}, \tag{26}$$

where T is a period of the pulse wave.

Referring to Eq. (18), we obtain the cell output $y_{ij}(t)$

$$y_{ij}(t) = \begin{cases} v(t) = M(q(t)) \times I_p(t), & I_p > 0, \\ 0, & I_p < 0. \end{cases} \tag{27}$$

Therefore, if we apply a positive current pulse $I_p(t)$ with a height of 1 and a width of 1 into the memristor, that is,

$$\left. \begin{aligned} & I_p(t) = 1, \\ & q(t) = \int_0^t I_p(t)dt = \int_0^t 1dt = t, \end{aligned} \right\} \tag{28}$$

for $0 \le t \le 1$, we would obtain

$$v(t) = M\big(q(t)\big) \times I_p(t) = M\big(q(t)\big) \times 1 = M\big(q(t)\big). \tag{29}$$

Thus, we can *measure (read)* the memristance $M(q(t))$ from the voltage $v(t)$ across the memristor by applying a positive current pulse $I_p(t)$ to the cell. Note that negative current pulses in Fig. 5 are used to discharge the memristor (that is, to reset the charge of the memristor), which is stored in the memristor by positive current pulses. If the negative current pulses are not applied to the cell, the charge of the memristor continues to increase as shown in Fig. 6.

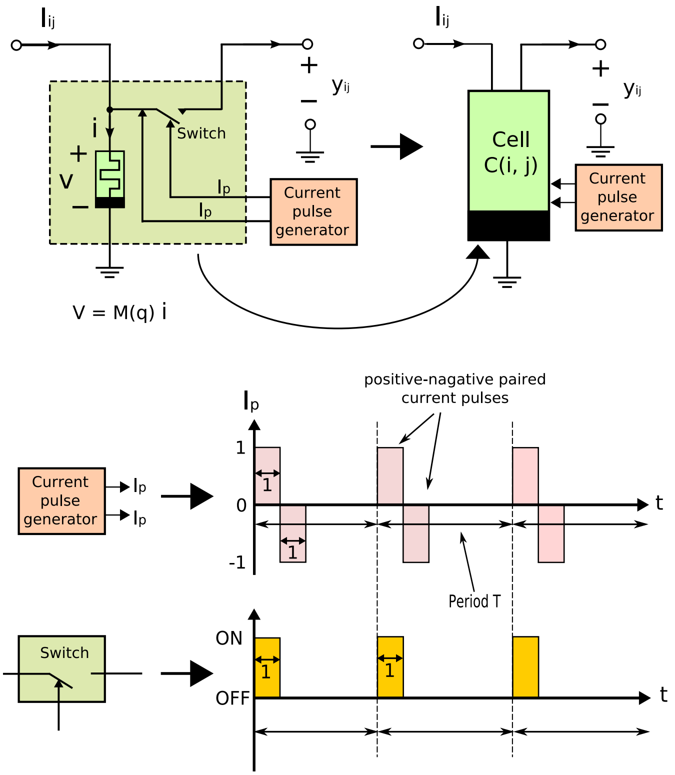


Fig. 5 A basic memristor cell $C(i, j)$ which consists of a memristor and a switch (*top*). A current pulse generator provides a *current pulse wave* (I_p) to a memristor and a switch. The pulse wave I_p consists of positive-negative paired current pulses (*middle*). A switch is turned on when (and only when) a positive current pulse is applied to the cell (*bottom*)

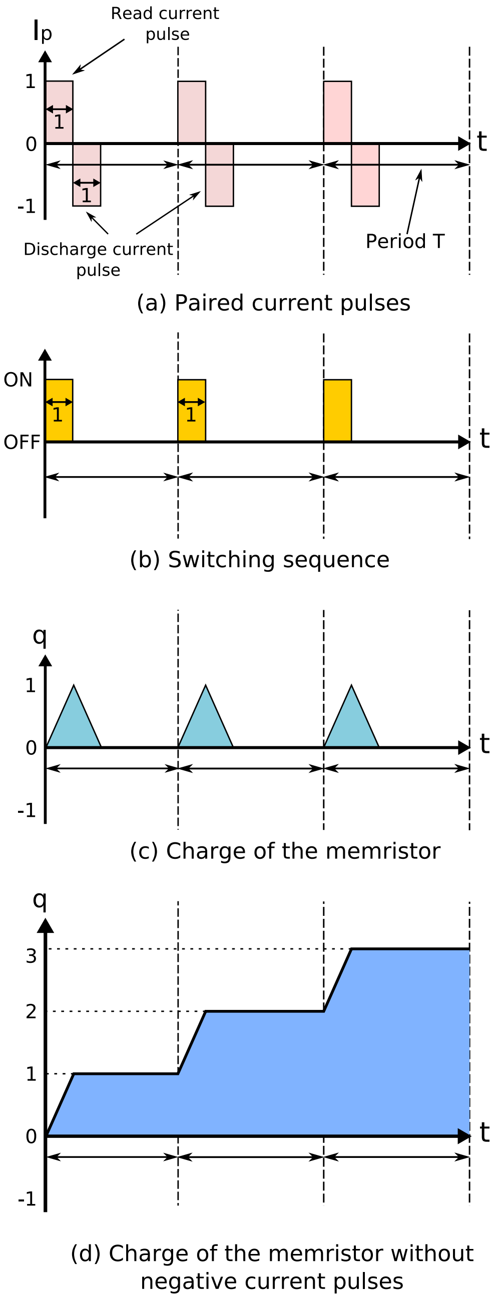
5.2 Logical Operations

In order to realize logical operations, we consider a memristor cell with an input as shown in Fig. 7. The characteristic of the memristor is illustrated in Fig. 8. The input current u_{ij} is applied to the cell by a voltage-controlled current source, whose output current has the same value as the controlled voltage u_{ij}. In this case, each cell $C(i, j)$ works independently, since there are no inputs from neighbors.

We next study the response of the cell $C(i, j)$ when the input u_{ij} in Fig. 9 is applied to the cell. The input pulse u_{ij} consists of positive-negative paired pulses. The negative pulses are used to discharge the memristor, which is stored in the memristor by the positive input pulses. Note the following (see Fig. 9):

1. *Positive* input pulses must be applied to the cell *before* a read pulse I_p is applied to the cell.
2. *Negative* input pulses must be applied to the cell *after* a read pulse I_p is applied to the cell.

Fig. 6 Switching sequence for a current pulse and a charge q of the memristor. (**a**) A current pulse I_p consists of positive-negative paired current pulses. (**b**) The switch is turned on when (and only when) a positive current pulse is applied. (**c**) The negative current pulse discharges the memristor. (**d**) The charge of the memristor continues to increase if the negative current pulses are not applied to the cell



The memristance $M(q)$ illustrated in Fig. 8 is defined by

$$M(q) = \begin{cases} 1, & 1 \le q \le 2, \\ 0, & \text{else}. \end{cases} \tag{30}$$

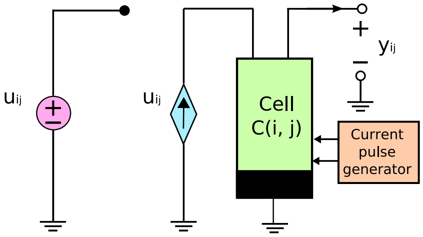


Fig. 7 A memristor cell $C(i, j)$ with an input u_{ij} (a voltage-controlled current source). A diamond-shape symbol denotes a voltage-controlled current source, whose output current has the same value as the controlled voltage u_{ij}

From Eqs. (27)–(30), the cell output y_{ij} is given by

$$y_{ij} = \begin{cases} 1, & 1 \leq q_{ij} \leq 2 \text{ and } I_p > 0, \\ 0, & \text{else.} \end{cases} \tag{31}$$

If the charge q_{ij} satisfies the condition: $1 \leq q_{ij} \leq 2$ and $I_p > 0$, then the cell generates the output pulse (see Fig. 9). Hence, if we apply the input u_{ij} corresponding to the sequence “11”, “01”, “00”, and “10” to the cell, we would obtain the output sequence “0”, “1”, “0”, and “1”, respectively. That is, we obtain the table:

input	“11”	“01”	“00”	“10”
output	0	1	0	1

(32)

Thus, this cell performs the logical operation “XOR”, which is defined by the following truth table:

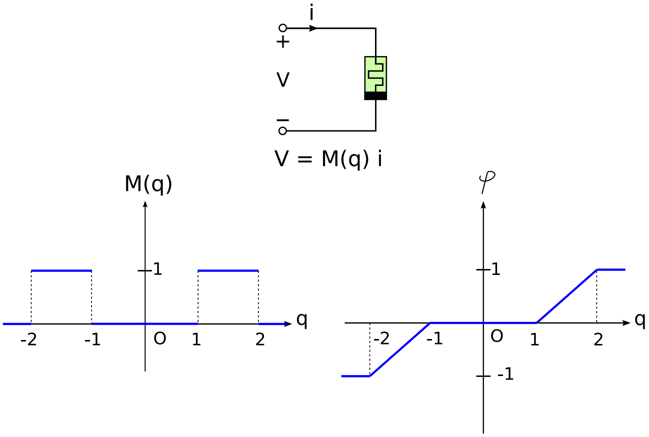


Fig. 8 A passive charge-controlled memristor (*top*). A memristance $M(q)$ (*bottom left*) and a q–φ characteristic curve of a passive charge-controlled memristor (*bottom right*) are illustrated

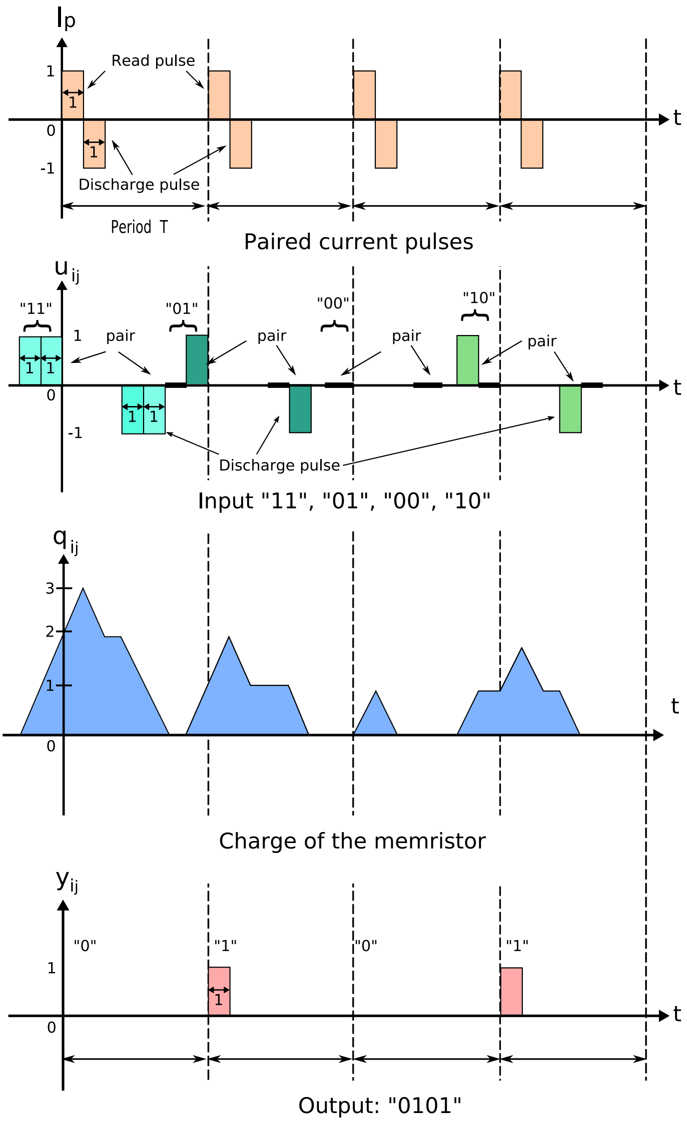


Fig. 9 XOR operation. Time sequences of the current pulse I_p, input u_{ij}, charge q_{ij}, and cell output y_{ij} are illustrated from *top* to *bottom*. The cell generates an output pulse when $1 \leq q_{ij} \leq 2$ and $I_p = 1$

XOR

u_{ij}		y_{ij}
0	0	0
0	1	1
1	0	1
1	1	0

(33)

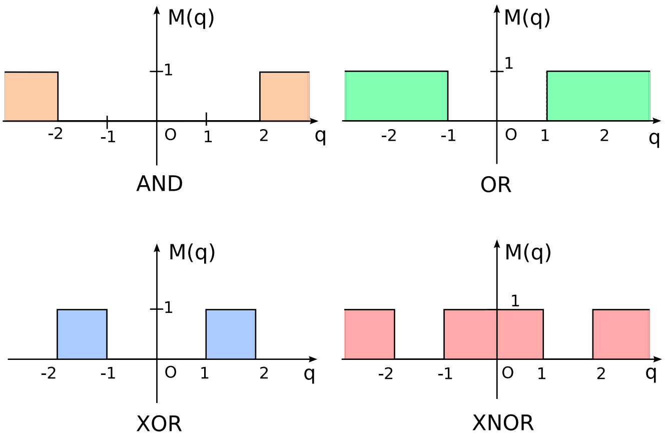


Fig. 10 Memristance $M(q)$ for the logical operations AND, OR, XOR and XNOR

Similarly, the memristance $M(q)$ illustrated in Fig. 10 can perform the logical operations "AND", "OR", and "XNOR", which are defined by the following truth tables:

AND

u_{ij}		y_{ij}
0	0	0
0	1	0
1	0	0
1	1	1

OR

u_{ij}		y_{ij}
0	0	0
0	1	1
1	0	1
1	1	1

XNOR

u_{ij}		y_{ij}
0	0	1
0	1	0
1	0	0
1	1	1

(34)

Compare the time sequences of the XOR operation and the AND operation in Figs. 9 and 11. Observe that the only difference between them is the "output sequence".

The memristance $M(q)$ for the logical operations "AND", "OR", "XOR" and "XNOR" can be described by the following even functions:

Operation	$M(q)$
AND	$u(\lvert q\rvert - 2)$
OR	$u(\lvert q\rvert - 1)$
XOR	$u(\lvert q\rvert - 1) - u(\lvert q\rvert - 2)$
XNOR	$1 - u(\lvert q\rvert - 1) - u(\lvert q\rvert - 2)$

(35)

Note that there are many distinct but equivalent memristances, which can perform the same logical operations. For example, if we define the memristance $M_i(q)$

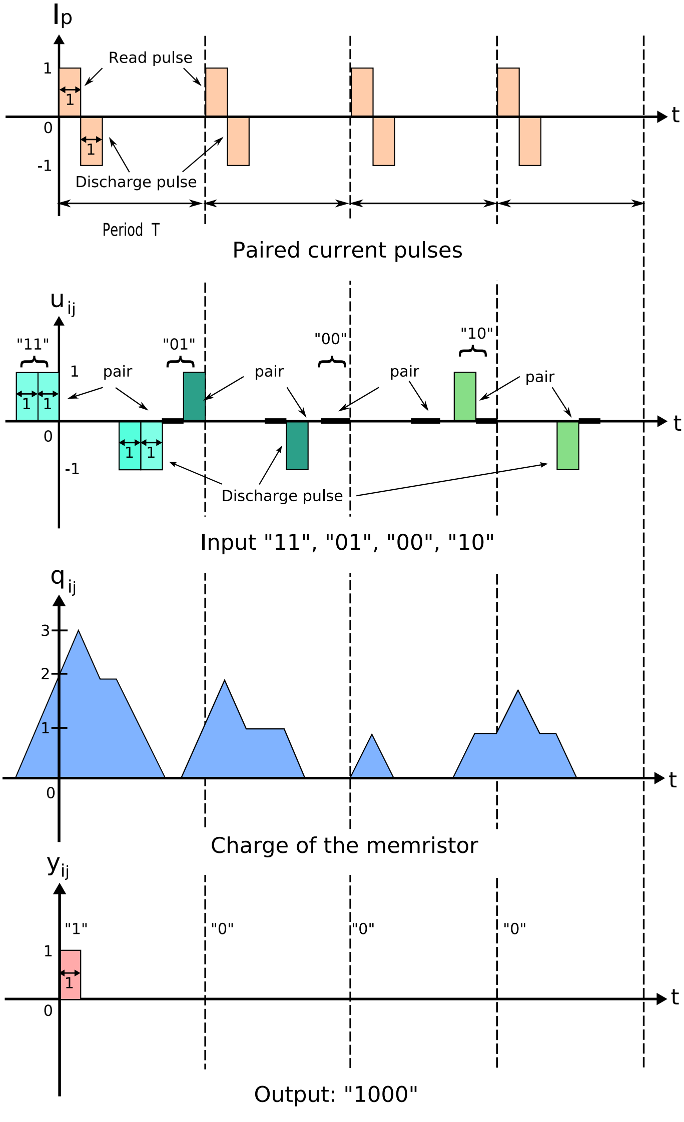


Fig. 11 AND operation. Time sequences of the current pulse I_p, input u_{ij}, charge q_{ij}, and cell output y_{ij} are illustrated from *top* to *bottom*. The cell generates an output pulse when $q_{ij} \geq 2$ and $I_p = 1$

$$M_0(q) = \begin{cases} 1, & |q| < 1, \\ 0, & \text{else}, \end{cases}$$

$$M_1(q) = \begin{cases} 1, & 1 \leq |q| < 2, \\ 0, & \text{else}, \end{cases} \tag{36}$$

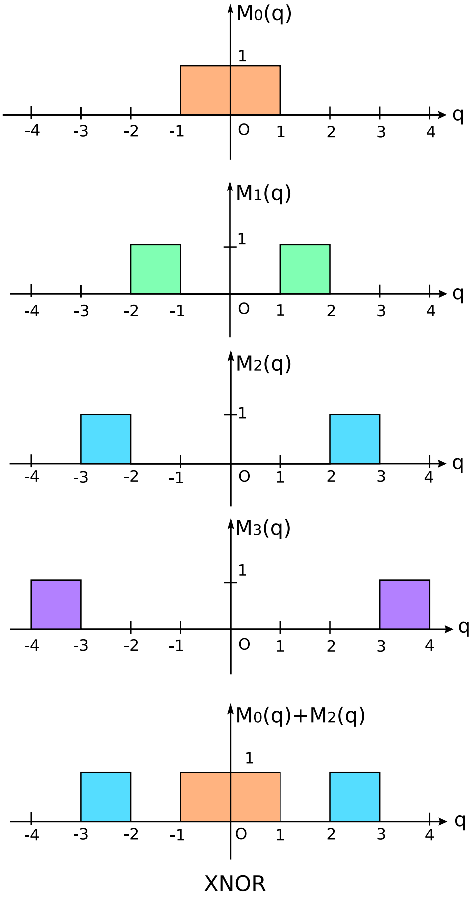


Fig. 12 Characteristic of memristances $M_i(q)$ $(i = 1, 2, 3)$. The memristance $M_0(q) + M_2(q)$ can perform the XNOR operation

$$M_2(q) = \begin{cases} 1, & 2 \le |q| < 3, \\ 0, & \text{else}, \end{cases}$$

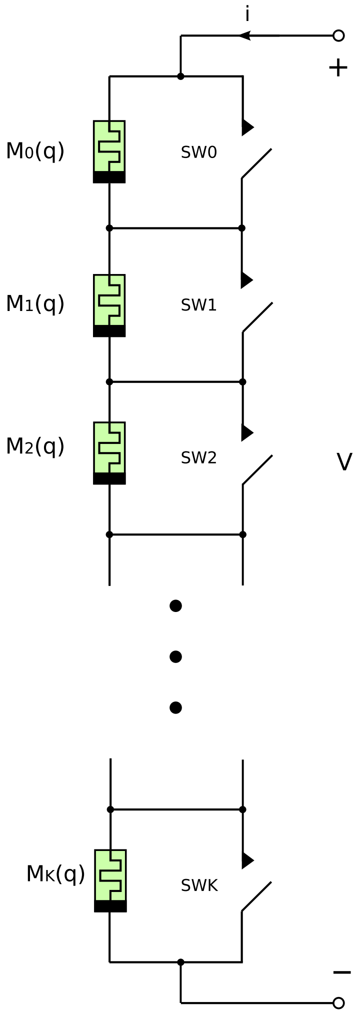
$$\vdots$$

$$M_i(q) = \begin{cases} 1, & i \le q < i+1, \\ 0, & \text{else}, \end{cases}$$

the memristance for the logical operation XNOR is given by $M_0(q) + M_2(q)$ as shown in Fig. 12.

Similarly, the memristances for the logical operations “AND”, “OR”, and “XOR” are given by the table:

Fig. 13 A circuit consisting of memristors and switches



Operation	Memristance
AND	$M_2(q)$
OR	$M_1(q) + M_2(q)$
XOR	$M_1(q)$
XNOR	$M_0(q) + M_2(q)$

(37)

5.3 Series Connection of Memristors

Consider the circuit of Fig. 13, which consists of memristors and switches. If two memristors with memristance $M_0(q)$ and $M_2(q)$ are connected in series as shown

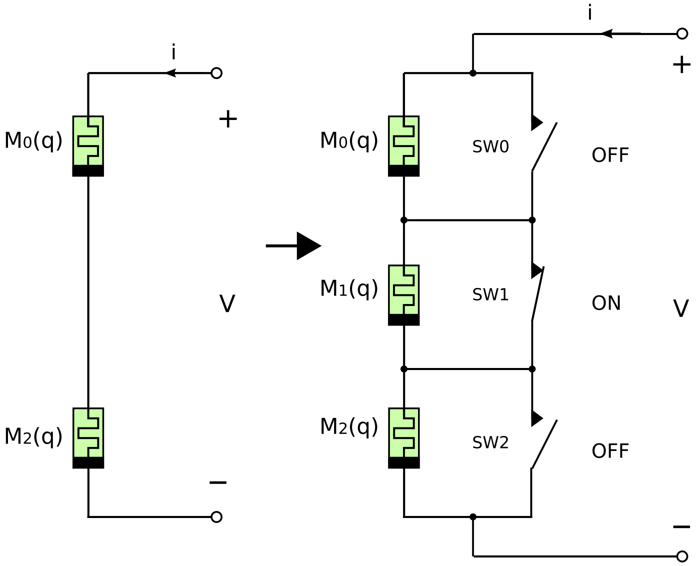


Fig. 14 A circuit realizing the memristance $M_0(q) + M_2(q)$

in Fig. 14, then we have

$$v = M_0(q)i + M_2(q)i = \big(M_0(q) + M_2(q)\big)i. \tag{38}$$

Thus, two series memristors are equivalent to a single memristor whose memristance is the sum of memristances, that is, $M_0(q) + M_2(q)$. It follows that these series memristors can perform the XNOR operation. We also realize many kinds of memristances by using the circuit in Fig. 13.

5.4 Asynchronous Inputs

Consider the memristor cell with asynchronous inputs from only eight nearest neighbors as shown in Fig. 15. The weighting coefficients $b_{k,l}$ $(k,\ l \in \{-1, 0, 1\})$ are usually listed as entries in the following table:

$$B \triangleq \begin{array}{|c|c|c|} \hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array} \tag{39}$$

Assume that the memristor cell satisfies the conditions:

1. Only two inputs $u_{i,j}$ and $u_{i-1,j}$ are available.

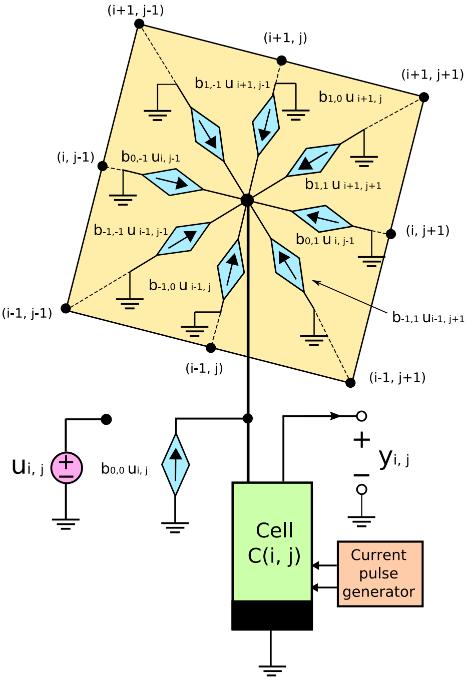


Fig. 15 A memristor cell $C(i, j)$ with asynchronous inputs from eight neighbor cells. Weighting coefficients $b_{k,l}$ $(k, l \in \{-1, 0, 1\})$ are some constants

2. Effective weighting coefficients $b_{k,l}$ are equal to 1. Thus, the weighting coefficients $b_{k,l}$ can be written as

$$B = \begin{array}{|c|c|c|}\hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \tag{40}$$

3. The memristance $M(q)$ is defined by Eq. (30), that is,

$$M(q) = \begin{cases} 1, & 1 \le q \le 2, \\ 0, & \text{else}. \end{cases}$$

Then, by applying the two asynchronous input sequence (the input $u_{i-1,j}$ precedes the input $u_{i,j}$):

$u_{i,j}$	1	1	0	0
$u_{i-1,j}$	1	0	0	1

(41)

to this cell, we obtain the output sequence:

$y_{i,j}$	0	1	0	1

(42)

as shown in Fig. 16.

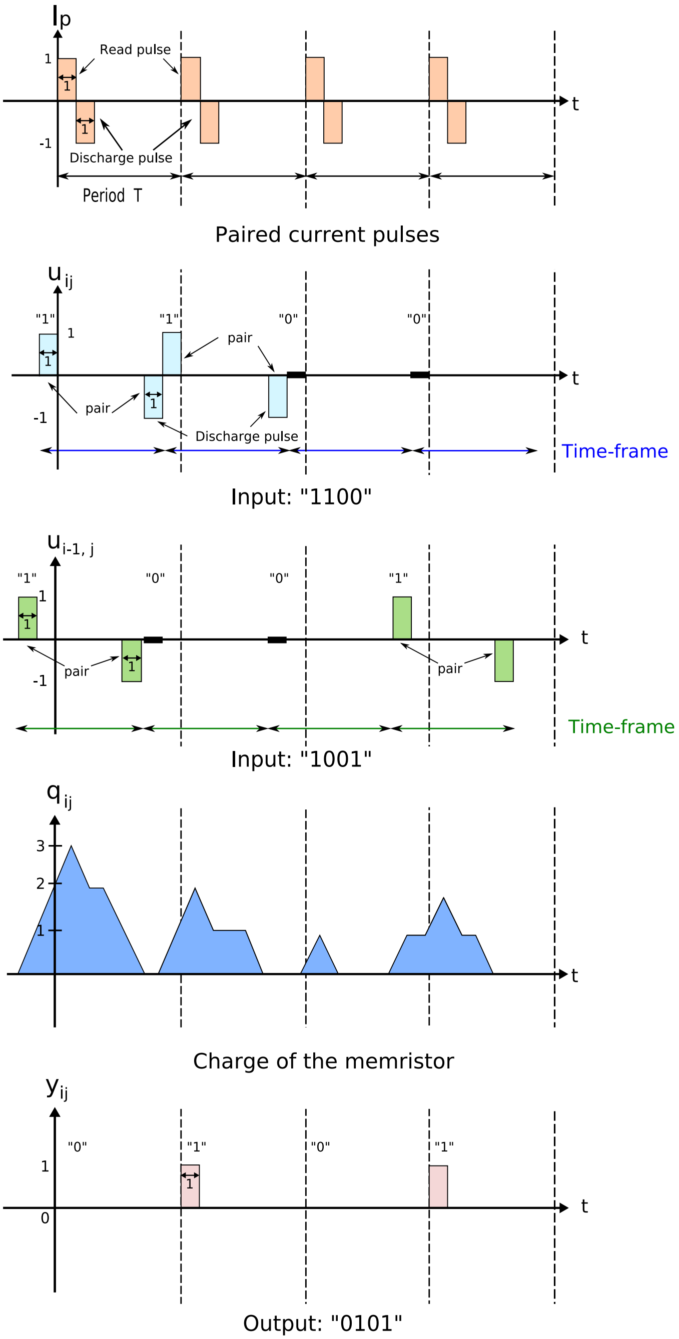


Fig. 16 Time sequences of the memristor cell with two inputs u_{ij} and u_{i-1j}. The pulse sequences of inputs u_{ij} and u_{i-1j} indicate the inputs “1100” and “1001”, respectively. The cell generates an output pulse when $1 \leq q_{ij} \leq 2$ and $I_p = 1$. Thus, we get the output sequence “0101”

That is, we have the following table:

$u_{i,j}$	1	1	0	0
$u_{i-1,j}$	1	0	0	1
$y_{i,j}$	0	1	0	1

(43)

It can realize the truth table of XOR:

XOR

$u_{i,j}$	$u_{i-1,j}$	y_{ij}
0	0	0
0	1	1
1	0	1
1	1	0

(44)

Observe from Fig. 16 that inputs $u_{i,j}$ and $u_{i-1,j}$ have different *time-frames*, which provide timing information.[1] Therefore, the inputs are not necessarily applied to the cell simultaneously.

6 Memristor Cellular Automaton

We propose a *memristor cellular automaton* in Fig. 17, which is quite similar to the realization of a cellular neural network (CNN) cel [2, 5]. A memristor cellular automaton consists of a memristor cell $C(i, j)$, a current pulse generator, a signal generator, and voltage-controlled current sources denoted by a pink-colored diamond-shape symbol. A signal generator provides the signal $z_{ij}(t)$ satisfying the equation

$$z_{ij}(t) = y_{ij}(t - T) - y_{ij}(t - 2T + 1), \tag{45}$$

where $y_{ij}(t - T)$ and $-y_{ij}(t - 2T + 1)$ corresponds to a positive pulse and a negative pulse in Fig. 18, respectively.

A negative pulse is used to reset the charge of the memristor, stored by a positive pulse. The charge $q_{ij}(t)$ stored in the memristor during the period $[nT, nT + \Delta t]$ $(n = 1, 2, \ldots)$ is given by

$$\begin{aligned} q_{ij}(nT + \Delta t) &= \sum_{k,l \in (-1,0,1)} a_{k,l} z_{i+k,j+l}(nT) + \Delta t \\ &= \sum_{k,l \in (-1,0,1)} a_{k,l} y_{i+k,j+l}\big((n-1)T\big) + \Delta t, \end{aligned} \tag{46}$$

[1] A time-frame is used to identify a start and end of a set of current pulses.

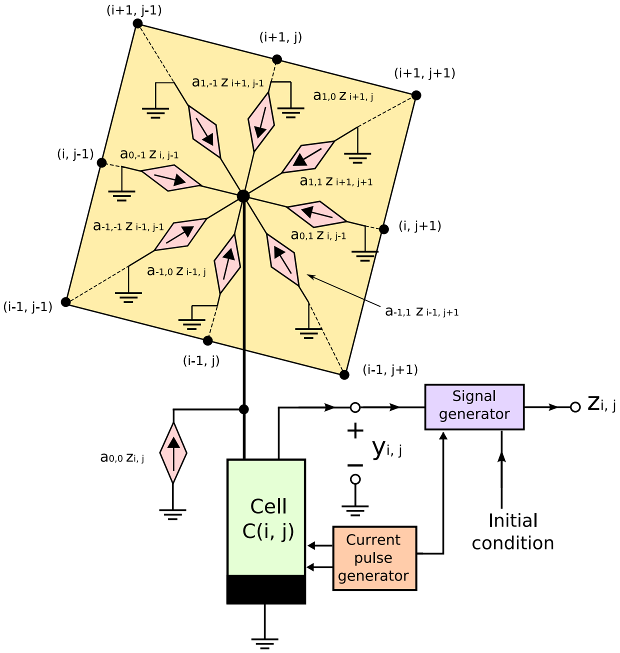


Fig. 17 A memristor cellular automaton cell, which consists of a memristor cell $C(i, j)$ (*light green*), a current pulse generator (*salmon pink*), a signal generator (*medium purple*), and voltage–controlled current sources denoted by a diamond-shape symbol (*pink*)

where $0 < \Delta t < 1$. The last term Δt indicates the charge stored by a positive current pulse (see Fig. 6). Thus, from Eqs. (27) and (29), we have the relation

$$y_{ij}(nT + \Delta t) = \begin{cases} M(q_{ij}(nT + \Delta t)) & \\ \quad = M(\sum_{k,l \in (-1,0,1)} a_{k,l} y_{i+k,j+l}((n-1)T) + \Delta t), & I_p > 0, \\ 0, & I_p < 0. \end{cases} \tag{47}$$

We assume that $M(q_{ij}(nT + \Delta t))$ does not change for $0 < \Delta t < 1$ (that is, the time period when a read pulse is applied to a memristor cell).

6.1 Rule 126 Memristor Cellular Automaton

Consider a cellular automaton, which consists of a *one-dimensional* array of cells $C(i)$. The local rule 126 was defined by

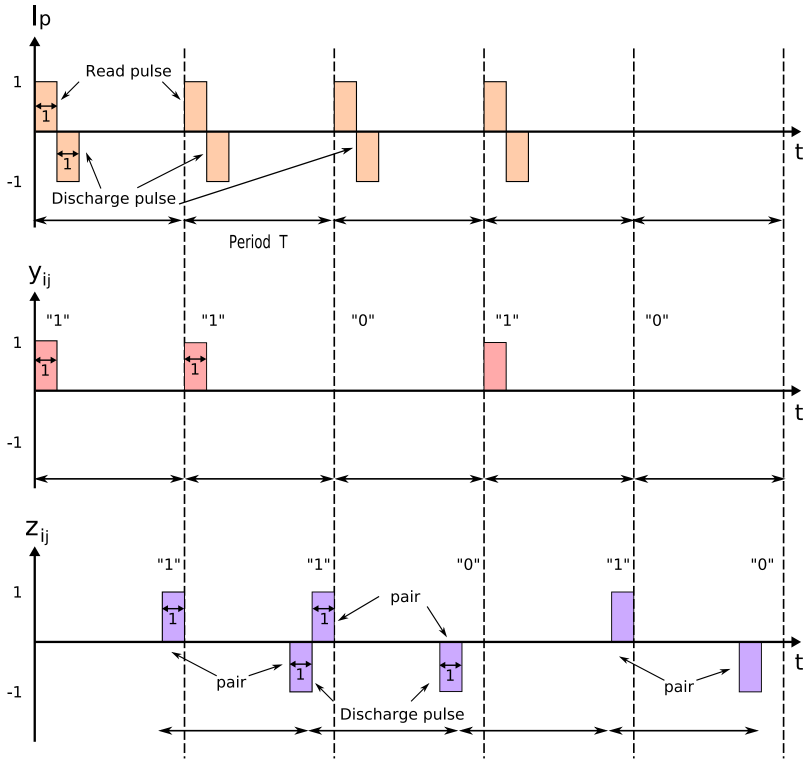


Fig. 18 Time sequences of the memristor cellular automaton. The current pulse I_p, output y_{ij} of cell $C(i, j)$, and output z_{ij} of the signal generator are illustrated from *top* to *bottom*

Rule 126

$\sum_{k=-1}^{1} y_{i+k}((n-1)T) \bullet 2^{1-k}$	7	6	5	4	3	2	1	0
new state of center cell: $y_i(nT)$	0	1	1	1	1	1	1	0

(48)

where $y_i(nT)$ denotes a state variable of cell "$C(i)$" at the discrete-time nT. Applying Eq. (8) to Eq. (47), we can obtain the difference equation:

$$\begin{aligned} y_i(nT + \Delta t) = u\Bigg(\mathit{floor}\Bigg(\bigg| \sum_{k \in (-1,0,1)} a_k y_{i+k}\big((n-1)T\big) + \Delta t \bigg| \Bigg) \Bigg) \\ - u\Bigg(\mathit{floor}\Bigg(\bigg| \sum_{k \in (-1,0,1)} a_k y_{i+k}\big((n-1)T\big) + \Delta t \bigg| - 6 \Bigg) \Bigg), \end{aligned} \quad (49)$$

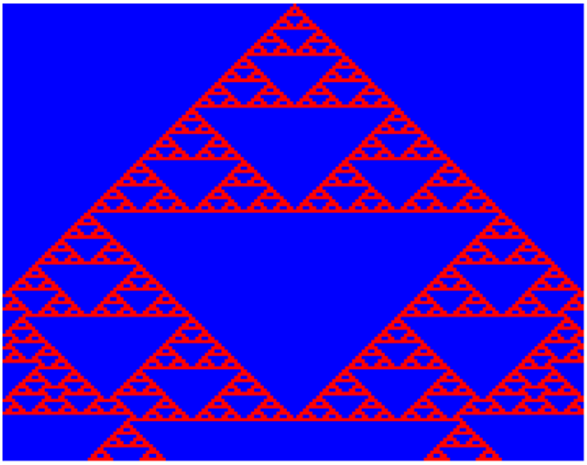


Fig. 19 Evolution of a rule 126 memristor cellular automaton

where $I_p > 0$, and the weighting coefficients a_k and the memristance $M(q)$ are given by

$$A \triangleq \boxed{a_{-1}}\boxed{a_0}\boxed{a_1} = \boxed{4}\boxed{2}\boxed{1}, \tag{50}$$

and

$$M(q) = u\big(floor(|q|)\big) - u\big(floor(|q|) - 6\big), \tag{51}$$

respectively. The memristance $M(q)$ can also be written as

$$M(q) = \sum_{i=1}^{6} M_i(q) = \begin{cases} 0, & \text{if } q = 0, 7, \\ 1, & \text{else,} \end{cases} \tag{52}$$

where $M_i(q)$ was defined by Eq. (36). Since we apply a positive current pulse to the cell in order to read the memristance $M(q)$ (thus, the charge $q(t)$ is increased by a positive current pulse), the floor functions are inserted into Eq. (51).

The evolution of Eq. (49) is illustrated in Fig. 19, where the two states "1" and "0" are colored "red" and "blue", respectively.[2]

[2] All evolution images and processed images in this paper are not obtained by the real memristor cellular automaton circuits, but by computer simulations.

Since the output $y_i((n-1)T + \Delta t)$ does not change for $0 < \Delta t < 1$, we obtain

$$y_i\big((n-1)T + \Delta t\big) = y_{ij}\big((n-1)T\big), \tag{53}$$

and

$$\begin{aligned} & \mathit{floor}\left(\left|\sum_{k \in (-1,0,1)} a_k y_{i+k}\big((n-1)T\big) + \Delta t\right|\right) \\ &= \mathit{floor}\left(\left|\sum_{k,l \in (-1,0,1)} a_k y_{i+k}\big((n-1)T\big)\right|\right) \\ &= \mathit{floor}\left(\left|\sum_{kl \in (-1,0,1)} a_{k,l} y_{i+k,j+l}\big((n-1)T + \Delta t\big)\right|\right) \\ &= \mathit{floor}\left(\left|\sum_{k \in (-1,0,1)} a_k y_{i+k}(\tau_{n-1})\right|\right), \end{aligned} \tag{54}$$

where $0 < \Delta t < 1$ and $\tau_n \triangleq nT + \Delta t$. Since all state variables y_{ij} and weighting coefficients $a_{k,l}$ are non-negative integers, the floor functions and absolute functions can be removed from Eq. (49). Thus, we have the equivalent difference equation for rule 126:

$$\begin{aligned} y_i(\tau_n) = u & \left(\sum_{k \in (-1,0,1)} a_k y_{i+k}(\tau_{n-1})\right) \\ & - u\left(\sum_{k \in (-1,0,1)} a_k y_{i+k}(\tau_{n-1}) - 6\right), \end{aligned} \tag{55}$$

or

$$y_i(\tau_n) = \sum_{i=1}^{6} M_i\left(\sum_{k \in (-1,0,1)} a_k y_{i+k}(\tau_{n-1})\right). \tag{56}$$

6.2 Sierpinski Memristor Cellular Automaton

In a multistate memristor cellular automaton, each cell has one out of m possible states represented by numbers $0, 1, 2, \ldots, m-1$. For example, a 4-state (or 4-color) Sierpinski cellular automaton can generate Pascal's triangle and has four states $\{0, 1, 2, 3\}$ [13]. The rule for this cellular automaton can be written as follows:

Rule of a 4-state Sierpinski cellular automaton

$y_{i-1}((n-1)T)$	$y_i((n-1)T)$	$y_{i-1}((n-1)T) + y_i((n-1)T)$	$y_i(nT)$
0	0	0	0
0	1	1	1
0	2	2	2
0	3	3	3
1	0	1	1
1	1	2	2
1	2	3	3
1	3	4	0
2	0	2	2
2	1	3	3
2	2	4	0
2	3	5	1
3	0	3	3
3	1	4	0
3	2	5	1
3	3	6	2

(57)

The memristance and the difference equation which can generate this rule are given by

$$M(q) = \mathit{floor}\big(|q|\big) \pmod 4, \tag{58}$$

and

$$y_i(nT + \Delta t) = \mathit{floor}\left(\left| \sum_{k \in (-1,0,1)} a_k y_{i+k}\big((n-1)T\big) + \Delta t \right|\right) \pmod 4, \tag{59}$$

respectively, where $m \pmod 4$ is the remainder, on division of m by 4, and the weighting coefficient matrix $A = [a_{k,l}]$ is given by

$$A = \begin{array}{|c|c|c|}\hline a_{-1} & a_0 & a_1 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline \end{array}. \tag{60}$$

Evolutions of Eq. (59) is illustrated in Fig. 20. We used the following palette for the state of cells:

Color palette

state	0	1	2	3
color	orange	blue	red	green

(61)

Since all state variables y_{ij} and weighting coefficients $a_{k,l}$ are non-negative integers, Eq. (59) can be transformed into an equivalent difference equation

$$y_i(\tau_n) = \sum_{k \in (-1,0,1)} a_k y_{i+k}(\tau_{n-1}) \pmod 4, \tag{62}$$

where $\tau_n \triangleq nT + \Delta t$.

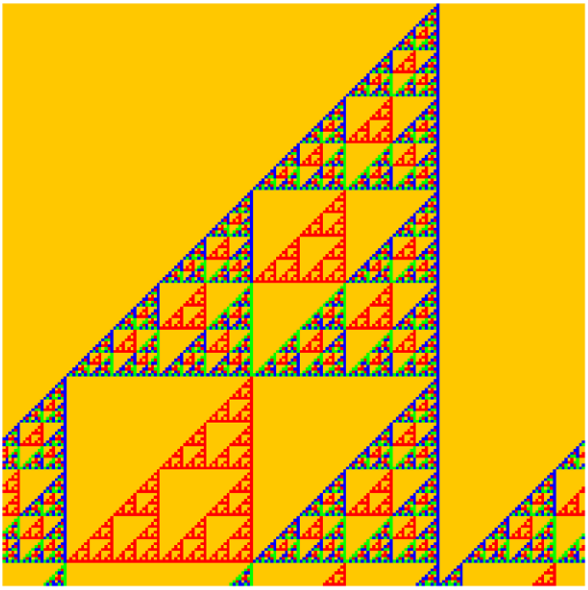


Fig. 20 Evolution of a 4-state Sierpinski memristor cellular automaton

6.3 Totalistic Two-Dimensional Memristor Cellular Automaton

A *totalistic* cellular automaton is a special class of cellular automata in which the rules depend on the total values of cells in a neighborhood. The evolution of a totalistic cellular automaton can be described by a table specifying the state of the neighbors (counting itself). Each of the cells can be in one of two states; black (cell value 1) or white (cell value 0). If each cell has nine neighbors (counting itself), there are 10 possible patterns for a neighborhood, and there are $2^{10} = 1024$ possible rules [13].

Consider the following totalistic rules for the two-dimensional cellular automaton with 9-neighbor cells:

Rule 797

$\sum_{k=-1}^{1} \sum_{m=-1}^{1} y_{i+k,j+m}((n-1)T)$	9	8	7	6	5	4	3	2	1	0
new state of center cell: $y_{ij}(nT)$	1	1	0	0	0	1	1	1	0	1

(63)

Since 797 is written “1100011101” in binary representation, the above rule is referred to as “rule 797”. The memristance and the difference equation which can generate this rule are given by

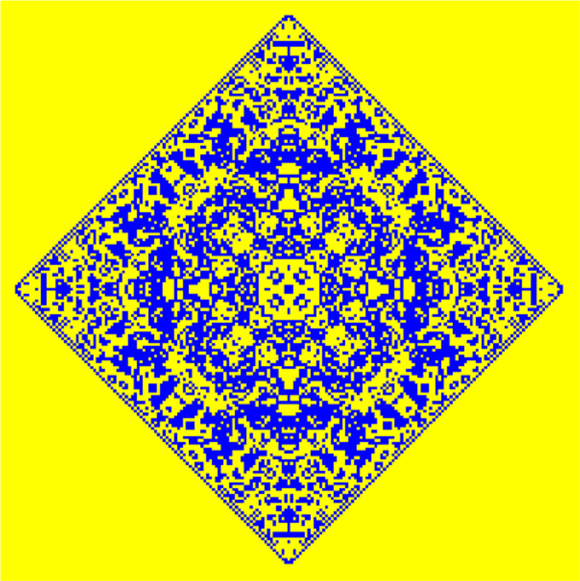


Fig. 21 Evolution of a rule 797 two-dimensional cellular automaton

$$M(q) = M_0(q) + M_2(q) + M_3(q) + M_4(q) + M_8(q) + M_9(q)$$
$$= \begin{cases} 1, & \text{if } q = 0, 2, 3, 4, 8, 9 \\ 0, & \text{if } q = 1, 5, 6, 7, \end{cases} \tag{64}$$

and

$$y_{ij}(nT + \Delta t) = M\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) + \Delta t \Bigg), \tag{65}$$

respectively, and the weighting coefficient matrix $A = [a_{k,l}]$ is given by

$$A = \begin{array}{|c|c|c|} \hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}. \tag{66}$$

Our computer simulation of the totalistic cellular automaton is shown in Fig. 21, where the two states "1" and "0" are colored "yellow" and "blue", respectively. Observe that it has a complex pattern. Since all state variables y_{ij} and weighting coefficients $a_{k,l}$ are non-negative integers, Eq. (59) can be transformed into an equivalent difference equation

$$y_{ij}(\tau_n) = M\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}(\tau_{n-1}) \Bigg), \tag{67}$$

where $\tau_n \triangleq nT + \Delta t$.

6.4 Horizontal Hole Detection Memristor Cellular Automaton

Horizontal hole detection cellular automaton detects the number of horizontal holes from each horizontal row of a binary image. A string of adjacent white pixels in a horizontal row is called a *horizontal hole* if each end is terminated at least one black pixel [2]. The rule for this cellular automaton can be written as follows:

Rule of a horizontal hole detection cellular automaton

$y_{i-1,j}((n-1)T)$	$y_{i,j}((n-1)T)$	$y_{i+1,j}((n-1)T)$	w	$y_{i,j}(nT)$
0	0	0	0	0
0	0	1	1	0
0	1	0	2	1
0	1	1	3	0
1	0	0	4	1
1	0	1	5	0
1	1	0	6	1
1	1	1	7	1

(68)

where '1" and "0" in the rule table denote "black" and "white", respectively, and

$$w = y_{i-1,j}\big((n-1)T\big) \bullet 2^2 + y_{i,j}\big((n-1)T\big) \bullet 2^1 + y_{i+1,j}\ \big((n-1)T\big) \bullet 2^0. \quad (69)$$

The memristance and the difference equation which can generate this rule are given by

$$\begin{aligned} M(q) &= M_2(q) + M_4(q) + M_6(q) + M_7(q) \\ &= \begin{cases} 1, & q = 2, 4, 6, 7, \\ 0, & q = 0, 1, 3, 5, \end{cases} \end{aligned} \quad (70)$$

and

$$y_{ij}(nT + \Delta t) = M\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) + \Delta t \Bigg), \quad (71)$$

respectively, where the weighting coefficient matrix $A = [a_{k,l}]$ is given by

$$A = \begin{array}{|c|c|c|} \hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 4 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}. \quad (72)$$

Our computer simulation of the horizontal hole detection cellular automaton is shown in Fig. 22. Observe that the horizontal holes are detected by this cellular automaton. Similarly, the weighting coefficient matrix $A = [a_{k,l}]$ for vertical hole

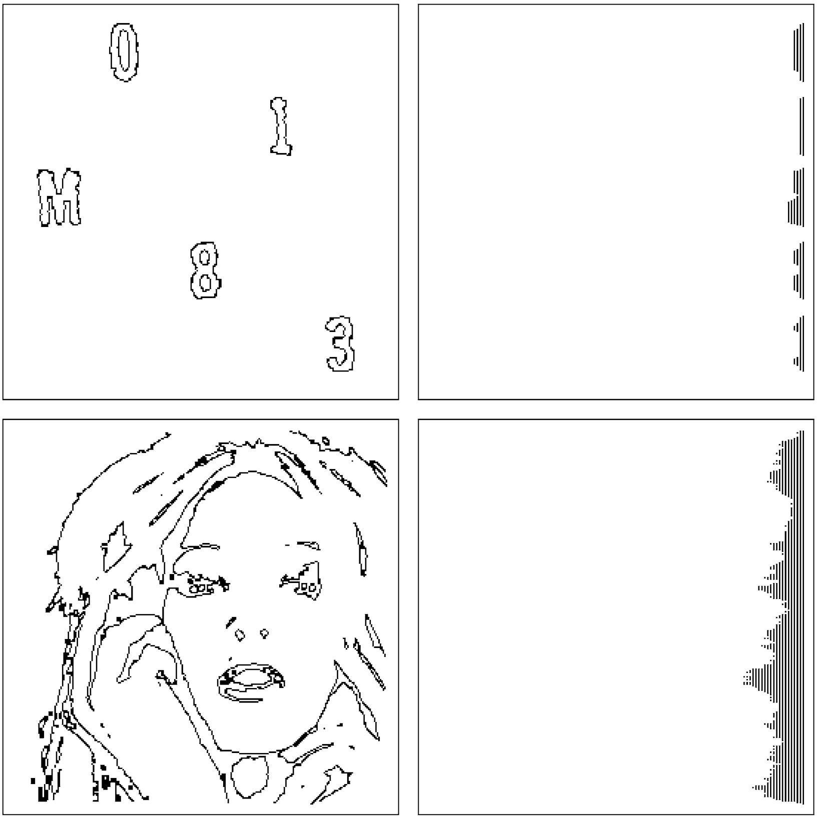


Fig. 22 A horizontal hole detection memristor cellular automaton. Initial state images and processed images are illustrated from *left* to *right*. The number of horizontal holes is indicated by the number of *vertical lines* on the *right images*

detection is given by

$$A = \begin{array}{|c|c|c|}\hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0 & 4 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 1 & 0 \\ \hline \end{array}. \tag{73}$$

Since all state variables y_{ij} and weighting coefficients $a_{k,l}$ are non-negative integers, Eq. (71) can be transformed into an equivalent difference equation

$$y_{ij}(\tau_n) = M\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l} y_{i+k,j+l}(\tau_{n-1}) + \Delta t \Bigg), \tag{74}$$

where $\tau_n \triangleq nT + \Delta t$.

6.5 Edge Detection Memristor Cellular Automaton

An edge is a jump in intensity from one pixel to the next. Define the rule of the edge detection as follows [11]:

1. if the number of black cells is equal to 6, 7, or 8, then the new state of the cell will be black.
2. in any other case, the new state of the cell will be white.

The truth table defining the above edge detection can be written as follows:

Rule of Edge Detection

$\sum_{k=-1}^{1}\sum_{m=-1}^{1} y_{i+k,j+m}((n-1)T)$	9	8	7	6	5	4	3	2	1	0
new state of center cell: $y_{ij}(nT)$	0	1	1	1	0	0	0	0	0	0

(75)

where $y_{ij}(nT)$ has two states {0, 1}, and "1" and "0" in the rule table denote "black" and "white" of the binary image, respectively. Here, we assumed that there are a few white cells in the neighborhood of an edge. The memristance which can generate the local rule for edge detection is given by

$$M(q) = M_6(q) + M_7(q) + M_8(q) = \begin{cases} 0, & q = 0, 1, 2, 3, 4, 5, 9, \\ 1, & q = 6, 7, 8, \end{cases} \tag{76}$$

or

$$M(q) = u\big(\mathit{floor}\big(|q|\big) - 5\big) - u\big(\mathit{floor}\big(|q|\big) - 8\big). \tag{77}$$

The difference equation is given by

$$\begin{aligned} y_{ij}(nT + \Delta t) = {} & u\left(\mathit{floor}\left|\sum_{k,l\in(-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) + \Delta t\right| - 5\right) \\ & - u\left(\mathit{floor}\left|\sum_{k,l\in(-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) + \Delta t\right| - 8\right), \end{aligned} \tag{78}$$

where the weighting coefficient matrix $A = [a_{k,l}]$ is given by

$$A = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}\,. \tag{79}$$

Our computer simulation of the edge detection cellular automaton is shown in Fig. 23. Observe that the edges of objects are extracted by this cellular automaton.

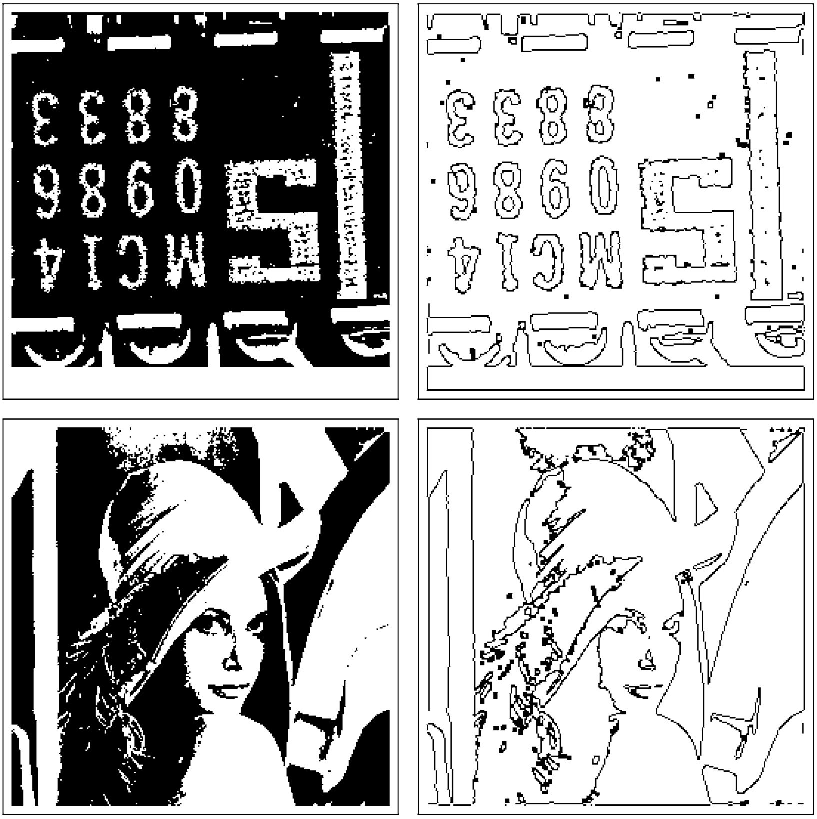


Fig. 23 An edge detection memristor cellular automaton. Initial state images and detected edge images are illustrated from *left* to *right*

Since all state variables y_{ij} and weighting coefficients $a_{k,l}$ are non-negative integers, Eq. (78) can be transformed into an equivalent difference equation

$$\begin{aligned} y_{ij}(\tau_n) = u\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l} y_{i+k,j+l}(\tau_{n-1}) - 5 \Bigg) \\ - u\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l} y_{i+k,j+l}\ (\tau_{n-1}) - 8 \Bigg), \end{aligned} \tag{80}$$

or

$$y_{ij}(\tau_n) = \sum_{i=6}^{8} M_i \Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l}\ y_{i+k,j+l}(\tau_{n-1}) \Bigg), \tag{81}$$

where $\tau_n \triangleq nT + \Delta t$.

6.6 Erosion

The basic erosion operator erodes away the boundaries of black pixels in binary images. Define the rule of the erosion operation as follows [11]:

1. if all cells are black, then the new state of the cell will be black.
2. in any other case, the new state of the cell will be white.

The truth table can be written as follows:

Rule of Erosion

$\sum_{k=-1}^{1}\sum_{m=-1}^{1} y_{i+k,j+m}((n-1)T)$	9	8	7	6	5	4	3	2	1	0
new state of center cell: $y_{ij}(nT)$	1	0	0	0	0	0	0	0	0	0

(82)

where $y_{ij}(nT)$ has two states {0, 1}, and"1" and "0" in the rule table denote "black" and "white", respectively. The memristance and the difference equation which can generate the local rule for erosion are given by

$$M(q) = u\big(\mathit{floor}\big(|q|\big) - 9\big) = \begin{cases} 1, & \text{if } |q| \geq 9, \\ 0, & \text{else}, \end{cases} \tag{83}$$

and

$$y_{ij}(nT + \Delta t) = u\left(\mathit{floor}\left| \sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) + \Delta t \right| - 9\right), \tag{84}$$

respectively, where the weighting coefficient matrix $A = [a_{k,l}]$ is given by

$$A = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}\,. \tag{85}$$

Our computer simulation of the erosion cellular automaton is shown in Fig. 24. Observe that the boundaries of black pixels in binary images are peeled off by this cellular automaton. Since all state variables y_{ij} and weighting coefficients $a_{k,l}$ are non-negative integers, Eq. (84) can be transformed into an equivalent difference equation

$$y_{ij}(\tau_n) = u\left(\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}(\tau_{n-1}) - 9 \right), \tag{86}$$

where $\tau_n \triangleq nT + \Delta t$.

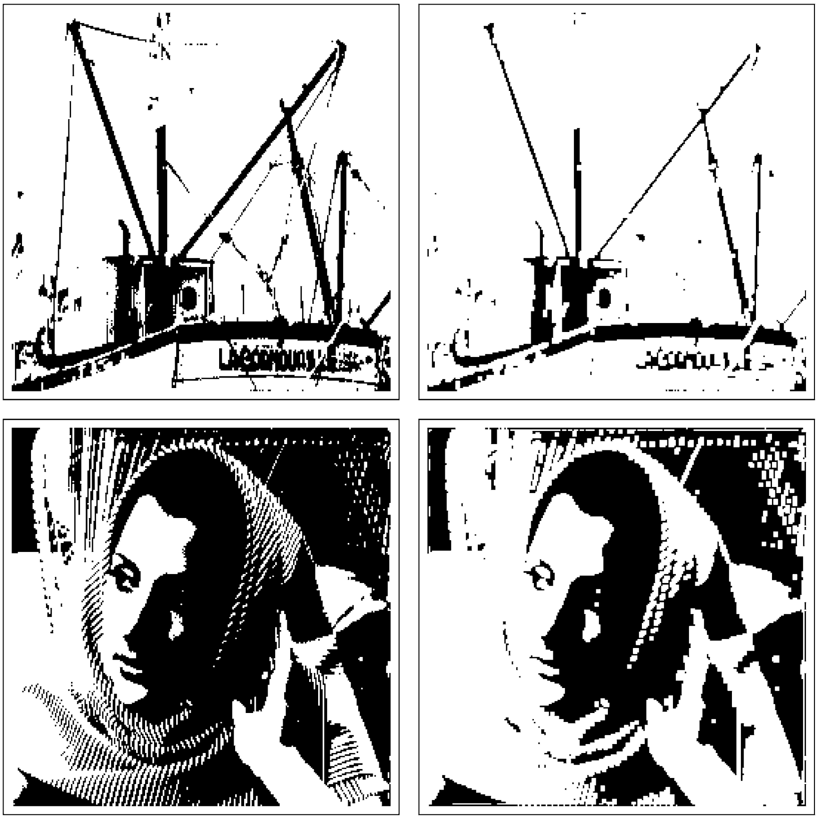


Fig. 24 An erosion memristor cellular automaton. Initial state images and erosion images are illustrated from *left* to *right*

6.7 Dilation

The basic dilation operation enlarges the areas of black pixels at their borders in binary images. Define the rule of totalistic dilation as follows [11]:

1. if all cells are white, then the new state of the cell will be white.
2. in any other case, the new state of the cell will be black.

The truth table can be written as follows:

Rule of Dilation

$\sum_{k=-1}^{1} \sum_{m=-1}^{1} y_{i+k,j+m}((n-1)T)$	9	8	7	6	5	4	3	2	1	0
new state of center cell: $y_{ij}(nT)$	1	1	1	1	1	1	1	1	1	0

(87)

where $y_{ij}(nT)$ has two states {0, 1}, and "1" and "0" in the rule table denote "black" and "white", respectively. The memristance and the difference equation which can generate the local rule for dilation are given by

$$M(q) = u\big(\mathit{floor}\big(|q|\big)\big) = \begin{cases} 1, & \text{if } |q| \geq 1, \\ 0, & \text{else}, \end{cases} \tag{88}$$

and

$$y_{ij}(nT + \Delta t) = u\left(\mathit{floor}\left|\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) + \Delta t\right|\right), \tag{89}$$

respectively, where the weighting coefficient matrix $A = [a_{k,l}]$ is given by

$$A = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}. \tag{90}$$

Our computer simulation of the dilation cellular automaton is shown in Fig. 25. Observe that the areas of black pixels at their borders in binary images are enlarged by this cellular automaton. Since all state variables y_{ij} and weighting coefficients $a_{k,l}$ are non-negative integers, Eq. (89) can be transformed into an equivalent difference equation

$$y_{ij}(\tau_n) = u\left(\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}(\tau_{n-1})\right), \tag{91}$$

where $\tau_n \triangleq nT + \Delta t$.

6.8 Laplacian Memristor Cellular Automaton

Consider the *Laplacian cellular automaton* with eight neighbors [11], whose local rule is given by

Rule of Laplacian Cellular Automaton

w	8	7	$\cdots$	2	1	0	−1	−2	$\cdots$	−7	−8
new state of center cell: $y_{ij}(nT)$	0	0	0	0	0	1	1	1	1	1	1

(92)

where $y_{ij}(nT)$ has two states {0, 1}, "1" and "0" in the rule table denote "black" and "white" in the binary image, respectively, and we set

$$w \triangleq \left(\sum_{k=-1}^{1} \sum_{m=-1}^{1} y_{i+k,j+m}\big((n-1)T\big)\right) - 9y_{i,j}\big((n-1)T\big). \tag{93}$$

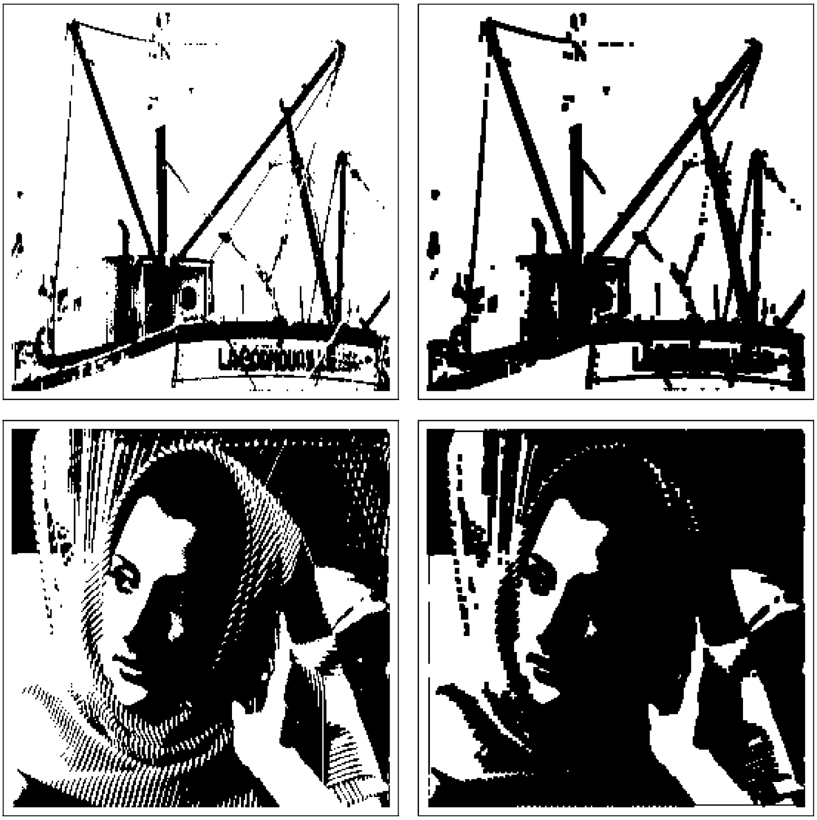


Fig. 25 A dilation memristor cellular automaton. Initial state images and dilation images are illustrated from *left* to *right*

The memristance and the difference equation which can generate the rule for Laplacian cellular automaton are given by

$$M(q) = 1 - u\big(\mathit{floor}(q)\big) = \begin{cases} 1, & \text{if } q < 1, \\ 0, & \text{else}, \end{cases} \tag{94}$$

and

$$y_{ij}(nT + \Delta t) = 1 - u\Bigg(\mathit{floor}\bigg(\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) + \Delta t\bigg)\Bigg), \tag{95}$$

respectively, where the weighting coefficient matrix $A = [a_{k,l}]$ is given by

$$A = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & -8 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}\,. \tag{96}$$

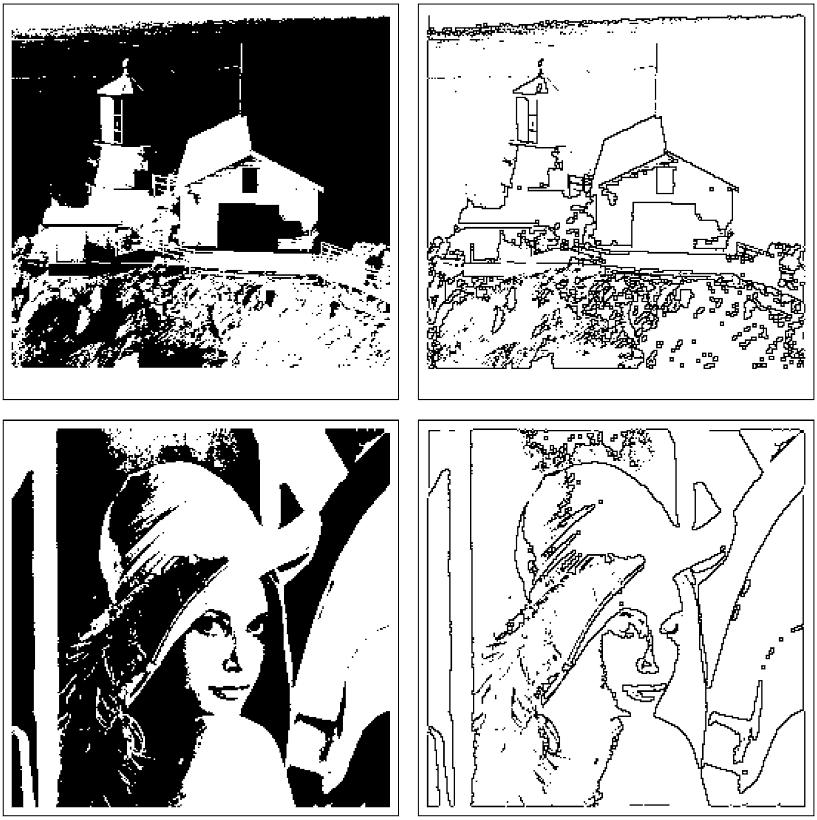


Fig. 26 A Laplacian memristor cellular automaton. Initial state images and processed images are illustrated from *left* to *right*

In this case, the memristance $M(q)$ is not an even function. Our computer simulation of the Laplacian memristor cellular automaton is shown in Fig. 26. Observe that the edges of objects are extracted by this cellular automaton. Note that Eq. (95) can be transformed into an equivalent difference equation

$$y_{ij}(\tau_n) = 1 - u\left(\sum_{k,l\in(-1,0,1)} a_{k,l}\, y_{i+k,j+l}(\tau_{n-1})\right), \tag{97}$$

where $\tau_n \triangleq nT + \Delta t$.

6.9 Sharpening Filter Memristor Cellular Automaton

Sharpening filters are used to enhance the edges of objects. It is well known that weighting coefficient matrix $A = [a_{k,l}]$ for a sharpening filter is given by

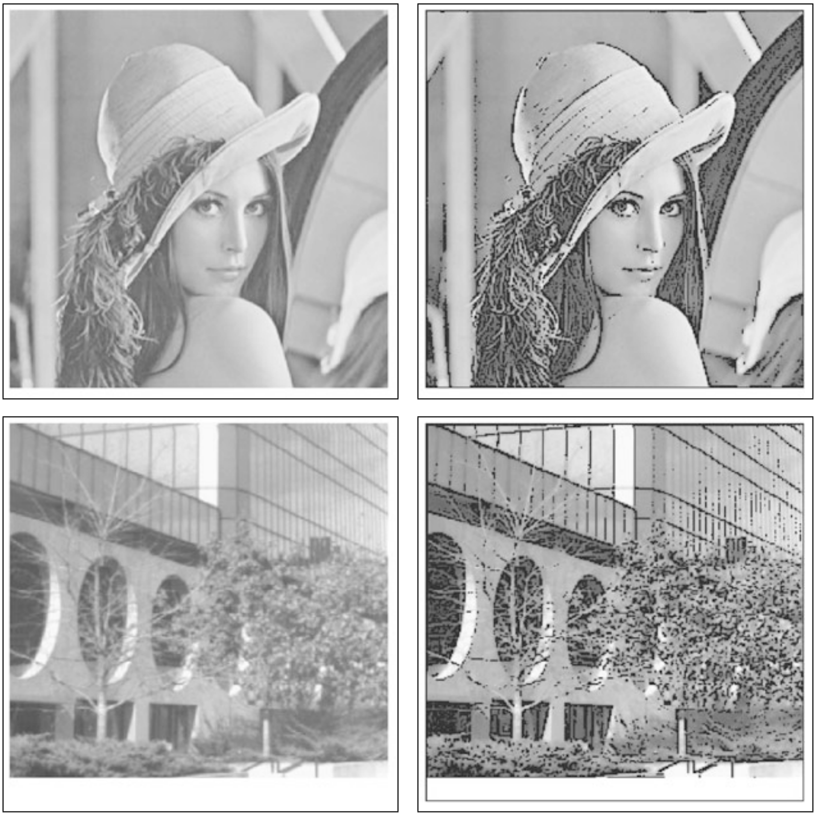


Fig. 27 A sharpening filter memristor cellular automaton. Initial state grayscale images and processed images are illustrated from *left* to *right*. The output binary image is superimposed on a grayscale image

$$A = \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}\,. \tag{98}$$

Consider the difference equation

$$y_{ij}(nT + \Delta t) = u\left(\mathit{floor}\left(\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) - c + \Delta t \right)\right), \tag{99}$$

where c is a constant.

Our computer simulation of the sharpening filter cellular automaton for $c = 30$ is shown in Fig. 27. Observe that the edges of objects are enhanced. In this simulation, the levels of a grayscale image range from 0 (black) to 255 (white), and the output binary image is superimposed on a given grayscale image. Note that Eq. (99) can be transformed into an equivalent difference equation

$$y_{ij}(\tau_n) = u\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}(\tau_{n-1}) - c \Bigg), \tag{100}$$

where $\tau_n \triangleq nT + \Delta t$ and c is an integer.

6.10 *Noise Removal Memristor Cellular Automaton*

Noise removal is a fundamental problem in image analysis. Let us remove the Gaussian noise[3] via the majority rule. The truth table can be written as follows [11]:

Rule of noise removal

$\sum_{k=-1}^{1}\sum_{m=-1}^{1} y_{i+k,j+m}((n-1)T)$	9	8	7	6	5	4	3	2	1	0
new state of center cell: $y_{ij}(nT)$	1	1	1	1	1	0	0	0	0	0

(101)

where $y_{ij}(nT)$ has two states {0, 1}, and “1” and “0” in the rule table denote “black” and “white”, respectively.

The memristance which can generate this rule is given by

$$M(q) = \sum_{5}^{9} M_i(q) = \begin{cases} 1, & \text{if } q = 5, 6, 7, 8, 9, \\ 0, & \text{else,} \end{cases} \tag{102}$$

or

$$M(q) = u\big(\mathit{floor}\big(|q| - 4\big)\big). \tag{103}$$

Thus, the difference equation is given by

$$y_{i,j}(nT + \Delta t) = \sum_{5}^{9} M_i \Bigg(\sum_{k,l \in (-1,0,1)} y_{i+k,j+m}\big((n-1)T\big) + \Delta t \Bigg), \tag{104}$$

or

$$y_{i,j}(nT + \Delta t) = u\Bigg(\mathit{floor}\Bigg(\Bigg| \sum_{k,l \in (-1,0,1)} y_{i+k,j+m}\big((n-1)T\big) + \Delta t \Bigg| - 4 \Bigg)\Bigg), \tag{105}$$

[3] Gaussian noise has a probability density function of the normal distribution (Gaussian distribution).

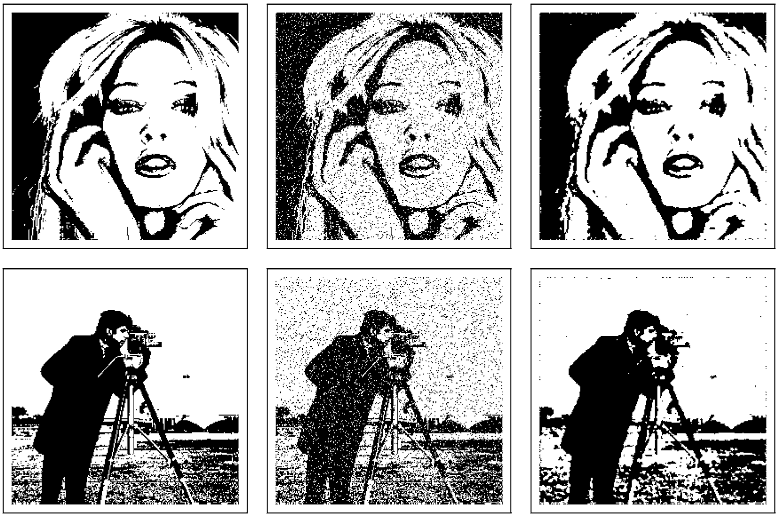


Fig. 28 A noise removal memristor cellular automaton. Binary images, Gaussian noise images (noise level 2 %), and noise removed images are illustrated from *left* to *right*

where $0 < \Delta t < 1$ and the weighting coefficient matrix $A = [a_{k,l}]$ is given by

$$A = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}. \tag{106}$$

Our computer simulation of the noise removal memristor cellular automaton is shown in Fig. 28. Observe that the noise is removed by this cellular automaton.

Since all state variables y_{ij} and weighting coefficients $a_{k,l}$ are non-negative integers, Eqs. (105) and (104) can be transformed into equivalent difference equations

$$y_{ij}(\tau_n) = u\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}(\tau_{n-1}) - 4 \Bigg), \tag{107}$$

and

$$y_{ij}(\tau_n) = \sum_{l=5}^{9} M_l \Bigg(\sum_{k,l \in (-1,0,1)} y_{i+k,j+m}(\tau_{n-1}) \Bigg), \tag{108}$$

respectively, where $\tau_n \triangleq nT + \Delta t$ and $0 < \Delta t < 1$.

6.11 Inverse Half-Toning Memristor Cellular Automaton

Inverse half-toning is used to convert a binary image into a grayscale image. The weighting coefficient matrix $A = [a_{k,l}]$, the memristance $M(q)$, and the difference equation for the inverse half-toning are given by

$$A = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}, \tag{109}$$

$$M(q) = \mathit{floor}\big(28|q|\big), \tag{110}$$

and

$$y_{ij}(nT + \Delta t) = \mathit{floor}\left(28\left|\sum_{k,l\in(-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big)\right| + \Delta t\right), \tag{111}$$

respectively.[4]

Our computer simulation of Eq. (111) is shown in Fig. 29. Observe that given binary images are converted into grayscale images (10-gradations).[5]

Since all state variables y_{ij} and weighting coefficients $a_{k,l}$ are non-negative integers, Eq. (111) can be transformed into an equivalent difference equation

$$y_{ij}(\tau_n) = \mathit{floor}\left(28 \sum_{k,l\in(-1,0,1)} a_{k,l}\, y_{i+k,j+l}(\tau_{n-1})\right), \tag{112}$$

where $\tau_n \triangleq nT + \Delta t$.

7 Memristor Cellular Automaton with Inputs

Consider memristor cellular automata shown in Figs. 30 and 31. Their dynamics are given by

$$\begin{aligned} y_{ij}(nT) = M\Bigg(& \sum_{k,l\in(-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) \\ & + \sum_{k,l\in(-1,0,1)} b_{k,l}\, u_{i+k,j+l}\big((n-1)T\big) + \Delta t\Bigg), \end{aligned} \tag{113}$$

[4] We can obtain Eqs. (110) and (111) using the approximation: $\mathit{floor}(\frac{255}{9}|q|) \approx \mathit{floor}(28|q|)$.

[5] 10-gradations: {0, 28, 56, 84, 112, 140, 168, 196, 224, 252} levels.

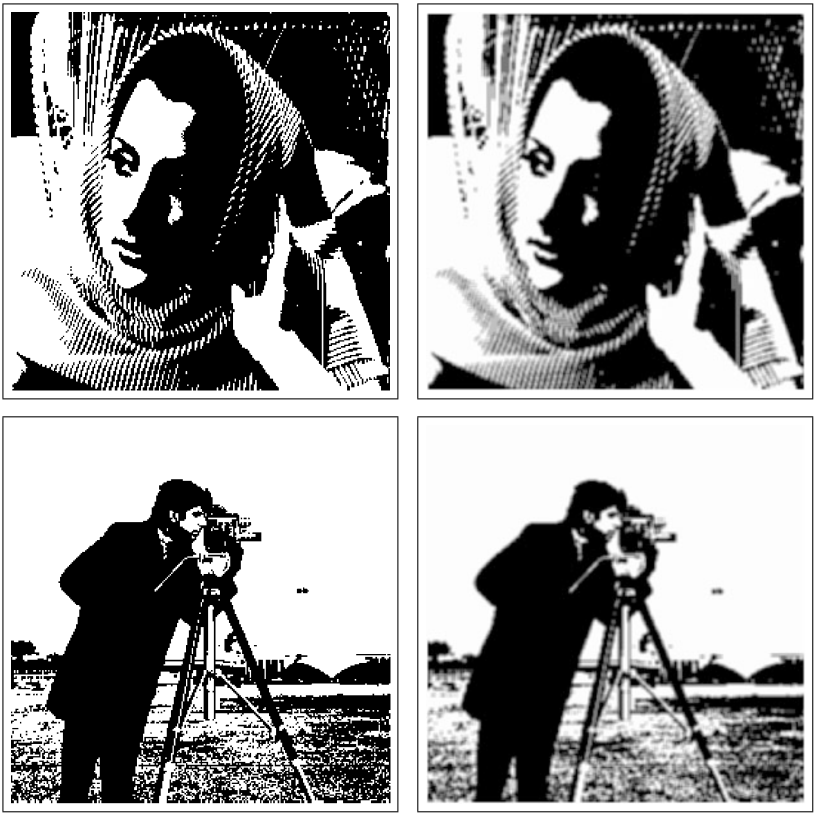


Fig. 29 Inverse half-toning memristor cellular automaton. Initial state binary images (*left*) are converted into 10-gradations grayscale images (*right*)

and

$$y_{ij}(nT) = M\Bigg(\sum_{k,l\in(-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) + \Delta t\Bigg) + \sum_{k,l\in(-1,0,1)} b_{k,l}\, u_{i+k,j+l}\big((n-1)T\big), \tag{114}$$

respectively, where $M(q)$ denotes the memristance. Since they have an input term, Eqs. (113) and (114) have a more generalized form than Eq. (47).

Consider next Eq. (49) with a *random binary noise input* $r(\tau_{n-1})$:

$$y_{ij}(\tau_n) = u\Bigg(\sum_{k,l\in(-1,0,1)} a_{k,l}\, y_{i+k,j+l}(\tau_{n-1}) + r(\tau_{n-1})\Bigg)$$

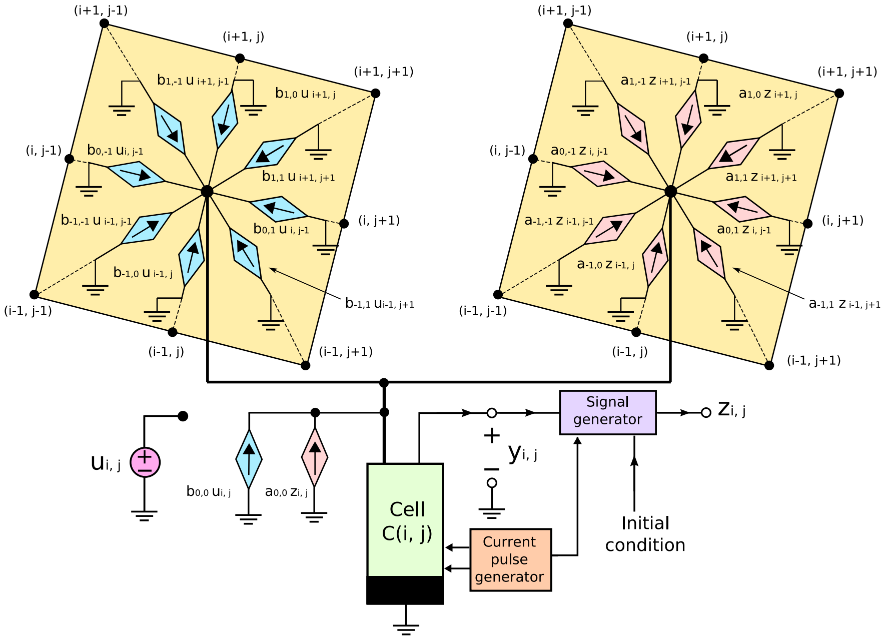


Fig. 30 A memristor cellular automaton, which consists of a memristor cell $C(i, j)$ (*light green*), a current pulse generator (*salmon pink*), a signal generator (*medium purple*), and voltage-controlled current sources for inputs (*light blue*) and outputs (*pink*)

$$-u\Bigg(\sum_{k,l\in(-1,0,1)} a_{k,l}y_{i+k,j+l}\;(\tau_{n-1}) + r(\tau_{n-1}) - 6\Bigg), \tag{115}$$

where $r(\tau_{n-1}) \in \{0, 1\}$. The above equation may be considered to be a special case of Eq. (113), if we set

$$M(q) = \big(\mathit{floor}(|q|)\big) - u\big(\mathit{floor}(|q|) - 6\big), \tag{116}$$

and

$$b_{k,l}u_{i+k,j+l}\big((n-1)T\big) = \begin{cases} r((n-1)T) & \text{if } (k,l) = (0,0), \\ 0 & \text{else.} \end{cases} \tag{117}$$

Observe that it has complex behavior as shown in Fig. 32.

8 Memristor Discrete-Time Cellular Neural Network

Consider the memristor cell shown in Fig. 33. The dynamics of the cell $C(i, j)$ is given by

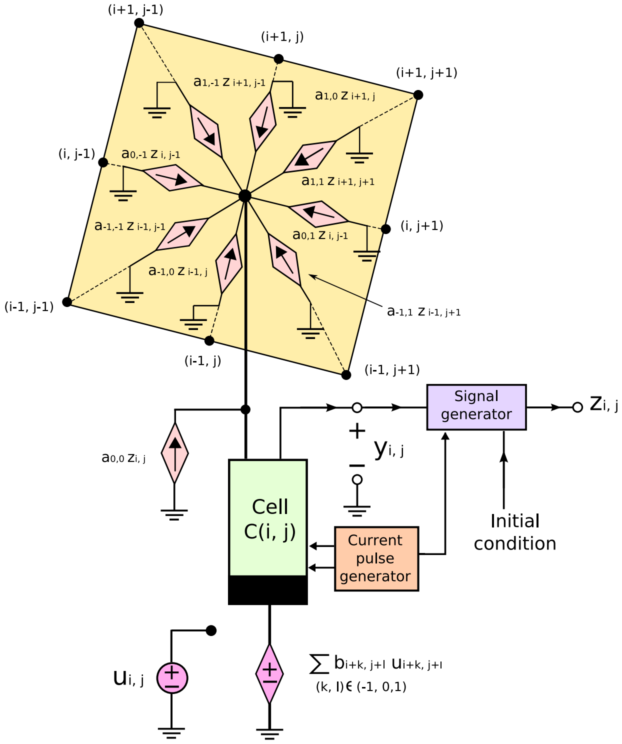
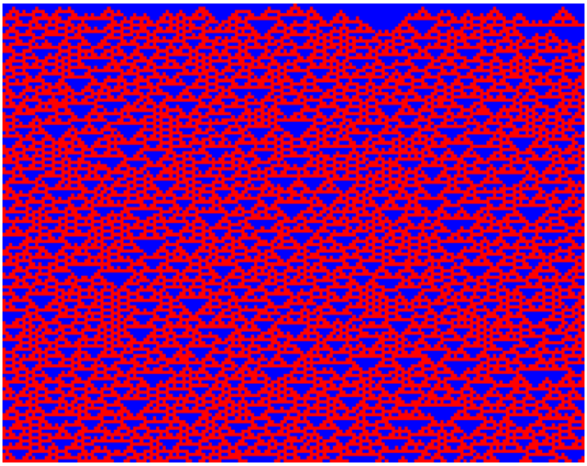


Fig. 31 A memristor cellular automaton, which consists of a memristor cell $C(i, j)$ (*light green*), a current pulse generator (*salmon pink*), a signal generator (*medium purple*), a voltage-controlled voltage source for inputs (*violet*), and voltage-controlled current sources for outputs (*pink*)

Fig. 32 Evolution of a rule 126 memristor cellular automaton with a random binary noise



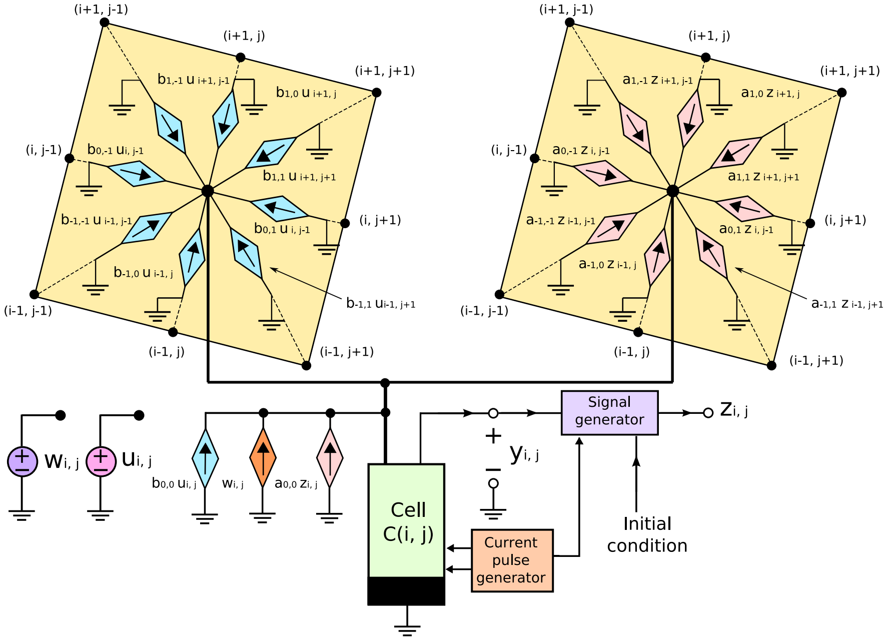


Fig. 33 A memristor discrete-time cellular neural network (DTCNN), which consists of a memristor cell $C(i, j)$ (*light green*), a current pulse generator (*salmon pink*), a signal generator (*medium purple*), and voltage-controlled current sources for inputs (*light blue*), outputs (*pink*), and a threshold (*chocolate*)

$$y_{ij}(nT) = M\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l} y_{i+k,j+l}\big((n-1)T\big) + \sum_{k,l \in (-1,0,1)} b_{k,l} u_{i+k,j+l}\big((n-1)T\big) + w_{ij} + \Delta t \Bigg), \tag{118}$$

where $0 < \Delta t < 1$.

In this cell, a new voltage-controlled current source for a threshold w_{ij} is connected to the cell $C(i, j)$ of Fig. 31. Thus, the difference between Eq. (113) and Eq. (118) is a threshold parameter w_{ij}. If we assume that

$$M(q) = u\big(floor(q)\big), \tag{119}$$

and all state variables y_{ij} and weighting coefficients $a_{k,l}$, $b_{k,l}$, and w_{ij} are integers, we would obtain the equation of a *memristor discrete-time cellular neural network* (Memristor DTCNN)

$$y_{ij}(\tau_n) = u\Bigg(\sum_{k,l \in (-1,0,1)} a_{k,l} y_{i+k,j+l}(\tau_{n-1}) + \sum_{k,l \in (-1,0,1)} b_{k,l} u_{i+k,j+l}(\tau_{n-1}) + w_{ij} \Bigg), \tag{120}$$

where $a_{k,l}$, $b_{k,l}$, and w_{ij} denote the feedback, control, and threshold template parameters, respectively. The matrices $A = [a_{k,l}]$ and $B = [b_{k,l}]$ are referred to as the feedback template A and the feedforward (input) template B, respectively. If we assume that w_{ij} is the same for the whole network, the template $\{A, B, w\}$ is fully specified by 19 parameters, which are the elements of two 3×3 matrices A, B, and a real number w, namely

$$A = \begin{array}{|c|c|c|}\hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array}\,, \qquad B = \begin{array}{|c|c|c|}\hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array}\,, \qquad w = \boxed{const.} \tag{121}$$

We remark that CNN and DTCNN can have the output state 1 or -1, however, the *memristor DTCNN* (120) have the output state 1 or 0. Thus, we may have to modify or adjust CNN templates, which are designed for the output state 1 or -1 [2, 5, 8–10].

We next show the relationship between the memristor DTCNN and the memristor cellular automata. The dynamics of DTCNN was given by

$$\begin{aligned} y_{ij}(nT) = M\Bigg(& \sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) \\ & + \sum_{k,l \in (-1,0,1)} b_{k,l}\, u_{i+k,j+l}\big((n-1)T\big) + w_{ij} + \Delta t \Bigg), \end{aligned} \tag{122}$$

If we set $w_{ij} = 0$, then we obtain the dynamics of the memristor cellular automata

$$\begin{aligned} y_{ij}(nT) = M\Bigg(& \sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) \\ & + \sum_{k,l \in (-1,0,1)} b_{k,l}\, u_{i+k,j+l}\big((n-1)T\big) + \Delta t \Bigg), \end{aligned} \tag{123}$$

Thus, the memristor DTCNN can perform all applications of the memristor cellular automata by modifying the memristance $M(q)$.

8.1 Dilation

The dilation operation enlarges the areas of black pixels at their borders in binary images. Define the template of the dilation operation as follows:

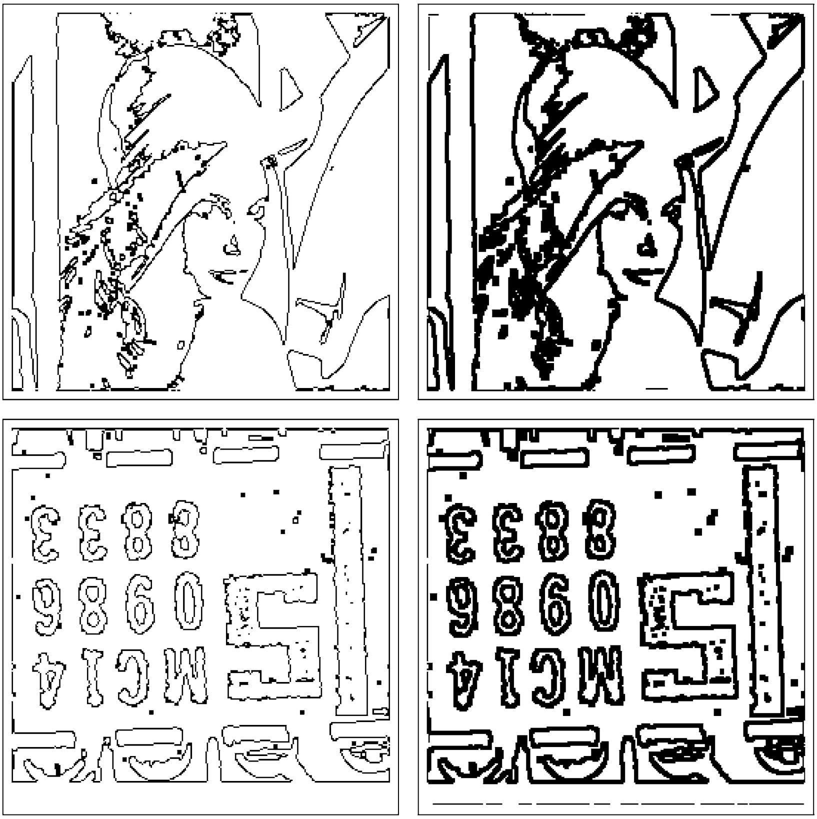


Fig. 34 Dilation memristor DTCNN. Boundaries of black pixels are enlarged. Input images and output images are illustrated from *left* to *right*

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}\,, \qquad B = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}\,, \qquad w = \boxed{0}\,. \tag{124}$$

Our computer simulation of the dilation operation is shown in Fig. 34. Observe that the boundaries of black pixels in binary images are enlarged. Compare the two templates (90) and (124), the two memristances (88) and (119), and Figs. 25 and 34.

8.2 Erosion

The erosion operator erodes away the boundaries of black pixels in binary images. Define the template of the erosion operation as follows:

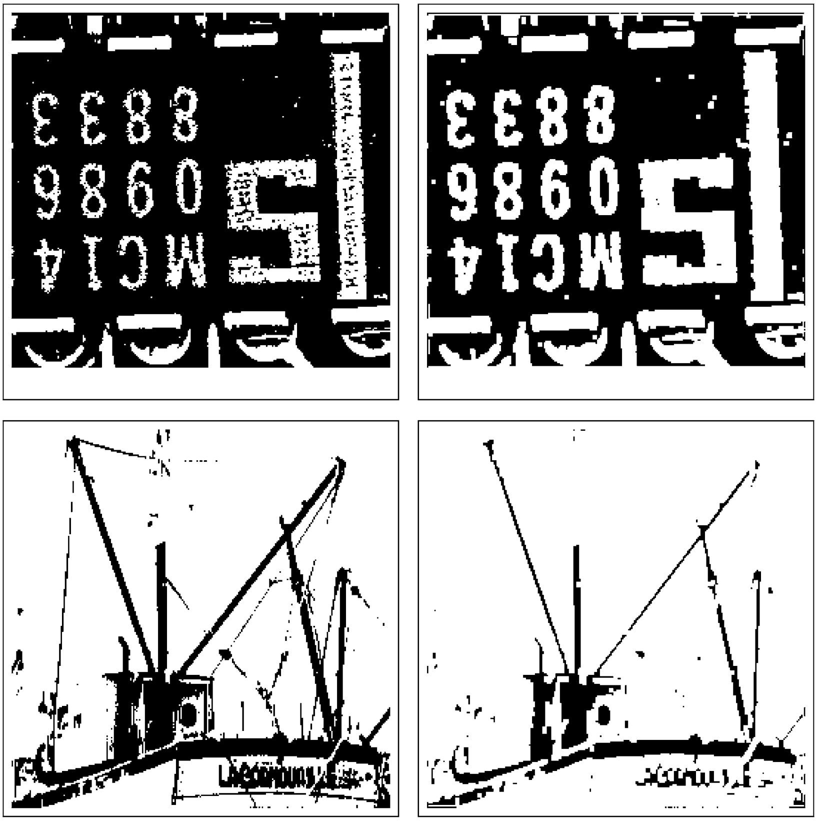


Fig. 35 Erosion memristor DTCNN. Boundaries of black pixels are peeled off. Input images and output images are illustrated from *left* to *right*

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}\,, \qquad B = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}\,, \qquad w = \boxed{-8}\,. \tag{125}$$

Our computer simulation of the erosion operation is shown in Fig. 35. Observe that the boundaries of black pixels in binary images are peeled off. Compare the two templates (85) and (125), the two memristances (83) and (119), and Figs. 24 and 35.

8.3 Edge Detection

The edge detection operator extracts edges of objects in a given binary image. Define the template of the edge operation as follows:

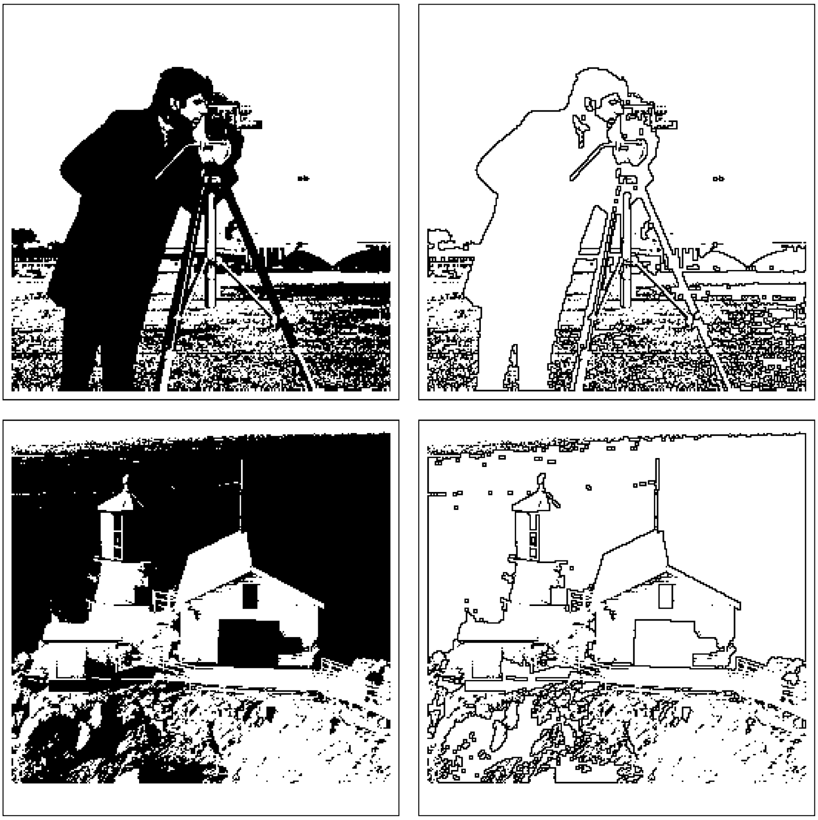


Fig. 36 Edges detection memristor DTCNN. Edge of objects are extracted. Input images and output images are illustrated from *left* to *right*

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & -8 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}, \qquad w = \boxed{0}. \tag{126}$$

Our computer simulation of the edge detection is shown in Fig. 36. Observe that edges of objects in a binary image are detected. Compare the two templates (78) and (126), the two memristances (76) and (119), and Figs. 23 and 36.

8.4 Right Edge Detection

The right edge detection operator extracts right edges of all binary objects which are at least two-pixels wide along the horizontal direction in a given binary image.

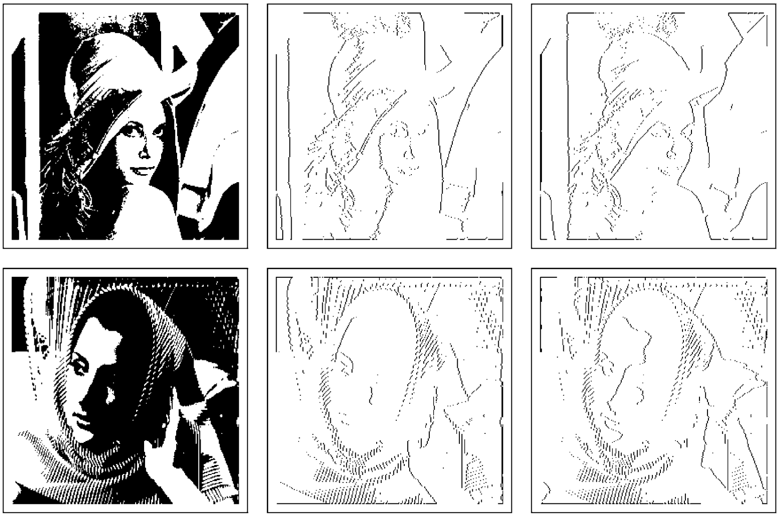


Fig. 37 *Right* and *left edges* detection memristor DTCNN. *Right edges* and *left edges* in binary images are detected. Input images, output images of *right edge* detection, and output images of *left edge* detection are illustrated from *left* to *right*

Define the template of the right edge operation as follows:

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & 1 & -1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{-1}. \tag{127}$$

Similarly, we can define the template of the right edge detection as follows:

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline -1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{-1}. \tag{128}$$

Our computer simulation of the right edge and left edge detection is shown in Fig. 37. Observe that right and left edges in binary images are detected.

8.5 *Face-Vase Illusion*

The face-vase illusion operator simulates the well-known visual illusion where the input image is perceived either as two symmetric faces, or as a vase, depending on the initial thought or attention, which is simulated by specifying a small patch of black pixels inside the object to be picked out. Define the template of the face-vase

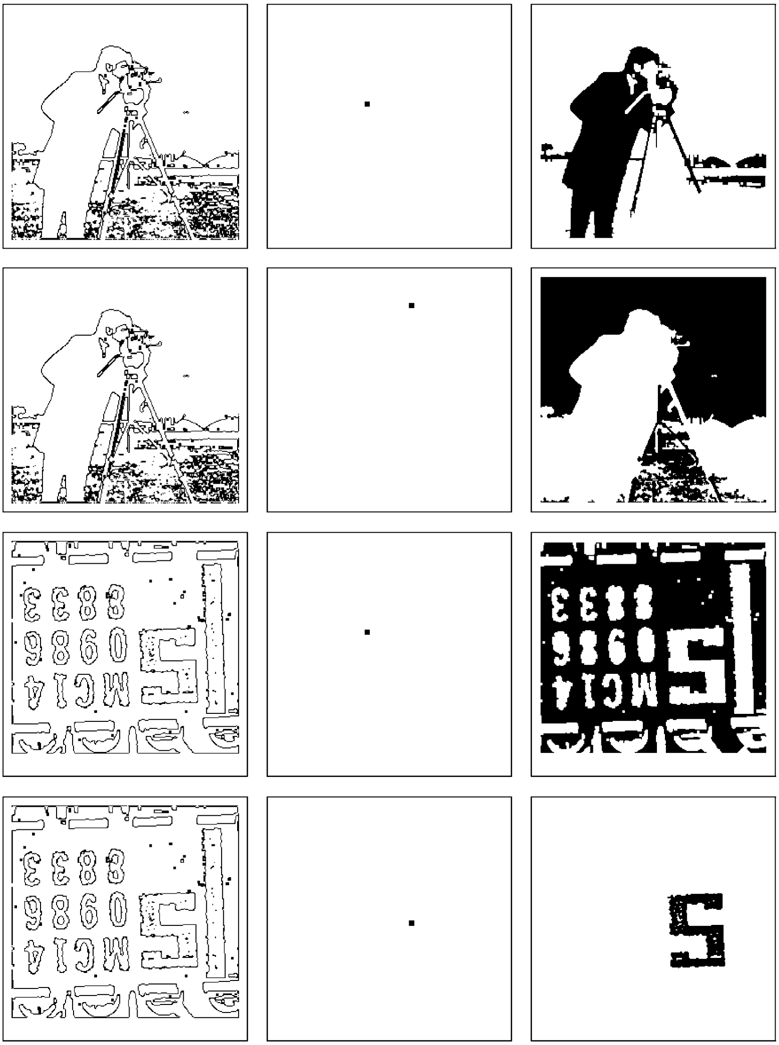


Fig. 38 Face-vase illusion memristor DTCNN. Complicated region is picked out by specifying a small patch of black pixels inside the object. Input images, initial states and output images are illustrated from *left* to *right*

illusion as follows:

$$A = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}\,, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & -6 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}\,, \qquad w = \boxed{0}\,. \tag{129}$$

Our computer simulation of the face-vase illusion is shown in Fig. 38. Observe that the complicated region is picked out by specifying a small patch of black pixels inside the object.

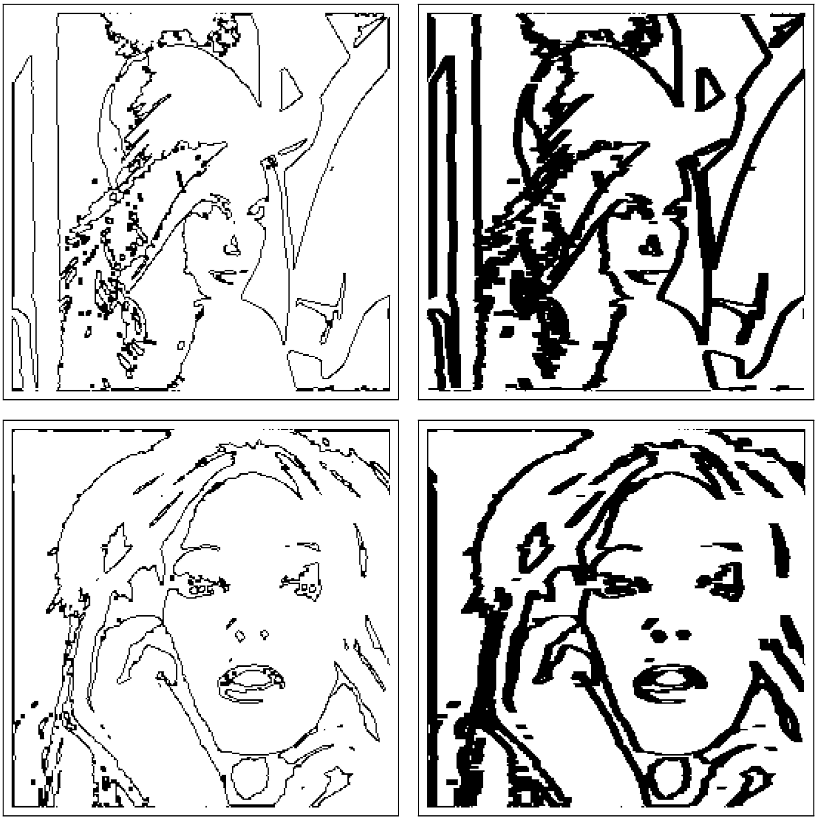


Fig. 39 Shadow projection memristor DTCNN. Shadow of all objects in a binary image is projected onto the *left* from the *right*. Input images and output images are illustrated from *left* to *right*

8.6 Shadow Projection

The shadow projection operator projects onto the left the shadow of all objects in a binary image from the right. Define the template of the shadow projection as follows:

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{0}. \tag{130}$$

Our computer simulation of the shadow projection is shown in Fig. 39. Observe that the shadow of all objects in a binary image is projected onto the left from the right.

8.7 Line Detection

Line detection is a fundamental problem in image analysis. Define the template of the horizontal line and vertical line detections as follows:

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline 2 & 2 & 2 \\ \hline -1 & -1 & -1 \\ \hline \end{array}, \qquad w = \boxed{0}. \tag{131}$$

and

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline -1 & 2 & -1 \\ \hline -1 & 2 & -1 \\ \hline -1 & 2 & -1 \\ \hline \end{array}, \qquad w = \boxed{0}. \tag{132}$$

respectively.

Our computer simulation of the line detection is shown in Fig. 40. Observe that the horizontal and vertical lines are detected.

8.8 Selected Objects Extraction

Selected objects extraction operator extracts an object marked by a binary input image. Define the template of the selected objects extraction as follows:

$$A = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 8 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{-8}. \tag{133}$$

Our computer simulation of the selected objects extraction is shown in Fig. 41. Observe that some selected objects marked by a rectangle are extracted.

8.9 Filled Contour Extraction

Filled contour extraction operator extracts an object which contains a boundary and which is marked by a binary input image. Furthermore, they are completely filled inside the interior of closed curves. Define the template of the selected objects extraction as follows:

$$A = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{-3}. \tag{134}$$

Our computer simulation of the filled contour extraction is shown in Fig. 42. Observe that all objects marked by a rectangle are extracted and the interior of all closed curves are filled with black pixels.

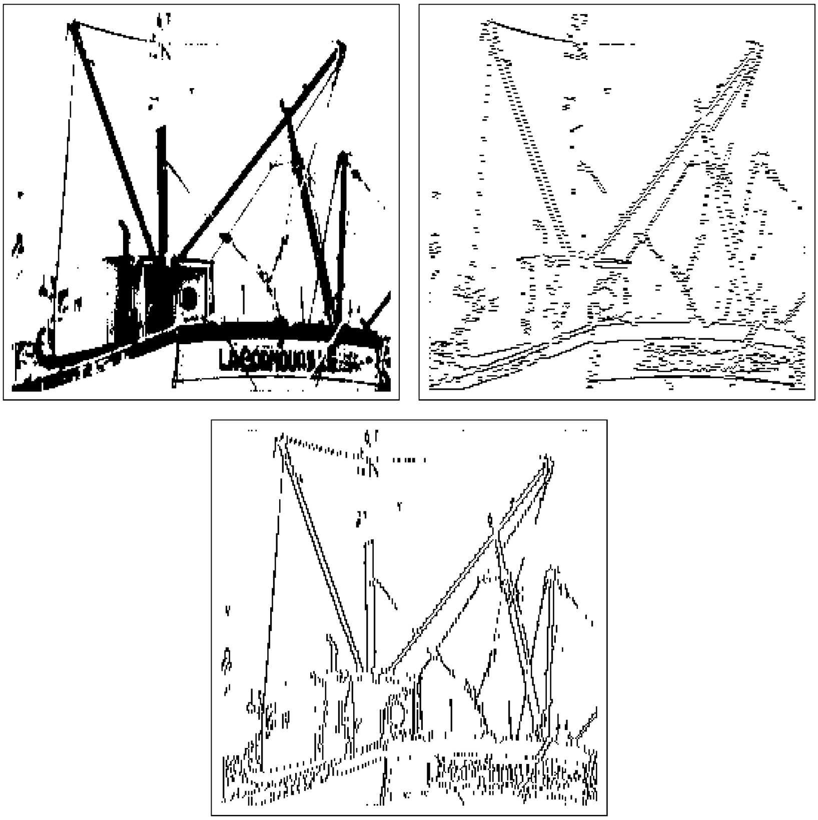


Fig. 40 Line detection memristor DTCNN. An initial state, a detected horizontal line image, and a detected vertical line image are illustrated from *left* to *right*

8.10 Horizontal Hole and Vertical Hole Detection

Horizontal (resp. vertical) hole detection operator detects horizontal (resp. vertical) holes. Define the template of the horizontal hole detection and vertical hole detection as follows:

$$A = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 2 & 2 & -3 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{0}. \tag{135}$$

and

$$A = \begin{array}{|c|c|c|}\hline 0 & 2 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & -3 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{0}. \tag{136}$$

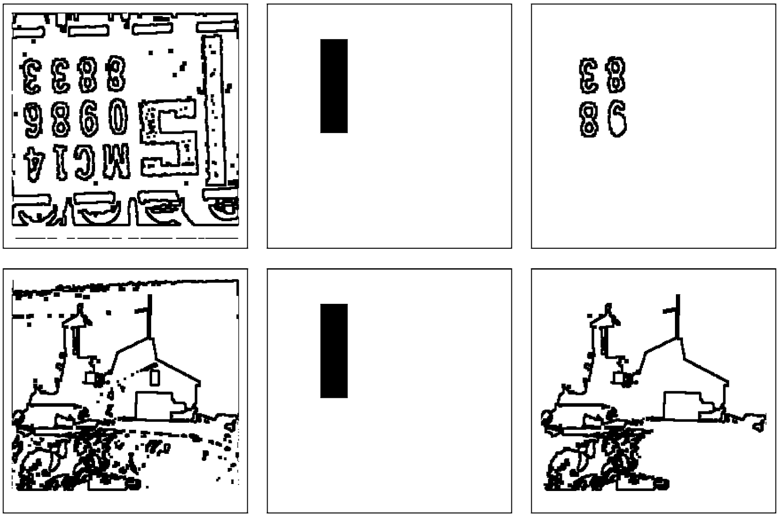


Fig. 41 Selected objects extraction memristor DTCNN. Some selected objects marked by a rectangle are extracted. Input images, initial state images, and output images are illustrated from *left* to *right*

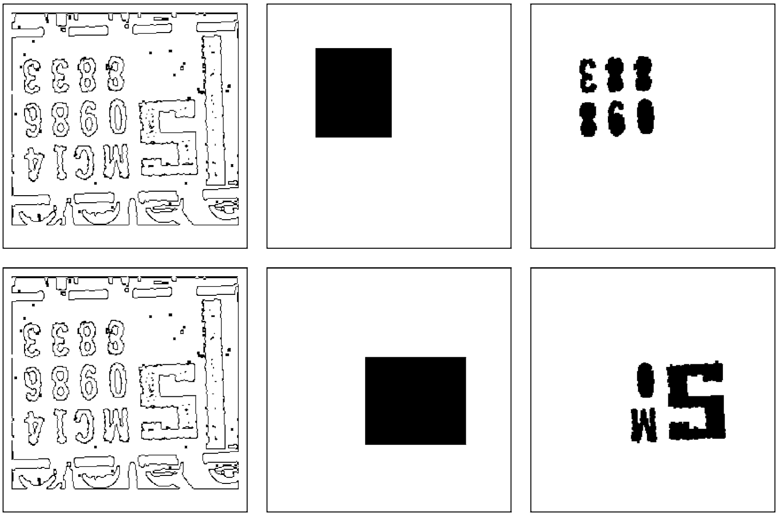


Fig. 42 Filled contour extraction memristor DTCNN. The regions which contain a boundary and which are marked by rectangles are extracted, while completely filling the interior of all closed curves. Input images, initial state images and output images are illustrated from *left* to *right*

Our computer simulation of the horizontal hole and the vertical hole detection is shown in Fig. 43. Observe that the horizontal holes and the vertical holes are detected by these templates.

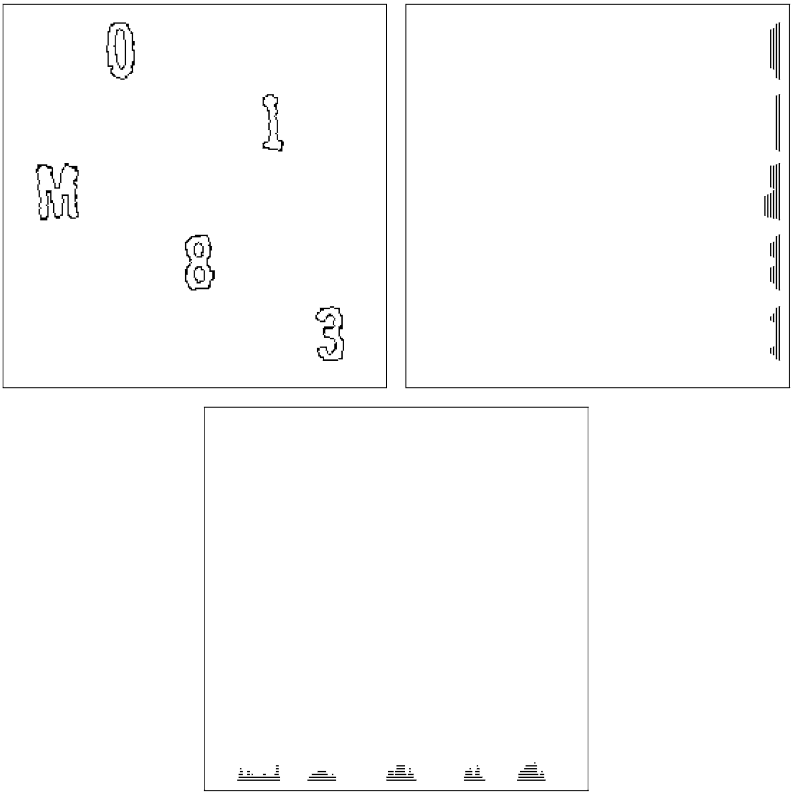


Fig. 43 A horizontal hole and vertical hole detection memristor DTCNN. The number of vertical lines (resp. horizontal lines) in the output images indicates the number of horizontal holes (resp. vertical holes). An initial state and output images for a horizontal hole and vertical hole detection are illustrated from *left* to *right*

9 Advanced Memristor DTCNN

Consider the memristor DTCNN cell shown in Figs. 44 and 45. If we apply a positive current pulse I_p with a height of 1 and a width of 1 into the memristor, we would obtain

$$\left.\begin{array}{l} v_1(t) = M_x(q)I_p(t) = M_x(q(t)) \times 1 = M_x(q(t)) = x_{ij}(t), \\ v_2(t) = M_y(q)I_p(t) = M_y(q(t)) \times 1 = M_y(q(t)) = y_{ij}(t), \end{array}\right\} \tag{137}$$

A signal generator provides the two signals $z_{ij}(t)$ and $p_{ij}(t)$ satisfying the equations

$$z_{ij}(t) = y_{ij}(t - T) - y_{ij}(t - 2T + 1), \tag{138}$$

and

$$p_{ij}(t) = x_{ij}(t - T) - x_{ij}(t - 2T + 1), \tag{139}$$

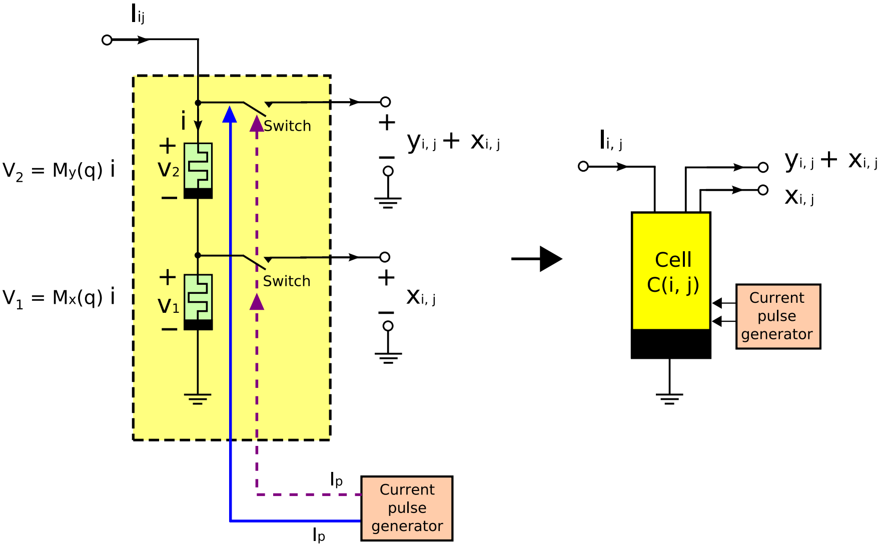


Fig. 44 A memristor cell $C(i, j)$, which consists of two memristors and a switch (*top*). It has two outputs x_{ij} and $y_{ij} + x_{ij}$. A current pulse generator provides a *current pulse wave* (I_p) to memristors and switches. Switches are turned on when (and only when) a positive current pulse is applied to the cell

respectively. If we assume that the memristance $M_x(q)$ satisfies

$$M_x(q) = \mathit{floor}\big(|q|\big), \tag{140}$$

the charge $q_{ij}(t)$ stored in the memristor during the period $[nT, nT + \Delta t]$ $(n = 1, 2, \ldots)$ is given by

$$\begin{aligned} q_{ij}(nT + \Delta t) = cx_{ij}\big((n-1)T\big) + \sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}\big((n-1)T\big) \\ + \sum_{k,l \in (-1,0,1)} b_{k,l}\, u_{i+k,j+l}\big((n-1)T\big) + w_{ij} + \Delta t, \end{aligned} \tag{141}$$

where $0 < \Delta t < 1$ and

$$\left.\begin{aligned} x_{ij}(nT) &= M_x(q_{ij}(nT)) = \mathit{floor}(|q_{ij}(nT)|), \\ y_{ij}(nT) &= M_y(q_{ij}(nT)). \end{aligned}\right\} \tag{142}$$

The last term Δt in Eq. (141) indicates the charge stored by a positive current read pulse[6] (see Fig. 6). The characteristic of the memristance $M_y(q)$ will be given later.

[6] During the period $[nT, nT + \Delta t]$, a positive current read pulse I_p with a height of 1 is applied into the memristor.

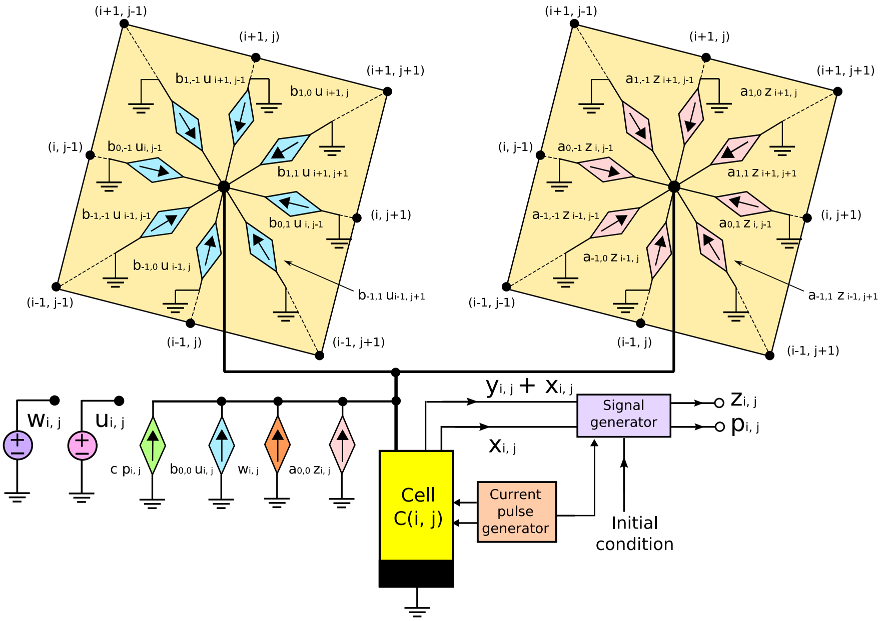


Fig. 45 A memristor DTCNN cell, which consists of a memristor cell $C(i, j)$ (*yellow*), a current pulse generator (*salmon pink*), a signal generator (*medium purple*), and voltage-controlled current sources for inputs (*light blue*), a cell state (*light green*), cell outputs (*pink*), and a threshold (*chocolate*)

For the sake of simplicity, assume that all state variables x_{ij}, y_{ij} and weighting coefficients a_{kl}, b_{kl}, w_{ij}, and c are non-negative integers. Then, by applying Eq. (142) to Eq. (141), we obtain the equation

$$\begin{aligned} x_{ij}(nT+\Delta t) &= \mathit{floor}\big(\big|q_{ij}(nT+\Delta t)\big|\big) \\ &= \mathit{floor}\Bigg(\Bigg|cx_{ij}\big((n-1)T\big) + \sum_{k,l\in(-1,0,1)} a_{k,l}y_{i+k,j+l}\big((n-1)T\big) \\ &\quad + \sum_{k,l\in(-1,0,1)} b_{k,l}u_{i+k,j+l}\big((n-1)T\big) + w_{ij} + \Delta t\Bigg|\Bigg) \\ &= cx_{ij}\big((n-1)T\big) + \sum_{k,l\in(-1,0,1)} a_{k,l}y_{i+k,j+l}\big((n-1)T\big) \\ &\quad + \sum_{k,l\in(-1,0,1)} b_{k,l}u_{i+k,j+l}\big((n-1)T\big) + w_{ij}, \end{aligned} \tag{143}$$

which is equivalent to a DTCNN equation

$$
\begin{aligned}
x_{ij}(\tau_n) = c x_{ij}(\tau_{n-1}) + \sum_{k,l \in (-1,0,1)} a_{k,l}\, y_{i+k,j+l}(\tau_{n-1}) \\
+ \sum_{k,l \in (-1,0,1)} b_{k,l}\, u_{i+k,j+l}(\tau_{n-1}) + w_{ij},
\end{aligned}
\tag{144}
$$

where $\tau_n \triangleq nT + \Delta t$. If we assume that w_{ij} is the same for the whole network, the template $\{A, B, w, c\}$ is fully specified by 20 parameters, which are the elements of two 3×3 matrices A, B, and real numbers w, c, namely,

$$
\begin{aligned}
A = \begin{array}{|c|c|c|} \hline a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ \hline a_{0,-1} & a_{0,0} & a_{0,1} \\ \hline a_{1,-1} & a_{1,0} & a_{1,1} \\ \hline \end{array}, \qquad
B = \begin{array}{|c|c|c|} \hline b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ \hline b_{0,-1} & b_{0,0} & b_{0,1} \\ \hline b_{1,-1} & b_{1,0} & b_{1,1} \\ \hline \end{array}, \\
w = \boxed{w_0}, \qquad c = \boxed{c_0}.
\end{aligned}
\tag{145}
$$

where w_0 and c_0 are constants.

9.1 Sandpile Cellular Automaton

The *sandpile* model is defined on a grid. Each grid point is associated with the height of a sandpile, which is defined below a limiting value. With each time interval, the height at one of the points increases. If a height exceeds the limiting value, the sand must be moved to nearby points until the height at all points are once again below the limiting value. The random placement of sand at a particular point may have no effect, or it may cause a cascading reaction that will affect every point on the grid.[7]

Let us define the rule of sandpile cellular automaton as follows:

1. Each cell has an integer state chosen between 0 and 7 for a von Neumann neighbors (a diamond-shaped neighborhood).
2. The number of neighbors having states larger than 3 is added to the middle cell.
3. In case the center cell is itself in a state larger than 3, the new state is reduced by 4.

The truth table of the sandpile can be written as follows:

Rule of Sandpile

$x_{ij}(\tau_{n-1})$	7	6	5	4	3	2	1	0
$x_{ij}(\tau_n)$	$\xi + 3$	$\xi + 2$	$\xi + 1$	w	$\xi + 3$	$\xi + 2$	$\xi + 1$	ξ

(146)

where ξ denotes the number of neighbors which have states larger than 3.

[7] For more details, see *"Bak-Tang-Wiesenfeld sandpile"* in Wikipedia, the free encyclopedia.

The above rule can be generated by the *difference equation* [11]

$$\begin{aligned} x_{i,j}(\tau_n) = x_{i,j}(\tau_{n-1}) + y_{i-1,j}(\tau_{n-1}) + y_{i,j-1}(\tau_{n-1}) \\ + y_{i,j+1}(\tau_{n-1}) + y_{i+1,j}(\tau_{n-1}) - 4y_{i,j}(\tau_{n-1}), \end{aligned} \tag{147}$$

where

$$\left. \begin{aligned} & y_{ij} = M_y(x_{ij}), \\ & M_y(q) = \sum_{i=4}^{7} M_i(q) = \begin{cases} 0 & \text{if } q = 1, 2, 3, \\ 1 & \text{if } q = 4, 5, 6, 7. \end{cases} \end{aligned} \right\} \tag{148}$$

Thus, the template of the sandpile DTCNN is given by

$$A = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{0}, \qquad c = \boxed{1}. \tag{149}$$

The evolution of the sandpile DTCNN is illustrated in Fig. 46, in which we used the following palette for the state of cells.

Color palette

state	0	1	2	3	4	5	6	7
color	white	orange	yellow	red	blue	blue	blue	blue

(150)

9.2 Game of Life

Game of Life is a two-dimensional cellular automaton. Each cell has one of two possible states, *live* (cell value 1) or *dead* (cell value 0). Each cell interacts with eight neighbors. The Game of Life proceeds according to the rule [6]:

1. *Survivals.* Any live cell with two or three live neighbors survives in the next generation.
2. *Deaths.* Any live cell with four or more live neighbors dies (is removed) from overpopulation. Every cell with one neighbor or none dies from isolation.
3. *Births.* Any dead cell with exactly three live neighbors comes to life.

The truth table defining the Game of Life is as follows:

Game of Life

sum of live neighbors	8	7	6	5	4	3	2	1	0
new state of center cell: $x_{ij}(t+1)$	0	0	0	0	0	1	$*$	0	0

(151)

The symbol $*$ means no change in the next generation. The rules of the Game of Life can be generated from the difference equation [11]

$$x_{ij}(t+1) = M_3(\xi) + M_2(\xi)x_{ij}(t), \tag{152}$$

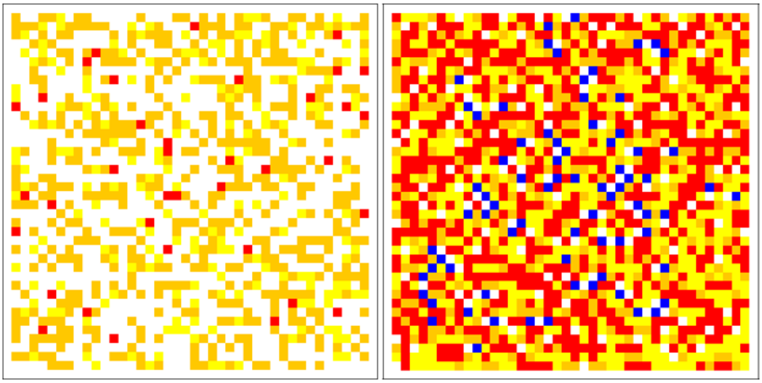


Fig. 46 Evolution of the sandpile DTCNN for $t = 100$ (*left*) and $t = 902$ (*right*) from a random initial pattern. A cascade process sets in (*blue points* spread on the grid) until all state values are below the threshold (*right*)

where

$$\xi \triangleq \left(\sum_{k=-1}^{1} \sum_{m=-1}^{1} x_{i+k,j+m}(t) \right) - x_{ij}(t). \tag{153}$$

Consider next the memristor cell in Fig. 47. Define the memristance

$$\left. \begin{aligned} M_a &= M_x(q) = \mathit{floor}(|q|), \\ M_b &= M_2(q), \\ M_c &= M_3(q). \end{aligned} \right\} \tag{154}$$

If we apply a positive current pulse I_p with a height of 1 and a width of 1 into the memristor, we would obtain

$$\left. \begin{aligned} v_1 &= M_a(q) I_p = M_x(q) = x_{ij}, \\ v_2 &= M_b(q) I_p = M_2(q) = y_{ij}, \\ v_3 &= M_c(q) I_p = M_3(q) = r_{ij}, \end{aligned} \right\} \tag{155}$$

where v_1, v_2, v_3, $M_a(q)$, $M_b(q)$, and $M_c(q)$ are illustrated in Fig. 47.

Thus, Eq. (152) can be realized by replacing the memristor cell in Fig. 44 with the one in Fig. 47, and providing the following signal from the signal generator

$$\begin{aligned} z_{ij}(t) &= M_3(q(t-T)) + M_2(q(t-T)) x_{ij}(t-T) \\ &\quad - M_3(q(t-2T+1)) - M_2(q(t-2T+1)) x_{ij}(t-2T+1) \\ &= r_{ij}(t-T) + y_{ij}(t-T) x_{ij}(t-T) \\ &\quad - r_{ij}(t-2T+1) - y_{ij}(t-2T+1) x_{ij}(t-2T+1). \end{aligned} \tag{156}$$

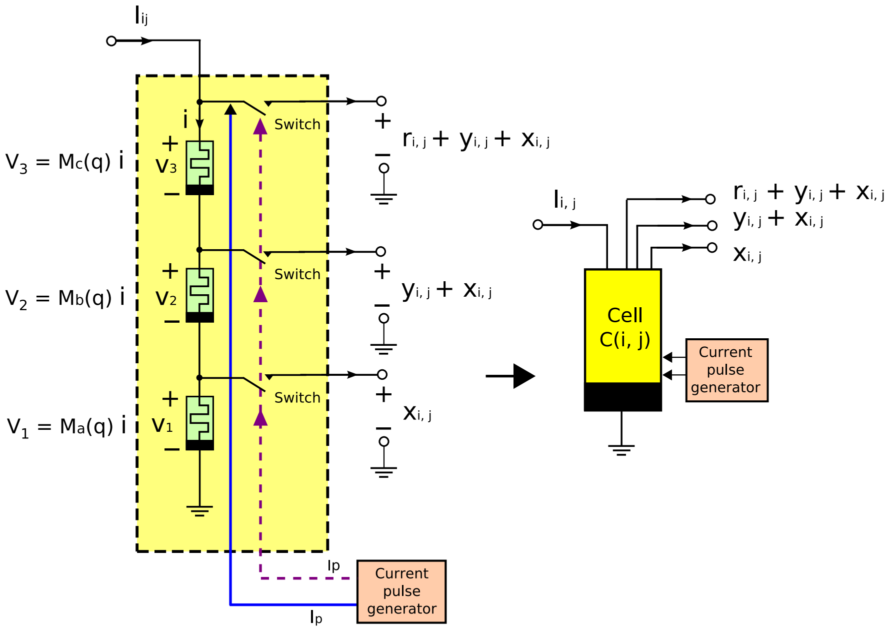


Fig. 47 A memristor cell $C(i, j)$, which consists of three memristors and a switch (*top*). It has three outputs x_{ij}, $y_{ij} + x_{ij}$, and $r_{ij} + y_{ij} + x_{ij}$. A current pulse generator provides a *current pulse wave* (I_p) to memristors and switches. Switches are turned on when (and only when) a positive current pulse is applied to the cell

In this case, the template of the Game of Life DTCNN is given by

$$A = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 0 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{0}, \qquad c = \boxed{0}. \tag{157}$$

Our computer simulation of a Game of Life is shown in Fig. 48.

9.3 Multitasking Capability

The basic erosion operator erodes away the boundaries of black pixels in binary images. The truth table can be written as follows [11]:

Rule of Totalistic Erosion

$\sum_{k=-1}^{1} \sum_{m=-1}^{1} x_{i+k,j+m}(\tau_{n-1})$	9	8	7	6	5	4	3	2	1	0
new state of center cell: $x_{ij}(\tau_n)$	1	0	0	0	0	0	0	0	0	0

(158)

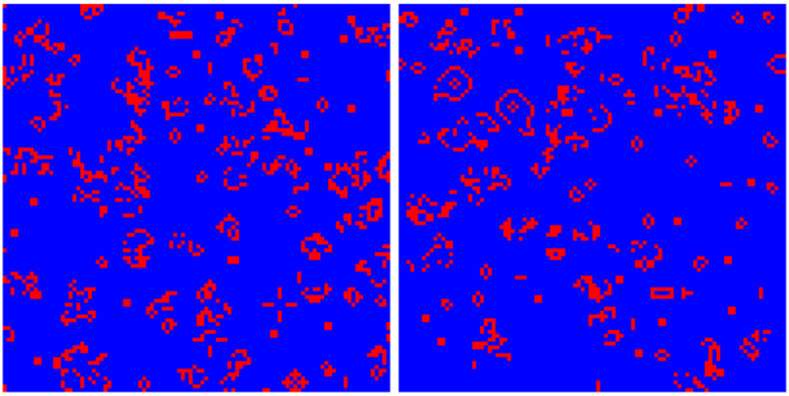


Fig. 48 Evolution of the Game of Life for 50 steps (*left*) and 100 steps (*right*) from a random initial pattern

where “1” and “0” in the rule table denote “black” and “white”, respectively. The above rule can be generated from the difference equation

$$x_{ij}(\tau_n) = M_9\left(\sum_{k,l\in(-1,0,1)} a_{k,l}x_{i+k,j+l}(\tau_{n-1})\right). \tag{159}$$

The basic dilation operation enlarges the areas of black pixels at their borders in binary images. The truth table can be written as follows [11]:

Rule of Totalistic Dilation

$\sum_{k=-1}^{1}\sum_{m=-1}^{1} x_{i+k,j+m}(\tau_{n-1})$	9	8	7	6	5	4	3	2	1	0
new state of center cell: $x_{ij}(\tau_n)$	1	1	1	1	1	1	1	1	1	0

(160)

where “1” and “0” in the rule table denote “black” and “white”, respectively. The above rule can be generated from the difference equation

$$x_{ij}(\tau_n) = 1 - M_0\left(\sum_{k,l\in(-1,0,1)} a_{k,l}x_{i+k,j+l}(\tau_{n-1})\right). \tag{161}$$

The truth table defining the edge detection was written as follows:

Rule of Edge Detection

$\sum_{k=-1}^{1}\sum_{m=-1}^{1} x_{i+k,j+m}(\tau_{n-1})$	9	8	7	6	5	4	3	2	1	0
new state of center cell: $x_{ij}(\tau_n)$	0	1	1	1	0	0	0	0	0	0

(162)

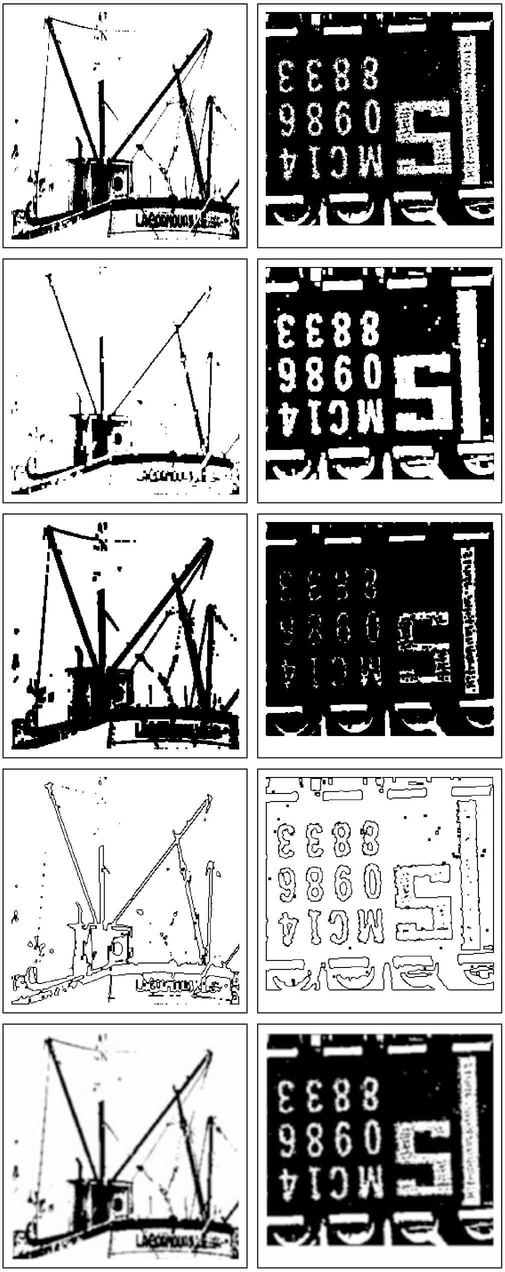


Fig. 49 Multitasking capability of a memristor cellular automaton. Binary images can be transformed into erosion images, dilation images, edge detection images, and half-toning images (illustrated from *top* to *bottom*) simultaneously with one operation

Here, we assumed that there are a few white cells in the neighborhood of edge. The above rule can be generated from the difference equation

$$x_{ij}(\tau_n) = \sum_{i=6}^{8} M_i \left(\sum_{k,l \in (-1,0,1)} a_{k,l} \, x_{i+k,j+l}(\tau_{n-1}) \right). \tag{163}$$

Consider again the memristor cell in Fig. 47. Define the memristance

$$\left.\begin{array}{l} M_a = M_9(q), \\ M_b = 1 - M_0(q), \\ M_c = M_6(q) + M_7(q) + M_8(q), \end{array}\right\} \tag{164}$$

which correspond to erosion, dilation and edge detection operations. Then, the DTCNN can perform an erosion operation, a dilation operation and an edge detection operation simultaneously with one operation. Furthermore, if we connect the memristor with the memristance

$$M(q) = \mathit{floor}\big(28|q|\big), \tag{165}$$

in series, it can also perform a half-toning operation simultaneously.

In this case, the template of the multitasking DTCNN is given by

$$A = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}, \qquad B = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}, \qquad w = \boxed{0}, \qquad c = \boxed{0}. \tag{166}$$

Our computer simulation of a multitasking operation is shown in Fig. 49.

10 Conclusion

In this paper, we have proposed the memristor cellular automaton and the memristor discrete-time cellular neural network, which can perform some logical operations, image processing operations and complex behaviors. There are many possible generalizations of these systems, which will be presented elsewhere.

Acknowledgements This work is supported in part by ONR grants No. N00014-07-1-0350 and N00014-09-1-0411.

References

1. Chua, L.O.: Memristor—the missing circuit element. IEEE Trans. Circuit Theory **CT-18**(5), 507–519 (1971)
2. Chua, L.O.: CNN: A Paradigm for Complexity. World Scientific, Singapore (1998)
3. Chua, L.O.: A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science, vols. I–II. World Scientific, Singapore (2006)
4. Chua, L.O., Kang, S.M.: Memristive devices and systems. Proc. IEEE **64**(2), 209–223 (1976)
5. Chua, L.O., Roska, T.: Cellular Neural Networks and Visual Computing. Cambridge University Press, Cambridge (2002)
6. Gardner, M.: Mathematical games—the fantastic combinations of John Conway's new solitaire game "life". Sci. Am. **223**, 120-123 (1970)
7. Itoh, M., Chua, L.O.: Designing CNN genes. Int. J. Bifurc. Chaos **13**(10), 2739–2824 (2003)

8. Itoh, M., Chua, L.O.: Advanced image processing cellular neural networks. Int. J. Bifurc. Chaos **17**(4), 1109–1150 (2007)
9. Itoh, M., Chua, L.O.: Memristor oscillators. Int. J. Bifurc. Chaos **18**(11), 3183–3206 (2008)
10. Itoh, M., Chua, L.O.: Imitation of visual illusions via OpenCV and CNN. Int. J. Bifurc. Chaos **18**(12), 3551–3609 (2008)
11. Itoh, M., Chua, L.O.: Difference equations for cellular automata. Int. J. Bifurc. Chaos **19**(3), 805–830 (2009)
12. Johnson, R.C.: Will memristors prove irresistible? EE Times **1538**, 30–34 (2008)
13. Peitgen, H.O., Jurgens, H., Saupe, D.: Chaos and Fractals: New Frontiers of Science. Springer, New York (1992)
14. Snider, G.S.: Self-organized computation with unreliable, memristive nanodevices. Nanotechnology **18**(36), 365202 (2007)
15. Strukov, D.B., Snider, G.S., Stewart, G.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
16. Tour, J.M., He, T.: The fourth element. Nature **453**, 42–43 (2008)
17. Wolfram, S.: Universality and complexity in cellular automata. Physica D **10**, 1–35 (1984)

Index

A. Adamatzky, L. Chua (eds.), *Memristor Networks*,
DOI 10.1007/978-3-319-02630-5,