Neural Networks 106 (2018) 223–236

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Evaluation of the computational capabilities of a memristive random network (MN^3) under the context of reservoir computing

Laura E. Suarez [a,b], Jack D. Kendall [b], Juan C. Nino [b,*]

[a] Department of Industrial Engineering, Universidad de los Andes, Bogotá, Colombia
[b] Department of Materials Science and Engineering, University of Florida, Gainesville, FL 32601, USA

ARTICLE INFO

Article history:
Received 3 January 2018
Received in revised form 17 May 2018
Accepted 10 July 2018
Available online 25 July 2018

Keywords:
Reservoir Computing
Memristive Networks
Neuromorphic Engineering
Speech Recognition

ABSTRACT

This work presents the simulation results of a novel recurrent, memristive neuromorphic architecture, the MN^3 and explores its computational capabilities in the performance of a temporal pattern recognition task by considering the principles of the reservoir computing approach. A simple methodology based on the definitions of ordered and chaotic dynamical systems was used to determine the separation and fading memory properties of the architecture. The results show the potential use of this architecture as a reservoir for the on-line processing of time-varying inputs.

1. Introduction

The continuous improvement in the performance and processing power of current computing devices has been mainly driven by the increase in the number of transistors that can be placed in a computer chip. In 1970, Intel co-founder Gordon Moore observed that this increase follows an exponential trend, such that the density of transistors per chip is doubled approximately every two years (Moore, 1965, 1975). However, this exponential tendency is unlikely to remain indefinitely. Several studies have suggested that the miniaturization of transistors will soon face technological and physical limitations and challenges (Waldrop, 2016). In addition to this, the architecture of current computing devices suffers from what is known as the von Neumann bottleneck problem, a limitation in the data transfer rate due to the physical separation between the memory and the processing units. This configuration results in reduced computational power and high energy consumption.

These two limiting factors and the continuously increasing demand for higher computational power by today's Big Data require different and unconventional computing architectures and approaches. One such example is neuromorphic systems, a computing approach inspired by the information processing characteristics of complex biological brains (Aleksander & Morton, 1989; Mead, 1990).

In contrast to modern computers, the massively interconnected architecture of the human brain allows it to reach far better performance thanks to the parallel interaction of thousands of neurons involved in a computation. Furthermore, there is no distance between memory and processing in the human brain; these two processes are equivalent and occur at a local level in the synaptic connections between neurons. This entails a more efficient storage and recall since the information exchange is direct and continuous, as opposed to the inherent latency in current von Neumann architectures.

By emulating the parallel architecture and network dynamics exhibited in biological neural circuits, neuromorphic systems demonstrate properties such as high compactness, robustness to noise, fault tolerance and computational and energy efficiency similar to those observed in biological systems (Aleksander & Morton, 1989; Jo et al., 2010; Mead, 1990; Mead & Ismail, 2012; Morabito, Andreou, & Chicca, 2013).

Recent advances in nanoelectronics have made possible the fabrication of devices that exhibit memristor-like characteristics. A memristor has been proposed as the fourth fundamental circuit element, after the resistor, the inductor and the capacitor (Chua, 1971). It was first theorized by Leon Chua based on symmetry principles. Essentially, a memristor is a two-terminal electronic device whose conductance can be incrementally modified according to the magnitude and direction of the applied voltage across or current through the device. Memristors can also work as non-volatile memories, which means they can store their resistance value even in the absence of a power source (Dongale et al., 2016; Kim et al., 2012). Given their adaptive behavior, memory storage capability, and nanoscale dimensions, these new memristive devices are ideal

* Correspondence to: Department of Materials Science and Engineering, University of Florida, Gainesville, FL 32601, USA
E-mail addresses: laura.suarez@mail.mcgill.ca (L.E. Suarez), jackdkendall@ufl.edu (J.D. Kendall), jnino@mse.ufl.edu (J.C. Nino).

https://doi.org/10.1016/j.neunet.2018.07.003
0893-6080/

for the emulation of synapses in ultra-high-density and compact artificial neuromorphic architectures (Jo et al., 2010; Snider, 2007, 2008).

Even though memristors have characteristics that make them ideal for the emulation of synaptic connections in neuromorphic hardware, their implementation has been hindered by the high device variability, mainly due to the internal conduction mechanisms of the device, such as filament formation and oxidation processes, as well as variability introduced during the manufacturing process. This is the case for systems whose operation depends on the training of the connections. For this reason, the practical use of memristors in neuromorphic systems requires a type of learning and network architecture that are highly tolerant to device imperfections and naturally embrace variability in memristor processing (Bürger & Teuscher, 2013; Goudarzi, Lakin, Stefanovic, & Teuscher, 2014). One such framework is the reservoir computing (RC) architecture (Jaeger, 2001; Maass, Natschlager, & Markram, 2002).

In the RC architecture, a temporal signal is projected into a high-dimensional space by a nonlinear dynamical system, or reservoir, that consists of a random, recurrent network of nonlinear units, which acts as a short-term memory for the signal, and thus contains information about past and present inputs. At the final stage, a memoryless linear classifier is trained on the output of the reservoir. Therefore, the connections within the reservoir are not trained; rather, the performance of the reservoir mainly depends on its global dynamical properties, and not on the individual properties of any device. These characteristics make the reservoir computing framework ideal for recurrent memristive networks.

To demonstrate the potential of this approach, this paper presents the simulation results of a novel recurrent, memristive neuromorphic architecture: the MN^3 (Memristive Nanofiber Neural Network), and its use for temporal pattern recognition within the framework of reservoir computing. This architecture is described in a recent patent by the authors, details of which can be found in Nino and Kendall (2017). Briefly, the system proposed in Nino and Kendall (2017) consists of a mat of electrospun memristive nanofibers, which can be spatially arranged to form an artificial neural network architecture that exhibits random, recurrent connections in a highly dense and compact structure, all features of biological neural networks. Given its recurrent and dynamic connectivity, the architecture proposed is expected to exhibit the requisite properties for the efficient reservoir design in the field of reservoir computing. By dynamic connectivity we refer here to the modifiable nature of the conduction strength of the memristive connections. However, the presence or absence of a connection is always fixed.

Acquiring a thorough theoretical understanding of the proposed neuromorphic hardware architecture (MN^3) is essential for evaluating its potential as a neuromorphic processor. Valuable insight into the behavior and dynamics of the network can be gained via numerical simulations. Addressing this need, we have developed a simulation tool in Python that can construct various instances of the proposed memristive neuromorphic architecture, simulate their dynamics, and evaluate their computational properties.

Prior studies focusing on networks of memristors designed to operate under the RC approach have been reported in the literature (Avizienis et al., 2012; Konkoli & Wendin, 2013a, b; Kulkarni & Teuscher, 2012; Oskoee & Sahimi, 2011; Sillin et al., 2013). However, most of them focused on the theoretical electrical characterization of these networks, and few of them explored their computational capabilities in the performance of real-world tasks. The efforts of the present work are thus emphasized on both providing details about the modeling and simulation of the MN^3 architecture, and the exploration of its computational capabilities in a speech recognition task by considering the operating principles of the reservoir computing (RC) framework. In addition, the modeling approach implemented in the present work introduces the effects of component mismatches and temporal sources of variability, which are critical in real memristive devices.

One of the first reported attempts to model and simulate a network of memristors was made by Oskoee and Sahimi (2011). They carried out numerical simulations of electrical currents in a two-terminal, square lattice of memristors and resistors that emulates a large size, two-phase material with memristive properties. The memristor model used in Oskoee and Sahimi (2011) is the moving-wall model proposed by Strukov in Strukov, Snider, Stewart, and Williams (2008), but they included a nonlinear window function in order to account for nonlinearities at the boundaries of the insulated layer. Simulation results showed various interesting behaviors: weak and strong memristive regimes, a possible first-order transition at the percolation threshold, nonlinear dynamics evidenced by the generation of second harmonics in the strong memristive regime, and dependence of the network's strength on the frequency. One of the limitations of that work was the restricted type of connectivity of the network, which constrained the maximum number of edges per node to four.

Some of the results obtained by Oskoee and Sahimi in Oskoee and Sahimi (2011), such as the manifestation of a soft and a hard switching regime and the generation of higher harmonics, were further confirmed in a later work by Sillin et al. (2013). In that work, a network of atomic switches, which contains elements of a nearest-neighbor, random network, was simulated. In contrast to Oskoee and Sahimi's work, the number of edges per node of the network proposed in Sillin et al. (2013) was not constrained to a certain number. Interestingly, in order to model the atomic switches, Sillin et al. used a memristor model similar to the one proposed by Oskoee and Sahimi in Oskoee and Sahimi (2011), but they included a dissolution and a stochastic term in the derivative of the state variable, which accounted for the breakdown and the variability of the conductive pathways, respectively. Additionally, Sillin et al. showed that the atomic switch network could potentially serve as a reservoir in the context of RC by demonstrating its performance in a waveform generation task (Sillin et al., 2013).

Another reported attempt to simulate a network of memristors is the work of Kulkarni and Teuscher (2012). They built a software framework that allowed them to create memristor networks, to simulate and evaluate them in NgSpice, a mixed-level/mixed-signal circuit simulator. As Sillin et al. (2013), Kulkarni and Teuscher suggested that the random network of memristors could be used as a reservoir in the context of RC, and successfully demonstrated its performance in a pattern recognition and associative memory tasks using synthetic signals, and a genetic algorithm to train the readout module.

There is also the work proposed by Konkoli and Wendin (2013a). They created a generic software tool for the simulation of the dynamics of sparse multi-terminal memristive networks, and they implemented the threshold memristor model initially proposed in Pershin and Di Ventra (2010). This software tool was used in a later work by the authors to evaluate the computational capabilities of a network of memristors in the context of RC (Konkoli & Wendin, 2013b). They proposed the hypothesis that if the nonlinear frequency response of the reservoir cannot be approximated by a linear combination of delayed inputs, then the quality of the reservoir is good. For testing this hypothesis, they proposed a dissimilarity measure in the frequency domain between the real output voltage at the internal nodes of the network, and the output produced by the linear combination of the delayed inputs. The more dissimilar these two signals are, the better is the quality of the reservoir. They used the same dissimilarity measure for the conductance and the voltage between internal nodes, and found

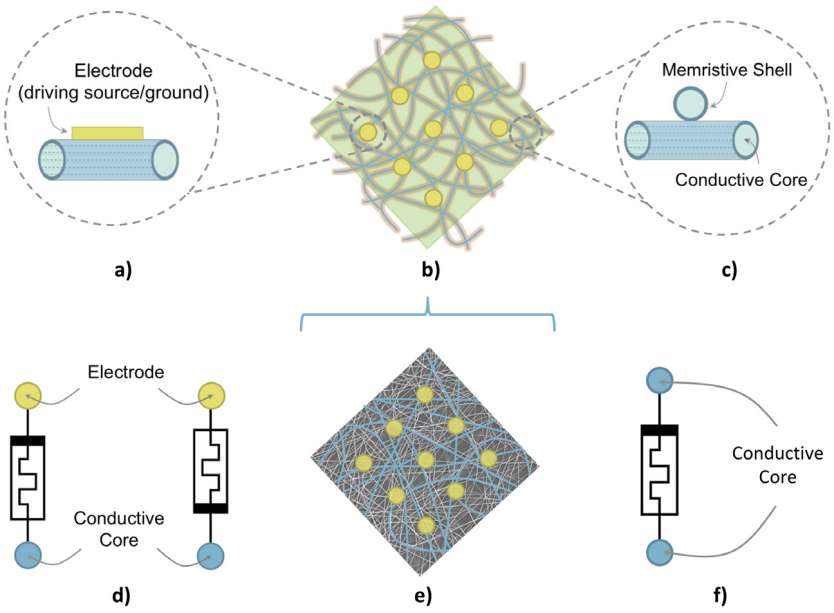

Fig. 1. Electrospun memristive nanofiber neural network architecture MN^3. (a) Detailed view of a connection between a memristive nanofiber and an external electrode. (b) Schematic of the network of memristive nanofibers. (c) Detailed view of a connection between two memristive nanofibers; each nanofiber consists of an internal conductive core coated with a memristive shell. (d) The connection between an electrode and a nanofiber is modeled as a single memristor. (e) SEM image of the electrospun memristive nanofiber with depicted electrodes. (f) The connection between two memristive nanofibers is modeled as a single memristor. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that these two variables present higher dissimilarity measures than the voltage at the internal nodes (Konkoli & Wendin, 2013b).

One of the most recent works on the application of memristive networks using the RC architecture was the one proposed by Kudithipudi, Saleh, Merkel, Thesing, and Wysocki (2016). They developed a software-based neuromemristive RC architecture, with doubly twisted toroidal structure, and they showed the applicability of this architecture for biosignal processing applications. Kudithipudi et al.'s work will be later used as a benchmark for the performance of the MN^3 architecture here presented.

In order to better contrast the current work with prior efforts, the structure of this paper is as follows: first, we provide a description of the MN^3 architecture, followed by details about the modeling and the simulation of the network. Next, we evaluate the computational capabilities of the network by assessing its performance in a speech recognition task under the paradigm of reservoir computing. Finally, we report our conclusions regarding the computational capabilities of the architecture proposed and provide a guidance to continue exploring and harnessing the dynamics of memristive networks in future work.

2. Materials and methods

This section presents a detailed description of the system, as well as the considerations and assumptions that were taken into account for the modeling and simulation of the network.

2.1. The architecture

The MN^3 architecture consists of a mat of electrospun memristive nanofibers, which can be spatially arranged to form an artificial neural network architecture that exhibits random, recurrent connections, in a highly dense and compact structure.

A schematic of the MN^3 architecture is shown in Fig. 1. The detailed views illustrate the cross section of a single fiber, as well as the configuration of the connection between two fibers at an intersection (c), and the connection between a fiber and an electrode (a). Each fiber (blue lines in b and e) is made up of a conductive core coated with a memristive shell. It is worth noting that the connections at the intersection of two nanofibers (Fig. 1(c)) could have also been modeled as 2 memristors in series with different polarities. We did build a preliminary model of the network taking into account this assumption. Since results did not present observable changes in terms of prediction performance (not shown in this work), we decided to work with the simplified version of the model (i.e., assuming that connections at the intersection of two nanofibers can be modeled as a single memristor), which resulted to be less computationally expensive.

While this architecture is a physical embodiment, it can also be conceived as an electrical circuit that, at a higher level of abstraction, represents the concept of an artificial neural network in the context of machine-learning. Fig. 2 provides a detailed sketch of the three different representations for the proposed MN^3 architecture.

2.2. Modeling the system as a graph

For modeling and simulation purposes, the electrical circuit representation of Fig. 2(b) was used. We adopted a graph-based approach to model the circuit as a network of memristive links that connect an array of external (controllable) and internal (non-controllable) nodes. The set of external nodes may either represent an electrode connected to a driving source or a ground; and the set of internal nodes may represent an accessible electrode, or the conductive core of a fiber (non-accessible nodes). The total set of nodes is denoted by $\mathcal{N}$. The set of links of the network, denoted by $\mathcal{L}$, represent physical memristors that connect pairs of nodes.

2.3. Modeling connectivity

The connectivity of the network is defined by an $\mathcal{N}\times\mathcal{N}$ adjacency matrix. The i, jth element of this matrix is 1 if there is a memristive connection between the pair of nodes i, j, otherwise the i, jth element is zero. The assignment of the connections was determined by a random variable that follows a Bernoulli distribution with parameter p, where p approximates the average probability of two

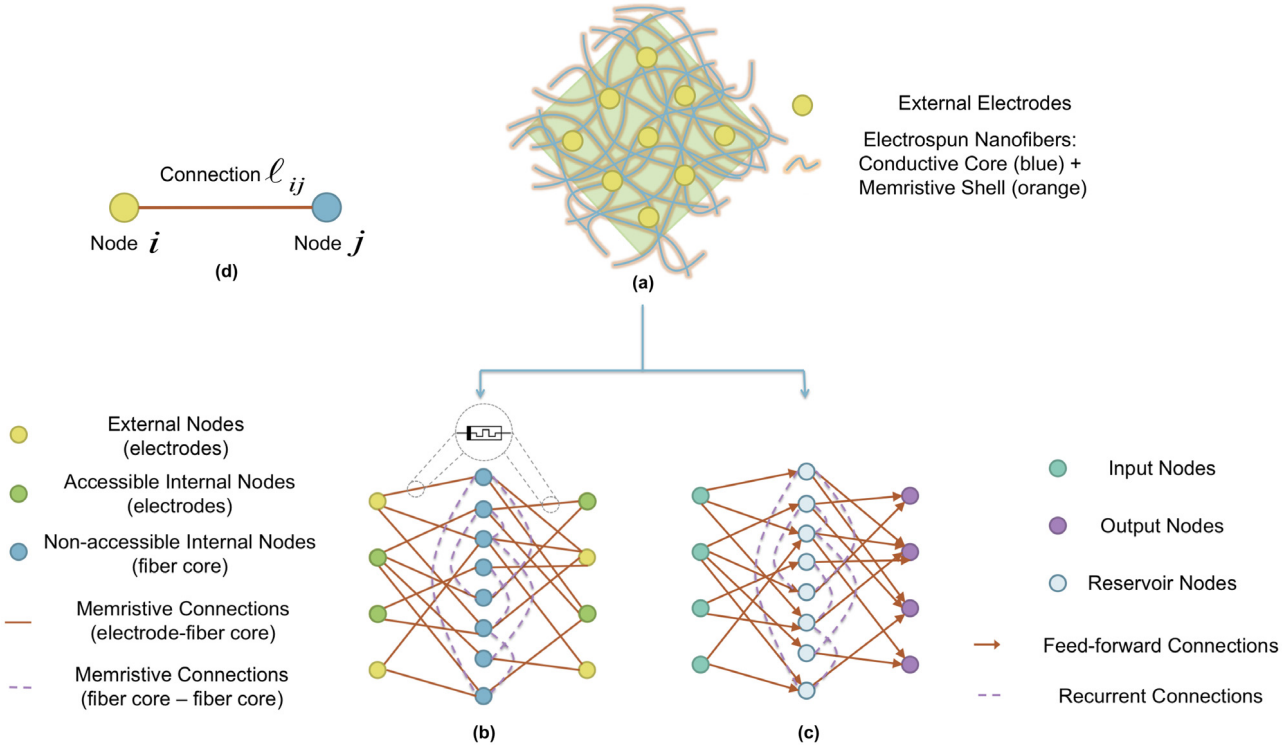

Fig. 2. Physical (a), electrical (b), and artificial neural network (c) representation of the memristive nanofiber neural network MN^3; (d) representation of a memristive connection between the nodes i and j.

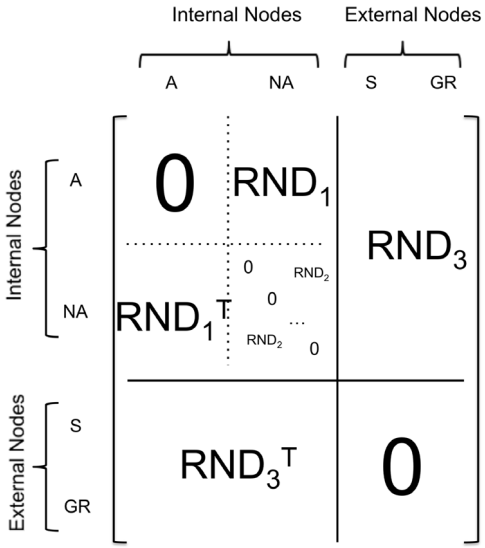

Fig. 3. General structure of the node-to-node adjacency (connectivity) matrix for the MN^3 architecture. Internal nodes can either be: accessible (A) or non-accessible (NA); external nodes can either be a source (S) or a ground (GR).

fibers being in direct contact when arbitrarily arranged in space during the electrospinning process. Given the physical architecture of the MN^3 network (Figs. 1 and 2), the connectivity matrix would have the form depicted in Fig. 3.

2.4. The algorithm

The algorithm used for the temporal evolution of the network under an external applied voltage is a simplified version of the one proposed in Konkoli and Wendin (2013a), and is summarized below:

1. Random initialization of the conductance/resistance values of the memristive connections.
2. For fixed conductance/resistance values, determine the voltage at the internal nodes using Kirchhoff's Current Law (KCL).
3. Determine the voltage across each memristive connection using the results obtained in step 2.
4. Using Euler's method, for a short delta of time ($dt = 1 \times 10^{-4}$ s), update the value of the conductance/resistance of each memristive connection using the voltages obtained in step 3.
5. Go back to step 2 until the end of the external applied voltage.

This algorithm was implemented in Python.

2.5. The memristor model

Several physical and chemical mechanisms have been found to be responsible for the behavior of memristors and memristive devices. Some of these are: nanomechanical effects, molecular switching effects, electrostatic/electronic effects, electrochemical metallization effects, valence change effects, thermochemical effects, phase-change effects, magnetoresistive effects and ferroelectric tunneling effects. A detailed review on these mechanisms is provided in Waser, Dittmann, Staikov, and Szot (2009). Based on these mechanisms, several memristor models can be found in the literature (Abdalla & Pickett, 2011; Berdan, 2013; Chang et al., 2011; Laiho, Lehtonen, Russel, & Dudek, 2010; Linn, Siemon, Waser, & Menzel, 2014; Nugent & Molter, 2014; Pershin & Di Ventra, 2010; Pickett et al., 2009; Vourkas & Sirakoulis, 2013; Williams, Pickett, & Strachan, 2013; Yakopcic, Taha, Subramanyam, & Pino, 2013).

In Nugent and Molter (2014), authors proposed a generalized memristive device model capable of accurately reproducing the empirical behavior of several memristive devices including the Ag-chalcogenide, AIST, GST, and WO_x devices. Given its versatility, we used this model for simulation purposes. This semi-empirical model assumes that the current through the device comes from both a memory-dependent current component, I_m, and a Schottky current, I_s in parallel:

$$I = \phi I_m(V, t) + (1 - \phi) I_s(V), \tag{1}$$

where $\phi \in [0, 1]$.

The Schottky component, $I_s(V)$, represents the Schottky barrier that many memristive devices contain and that is formed at the metal–semiconductor junction. On the other hand, the memory component, I_m, comes from the notion that memristors can be represented as a collection of conducting channels that switch between states with a different resistance value. The resistance of the device can be modified through the application of an external voltage that causes the channels to transition between low and high conducting states. Thus, each conducting channel can be treated as a metastable switch (MSS), and the conductance of the device is defined by a collection of these MSSs that capture the memory effect of the memristor.

In this model, the low and high conducting states of each channel are separated by a potential barrier. If we let the barrier potential be the reference potential $V = 0$, then the probability that the MSS transitions from the B state to the A state is given by:

$$P_A = \alpha \frac{1}{1 + e^{-\beta(V - V_A)}} = \alpha \Gamma(V, V_A). \tag{2}$$

And the probability that the MSS transitions from the A state to the B state is given by: $P_B = \alpha(1 - \Gamma(V, -V_B))$. Here, $\beta = \frac{q}{kT} = (V_T)^{-1}$, where V_T is the thermal voltage, $\alpha = \frac{\Delta t}{t_c}$ is the ratio of the time step period Δt to the characteristic time scale of the device t_c, and V is the voltage across the switch.

The intrinsic electrical conductance of each MSS is given by G_A and G_B. The total memristor conductance is given by the sum over each MSS:

$$G_m = N_A G_A + N_B G_B = N_B (G_B - G_A) + N G_A, \tag{3}$$

where N_A is the number of MSSs in the A state, N_B is the number of MSSs in the B state, and $N = N_A + N_B$. Therefore, a memristor is modeled as a collection of N MSSs that evolve in discrete time steps, Δt. At each time step, a subset of the MSSs in the A state will transition to the B state, and vice versa. The probability that x MSSs will transition out of a population of n MSSs is given by a binomial distribution:

$$P(x, n) = \frac{n!}{x!(n - 1)!} p^x (1 - p)^{n - x} \tag{4}$$

where p is the probability that a MSS will transition states.

The change in conductance of a memristor is thus modeled as a probabilistic process in which the number of switches that transition between A and B states is drawn from a binomial distribution with a center at np and variance $np(1 - p)$. This random process accounts for the intrinsic dynamic stochastic behavior of memristive devices. The update of the memristor conductance is thus given by the contribution of two random variables, RV_A and RV_B, drawn from two binomial distributions:

$$\Delta N_B = RV_A(N_A, P_A) - RV_B(N_B, P_B) \tag{5}$$

The final update to the conductance of the memristor is thus given by:

$$\Delta G_m = \Delta N_B (G_B - G_A) \tag{6}$$

Table 1
Parameters for the Generalized Metastable Switch Model (MSS) of an Ag-chalcogenide memristive device.

Model parameter	Distribution
t_c [ms]	$t_c \sim$ Normal $(\mu = 0.32 \times 10^{-3}, \sigma = 0.1\mu)$
$W_{off} = G_A N_A$ [mS]	$W_{off} \sim$ Normal $(\mu = 0.91 \times 10^{-3}, \sigma = 0.1\mu)$
$W_{on} = G_B N_B$ [mS]	$W_{on} \sim$ Normal $(\mu = 0.87 \times 10^{-2}, \sigma = 0.1\mu)$
V_A [V]	$V_A \sim$ Normal $(\mu = 0.17, \sigma = 0.1\mu)$
V_B [V]	$V_B \sim$ Normal $(\mu = 0.22, \sigma = 0.1\mu)$
N	$N \sim$ Normal $(\mu = 10\,000, \sigma = 0.1\mu)$

To include device-to-device variability effects, the parameters of the model, as well as the initial conditions of each of the elements in the network were drawn from the probability distributions described in Table 1. The values of these parameters are fitted to an Ag-chalcogenide memristive device (Nugent & Molter, 2014).

2.6. Computational capabilities of the MN^3 as a reservoir: the isolated digit speech recognition task

To test the computational capabilities of the MN^3 architecture, we simulated an instance of the network to perform a temporal pattern recognition task using the network as a reservoir under the principles of the RC approach. The task, commonly known as the isolated digit speech recognition task, consists in the supervised classification of isolated spoken digits from '0' to '9'. Here, the term 'supervised' refers to the *supervised learning paradigm* as used in the context of machine-learning. Therefore, the linear classifier used for the classification of the isolated spoken digits (after their transformation by the memristive reservoir) was trained on the labeled data set described below.

The data set used for this task was taken from the TI46 corpus (Doddington & Schalk, 1981), which consists of 500 samples; each digit was recorded 10 times by 5 female speakers. The raw signals of this data set are usually preprocessed for speech recognition purposes. The preprocessing consists in a transformation of the signal to the frequency domain and a selective filtering based on psychoacoustic properties of the human ear (Vandoorne et al., 2014). The version of the data set used in the present work was preprocessed using the Lyon ear model described in Lyon and Shamma (1996), applying a decimation of the input signals with a factor of 128. This preprocessing step decomposes the signal of each digit into 77 separate temporal signals, each corresponding to a different frequency component. Fig. 5(b) illustrates an example of the resulting temporal signals for each of the frequency components (each color in Fig. 5(b) represents a different frequency component) for one of the 10 samples of digit '1'.

The instance of the MN^3 architecture simulated for this task consisted of a network with a total of 400 internal nodes (100 accessible and 300 non-accessible nodes; we simulated as large a network was practical), and 82 external nodes (77 sources, each corresponding to one of the 77 temporal frequency components of the input signals. We incorporated 5 grounded nodes to encourage a more evenly distributed current throughout the network). A schematic graph of the architecture is shown in Fig. 4. Connections between pairs of nodes were randomly generated using a Bernoulli distribution with $p = 0.9$. The parameters of each of the memristive connections, as well as the initial conductance states, were drawn from the probability distributions described in Table 1.

Each one of the 77 temporal frequency components (Fig. 5(b)) of the 500 samples was fed as an input voltage (scaled by a factor of 1×10^3) to one of the 77 source nodes of the simulated MN^3 architecture. Network states, i.e., the voltages at the internal nodes (both accessible and non-accessible), were recorded at each point in time. For the classification of the spoken digits, 10 distinct linear regression models were trained in parallel (Fig. 5(d)), one per

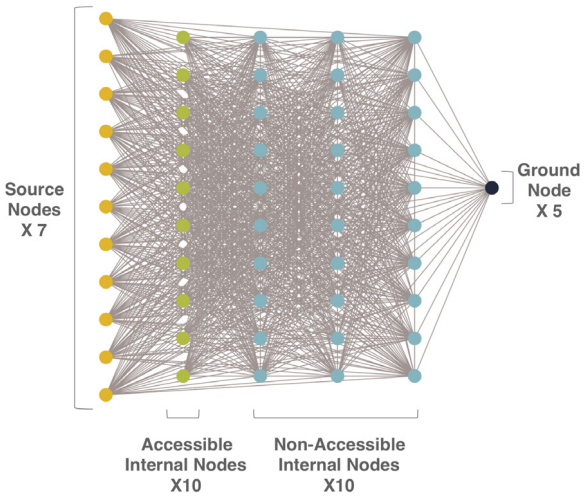

Fig. 4. Network architecture used for the isolated digit speech recognition task. The simulated network consisted of 482 nodes: 77 external driving sources, 100 accessible internal nodes, 300 non-accessible internal nodes and 5 grounded nodes. The spatial location of the nodes serves just for representational purposes.

digit, using as input the memristive network states (Fig. 5(c)). The data set was divided into training (450 samples) and test sets (50 samples). The test set was used to provide a reliable estimate of the prediction accuracy of the model. Since there is a prediction value for each point in time, a winner-take-all approach was used to determine which digit was spoken during the test procedure. Given the large number of input features, we used Ridge regression to penalize those input features that are less important in the prediction of the digit label. Fig. 5 shows a scheme for the isolated digit speech recognition task.

2.7. Computational capabilities of the MN^3 as a reservoir: the separation and fading memory properties

According to the RC framework, a reservoir should possess the two following properties to be computationally useful: fading memory and separation (Jaeger, 2001; Maass et al., 2002).

The separation property (SP) is related to the computational power[1] of the reservoir; it refers to the ability of the reservoir to generate separate network states in response to different input signals, i.e. being able to map different input streams into significantly different trajectories in the phase space of the reservoir (Jaeger, 2001; Maass et al., 2002). The fading memory property (FMP), on the other hand, is related to the information storage capacity of the reservoir; it refers to the ability of the reservoir to integrate temporal information of the input history into its current state (Jaeger, 2001; Maass et al., 2002). This ability of the reservoir is determined by the range of stable patterns that the network can generate.

Previous works have demonstrated that these two properties are strongly related to the dynamics of the reservoir (Beggs, 2008; Kauffman, 1993; Langton, 1990; Legenstein, 2005; Legenstein & Maass, 2007). The diversity in the number of states associated to the SP is a characteristic feature of systems operating in a chaotic regime (Legenstein & Maass, 2007), whereas the ability to produce stable state patterns associated to the FMP is a characteristic of systems operating in an ordered regime. Therefore, if a system operates under a chaotic dynamical regime, its state should be sensitive to changes in the input when the initial conditions are held constant (Legenstein & Maass, 2007). In contrast, if a system operates under an ordered dynamical regime, regardless of the differences in the initial conditions, state differences should approach zero if there are no longer any differences in the inputs (Legenstein & Maass, 2007). Accordingly, a reservoir can be conceived as a time-dependent pattern generator, and the evaluation of its computational capabilities essentially relies on the analysis of these spatiotemporal patterns for characterizing its dynamical regime.

Using the conditions for ordered and chaotic dynamics in autonomous dynamical systems, different analyses were performed to determine the separation and fading memory properties of the MN^3 architecture, and the extent to which these properties can explain its performance in the isolated digit speech recognition task.

First, to quantify the SP of the memristive reservoir, we evaluated the evolution of the distance between the trajectories (in phase space) of the network states, as defined by the voltages at the accessible internal nodes,[2] generated by two different inputs (i.e., two different digits), while holding the initial conditions constant (i.e., the initial values of the conductance). Each input signal consisted of a consecutive sequence of 11 samples of the exact same digit (see Fig. 6(a)). These input sequences were generated for digits '0', '1', '2', and '3', arbitrarily chosen, and were independently fed into the same simulated instance of the MN^3 architecture (i.e., an instance of the MN^3 network with the same physical parameters and connectivity matrix). To isolate the effects due to variations in the initial conditions of the system, the network was reset to the same conductance state for each input sequence. As with the speech recognition task, for every run, voltages at the internal nodes were recorded for each point in time, and the distance between trajectories in phase space of the network states for all six possible paired combinations of the four given input sequences was estimated.

The method used to quantify the FMP of the memristive reservoir is slightly different to the one used for the SP. In this case, since we are interested in evaluating the sensitivity of the system to variations in its initial conditions[3] when the input conditions are held constant, we assessed the evolution of the distance between trajectories (in phase space) of the network states (as defined by the voltages at the accessible internal nodes) generated by the exact same input sequence, but for different initial conditions, i.e., different initial values of the conductance. We used as inputs the same four sequences described above for the SP (see Fig. 6(c)). Each of these input sequences was independently fed twice into an identical simulated instance of the MN^3 architecture (i.e., same physical parameters and connectivity matrix), but this time the initial conductance values were randomly reassigned in every run. As with the speech recognition task and the SP, for every run, voltages at the internal nodes were recorded for each point in time, and the distance between trajectories in phase space of the network states for all pairs of simulated initial conditions, i.e., for each input sequence under (two) different initial conductance values, was estimated.

We used the definition of Euclidean distance as a measure to quantify both the SP and the FMP. Thus, at time t, the distance $d_{u,v}$ between trajectories (in the phase space) for network states $U(t) \epsilon R^N$ an$V(t) \epsilon R^N$ is given by:

$$d(t)^2_{u,v} = \sum_{i=1}^{N} [u_i(t) - v_i(t)]^2 \tag{7}$$

[1] There is not a unique definition for the computational power of a machine. However, a machine is said to be computationally powerful depending on the amount of operations that it can perform, i.e., the number of mappings between inputs and outputs.

[2] The state of the system was defined only in terms of the voltage at the internal nodes, since they are of practical relevance.

[3] Since the network proposed here is an input-driven dynamical system, a large number of its state variables are zero unless there is an external driving force. This is the case for all voltages and currents, except for the conductance of the memristive connections. Consequently, the sensitivity of the network to variations in the initial conditions was evaluated with respect to the conductance state of the memristors.

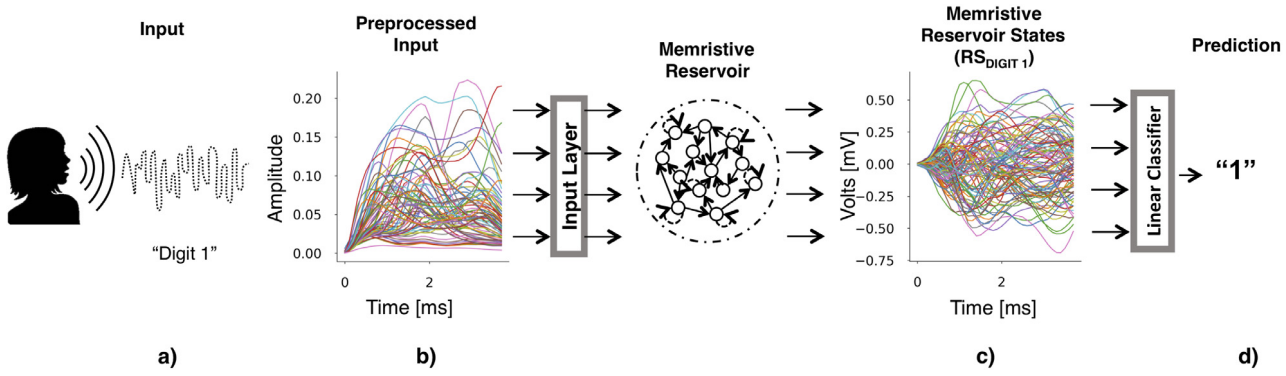

Fig. 5. The isolated digit speech recognition task. (a) Toy example of the original input signal before preprocessing. (b) Real sample input signal for digit '1' after preprocessing; each color codes for a different frequency component (77 frequency components were used in total for all the digits). (c) Resulting memristive reservoir states for a sample of digit '1'. (d) Prediction of digit label by the linear classifier (Ridge regression).

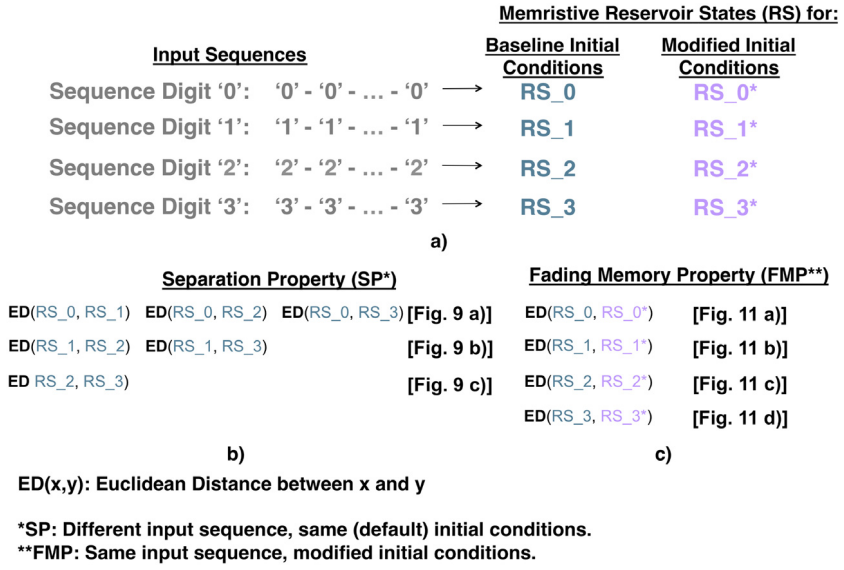

Fig. 6. Schematic of the methodology used to measure the separation property (SP) and fading memory property (FMP) of the MN^3 architecture. The sequences of digits used as inputs for the reservoir are depicted in (a) (upper left), with their corresponding resultant reservoir states (RS) (upper right), for both the default and the modified initial conditions. The different combinations of resultant reservoir states used to test the SP and the FMP are shown in (b) and (c), respectively.

In this case, $u_i(t)$ and $v_i(t)$ represent the voltage at the ith internal node at time t for network states $U(t)$ and $V(t)$, respectively, and N corresponds to the total number of internal nodes used in the calculation of the distance. A scheme of the methodology to quantify the SP and the FMP is depicted in Fig. 6.

3. Results and discussion

3.1. The isolated digit speech recognition task

Speech recognition is intrinsically a nonlinearly separable problem. Nevertheless, according to the operating principles of the RC approach, the reservoir should act as a filter that performs a nonlinear projection of the input space into a higher dimensional space, and by doing this, it converts nonlinearly separable input data into a linearly separable one, thus reducing the complexity of the problem.

Results for the isolated digit speech recognition task are shown in Fig. 7. The classification task was carried out for two different scenarios: first, using only the voltages at the accessible internal nodes (100 nodes; Fig. 7(a)), and second, using the voltages at all the internal (both the accessible and non-accessible) nodes (400 nodes; Fig. 7(b)). As a control, the same classification task was performed using the original preprocessed signals (Fig. 7(c)). In addition, for benchmark purposes, the results using the network states of a (non-memristive) reservoir composed of 100 leaky integrator neurons[4] were included (Fig. 7(d)) (Verstraeten et al., 2012).

The high classification performance presented in Fig. 7(c) corresponding to the original preprocessed signals shows that the frequency domain transformation step of the preprocessing stage reduces already in large part the complexity of the problem by converting the signal in a quasi-linearly separable problem. In this case, the precision score is 1.0 for 6 of the 10 digits, and 0.81 on average for the 4 remaining digits. When compared to the results of the memristive reservoir using only the voltages at the accessible internal nodes (Fig. 7(a)), the number of digits whose precision score is 1.0 goes up to 8, and the two remaining digits show an averaged precision score of 0.70. Nevertheless, if all the internal nodes (i.e., the accessible and non-accessible nodes) of the memristive reservoir are used, the precision is 1.0 for all the classes (Fig. 7(b)). This improvement in the classification performance of the memristive reservoir when all the internal nodes are included

[4] A leaky integrator neuron is a standard neuron with a first-order low-pass filter added to its output.

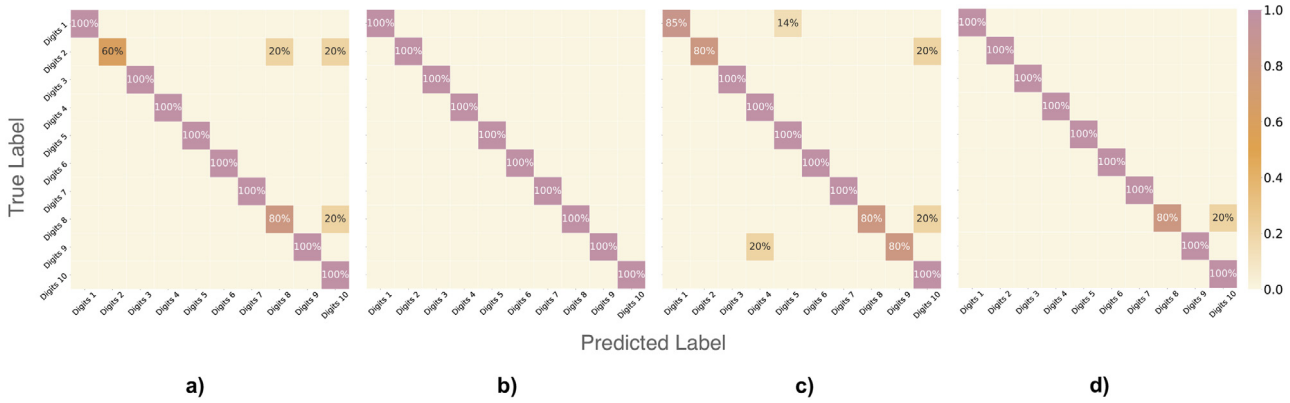

Fig. 7. Classification performance (confusion matrix) for the isolated digit speech recognition task using the voltage at (a) the accessible internal nodes of the memristive reservoir (100 nodes), at both (b) the accessible and non-accessible internal nodes of the memristive reservoir (400 nodes); (c) the original preprocessed data, and (d) the states of a reservoir made-up of leaky integrator units (Verstraeten et al., 2012).

could be due to either an increase in the number of input features, or to special discriminative features provided by the voltage at the non-accessible internal nodes. The dynamics of the latter (non-accessible internal nodes) are likely to be different due to their embedding in the physical matrix where they directly interact with each other (Fig. 3), in contrast to the accessible nodes, which do not interact at all.

The high classification performance corresponding to the memristive reservoir states (Fig. 7(a) and (b)) suggests that the projection made by the memristive reservoir on the preprocessed signals is further decreasing the complexity of the problem by increasing the separability of the signals. To test this, we trained multiple classifiers with different sizes for the training set using as input the memristive reservoir states, and the original preprocessed signals. If the memristive reservoir is indeed increasing the separability of the signals, then the linear classifier should be able to attain high precision and recall scores when trained with fewer samples. Results are shown in Fig. 8.

Fig. 8(c) shows that the classification performance in terms of both the precision and recall scores notably declines as the size of the training set is reduced when the original preprocessed signals are used as input. In contrast, the classification performance for the memristive reservoir states, using only the accessible internal nodes (Fig. 8(b)), and both the accessible and non-accessible internal nodes (Fig. 8(a)), barely decreases with the reduction of the training set. However, the case in which the voltage at both the accessible and non-accessible internal nodes are used (Fig. 8(a)), seems to outperform the case in which only the voltage at the accessible internal nodes is used (Fig. 8(b)). In the former case, the performance in terms of both precision and recall only shows a marked decline when the training size is just 30% of the whole sample size, i.e., 150 out of 500 samples. Altogether, these results show that the projection performed by the memristive reservoir on the original preprocessed signals does reduce the complexity of the problem by improving the separability of the preprocessed signals.

From the reported attempts on simulation of network of memristors used as reservoirs within the RC framework, Kudithipudi et al.'s work (Kudithipudi et al., 2016) is perhaps the most suitable to be used as benchmark for the MN^3 architecture here proposed. In Kudithipudi et al. (2016), Kudithipudi et al. used a software-based memristive RC architecture for epileptic seizure detection and EMG prosthetic finger control; they reported an accuracy of 90% and 84%, respectively. Although the complexity of the multi-label isolated spoken digit recognition task here presented, and the biosignal processing tasks performed by Kudithipudi et al. are not by any means directly comparable, they both present relatively good prediction performance when compared to other non-memristive, non-RC architectures such as the ones presented in Chapaneri and Jayaswal (2013) and Limkar, Rao, and Sagvekar (2012), which use dynamic time warping (DTW) and Mel frequency cepstrum (MFCC) for a similar isolated spoken digit recognition task, reporting classification accuracies of 90.5% and 99.16%, respectively.

3.2. The separation property (SP)

The distance between trajectories (in phase space) of the network states of the memristive reservoir, generated by two different input sequences of digits under the same initial conductance states, for all possible combinations of inputs (see Fig. 6(b)), is shown in Fig. 9. For comparison purposes, the same analysis described in Section 2.7 for the evaluation of the SP was carried out using the network states of the same (non-memristive) reservoir presented in Section 3.1 (cyan line in Fig. 9). In addition, the separation between the input sequences as described by the original preprocessed signals was also estimated (light blue line in Fig. 9).

A reservoir is said to have the SP if the distance between trajectories of the network states neither decays to zero nor diverges exponentially with time (Jaeger, 2001; Maass et al., 2002). Fig. 9 shows that for all possible combinations of input sequences, the separation between trajectories of the network states of the memristive reservoir is maintained in time: it neither decreases nor exponentially diverges (dark blue line in Fig. 9). In addition, compared to the results for the reservoir composed of leaky integrator neurons, and the original preprocessed input signals, the memristive reservoir consistently presents a higher SP.

To test whether the SP of the memristive reservoir is responsible for its increased performance when all the internal nodes, accessible and non-accessible, are included in the classification analysis (as shown in Fig. 7(a) and (b) for the speech recognition task), we studied the effect of the number of internal nodes under consideration on the estimation of both the SP and the classification performance of the memristive reservoir. Results of these analyses are shown in Fig. 10.

Fig. 10 shows that there is a positive correlation between the number of internal nodes considered and the classification performance of the memristive reservoir. The higher the number of internal nodes included for the classification task, the higher is the performance of the classifier in terms of both precision and recall. Furthermore, Fig. 10(b) suggests that the SP of the memristive reservoir is susceptible to the number of internal nodes considered for its estimation. This result is not entirely surprising. Increasing the number of internal nodes for the estimation of the SP naturally leads to an increase in the number of dimensions describing the

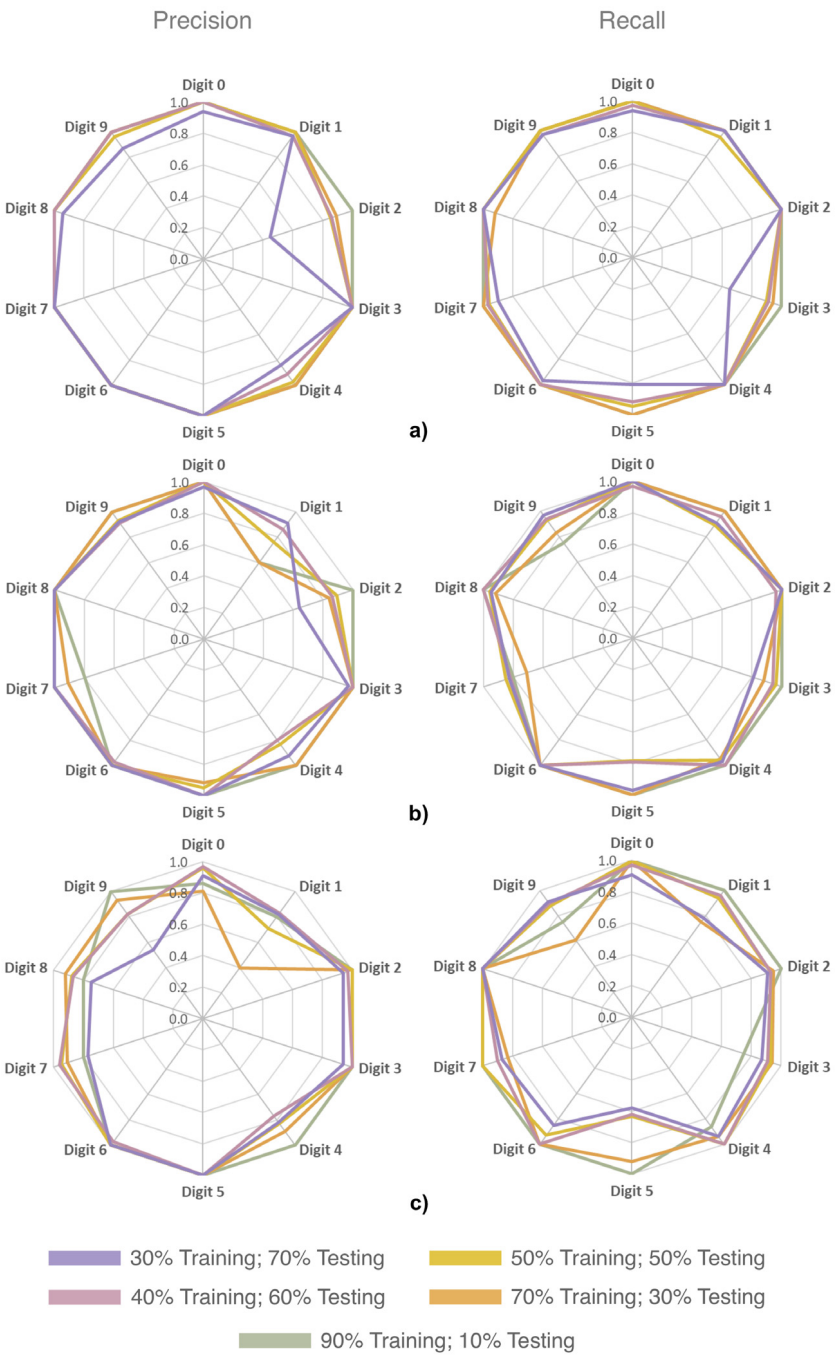

Fig. 8. Classification performance (in terms of precision and recall for each digit label) as a function of the training size for the isolated digit speech recognition task using the voltage at (a) the accessible internal nodes of the memristive reservoir (100 nodes), at both (b) the accessible and non-accessible internal nodes of the memristive reservoir (400 nodes); and (c) the original preprocessed data.

state of the network, thus positively contributing to the separation between two network states. However, this only holds true if the new added dimensions provide new information about the input space, and that depends on the degree of nonlinearity of the memristive reservoir. This will be further analyzed in Section 3.4.

Overall, these results suggest that the improvement in the classification performance of the memristive reservoir could have been, at least, partially driven by an increase in the SP of the system potentially induced by the augmentation of the dimensionality due to the inclusion of more internal nodes in the prediction of the spoken digits' labels. However, as we will show in Section 3.4, an increase in the SP purely driven by the augmentation of the dimensionality is not a sufficient condition for obtaining a good classification performance. In addition, a nonlinear component should be present.

3.3. The fading memory property (FMP)

Fig. 11 shows the distance between trajectories (in phase space) of the network states of the memristive reservoir, generated by the same input sequence (i.e., same digit) under different initial conductance states. Results for the reservoir made of leaky integrator units (the same one used in Fig. 7(d) and (cyan line of) Fig. 9) were not included here because in the source code used for the simulation of this type of reservoir, the states of the network were by default initialized with the same values. Likewise, the

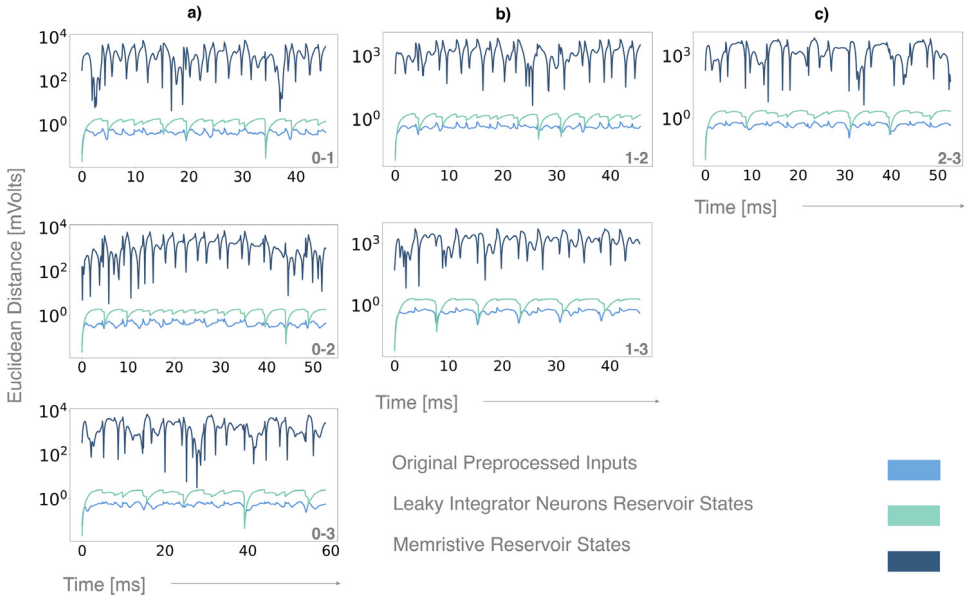

Fig. 9. Quantification of the separation property (SP). Euclidean distance between the trajectories (in phase space) of the memristive reservoir (using only the voltage at the accessible internal nodes; dark blue line), and the reservoir formed by leaky integrator units (cyan line). For comparison, the Euclidean distance using the original preprocessed data is also shown (light blue line). The distance between digit '0' and digits '1', '2' and '3' is shown in (a); the distance between digit '1' and digits '2' and '3' is shown in (b), and the distance between digits '2' and '3' is shown in (c). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

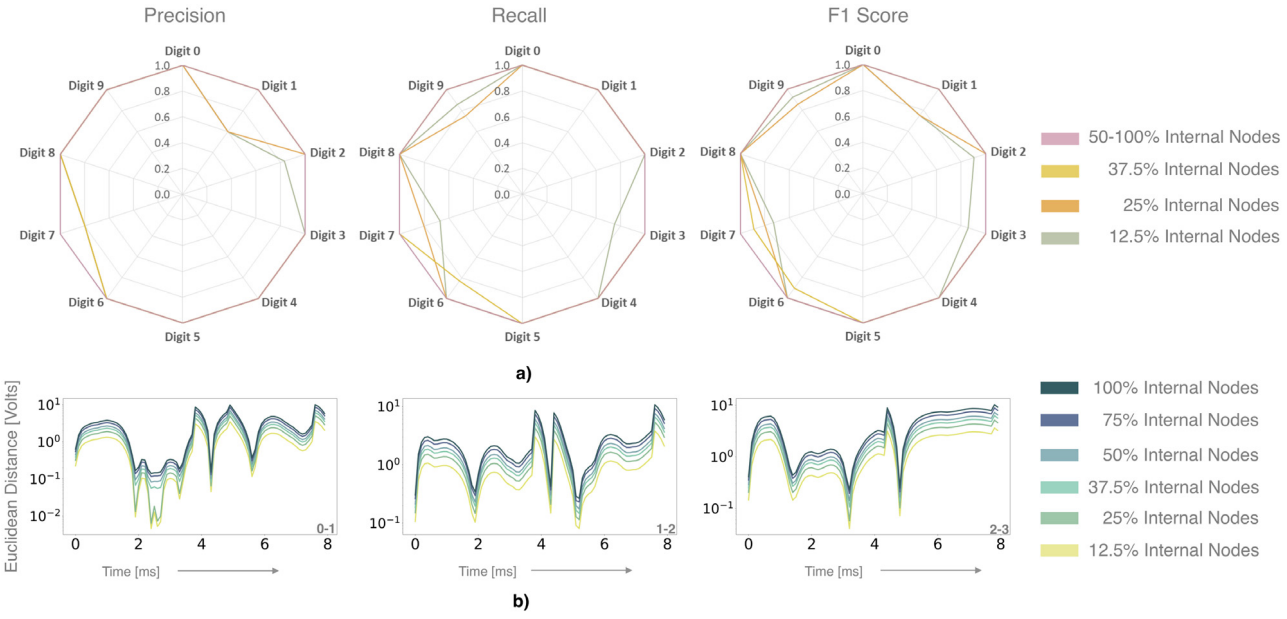

Fig. 10. Increase in the classification performance of the memristive reservoir, potentially induced by an increase in the separation property (SP) due to an augmentation in the dimensionality (i.e., the number of internal nodes used to describe the state of the system). The classification performance (in terms of precision, recall and F1 score for each digit label) and the separation property (SP), as a function of the number of internal nodes used to describe the state of the memristive reservoir are shown in (a) and (b), respectively.

distances between the original preprocessed input signals were also excluded because evidently the distance between a signal and itself will be in all cases zero.

A reservoir exhibits the FMP if variations introduced in the initial conditions of the network lead to a decaying distance between trajectories (in the phase space) of the network states produced by the same input (Legenstein & Maass, 2007; Maass, Natschlager, & Markram, 2004). Fig. 11 shows that, during the transient stage of the network's response (i.e. at approximately the first 4 ms), the Euclidean distance is higher compared to later stages. This means that, although the inputs are the same (in which case one would expect the network states to be the same), the network is able to 'remember' that the initial conditions were different by giving rise to temporarily different network states. Once the network stabilizes and reaches a cyclic steady state (i.e., from 4 to 15 ms approximately), this distance approaches zero, meaning that the network states tend to be more similar, and thus the reservoir is not able to 'remember' anymore the differences in the initial conditions, i.e. the memory fades away. Further, although the distance does not fully decay to zero, it is on average significantly lower (one to four orders of magnitude) compared to the distances between trajectories generated by two different input conditions (Fig. 9),

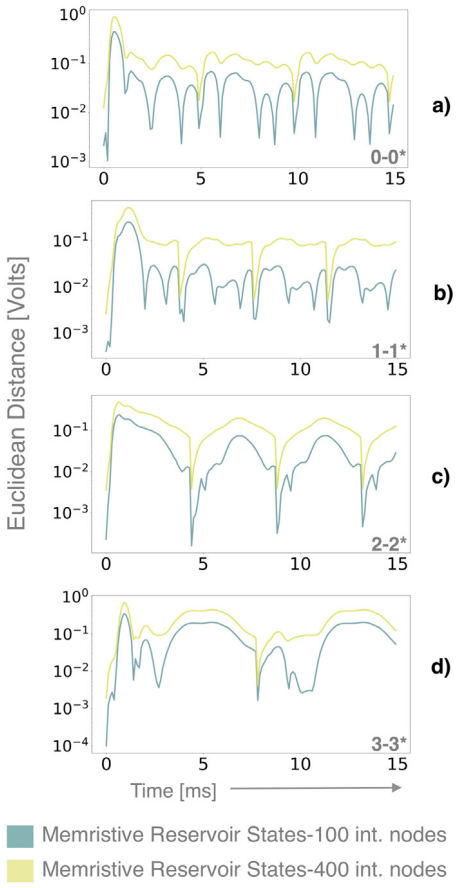

Fig. 11. Quantification of the fading memory property (FMP). Euclidean distance between the trajectories (in phase space) of the memristive reservoir using the voltage at the accessible internal nodes (dark green); and the voltage at both the accessible and non-accessible internal nodes (light green). Figures (a) through (d) show the distance between the reservoir states generated by each digit sequence ('0', '1', '2' and '3') and the same sequence under different initial conditions ('0*', '1*', '2*' and '3*'), i.e., different initial conductance states, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which is clearly a desirable behavior for any type of reservoir. These results suggest that the memristive reservoir is able to erase spurious information contained in the initial conditions of the system, and thus serves as evidence that the network possess the FMP.

Fig. 11 also shows that the number of internal nodes considered in the estimation of the distance between network states does not seem to have a significant effect on the time that it takes to the network to reach a steady state. This means that the stabilization time, as expected, does not depend on the number of variables included, but it is rather a property of the system that can be tuned with the parameters and connectivity of its constituent memristive elements.

3.4. Nonlinearity and the separation property

In Section 3.1 it was suggested that the improvement in the classification performance of the memristive reservoir could be potentially associated to an increase in the SP, induced by the inclusion of more internal nodes under consideration. However, it was also suggested that this increase in performance, driven by an increase in the SP of the system, is not entirely explained by the augmentation of the dimensionality, but it also depends on the degree of nonlinearity of the network.

To test the validity of this hypothesis, the same analyses carried out in Sections 3.1 and 3.2 were applied to a reservoir made-up entirely of resistive elements with the same architecture and similar electrical properties (i.e., conductance/resistance values) of the memristive version presented in Sections 3.1 and 3.2. This resistive reservoir was used as a control to control for nonlinearity. Results of these analyses are shown in Fig. 12.

In contrast to the results presented for the memristive version of the reservoir (Fig. 10(a)), Fig. 12(a) shows that the classification performance of the resistive reservoir is highly degraded as more internal nodes are included in the prediction of the digits' labels. Notwithstanding, Fig. 12(b) shows that the SP of the resistive reservoir, as measure by the Euclidean distance, increases with the number of internal nodes under consideration. These results prove that an increase in the SP, induced by the augmentation of the dimensionality, may be a necessary but not a sufficient condition to perform well in the speech recognition task.

In addition, an interesting observation from Fig. 12 is that the Euclidean distance between network states, given two distinct input sequences, is on average higher for the resistive reservoir, compared to the distance exhibited by the other systems laid out in this work (Fig. 9), such as the memristive and the leaky integrator reservoirs, which presented a higher classification performance (Fig. 7).

Although both the memristive and the resistive versions of the reservoir possess the exact same architecture (i.e., number of nodes and connectivity matrices) and physical parameters, the nature of their constituent operating units is fundamentally different. Unlike resistors, memristors are intrinsically nonlinear elements with memory. The poor performance of the resistive reservoir regardless of its high SP values, compared to the memristive version and the leaky integrator reservoir considered in the present work, is mainly because it does not exhibit memory properties, and its behavior is, by definition, linear.

To assess the level of nonlinearity of the MN^3 architecture, we used the method applied in Maass et al. (2002) to quantify the degree of nonlinearity in the response of a mixed network of linear and nonlinear resistors. This method is based on a physical phenomenon known as high harmonic generation (HHG) and is associated with systems that produce nonlinear frequency transformations. Therefore, we assessed whether the MN^3 architecture exhibits a nonlinear frequency response by looking for the presence of harmonics higher than the fundamental frequency of excitation of the input voltage in the output current of the network. This analysis was performed based on the Fourier transform of the signal.

For this purpose, a random memristive network with 2 external nodes (one source and one ground) and 400 (100 accessible and 300 non-accessible) internal nodes with a probability of connection $p = 0.90$ was simulated using a 10 Hz sinusoidal input signal with different levels of voltage amplitude. The HHG analysis was performed on the output current collected over 5 cycles of the input signal. For each amplitude level, the network was reset to the same initial state. For comparison purposes, this same analysis was carried out for the resistive version of the reservoir using the exact same architecture and physical parameters. Results are presented in Fig. 13.

Fig. 13 (top) shows that the dynamics of the memristive reservoir do promote the generation of higher harmonics in the output response of the system. This means that the memristive reservoir does perform a nonlinear projection of the input space into its network space, which is a desirable characteristic since it enhances the SP of the system. On the other hand, Fig. 13 (bottom) shows that the resistive reservoir is incapable of generating higher harmonics in its output response, which is an expected behavior, given the linear nature of its resistive constituent units.

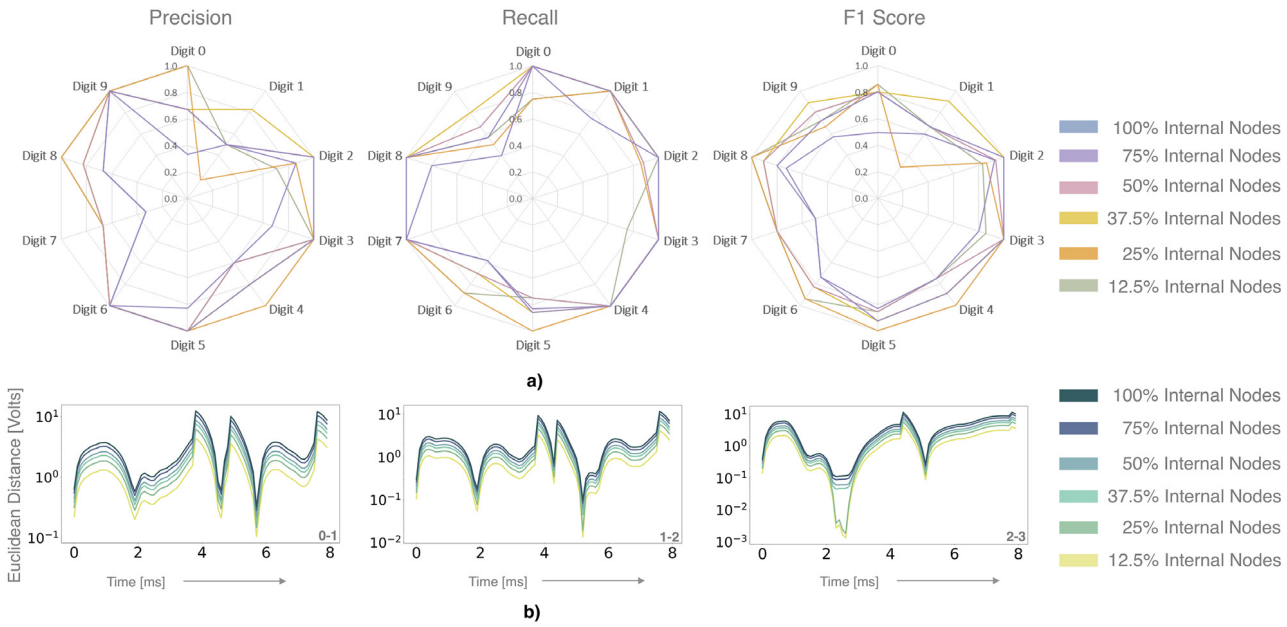

Fig. 12. Decrease in the classification performance of the resistive reservoir despite an increase in the separation property (SP) due to the augmentation of the dimensionality (i.e., the number of internal nodes used to describe the state of the system). The classification performance (in terms of precision, recall and F1 score for each digit label) and the separation property (SP), as a function of the number of internal nodes used to describe the state of the resistive reservoir are shown in (a) and (b), respectively.

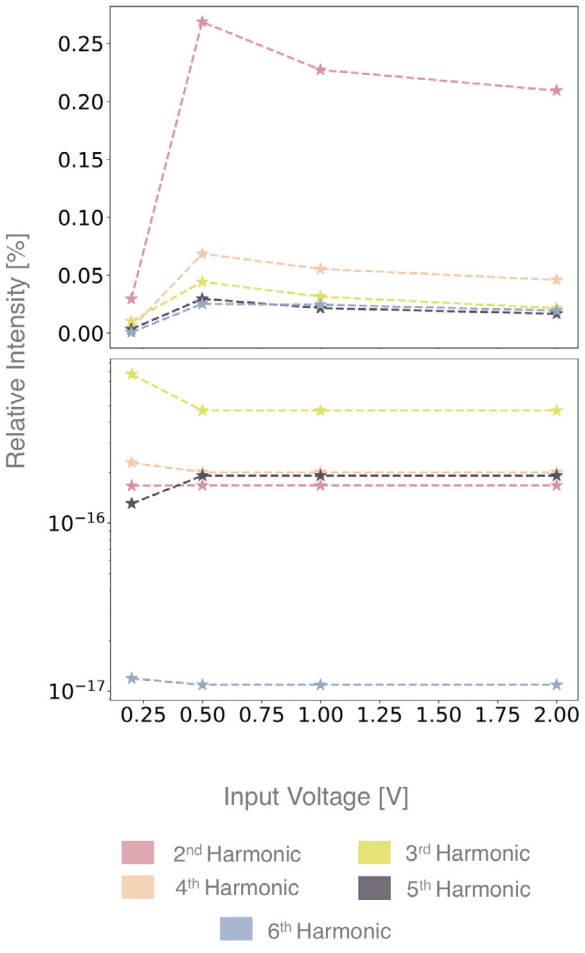

Fig. 13. Higher harmonic generation (HHG) analysis. Higher harmonics to fundamental intensity ratio for different input bias levels. Results of the HHG analysis for the memristive reservoir (top), and for the resistive reservoir (bottom).

In Fig. 13 (top), it can also be observed that the intensity of higher harmonics is attenuated by the amplitude of the input voltage. This decrease can be explained by the decline in the amount of hysteresis and nonlinearity of the memristors composing the reservoir. Fig. 14 shows the current–voltage (IV) curve of a single memristor for different amplitudes in the input voltage using a 100 Hz sinusoidal signal. As the amplitude of the input voltage increases, the memristor completes its set/reset process and does not change any further, temporarily exhibiting a lower hysteresis and thus less nonlinear behavior. This decrease in nonlinearity at the microscopic level suppresses higher harmonic generation at the global network scale (Oskoee & Sahimi, 2011; Sillin et al., 2013).

4. Conclusions and future directions

In this work, we simulated a random network of memristive elements as a reservoir for the solution of the isolated digit speech recognition task. We showed that the proposed architecture is not only capable of successfully carrying out this task, but it also outperforms the alternative proposed architectures. In addition, we showed that its performance can be explained by a combination of its separability, the presence of a fading memory and its nonlinear behavior. Overall, results from this work strongly suggest that the MN^3 architecture is a plausible alternative for the construction of kernels that operate under the principles of RC, and that, in order to perform more complex tasks, its performance could be potentially improved by tuning its parameters to increase its SP and FMP.

Since memristive networks are input-driven systems, their behavior largely depends on the characteristics of the input. In contrast with autonomous dynamical systems, whose dynamics are uniquely determined by the parameters of their constitutive elements, the dynamics of memristive networks are strongly determined by the nature and properties of the input. Thus, controlling their behavior and taking advantage of their dynamics for computational purposes requires adequate encoding techniques of the input signals. For future work, it may be interesting to study the effects of the type of input signal, as well as the effect of connectivity density and connectivity patterns between nodes on the overall behavior of the network, since these factors appear to have nontrivial effects (Kang, Goh, Lee, & Kim, 2004).

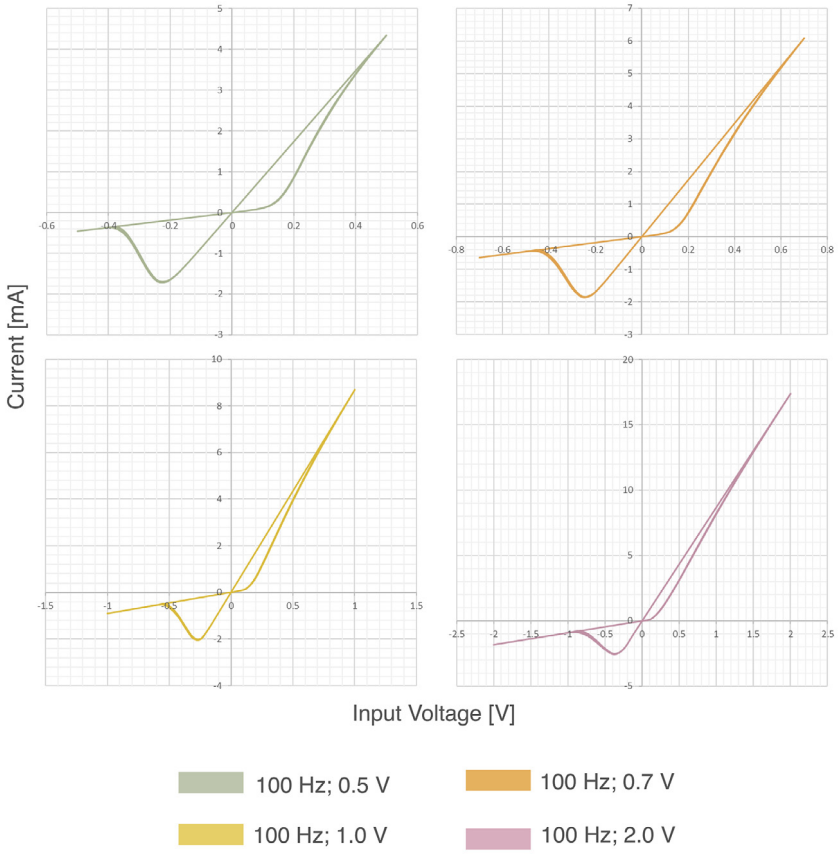

Fig. 14. Theoretical current–voltage (IV) curve of a single simulated memristor (using the model proposed in Nugent and Molter (2014) and explained in the Methods section), using a 100 Hz sinusoidal input voltage, for different amplitude levels. The behavior of a single memristor becomes more linear as the amplitude of the input voltage increases.

Although we introduced stochastic terms to induce device-to-device and temporal variability, we did not study the effect of these terms in the overall behavior and performance of the network. In the case of memristive networks these sources of variability are a critical factor since it is well known that the manufacturing processes of these devices introduce important device-to-device variations, and at the same time, these devices have shown to present a poor consistent behavior over repetitive trials (Bill & Legenstein, 2014).

Despite the simplicity of the methods presented here for the evaluation of the SP and the FMP, they are a good first approach to characterize the dynamic behavior of a specific reservoir architecture. However, we encourage the use of more sophisticated methods. For instance, the FMP of the reservoir could be better evaluated by assessing its performance in the NARMA task (Jaeger, 2003; Steil, 2005; Verstraeten, Schrauwen, d'Haene, & Stroobandt, 2007).

Acknowledgments

This work was partially supported by the Office of Technology Licensing at University of Florida (CF Award 15244). This material is based upon work supported by the U.S. National Science Foundation under Grant No. ECCS-1709641.

The authors gratefully acknowledge the collaboration provided by Golia Shafiei (McGill University) and Ross Markello (McGill University) in the design and formatting of the manuscript figures, as well as their proofreading and language editing.

References

Abdalla, H., & Pickett, M. D. (2011). SPICE modeling of memristors. In *2011 IEEE international symposium of circuits and systems (ISCAS)* (pp. 1832–1835).

Aleksander, I., & Morton, H. (1989). Analog VLSI and neural systems - mead, C, (in English). *Nature, Book Review*, *340*(6230), 198–198.

Avizienis, A. V., et al. (2012). Neuromorphic atomic switch networks. *PloS One*, *7*(8), e42772.

Beggs, J. M. (2008). The criticality hypothesis: how local cortical networks might optimize information processing. *Philosophical Transactions of the Royal Society A-Mathematical Physical and Engineering Sciences*, *366*(1864), 329–343.

Berdan, R., et al. (2013). Temporal processing with volatile memristors. In *2013 IEEE international symposium on circuits and systems (ISCAS2013)* (pp. 425–428).

Bill, J., & Legenstein, R. (2014). A compound memristive synapse model for statistical learning through STDP in spiking neural networks. *Frontiers in Neuroscience*, *8*.

Bürger, J., & Teuscher, C. (2013). Variation-tolerant computing with memristive reservoirs. In *2013 IEEE/ACM international symposium on nanoscale architectures (NANOARCH)* (pp. 1–6).

Chang, T., Jo, S.-H., Kim, K.-H., Sheridan, P., Gaba, S., & Lu, W. (2011). Synaptic behaviors and modeling of a metal oxide memristive device. *Applied Physics A*, *102*(4), 857–863.

Chapaneri, S. V., & Jayaswal, D. J. (2013). Efficient speech recognition system for isolated digits. *International Journal Computer Science and Engineering Technologies*, *4*(3), 228–236.

Chua, L. (1971). Memristor-The missing circuit element. *IEEE Transactions on Circuit Theory*, *18*(5), 507–519.

Doddington, G. R., & Schalk, T. B. (1981). Computers: Speech recognition: Turning theory to practice: New ICs have brought the requisite computer power to speech technology; an evaluation of equipment shows where it stands today. *IEEE Spectrum*, *18*(9), 26–32.

Dongale, T. D., et al. (2016). TiO2 based nanostructured memristor for RRAM and neuromorphic applications: a simulation approach. *Nano Convergence, Journal*, *3*(1), 16.

Goudarzi, A., Lakin, M. R., Stefanovic, D., & Teuscher, C. (2014). A model for variation- and fault-tolerant digital logic using self-assembled nanowire architectures. In *2014 IEEE/ACM international symposium on nanoscale architectures (NANOARCH)* (pp. 116–121).

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. In *Bonn, Germany: German national research center for information technology GMD technical report, Vol. 148* (p. 34).

Jaeger, H. (2003). Adaptive nonlinear system identification with echo state networks. In *Advances in neural information processing systems* (pp. 593–600).

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., & Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Letters*, *10*(4), 1297–1301.

Kang, B., Goh, K. I., Lee, D. S., & Kim, D. (2004). Complex networks: structure and dynamics. *Sae Mulli*, *48*(2), 115–141.

Kauffman, S. A. (1993). *The origins of order: Self organization and selection in evolution*. USA: Oxford University Press.

Kim, K. H., et al. (2012). A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications, (in English). *Nano Letters*, *12*(1), 389–395.

Konkoli, Z., & Wendin, G. (2013a). A generic simulator for large networks of memristive elements. *Nanotechnology*, *24*(38), 384007.

Konkoli, Z., & Wendin, G. (2013b). Toward bio-inspired information processing with networks of nano-scale switching elements, arXiv preprint arXiv:1311.6259.

Kudithipudi, D., Saleh, Q., Merkel, C., Thesing, J., & Wysocki, B. (2016). Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing. *Frontiers in Neuroscience*, *9*, 502.

Kulkarni, M., & Teuscher, C. (2012). Memristor-based reservoir computing. In *2012 IEEE/ACM international symposium on nanoscale architectures (NANOARCH)* (pp. 226–232).

Laiho, M., Lehtonen, E., Russel, A., & Dudek, P. (2010). Memristive synapses are becoming reality. *The Neuromorphic Engineer*, 1–3.

Langton, C. G. (1990). Computation at the edge of chaos: phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, *42*(1), 12–37.

Legenstein, T. (2005). At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks. In *Advances in neural information processing systems 17: Proceedings of the 2004 conference, Vol. 17* (p. 145). MIT Press.

Legenstein, R., & Maass, W. (2007). What makes a dynamical system computationally powerful. In *New directions in statistical signal processing: From systems to brain* (pp. 127–154).

Limkar, M., Rao, R., & Sagvekar, V. (2012). Isolated digit recognition using MFCC and DTW, Vol. 1, Mumbai University, India (pp. 59–64).

Linn, E., Siemon, A., Waser, R., & Menzel, S. (2014). Applicability of well-established memristive models for simulations of resistive switching devices. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *61*(8), 2402–2410.

Lyon, R., & Shamma, S. (1996). Auditory representations of timbre and pitch. In *Auditory computation* (pp. 221–270). Springer.

Maass, W., Natschlager, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations, (in English). *Neural Computation*, *14*(11), 2531–2560.

Maass, W., Natschlager, T., & Markram, H. (2004). Fading memory and kernel properties of generic cortical microcircuit models, (in English). *Journal de Physiologie (Paris)*, *98*(4–6), 315–330.

Mead, C. (1990). Neuromorphic electronic systems, (in English). *Proceedings of the IEEE*, *78*(10), 1629–1636.

Mead, C., & Ismail, M. (2012). *Analog VLSI implementation of neural systems*. Springer Science & Business Media.

Moore, G. (1965). Cramming more components onto integrated circuits. *IEEE Solid-State Circuits Society Newsletter*, *11*(5), 33–35.

Moore, G. (1975). Progress in digital integrated electronics. In *SPIE milestone series MS, Vol. 178* (pp. 179–181).

Morabito, F. C., Andreou, A. G., & Chicca, E. (2013). Neuromorphic engineering: From neural systems to brain-like engineered systems, (in English). *Neural Networks, Editorial Material*, *45*, 1–3.

Nino, J. C., & Kendall, J. D. (2017). Memristive nanofiber neural netwoks, US Patent 20170098156 A1.

Nugent, M. A., & Molter, T. W. (2014). AHaH computing-from metastable switches to attractors to machine learning. *PloS One*, *9*(2), e85175.

Oskoee, E. N., & Sahimi, M. (2011). Electric currents in networks of interconnected memristors, (in English). *Physical Review E*, *83*(3), 8, Art. no. 031105.

Pershin, Y. V., & Di Ventra, M. (2010). Experimental demonstration of associative memory with memristive neural networks. *Neural Networks*, *23*(7), 881–886.

Pickett, M. D., et al. (2009). Switching dynamics in titanium dioxide memristive devices. *Journal of Applied Physics*, *106*(7), 074508.

Sillin, H. O., et al. (2013). A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. *Nanotechnology*, *24*(38), 384004.

Snider, G. S. (2007). Self-organized computation with unreliable, memristive nanodevices. *Nanotechnology*, *18*(36), 365202.

Snider, G. S. (2008). Spike-timing-dependent learning in memristive nanodevices. In *2008 IEEE international symposium on nanoscale architectures* (pp. 85–92). IEEE.

Steil, J. J. (2005). Memory in backpropagation-decorrelation O(N) efficient online recurrent learning. In *Artificial neural networks: Formal models and their applications–ICANN 2005* (pp. 649–654). Berlin, Heidelberg: Springer Berlin Heidelberg.

Strukov, D. B., Snider, G. S., Stewart, D. R., & Williams, R. S. (2008). The missing memristor found. *Nature*, *453*(7191), 80–83.

Vandoorne, K., et al. (2014). Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature Communications*, *5*.

Verstraeten, D., Schrauwen, B., d'Haene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural Networks*, *20*(3), 391–403.

Verstraeten, D., Schrauwen, B., Dieleman, S., Brakel, P., Buteneers, P., & Pecevski, D. (2012). Oger: modular learning architectures for large-scale sequential processing. *Journal of Machine Learning Research (JMLR)*, *13*, 2995–2998.

Vourkas, I., & Sirakoulis, G. (2013). On the analog computational characteristics of memristive networks. In *2013 IEEE 20th international conference on electronics, circuits, and systems (ICECS)* (pp. 309–312).

Waldrop, M. (2016). More than Moore. *Nature*, *530*(7589), 144–147.

Waser, R., Dittmann, R., Staikov, G., & Szot, K. (2009). Redox-based resistive switching memories–nanoionic mechanisms, prospects, and challenges. *Advanced Materials*, *21*(25–26), 2632–2663.

Williams, R. S., Pickett, M. D., & Strachan, J. P. (2013). Physics-based memristor models, In *2013 IEEE international symposium on circuits and systems (ISCAS2013)* (pp. 217–220).

Yakopcic, C., Taha, T. M., Subramanyam, G., & Pino, R. E. (2013). Generalized memristive device SPICE model and its application in circuit design. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, *32*(8), 1201–1214.