

Investigations of the staircase memristor model and applications of memristor-based local connections

Xiao Yang¹ · Wanlong Chen¹ · Frank Z. Wang¹

Received: 30 April 2015 / Revised: 23 November 2015 / Accepted: 10 March 2016 / Published online: 31 March 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract The memristor is postulated by Chua (IEEE Trans Circuit Theory, 18(5):507–519, 1971) as the fourth fundamental passive circuit element and experimentally validated by HP labs in 2008. It is an emerging device in nano technology that provides low power consumption as well as high density. In this paper, the staircase memristor model is discussed and investigated with the HP memristor model. By connecting HP memristor models in series or parallel, a staircase memristor model could be constructed and demonstrate staircase behavior. By comparing the staircase memristor model to the general HP memristor model, distinctions between them are demonstrated and this lends themselves to different applications. Further to the memristor-based cellular neural network (CNN), the structure is modified and applied to the echo state network (ESN) where memristors are used as local connections. By this means, the ESN benefits from the simple structure, non-volatility and low power feature of memristors and therefore the complexity and size of the original ESN architecture can be reduced. Meanwhile, the simpler structure still has satisfactory performance in applications compared with the original ESN.

Keywords Nano technology · Memristor · Staircase · Delayed-switching effect · Ferroelectric memristor · Bio-inspired circuits · Cellular neural networks · Echo state networks

1 Introduction

Since the claim that the missing memristor was found [24], memristive devices have become one of the leading areas in nano technology because of its promising attributes of nano-scale size, low power feature and simple structure. Indeed, it also has been considered as one of the potential candidates to substitute the CMOS technology which is reaching the bottleneck in terms of size.

The memristor was first postulated as the fourth fundamental passive circuit element by Chua [8]. Following his concept, there are four fundamental circuit elements and each element represents a two-variable relationship between the four basic circuit variables, namely, charge q , current i , flux φ and voltage v . In addition to the known links which are the resistor, capacitor and inductor, the postulate of the memristor reveals the missing link between flux φ and charge q . However, a physical relation between φ and q is not necessary [9].

Since the memristor represents the relation between φ and q , it could also be expressed by the derivative of φ (voltage v) and the derivative of q (current i). For a charge-controlled memristor, the voltage across it is given by

$$\begin{cases} v(t) = M(q(t)) \cdot i(t) \\ M(q) = d\varphi(q)/dq \end{cases} \quad (1)$$

where $M(q(t))$ is called memristance and has the unit of Ohms. Likewise, a memristor can be controlled by the flux, which gives

$$\begin{cases} i(t) = G(\varphi(t)) \cdot v(t) \\ G(\varphi) = dq(\varphi)/d\varphi \end{cases} \quad (2)$$

where $G(\varphi(t))$ is called memductance which has the unit of Siemens. Thus, the memristance or memductance is

✉ Xiao Yang
Xiao.Yang@kent.ac.uk

¹ School of Computing, University of Kent,
Canterbury CT2 7NF, UK

determined by the time integral of voltage or current. A typical memristor has polarity which influences its memristance. A boarder class called memristive system was proposed later by Chua, in which, memristor is just a special case [10].

Beyond the postulate, the memristor has been found in nano scale by several research groups utilizing different materials such as the titanium-dioxide based memristor [24], the ferroelectric memristor [5], the tungsten-oxide based memristor [4] and the diamond-like carbon based memristor [6]. The memristor with staircase behavior is firstly introduced by Chua [8] and emulated by non-linear resistors and Zener diodes. However, with the development of the memristor, several multi-state memristors were found with intrinsic multilevel resistance, such as the FeO_x based memristor which can be tuned and controlled by external electric conditions [3]. Moreover, the Cu_xO based memristor can also provide multilevel resistive switching [27]. Especially in the ferroelectric memristor, the variation of its state variable is like a staircase by applying a consecutive pulse signal. Such memristors with staircase behavior is distinct from general memristors and exhibit several delays during the switching. In ferroelectric memristors, it is argued that the ferroelectric domain dynamics dominates the variations and thus the wavy variation signals the presence of several areas with different dynamics. In this case, the peculiar delay behavior is caused by nucleation effects in ferroelectric barriers since nucleation centers need to be activated, which yields delays.

Such memristors are particularly useful in some applications which are introduced in [30] where it presents using staircase memristors in generating staircase waveform and cellular neural networks (CNN). In more general cases, applications of the memristor lie in the fields of hybrid dynamic random-access memories (DRAM), content-addressable memories (CAM), programmable logic circuits and neural networks. One of HP laboratories [2] announced the development of the memristor crossbar in computer memory and its ability of operating logic computation, which leads researchers to pursuing more potential applications of memristors. Recently, a memristor based CAM cell [7, 14] was proposed to improve the density and power consumption of CAM. Beyond CAM, the memristor is a potential element to mimic the synapses in neural networks with the benefits of nano-scale structure and non-volatility [19, 20, 29]. It works as the connection sites and represents the connection strength between neurons.

In this paper, the staircase memristor model is further investigated with the HP memristor model to develop two distinct models. In order to show distinctions between the staircase memristor model and the HP memristor model, a comparison is taken by programming both models to some specific resistance values in software simulations. Further

to the proposed CNN structure with memristive connections, the structure is modified and applied to echo state networks (ESN) as the local connections of neurons in the reservoir which is a collection of states of neuron activations between input and output layers. By utilizing memristor-based CNN structure as local connections between neurons, the connectivity complexity can be significantly reduced while maintaining a satisfactory performance.

2 The HP memristor model

Although there is a plenty of variations of memristor models, the primary one in use is the memristor model proposed by Strukov et al. [24]. The HP memristor is fabricated by two Pt nodes and a thin semiconductor film which is sandwiched between nodes as shown in Fig. 1. Within the semiconductor film, it consists of an undoped, insulating TiO_2 layer and a doped oxygen-poor TiO_{2-x} layer. In this case, the effective transport mechanism in titanium-dioxide based memristor devices is through the drift of oxygen vacancies originating within an oxygen deficient layer of TiO_{2-x} and therefore shifts the dividing boundary between TiO_2 and TiO_{2-x} layers. Specifically, the semiconductor has a region with a high concentration of dopants having low resistance R_{on} and conversely the remainder has a low dopant concentration having much higher resistance R_{off} . By applying an external bias voltage, the boundary between doped and undoped regions will move towards the undoped region and therefore the width w of the doped region will increase until it reaches the total width D of the semiconductor film and switches to low resistance R_{on} .

Based on observations and experiments, the following Eq. (3) is proposed by Strukov et al. [24] to model the HP memristor.

$$v(t) = \left(R_{on} \frac{w(t)}{D} + R_{off} \left(1 - \frac{w(t)}{D} \right) \right) i(t) \quad (3)$$

where $w(t)$ is the width of doped region at time t . D is the full width or thickness of the semiconductor film. $i(t)$ and $v(t)$ imply the applied current and voltage which pass through the memristor. However, it is not sufficient to describe the behaviors of HP's memristor since the term

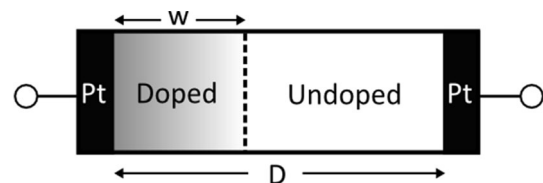


Fig. 1 Schematic of the titanium-dioxide based memristor proposed by HP

$w(t)$ is unknown. Hence, another Eq. (4) was given to define $w(t)$

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{on}}{D} i(t) \tag{4}$$

where μ_v is the average ion mobility.

Since μ_v , R_{on} and D are constant parameters, it exhibits linear ionic drift in the film. However, in nano-scale devices, a small voltage can produce significant non-linearities in ionic transport. Thus, to model the boundary condition and non-linear drift when w is approaching either boundaries of the device, a window function $f(x)$ is often multiplied to (4) which gives

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{on}}{D} f(x) i(t) \tag{5}$$

By multiplying the window function $f(x)$, w will drift non-linearly when w is approaching either boundary of 0 or D . In order to model different and more sophisticated memristor dynamics, several models were proposed in the literature based on the HP memristor with different window functions such as Joglekar’s model [16], Biolek’s model [1], the boundary condition memristor (BCM) model [13] and the threshold adaptive memristor (TEAM) model [18]. However, the window function used by HP is investigated in this paper.

2.1 The HP memristor with a single stair

Based on the experimental results of the existing HP memristor model with non-linear drift, it exhibits the staircase behavior because of the boundary effect. Therefore, by applying the window function $w(1-w)/D^2$ proposed in [24] to replace $f(x)$ in the right side of (5), it leads to the following equation

$$\frac{dw}{w(1-w)} = \mu_v \frac{R_{on}}{D} i \frac{1}{D^2} dt \tag{6}$$

Then, both sides could be integrated which yields

$$-\ln \left| \frac{-w+1}{w} \right| + C = \mu_v \frac{R_{on}}{D^3} q \tag{7}$$

Since D is normalized to 1 and $0 < w < 1$, the absolute symbol can be removed. If we assume that the initial value of charge q is $q_0 = q(0) = 0$, we have

$$\frac{-w+1}{w} = e^{-q\mu_v \frac{R_{on}}{D^3} + C} \tag{8}$$

Finally, by simplifying (8), the state variable s is obtained and has the following relation with charge q

$$s = w = \frac{1}{1 + e^{-qk_p + \rho}} \tag{9}$$

where $k_p = \mu_v \frac{R_{on}}{D^3}$ and ρ is the constant C . In this case, the

state variable s models the boundary effect of the memristor and will be used in the proposed staircase memristor model.

The derivative of (9) gives the change rate of state variable s

$$\frac{ds}{dq} = \frac{k_p e^{-qk_p + \rho}}{(1 + e^{-qk_p + \rho})^2} \tag{10}$$

where k_p denotes the propagation speed of charge in memristor. ρ/k_p is a constant term which determines the middle point of the transition period of the memristor between ON and OFF. By substituting (9) into

$$R(q) = sR_{on} + (1-s)R_{off} \tag{11}$$

and

$$v(t) = R(q)i(t) \tag{12}$$

the HP memristor model is obtained with a single stair which is controllable by varying the parameters k_p and ρ . If k_p is a large number, it requires less charges to change the state of the memristor. By varying ρ , a virtual threshold (where the state begins to change much more) is controlled and therefore a larger ρ results in that more charges are required to switch the memristor. Because of the boundary effect, state of the HP memristor slowly evolves when it approaches either boundary. This effect is useful in modeling a staircase memristor since it could mimic a single stair in a staircase memristor.

3 The staircase memristor model

The concept of the staircase memristor model derives from the “delayed-switching effect” of piece-wise linear memristors [25]. This effect indicates that switching in a memristor takes place with a time delay because the memristor possesses certain inertia [26]. A staircase memristor model is considered to have a delayed-switching effect between several somewhat stable resistance levels, and hence the variation of its state is like a staircase. In practice, memristors with multi-level resistance are observed in [3, 5, 27] and a theoretical SPICE model was proposed in [28]. In particular, in ferroelectric memristors, significant delays of resistive states are observed. These delays are beneficial to some applications, such as programming or maintaining a memristor at a specific and stable value. Therefore, in this section we demonstrate a staircase memristor model inspired by the ferroelectric memristor. The advantage of the proposed model is it could show the explicit relation between flux ϕ and charge q which is very important for memristor studies.

3.1 Modelling a staircase memristor

A staircase memristor model is obtained by dividing the $q - \varphi$ curve into several linear segments, which implements a piece-wise linear memristor. Since the slope of the $q - \varphi$ curve denotes the memristance of the memristor, the same number of stairs on its memristance can be observed as shown in Fig. 2(a).

If the transition period from ON to OFF or OFF to ON of a memristor consists of a number of linear segments, the same number of stairs on its memristance or memductance can be observed. Therefore, the $q - \varphi$ curve can be partitioned into several segments with different slopes, as shown in Fig. 2(a), and staircase behavior is observed. The resulting curves such as I–V curve shown in Fig. 2(b) is better than practical memristors with multilevel resistance such as [3, 5, 27] because of the limitation of theoretic studies. Due to the lack of data of practical memristors, the comparison between the model and practical memristors is limited. However, the staircase behavior shown in the ferroelectric memristor can be reproduced by proposed model as shown in Fig. 2(c).

In case of a staircase memristor, there will be several such regions which leads its state varies like a staircase. Hence, assuming all the regions have the same characteristics, for example the same width and propagation speed, it gives the following equation:

$$s = \sum_{i=1}^N i \frac{S_{max}}{N} + s(i) \tag{13}$$

$$= \sum_{i=1}^N i \frac{S_{max}}{N} + \frac{S_{max}}{N} \frac{1}{1 + e^{-qk_p + \rho + Q_{min}^i}}$$

where s is the state variable of a staircase memristor and varies from “0” to “1”. Theoretically, more than one region is prohibited to activate at same time in order to produce proper staircases.

Therefore, a heaviside function is multiplied to (13) which gives

$$s = \sum_{i=1}^N H(i) \left(i \frac{S_{max}}{N} + \frac{S_{max}}{N} \frac{1}{1 + e^{-qk_p + \rho + Q_{min}^i}} \right) \tag{14}$$

where

$$H(i) = H(| -k_p q | - Q_{min}^i) \tag{15}$$

S_{max} is the maximum value of the state variable s . N is the number of the regions in the memristor. Q_{min}^i denotes the minimum quantity of charge required to enter current region i . If the total charge exceeds Q_{min}^i , region i is activated. In particular cases where the regions have different proportions of total thickness, the term $i \frac{S_{max}}{N}$ will be replaced by a varied number according to the proportion of the region i .

3.2 The series and parallel forms of staircase memristor models

Following the equations of the state variable s , there are two distinct approaches to construct a staircase memristor model. In order to demonstrate the two distinct approaches, a simplified conceptual schematic diagram is shown in Fig. 3, several HP memristor models with a single stair could be connected in series or parallel to construct a staircase memristor model. This schematic diagram gives

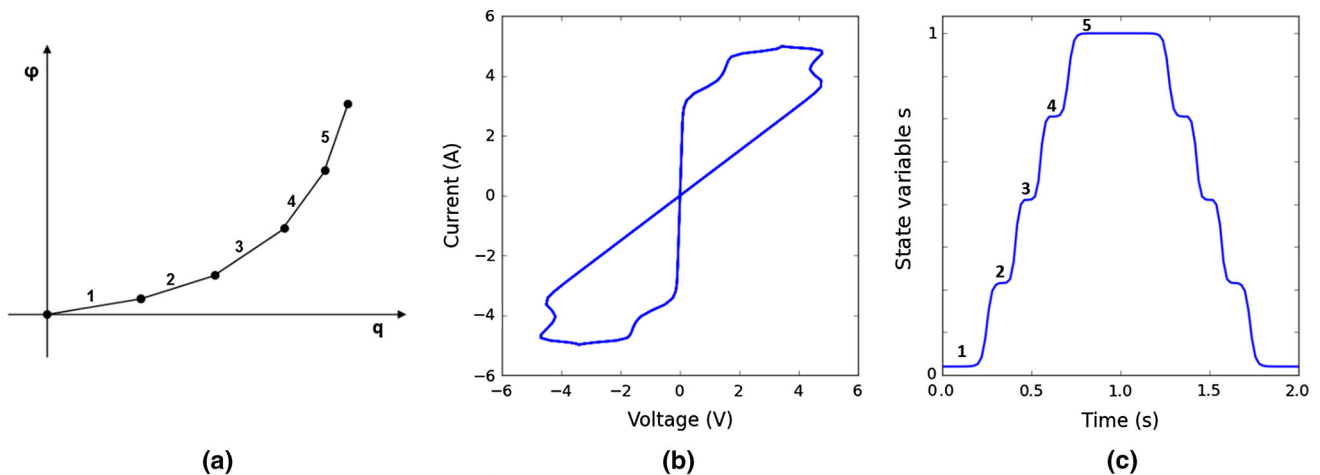


Fig. 2 Simulation results of the staircase memristor model. **a** A $q - \varphi$ curve is divided into five segments which represent five memristance values. It means this staircase memristor has 4 regions which are region 1: {1 → 2}, region 2: {2 → 3}, region 3: {3 → 4}, region 4: {4 → 5}. **b** A pinched hysteresis loop of current and voltage of

staircase memristor. **c** By applying a periodic sinusoidal signal, the state variable s of the staircase memristor varies like a staircase. Parameters used here are: $N = 4, Q_{min}^1 = 0, Q_{min}^2 = 14, Q_{min}^3 = 28, Q_{min}^4 = 42, k_p = 20, \rho = 10, R_{off} = 40, R_{on} = 1$.

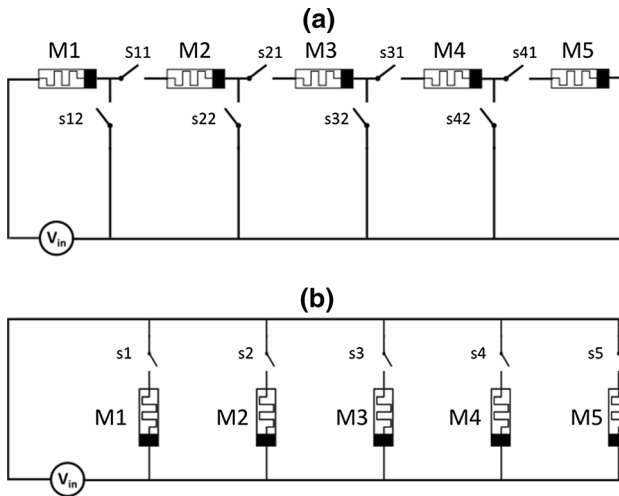


Fig. 3 Schematic of conceptual circuits of staircase memristor models in series and parallel forms

the idea of the two approaches rather than actual circuit implementations. In practice, a numerical simulation is applied by using Python and Numpy instead of Spice.

In Fig. 3(a), it demonstrates a staircase memristor model with 6 resistance levels, and each HP memristor model is activated one by one. At the first, switch s_{12} is turned on and s_{11} is turned off and therefore only the memristor M1 is connected to the voltage source. When M1 is switched from ON to OFF, switch s_{12} is turned off. Then the switches s_{11} and s_{22} are turned on, which only connects M2 to the voltage source besides M1. Until all the memristors are connected and switched to the OFF state, a staircase memristor model with 6 resistance levels is achieved. This mechanism could be described by the following equation.

$$R(q) = sR_{on} + (1 - s)R_{off} \tag{16}$$

In contrast, HP memristor models are connected in parallel in Fig. 3(b). In this case, the switch s_1 is connected to the

voltage source initially. When M1 is switched from ON to OFF, switch s_2 is turned on to connect M2 to the voltage source. Until all the memristors are connected and switched their states, a staircase memristor model is achieved. Since the memristors are connected in parallel, it could be described by

$$G(q) = sG_{on} + (1 - s)G_{off} \tag{17}$$

In a word, the conceptual circuit works somewhat like a digital potentiometer which is built by the classical HP memristors.

3.3 Comparisons of HP and staircase memristor models

A comparison between a staircase memristor model and a general HP memristor model is shown in Table 1, which describes the errors between the expected resistance levels and the actual resistance levels obtained by applying the same pulse signal with an amplitude $A = 5\text{ V}$ and a duty cycle $D_c = 0.5$. The errors ϵ is measured by

$$\epsilon = \frac{|R_a - R_e|}{R_e} \times 100 \tag{18}$$

where R_a is the actual resistance obtained and R_e is the expected resistance. The HP memristor model proposed in [24] and described by (3) and (4) is used for comparison with parameters $R_{on} = 1\ \Omega$, $R_{off} = 400\ \Omega$, $\mu_v = 5 \times 10^{-2}\text{ m}^2\text{s}^{-1}\text{V}^{-1}$ and the width D is normalized to 1. In contrast to general HP memristor model, a staircase memristor model proposed in this paper and defined by (14) with 5 resistance levels is used. It models a staircase memristor containing five resistance levels according to the ferroelectric memristor and other multi-level memristive devices which exhibit five-level resistance states [5, 27].

Both the staircase memristor model and the general HP memristor model are programmed towards the expected

Table 1 Staircase memristor model vs general HP memristor model. It measures differences between expected resistance values and actual resistance values of both staircase and general HP memristor models in percentage. The base frequency f of the applied pulse signal is $\frac{1}{2\pi}$ with a duty cycle $D_c = 0.5$

Resistance Expected	R = 100 Ω		R = 200 Ω		R = 300 Ω	
	Staircase (%)	HP (%)	Staircase (%)	HP (%)	Staircase (%)	HP (%)
Frequency (Hz)						
$10f$	0.74	1.5	0.245	0.5	0.017	0.167
$9f$	0.74	6.5	0.245	1.0	0.007	1.33
$8f$	0.74	7.0	0.245	2.25	0.003	0.67
$7f$	0.74	4.5	0.245	3.0	0.1	2.33
$6f$	0.74	4.0	0.245	4.75	0.11	1.5
$5f$	0.74	1.0	0.245	0.495	0.1	0.0
$4f$	0.74	14.5	0.245	2.25	0.11	3.33
$3f$	0.72	13.5	0.245	4.5	0.1	1.33
$2f$	0.71	14.5	0.245	10.0	0.11	8.5
f	0.72	13.5	0.185	25.5	0.1	8.33

resistance levels $R = 100 \Omega$, $R = 200 \Omega$ and $R = 300 \Omega$. By decreasing the frequency of the pulse signal from $10f$ to f , the error of the general HP memristor grows significantly. It shows that the general HP memristor model is very sensitive to the frequency, however, the staircase memristor model's resistance is very reliable with a much smaller fluctuation around the expected resistance level. By varying the expected resistance level, the staircase memristor model still has reliable performance. In contrast, the errors of the general HP memristor model fluctuate a lot because of the non-linearity. As mentioned previously, the change rate of memristance is a non-linear function with respect to charge q and therefore if the expected resistance is quite far from either boundary, the memristance changes significantly. In this case, since a small increment in charge q will result in a large increment in actual resistance, low frequency signal yields large error.

In this comparison, the frequency range from $\frac{1}{2\pi}$ to $\frac{5}{\pi}$ is selected to show the significant difference between HP and staircase models. In fact, the staircase model can work in a much higher frequency and the accuracy will be improved for both the HP model and the staircase model as observed in Table 1 when the frequency increases. However, staircase model still outperforms HP model in this case. The comparison result implies that the staircase memristor is more reliable than the general HP memristor if a specific resistance level is required.

4 Staircase memristors in CNN circuits

CNN is biologically motivated neural networks and important for applications in practice [11, 12]. The mainly uniform processing elements, called cells or artificial neurons, are placed on a regular geometric grid (with a square, hexagonal, or other patterns). The structure of CNN is defined as “Any cell in a cellular neural network is connected only to its neighbor cells” [12]. Using staircase memristors in CNN circuit is worth investigating because it allows a easier way to program the CNN template as well as high density. Figure 4 shows the schematic diagram of using staircase memristors in a CNN circuit.

Staircase memristors are used as the templates of the CNN circuit and represent the connections or the weights between CNN cells. In Fig. 4, all the inputs and outputs of neighbor cells are weighted by staircase memristors and then summed separately. The weighted and summed inputs and outputs of neighbor cells will contribute to the state of the cell (i, j) . In order to program the staircase memristor according to different applications, the switch S3 and S4 will be disconnected to isolate staircase memristors from the CNN circuit. It avoids the influence from the CNN cell.

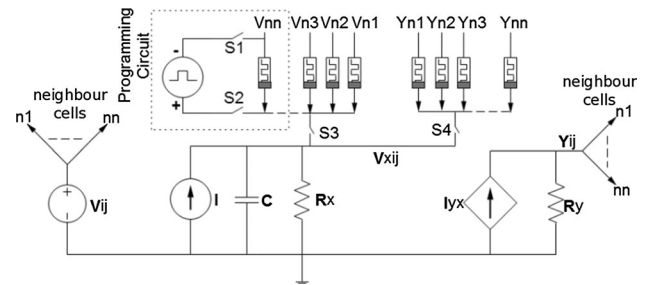


Fig. 4 A schematic of using staircase memristors in the CNN circuit. Each staircase memristor is a connection between the cell (i, j) and one of its neighbors. Cell (i, j) receives all the weighted and summed inputs and outputs from its neighbors and propagate its input V_{ij} and output Y_{ij} to all its neighbors. When programming a staircase memristor, S3 and S4 will be turned off to isolate staircase memristor to avoid influence from CNN circuit. S1 and S2 will be turned on to connect staircase memristors to the programming circuit

S1 and S2 will be turned on to connect the required staircase memristor and the provided pulse signal. Hence, the memristance of staircase memristor can be varied by controlling the duration of the pulse. After all, the CNN cell interacts with its neighbors via the programmed memristive templates until the template has to be changed again.

Based on the CNN circuit in Fig. 4, the software based simulation was presented in [30] to simulate the CNN circuit with staircase memristors in typical machine vision tasks of noise removal and edge detection. The proposed circuit structure with memristors can be adapted to other networks such as echo state networks and therefore the neurons are locally connected by memristors.

5 The memristor-based CNN structure in reservoir computing

Reservoir computing is an exciting approach that aims to overcome the training problem that exists in traditional recurrent neural networks (RNNs). It is well-known that training RNNs is inherently difficult even with the important yet powerful error back-propagation (BP) algorithm. It is a time-consuming and computationally expensive job to train RNNs, however there is still a possibility that the training may fail to converge. In the paradigm of reservoir computing, a “reservoir” is a collection of states of neuron activations between the input and output layers. It is generated with random connection weights and used to extract features from the input signals. Distinct from other neural networks, only the readout weights between the “reservoir” and the output layer are trained. The term “reservoir computing” comes mainly from the echo state network (ESN) [15] and the liquid state machine (LSM) [22] which share the concept of a “reservoir”. In principle, a

“reservoir” is an excitable, dynamical medium and plays an important role in reservoir computing networks. Theoretically, any dynamical systems with rich dynamics are capable of building a reservoir. Since a memristive system is also a non-linear dynamical system, using memristors as reservoir components in the ESN has been investigated by [17]. The graph-based approach is used to represent the reservoir network implemented by memristors. However, we propose an echo state network that is based on the memristive CNN structure where memristors are used as the local connections between nodes in the reservoir.

5.1 The reservoir with memristor-based local connections

In the original ESN, the given training input signal and target output signal are defined by $\mathbf{u}(n) \in \mathbb{R}^{N_u}$ and $\mathbf{y}^{target}(n) \in \mathbb{R}^{N_y}$ respectively. n is the discrete time in the dataset with values $n = 1, 2, 3, 4, \dots$. N_u and N_y are the number of inputs and outputs in the network respectively. The components of the reservoir are RNN type units with leaky-integrated discrete-time continuous values. The typical update equations are

$$\tilde{\mathbf{x}}(n) = \tanh(\mathbf{W}^{in}[1; \mathbf{u}(n)] + \mathbf{W}\mathbf{x}(n - 1)) \tag{19}$$

where $\tilde{\mathbf{x}}$ denotes the update of reservoir components, which collects both the inputs and the states of other units. $[1; \mathbf{u}(n)]$ denotes the vertical vector concatenation of vectors 1 and $\mathbf{u}(n)$.

The new states of the units are defined by

$$\mathbf{x}(n) = (1 - \alpha)\mathbf{x}(n - 1) + \alpha\tilde{\mathbf{x}}(n) \tag{20}$$

where $\mathbf{x}(n) \in \mathbb{R}^{N_x}$ is a vector of reservoir neuron activations at time step n . α is the leaking rate of the neuron, which is normally within the range (0, 1]. \mathbf{W}^{in} is the input weight matrix containing the connection weights between inputs and the reservoir neurons, thus it has the size of $N_x \times (1 + N_u)$. \mathbf{W} is the recurrent weight matrix which consists of connection weights between the reservoir neurons and has the size of $N_x \times N_x$, which implies that the reservoir neurons are fully connected.

The output \mathbf{y}_n is defined by

$$\mathbf{y}_n = \mathbf{W}^{out}[1; \mathbf{u}(n); \mathbf{x}(n)] \tag{21}$$

Thus, the output weight matrix \mathbf{W}^{out} has a size of $N_y \times (1 + N_u + N_x)$. So far, the work-flow of original ESN is defined, and there are 3 main differences compared to the CNN:

1. the network is randomly connected instead of locally connected
2. the network weights are randomly generated instead of a space-invariant template

3. the output is a linear function instead of a piece-wise linear function

Since the memristor-based CNN structure is used as the reservoir, only the reservoir network is adjusted to adopt the proposed structure.

From the definitions of the states of the units in (20) and its update in (19), the state vector $\mathbf{x}(n)$ is determined by its previous state $\mathbf{x}(n - 1)$, the input $\mathbf{u}(n)$ and states of other units. Thus, according to the definition of CNN, the reservoir network is redefined to have a regular geometric grid and local connections by

$$\sum_{k=1}^M \sum_{l=1}^N \mathbf{W}^x(i, j; k, l)\mathbf{x}(n - 1) \tag{22}$$

where we assume that the reservoir cell (i, j) has a neighborhood size of $M \times N$ neighbors and then the update equation (19) can be rewritten as

$$\tilde{\mathbf{x}}(n) = \tanh(\mathbf{W}^{in}[1; \mathbf{u}(n)] + \sum_{k=1}^M \sum_{l=1}^N \mathbf{W}^x(i, j; k, l)\mathbf{x}(n - 1)) \tag{23}$$

where \mathbf{W}^x is the matrix that denotes the local connections and is implemented by the memristors. This structure is slightly different from the traditional CNN which has a feedback loop containing the outputs of the CNN neurons. The feedback loop in traditional CNN is taken out because the reservoir size is independent of the input size or output size which may not have neighbors. Based on the proposed approach, the basic network is illustrated in Fig. 5 where the reservoir is implemented using local connections. If a reservoir has 100 neurons, the original ESN has 100×100 connections, however, this approach only has 100×8 connections. Therefore, the required connections are significantly reduced.

5.2 The benchmark task

In order to evaluate the performance of the proposed memristor-based reservoir with CNN structure, we use the Mackey–Glass time series dataset in this task. The dataset

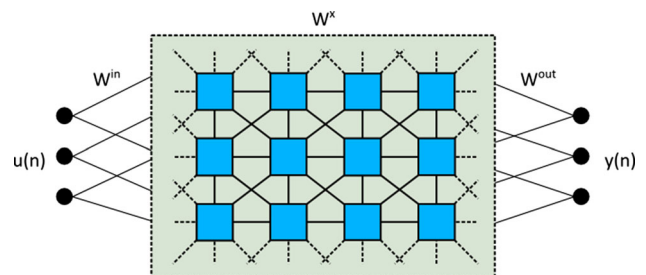


Fig. 5 A reservoir with local connections which are implemented by memristors

is generated from the Mackey–Glass equation which is a non-linear time delay differential equation defined by

$$dm/dt = \beta \frac{m_\tau}{1 + m_\tau^n} - \gamma m \quad (24)$$

where m_τ is the value of m at the time $(t - \tau)$ and τ denotes the delay of the Mackey–Glass system. This equation is used by [23] to describe the physiological control system where m denotes the concentration of circulating blood cells. However, we only focus on the data itself rather than the physiological representations. The parameters selected to generate the required dataset are $\beta = 0.2$, $\gamma = 0.1$, $n = 10$ and $\tau = 17$ which gives mild chaos. In this task, the network aims to learn the generated dataset and predict the future values after training.

5.3 Experimental setup

Before the experiment, the dataset is divided into two separate parts which are the training set and test set. Each set contains 2000 values but only the values in the training set are used for training the network. The test set is used for comparing the actual results, thus evaluating the performance of the network in this prediction task. The whole process of the experiment of memristive ESN is:

1. generating a reservoir with the size of 32×32
2. programming the memristive connections to random values in the range of $[-0.5, 0.5]$
3. running the training set and collecting the activation states of reservoir neurons using (23) and (20) with a leakage rate $\alpha = 0.4$
4. training the readout weights using ridge regression with a regularization coefficient 1.0×10^{-8}
5. running the test set and evaluating the performance using mean-squared error (MSE)

Since the Mackey–Glass equation only generates a time-series dataset, the network only has one input and one output. For the purpose of comparison, the original ESN is generated using the Python code developed by Mantas [21].

5.4 Results

In order to evaluate the performance of the memristor based ESN with CNN structure (MCNN ESN), 10 running results are obtained for the proposed ESN with memristive CNN structure, the original ESN structure with memristive connections and the original ESN as shown in Table 2 using Python 2.7, Oger toolbox 1.1.3 and script developed by Mantas [21]. The results are measured using mean-squared error (MSE) as shown in (25) which computes the

Table 2 Simulation results of all ESN networks in 10 trials

Test no.	MCNN MCNN ESN	Memristive ESN	Original ESN
1	2.04×10^{-2}	2.95×10^{-2}	1.57×10^{-2}
2	1.11×10^{-2}	2.02×10^{-2}	1.94×10^{-2}
3	1.69×10^{-2}	1.46×10^{-2}	1.16×10^{-2}
4	2.15×10^{-2}	1.00×10^{-2}	1.60×10^{-2}
5	3.61×10^{-2}	1.84×10^{-2}	9.22×10^{-3}
6	1.93×10^{-2}	1.40×10^{-2}	1.29×10^{-2}
7	2.12×10^{-2}	3.22×10^{-2}	3.02×10^{-2}
8	2.34×10^{-2}	1.47×10^{-2}	2.74×10^{-2}
9	3.06×10^{-2}	1.50×10^{-2}	2.16×10^{-2}
10	1.93×10^{-2}	3.03×10^{-2}	1.67×10^{-2}
Average	2.20×10^{-2}	1.99×10^{-2}	1.81×10^{-2}

In all cases, connection weights are randomly generated in the range of $[-0.5, 0.5]$. Results are measured by mean-squared error (MSE)

differences between the predicted results and the test set of Mackey–Glass dataset.

$$MSE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (\hat{Y}_i - Y_i)^2 \quad (25)$$

It is noticed that, in Fig. 6(a), the predicted signal is somewhat shifted from the target signal and thus a relatively large error is expected in this experiment. For all cases, the average MSE is around 2.0×10^{-2} in Table 2. Therefore, the average root-mean-squared error (RMSE) is around 1.4×10^{-1} . In order to measure the average RMSE to the scale of the target signal, the normalized RMSE can be obtained by

$$\frac{RMSE}{|m_{max} - m_{min}|} \quad (26)$$

where m_{max} and m_{min} are the maximum and minimum values of the target signal respectively. It gives that the normalized RMSE is approximately 17.5% which confirms the observed error in Fig. 6(a).

By setting the same reservoir size with 1024 neurons and the same leaking rate $\alpha = 0.4$, it shows that the original ESN has a better performance than both the ESN with memristive connections and the proposed ESN. The potential reasons for the reduced accuracy might be three-fold. First, due to the weight accuracy of memristive connections, actual weights of memristive connections are slightly different from expected values as shown in Table 1. By comparing the results of the original ESN and the same ESN with memristive connections, the limitation in weight accuracy explains that the original ESN has better performance than the ESN with memristive connections since there is no difference in the network

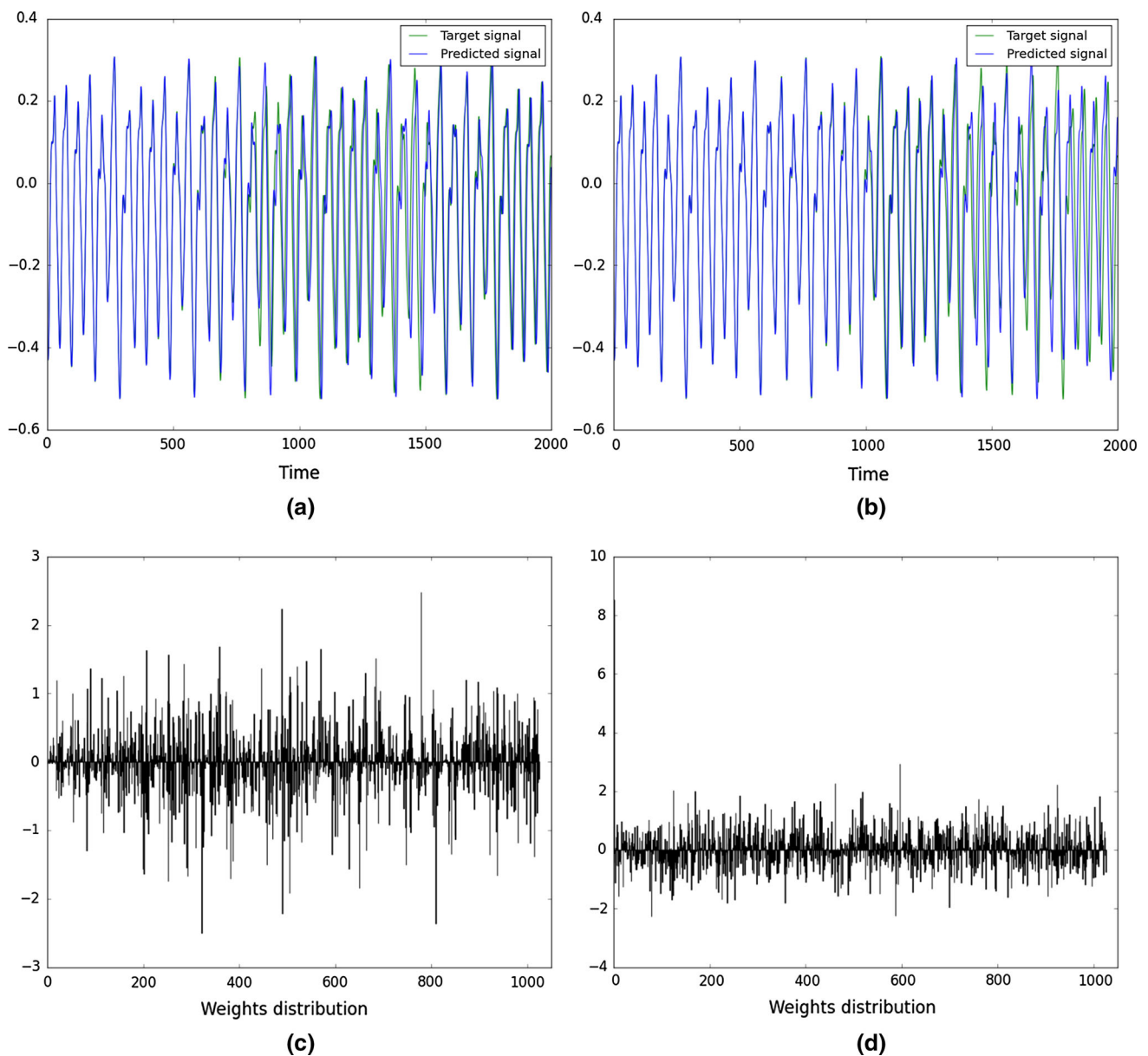


Fig. 6 Benchmark task results of the memristive ESN and the original ESN. **a, b** Demonstrate the deviation between target signal and predicted signal. **c, d** Show the distribution of the readout weights

structure. Second, this might be caused by the simplified CNN structure of the proposed reservoir. Since the proposed ESN is conceptually simple and computationally inexpensive, successfully applying ESN is sometimes empirical. Therefore, the further simplified ESN with a memristor-based CNN structure may lead to more stability problems than the original ESN and yield a slightly worse performance. Third, in this comparison, the weight matrix of the reservoir in original ESN is optimized through normalizing and setting its spectral radius. However, in the proposed ESN, the weight matrix of the reservoir is not optimized thus there are opportunities to improve its performance by a proper optimization.

By investigating the example results shown in Fig. 6(c), (d), the readout weights \mathbf{W}^{out} of the memristive reservoir is very similar to the original ESN’s range $(-2, 2)$. However, in some trials, the readout weights of the memristive reservoir is obviously greater than the original ESN’s for example in the range $(-6, 6)$. According to the practical guide [21], large output weights \mathbf{W}^{out} may imply that the solution is sensitive and unstable because a tiny difference will be amplified by the output weights and lead to large deviations from the expected values. Therefore, the average performance of proposed ESN is slightly sensitive than the original ESN. For the purpose of improving the performance, a practical approach is selecting the parameters

carefully and tuning the parameters manually or automatically through grid search which exhaustively searches for proper parameters by comparing the performance metric. Considering the very simplified CNN structure with memristors, the proposed ESN structure is promising for some specific tasks which require smaller size and less computation, and it is worth further investigation.

6 Conclusion

By deriving the HP memristor with a single stair, the staircase memristor model is investigated in this paper based on the delayed-switching effect and the experimental results of discovered memristors. The simulation results are very similar to the recently discovered ferroelectric memristors. A comparison of the staircase memristor model and HP memristor model is given to inspect the distinctions between them. The staircase memristor model is assumed to have multiple regions and each region has a delayed-switching effect. Before the threshold of a region is reached, the staircase memristor model remains at a somewhat stable state. By varying the parameters, the features of regions can be modified according to different characteristic of staircase memristors which will result in different staircase scenarios. The proposed CNN structure with memristor-based local connections is adapted to ESNs in order to reduce the connection complexity. By modifying the connections, neurons in the reservoir of a ESN are locally connected in the form of the proposed CNN structure rather than randomly or fully connected. Without optimization of the proposed memristive ESN, its performance in the benchmark test is comparable with the original ESN. However, there exists some limitation of practical memristors and staircase memristors. In fact, current practical staircase memristors, such as [3, 5, 27] can only have limited multilevel resistance. Therefore, for resistance beyond these levels, the accuracy must be concerned and evaluated in the future. Another example of limitations is the switching frequency. Because of the delayed-switching effect of staircase memristors, switching a memristor takes place with a time delay which will affect applications which require a high switching frequency. Since the memristor is still in the early stage compared to other emerging technology, these limitations might be improved and overcome in the future. Therefore, using memristors in building local connections in CNN and ESN circuits is worth discussing to enjoy the benefits of memristors such as high density and low power consumption. Overall, the results given in this paper demonstrate the potential of memristors in bio-inspired electrical and electronic circuits.

Acknowledgments The authors would like to thank Prof. Chua for his support and help on our work. We also would like to thank Prof. Jones and Dr. Corinto for giving us great help and comments.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Biolek, Z., Biolek, D., & Biolkova, V. (2009). SPICE model of memristor with nonlinear dopant drift. *Radioengineering*, 18(2), 210–214.
2. Borghetti, J., Snider, G. S., Kuekes, P. J., Yang, J. J., Stewart, D. R., & Williams, R. S. (2010). ‘memristive’ switches enable ‘stateful’ logic operations via material implication. *Nature*, 464(7290), 873–876.
3. Chang, Y. F., Chang, T. C., & Chang, C. Y. (2011). Investigation statistics of bipolar multilevel memristive mechanism and characterizations in a thin FeO_x transition layer of $TiN/SiO_2/FeO_x/Fe$ structure. *Journal of Applied Physics*, 110(5), 053703.
4. 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.
5. Chanthbouala, A., Garcia, V., Cherifi, R. O., Bouzehouane, K., Fusil, S., Moya, X., et al. (2012). A ferroelectric memristor. *Nature Materials*, 11(10), 860–864.
6. Chen, Y. J., Chang, K. C., Chang, T. C., Chen, H. L., Young, T. F., Tsai, T. M., et al. (2014). Resistance switching induced by hydrogen and oxygen in diamond-like carbon memristor. *Electron Device Letters, IEEE*, 35(1), 1016–1018.
7. Chen, W., Yang, X., & Wang, F. Z. (2013). Delayed switching applied to memristor content addressable memory cell. *Proceedings of the World Congress on Engineering*, 1, 354–357.
8. Chua, L. (1971). Memristor: the missing circuit element. *IEEE Transactions on Circuit Theory*, 18(5), 507–519.
9. Chua, L. (2011). Resistance switching memories are memristors. *Applied Physics A: Materials Science and Processing*, 102, 765–783.
10. Chua, L., & Kang, S. M. (1976). Memristive devices and systems. *Proceedings of the IEEE*, 64(2), 209–223.
11. Chua, L., & Yang, L. (1988). Cellular neural networks: theory. *IEEE Transactions on Circuits and Systems*, 35(10), 1257–1272.
12. Chua, L. O., & Yang, L. (1988). Cellular neural networks: Applications. *IEEE Transactions on Circuits and Systems*, 35(10), 1273–1290.
13. Corinto, F., & Ascoli, A. (2012). A boundary condition-based approach to the modeling of memristor nanostructures. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 59(11), 2713–2726.
14. Eshraghian, K., Cho, K. R., Kavehei, O., Kang, S. K., Abbott, D., & Kang, S. M. S. (2011). Memristor MOS content addressable memory (MCAM): Hybrid architecture for future high performance search engines. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 19(8), 1407–1417.
15. Jaeger, H. (2001). The “echo state” approach to analysing and training recurrent neural networks—with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148, 34.
16. Joglekar, Y. N., & Wolf, S. J. (2009). The elusive memristor: Properties of basic electrical circuits. *European Journal of Physics*, 30(4), 661.

17. Kulkarni, M., Teuscher, C. (2012). Memristor-based reservoir computing. In *2012 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)* (pp. 226–232).
18. Kvatinsky, S., Friedman, E., Kolodny, A., & Weiser, U. (2013). TEAM: Threshold adaptive memristor model. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1), 211–221.
19. Lehtonen, E., Laiho, M. (2010). Cnn using memristors for neighborhood connections. In *2010 12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA)* (pp. 1–4).
20. Linares-Barranco, B., Serrano-Gotarredona, T., Camunas-Mesa, L. A., Perez-Carrasco, J. A., Zamarreno-Ramos, C., & Masquelier, T. (2011). On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. *Frontiers in Neuroscience*, 5, 1–22.
21. Lukoševičius, M. (2012). A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade* (pp. 659–686). Springer.
22. Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11), 2531–2560.
23. Mackey, M., & Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science*, 197(4300), 287–289.
24. Strukov, D. B., Snider, G. S., Stewart, D. R., & Williams, R. S. (2008). The missing memristor found. *Nature*, 453(7191), 80–83.
25. Wang, F., Helian, N., Wu, S., Lim, M., Guo, Y., & Parker, M. (2010). Delayed switching in memristors and memristive systems. *IEEE Electron Device Letters*, 31(7), 755–757.
26. Wang, F. Z., Helian, N., Wu, S., Yang, X., Guo, Y., Lim, G., et al. (2012). Delayed switching applied to memristor neural networks. *Journal of Applied Physics*, 111(7), 07E317–07E317.
27. Wang, S. Y., Huang, C. W., Lee, D. Y., Tseng, T. Y., & Chang, T. C. (2010). Multilevel resistive switching in $Ti/Cu_xO/Pt$ memory devices. *Journal of Applied Physics*, 108(11), 114110.
28. Xu-Dong, F., Yu-Hua, T., & Jun-Jie, W. (2012). SPICE modeling of memristors with multilevel resistance states. *Chinese Physics B*, 21(9), 98901.
29. Yang, X., Chen, W., Wang, F. Z. (2013). A supervised spiking time dependant plasticity network based on memristors. In: *2013 IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)* (pp. 447–451).
30. Yang, X., Chen, W., Wang, F. (2014). The staircase memristor and its applications. In *2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS)* (pp. 259–262)



Xiao Yang received the B.Eng. and M.Sc. degrees in electronic engineering in 2008 and 2009 from University of Hull, U.K. He is currently working towards the Ph.D. degree in computer science in Future Computing Group, University of Kent, U.K. From 2010 to 2014 he enjoyed a great work experience in School of Computing, University of Kent and supervised undergraduate level courses such as Object Oriented Programming and Project Planning. His

research activity is mainly in the area of neural networks, memristors, memristive devices and circuits.



research activity is mainly in the area of neural networks, memristors, memristive devices and circuits.

Wanlong Chen received the B.S. degree (first class honours) in computer science from the University of Hertfordshire, Hatfield, U.K., in 2011. Since 2011, he has been working as a graduate student researcher towards a Ph.D. degree in computer science in the University of Kent, Canterbury, Kent, U.K. He is currently with the future-computing group in the school of computing in the University of Kent, working on green computing and future computing via memristors. His research interests include modelling and simulation of memristive devices and circuits; memristors, memristive devices and systems; emerging non-volatile memories.



Professor in Future Computing and Head of School of Computing, University of Kent, U.K. The School of Computing was formally opened by Her Majesty the Queen. His research interests include memristor as a new computing paradigm, unconventional computing, green computing, grid/cloud computing, and data storage and data communication, etc. He has been invited to deliver keynote speeches and invited talks to report his research worldwide, for example at Princeton University, Carnegie Mellon University, CERN, Hong Kong University of Science and Technology, Tsinghua University (Taiwan), Jawaharlal Nehru University, Aristotle University, and University of Johannesburg. In 1996, he designed and developed a new type of random access memory using the spin-tunnelling effect at Tohoku University, Japan. This device was the first of its kind worldwide. Prof Wang and his team have won an ACM/IEEE Super Computing finalist award. He was elected as the Chairman (U.K. and Republic of Ireland Chapter) of the IEEE Computer Society in 2005. He is a Fellow of the British Computer Society. He has served the Irish Government High End Computing Panel for Science Foundation Ireland (SFI) and the U.K. Government EPSRC e-Science Panel.