Accepted Manuscript

Journal of Circuits, Systems, and Computers

Article Title:	Programming of Memristive Artificial Synaptic Crossbar Network using PWM Techniques
Author(s):	P Michael Preetam Raj, Arvind Subramaniam, Shashank Priya, Souri Banerjee, Souvik Kundu
DOI:	10.1142/S0218126619502013
Received:	17 April 2018
Accepted:	08 November 2018
To be cited as:	P Michael Preetam Raj *et al.*, Programming of Memristive Artificial Synaptic Crossbar Network using PWM Techniques, *Journal of Circuits, Systems, and Computers*, doi: 10.1142/S0218126619502013
Link to final version:	https://doi.org/10.1142/S0218126619502013

World Scientific
www.worldscientific.com

Journal of Circuits, Systems, and Computers

Programming of Memristive Artificial Synaptic Crossbar Network using PWM Techniques

P Michael Preetam Raj[†], Arvind Subramaniam[†], Shashank Priya[‡], Souri Banerjee[§], Souvik Kundu[†,*]

[†]*Department of Electrical & Electronics Engineering, BITS-Pilani Hyderabad Campus, Hyderabad, 500078, India*

[‡]*Center for Energy Harvesting Materials and Systems, Department of Mechanical Engineering, Virginia Tech, Blacksburg - 24061, USA*

[§]*Department of Physics, BITS-Pilani Hyderabad Campus, Hyderabad - 500078, India*

[*]*souvikelt@gmail.com; souvik.kundu@hyderabad.bits-pilani.ac.in*

Memristor resembles an artificial synapse and is considered to be basic electronic element for realizing neuromorphic circuits. In this work, a systematic investigation was conducted on memristor-based resistance-programming circuits to write analog data into a memristor utilizing pulse width modulation techniques. The high-frequency sinusoidal signal was utilized to read the data in form of its electronic resistance. An optimum circuit configuration demonstrated multilevel stable resistive states, which are analogous to the connection weights in the human synapse. In order to modulate these memristive weights for representing the learning activities in human brain synapse, it was identified that the pulse width modulation technique is superior as compared to spike-timing-dependent plasticity. Further, the above analysis was utilized in training the memristor to update its resistive weights in consonance with its learning, analogous to that in a neural network. Further, the memristive crossbar architecture was utilized to implement a real-time application in Econometrics, where an array of memristors were utilized to learn and update the purchase trends of an N × L matrix of customers. The proposed circuits possess the advantages of high packing density, low power consumption and non-volatility, and also pave the way for developing future neuromorphic circuits.

Keywords: Memristor; Pulse Width Modulation; SPDT Switch; Memristive Neural Network; Crossbar Network.

1. Introduction

The power consumption and thermal management of transistors are becoming challenging for both miniaturization and implementation of nanoelectronic devices and circuits. [1] Memristor offers higher circuit density (> 40% improvement) and much lower power consumption (one-hundredth times), which provides an intriguing alternative for complementary metal oxide semiconductor (CMOS) transistor-based devices. [2] Owing to this potential, memristors have attracted wide interest for the design of neuromorphic systems and various other electronics. [3] Memristors have been utilized in chaos and biomimetic and neuromorphic circuits, specifically for emulation of synaptic behavior. [4] There are several different synaptic properties such as long-term potentiation (LTP),

long-term depression (LTD), retention, spike-timing-dependent plasticity (STDP), and integrate-and-fire [5], which are emulated by artificial devices and are crucial in a neural network. [6] The spike signals in STDP have the drawback related to the dependence of memristance on the neighboring circuit elements. [7] It is important to mention that the resistance change in the memristor is triggered by the average value of the current flowing through it. One can notice that the average value of a practical spike signal cannot be determined precisely in comparison with that of the pulse signal. [8] In addition, the important function of the reversal of LTP or LTD, to move the system to its default settings, is not available with the use of spike signals in STDP. [9,10] In order to overcome the abovementioned problems associated with the STDP, in this work, pulse width modulation (PWM) techniques were utilized for programming the memristor.

As mentioned earlier, memristors are capable of performing similar logic functions as multiple transistors, thereby, making them a promising component for improving computational capabilities.[11] One needs a square of the number of transistors (n^2) to that of the number of memristors (n), resulting in a hundred times lower power consumption and three times lower space, for a chip size of ~100 transistors.[11] Therefore, efforts were made to develop neuromorphic circuits based on memristors that can reduce the chip area and improve power efficiency. [12–14] Pershin et al. demonstrated human brain like memory in the neural network where two-memristor-emulating synapses were connected to mimic three electronic neurons. [15] However, the design has disadvantage in terms of the complex circuit design arising due to the requirement of analog-to-digital converter and a microcontroller. Although several studies have been conducted on memristor based neuromorphic systems, it is important to consider the practicality in terms of crossbar network based large scale applications.

One of the important operations in the human brain is updating synaptic weights, due to learning. [16] This function of updating the weights can be achieved by logic computing memristive systems. [4] In literature, researchers have demonstrated several memristive computational logic systems by analyzing the features of the human brain. [17–20] A memristor-based bridge synapse circuit capable of learning and implementing fuzzy logic was recently demonstrated experimentally. [20] However, this design has the drawback of area overhead as the memristive bridge requires multiple memristors. [20] Implementation of artificial biological systems has been achieved by utilizing CMOS neurons and memristive synapses, which perform STDP function. [21,22] Despite these prior advancements, the basic foundational theory for using a memristor in electronic systems that mimic the human brain is still underdeveloped and the two-level based design of the memristive circuits remains computationally inefficient. [23]

In this work, the overarching goal is to investigate the memristor-based neural network concentrated on bio-inspired systems. PWM was used instead of spike signals to program memristance, resulting in multilevel programmable resistive states. These states are analogous to the connection weights in the human brain synapse. Since the PWM signals generate a predetermined average current to precisely program the memristor, it is

noteworthy to mention that the proposed PWM based memristive networks have the advantage of making the memristor immune to variations in the circuit elements connected to it. Finally, a memristive circuit was developed by utilizing the concept of controlled resistance programming rates and network feedback techniques, which can mimic brain cells. Efforts were devoted to train the memristive crossbar network to learn and perform the computations in accordance with real-time marketing strategies.

2. Simulations

LTspice XVII simulator was used in simulating the electrical characteristics of the memristive circuits. The physical memristive structure comprises of a two-layered (with and without oxygen vacancies) 10 nm thin TiO_2 film, sandwiched between two platinum contacts. The same memristor design was manufactured in 2008 by Hewlett-Packard Labs. This memristor device model is available in LTspice. [24] Memristor components were included in the LTspice directory and it was utilized in constructing as well as simulating novel logic circuits.

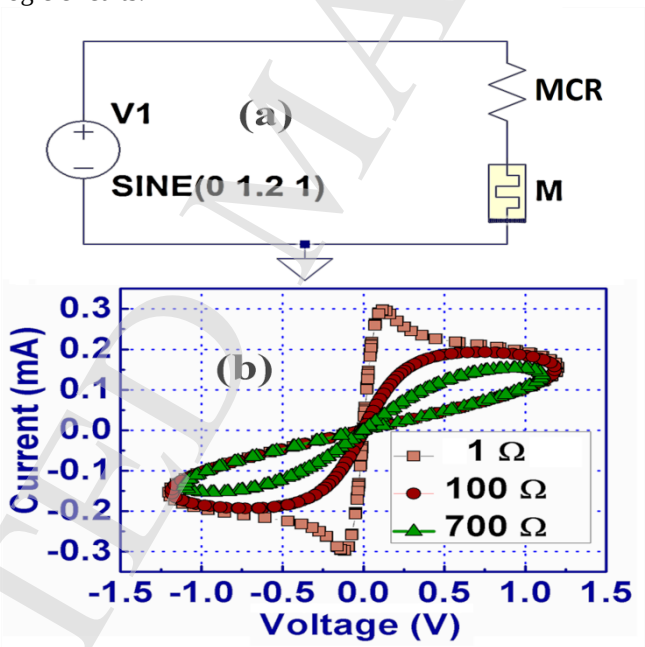

Fig. 1. (a) Schematic of the simulated MCR based memristor circuit, and (b) its I-V characteristics. The variation in the resistive value of the MCR modulates the shape of the pinched hysteresis loop. An input sinusoidal signal with zero DC offset was utilized to identify these characteristics.

3. Results & Discussions

Unlike other components such as resistors, inductors, capacitors, diodes, and transistors; memristors do not have a constant shape of current – voltage (I-V) characteristics, as their resistive state changes dynamically. [25] Further, varying the

resistive parameters of the elements connected to it in a circuit influences its I-V characteristics, as shown in Fig. 1(a). Memristance is the resistance obtained from memristor's I-V characteristics; therefore it is the internal component of memristor. Whereas memristance controlling resistance (MCR) is the series resistance (external component) connected in series with the memristor (as shown in Fig. 1(a)). One can vary the current by tuning the voltage V1and the current passed through the MCR first and then transited into the memristor. The addition of a series MCR with variation of its value results in different I-V characteristics (shown in Fig. 1(b)). From the Fig. 1(b), it was observed that the pinched hysteresis loop and the low resistance state of the memristor are controlled by using the MCR. The pinched hysteresis is due to the presence of internal negative resistance in memristor. [26] Negative resistance refers to the case where the increase in voltage causes decrease in current or vice versa. In general, the negative resistance components are represented by the negative slopes in I-V characteristics. This negative resistance can be altered so that the shape of I-V characteristics can be modulated by adding an MCR, as shown in Fig. 1(b).

It is noteworthy to mention that increasing the value of MCR results in a narrow pinched hysteresis loop, whereas reducing its value results in wider loops. This is owing to the fact that increases in the value of MCR results in reduced current through the memristor. The reduction in the current causes a slower drift of the positively charged oxygen vacancies, which ultimately results in narrow pinched hysteresis loop. [11] These results reveal the ability to precisely change and control the shape of I-V characteristics by simply varying the value of MCR. Moreover, these results are suitable for a variety of applications where the modulation of the shape of the I-V curve is important. Controlled variation of memristance is also useful for electronic applications such as active filters for signal processing. [27]

In order to control the memristance or fix its value, Williams demonstrated that it could be done by varying the applied signal frequency. [11] However, such systems are not internally controllable since the applied frequency is an external signal, making this technique unsuitable for automation. Here, we demonstrate that the memristance can be precisely controlled by varying MCR. Since MCR is an internal component of the system, this technique is suitable for automated control. It is noteworthy to mention that a single memristor can replace 2 op-amps, 14 transistors, 10 resistors and 4 capacitors [28], hence use of a single memristor in a circuit reduces the number of components, their power consumption, and ultimately the size of circuitry. These features were obtained when memristor was utilized in memory and neuromorphic computational circuits. [12–14]

From Fig. 2, it can be seen that the memristance was periodically varying with the applied voltage. In this work, the initial resistance of the memristor is found to be 11 kΩ, which is same as reported by Biolek et al. [24] The memristance deviates from this value based on the polarity at which it is connected in the circuit. The deviation in memristance is due to the drift of oxygen vacancies within the device. [25] Under forward bias, when the MCR value was maintained at 1 Ω, the memristance varied from 11 kΩ

to 200 Ω (Fig. 2(a)). When the value for MCR was varied to 100 Ω and then to 1kΩ, the memristance changed from 11 kΩ to 1.25 kΩ and then 11 kΩ to 3.80 kΩ, as shown in Fig. 2(b) and Fig. 2(c), respectively. Thus, from the above results, it can be observed that the variation of MCR changes the resistive swing of the memristor. However, for all the cases, the upper limit of the memristance is fixed at 11 kΩ, which is the initial resistive state of the memristor. [24] When the input sinusoidal voltage rises with time, there is a reduction in the memristance owing to the drift of positively charged oxygen vacancies (in TiO_2 layer) towards the ground terminal. [25] The reason for the increase in the lower limit of the memristance is due to the impact of MCR value on the internal negative resistance of the memristor.

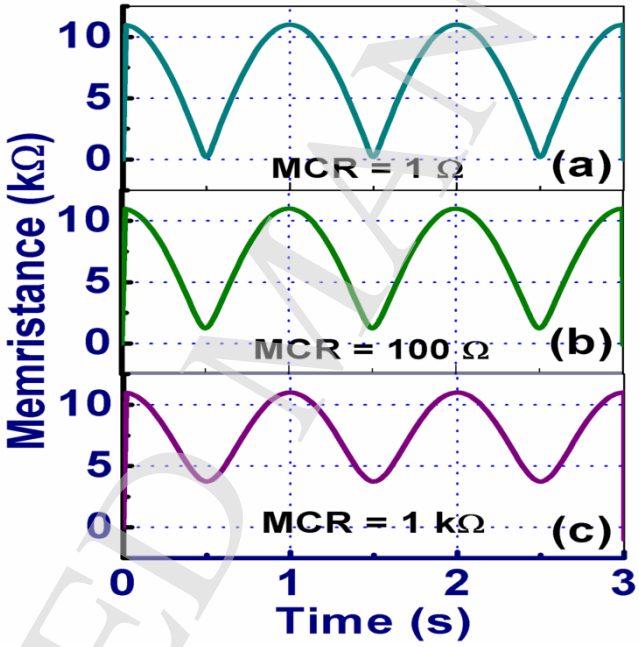

Fig. 2. Variation of memristance as a function of time, for an MCR value of (a) 1 Ω, (b) 100 Ω and (c) 1 kΩ. The variation of MCR changes the resistive swing of the memristor.

The MCR value adds on to the existing negative resistance which ultimately reduces the net negative resistance, thus bringing down the memristive swing, which can be found in Fig. 2 (a,b, and c). Unlike the situation in Fig. 2, the memristance can be increased beyond 11 kΩ by exchanging the two terminal connections of the memristor as depicted in Fig. 3.

In order to develop a memristive neural network, it is essential to consider the functionalities of a real human brain cell. An external signal received by the human sensors results in the firing of a neuron in the brain. This activated neuron, in turn, fires the neighboring neurons, once the threshold value is exceeded, and the process

continues. [29] Based on the results at the output layered neurons, human beings are able to perform a series of parallel activities. The firing of a neuron is analogous to turning ON the memristor. [30] The learning in the brain takes place due to the update of weights in the synapse, which is analogous to the update of the resistance in a memristor. [4] Thus, it can be understood that a brain cell can be mimicked by memristive circuits. [31,32] A circuit was developed where memristor was programmed using pulse ON time variations (also known as PWM) as shown in Fig. 3(a, and b). Once learning has occurred, the memristor is switched to make the connection with the sinusoidal voltage source (to measure memristance) by using single pole double throw (SPDT) switch. A 1.20 V direct current (DC) voltage is utilized to write the data (update memristance) into the memristors. The read operation was performed by measuring the memristance by utilizing the relation given below

$$M = \frac{v_{mem}}{i_{mem}} \tag{1}$$

where M is the memristance, v_{mem} is the ac voltage across the memristor and i_{mem} is the ac current through the memristor. In this case, a 20 mV (frequency 1 kHz) ac signal is utilized to perform the read (measuring memristance) operation. In order to minimize the effect of noise on the memristive circuits, it is essential to eliminate the average and low-frequency noise components by utilizing high pass filtered signals for read operations. This technique increases the number of stable resistive states within the memristor.

The choice of the low voltage for the read cycle is mainly due to controlled movement of the oxygen vacancies within the TiO_2 layers. [11] If one uses a very large voltage to perform the read operation, the oxygen vacancies get disrupted and the data within the memristor changes. On the other side, the memristance varies with the average value of the current flowing through it. Thus, in order to avoid a destructive read operation, it is essential that the signal should be based on low magnitude pure alternating current. Therefore, a signal with amplitude 20 mV and high frequency pure sine wave with zero DC offset was found to be suitable for the read operations. The DC

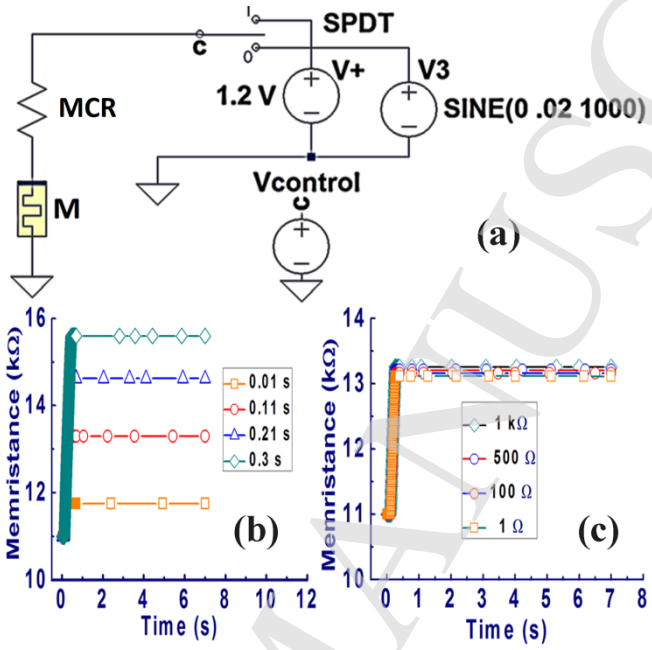

Fig. 3. (a) Schematic of pulse on-time variation based memristor circuit, and its (b) resistance-time characteristics. It can be seen that the data can be written in the form of memristance. Different values of pulse ON duration (T_{ON}) correspond to different memristances. (c) Resistance-time characteristics due to variation of series resistance for a given T_{ON}. It can be seen that the PWM technique is immune to variations in MCR. A 20 mV sinusoidal signal with a frequency greater than 1 KHz can be utilized to perform the read operation.

input facilitates the potentiation and depression effects of the memristor. This is due to the fact that the memristor tends to deviate to either ON state or OFF state in the presence of a DC voltage, depending upon its polarity. However, this deviation represents the potentiation and depression effects in the human brain. Since the memristance changes with time, in accordance with the applied voltage (Fig. 2), it was found that the duration of the applied signal determines the value of memristance. This was the basis for proposing the technique for resistance programming in a memristor, controlled by the pulse ON duration, which is an average-valued signal. The ON time of 1 V pulse signal (indicated by Vcontrol in Fig. 3(a)) was varied from 0.01 s to 0.30 s with a step size of 0.10 s to program the memristor and the memristance was retained beyond 7 s, as shown in Fig. 3(b) where MCR was fixed at 100 Ω. For 0.01 s pulse ON time, the memristance was programmed to be ~11.70 kΩ, which can be changed to ~15.50 kΩ for 0.30 s pulse ON time. No variation in the memristance over measured time (straight lines parallel to time axis) indicates long-term data retention. Thus, Fig. 3(b) demonstrates the role of PWM in precisely programming the memristor. The variation of memristance with the pulse ON time can be expressed by Eq. (2).

$$M = \left(12.9878 \times T_{ON} + 11.5690\right)\ \text{k}\Omega \qquad (2)$$

where T_{ON} is the pulse ON duration. Although one can control the memristor by utilizing MCR, in certain cases the parasitic series resistances alter the net value of the MCR resulting in unwanted modulation of memristance. [7] Hence it is desirable to make the circuit immune to the variations of the MCR and the programming is only due to the PWM. Importantly, it is shown in Fig. 3(c) that the pulse ON time variation technique is independent of the MCR value. In this case, the ON duration of the pulse signal in Fig. 3(a) is fixed to a value of 0.10 s and MCR was varied from ~0 Ω to 1 kΩ (Fig. 3(c)). If the MCR is varied from ~0 Ω to 1 kΩ, using the pulse on-time variation technique, one can notice that all the memristances get overlapped. This shows the effectiveness of this technique as there is no variation in the memristance even though other elements are varied (MCR represents the resistance provided by other circuit elements / parasitic resistance). This indicates that the PWM techniques are independent of the MCR value. This is because, unlike for a sine wave, the voltage and current values in a pulse signal change abruptly, making the device follow its natural electronic characteristics. This results in a huge benefit in terms of the stability through the memristive programming utilizing the pulse signal, as the memristance is no longer affected by the variations of the neighboring circuit elements (as shown in Fig. 3(c)). Resistance programming using a different PWM technique as shown in Fig. 4(a) achieves an outcome similar to that in Fig. 3(a). The results will be contrary if one applies an STDP signal instead of PWM. For an STDP, there will be only two states (ON or OFF) of the memristance values. However, obtaining the programmed multilevel stable states is not possible. This is because triggering a memristor by utilizing STDP signal depends only upon the polarity of the spike signal. Therefore, only two memristive states are possible due to the positive and negative spike triggering. [7] Unlike that of a spike signal, one can change the duty cycle of a pulse signal, which is used to precisely program the memristor for multilevel stable memory states. The number of these levels (nl) can be expressed by Eq. (3), which relates nl to the *off* resistance (R_{off}), *on* resistance (R_{on}), and maximum resistive drift (R_{dmax}).

$$nl = \left(\frac{R_{off} - R_{on}}{R_{d\max}} \right) \tag{3}$$

Here, R_{dmax} depends upon the average value of the noise signal (V_{ndc}) applied to the memristor over a time (t) as expressed in Eq. (4).

$$R_{\mathrm{dmax}} \propto \left(V_{ndc} \times t \right) \tag{4}$$

Automation and updating of the memristance value are possible when the comparator (U1) based circuit shown in Fig. 4(a), is used instead of the Vcontrol source in Fig. 3(a). Utilizing this technique, memristance was programmed using the PWM technique, which is shown in Fig. 4(b), corresponding to the different values of the voltage source V2 (V1

is a triangle signal as shown in Fig. 4(a)). When V2 is kept less than V1, the output goes high, whereas the output comes to a low value when V2 exceeds V1. Thus, modulating the value of V2, the net time range of V1 in comparison to V2 gets changed, which results in a PWM signal. The dependence of the voltage source V2 on the memristance is analysed and can be determined using Eq. (5).

$$M = \left(-1.5938 \times V2 + 16.0695\right) \text{k}\Omega \tag{5}$$

The circuit shown in Fig. 4(a) is useful in the construction of the memristive neural network, which is a complete automated system, as depicted in Fig. 5(a). The inputs and outputs were represented in the form of the voltage signals, and the memristance changes with the ON duration of the pulse signal (Fig. 3(b)), Hence, it was essential to convert the voltage signal into the corresponding pulse signals with changeable ON duration by utilizing the PWM. The major function of a neural network is update of the synaptic weight based on the error signal (the difference between the known and obtained outputs). [16] The schematic shown in Fig. 5(a) represents this functionality of the memristive neural network, which performs the above-mentioned computations. The

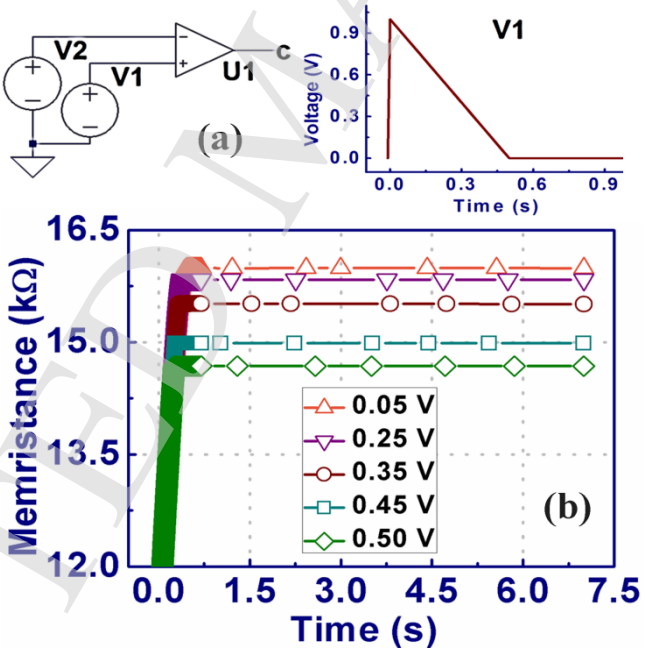

Fig. 4. (a) Effect of using a pulse width modulator, and (b) its resistance-time characteristics. Memristance varies with the change in the dc voltage V2, as its comparison with V1 results in a PWM signal.

input, in form of a voltage signal, is fed to the memristor circuit. The difference between the known output and the output of the memristor circuit (error signal) is multiplied by

the learning rate and the input. This resulted value is added to the voltage signal corresponding to the previous weight and the overall result feeds the input to the PWM.

The obtained signal, at the output of the PWM block, is utilized to update the memristance. It is noteworthy to mention that a known output is basically the user-defined output during training of the neural network. The proposed neural network (analogous to the natural neural networks) consists of two paths, viz. back propagation and forward path. [33] The role of the back propagation is to update the memristance based on previous input and output experiences to facilitate learning. On the other side, the forward path is utilized to perform decisions based on the learning experience. It is important to mention that the forward path consists of the input, memristor circuit and output as shown in Fig. 5(b). The remaining portion is employed for the update of the memristance through back propagation.

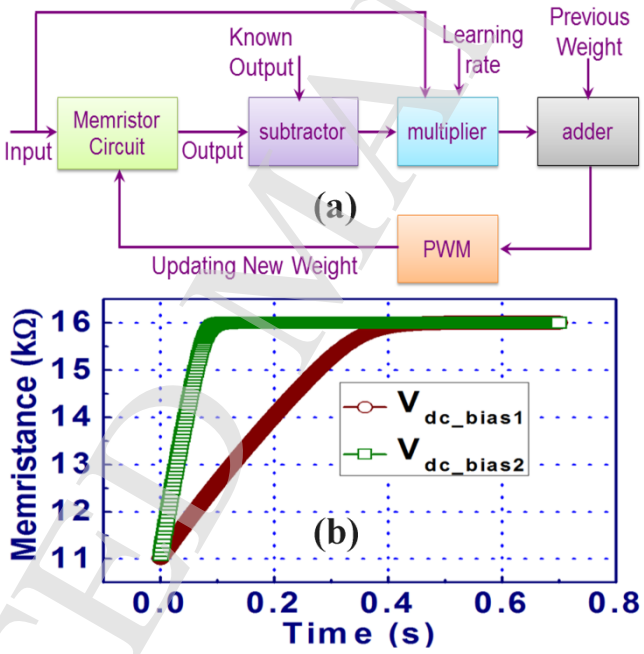

Fig. 5. (a) Schematic of the functionality of proposed memristive neural network, and (b) control of the rate of resistive programming. The learning rate can be modulated by changing the external dc bias voltage which is applied to the memristor.

Human beings set priority to different tasks and perform them at different speeds. The ability to automatically modulate the learning rates based on previous experiences is an important reason for successful functionality. [34-36] Thus, the rate of resistive switching analogous to the learning rate in human brains, is an important parameter in a memristive neural network. An automated modulation of resistive switching rate is thereby suitable technique for human-like behavior required for the next generation of humanoid robots. The voltage controlled resistive switching rate modulation which corresponds to the

variation of the memristive DC bias voltage is shown in Fig. 5(b). As shown in this figure, the learning rate is found to be slow when Vdc_bias1 is applied and becomes fast for Vdc_bias2, where Vdc_bias2 is greater than Vdc_bias1. One can also develop an automated feedback mechanism, which modulates the value of the bias voltage. It leads to trigger the learning rates, making the neural network perform various tasks with different mechanisms, according to its learning and experience.

Usually, in a neural network system, the learning rate should be low enough that the weights converge to a precise value, but high enough to spend less time training it. [37] Interestingly, utilizing our memristive neural network, one could tune the learning rates as per required applications. If one uses the conventional CMOS or transistor-based circuit, this important feature cannot be afforded as the number of circuit elements will get increased over twenty times as compared to memristors. [38] This occurs due to the fact that data storage and state transitions are necessary for the neural net computation, which requires transistor based flip-flops and a combinational circuit. [39] However, the tasks performed by these flip flops can be replaced by a single memristor. Thus, the memristor-based neural network reduces the overall complexity of the system. [40]

A broad practical application of the proposed technique could be in predictive analytics. For example, a sales manager wants to modulate sales strategy in accordance with the change in purchase frequency of a customer. In such case, our proposed memristive neural network can be assigned to the customer. In accordance with the prior purchases, the neural network gets trained to estimate the future purchases, which can be utilized to make changes in the sales strategies. The function y = Ax is considered for this application, where A, x, and y represent the memristive synaptic weight, input and output, respectively. In order to teach this relation to a memristor, one ought to consider a value for A. It is assumed that a customer usually procures at least seven products in every week (say 1 product per day). Hence, the value A = 7 (7 products per week) is most common for this application. On the other side, let the initial random weight be 6, while the learning rate is 0.20 (say). The input and output training vectors, for this function are [1 2 3 4 ……] and [7 14 21 28 ……..], respectively, which correspond to the relation y = 7x. It is noteworthy to mention that the calibration of the outputs in terms of the memristance range is achieved by considering a constant multiplication factor of 2 kΩ. This factor is useful in maintaining the memristor to operate within a predefined resistive swing.

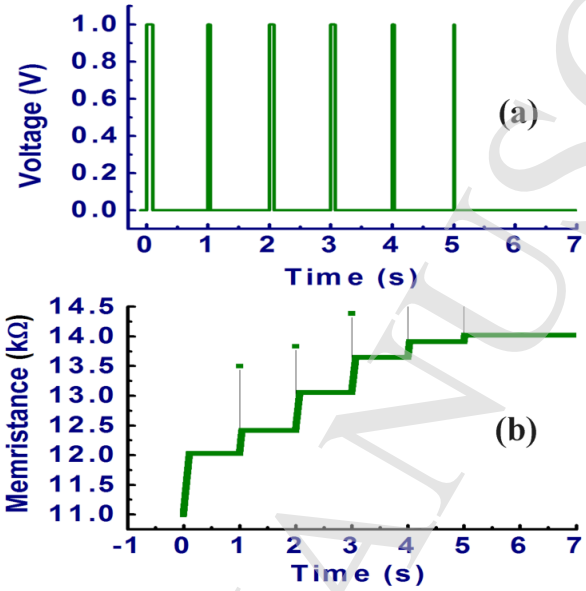

Fig. 6. Variation of (a) the PWM signal and (b) the memristance as a function of time. The PWM signal was utilized to systematically update the memristance.

In this work, we demonstrate the systematic update of memristance to 14 kΩ analogous to the update of synaptic weights (due to learning) in human brain. In this manner, the memristor learns the purchase frequency of a customer. For the abovementioned input vector, the output of PWM block is shown in Fig. 6(a). The memristance updation which corresponds to the input signal in Fig. 6(a)) is tabulated in Table 1. These computations are in accordance with the block diagram shown in Fig. 5(a). Table 1 presents the numerical values which represent the systematic update of memristance in accordance with its learning. In this case, it can be seen that the memristor learns the relation y = 7x, at the end of 5 iterations. It can be seen that the memristance gets saturated at 14 kΩ. No further increment in the memristance represents the end of numerical iterations. The presence of noise could introduce an error in the memristance value. However, if one maintains the minimum allowed memristive drift greater than R_{dmax}, the false triggering caused by the noise signal can be avoided. The PWM signal obtained in Fig. 6(a) was applied to the circuit shown in Fig. 3(a) (in place of the Vcontrol shown in Fig. 3(a)). The output results across memristor in the systematic update of memristive weight (to 14 kΩ) are shown in Fig. 6(b).

Table 1. Stepwise update of memristive weights.

S. No.	Input	Previous weight	Output	Memristance (kΩ)
1	1	6	6.2	12.4
2	2	6.2	13.04	13.04
3	3	6.52	20.424	13.616
4	4	6.808	27.8464	13.9232
5	5	6.9616	35	14
6	6	7	42	14
7	7	7	49	14

As per the results in Table 1, the deviation in the purchase rate triggers the corresponding change in the memristance weights. Thus, the neural network learns and keeps track of the changes in the purchase trend of customers. In a real-time situation, a sales manager is expected to analyze a large group of customers simultaneously. This can be implemented by utilizing the crossbar array of memristors, where an individual memristor in the crossbar was assigned to track the purchasing trend of one customer.

The memristive crossbar architecture for real-time applications, which facilitates systematic weight update is shown in Fig. 7. The PWM technique shown in Fig. 3(a) was applied to the crossbar structure. In the crossbar architecture, an individual memristor was selected by accessing its row first and then its column. The selected row can be connected to the programming voltage sources shown in Fig. 7. The selected column is grounded, while the other rows and columns are open circuited. Thus, only the selected memristor has a closed circuit connection, while the other unselected memristors in the crossbar are open circuited. The data in the selected memristors can be written or read in order to facilitate the artificial synaptic training. However, at the same time, the data in the unselected memristors is unaltered since the electronic circuit functions take place only in the closed circuit loop paths. [41] Therefore, for half accessed devices, only the selected memristors will perform, whereas the data in the unselected devices will remain unaltered. When the selected memristor is trained, the signal is disconnected from that memristor and the access is provided to the neighboring memristor by utilizing the row and column selector switches. In this manner, a total of N × L customers can be analyzed simultaneously, in form of an N × L matrix, where each of the memristor was utilized to learn the purchase trend of an individual customer. When the purchase frequency of a customer gets changed, the memristor corresponding to the customer in the crossbar structure gets connected to the programming voltage sources through row and column

selector switches (Fig. 7), eventually update the memristance. It is important to mention that time division multiplexing was utilized for providing access to the individual memristor and effectively programming the crossbar array. Further, the coalesced effect of all the memristive synaptic updates can be employed to evaluate the characteristics of the entire N × L cluster, which is useful to make decisions corresponding to the cluster as a whole. The size of the crossbar array should be such that the number of memristors is greater than or equal to the total number of customers. For example, a 32×32 size crossbar network could be utilized to analyse a maximum of 1024 customers. Since the key component of this crossbar network is memristor, as expected, the network resulted in low power consumption, high packing density, and non-volatility. [2] Hence, it would be easier to embed the network even inside an integrated circuit.

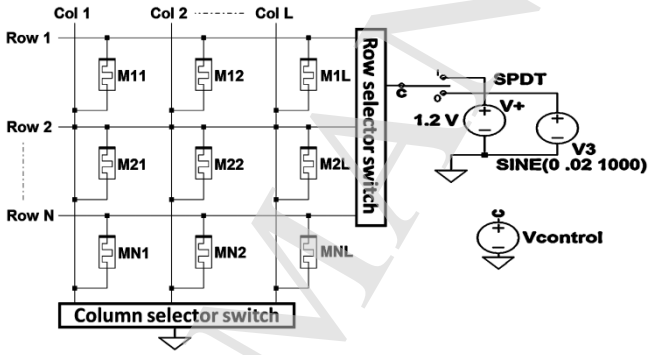

Fig. 7. The memristive architecture for predictive analytics, The proposed structure can be utilized to track N × L customers and learn their purchase trends in order to improvise sales management.

4. Conclusions

In this paper, we demonstrate a controlled resistive state transition. We have developed an electronic circuit utilizing a PWM signal, which can provide stable memristance values. Since memristance depends on the Pulse ON-time duration of the applied voltage, we have identified that PWM technique is best suited for memristance programming. Ultimately, we have developed a memristive circuit that mimics the features of a human brain. One of the important findings is the ability of this circuit to modulate its learning rate to set priorities between different tasks. Moreover, a practical implementation of memristive neural network to perform linear mathematical operations in a crossbar network was implemented. Further research is needed to investigate the memristive artificial synaptic crossbar network to test its overall functionality to resemble a human brain. It is noteworthy to mention that one could replace TiO_2 based memristors with other ferroelectric or metal-oxides (*say* HfO_2) memristors and develop the models associated with those materials. In this way, the device performance parameters such as on/off ratio, retention time, endurance, set/reset voltages, etc. can be further improved.

The efforts can also be envisaged to increase the range of memristance (presently it is 100 Ω to 16 kΩ) or memristive states by utilizing other oxides memristors. The effect of different noises on the circuit performance can be studied in order to understand the efficacy of the circuit for high frequency applications. Results presented here provide a pathway for future memristor based artificial brains for humanoid robots that are less complex, non-volatile and consume lower power.

References

1. P. S. Peercy, The drive to miniaturization, *Nature*. 406 (2000) 1023–1026. doi:10.1038/35023223.
2. D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi, C.S. Hwang, Memristors for Energy-Efficient New Computing Paradigms, *Adv. Electron. Mater*. 2 (2016) 1600090. doi:10.1002/aelm.201600090.
3. W. Zhao, D. Querlioz, J. O. Klein, D. Chabi, C. Chappert, Nanodevice-based novel computing paradigms and the neuromorphic approach, in: 2012 *IEEE Int. Symp. Circuits Syst.*, IEEE, 2012: pp. 2509–2512. doi:10.1109/ISCAS.2012.6271812.
4. S. J. Choi, G. B. Kim, K. Lee, K. H. Kim, W. Y. Yang, S. Cho, H. J. Bae, D. S. Seo, S. I. Kim, K. J. Lee, Synaptic behaviors of a single metal–oxide–metal resistive device, *Appl. Phys. A*. 102 (2011) 1019–1025. doi:10.1007/s00339-011-6282-7.
5. M. Prezioso, I. Kataeva, F. Merrikh-Bayat, B. Hoskins, G. Adam, T. Sota, K. Likharev, D. Strukov, Modeling and implementation of firing-rate neuromorphic-network classifiers with bilayer Pt/Al2O3/TiO2−x/Pt Memristors, *in: 2015 IEEE Int. Electron Devices Meet., IEEE*, 2015: p. 17.4.1-17.4.4. doi:10.1109/IEDM.2015.7409719.
6. A. Thomas, Memristor-based neural networks, *J. Phys. D. Appl. Phys.* 46 (2013) 093001. doi:10.1088/0022-3727/46/9/093001.
7. C. Zamarreño-Ramos, L. A. Camuñas-Mesa, J. A. Pérez-Carrasco, T. Masquelier, T. Serrano-Gotarredona, B. Linares-Barranco, On Spike-Timing-Dependent-Plasticity, Memristive Devices, and Building a Self-Learning Visual Cortex, *Front. Neurosci.* 5 (2011) 26. doi:10.3389/fnins.2011.00026.
8. F. Wood, M. J. Black, A nonparametric Bayesian alternative to spike sorting, *J. Neurosci. Methods*. 173 (2008) 1–12. doi:10.1016/J.JNEUMETH.2008.04.030.
9. M. Coesmans, J. T. Weber, C. I. De Zeeuw, C. Hansel, Bidirectional Parallel Fiber Plasticity in the Cerebellum under Climbing Fiber Control, *Neuron.* 44 (2004) 691–700. doi:10.1016/J.NEURON.2004.10.031.
10. W. Zhang, D. J. Linden, The other side of the engram: experience-driven changes in neuronal intrinsic excitability, *Nat. Rev. Neurosci.* 4 (2003) 885–900. doi:10.1038/nrn1248.
11. R. S. Williams, How We Found the Missing Memristor, *in: Memristors Memristive Syst., Springer New York, New York, NY*, 2014: pp. 3–16. doi:10.1007/978-1-4614-9068-5_1.
12. G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, T. Prodromakis, Integration of nanoscale memristor synapses in neuromorphic computing architectures, *Nanotechnology*. 24 (2013) 384010. doi:10.1088/0957-4484/24/38/384010.
13. M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, K. N. Salama, Memristor-based memory: The sneak paths problem and solutions, *Microelectronics J.* 44 (2013) 176–183. doi:10.1016/J.MEJO.2012.10.001.

14. K. H. Kim, S. Gaba, D. Wheeler, J. M. Cruz-Albrecht, T. Hussain, N. Srinivasa, W. Lu, A Functional Hybrid Memristor Crossbar-Array/CMOS System for Data Storage and Neuromorphic Applications, *Nano Lett.* 12 (2012) 389–395. doi:10.1021/nl203687n.
15. Y. V. Pershin, M. Di Ventra, Experimental demonstration of associative memory with memristive neural networks, *Neural Networks.* 23 (2010) 881–886. doi:10.1016/J.NEUNET.2010.05.001.
16. A. K. Jain, Jianchang Mao, K.M. Mohiuddin, Artificial neural networks: a tutorial, *Computer (Long. Beach. Calif).* 29 (1996) 31–44. doi:10.1109/2.485891.
17. S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, U. C. Weiser, Memristor-Based Material Implication (IMPLY) Logic: Design Principles and Methodologies, *IEEE Trans. Very Large Scale Integr. Syst.* 22 (2014) 2054–2066. doi:10.1109/TVLSI.2013.2282132.
18. Y. Hirose, H. Hirose, Polarity-dependent memory switching and behavior of Ag dendrite in Ag-photodoped amorphous As_2S_3 films, *J. Appl. Phys.* 47 (1976) 2767–2772. doi:10.1063/1.322942.
19. G. S. Snider, Self-organized computation with unreliable, memristive nanodevices, *Nanotechnology*. 18 (2007) 365202. doi:10.1088/0957-4484/18/36/365202.
20. S. P. Adhikari, Changju Yang, Hyongsuk Kim, L. O. Chua, Memristor Bridge Synapse-Based Neural Network and Its Learning, *IEEE Trans. Neural Networks Learn. Syst.* 23 (2012) 1426–1435. doi:10.1109/TNNLS.2012.2204770.
21. T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, B. Linares-Barranco, STDP and STDP variations with memristors for spiking neuromorphic learning systems, *Front. Neurosci.* 7 (2013) 2. doi:10.3389/FNINS.2013.00002.
22. T. Serrano-Gotarredona, T. Prodromakis, B. Linares-Barranco, A Proposal for Hybrid Memristor-CMOS Spiking Neuromorphic Learning Systems, *IEEE Circuits Syst. Mag.* 13 (2013) 74–88. doi:10.1109/MCAS.2013.2256271.
23. X. Wang, Q. Shao, P. S. Ku, A. Ruotolo, A memristive diode for neuromorphic computing, *Microelectron. Eng.* 138 (2015) 7–11. doi:10.1016/J.MEE.2014.12.008.
24. Z. Biolek, D. Biolek, V. Biolková, SPICE model of memristor with nonlinear dopant drift. *Radioengineering*, 18 (2009) 210–4.
25. D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams, The missing memristor found, *Nature*. 453 (2008) 80–83. doi:10.1038/nature06932.
26. A. S. Alexandrov, A. M. Bratkovsky, B. Bridle, S. E. Savel'ev, D. B. Strukov, R. Stanley Williams, Current-controlled negative differential resistance due to Joule heating in TiO_2, *Appl. Phys. Lett.* 99 (2011) 202104. doi:10.1063/1.3660229.
27. F. Corinto, A. Ascoli, Memristive diode bridge with LCR filter, *Electron. Lett.* 48 (2012) 824. doi:10.1049/el.2012.1480.
28. L. Chua, Memristor-The missing circuit element, *IEEE Trans. Circuit Theory.* 18 (1971) 507–519. doi:10.1109/TCT.1971.1083337.
29. M. Markou, S. Singh, Novelty detection: a review—part 2:: neural network based approaches, *Signal Processing*. 83 (2003) 2499–2521. doi:10.1016/J.SIGPRO.2003.07.019.
30. M. Al-Shedivat, R. Naous, G. Cauwenberghs, K. N. Salama, Memristors Empower Spiking Neurons With Stochasticity, *IEEE J. Emerg. Sel. Top. Circuits Syst.* 5 (2015) 242–253. doi:10.1109/JETCAS.2015.2435512.
31. H. de Garis, M. Korkin, The CAM-Brain Machine (CBM): an FPGA-based

hardware tool that evolves a 1000 neuron-net circuit module in seconds and updates a 75 million neuron artificial brain for real-time robot control, *Neurocomputing*. 42 (2002) 35–68. doi:10.1016/S0925-2312(01)00593-8.

32. P. Suffczynski, F. Wendling, J. J. Bellanger, F. H. L. Da Silva, Some Insights Into Computational Models of (Patho)physiological Brain Activity, *Proc. IEEE.* 94 (2006) 784–804. doi:10.1109/JPROC.2006.871773.
33. J. D. Paola, R. A. Schowengerdt, A detailed comparison of backpropagation neural network and maximum-likelihood classifiers for urban land use classification, *IEEE Trans. Geosci. Remote Sens.* 33 (1995) 981–996. doi:10.1109/36.406684.
34. B. Hayes-Roth, An architecture for adaptive intelligent systems, *Artif. Intell.* 72 (1995) 329–365. doi:10.1016/0004-3702(94)00004-K.
35. B. Hayes-Roth, Architectural foundations for real-time performance in intelligent agents, *Real-Time Syst.* 2 (1990) 99–125. doi:10.1007/BF01840468.
36. D. L. Hintzman, Human Learning and Memory: Connections and Dissociations, *Annu. Rev. Psychol.* 41 (1990) 109–139. doi:10.1146/annurev.ps.41.020190.000545.
37. I. Kaastra, M. Boyd, Designing a neural network for forecasting financial and economic time series, *Neurocomputing.* 10 (1996) 215–236. doi:10.1016/0925-2312(95)00039-9.
38. H. Kaeslin, Digital integrated circuit design: from VLSI architectures to CMOS fabrication. *Cambri. Univ. Press*, 43 (2008). doi:10.1109/JSSC.2008.917523.
39. A. M. Kamboh, M. Raetz, K. G. Oweiss, A. Mason, Area-Power Efficient VLSI Implementation of Multichannel DWT for Data Compression in Implantable Neuroprosthetics, *IEEE Trans. Biomed. Circuits Syst.* 1 (2007) 128–135. doi:10.1109/TBCAS.2007.907557.
40. Q. Xia, W. Robinett, M.W. Cumbie, N. Banerjee, T. J. Cardinali, J. J. Yang, W. Wu, X. Li, W. M. Tong, D. B. Strukov, G. S. Snider, G. Medeiros-Ribeiro, R. S. Williams, Memristor−CMOS Hybrid Integrated Circuits for Reconfigurable Logic, *Nano Lett.* 9 (2009) 3640–3645. doi:10.1021/nl901874j.
41. S. P. Mohanty, X. Li, H. Li, Y. Cao, Guest Editorial Special Issue on Nanoelectronic Devices and Circuits for Next Generation Sensing and Information Processing, *IEEE Trans. Nanotechnol.* 16 (2017) 383–386. doi:10.1109/TNANO.2017.2680420.