Article

Linearized Programming of Memristors for Artificial Neuro-Sensor Signal Processing

Changju Yang and Hyongsuk Kim *

Department of Electronic Engineering, College of Engineering at Chonbuk National University,
567 Baekje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do 54896, Korea; ychangju@jbnu.ac.kr
* Correspondence: hskim@jbnu.ac.kr; Tel.: +82-63-270-2477

Abstract: A linearized programming method of memristor-based neural weights is proposed. Memristor is known as an ideal element to implement a neural synapse due to its embedded functions of analog memory and analog multiplication. Its resistance variation with a voltage input is generally a nonlinear function of time. Linearization of memristance variation about time is very important for the easiness of memristor programming. In this paper, a method utilizing an anti-serial architecture for linear programming is proposed. The anti-serial architecture is composed of two memristors with opposite polarities. It linearizes the variation of memristance due to complimentary actions of two memristors. For programming a memristor, additional memristor with opposite polarity is employed. The linearization effect of weight programming of an anti-serial architecture is investigated and memristor bridge synapse which is built with two sets of anti-serial memristor architecture is taken as an application example of the proposed method. Simulations are performed with memristors of both linear drift model and nonlinear model.

1. Introduction

Memristor is a new circuit element postulated by Leon Chua in 1971 [1] and fabricated recently by the Stanley Williams group [2] from Hewlett-Packard (HP). It exhibits excellent features of both memory [3–6] and neuromorphic applications [2,7–11]. For memory applications, it is nonvolatile and has an extremely small size of a few nanometers [3–5]. For neuromorphic applications, it has features of pulse-based operation and adjustable resistance, which are ideal for tuning the synaptic weights of neuromorphic cells [2,7–10,12,13].

A convenient way of programming a memristor with a certain value is by applying a rectangular voltage pulse whose magnitude of voltage is constant during pulse period and zero during non-pulse period. However, the variation of its resistance is a nonlinear function of the applied voltage pulse width even it is a linear model. We call the resistance of memristor memristance.

Fortunately, when two memristors with opposite polarities are combined together, the nonlinearity of memristance is reduced dramatically due to the complementary action of two memristors. Kim et al. presented an efficient weighting circuit for synaptic operation by building a bridge structure combining two opposite anti-serial memristor circuits [7].

Waser's group reported a fabrication result of complementary resistive switch (CRS) consisting of two back-to-back (anti-serial) memristive elements for the construction of large passive crossbar arrays by solving the sneak path problem [14]. Later, the CRS architecture has been further investigated via an analytical approach [15]. T. Liu et al. also reported the Current-Voltage (I-V) characteristics

of antiparallel resistive switches (APRS) that strongly depend on the parameters of the individual switches [16].

In this paper, we propose a linearized programming method utilizing an anti-serial architecture. To program a target memristor, the same type of a subsidiary memristor is prepared and connected to the target memristor in series with opposite polarity. Since composite memristance of the anti-serial circuit is a constant value, the current through the circuit is constant. It follows that the memristance variation of the individual memristor is a linear function of pulse width since the memristance variation is a linear function of charge.

The principle of such linear programming is explained theoretically and verified via simulations in this paper. Section 2 describes a reason that memristor is a useful element for building a neural synapse. Section 3 demonstrates nonlinearity in programing of a memristor with rectangular voltage pulses. To resolve the nonlinearity problem of voltage controlled memristor, an anti-serial architecture is proposed in Section 4. Simulation results to support the proposed idea are provided in Section 4. Section 5 is the conclusion.

2. Memristor as a Promising Element for the Implementation of Neural Synapses

In biological neural systems, each neuron is connected to other neuron through a synapse between them. A synapse is a very special place in a neural cell, where memory and analog multiplication are performed. Figure 1 shows input and output connections of a biological neuron where inputs (dendrites) are denoted as "b" and output (axon) is denoted as "a". Typically, about 10,000 inputs (dendrites) are connected at a single neuron. Thus, the same number of synaptic weights is needed in a neuron. Therefore, the implementation of huge number of neural synapses in a chip is a big roadblock for the development of an artificial neuron system. To make matters worse, one synaptic circuit is composed of many transistors as shown in Figure 2. Therefore, the implementation of an artificial neural system that mimics a biological neural system is seemingly far beyond our reach with the current circuit technologies.
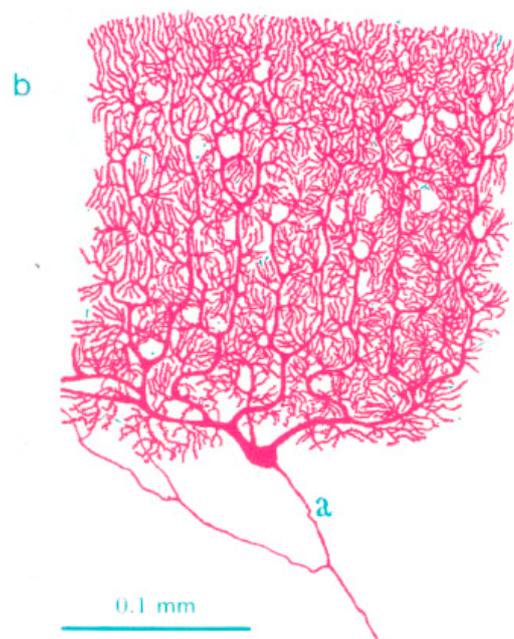
Figure 1. Inputs and outputs of a biological neuron where inputs (dendrites) are denoted as "b" in the figure and outputs (axons) are denoted as "a". Since one synapse is connected at each input (dendrite), a huge number of synaptic weights are composed at a neuron.
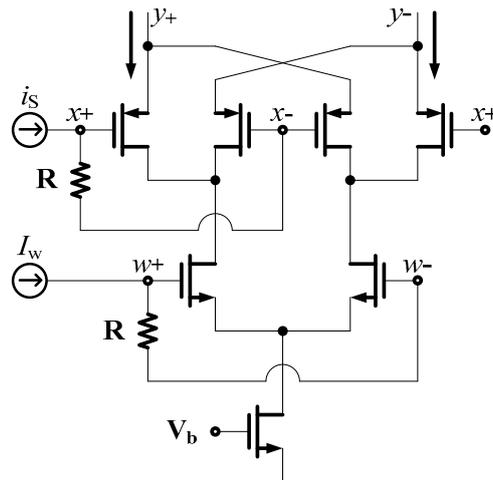
Figure 2. An analog multiplier employed to implement Cellular Neural Networks [17,18].

Figure 3 shows a structure of a memristor fabricated successfully by HP [2]. In an HP TiO$_2$ (Titanium dioxide) memristor model [2], an undoped region with highly resistive TiO$_2$ and a doped region with a highly conductive oxygen vacancy TiO$_{2-x}$ layer are sandwiched between two platinum electrodes. When a voltage or current signal is applied to the device, the border line between the doped and undoped layers shifts as a function of the applied voltage or current. In consequence, the resistance between the two electrodes is altered. Figure 3b,c are equivalent circuits.
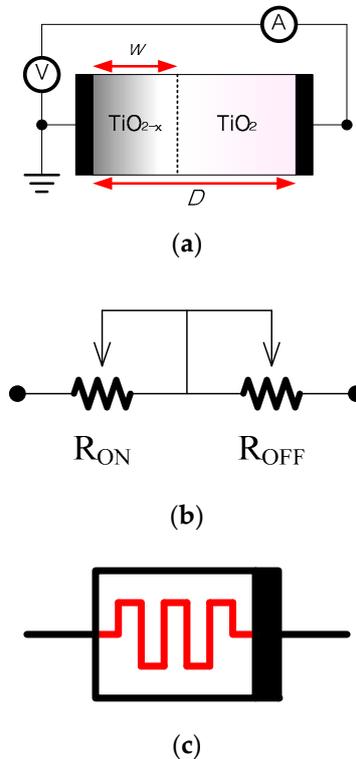

(**a**)

(**b**)

(**c**)

Figure 3. (**a**) Structure of TiO$_2$ memristor, TiO$_{2-x}$ and TiO$_2$ layers are sandwiched between two platinum electrodes. When a voltage/current is applied, its memristance is altered; (**b**) equivalent circuit and (**c**) symbol of the Memristor.

In the TiO$_2$ memristor, a thin titanium dioxide (TiO$_2$) layer and a thin oxygen-poor titanium dioxide (TiO$_{2-x}$) layer are sandwiched between two platinum electrodes. When a voltage or current is applied to the device, the resistance between the two electrodes is altered.

The memristor is defined [7] by

$$v(t) = R(t)i(t) = \frac{d\varphi}{dt} \cdot \frac{dt}{dq}i(t) \tag{1}$$

where $\varphi(t)$ and $q(t)$ denote the flux and charge, respectively, at time t. Thus, the resistance can be interpreted as the slope at the operating point $q = q_Q$ at time t on the memristor φ_q curve. If the φ_q curve is nonlinear; the resistance will vary with the operating point.

Since the flux φ is defined by $\varphi(t) = \int_{-\infty}^{t} v(\tau)\, d\tau$, the resistance of the memristor, called the memristance, M, can be controlled by applying a voltage or current signal across the memristor, where

$$R = M = \left. \frac{d\varphi}{dq} \right|_{(q_Q \cdot \varphi_Q)} \tag{2}$$

Equation (2) shows that a memristor is a kind of resistor that is variable depending upon an operation point on a charge and flux plane. In this sense, it is a programmable resistance.

Let the input of a memristor be a current and the voltage across the memristor be the output of a single memristor circuit as shown in Figure 4. Then, according to Ohm's law, the voltage output is an analog multiplication between the current input and the resistance of a memristor as in Equation (3).

$$v = i \times M \tag{3}$$
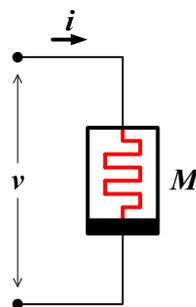
Figure 4. Memristor as an ideal element for a neural synapse where voltage across the memristor is a multiplication between an input current and a memristance.

Therefore, memristor is, in fact, an analog multiplier. Note that the resistance of a memristor is distinguished from that of an ordinary resistor in the sense that its resistance is programmable. It follows that the resistance of a memristor is called memristance. Since the memristance of a memristor can be altered (programmed) by input voltage/current and an analog multiplication is performed within a single device, the memristor is known as an ideal element for the implementation of synapses.

3. Nonlinearity in Memristor

In the TiO$_2$ memristor, a thin TiO$_2$ layer and a thin oxygen-poor TiO$_{2-x}$ layer are sandwiched between two platinum electrodes. The TiO$_2$ layer and the TiO$_{2-x}$ layer are referred to as un-doped, and doped layers, respectively. When a voltage or current is applied to the device, the dividing line between the TiO$_2$ and TiO$_{2-x}$ layers shifts as a function of the applied voltage or current. As a result, the resistance between the two electrodes is altered.

Let D and w denote the thickness of the sandwiched area and the doped area (oxygen deficient area) in the TiO$_2$ memristor, respectively, and let R_{ON} and R_{OFF} denote the resistances at high and low dopant concentration areas, respectively.

The relationship between the flux and the charge of the TiO$_2$ memristor is given by [2]

$$\varphi(t) = R_{off}\left\{ q(t)\left[1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right] - \frac{\mu_v R_{ON}}{2D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)q(t)^2 \right\} + \varphi_0 \tag{4}$$

where μ_v is the dopant mobility and $w(t)/D$ is the state variable x.

The memristance M of a memristor can be computed with $M = \frac{d\varphi}{dq}$ as

$$M = \frac{d\varphi}{dq} = R_{off}\left\{\left[1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right] - \frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)q(t)\right\} \tag{5}$$

Assume that the applied input is a current $i(t)$, then, Equation (5) is

$$M = R_{off}\left\{\left[1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right] - \frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)\int i(t)dt\right\} \tag{6}$$

The derivative of Equation (6) with respect to time gives

$$\frac{dM(t)}{dt} = -R_{off}\frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)i(t) \tag{7}$$

It follows from Equation (7)

$$M(t)\frac{dM(t)}{dt} = -R_{off}\frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)v(t) \tag{8}$$

Assume that a constant voltage V is applied as an input voltage. Integrating both sides of Equation (8) results in

$$\frac{M(t)^2}{2} = C - R_{off}\frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)V \cdot t \tag{9}$$

where C is the constant of integration.

It follows from Equation (9) that $M(t)$ can be written as,

$$M(t) = \sqrt{2\left\{C - R_{off}\frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)V \cdot t\right\}} \tag{10}$$

Equation (10) exhibits a fact that memristance is a nonlinear function of time t. Thus, programming the memristor with an arbitrary value is very difficult.

Differently from the linear drift model described above, the nonlinear phenomenon appears often at the boundaries of nano-scale devices; with even the small voltage applied across nanometer devices, a large electric field is produced, and, therefore, the ion boundary position is moved in a non-linear fashion in nano-scale devices [14].

Several different types of nonlinear memristor models have been investigated [11,19,20]. One of them is the window model in which the state equation is multiplied by window function $F_p(w)$, namely

$$\frac{dw(t)}{dt} = \mu_v\frac{R_{ON}}{D}i(t)F_p(w) \tag{11}$$

where p is an integer parameter and $F_p(w)$ is defined by

$$F_p(w) = 1 - \left(2\frac{w}{D} - 1\right)^{2p} \tag{12}$$

This is called the nonlinear drift model or memristive model. It is difficult to find the solution satisfying both Equations (11) and (12) analytically. However, $w(t)$ can be computed numerically as

$$w(t + \Delta t) = \mu_v \frac{R_{ON}}{D} \left(1 - \left(2\frac{w}{D} - 1 \right)^{2p} \right) \Delta q + w(t) \tag{13}$$

where Δq is the charge increment fed to the memristor during the time interval Δt and computed by integrating the input current as

$$\Delta q = \int I(t)dt = I\Delta t \tag{14}$$

Substituting the value of w from Equation (7) in Equation (8), we get

$$M \approx \frac{R_{ON}}{D} w_0 \left(1 - \frac{R_{OFF}}{R_{ON}} \right) + \frac{R_{ON}}{D} K\Delta q \times F_p(w) \left(1 - \frac{R_{OFF}}{R_{OFF}} \right) \tag{15}$$

The current voltage relationship can be obtained as

$$v(t) = \left\{ \frac{R_{ON}}{D} w_0 \left(1 - \frac{R_{OFF}}{R_{ON}} \right) + \frac{R_{ON}}{D} K\Delta q \times F_p(w) \left(1 - \frac{R_{OFF}}{R_{OFF}} \right) \right\} i(t) \tag{16}$$

Figure 5 shows the graphs of the memristance vs. time and the memristance vs. time charge of the nonlinear models of memristors when a rectangular pulse is applied. The memristance is more nonlinear about time than that of charge. In addition, as the number p becomes smaller, the nonlinearity increases. On the other hand, as the integer p increases, the model tends to the linear model.
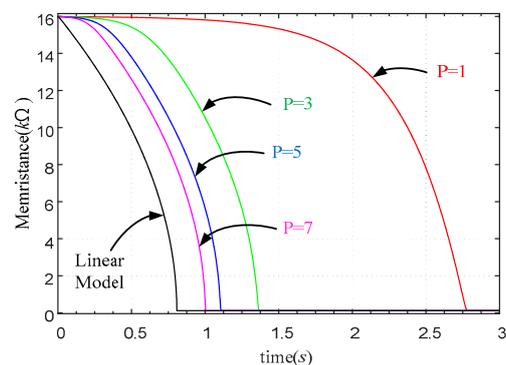

(a)
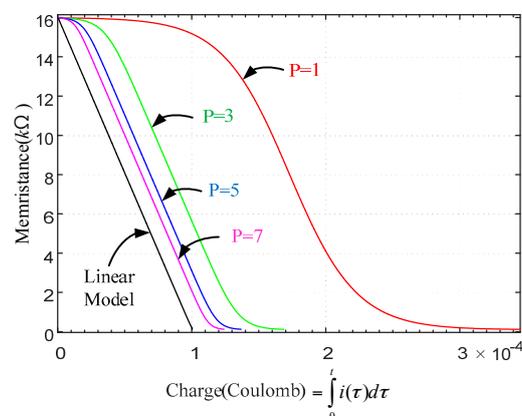
(b)

Figure 5. Memristance variation of a nonlinear memristor of (Equation of Window function) about: (**a**) time; (**b**) charge when a rectangular pulse is applied.

4. Linearization in Memristor Programming with Anti-Serial Architecture

Anti-serial memristor circuit is a circuit of two memristors in serial connection with opposite polarities as shown in Figure 6. When a positive voltage (or current) signal is applied to a circuit with two memristors connected in series, but with opposite polarities, then the memristance of M1 decreases, whereas the memristance of M2 increases. As a result, the composite memristance becomes constant, due to their complementary action [21].

(**a**)

(**b**)
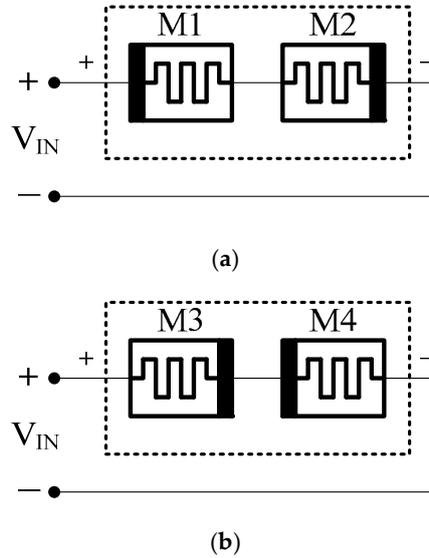
Figure 6. Memristor circuit with two memristors connected anti-serially. (**a**) M1 and M2 connected back to back in series; (**b**) serial connection with opposite polarities using M3 and M4.

Let us assume that the polarity of M1 is the same as that of the predefined reference polarity, and the polarity of M2 is opposite to that of the reference. If charge $q(t)$ is injected into the positive terminal of the composite device, it acts as positive charge for M1, whereas it acts as negative charge for M2. Thus,

$$q_2(t) = -q(t) \tag{17}$$

Similarly, the sign of flux $\varphi_2(t)$ is opposite to the reference, i.e.,

$$\varphi_{c2}(t) = -\varphi_2(t) \tag{18}$$

Thus, flux $\varphi_1(t)$ and $\varphi_2(t)$ of memristor M1 and M2 can be written as functions of charge $q(t)$, as

$$\varphi_1(t) = R_{off} \quad \left\{ q(t) \left[1 + \frac{w_{01}}{D} \left(\frac{R_{on}}{R_{off}} - 1 \right) \right] \right. \\ \left. - \frac{\mu_v R_{on}}{2D^2} \left(1 - \frac{R_{on}}{R_{off}} \right) q(t)^2 \right\} + \varphi_1(0) \tag{19}$$

$$\varphi_2(t) = R_{off} \quad \left\{ q(t) \left[1 + \frac{w_{02}}{D} \left(\frac{R_{on}}{R_{off}} - 1 \right) \right] \right. \\ \left. + \frac{\mu_v R_{on}}{2D^2} \left(1 - \frac{R_{on}}{R_{off}} \right) q(t)^2 \right\} - \varphi_2(0) \tag{20}$$

The total flux $\varphi_C(t)$ is the sum of $\varphi_1(t)$ and $\varphi_2(t)$.

When two memristors are assumed to be identical, and they are in the stable composite memristance state, the flux of the composite memristor becomes

$$\varphi_C(t) = 2R_{off}q(t) \left[1 + \frac{w_0}{D} \left(\frac{R_{on}}{R_{off}} - 1 \right) \right] \tag{21}$$

where $w_{01} = w_{02} = w_0$. Furthermore, the memristance of the composite memristor can be obtained by differentiating Equation (21) with respect to $q(t)$, as

$$M_C = \frac{d\varphi_C(t)}{dq(t)} = 2\left\{R_{off}\left[1 + \frac{w_0}{D}\left(\frac{R_{on}}{R_{off}} - 1\right)\right]\right\} = 2M_O \tag{22}$$

where, M_C is the composite memristance, and M_0 is a constant value of $R_{off}\left[1 + \frac{w_0}{D}\left(\frac{R_{on}}{R_{off}} - 1\right)\right]$.

Note that M_C in Equation (22) is a constant, since all the related parameters of M_0 are constant.

Let M2 is a target memristor to program. From Equation (6), the expression of memristance M_{Target} is

$$M_{\text{target}} = R_{off}\left\{\left[1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right] - \frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)\int i(t)dt\right\} \tag{23}$$

when a rectangular voltage pulse with V volt is applied, current $i(t)$ during a non-zero pulse period can be computed as

$$i(t) = \frac{V}{M_O} \tag{24}$$

Plugging Equation (24) into Equation (23), we obtain

$$M_{\text{target}} = R_{off}\left\{\left[1 + \frac{w_0}{D}\left(\frac{R_{ON}}{R_{OFF}} - 1\right)\right] - \frac{\mu_v R_{ON}}{D^2}\left(1 - \frac{R_{ON}}{R_{OFF}}\right)\frac{V}{M_O}t\right\} \tag{25}$$

All the parameters of the right side of Equation (25) are constant except time t. Therefore, it is a linear function of time. Comparing Equations (25) with (10) which is a nonlinear equation about time t, programing a memristor with Equation (25) is much easier than with Equation (10).

To program a target memristor with this method, the same type of a subsidiary memristor is prepared and connected to the target memristor in series with opposite polarity as in Figure 7. Since composite memristance of the anti-serial circuit is a constant value, the current through the circuit is constant. It follows that the memristance variation of the individual memristor is a linear function of pulse width since the memristance variation is a linear function of charge.
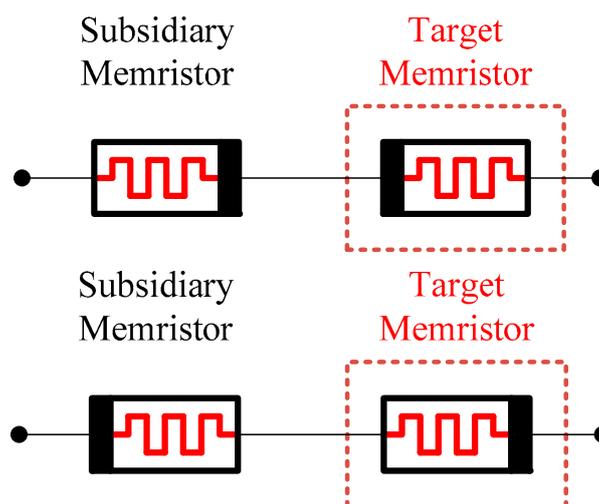

Figure 7. Proposed linearized programming method with anti-serial architecture. It is a circuit of two memristors in serial connection with opposite polarities. Though the individual behavior of memristance variation of two memristors is nonlinear about time, it becomes linear in an anti-serial connection due to the complementary action of two memristors with opposite polarities.

5. Application of the Anti-Serial Memristor Architecture to the Weight Programming of a Memristor Bridge Synapse

Anti-serial connections of memristor circuit are utilized to build a memristor bridge weighting circuit [7], which can be programmed linearly due to the cooperation of two sets of anti-serial memristor circuits.

The memristor bridge circuit consists of four identical memristors with different polarities indicated in Figure 8. When a positive or a negative pulse $V_{in}(t)$ is applied at the input, the memristance of each memristor is increased or decreased linearly depending upon its polarity. For instance, when a positive pulse is applied as input, the memristances of M1 and M4 (whose polarities are forward-biased) will decrease. On the other hand, the memristances of M2 and M3 (whose polarities are reverse-biased) will increase. It follows that the voltage V_A at *node A* (with respect to ground) becomes larger than the voltage V_B at *node B* for a positive input signal pulse. In this case, the circuit produces a positive output voltage V_{out} representing a positive synaptic weight.
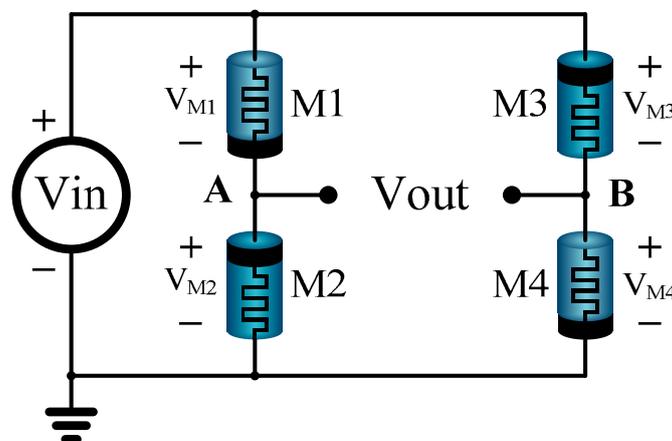
Figure 8. An application of the proposed anti-serial memristor circuit to the implementation of neural networks. With two different types of anti-serial memristor circuits, synaptic weights of neural networks can be programmed linearly about time when a direct current DC voltage signal is applied for the programing of a neural synapse.

On the other hand, when a negative strong pulse is applied, the memristances are varied in the opposite direction and the voltage at *node B* becomes larger than that at *node A*. In this case, the circuit produces a negative output voltage V_{out} representing a negative synaptic weight.

6. Simulation

The linearity in programing with several memristor circuits such as single and anti-serial circuit has been tested. The memristor models employed for these simulations are a linear drift model of HP TiO$_2$ [2] and a nonlinear model with window function with $p = 1$ [16].

When we want to program a memristor to a certain memristance value, one of the easiest ways is by applying a constant voltage or current for a certain length of time. If memristance variation about time is linear, the desired value of memristance can be programmed easily since the programmed memristance would be proportional to the width of a voltage pulse.

Figure 9 shows the memristance variation about time when a constant voltage of 1 V is applied to a TiO$_2$ memristor model. Though the memristance curve is linear about charge, it is non-linear about applied time (pulse width) as shown with a solid line in Figure 9. Such a nonlinearity about time makes the programing of a memristor difficult. The desirable memristance curve is the dotted line.
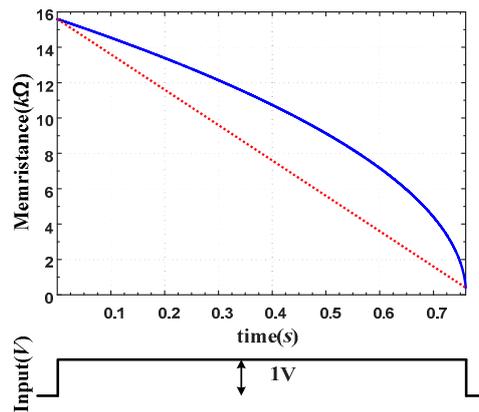
Figure 9. Nonlinear variation of the memristance about time of TiO$_2$ memristor model when a constant voltage (rectangular pulse) is applied.

Figure 10 shows a memristance variations of two memristors in the proposed anti-serial connection in Figure 7 when a constant voltage source is applied. Upon applying a constant voltage source to the anti-serial circuit, the subsidiary and target memristors are cooperating in complimentary fashion. The resultant memristance variation is linearized about time as shown in the figure.
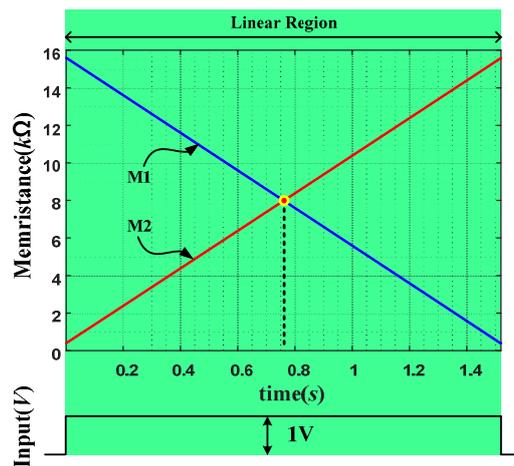
Figure 10. Linear variation of memristance when a constant voltage source is applied to the proposed anti-serial connection of two memristors.

Simulations to demonstrate the linearity in programming the memristor bridge synapse circuit have also been performed. If the memristance variation of each memristor of anti-serial memristor circuit is linear, the voltage change at each node of the circuit is supposed to be linear. In consequence, the weight of the memristor bridge synapse can be programmed linearly since the memristor bridge synapse circuit is composed of two different sets of anti-serial circuits.

Figure 11 shows a weight programming scenario where Figure 11a is the changes of two voltages (*Vp* and *Vn*) at middle points starting with 15.6 kΩ. When +1 V is applied for a long time as in Figure 11e, memristances of M1 and M4 are reduced gradually until 400 Ω is reached while M2 and M3 are kept with its highest memristance 15.6 kΩ. The change of memristance is a nonlinear function about time during this period. Then, when −1 V is applied afterwards, the memristances of M2 and M3 are reduced linearly while those of M1 and M4 are increased until M2, M3 and M1, M4 reach the minimum and maximum values, respectively. Then, when +1 V is applied again, the memristances of M2 and M3 are increased while those of M1 and M4 are decreased linearly.
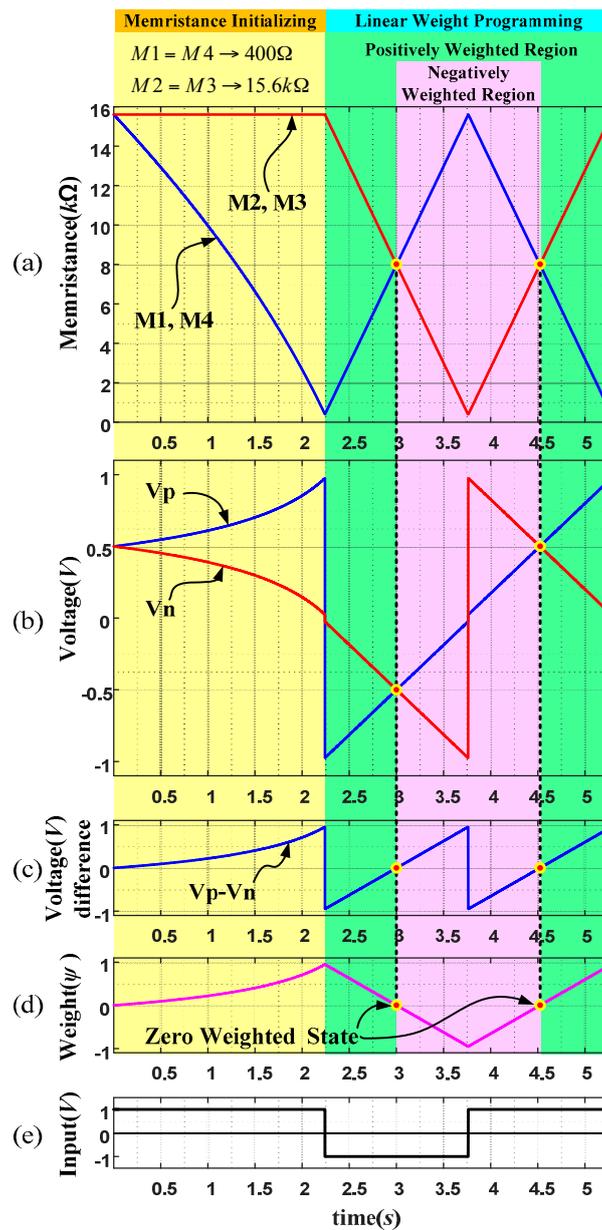
Figure 11. Linear programming of the memristor bridge synapse that is composed of two different types of anti- serial linear memristors. (**a**) variation of memristances; (**b**) changes of voltages *Vp* and *Vn*; (**c**) difference of middle voltage (*Vp* − *Vn*); (**d**) weight changes of the memristor bridge synapses; and, (**e**) wide pulse for programming.

Figure 11b shows the changes of voltages *Vp* and *Vn* which are the voltages at middle points of two anti-serial circuits during the period of Figure 11a. As seen in the middle of the figure, voltage changes linearly about time due to the linear change of memristances in Figure 11a. As the result, the voltage difference of middle voltage (*Vp* − *Vn*) is also linear as in Figure 11c and finally, weight changes of the memristor bridge synapses becomes also linear as in Figure 11d.

The effect of the nonlinearity of memristor models to the performance of our memristor circuit has also been investigated. Figure 12a shows the memristance variations for each memristor of anti-serially connected non-linear memristor circuit. It is assumed that M1 and M2 have initial values of 15.6 kΩ and 400 Ω, respectively. When a positive DC input with +1 V is applied at the anti-serial circuit, the memristance of M1 is reduced and that of M2 is increased. Observe the region

at the middle where graphs are almost linear though some nonlinear regions can be seen at the end of the curves. Since weighting operations are performed only in the linear region at the center as indicated with a box, linear programming can be performed with the proposed anti-serial architecture. Meanwhile Figure 12b is the variations of memristances when the programming signal is applied at individual memristors. As seen in the figure, the memristance changes as functions of time are highly nonlinear in all the range of the curves.


(**a**)

(**b**)
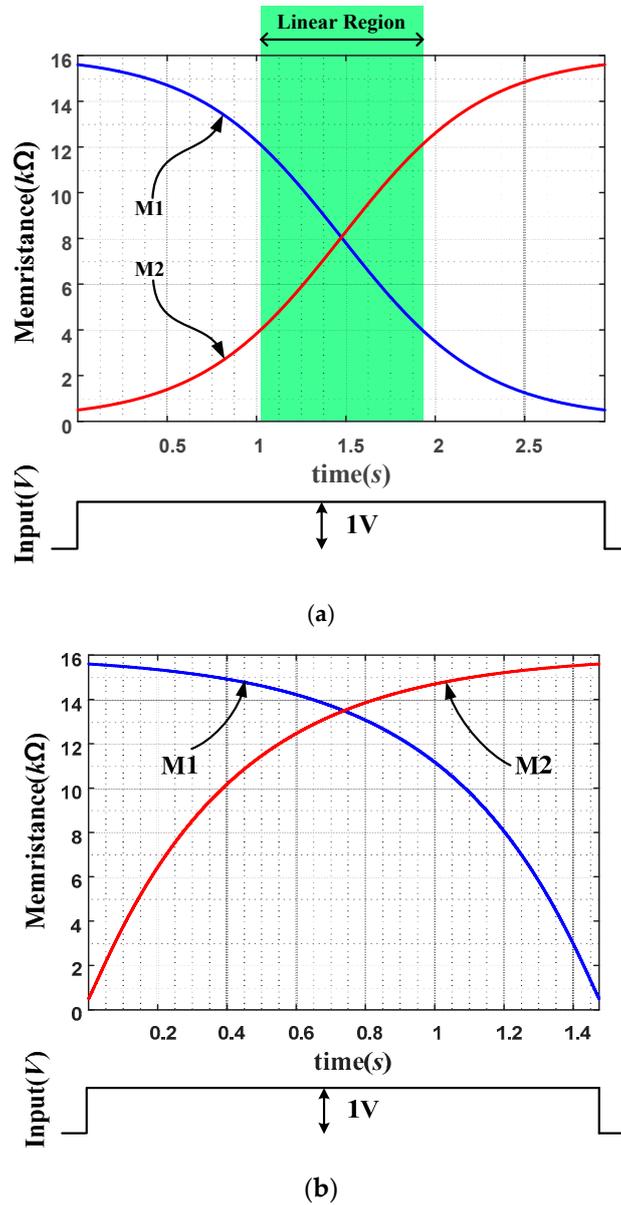
Figure 12. Linearized variation of memristance at the middle of the graph when a constant voltage source is applied at an anti-serial connection of two nonlinear memristors (**a**) and nonlinear variation of memristance of individual memristors for comparison when the programming signal is applied at each memristor individually (**b**).

Comparing Figures 12a and 12b, the memristances of anti-serial memristors are linearized significantly than those without the anti-serial connection.

The effect of weight programming of memristor bridge synapses with non-linear model of memristors is also investigated. Figure 13 shows parameter variation of a memristor bridge synapse

while the programming input signal as in Figure 13e is applied. All the necessary arrangements are the same as the linear memristor case in Figure 11 except that nonlinear memristors are employed.

Observe the memristance especially in Figure 13a and weight variation in Figure 13d at the time periods of [5.355, 6.247] s and [8.318, 9.209] s are all linearized very much.
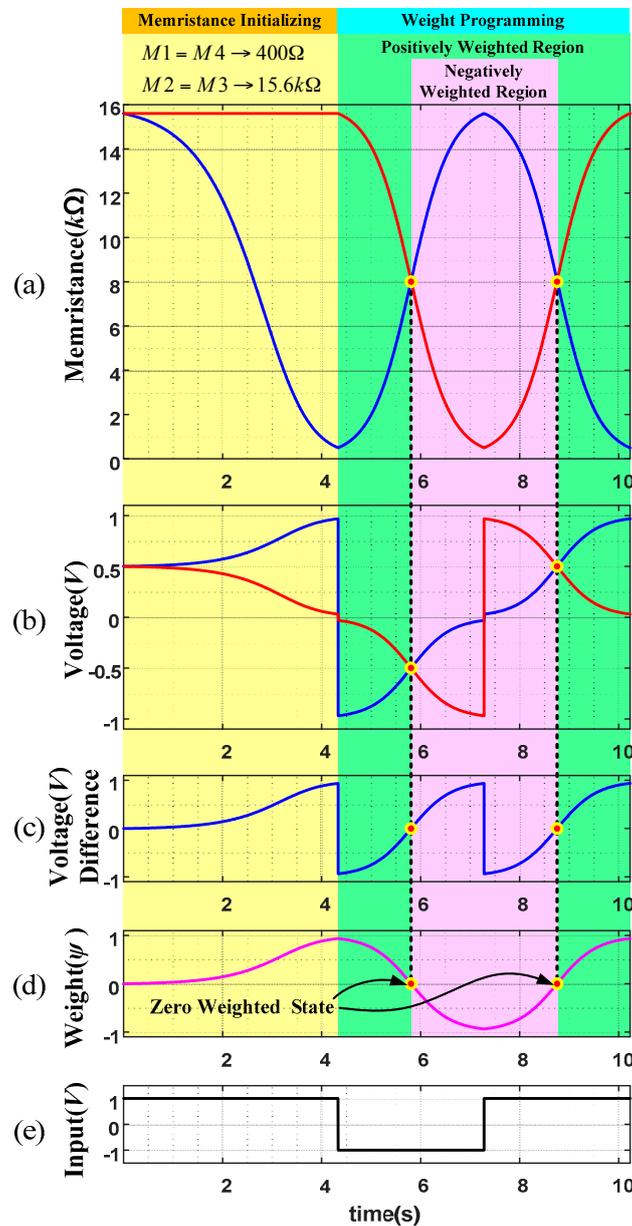
Figure 13. Linearized programming of the memristor bridge synapse which is composed of two different types of anti-serial non-linear memristors. (**a**) variation of memristances; (**b**) changes of voltages *Vp* and *Vn*; (**c**) difference of middle voltage (*Vp* − *Vn*); (**d**) weight changes of the memristor bridge synapses; and, (**e**) wide pulse for programming.

7. Conclusions

In neuromorphic applications of memristors, a linear programming of memristance about time is important. In this paper, we proposed a method utilizing an anti-serial architecture.

Anti-serial architecture is a serial connection of two memristors with opposite polarities. It exhibits linearization in programming due to a complimentary action of two memristors; when the memristance

of one memristor increases, the other decreases. Since composite memristance of the anti-serial circuit is a constant value, the current through the circuit is constant. It follows that the memristance variation of the individual memristor is a linear function about pulse width since the memristance variation is a linear function of charge.

Our proposed idea of linear programming of a memristor is by employing an additional subsidiary memristor with an opposite polarity when programming a target memristor is needed. When programming a target memristor is needed, a subsidiary memristor with an opposite polarity is prepared so that the subsidiary and target memristors construct an anti-serial architecture.

The validity of the proposed idea has been proved with linear drift model of HP TiO$_2$ memristor. In addition, it has been applied in building a memristor synapse circuit that is composed of two different sets of anti-serial architectures. Due to the anti-serial architecture, weights have been programmed linearly about applied pulse width.

The proposed architecture has also been tested with memristor models of highly nonlinear characteristics. Memristances of the anti-serial memristor circuits and weights of memristor bridge synapse circuits are all linearized significantly around zero memristance and zero weight regions, respectively.

Author Contributions: Changju Yang designed the circuits, performed the experiments and conducted the data analysis. All authors wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chua, L.O. Memristor-The missing circuit element. *IEEE Trans. Circuit Theory* **1971**, *18*, 507–519. [CrossRef]
2. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S. The missing memristor found. *Nature* **2008**, *453*, 80–83. [CrossRef] [PubMed]
3. Aono, M.; Hasegawa, T. The Atomic Switch. *Proc. IEEE* **2010**, *98*, 2228–2236. [CrossRef]
4. Kozicki, M.N.; Gopalan, C.; Balakrishnan, M.; Mitkova, M. A Low-Power Nonvolatile switching element based on copper-tungsten oxide solid electrolyte. *IEEE Trans. Nanotechnol.* **2006**, *5*, 535–544. [CrossRef]
5. Waser, R.; Aono, M. Nanoionics-based resistive switching memories. *Nat. Mater.* **2007**, *6*, 833–840. [CrossRef] [PubMed]
6. Buscarino, A.; Fortuna, L.; Frasca, M.; Gambuzza, L.V. A chaotic circuit based on Hewlett-Packard memristor. *Chaos* **2012**, *22*, 023136. [CrossRef] [PubMed]
7. Kim, H.; Sah, M.P.; Yang, C.; Roska, T.; Chua, L.O. Memristor bridge synapses. *Proc. IEEE* **2012**, *100*, 2061–2070. [CrossRef]
8. Kim, H.; Sah, M.P.; Yang, C.; Roska, T.; Chua, L.O. Neural Synaptic Weighting with a pulse-based memristor circuit. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2012**, *59*, 148–158. [CrossRef]
9. Jo, S.H.; Chang, T.; Ebong, I.; Bhadviya, B.B.; Mazumder, P.; Lu, W. Nanoscale memristor device as synapse in Neuromorphic Systems. *Nano Lett.* **2010**, *10*, 1297–1301. [CrossRef] [PubMed]
10. Snider, G. Self-organized computation with unreliable, memristive nanodevices. *Nanotechnology* **2007**, *18*, 365202. [CrossRef]
11. Choi, H.; Budhathoki, R.K.; Park, S.; Yang, C.; Kim, H. Linear Programming of Voltage-Controlled Memristors with an Anti-Serial Memristor Circuit. In Proceedings of the 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, Portugal, 24–27 May 2015; pp. 1142–1145.
12. Gambuzza, L.V.; Buscarino, A.; Fortuna, L.; Frasca, M. Memristor-Based Adaptive Coupling for Consensus and Synchronization. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2015**, *62*, 1175–1184. [CrossRef]
13. Bilotta, E.; Pantano, P.; Vena, S. Speeding up cellular neural network processing ability by embodying memristors. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**. [CrossRef] [PubMed]
14. Linn, E.; Rosezin, R.; Kügeler, C.; Waser, R. Complementary resistive switches for passive nanocrossbar memories. *Nat. Mater.* **2010**, *9*, 403–406. [CrossRef] [PubMed]

15. Kavehei, O.; Al-Sarawi, S.; Cho, K.; Eshraghian, K.; Abbott, D. An Analytical Approach for Memristive Nanoarchitectures. *IEEE Trans. Nanotechnol.* **2012**, *11*, 374–385. [CrossRef]

16. Liu, T.; Kang, Y.; Verma, M.; Orlowski, K. Switching Characteristics of Antiparallel Resistive Switches. *IEEE Electron Device Lett.* **2012**, *33*, 429–431. [CrossRef]

17. Dominguez-Castro, R.; Espejo, S.; Rodriguez-Vazquez, A.; Carmona, R.A.; Foldesy, P.; Zarandy, A.; Szolgay, P.; Sziranyi, T.; Roska, T. A 0.8-μm CMOS two-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage. *IEEE J. Solid-State Circuits* **1997**, *32*, 1013–1026. [CrossRef]

18. Cruz, J.M.; Chua, L.O. A 16 × 16 Cellular Neural Network Universal Chip: The First Complete Single-Chip Dynamic Computer Array with Distributed Memory and with Gray-Scale Input-Output. *Analog Integr. Circuits Signal Process.* **1998**, *15*, 227–237. [CrossRef]

19. Joglekar, Y.N.; Wolf, S.J. The elusive memristor: Properties of basic electrical circuits. *Eur. J. Phys.* **2009**, *30*, 661–685. [CrossRef]

20. Pickett, M.D.; Strukov, D.B.; Borghetti, J.L.; Yang, J.J.; Sinder, G.S.; Stewart, D.R.; Williams, R.S. Switching dynamics in titanium dioxide memristive devices. *J. Appl. Phys.* **2009**, *106*, 074508. [CrossRef]

21. Budhathoki, R.K.; Sah, M.P.; Adhikari, S.P.; Kim, H.; Chua, L.O. Composite Behavior of Multiple Memristor Circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2013**, *60*, 2688–2700. [CrossRef]