electronics

MDPI

Article

Analysis of Memristor-Based Neural Networks and Logic Circuits for Artificial Intelligence Using Standard and Improved Memristor Models

Stoyan Kirilov *, Georgi Tsenov and Valeri Mladenov

Department of Fundamentals of Electrical Engineering, Faculty of Automatics, Technical University of Sofia, 8 Kliment Ohridski Blvd., 1000 Sofia, Bulgaria; gogotzenov@tu-sofia.bg (G.T.); valerim@tu-sofia.bg (V.M.)
* Correspondence: s_kirilov@tu-sofia.bg; Tel.: +359-2-965-3319

Abstract

Memristors are state-of-the-art electronic elements with nano sizes, about 3 nm dimensions, with very good nano-second switching and memory properties, low power usage of about 100 μW, and good compatibility with the current technology of CMOS-integrated chips and circuits. These components are potentially applicable in T-byte memory arrays, artificial neural networks, logic gates and many other digital and analog electronic schemes and devices for artificial intelligence. This paper presents the application of some simple and fast-operating modified memristor models with activation thresholds in neural networks and logic circuits. MATLAB ver. 2016a and LTSPICE ver. XVII products are used for the analysis of memristor neural nets and logical circuits for artificial intelligence. Several simple, accurate and fast-operating existing modified memristor models, together with several frequently used standard memristor models, are utilized for the associated analyses and simulations. A comparison between the used memristor models is conducted. The considered memristor models are tuned, using experimentally recorded *i*-*v* relations of tungsten-sulfide Knowm memristors. An accurate functioning of the analyzed neural nets and logic functions is confirmed by the derived results. The considered modified memristor models, neural networks and logic schemes are important in modeling and analysis of memristor-based circuits for ultra-high-density artificial intelligence-integrated chips.

Keywords: memristor neural network; memristor-based logic circuits; artificial intelligence; improved memristor models

Check for updates

Academic Editors: Paris Kitsos and Jinjun Feng

Received: 15 April 2026
Revised: 3 June 2026
Accepted: 15 June 2026
Published: 18 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Memristors are promising nano-size electronic elements with very good memory and switching features. They were predicted by L. Chua in 1971 as the missing physical connection between the magnetic flux and electric charge for the main passive two-terminal elements [1–3]. The memory properties of memristors are connected to the accumulation of electric charge and retention of their state after turning the sources off. Different types of memristors are available [4–6]. The scientific literature reports memristors based on titanium dioxide [3], hafnium oxide [4], tantalum oxide [5,6], niobium oxide [2], and ferroelectric, polymeric, spintronic and other composite materials [7–10]. In recent years, memristors that are based on metal oxides have been under intensive investigation. They have stable characteristics, nano-dimensions, significantly low power consumption, and a strong compatibility with the present CMOS ultra-high-density integrated chips

and circuits [10]. Memristors are potentially applicable in different electronic circuits like in neural networks [11,12], memory arrays, reconfigurable electronic circuits and devices—amplifiers, oscillators, filters, logic gates and many others [13–16]. In memristor-based memory crossbars, they are used for storing logical information. Their switching properties are significant for fast commutation in such schemes [17–19]. In neural networks, memristors are mainly used as synapses for storing the synaptic weights [18,19]. The preliminary design and analysis of complex memristor-based electronic circuits require appropriate software products, such as SPICE and fast-operating memristor models [20,21]. In recent years, LTSPICE has been desired by many design engineers, owing to its free license, user-friendly and simple interface, and improved convergence compared to the other SPICE family products [21]. The number of electronic elements and their connections is not restricted in LTSPICE environments and this makes computer simulations of complex electronic circuits possible. LTSPICE software is available for free download on the website of Analog Devices.

Some standard memristor models, such as those of Stanley Williams [1], Joglekar [22], Biolek [23], Laiho–Lehtonen, Knowm [24,25] and many others [26,27], have specific drawbacks, alongside their positive features. The Williams, Joglekar and Biolek memristor models are simple and fast-operating, but their accuracy is low in comparison with other models and they do not completely represent some specific physical phenomena, such as the nonlinear ionic dopant drift at high-level voltage signals. Laiho–Lehtonen [26,27] and Knowm [24] models and some memristor models for tantalum-oxide memristors [5,6] have good precision, but they contain many mathematical operations and hence, they require significant hardware resources for their simulations. Such memristor models are not quite appropriate for analysis and simulations of complex memristor-based electronic circuits. For avoidance of the main drawbacks of several standard memristor models, some modified and improved memristor models are proposed and used [28–30]. They are mainly based on the standard Joglekar, Biolek and Lehtonen–Laiho memristor models. They are improved by simplification and approximation of their main equations. For a partial decrease in some related convergence issues, a smooth and differentiable logarithmic–sigmoidal step-like function is applied in the modified models instead of the standard Heaviside step function [21,27]. The memristor models that are modified can operate correctly at low, middle and high-frequency signals. Activation voltage thresholds are applied in many of the modified memristor models, and thus they enable their application in neural networks, where different operating modes are needed for tuning the synaptic weights and for utilization of the neural networks at fixed synaptic weights [27].

The main purpose of this article is to express a detailed comparison between some of the existing modified memristor models, together with several standard models, applied in memristor-based neural networks and logic circuits for systems with artificial intelligence.

To achieve this purpose, several related research tasks are defined and resolved. The memristor models under analysis are applied and analyzed in a memristor-based neuron, a neural network with memristor-based synapses, and several memristor-based logic gates and circuits. A detailed comparison of the modified memristor models is performed, together with the standard Joglekar, Biolek, and Knowm models, utilizing some important criteria for comparison, such as the simulation time, the related model's complexity, the switching properties, and the operating frequencies [21,26]. The main advantages of the considered memristor models that are modified, compared to the Joglekar and Biolek models, include a higher speed of operation, excellent switching properties, good accuracy, higher operating frequencies, and their applicability for analysis of neural networks and logic circuits. Compared to the standard Knowm memristor model, the modified models under analysis are more simplified, fast-operating and have lower simulation time. The

LTSPICE code of the considered standard and the modified memristor models and circuits is readily available for download at the following GitHub free version page: https://github.com/mladenovvaleri/Advanced-Memristor-Modeling-in-LTSpise [21] (accessed on 7 April 2026).

The rest of the article structure follows with Section 2 that shortly describes the memristors' operation and function, their main standard models, some existing modified models, their tuning, according to experimental data of Knowm tungsten-sulfide memristors, and their LTSPICE implementation. Section 3 presents a memristor-based neuron, a simple memristor-based neural network and their analysis. In Section 4, memristor-based logic gates and circuits are analyzed. A brief comparison is presented of the used standard and modified memristor models in Section 5. A discussion is included in Section 6, and Section 7 is the conclusion of the article.

2. Memristors—Structure, Operation, Modeling, Tuning and LTSPICE Implementation

2.1. Memristor Structure and Operation

Memristors are passive, nonlinear and two-terminal electronic elements with memory and switching properties [1]. They are mainly based on oxides and chalcogenides of transition metals, partially doped with oxygen deficiencies [1,31]. A memristor's resistance $M(x)$, also known as memristance, depends on the stored electric charges, which are proportional to the integral of the applied voltage over time [1].

When the external electric sources are off, the memristor retains its previous state, representing memory effects. The state variable x is one of the main important quantities related to memristors, and it represents the accumulated charges and the corresponding memristance. It is limited in the range between zero and one, owing to geometrical considerations. Under an applied voltage signal, a memristor can change its memristance within a broad range, which is related to its switching properties [26,27].

Usually, memristors can operate in soft-switching and hard-switching modes [27]. If the state variable remains within its internal bounds, the memristor operates in a soft-switching regime. When the state variable reaches either limit, the device enters hard-switching, in which it effectively behaves as a rectifying element [1–3].

2.2. Memristor Modeling

2.2.1. General Model for Memristors

In real memristor elements, the physical processes in the electromagnetic field are quite complex and can be approximately described. The electric current in memristors is related to ionic transport, which in most cases is a nonlinear phenomenon, dependent on the applied voltage. Sometimes, the nonlinear ionic dopant drift is represented by suitable polynomial functions in the state differential equation of memristor models. In some cases, memristor models are based on the processes of filament formation and rupture in the memristor nanostructures. In experimental analyses of metal–oxide and tungsten-chalcogenide Knowm memristors, physical ionic transport is related to some extent to thermal effects, when the physical parameters slightly change with the temperature in the room-temperature range. The switching dynamics and properties of the memristor models that are under analysis are linked to the corresponding operational behavior of the device and accurate representation of experimental current–voltage characteristics.

At the same time, physics-based memristor models are quite complex and their SPICE implementations are related to many convergence problems, which make simulations of electronic circuits almost impossible. To solve such problems, simplified and quick-operating memristor models have been frequently used in recent years for the analysis and

design of memristor-based electronic circuits. They are based on physical models but are quite simplified and approximated [21].

Usually, a general model of the memristor element can be expressed with a two-equation system, similar to the following System (1) [27]:

$$\left| \begin{array}{l} i = v[M(x)]^{-1}, x_{t=0} = x_0 \\ \frac{dx}{dt} = ki(v) f_W(x) \end{array} \right. \quad (1)$$

where i and v are the memristor current and voltage, x_0 is the initial value of the state variable x, k is a parameter dependent on the physical properties of the used material, and $f_W(x)$ is a window function, applied for restriction of the state variable x and for a limited representation of some specific boundary effects [26].

In the next sub-section, some standard memristor models are considered.

2.2.2. Standard Memristor Models

One of the well-known and frequently used standard memristor models is the Joglekar model [22]. It is provided in the following System (2) [22]:

$$\left| \begin{array}{l} i = v[R_{ON}x + R_{OFF}(1-x)]^{-1} \\ \frac{dx}{dt} = ki\left[1 - (2x-1)^{2p}\right], x_{t=0} = x_0 \end{array} \right. \quad (2)$$

where R_{ON} and R_{OFF} are the limiting ON-state and OFF-state resistances of the memristor, and p is a positive integer exponent, determining the nonlinearity of the Joglekar window function [22,26].

Another frequently used standard memristor model is that of Biolek [23]. It is described with (3) [23]:

$$\left| \begin{array}{l} i = v[R_{ON}x + R_{OFF}(1-x)]^{-1} \\ \frac{dx}{dt} = ki\left[1 - (x - stp(-i))^{2p}\right], x_{t=0} = x_0 \end{array} \right. \quad (3)$$

where "stp" is the classical Heaviside step function [23].

Biolek and Joglekar models of memristors are without activation thresholds and due to this, they are not applicable in neural networks, where different operating modes are required. They are applicable in memristor-based logic gates, where memristors usually operate in a hard-switching mode.

The next standard model under consideration is the Knowm memristor model [24]. It was specially created by Nugent and Molter for modeling of self-directed channel chalcogenide-based memristors [24,25]. It is represented by the following System (4) [24]:

$$\left| \begin{array}{l} i = v\left(\frac{x}{R_{ON}} + \frac{1-x}{R_{OFF}}\right); x_{t=0} = x_0 \\ \frac{dx}{dt} = \frac{1}{\tau}\left[\frac{1-x}{1+\exp(-\beta(v-v_{ON}))} - \frac{x}{1+\exp(\beta(v-v_{OFF}))}\right] \end{array} \right. \quad (4)$$

where τ is a time constant, β is a tuning parameter of the model, and v_{ON} and v_{OFF} are voltage thresholds [24].

A standard memristor model with voltage thresholds [32,33].

In [32,33], a memristor model that is voltage-controlled with voltage thresholds is proposed. It is provided in the following System (5) [32]:

$$\begin{cases} i = v[R_{ON}x + R_{OFF}(1-x)]^{-1}, x_{t=0} = x_0 \\ \frac{dx}{dt} = k\frac{i_{off}}{i-i_0}\left[1-(2x-1)^{2p}\right], v > v_{tplus} > 0 \\ \frac{dx}{dt} = 0, v_{tminus} \le v \le v_{tplus} \\ \frac{dx}{dt} = k\frac{i}{i_{on}}\left[1-(2x-1)^{2p}\right]c, v < v_{tminus} < 0 \end{cases} \quad (5)$$

where the first equation relates to the current and voltage and is the same as in the Joglekar model [22]. The state differential equation is divided into three sub-expressions, according to the value of the applied voltage. If the memristor voltage v is higher than the positive threshold v_{plus}, then the rate of change in the state variable is expressed with the second expression of the system. When the memristor voltage is between the activation thresholds v_{minus} and v_{plus}, then the state variable remains unchanged and its derivative is zero. If the voltage is lower than the negative voltage threshold, then the final equation is considered. The quantities i_{off}, i_0 and i_{on} are tuning parameters.

This standard model [32] is applicable in implementation of synapses in neural networks. In the process of the usage of the neural network, the memristors are used as synaptic weights with constant memristances, and they must operate with voltages that are between the activation thresholds. During the phase of learning of memristor neural networks, the synaptic weights are updated, and this is related to the change in their memristances by pulses. In this case, the applied pulses must be of sufficient amplitude to exceed the positive threshold v_{plus} or lower than the negative threshold v_{minus}.

In the next sub-section, several existing modified memristor models are presented.

2.2.3. Existing Modified and Improved Memristor Models

An existing modified and improved memristor model, abbreviated as model A_{11}, is presented with the following System (6) [28]:

$$\begin{cases} i = v[R_{ON}x + R_{OFF}(1-x)]^{-1} \\ \frac{dx}{dt} = k_1 \sinh(k_2 v)\left[1-(x-stp(-i))^{2p}\right], |v| \ge v_{thr} \\ \frac{dx}{dt} = 0, |v| < v_{thr}, x_{t=0} = x_0 \end{cases} \quad (6)$$

where k_1 and k_2 are tuning parameters, and v_{thr} is an activation voltage threshold. This model is based on both Biolek and Laiho–Lehtonen models. It is simpler compared to the Lehtonen–Laiho model. Introducing an activation threshold enables the memristor model to operate in different modes [27]. If the memristor voltage magnitude is under the activation threshold, the device acts as a linear resistor; otherwise, the state variable varies proportionally to the time-integrated applied voltage, and the model has memory and switching properties [27]. The first mode is related to the phase of usage of the memristor-based neural networks, when all the synaptic weights are already adjusted and they have constant values. The respective memristances are also with constant values. The second one is needed in the phase of adjustment of the synaptic weights, where the memristances and the corresponding weights are tuned by positive or negative voltage pulses.

Another existing modified and improved memristor model, denoted as B_{mod}, is presented with the following System (6) [29]

$$\begin{cases} i = v \cdot [G_{ON}x + G_{OFF}(1-x)]; x(t=0) = x_0 \\ \frac{dx}{dt} = k_1 v^m \sin^2(\pi x)\{1+\exp[k_2(|v| - v_{thr})]\}^{-1} \end{cases} \quad (7)$$

where G_{ON} and G_{OFF} are the ON-state and OFF-state conductances, also known as memductances, k_1 and k_2 are tuning parameters, and v_{thr} is the activation threshold. In this model, the standard Hann window function is used.

A modified memristor model D_4 [30].

The modified memristor model D_4 [30] is a simple model with a piecewise-linear state–flux relation and self-restriction of the state variable. This model has no window function. It can be expressed with the following System (8) [30]:

$$\left| \begin{array}{l} v = i[R_{ON}x + R_{OFF}(1-x)] \\ \Psi(t) = \frac{2}{c}x_0 - \frac{1}{c} + \int_0^t v(\tau)d\tau \\ x(\Psi) = \frac{1}{2}\left[\frac{1}{2}c\left(\left|\Psi + \frac{1}{c}\right| - \left|\Psi - \frac{1}{c}\right|\right) + 1\right] \end{array} \right. \quad (8)$$

where Ψ is the magnetic flux linkage, expressed as a time integral of the applied voltage, and c is a fitting parameter. In the state algebraic equation, the memristor state variable x is directly represented by an analytical expression as a monotone-increasing and piecewise-linear function of flux linkage [30].

The modified memristor model D_4 does not include activation thresholds and due to this, it is applied only in logic gates and circuits. Only models containing activation thresholds are applicable in neural networks as well.

2.3. Experimental Equipment and Measurements

A schematic representation of the measurement setup, the auxiliary components, the Knowm SDC (Self-directed channel tungsten sulfide) memristor being tested (Knowm Inc., Santa Fe, NM, USA), and their interconnections is shown in Figure 1a. The analyzed memristors are embedded within an integrated circuit. A digital oscilloscope XDS3204AE (OWON Technology, Eindhoven, The Netherlands) and an experimental breadboard shown in Figure 1b are used in the setup. One memristor from the integrated circuit is selected for detailed measurements and analysis. The memristor voltage is generated by a signal generator XDG2060 (OWON Technology, Eindhoven, The Netherlands) presented in Figure 1c and applied to the horizontal channel of the oscilloscope. The electric circuit of the laboratory setup is shown in Figure 1d. A series resistor R = 10 kΩ is included to limit the memristor current and protect the device from high-intensity electrical surges. The voltage drop across this resistor, which is proportional to the memristor current, is fed into the vertical channel of the oscilloscope.

This circuit enables observation and recording of the memristor's *i-v* characteristics under various sinusoidal and pulse excitations with different amplitudes and frequencies. The resulting *i-v* curves are displayed on the oscilloscope screen. As is typical for Knowm devices, the measured memristors exhibit asymmetric current–voltage characteristics.

The acquired data are stored in .csv files, which may be used directly for parameter estimation or first converted to .xls format. These files serve as sources of voltage and current waveforms during the estimation procedures performed in MATLAB, which is described in the following sub-section. The first .xls file contains two columns: the time vector—sampled with a 100 ps step corresponding to the 10 GHz sampling frequency (denoted as t_{out} in Simulink)—and the sampled voltage signal applied to the Simulink memristor model. The second file has the same structure, with its second column containing the experimentally measured memristor current, denoted as i_{mes}.

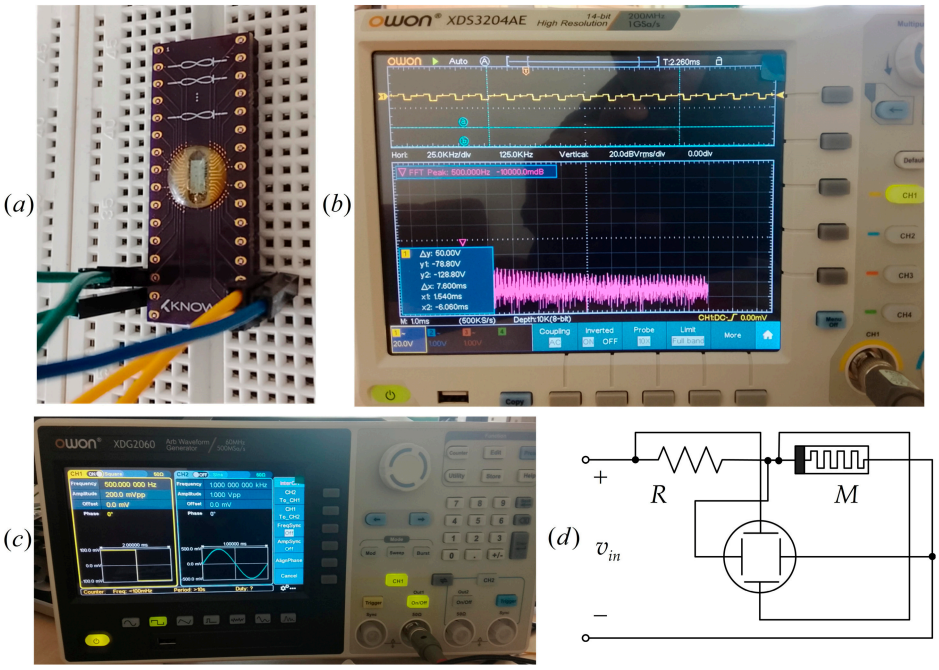


Figure 1. (**a**) Laboratory scheme for obtaining of the current–voltage characteristics of self-directed channel Knowm memristors. (**b**) A digital scope. (**c**) A signal generator. (**d**) An electric scheme of the used laboratory equipment, including a resistor R = 10 kΩ for restriction of the memristor current, and a 4–channel digital oscilloscope with a maximum operating frequency of 100 MHz, a sampling frequency of 1 GHz, a time resolution of 1 ns and a voltage resolution of 2 mV/div in sinusoidal mode with an operating frequency of 10 kHz.

2.4. Tuning Memristor Models in MATLAB-Simulink [34]

For memristor model tuning, the experimental *i-v* relations of self-directed channel Knowm manufactured memristors are used. The adjustment is conducted in a MATLAB-Simulink simulator [34]. Parameter extraction is based on variation in the values of memristor model parameters and a gradient descending algorithm, till the optimal values of the tuning parameters are obtained and the Root Mean Square Error (RMSE) between the experimental and the simulated current–voltage relations is minimized.

A pseudo-code of the tuning process is presented below.

1. Measuring and recording of input data—voltage and current of the Knowm memristors under analysis;
2. Generation and utilization of a Simulink memristor model, containing measured current data for comparison with the simulated data;
3. Simulation and changing the parameters of the model, till the minimum of the error between the experimental and the simulated currents is obtained, or till the previously defined number of iterations is finished. The error is also known as the cost function.

2.4.1. Tuning the Standard Knowm Memristor Model [24,34]

The results from tuning the standard Knowm memristor models are presented in Figure 2. The parameter trajectories during the extraction procedure are presented in Figure 2a. The resulting time waveforms of the memristor voltage, the state variable, and the experimental and simulated currents are shown in Figure 2b. In Figure 2c, the corresponding experimental and simulated current–voltage characteristics are presented and compared, along with the extracted optimal model parameter values.

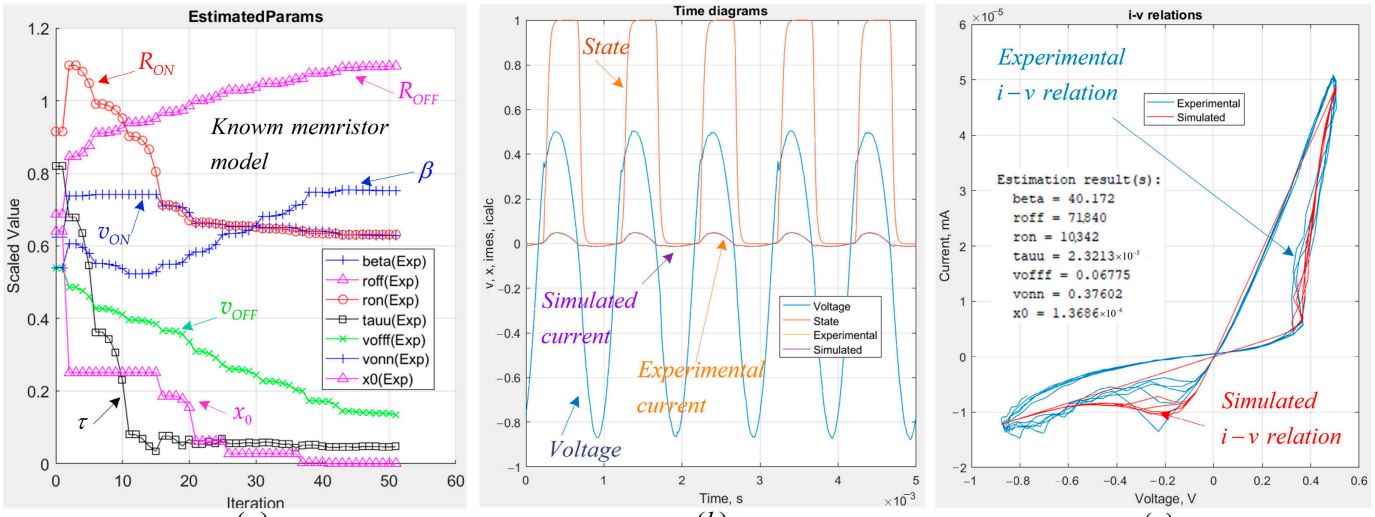


Figure 2. (**a**) Parameter trajectories during the adjustment of the standard self-directed channel Knowm memristor model. (**b**) Time-domain diagrams of the voltage, state variable, and both measured and simulated memristor currents. (**c**) The experimental and modeled *i-v* responses.

2.4.2. Tuning the Modified Memristor Model A_{11} [28]

The results derived from adjustment of the modified memristor model A_{11} [28] are presented in Figure 3. The curves of parameter trajectories obtained during the estimation procedure are shown in Figure 3a. Figure 3b presents the time-resolved waveforms of the applied voltage, the memristor state variable, and both the simulated and measured currents. Figure 3c displays and compares the experimental and simulated current–voltage characteristics, together with the optimally identified model parameter values.

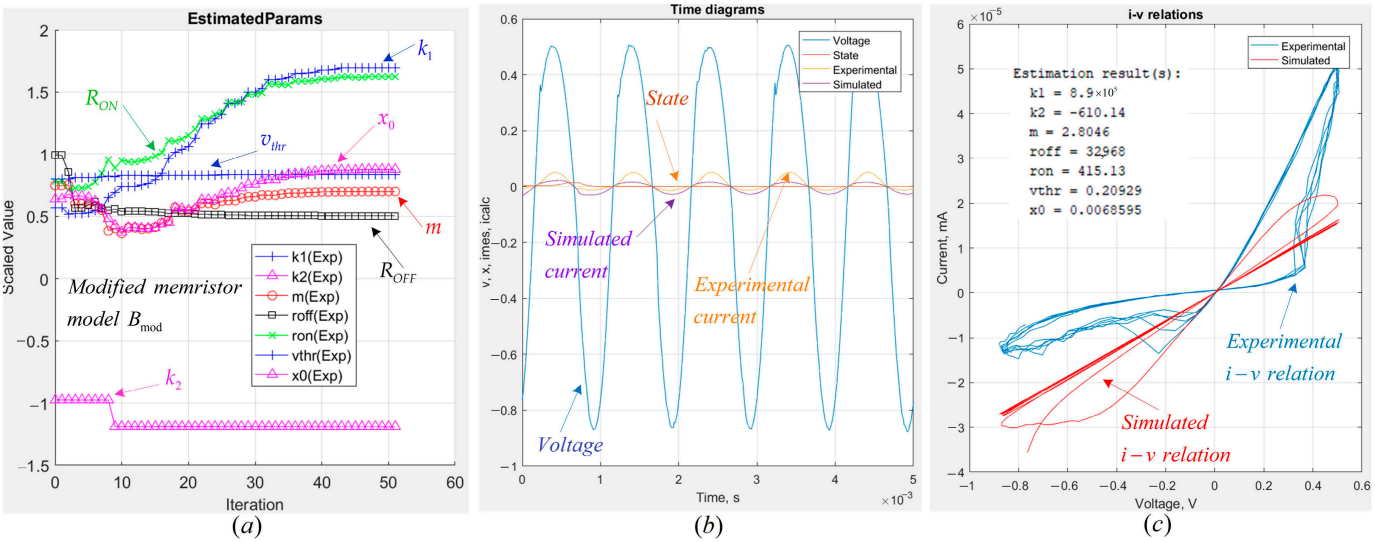


Figure 3. (**a**) Parameter trajectories during adjustment of the modified memristor model A_{11}, using experimental *i-v* relations of Knowm memristors. (**b**) Time-domain diagrams of the memristor voltage, state variable, and the experimental and simulated currents. (**c**) The experimental and simulated current–voltage characteristics of the memristors under study.

The other memristor models considered in this study are tuned using the same procedure. The identified model parameters are subsequently incorporated into their respective LTSPICE memristor implementations. The next sub-section details the LTSPICE implementation and analysis of the previously discussed models.

2.5. LTSPICE Implementation and Analysis of Memristor Models

Based on the previously described mathematical representations of the standard and modified memristor models, the respective LTSPICE library models are constructed. Figure 4 illustrates the equivalent schematic of a general memristor model.

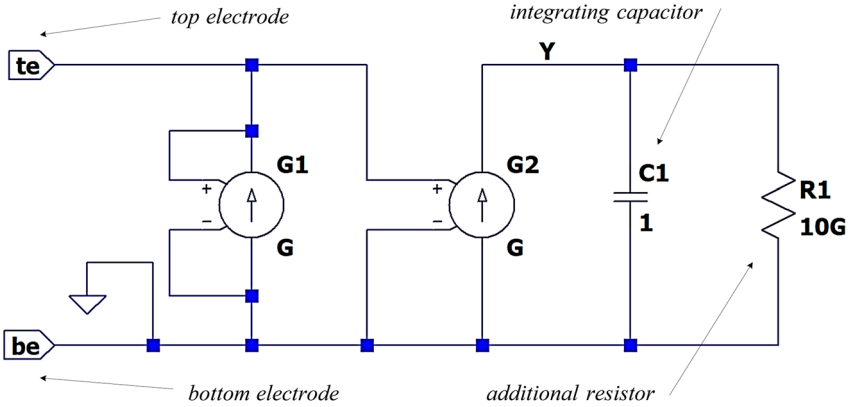


Figure 4. LTSPICE implementation of a memristor model.

The model is based on two voltage-controlled current sources and an integrating capacitor [21,27]. A dependent current source, denoted as G_2, is connected to the integrating capacitor C_1. This source generates a current, which is proportional to the time derivative of the memristor state variable, according to the respective state differential equation of the memristor model. This current is integrated by the capacitor C_1, whose voltage V(Y) is then proportional to the memristor state variable. Usually, an additional high-ohm resistor R_1 is used. It is attached in parallel to the integrating capacitor C_1, for partial avoidance of convergence issues. One of the electrodes of the capacitor is grounded and the other terminal is related to the voltage, proportional to the memristor state variable. It is used by a second voltage-controlled current source, which represents the current memristor current. It is related to the ON-state and OFF-state resistances of the memristor and its state variable. The second current source is connected in parallel to the memristor terminals. According to the described structure, every memristor model could be expressed by a code. As an example, the LTSPICE code of Joglekar memristor model is presented below.

1. subckt K_2 te be Y
2. params ron = 220 roff = 41 $\times$ 10^3 k = 2.1 $\times$ 10^4 pp = 5 x0 = 0.1
3. Cint Y 0 {1}
4. IC V(Y) = x0
5. Rad Y 0 10G
6. Gy 0 Y value = {(k $\times$ V(te,be) $\times$ (1/(ron $\times$ (V(Y)) + roff $\times$ (1 $-$ V(Y)))) $\times$ (1$-$ pow ((2 $\times$ (V(Y)) $-$ 1),(2 $\times$ pp)))))}
7. G1 te be value = {V(te,be) $\times$ ((1/(ron $\times$ (V(Y)) + roff $\times$ (1 $-$ V(Y))))))}
8. ends K_2

The above-discussed code could be used for the generation of a memristor circuit element [21]. The LTSPICE code can easily be changed and might be used in other similar SPICE simulators.

The memristor models under analysis are analyzed in sinusoidal and pulse modes for their comparison, in both MATLAB and LTSPICE. Figure 5 presents the process of tuning the memristor models with positive voltage pulses. The impulses have an amplitude of 0.3 V, a frequency of 500 Hz and a duty cycle of 50%. The time diagram of the pulse sequence is presented in Figure 5a. The corresponding change in the memristance, according to the models under analysis—K_2, K_3, A_{11} and D_4—is visualized in Figure 5b. The change in

memristance during the pulses is almost linear. Model D_4 ensures that the memristance is tuned in a broad range. The difference in the time diagrams is related to the use of different window functions. During the pauses, the memristance does not change, which confirms the memory properties of the models.

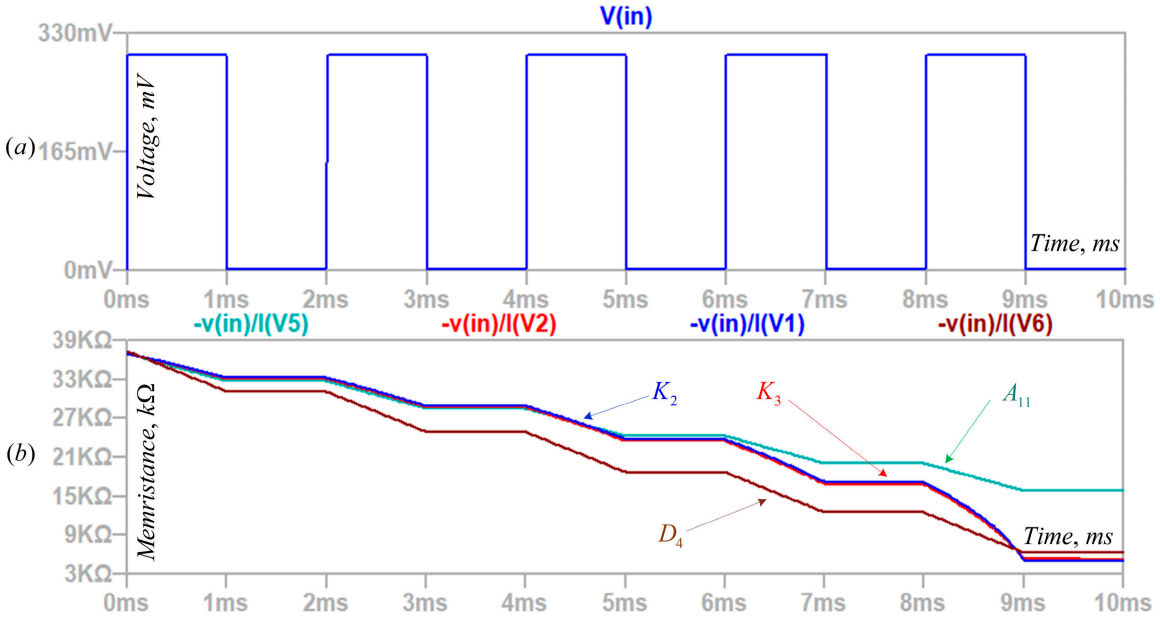


Figure 5. Tuning the memristance of the standard models K_2 and K_3, and the modified models A_{11} and D_4. (**a**) Time diagram of the applied voltage; (**b**) time diagram of the respective memristances.

In the following section, a memristor-based neural network and its analysis are presented.

3. A Memristor-Based Neuron, a Neural Network and Their Analysis

3.1. Memristor-Based Artificial Neurons [12,35]

A block diagram of a simple memristor-based neuron [12,35] is given in Figure 6.

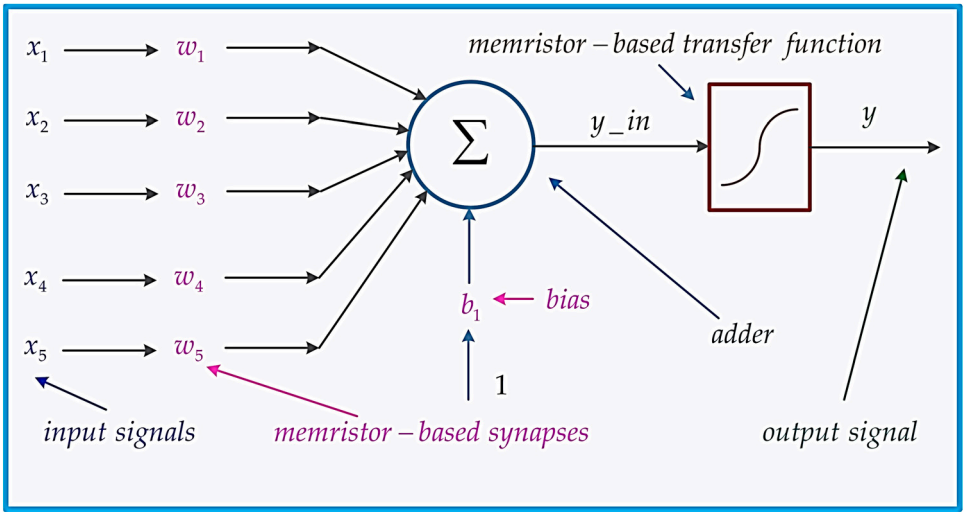


Figure 6. A block schematic of the memristor-based artificial neuron under analysis.

The input signals are denoted as x_1, x_2, x_3, x_4 and x_5. The considered neural network has five inputs and five synaptic weights. The weights are defined as w_1, w_2, w_3, w_4 and w_5. An additional synapse denoted as b_1 is used for the bias signal. The synapses are based on memristors and operational amplifiers. The adder is based on operational

amplifiers and reference memristors with a memristance of 10 kΩ. The output signal of the adder is directed to a memristor-based transfer function. It is based on MOS transistors and memristors.

Figure 7 presents in detail the principal schematic of the synaptic bonds and the summator of a memristor-based neuron. The circuit ensures implementation of separate positive and negative synaptic weights. In the provided example the negative synaptic weights are implemented by the memristors M_1, M_2 and M_3. The memristors M_4 and M_5 are related to positive synaptic weights. The memristor M_6 ensures biasing of the weighted sum of the input signals. The memristors M_7, M_8 and M_9 are reference elements with a resistance of 10 kΩ. The memristor M_8 is used as a feedback element as well. The memristors are selected using the crossbar on the left. The output signal *y_in* is a weighted sum of the inputs and can be expressed as follows—Formula (9) [35]:

$$y_in = v_{out_OA1} = -\frac{M_7}{M_1}x_1 - \frac{M_7}{M_2}x_2 - \frac{M_7}{M_3}x_3 - \frac{M_7}{M_8}\left(-\frac{M_9}{M_4}x_4 - \frac{M_9}{M_5}x_5 - \frac{M_9}{M_6}x_b\right) \quad (9)$$

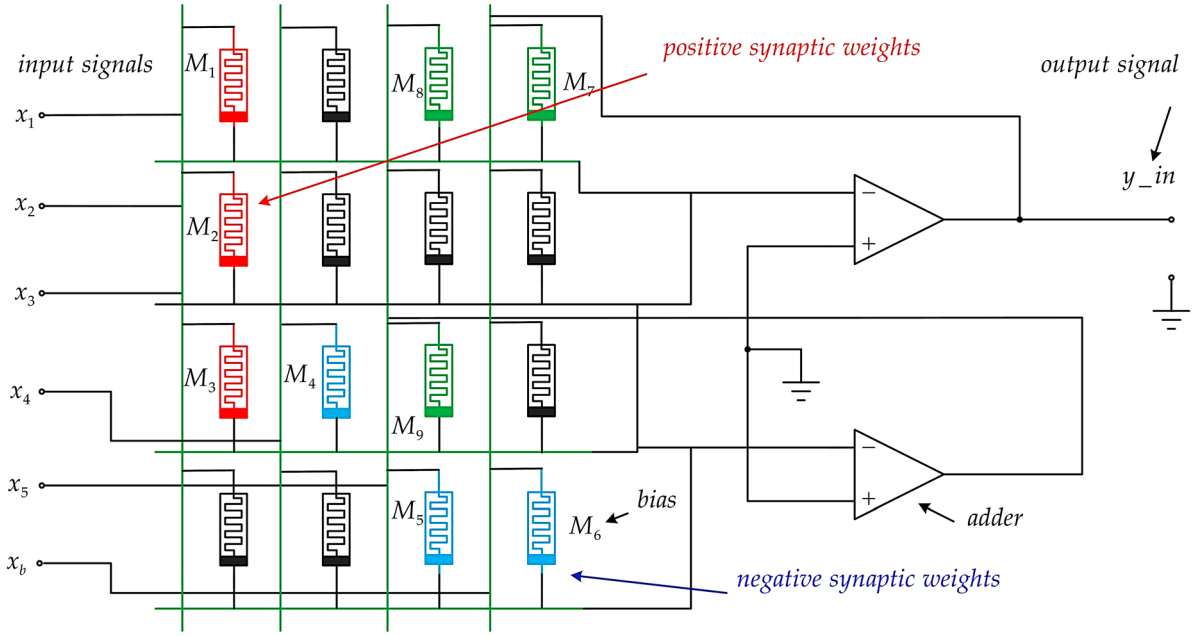


Figure 7. An illustrative diagram of the memristor synaptic elements under study and the corresponding adder structure.

After a little algebraic transformation of (9), Formula (10) is obtained:

$$y_in = v_{out_OA1} = -\frac{M_7}{M_1}x_1 - \frac{M_7}{M_2}x_2 - \frac{M_7}{M_3}x_3 + \frac{M_7}{M_4}x_4 + \frac{M_7}{M_5}x_5 + \frac{M_7}{M_6}x_6 \quad (10)$$

where the coefficients near the signals x_1–x_b are the respective synaptic weights—Formula (11):

$$w_1 = -\frac{M_7}{M_1};\ w_2 = -\frac{M_7}{M_2};\ w_3 = -\frac{M_7}{M_3};\ w_4 = \frac{M_7}{M_4};\ w_5 = \frac{M_7}{M_5};\ b = \frac{M_7}{M_6} \quad (11)$$

Noticeably, the synaptic weights w_1, w_2, and w_3 are negative ones, and the weights w_4, w_5 and w_6 are positive ones [35].

During the phase of usage of the neuron, the memristors operate as linear resistors. For this purpose, their voltage must be lower than the activation threshold v_{thr}. In the procedure of tuning the synaptic weights, a sequence of positive or negative voltage pulses is used to change the memristance of the respective memristors [32,35–38].

In the next sub-section, analytical determination of the needed impulse length at a given amplitude is presented for both the modified memristor models B_{mod} and A_{11}.

3.2. Analytical Determination of the Pulse Duration and a Comparison with Simulation Results for Both the Modified Memristor Models B_{mod} and A_{11}

For the modified memristor model B_{mod}, for applied voltage, whose absolute value exceeds the memristor activation threshold v_{thr}, the state differential equation from System (7) is transformed in a simpler form—Equation (12) [29]:

$$\frac{dx}{dt} = k_1 v^m \sin^2(\pi x) \tag{12}$$

After a separation of the state variable x and the time variable t, the following formula is derived and integrated in limits, where x is changed between x_1 and x_2, and the time variable from t_1 to t_2—Formula (13):

$$\frac{dx}{\sin^2(\pi x)} = k_1 v^m dt \Rightarrow \int_{x_1}^{x_2} \frac{1}{\sin^2(\pi x)} dx = k_1 v^m \int_{t_1}^{t_2} dt = k_1 v^m T_{imp} \tag{13}$$

where the duration of the tuning pulse could be obtained in the following form—Formula (14):

$$T_{impB_{\mathrm{mod}}} = \frac{-\frac{\cot g(\pi x_2)}{\pi} + \frac{\cot g(\pi x_1)}{\pi}}{k_1 v^m} \tag{14}$$

For additional confirmation of the correctness of the obtained length of the pulse, a simulation in LTSPICE simulator is conducted, for m = 3, k_1 = 10,000, v = 0.6 V, x_1 = 0.1 and x_2 = 0.916. The time diagrams of the memristor voltage and the state variable are presented in Figure 8. The length of the pulse is:

$$T_{impB_{\mathrm{mod}}} = \frac{-\frac{\cot g(\pi x_2)}{\pi} + \frac{\cot g(\pi x_1)}{\pi}}{k_1 v^m} = \frac{-\frac{\cot g(\pi \cdot 0.916)}{\pi} + \frac{\cot g(\pi \cdot 0.1)}{\pi}}{10000 \cdot 0.6^3} = 0.999\ ms$$

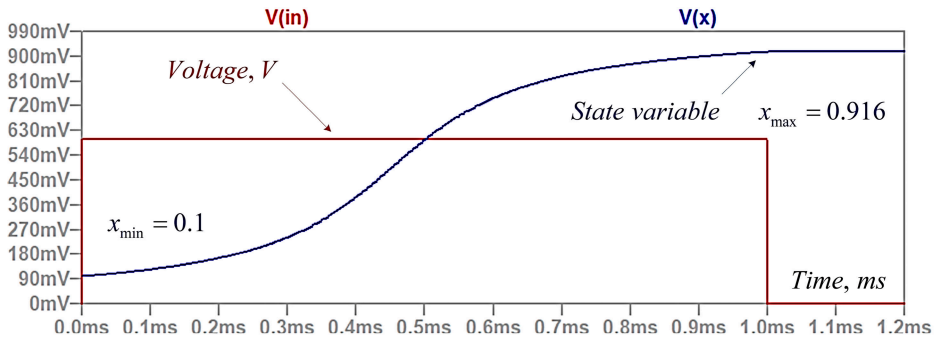


Figure 8. Time diagram of the memristor pulse voltage and the related change in the memristor state variable, confirming the proper tuning of model B_{mod}, according to Formula (14).

The obtained value of the needed pulse duration at a level of 0.6 V is very near to that in LTSPICE—1 ms—and confirm the correctness of the results.

For the next modified memristor model A_{11}, the state differential equation from System (6) is represented with Equation (15) [28]:

$$\frac{dx}{dt} = k_1 \sinh(k_2 v)\left[1 - (x - stp(-i))^{2p}\right] \tag{15}$$

For positive voltage and current, and for $p = 1$, the following expression is valid:

$$i > 0 \Rightarrow -i < 0 \Rightarrow stp(-i) = 0 \tag{16}$$

In this case, the state-variable differential equation defining the memristor model A_{11} could be simplified and represented in the following way:

$$\frac{dx}{dt} = k_1 \sinh(k_2 v)\left(1 - x^2\right) \tag{17}$$

This follows a separation of the state variable x and the time variable t, and an integration process, using the limits of the state variable—x_1 and x_2—and the time limits from t_1 to t_2:

$$\int_{x_1}^{x_2} \frac{1}{1 - x^2} dx = k_1 \sinh(k_2 v) \int_{t_1}^{t_2} dt = k_1 \sinh(k_2 v) T_{imp} \tag{18}$$

After analytical integration of this equation, the following expression for the needed pulse duration is obtained—Formula (19):

$$T_{impA_{11}} = \frac{1}{2} \frac{1}{k_1 \sinh(k_2 v)} \ln\left[\frac{(x_2 + 1)(1 - x_1)}{(1 - x_2)(x_1 + 1)}\right] \tag{19}$$

The numerical value of the pulse duration is:

$$T_{impA_{11}} = \frac{1}{2} \frac{1}{568.8 \sinh(1.966 \cdot 0.6)} \ln\left[\frac{(0.916 + 1)(1 - 0.1)}{(1 - 0.916)(0.1 + 1)}\right] = 0.9991 ms$$

For extra approval of the precision of the derived length of the impulse, LTSPICE simulation is conducted, for $k_1 = 568.8$, $k_2 = 1.966$, $p = 1$, $v = 0.6$ V, $x_1 = 0.1$ and $x_2 = 0.916$. The length of the pulse and the tuning process are presented in the time diagram in Figure 9.

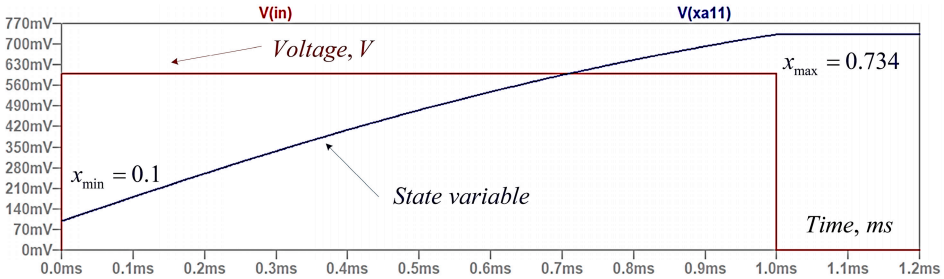


Figure 9. Time diagram of the memristor pulse voltage and the change in the state variable, confirming the proper tuning of memristor model A_{11}, according to Formula (19).

The derived value of the impulse duration at a voltage level of 0.6 V is very close to that in the LTSPICE simulator and confirm the accuracy of the obtained results.

3.3. Memristor-Based Transfer Function with MOS Transistors [35]

Memristor-based transfer functions are important modules in many kinds of artificial neural networks [39,40]. The hardware implementation of the modified logarithmic–sigmoidal transfer function under analysis is shown in Figure 10 for visualization and for further discussion. The activation function is realized using two MOS transistors, denoted as T_1 and T_2, connected in anti-parallel, a memristor element M, and a DC voltage source [35]. The MOS transistors are n-type devices selected from the standard LTSPICE library. Each transistor is configured as a diode by directly shorting its gate and source terminals. Their anti-parallel connection produces a nonlinear and symmetric

activation characteristic. The memristor M is placed in series with the transistors, and the voltage drop across it corresponds to the difference between the input and output signals. The DC voltage source V_1 provides a shift of the output voltage level [35]. The resulting input–output relationship closely resembles a logarithmic–sigmoidal transfer function of the form: $y = 1/(1 + exp(-k \times y_in)$. Additional voltage divider implemented with the resistors R_1 and R_2 is used for ensuring a signal lower than the memristor activation threshold for the neurons after the transfer function. The output operational amplifier is a voltage repeater and acts as a buffer stage.

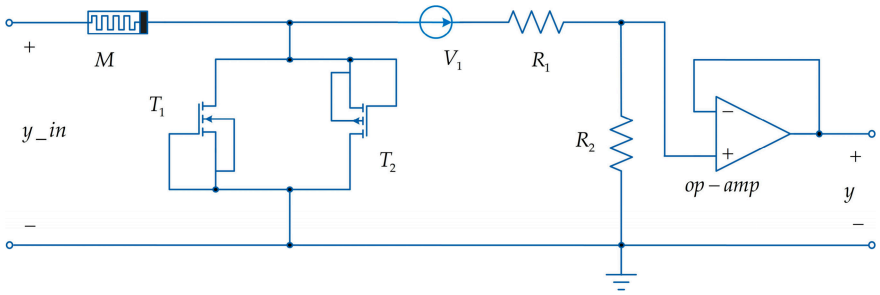


Figure 10. A principal schematic of the memristor-based transfer function under analysis.

The LTSPICE implementation of the discussed memristor-based transfer function is presented in Figure 11a. To obtain the optimal value of the parameter k, several simulations and comparisons are made, using the controlled voltage source B_1. Its voltage presents a modified logarithmic–sigmoidal activation function, presented in Figure 11b. It is established that for $k = 5$ and memristance $m_0 = 200$ Ohms, the best closeness between the discussed transfer functions is obtained. The derived expression for the voltage of the controlled voltage source B_1 is: $v_{out} = 1/(1 + exp(-5 \times v_{in}))$.

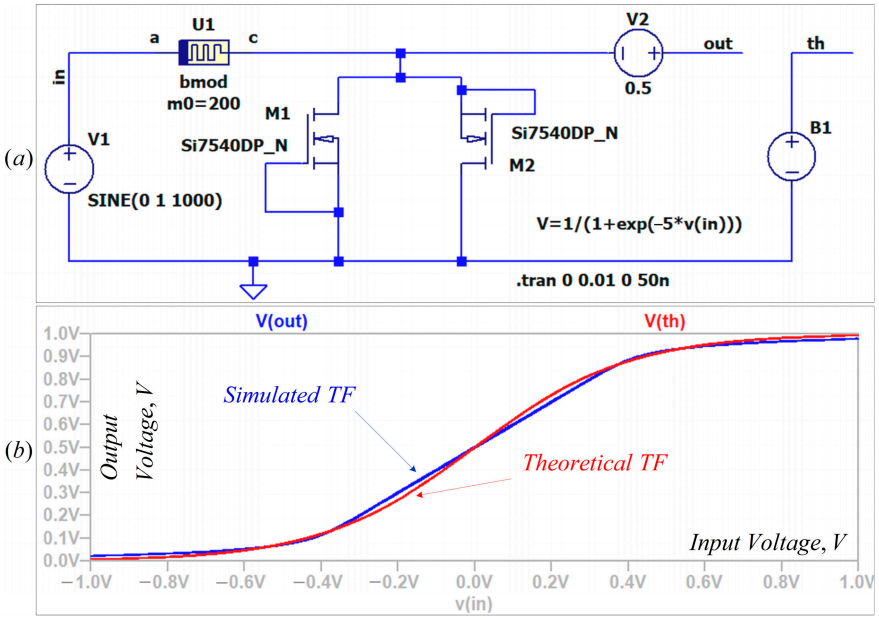


Figure 11. (**a**) A modified tangent–sigmoidal transfer function implemented in LTSPICE simulator, based on memristors and MOS transistors; (**b**) a comparison of the obtained activation function, presented by a voltage-controlled source with a voltage signal $v_{out} = 1/(1 + exp(-5 \times v_{in}))$.

3.4. A Memristor-Based Neural Network

In recent years, memristor-based artificial neural networks have been under intensive analyses [41–43]. A block scheme of the considered memristor-based neural net under investigation is shown in Figure 12 for its visualization and for further discussion on its

structure and specific operation. It contains five input nodes for the input signals x_1, x_2, x_3, x_4 and x_5. The neural net's hidden layer has four memristor-based neurons—N_1, N_2, N_3 and N_4. The bias signals are expressed as b_1, b_2, b_3 and b_4. The synaptic weights are presented by w_{ij}, where i = 1–4 is the number of the corresponding artificial neurons, and j is the number of the input voltage signal. The adders are applied to sum the weighted input voltage signals. The outputs of the adders are expressed as s_1, s_2, s_3 and s_4. They are directed to the logarithmic–sigmoidal transfer functions tf_1, tf_2, tf_3 and tf_4. The signals tf_1, tf_2, tf_3 and tf_4 are directed to the output layer of the memristor neural net, which has two neurons—N_{out1} and N_{out2}. The weights of the synapses are denoted by v_{kp}, where k is the number of the output neuron, and p is the number of the corresponding input signal for the neuron. The biasing signals directed to the adders are represented as b_{out1} and b_{out2}. The output signals of the adders are denoted as s_{1out} and s_{2out}. The corresponding activation functions tf_{out1} and tf_{out2} are of type "purelin" [35]. They are implemented with buffering operational amplifiers. The output voltage signals of the memristor neural net are expressed as y_1 and y_2.

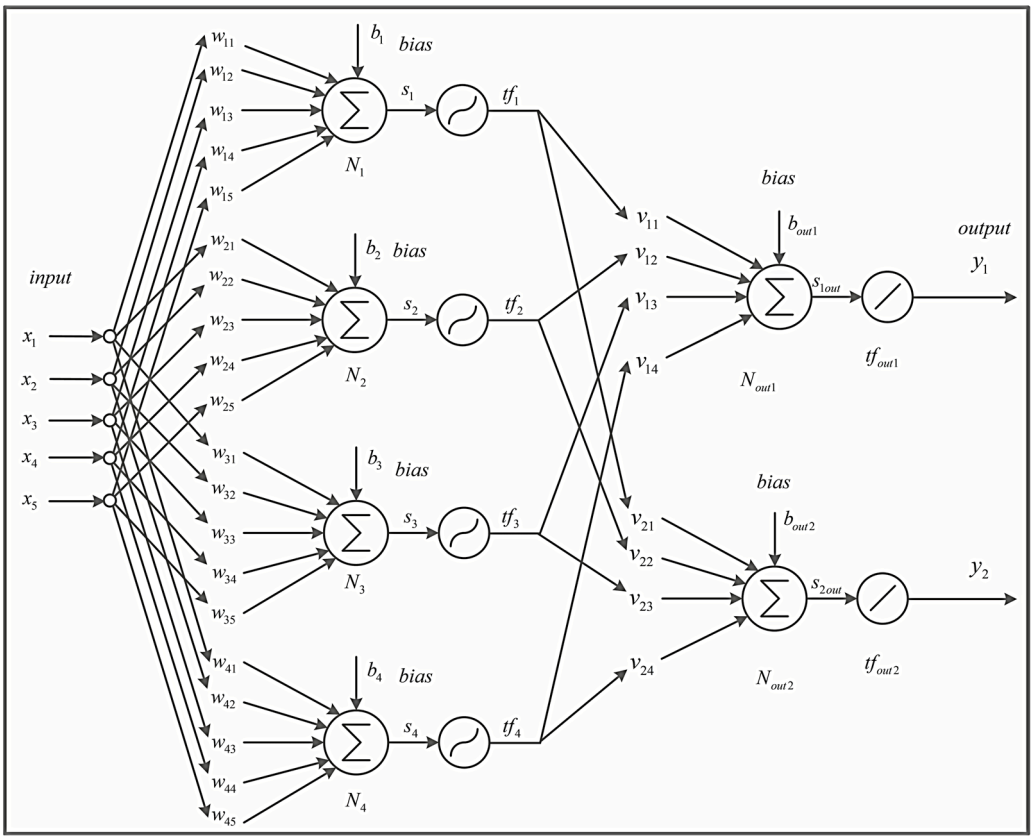


Figure 12. A structural diagram of the considered memristor-based neural network with five inputs, a hidden layer of four neurons, and an output layer of two neurons.

The operation of the discussed neural network that is memristor-based is founded on its training in a MATLAB simulator, using a feed-forward and error backpropagation algorithm to tune the weights of the synapses [11]. The derived values of the synaptic weights are utilized for adjustment of the memristances and for implementation of the neural net in the LTSPICE simulator. The input voltage signals are sequences of rectangular impulses. The duration of an impulse is 800 μs. The corresponding duty cycle of the pulse sequence is 50%.

The levels of the input voltage signals and the bias signal are presented in Table 1 for four different cases. The bias signal is chosen with a level of 100 mV, and it does not go beyond the memristor activation threshold v_{thr}.

Table 1. Input and bias signals for the considered memristor neural network, implemented in LTSPICE simulator.

Case	Input Signals, mV					Bias Signal, mV
	v_{in1}	v_{in2}	v_{in3}	v_{in4}	v_{in5}	v_{bias}
1	−10	−30	−20	10	20	100
2	10	−20	−10	20	10	100
3	−30	−10	−20	30	20	100
4	30	−40	20	−10	30	100

Table 2 presents the weights of the memristive synapses of the hidden layer of the memristor-based neural network. The rows are related to the neurons, and the columns represent the respective synapses and their weights.

Table 2. Weights for the synapses in the hidden layer of the memristor neural net in LTSPICE.

	Synaptic Weights for the Hidden Layer of the Neural Network					
Neuron	w_1	w_2	w_3	w_4	w_5	w_{bias}
N_1	−0.25	−0.2	−0.5	1	0.25	0.2
N_2	−1	−0.5	−2	0.5	1	0.25
N_3	−0.5	−0.25	−1	1	2	0.25
N_4	−0.2	−1	−2	0.25	0.5	1

Table 3 presents the resistances of the memristors, which are responsible for realization of the synaptic weights.

Table 3. Memristances related to the synapses in the neural network's hidden layer in LTSPICE.

	Input Memristances, kΩ					
Neuron	M_1	M_2	M_3	M_4	M_5	M_{bias}
N_1	40	50	20	10	40	50
N_2	10	20	5	20	10	40
N_3	20	40	10	10	5	40
N_4	50	10	5	40	20	10

The weights of the synapses between the hidden layer and the output layer are presented in Table 4.

Table 4. Synaptic weights for the output layer of the memristive neural net.

	Synaptic Weights for the Output Layer of the Memristive Neural Net				
Neuron	w_{1out}	w_{2out}	w_{3out}	w_{4out}	$w_{biasout}$
N_{1out}	−1	−0.2	1	0.25	1
N_{2out}	−0.5	−1	2	0.5	0.2

The memristances of the corresponding memristors that ensure the synaptic bonds between the hidden and output layers are presented in Table 5.

Table 5. Memristances for the synapses for the output layer of the memristor neural net in LTSPICE.

	Memristances for the Synapses for the Output Layer of the Memristor Neural Net, in kΩ				
Neuron	**M_{1out}**	**M_{2out}**	**M_{3out}**	**M_{4out}**	**$M_{biasout}$**
N_{1out}	10	50	10	40	10
N_{2out}	20	10	5	20	50

A Simulink schematic of the considered memristor-based neural network is presented in Figure 13. It corresponds to the neural network presented in Figure 12. The Simulink model of the neural network is proposed and analyzed for additional confirmation of the proper operation of the considered memristor-based neural network. The input signals are implemented with source modules, related to Microsoft Excel worksheets, containing the time and voltage variables for the inputs. The synapses are represented with standard gain elements. The adders are realized with the standard building blocks "sum" in Simulink environment. The corresponding transfer function is implemented, using a modified logarithmic–sigmoidal MATLAB function.

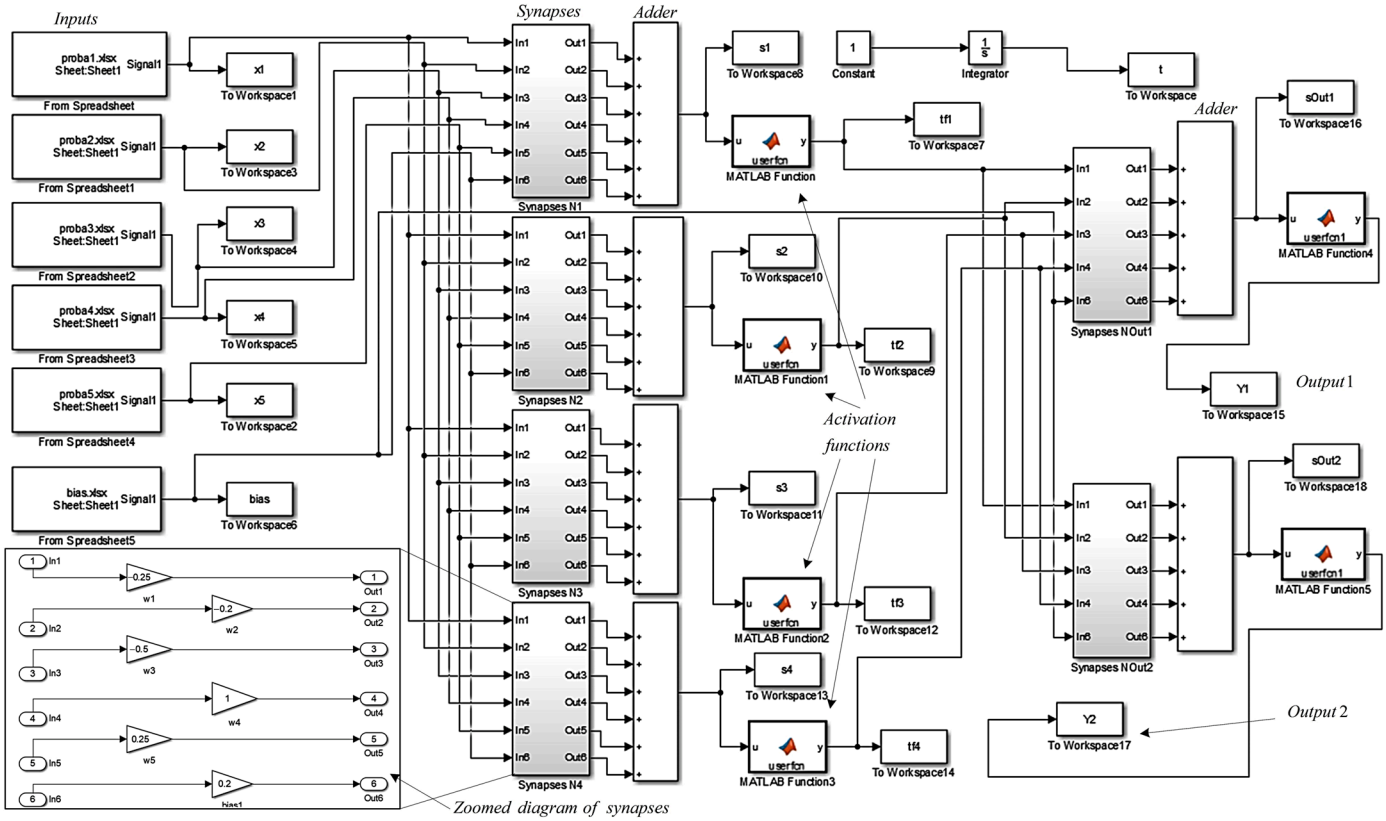


Figure 13. Simulink scheme of the considered memristor-based neural network.

The time-domain diagrams of the input signals, according to Table 1, are presented in Figure 14. The maximal simulation time is 5600 μs. The time step is 10 ns.

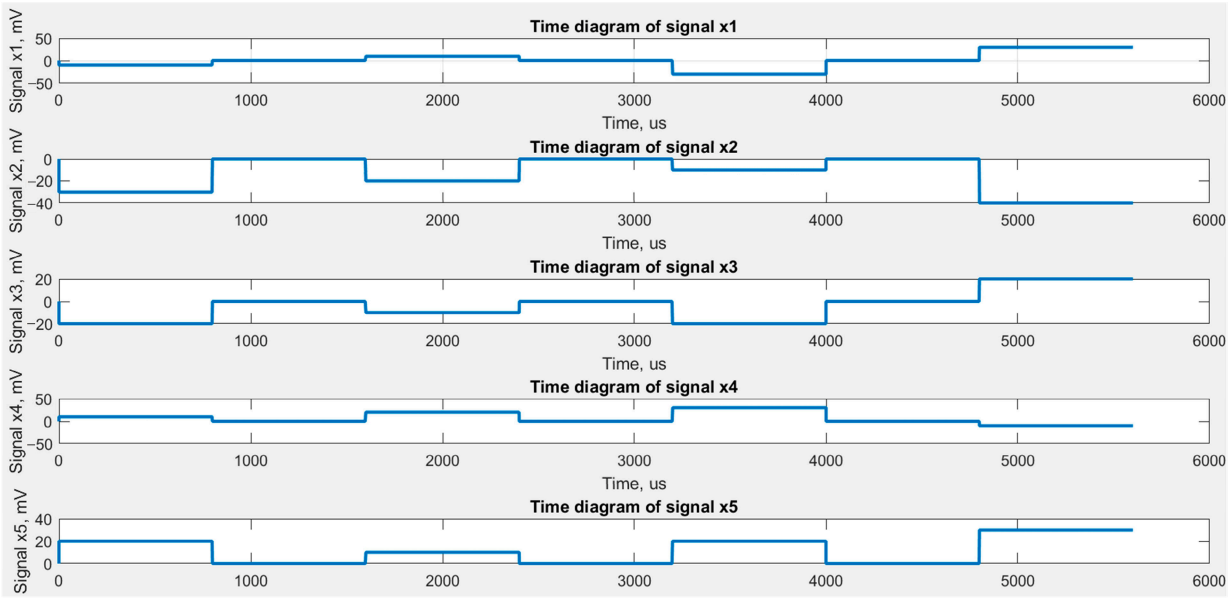


Figure 14. Time diagrams of the input signals of the considered memristor-based neural network.

The time diagrams of the signals after the adders—s_1, s_2, s_3, s_4—and the signals after the transfer functions—tf_1, tf_2, tf_3, tf_4—are presented in Figure 15.

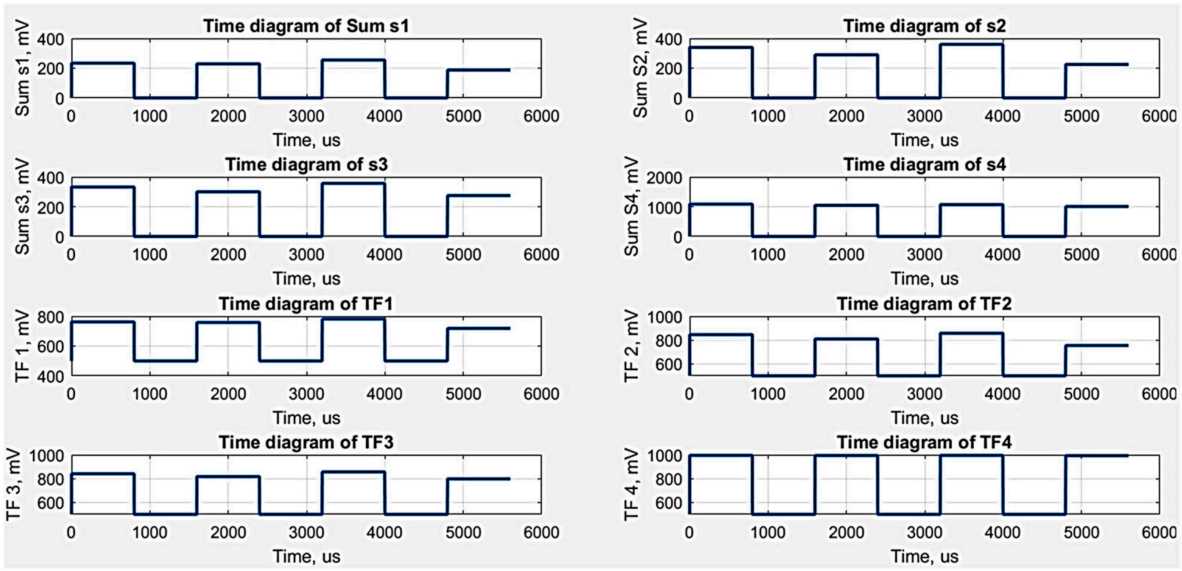


Figure 15. Time diagrams of the signals after the adders—s_1, s_2, s_3, s_4—and after the transfer functions—tf_1, tf_2, tf_3, tf_4.

The output signals after the adders—$SumOut_1$, $SumOut_2$—and after the output transfer functions—Y_1, Y_2—are visualized in Figure 16.

In Table 6, a comparison is proposed between the results derived for Case 1 in MATLAB, Simulink and LTSPICE simulators. A good similarity between the derived results is obtained. The maximal discrepancy between the output signals corresponds to a relative error of about 3%, which is an acceptable one for the analyses considered. The results related to the other three cases of signal levels are presented for comparison and further comments in Tables 7–9, described in Table 1, respectively.

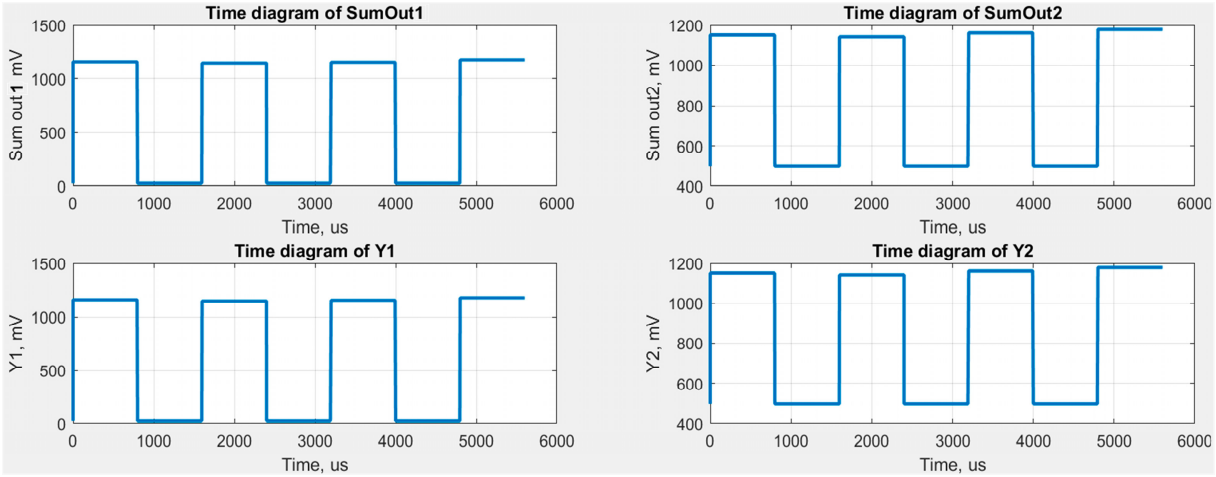


Figure 16. Time diagrams of the output signals of the memristor neural network: after the output adders—$SumOut_1$, $SumOut_2$—and after the output transfer functions Y_1, Y_2.

Table 6. A comparison between the results, obtained after analyses of the memristor neural net for Case 1.

Signals	**S_1**	**S_2**	**S_3**	**S_4**	**tf_1**	**tf_2**	**tf_3**	**tf_4**	**Y_1**	**Y_2**
Unit	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**
Simulink	0.2335	0.3400	0.3325	1.0845	0.7627	0.8455	0.8406	0.9956	1.1577	1.1521
LTSPICE	0.2335	0.3400	0.3325	1.0845	0.7645	0.8471	0.8402	0.9941	1.1571	1.1532
MATLAB	0.2335	0.3400	0.3325	1.0845	0.7627	0.8455	0.8406	0.9956	1.1577	1.1521

Table 7. A Comparison of the results, derived after the analyses of the memristor neural net for Case 2.

Signals	**S_1**	**S_2**	**S_3**	**S_4**	**tf_1**	**tf_2**	**tf_3**	**tf_4**	**Y_1**	**Y_2**
Unit	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**
Simulink	0.2290	0.2900	0.3000	1.0480	0.7586	0.810	0.8176	0.9947	1.1457	1.1432
LTSPICE	0.2290	0.2901	0.3000	1.0480	0.7602	0.8118	0.8171	0.9940	1.1455	1.1435
MATLAB	0.2290	0.2900	0.3000	1.0480	0.7586	0.810	0.8176	0.9947	1.1457	1.1432

Table 8. A comparison between the results, obtained after analyses of the memristor neural net for Case 3.

Signals	**S_1**	**S_2**	**S_3**	**S_4**	**tf_1**	**tf_2**	**tf_3**	**tf_4**	**Y_1**	**Y_2**
Unit	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**
Simulink	0.2545	0.3600	0.3575	1.0735	0.7812	0.8581	0.8566	0.9954	1.1527	1.1622
LTSPICE	0.2545	0.3600	0.3575	1.0735	0.7815	0.8592	60.8578	0.9951	1.1521	1.1626
MATLAB	0.2545	0.3600	0.3576	1.0735	0.7812	0.8581	0.8566	0.9954	1.1527	1.1622

Table 9. A comparison of the results, obtained after the analyses of the memristor neural net for Case 4.

Signals	S_1	S_2	S_3	S_4	tf_1	tf_2	tf_3	tf_4	Y_1	Y_2
Unit	V	V	V	V	V	V	V	V	V	V
Simulink	0.1880	0.2250	0.2750	1.0065	0.7191	0.7549	0.7982	0.9935	1.1765	1.1787
LTSPICE	0.1880	0.2250	0.2750	1.0065	0.7203	0.7558	0.7977	0.9939	1.1772	1.1783
MATLAB	0.1880	0.2250	0.2750	1.0065	0.7191	0.7549	0.7982	0.9935	1.1765	1.1787

In the next section, several memristor-based logic gates and circuits are discussed and analyzed, using the standard and modified models under analysis.

4. Memristor-Based Logic Gates and Circuits

Memristor-based logic gates and circuits are significant modules of more complex logic devices and digital schemes for computing [44–47]. In the next sub-section, some simple memristor-based logic gates will be considered, based on the commented above standard and modified memristor models.

4.1. Memristor-Based Logic Gate OR

A memristor-based logic gate OR is presented in Figure 17a. It contains two voltage sources—V_1 and V_2. They generate sequences of rectangular voltage pulses with a maximal level of 1 V and a minimal level of zero, corresponding to the logical levels of one and zero. The input signals are denoted as in_1 and in_2. An additional resistor R_1 with a high resistance is used and connected to the output. The output signal is denoted as out_{D4}. Memristors are modeled with the standard models of Joglekar (K_2), Biolek (K_3), Knowm, and the modified model D_4.

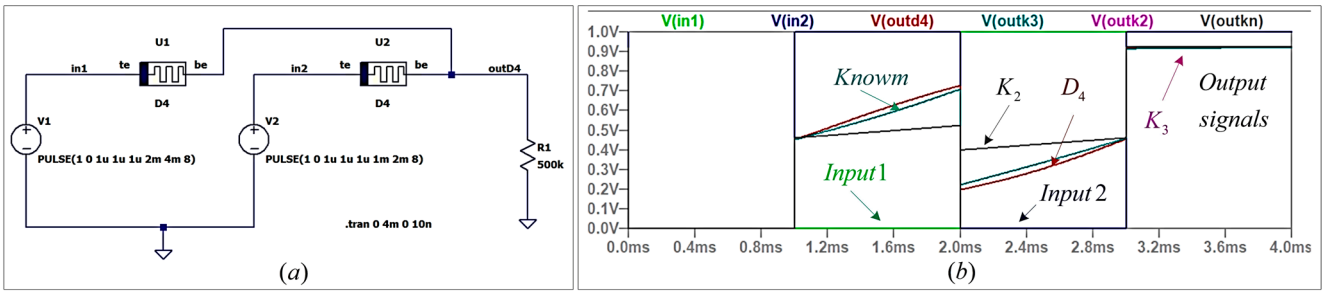


Figure 17. (**a**) A schematic of a two-input memristor-based "OR" gate in LTSPICE simulator. (**b**) Time diagrams of the input and output signals for the considered OR gate, according to the applied memristor models under analysis—Joglekar (K_2), Biolek (K_3), Knowm and D_4.

The time diagrams of the input signals are presented in Figure 17b. In the same sub-figure, the time diagrams of the output signal according to the applied memristor models are shown. In this case, the considered memristor models operate in a hard-switching mode. The switching properties are important for the quick change in the memristance at the applied voltage pulses. In this sense, the Knowm model and D_4 model are preferable, compared to Joglekar and Biolek memristor models. For analysis and simulations of large memristor-based circuits and devices, simple and fast-operating memristor models are needed. In this case, the modified model D_4 could be utilized for analysis of such electronic circuits. The standard Knowm model has better accuracy, when fitting experimental *i-v* relationships of Knowm self-directed channel memristors, but it has higher complexity than

model D_4, due to the use of many mathematical operations, and it is not very appropriate for analysis of complex circuits.

4.2. Memristor-Based Logic Gate AND

A schematic of a memristor-based logic gate AND is presented in Figure 18a. It has two inputs, denoted by in_1 and in_2. An additional DC voltage source V_2 is applied. To obtain a logical one at the output, both the memristors must be in the OFF-state, and this mode corresponds to applied logical ones to the inputs. In the other three cases, when at least one of the input signals is zero, one of the memristors is in the ON-state and the output voltage is near zero. In this logic circuit, the memristors operate in a hard-switching mode. In Figure 18b, the time diagrams of the input and output signals are presented for their visualization and comparison. In this case, the memristor models under analysis have similar behavior.

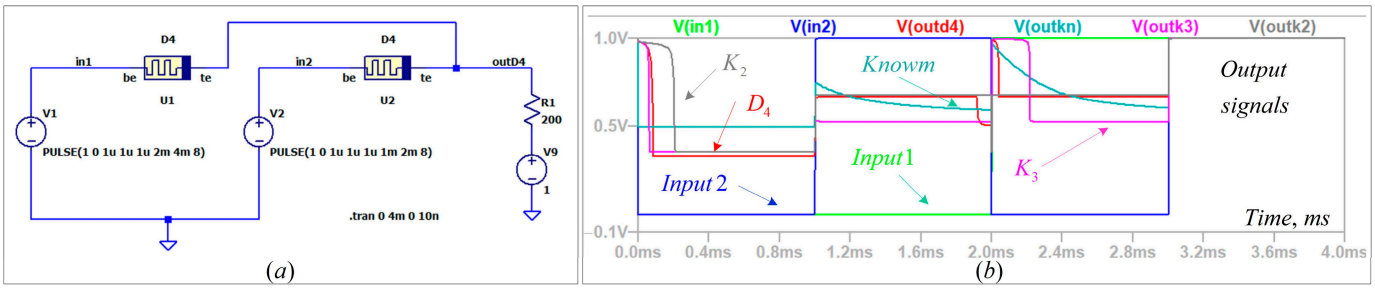


Figure 18. (**a**) A schematic of a two-input memristor-based "AND" gate in LTSPICE environment. (**b**) Time-domain diagrams of the input and output signals of the AND gate.

4.3. Memristor-Based Logic Gate XOR [32,33]

In Figure 19, a schematic of a memristor-based XOR circuit is presented. It contains two input signals, generated by the voltage sources V_1 and V_2, two invertors, implemented by the elements A_1 and A_2, two dependent voltage sources B_1 and B_2, four memristors and two N-channel MOS transistors. The two MOS transistors M_1 and M_2 create a differential amplifier [32,33]. The voltage source V_3 is used for a DC power supply of the MOS transistors.

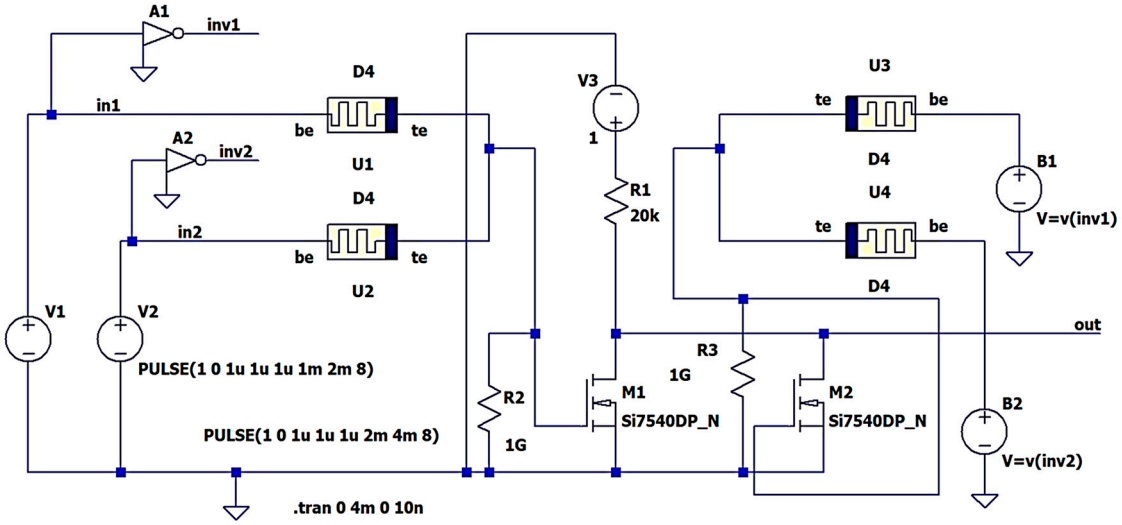


Figure 19. A schematic of XOR logic gate with memristors and MOS transistors in LTSPICE.

The time diagrams of the input and the output signals are presented in Figure 20.

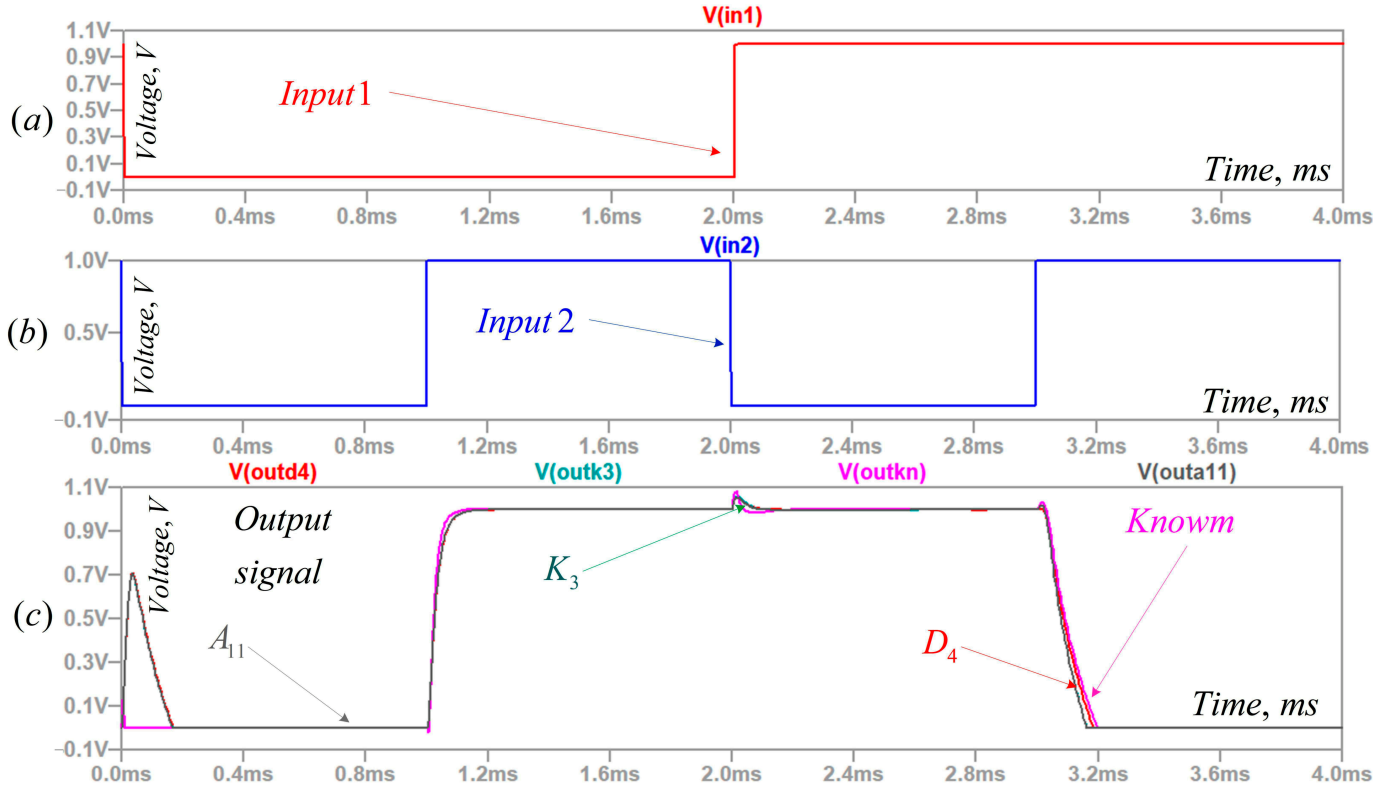


Figure 20. Time diagrams of the signals for the XOR gate: (**a**) input signal 1; (**b**) input signal 2; (**c**) output signals, according to the used memristor models K_3, Knowm, A_{11} and D_4.

The input signals for the XOR memristor logic gate are presented in Figure 20a,b. The corresponding output signal is presented in Figure 20c.

In this case, the memristors operate in a hard-switching mode. The memristor models under analysis are with similar behavior in the considered case.

4.4. Memristor-Based Full Adder

Figure 21 presents a schematic of a memristor-based full adder [32]. It contains six memristors, three voltage sources, and three MOS transistors. In Figure 22 the time diagrams of the corresponding signals are presented.

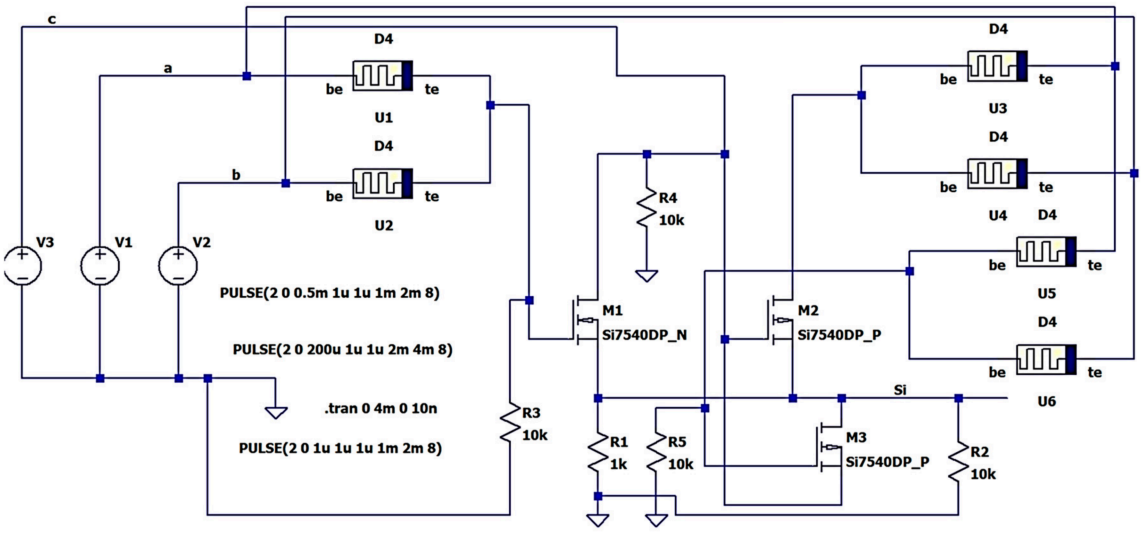


Figure 21. A schematic of a memristor-based adder with MOS transistors.

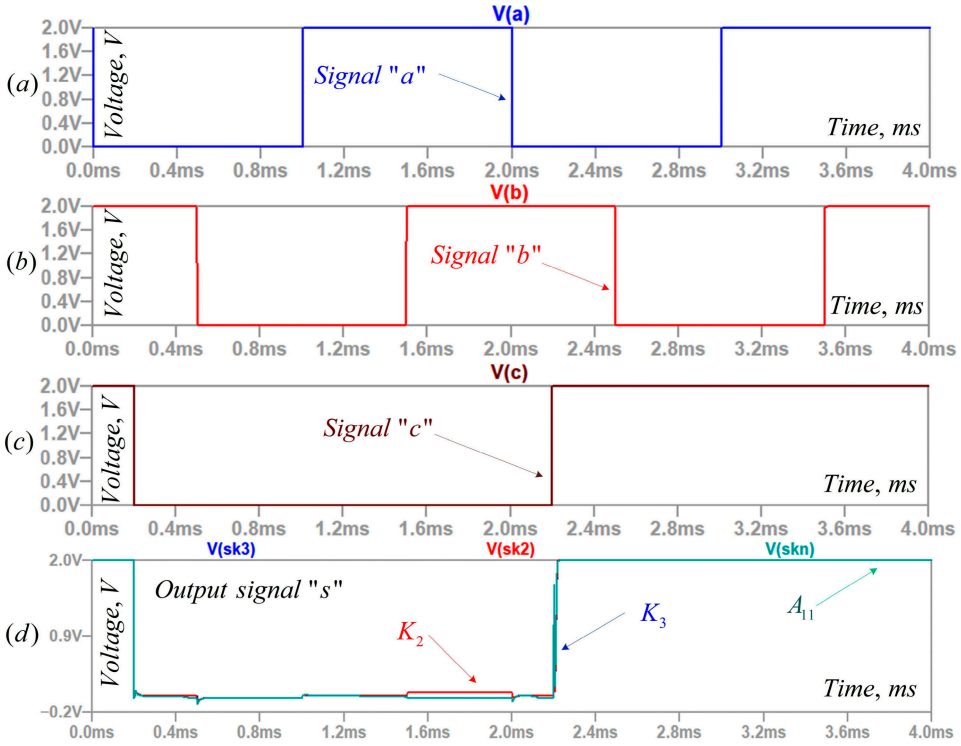


Figure 22. Time diagrams of the signals for the memristor-based adder; (**a**) the input signal $V(a)$; (**b**) the input signal $V(b)$; (**c**) the input signal $V(c)$; (**d**) the output signal $V(sk_2)$.

Figure 22a presents the voltage signal, denoted by "*a*". Figure 22b,c present the time diagrams of the signals "*b*" and "*c*". The output signal "*s*" is presented in Figure 22d. Here, the memristor models applied in the circuit have similar behavior. The memristors operate in a hard-switching mode.

5. A Comparison of the Applied Memristor Models

A comparison of the considered modified and standard memristor models is conducted, according to several important criteria—complexity, accuracy, switching features, simulation time and operating frequency [21,26]. The results are presented in Table 10, highlighting the main advantages of the modified models B_{mod}, A_{11} and D_4, compared to the considered standard models of Joglekar, Biolek and Knowm. The modified memristor models and the proposed one have lower complexity, high operating frequencies, good accuracy and switching properties, and they are more appropriate for analysis of complex memristor-based electronic devices and circuits.

Table 10. A short comparison of the considered standard memristor models—Joglekar, Biolek, Knowm—and the applied modified models D_4 [30], A_{11} [28] and B_{mod} [29].

Memristor Model	**Joglekar [22]**	**Biolek [23]**	**Knowm [24]**	**D_4 [30]**	**A_{11} [28]**	**B_{mod} [29]**
Switching properties	middle	satisfactory	good	good	good	Good
Frequency	low	middle	high	high	high	high
RMS error	2.68×10^{-5}	2.56×10^{-5}	1.31×10^{-5}	2.72×10^{-5}	2.15×10^{-5}	1.44×10^{-5}
Complexity	low	middle	high	low	low	low
Simulation time, ms	262.3	291.4	337.9	149.3	304.1	297.3

The analyses and simulations were conducted on a computer with AMD Ryzen 3 Pro, 2.1 GHz, 16 GB RAM and MS Win 64-bit OS. MATLAB ver. 2016a and LTSPICE ver. XVII were used as specialized software products.

The results from the comparison of the memristor models are presented in Table 10. After a comparison of the main properties and advantages of the modified memristor models, it could be concluded that the modified models A_{11}, B_{mod} and D_4 have lower simulation time, lower complexity, good switching properties and high operating frequencies. The presence of activation thresholds in the modified memristor models A_{11} and B_{mod} is very important for application in neural networks and logic gates for artificial intelligence, where the memristors must be able to operate as linear resistors and as switching elements. In this sense, some of the standard memristor models without activation thresholds are not suitable in such cases.

6. Discussion

After comparing the simulation results of the memristor-based neural network implemented in LTSPICE, MATLAB, and Simulink, an excellent agreement is observed among the signals following the adders, the transfer functions, and the final outputs. The maximum deviation between the output signal levels is below 3%. The network synapses are realized using single memristors together with operational amplifiers configured for summation through their inverting inputs, enabling the implementation of both positive and negative weights. The logarithmic–sigmoidal transfer function employed in the study is constructed using a metal–oxide memristor in combination with two anti-parallel MOS transistors. The analyses demonstrate a strong correspondence between all obtained results, confirming the accurate operation of the proposed memristor-based neural network.

7. Conclusions

The presented modified and standard memristor models are applied in neural networks and logic circuits for their analysis using MATLAB and LTSPICE simulators. The performance of the modified models B_{mod}, A_{11} and D_4 is evaluated by comparing them to Joglekar, Biolek, and Knowm standard memristor models. Several significant criteria are taken into account, such as the operating frequency, switching properties, complexity, accuracy and simulation time. The modified memristor models, together with the Knowm model, show better switching properties compared to the classical memristor models, making them more suitable for application in various complex circuits, owing to their simplified equations and a comparatively good accuracy. Most of the modified memristor models have slightly lower accuracy compared to physics-based memristor models, but they are quite simpler and faster and they are suitable for simulations and analysis of real memristor-based devices and circuits. The present work is a step forward in the field of modeling and analysis of memristor-based neural nets, logic gates and circuits for electronic systems with artificial intelligence.

Author Contributions: Conceptualization, V.M. and G.T.; methodology, V.M., G.T. and S.K.; software, V.M., G.T. and S.K.; validation, V.M. and G.T.; formal analysis, V.M., G.T. and S.K.; investigation, V.M., G.T. and S.K.; resources, V.M., G.T. and S.K.; data curation, G.T., S.K. and V.M.; writing—original draft preparation, V.M., G.T. and S.K.; writing—review and editing, V.M., G.T.; visualization, V.M., G.T. and S.K.; supervision, V.M., G.T. and S.K.; project administration, V.M. and G.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

MOS	Metal–Oxide Semiconductor Transistor
CMOS	Complementary Metal–Oxide Semiconductor Technologies
SPICE	Simulation Program with Integrated Circuits Emphasis
LTSPICE	Linear Technology SPICE software
MATLAB	MATrix LABoratory software
AI	Artificial Intelligence
RMSE	Root Mean Square Error
XOR	eXclusive OR logic function

References

1. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S. The missing memristor found. *Nature* **2008**, *453*, 80–83. [CrossRef] [PubMed]
2. Kumar, S.; Wang, Z.; Davila, N.; Kumari, N.; Norris, K.J.; Huang, X.; Strachan, J.P.; Vine, D.; Kilcoyne, A.L.; Nishi, Y.; et al. Physical origins of current and temperature controlled negative differential resistances in NbO_2. *Nat. Commun.* **2017**, *8*, 658. [CrossRef] [PubMed]
3. Mohammad, B.; Abi Jaoude, M.; Kumar, V.; Al Homouz, D.M.; Nahla, H.A.; Al-Qutayri, M.; Christoforou, N. State of the art of metal oxide memristor devices. *Nanotech. Rev.* **2016**, *5*, 311–329. [CrossRef]
4. Voronkovskii, V.A.; Aliev, V.S.; Gerasimova, A.K.; Islamov, D.R. Influence of HfO_x composition on hafnium oxide-based memristor electrical characteristics. *Mater. Res. Express* **2018**, *5*, 016402. [CrossRef]
5. Strachan, J.P.; Torrezan, A.C.; Miao, F.; Pickett, M.D.; Yang, J.J.; Yi, W.; Medeiros-Ribeiro, G.; Williams, R.S. State Dynamics and Modeling of Tantalum Oxide Memristors. *IEEE Trans. Electron Devices* **2013**, *60*, 2194–2202. [CrossRef]
6. Mladenov, V. A modified tantalum oxide memristor model for neural networks with memristor-based synapses. In *Proceedings of the 2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–4. [CrossRef]
7. Du, J.; Sun, B.; Yang, C.; Cao, Z.; Zhou, G.; Wang, H.; Chen, Y. Ferroelectric memristor and its neuromorphic computing applications. *Mater. Today Phys.* **2025**, *50*, 101607. [CrossRef]
8. Li, J.; Qian, Y.; Li, W.; Yu, S.; Ke, Y.; Qian, H.; Lin, Y.H.; Hou, C.H.; Shyue, J.J.; Zhou, J.; et al. Polymeric memristor based artificial synapses with ultra-wide operating temperature. *Adv. Mater.* **2023**, *35*, 2209728. [CrossRef] [PubMed]
9. Shao, Q.; Wang, Z.; Zhou, Y.; Fukami, S.; Querlioz, D.; Chua, L.O. Spintronic memristors for computing. *npj Spintron.* **2025**, *3*, 16. [CrossRef]
10. Sun, K.; Chen, J.; Yan, X. The future of memristors: Materials engineering and neural networks. *Adv. Funct. Mater.* **2021**, *31*, 2006773. [CrossRef]
11. Aggarwal, C. *Neural Networks and Deep Learning*; Springer International Publishing AG: Berlin/Heidelberg, Germany, 2018; p. 497.
12. Xu, W.; Wang, J.; Yan, X. Advances in memristor-based neural networks. *Front. Nanotechnol.* **2021**, *3*, 645995. [CrossRef]
13. Kvatinsky, S.; Wald, N.; Satat, G.; Kolodny, A.; Weiser, U.C.; Friedman, E.G. MRL—Memristor ratioed logic. In *Proceedings of the 2012 13th International Workshop on Cellular Nanoscale Networks and Their Applications*; IEEE: Piscataway, NJ, USA, 2012; pp. 1–6. [CrossRef]
14. Hoffer, B.; Rana, V.; Menzel, S.; Waser, R.; Kvatinsky, S. Experimental demonstration of memristor-aided logic (MAGIC) using valence change memory (VCM). *IEEE Trans. Electron Devices* **2020**, *67*, 3115–3122. [CrossRef]
15. Wang, X.; Yang, R.; Chen, Q.; Zeng, Z. An improved memristor-CMOS XOR logic gate and a novel full adder. In *Proceedings of the 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), Doha, Qatar, 4–6 February 2017*; IEEE: Piscataway, NJ, USA, 2017; pp. 7–11. [CrossRef]
16. Mladenov, V.; Kirilov, S. A Memristor Model with a Modified Window Function and Activation Thresholds. In *Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 1–5. [CrossRef]
17. Cho, K.; Lee, S.J.; Eshraghian, K. Memristor-CMOS logic and digital computational components. *Microelectron. J.* **2015**, *46*, 214–220. [CrossRef]
18. Thomas, A. Memristor-based neural networks. *J. Phys. D Appl. Phys.* **2013**, *46*, 093001. [CrossRef]

19. Aguirre, F.; Sebastian, A.; Le Gallo, M.; Song, W.; Wang, T.; Yang, J.J.; Lu, W.; Chang, M.F.; Ielmini, D.; Yang, Y.; et al. Hardware implementation of memristor-based artificial neural networks. *Nat. Commun.* **2024**, *15*, 1974. [CrossRef] [PubMed]
20. Yang, Y.; Lee, S.C. *Circuit Systems with MATLAB and PSpice*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
21. Mladenov, V. A Unified and Open LTSPICE Memristor Model Library. *Electronics* **2021**, *10*, 1594. [CrossRef]
22. Joglekar, Y.N.; Wolf, S.J. The elusive memristor: Properties of basic electrical circuits. *Eur. J. Phys.* **2009**, *30*, 661. [CrossRef]
23. Biolek, Z.; Biolek, D.; Biolkova, V. SPICE Model of Memristor with Nonlinear Dopant Drift. *Radioengineering* **2009**, *18*, 210–214.
24. Nugent, M.A.; Molter, T.W. AHaH computing–from metastable switches to attractors to machine learning. *PLoS ONE* **2014**, *9*, e85175. [CrossRef] [PubMed]
25. Bednarz, K.; Garda, B. Measurement and Modeling of Self-Directed Channel (SDC) Memristors: An Extensive Study. *Energies* **2024**, *17*, 5400. [CrossRef]
26. Ascoli, A.; Corinto, F.; Senger, V.; Tetzlaff, R. Memristor Model Comparison. *IEEE Circuits Syst. Mag.* **2013**, *13*, 89–105. [CrossRef]
27. Ascoli, A.; Tetzlaff, R.; Biolek, Z.; Kolka, Z.; Biolkova, V.; Biolek, D. The Art of Finding Accurate Memristor Model Solutions. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2015**, *5*, 133–142. [CrossRef]
28. Mladenov, V.; Kirilov, S. An Improved Model for Metal Oxide-Based Memristors and Application in Memory Crossbars. In *Proceedings of the 2022 18th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–4. [CrossRef]
29. Kirilov, S. A Simple Memristor Model for Memory Devices. In *Proceedings of the 2024 13th International Conference on Modern Circuits and Systems Technologies (MOCAST), Sofia, Bulgaria, 26–28 June 2024*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–4. [CrossRef]
30. Mladenov, V.; Kirilov, S. Memristor Models with Logarithmic-Sigmoidal and Piecewise-Linear Self-Limiting State-Flux Relations. In *Submitted for Presentation and Publishing in 33-Rd International IEEE Conference on Electronics Circuits and Systems (ICECS) 2026, Thessaloniki, Greece, 8–11 November 2026*; IEEE: Piscataway, NJ, USA, 2026.
31. Zhevnenko, D.A.; Meshchaninov, F.P.; Kozhevnikov, V.S.; Shamin, E.S.; Telminov, O.A.; Gornev, E.S. Research and Development of Parameter Extraction Approaches for Memristor Models. *Micromachines* **2021**, *12*, 1220. [CrossRef] [PubMed]
32. Zhang, Y.; Wang, X.; Li, Y.; Friedman, E.G. Memristive Model for Synaptic Circuits. *IEEE Trans. Circuits Syst. II Express Briefs* **2017**, *64*, 767–771. [CrossRef]
33. Zhang, Y.; Wang, X.; Friedman, E.G. Memristor-based circuit design for multilayer neural networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2017**, *65*, 677–686. [CrossRef]
34. Kim, P. Matlab Deep Learning. In *With Machine Learning, Neural Networks and Artificial Intelligence*; APress: New York, NY, USA, 2017.
35. Mladenov, V.; Kirilov, S. A Memristor Neural Network Based on Simple Logarithmic-Sigmoidal Transfer Function with MOS Transistors. *Electronics* **2024**, *13*, 893. [CrossRef]
36. Shooshtari, M.; Kim, S.-Y.; Pahlavan, S.; Rivera-Sierra, G.; Través, M.J.; Serrano-Gotarredona, T.; Bisquert, J.; Linares-Barranco, B. Advancing Logic Circuits with Halide Perovskite Memristors for Next-Generation Digital Systems. *SmartMat* **2025**, *6*, e70032. [CrossRef]
37. Shooshtari, M.; Serrano-Gotarredona, T.; Linares-Barranco, B. Review of Memristors for In-Memory Computing and Spiking Neural Networks. *Adv. Intell. Syst.* **2026**, *8*, e202500806. [CrossRef]
38. Shamsi, J.; Avedillo, M.J.; Linares-Barranco, B.; Serrano-Gotarredona, T. Hardware Implementation of Differential Oscillatory Neural Networks Using VO_2-Based Oscillators and Memristor-Bridge Circuits. *Front. Neurosci.* **2021**, *15*, 674567. [CrossRef] [PubMed]
39. Barraj, I.; Mestiri, H.; Masmoudi, M. Overview of memristor-based design for analog applications. *Micromachines* **2024**, *15*, 505. [CrossRef] [PubMed]
40. Chen, W.; Song, L.; Wang, S.; Zhang, Z.; Wang, G.; Hu, G.; Gao, S. Essential characteristics of memristors for neuromorphic computing. *Adv. Electron. Mater.* **2023**, *9*, 2200833. [CrossRef]
41. Cai, L.; Yu, L.; Yue, W.; Zhu, Y.; Yang, Z.; Li, Y.; Tao, Y.; Yang, Y. Integrated memristor network for physiological signal processing. *Adv. Electron. Mater.* **2023**, *9*, 2300021. [CrossRef]
42. Kim, K.; Lee, J. Stochastic memristor modeling framework based on physics-informed neural networks. *Appl. Sci.* **2024**, *14*, 9484. [CrossRef]
43. Barraj, I. A DTMOS-Based Memristor Emulator Circuit for Low-Power Biomedical Signal Conditioning. *Micromachines* **2026**, *17*, 328. [PubMed]
44. Vourkas, I.; Sirakoulis, G.C. Emerging memristor-based logic circuit design approaches: A review. *IEEE Circuits Syst. Mag.* **2016**, *16*, 15–30. [CrossRef]
45. Talati, N.; Gupta, S.; Mane, P.; Kvatinsky, S. Logic design within memristive memories using memristor-aided loGIC (MAGIC). *IEEE Trans. Nanotechnol.* **2016**, *15*, 635–650.

46. Huang, Y.; Li, S.; Yang, Y.; Chen, C. Progress on memristor-based analog logic operation. *Electronics* **2023**, *12*, 2486. [CrossRef]
47. Su, B.; Cai, J.; Zhang, Y.; Wang, Y.; Wang, S.; Wen, K. A 1T2M memristor-based logic circuit and its applications. *Microelectron. J.* **2023**, *132*, 105674.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.