

Article

A Memristor Neural Network Based on Simple Logarithmic-Sigmoidal Transfer Function with MOS Transistors

Valeri Mladenov  and Stoyan Kirilov 

Faculty of Automatics, Department Fundamentals of Electrical Engineering, Technical University of Sofia, 8 St. Kliment Ohridski Blvd, 1000 Sofia, Bulgaria; s_kirilov@tu-sofia.bg

* Correspondence: valerim@tu-sofia.bg; Tel.: +359-882-27-00-77

Abstract: Memristors are state-of-the-art, nano-sized, two-terminal, passive electronic elements with very good switching and memory characteristics. Owing to their very low power usage and a good compatibility to the existing CMOS ultra-high-density integrated circuits and chips, they are potentially applicable in artificial and spiking neural networks, memory arrays, and many other devices and circuits for artificial intelligence. In this paper, a complete electronic realization of an analog circuit model of the modified neural net with memristor-based synapses and transfer function with memristors and MOS transistors in LTSPICE is offered. Each synaptic weight is realized by only one memristor, providing enormously reduced circuit complexity. The summing and scaling implementation is founded on op-amps and memristors. The logarithmic-sigmoidal activation function is based on a simple scheme with MOS transistors and memristors. The functioning of the suggested memristor-based neural network for pulse input signals is evaluated both analytically in MATLAB-SIMULINK and in the LTSPICE environment. The obtained results are compared one to another and are successfully verified. The realized memristor-based neural network is an important step towards the forthcoming design of complex memristor-based neural networks for artificial intelligence, for implementation in very high-density integrated circuits and chips.

Keywords: memristor neural network; logarithmic-sigmoidal transfer function; artificial intelligence; metal-oxide memristors; memristor modeling; LTSPICE models



Citation: Mladenov, V.; Kirilov, S. A Memristor Neural Network Based on Simple Logarithmic-Sigmoidal Transfer Function with MOS Transistors. *Electronics* **2024**, *13*, 893. <https://doi.org/10.3390/electronics13050893>

Academic Editor: Antonio Di Bartolomeo

Received: 15 January 2024

Revised: 19 February 2024

Accepted: 23 February 2024

Published: 26 February 2024



Copyright: © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial neural networks have inspiration from electro-chemical communication between neural cells in the human brain and other biological neural systems [1,2]. Their applications have revealed power efficiency and promising utilization in very low-energy and mobile electronic schemes and devices [3,4]. Neuromorphic integrated chips have the ability to perform complex computations quickly and effectively, consuming minimal electric power [5–7]. In the past, Metal Oxide Semiconductor (MOS) transistor-based neural nets and optical neural networks were a leading approach for realizing analog neural nets, utilizing light, instead of electrical pulses to conduct neural computing, image and video processing, and other applicable tasks [8–10]. Some of the state-of-the-art improvements in this field are the neural nets based on memristors [11–14], which are generally used in the synaptic bonds for storing the synaptic weights [15–17].

Memristors predicted by Leon Chua [18] are electronic passive components with two electrodes that have the ability to store electric charges passed through their structure [19,20]. By applying electric pulses—voltage or current with definite amplitudes and durations—the resistance of memristors (also known as a *memristance*) could be altered. The memristors could work as tunable electronic resistors [21], [22]. The first material memristor based on titanium dioxide (TiO₂) was created in Hewlett-Packard scientific and research labs by Stanley Williams and his collaborators [21,23,24]. Different metal oxides, such as HfO₂, Ta₂O₅, Nb₂O₅, and many others substances are utilized for the realization of

memristor elements [25,26]. Some valuable properties of memristors are their low energy usage, very good switching and memory properties, high switching speed, nano-sizes, and good compatibility with the present day Complementary MOS (CMOS) integrated chips and circuits [27,28]. Memristors are potentially applicable in memory matrices, reconfigurable analog and digital circuits, neural nets, and many others [29,30]. They are related to storage and computing in a single electronic circuit element [31].

Several different categories of memristor-based synapses, artificial neurons, and neural networks are accessible in the related scientific works [32,33]. Some electronic synapses based on single memristor elements [7] and MOS transistors ensure only positive weights, which is a weakness of such realizations. More complicated schematics employ multiple memristors for each synapse [9], arranged in anti-parallel and bridge circuits. Such configurations enable the realization of positive, zero, and negative synaptic weights [9,12]. An arrangement containing five memristors and electronic switches forms a π -type memristor synapse [13]. Certain synapses within this category utilize operational amplifiers (op-amps) and metal-oxide-semiconductor (MOS) transistors as differential amplifiers [14]. In contrast to the advantage of realization of positive and negative weights, the main drawback lies in the higher number of memristors per single synapse. For synapse realization, usually three or more memristors are used, and several op-amps are included [15]. A possibility to realize each synaptic weight with just a single memristor is available, which could significantly reduce neural circuit complexity [34,35].

For analysis and simulation of memristor-based neurons and neural networks with a large number of memristors, fast-operating and simple SPICE models of memristors and MOS transistors are desirable, for simplification of the analyses and for decreasing the simulation time [36,37]. For this aim, a modified, enhanced, and simplified SPICE model of metal-oxide memristors with activation thresholds is utilized [35]. Along with the software products of the Simulation Program with Integrated Circuits Emphasis (SPICE) family, the LTSPICE simulator used here is a preferred one, owing to its user-friendly interface, free license, absence of artificial restrictions on the maximal number of the considered electronic elements and their connections, and very good convergence characteristics [33]. Owing to its benefits, the LTSPICE product is applied for the electronic simulations in the present work. For comparison and confirmation of the derived results, MATLAB, Simulink, and Neural Network Toolbox [32] are also utilized.

After a comprehensive reference investigation, it was established that complete circuits of memristor-based synaptic schemes, neurons, transfer functions and neural nets, and related programming codes and results, together with comparisons are not represented in the available scientific papers. It is also known that some issues with the realization of negative synaptic weights are reported [6]. This was the main motivation for proposing the implementation of a memristor-based neural network in LTSPICE, as the activation function was realized with a voltage-controlled voltage source in the LTSPICE simulator [34]. Subsequently, a single neuron with its activation function implemented with memristors and MOS transistors is also realized and proposed to the readers [35].

The goal of this paper is the realization of a whole electronic implementation of a memristor neural network in the LTSPICE environment. To achieve this purpose, the following tasks are appointed: the realization of a new electronic circuit based on memristor and MOS transistors, realizing a logarithmic-sigmoidal transfer function; the realization of a simple adder based on memristors and operational amplifiers with a minimal number of elements; the synthesis and analysis of a simple multi-layer neural network, utilizing memristors for the synaptic weights of positive and negative signs, and to employ a minimal number of electronic components in the LTSPICE environment [33]. An artificial neuron with two distinct inputs for the positive and for the negative synaptic weights is implemented by employing memristor elements and operational amplifiers. Single metal-oxide memristors are used to implement the synapses of the considered neural network. For the analyses and simulations, a simple, fast-operating, and accurate LTSPICE memristor model with an activation threshold is applied [35]. In this paper, the opportunity to implement

memristor-based neural network circuits and to make analyses and comparisons between different realizations of neural nets in SPICE simulators is proposed to the readers. Future work on the subject is predicted, considering the synthesis and analysis of more complex multi-layer memristor-based neural networks, as well as their practical implementation and comparison with other proposed realizations.

This paper is organized as follows. Section 2 presents memristors, their modeling, tuning, and a brief comparison between memristor models. The LTSPICE realization and analysis of the applied memristor model is considered in Section 3. The applied activation function based on memristors and MOS transistors is presented in Section 4. The LTSPICE memristor neuron, which is considered simple and fast-operating, is discussed in Section 5. Section 6 presents the offered simple neural network with memristors and MOS transistors. Section 7 presents the derived results in MATLAB, SIMULINK, and LTSPICE and their comparison. The conclusion shown in Section 8 summarizes the results and proposes future work on the topic.

2. Memristors—Modeling, Tuning, Comparison, and the Applied Memristor Model

For improvement and for understanding the main conception of the work, according to the structure, operation, modeling, tuning, and comparison of metal-oxide memristors [36,37], a short overview of the basic aspects of memristors is first presented.

2.1. A Short Depiction of Memristor Elements and Their Operation

Memristors are highly nonlinear passive one-port electronic elements with memory and switching properties. In the scientific reports, many memristors based on different materials, such as metal-oxides, polymeric, ferroelectric, spintronic and others, are described [17–24]. Metal-oxide memristors have a central position in the family of memristors, owing to their stable characteristics and parameters, and very good switching and memorizing properties [24]. A part of the memristor nanostructure is doped by oxygen vacancies, applying electroforming processes [18]. The memory and switching effects in memristors depend on their ability to undergo alterations in their resistance, proportional to their state variable, when voltage or current signals are applied [19]. The state variable of a memristor x indicates the ratio between the lengths of the doped layer and the whole memristor structure. The memristor has two limiting values of its resistance, usually denoted as R_{ON} (the ON-resistance state) and R_{OFF} (the OFF-resistance state).

The behavior of memristors in an electric field is quite different, in comparison to those of the classical electronic passive elements—the *resistor*, *inductor*, and *capacitor*. Owing to its specifics, the modeling of memristors is mandatory for their analysis by electronic simulators and applications in electronic schemes and devices [19,26].

2.2. The Process of Modeling of Memristors [19,24,30]

Each mathematical model of a memristor is built on two key equations [26]. The first one offers the i - v relation, and the second one connects the time derivative of state variable x and the memristor current i (or the memristor voltage v). A large collection of repeatedly used metal-oxide memristor models, enclosing those of Williams-Strukov [18], Joglekar [22] and Biolek [23], is represented by the next general set of Equation (1) [26]:

$$\begin{cases} i = v \cdot M(x)^{-1} \\ \dot{x} = k \cdot f(x) \cdot i \end{cases} \quad (1)$$

where i and v are memristor current and voltage, M is the so-called *memristance* (the *state-dependent resistance* of the memristor), x is the state variable of the memristive element, and k is a constant which is dependent on the main physical parameters of the element—the ionic drift mobility μ , the ON-state resistance, and the whole length D . The expression $f(x)$ is a window function, utilized for the limitation of the state variable in the interval

from 0 to 1 and for introducing the boundary effects in a hard-switching mode [26]. The Strukov–Williams memristor model [18] is described by the next set of Equation (2):

$$\begin{cases} i = v \cdot [R_{ON}x + R_{OFF}(1 - x)]^{-1} \\ \dot{x} = k \cdot f_{SW}(x) \cdot i; \quad f_{SW}(x) = -x(x - 1) \end{cases} \quad (2)$$

where R_{ON} and R_{OFF} are minimal and maximal values of memristance, and $f_{SW}(x)$ is a simple and low-order polynomial window function proposed by Williams and Strukov [18]. This classical model is a very simple one, and owing to the reduced number of the elementary mathematical operations, it has a very high operating speed [30,31]. Owing to the linear dependence between the time derivative of the state variable $\dot{x} = dx/dt$ and the memristor current i , this model could not express the different behavior of the memristors for voltages lower or higher than one volt [21,26]. The window function f_{sw} is related to the so-called *terminal state problems* [20,31]. This memristor model has a comparatively low accuracy in the simulation of complex and non-symmetrical experimental current–voltage characteristics of metal-oxide memristors. The Joglekar memristor model is one of the widely used standard metal-oxide memristor models [22] and it is presented by system (3):

$$\begin{cases} i = v \cdot [R_{ON}x + R_{OFF}(1 - x)]^{-1} \\ \dot{x} = k \cdot f_J(x) \cdot i; \quad f_J(x) = 1 - (2x - 1)^{2p} \end{cases} \quad (3)$$

where $f_J(x)$ is a polynomial window, proposed by Joglekar and Wolf [22]. This traditional model for metal-oxide memristors is a simple one, and it has a comparatively high functioning speed [24,31]. Due to the linear relation between the time derivative of state variable x and current i , this model fails to capture the distinct behavior of memristors under voltages lower or higher than one volt. Terminal state problems are related to the applied window function [22,24]. The polynomial window function that is used is an adjustable one and ensures a different slope, according to parameter p and the memristor state variable x .

The Biolek model for metal-oxide memristors, which is one of the mainly utilized standard memristor models, is presented by the next system (4) [23]:

$$\begin{cases} i = v \cdot [R_{ON}x + R_{OFF}(1 - x)]^{-1} \\ \dot{x} = k \cdot f_B(i, x) \cdot i; \quad f_B(i, x) = 1 - (x - \text{stp}(-i))^{2p}; \quad \text{stp}(i) = \begin{cases} 0, & i < 0 \\ 1, & i \geq 0 \end{cases} \end{cases} \quad (4)$$

where $f_B(x, i)$ is a window function, proposed by Biolek, and p is a positive integer [23]. This classic model is a comparatively simple and fast-operating one [24,30]. The window function correctly represents the boundary effects and is not related to terminal state issues [24]. This classical memristor model has a comparatively good accuracy in the modeling of experimental current–voltage relationships.

The Lehtonen–Laiho memristor model is a classic one, frequently applied for the modeling of metal-oxide memristors, and is expressed by the next set of Equation (5) [21]:

$$\begin{cases} i = [\exp(\gamma v) - 1] \cdot \chi + \beta \cdot x^n \cdot \sinh(\alpha v) \\ \dot{x} = a \cdot f_B(i, x) \cdot v^m \end{cases} \quad (5)$$

where m and n are integer coefficients, and γ , α , β , and χ are coefficients for tuning the memristor model [21]. This model is a relatively complex one and has a high operating rate [30,31]. Due to the highly nonlinear relation between the time derivative of state variable x and the memristor voltage v , this standard model correctly expresses the different behavior of metal-oxide memristors for signals higher or lower than 1 V [21]. Usually, in this model, the classical Biolek window is applied [24,31]. This model has a high accuracy.

The modified memristor model, applied in the present analyses and denoted by $A_{14\text{mod}}$ [35], is based on the Joglekar memristor model and the Lehtonen–Laiho model using the Hann window function $\sin^2(\pi x)$ [24,35]. The model has an activation threshold v_{thr} ,

applied in the state differential equation, using the standard Heaviside step function [23]. This modified memristor model is presented by the next set of Equation (6) [35]:

$$\begin{cases} i = v \cdot [R_{ON}x + R_{OFF}(1 - x)]^{-1} \\ \dot{x} = k \cdot \sin^2(\pi x) \cdot v^m \cdot \text{stp}(|v| - v_{thr}) \end{cases} \quad (6)$$

where k is a constant dependent on the physical parameters of the memristor nanostructure and m is an odd and integer exponent [35]. This modified memristor model has a simple mathematical structure, and the applied Hann window function is not related to terminal state problems. If the voltage v is lower than the activation threshold v_{thr} , then the time derivative of the state variable $\dot{x} = dx/dt$ is zero and the memristor behaves as a simple and linear resistor with a constant conductance [24,26]. This operating mode is used for the functioning of metal-oxide memristors in adjusted neural networks after finishing the training processes and establishing constant values of the synaptic weights [24]. The modified memristor model *A14 mod* is a simple one, with a high operating speed and good accuracy, and it is very appropriate for application in neural networks [35].

2.3. Fine Adjustment and Parameter Estimation of the Applied Memristor Model [24,31,32]

In this work, the authors analyze the considered modified model of a memristor based on metal oxides [35], which is fully allocated by Equation (6). This memristor model incorporates multiple parameters that allow its precise fine-tuning process. In [35], this modified memristor model is tuned according to experimental data of metal-oxide memristors [20]. In this work, it is adjusted in accordance with current–voltage dependencies, derived by experimental data of Known self-directed channel memristors [36]. A technique for altering the model's parameters till reaching the complete minimum of the Root Mean Square (RMS) error between the simulated and the experimental current of the memristor is applied [32]. Many scientists utilize simulated annealing and gradient descent algorithms for reaching the optimal values of the parameters of the memristor models [24,32]. The applied model is simple and applicable for simulations of memristor-based circuits and devices. The MATLAB-SIMULINK environment is utilized to extract parameters from the considered memristor model [32]. The optimization approach for tuning the metal-oxide memristor model involves modifying the coefficients and looking for the global minimum of the root mean square (RMS) error between the experimental and simulated current–voltage (i-v) characteristics. At each iteration step, one of the coefficients of the model alters by a small growth [32]. The RMS error between simulated and experimental i-v relations is computed. The other parameters of the model also change. After finalizing the tuning procedure, a graphical evaluation of the obtained current–voltage relationship and its nearness to the experimental one is also realized, paying attention to the form of the derived i-v pinched hysteresis loop and especially to the intervals of switching the resistance of the memristor element. The respective time diagrams of simulated and experimental memristor currents are also compared. An important condition for finalizing the adjustment process is the minimization of RMS error [32]. Extra simulations are conducted for obtaining the optimal model's coefficients, applying smaller increments for their alterations [32]. The precise values of the memristor model's parameters are also obtained, using the least squares method in MATLAB, applying Simulink Optimization Toolbox [32]. The parameter estimation process can be summarized in several main steps:

1. Initialization—choosing the initial values of the model's coefficients, and using these values, presented in the original references [36], and the parameters in the modified memristor model (6), after their evaluation in the Simulink and MATLAB environment [32];
2. Determination of the value of root mean square error (RMSE) for stopping the estimation process, the corresponding deviations, and the number of maximum iteration steps;

3. Starting simulation and computing the current of the memristor element according to the applied modified memristor model;
4. Evaluation of the RMS error (the cost function) between the calculated and experimental currents of the considered memristor;
5. Alteration of the model's parameters according to a gradient descent of RMS error;
6. When the determined RMS error is reached, or the maximal number of iteration steps are realized, the simulation process and evaluation of the derived model's parameters are finished. The voltage signals in the MATLAB-Simulink memristor model are previously sampled, the time step has a value of 100 ns. The experimental memristor current is represented by i_{mes} . The output of the Simulink memristor model is related to the simulated current of the memory element i_{calc} . The cost function S_{cost} is an algebraic sum of the squared differences between the experimental and calculated currents—Equation (7) [24,32]:

$$S_{cost}(i_{mes}, i_{calc}) = \sum_{k=1}^N [i_{mes}(k) - i_{calc}(k)]^2 \quad (7)$$

where $N = 100,000$ is the total samples of the considered signals, and k is the number of the current and considered sample. The criterion for finalizing the parameter estimation process is minimizing the cost function S_{cost} [24]. The obtained optimal values of the model's parameters are used for creation of the respective LTSPICE memristor library model, which is considered in the next section. The obtained values of the coefficients ensure that the RMS error is about 15%. The derived normalized parameters' trajectories of considered models A_{14mod} , K_2 , K_3 , and K_5 [19,31] and the obtained optimal parameters are presented in Figure 1 for their visual observation and comparison and for the expression of their alteration in time domain. The initial values of parameters are chosen, according to [36]. After finishing tuning of considered model A_{14mod} , the obtained optimal values of parameters are $k = 4.82 \times 10^5$; $x_0 = 0.00506$; $R_{on} = 1.05 \Omega$; $R_{off} = 50.027 \text{ k}\Omega$; $m = 3$; $v_{thr} = 0.205 \text{ V}$.

After the parameter estimation process, the considered memristor models K_2 , K_3 , K_5 , and A_{14mod} are analyzed in Simulink-MATLAB and the derived results are presented in Figure 2 for their visual comparison. The obtained normalized time diagrams of the applied voltage signal, and the corresponding experimental and simulated memristor currents and state variable after parameter extraction process are shown in Figure 2a for their visual expression and comparison. The next Figure 2b represents the corresponding experimental and simulated current–voltage characteristics, according to models A_{14mod} , K_2 , K_3 , and K_5 [19,31] for their comparison and visual observation.

For a short comparison of the considered modified metal-oxide memristor model to some of the standard existing and frequently utilized memristors (K_2 , K_3 , and K_5 [19,31]), several main criteria, such as the simulation time, the operation rate, accuracy and convergence [24,31], are considered in Table 1.

Table 1. A short comparison of considered metal-oxide memristor models—Joglekar (K_2), Biolek (K_3), Lehtonen–Laiho (K_5), and the applied modified memristor model A_{14mod} [19,31,35].

Model	Joglekar (K_2) [22]	Biolek (K_3) [23]	Lehtonen–Laiho (K_5) [21]	A_{14mod} [35]
Switching properties	middle	satisfactory	good	good
Frequency	low	low, middle	high	high
RMS error	17.6	16.3	3.4	14.2
Complexity	low	middle	high	low
Simulation time, ms	15.4	16.7	18.3	17.5

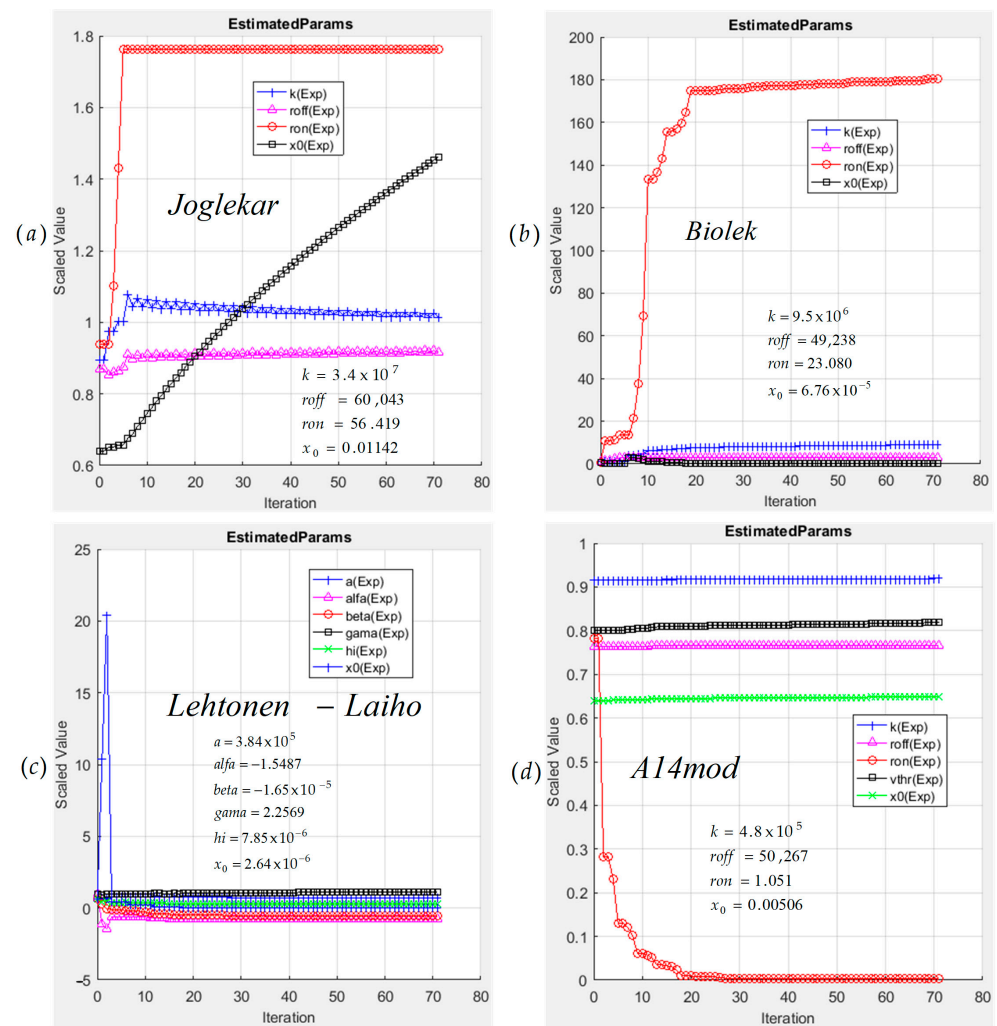


Figure 1. (a) Parameter trajectories during estimation—Joglekar (K_2); (b) Biolek (K_3); (c) Lehtonen–Laiho (K_5); (d) A_{14mod} .

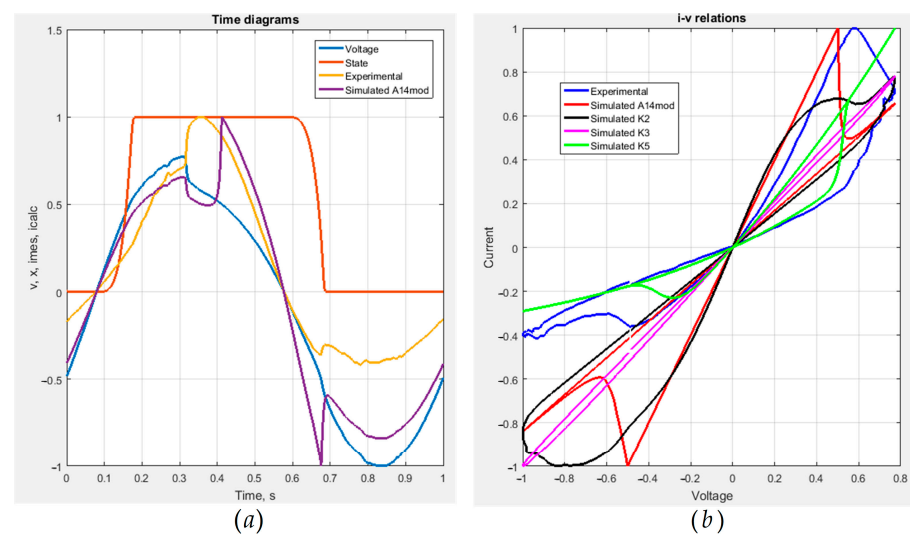


Figure 2. (a) Normalized time diagrams of experimental and simulated currents after parameter estimation process, voltage and state variable of modified memristor model A_{14mod} ; (b) A comparison of experimental and simulated normalized i - v relations after parameter estimation process—models A_{14mod} , K_2 , K_3 , and K_5 [19,31].

After the conducted comparison between the memristor models, it could be stated that the applied modified memristor model $A_{14\text{mod}}$ has a little bit lower accuracy than the Lehtonen–Laiho model, but it also has lower complexity, good switching properties, and high operating frequency [19,35]. According to the standard Joglekar and Biolek memristor models, the modified model $A_{14\text{mod}}$ has better switching characteristics and properties with a little bit higher complexity and simulation time. In state equations of Joglekar and Biolek models, the time derivative of state x is dependent on current i , which is related to the applied voltage v by a linear function, the first equation of models (2)–(4), which represents current–voltage characteristics. Due to this, the time derivative of x depends on voltage v . In contrast, in the Lehtonen–Laiho model and in the applied modified model $A_{14\text{mod}}$, the time derivative of state depends on a nonlinear polynomial of voltage v^m . Owing to this, the model has different behavior when the voltage is lower or higher than one volt. Voltages higher than one volt lead to a rapid change in x with the voltage. This behavior is very important when the memristor operates in a hard-switching mode—in logic gates, in synapses for neural networks, and in non-volatile memory devices. The type of the applied memristor model is important for training the networks and the adjustment of the synaptic weights of memristor-based neural networks by short voltage pulses. The proposed model is a simple and fast-operating one and has good accuracy when it is fitted according to experimental i – v relationships of real memristors [36]. Its simplicity allows designers and engineers to apply it for analysis and simulations of complex neural circuits with a large number of memristors.

In the next section, the generation and analysis of the LTSPICE memristor model, corresponding to the modified memristor model $A_{14\text{mod}}$, and its main properties are analyzed and discussed.

3. LTSPICE Library Model [35] and Analysis of the Applied Memristor Model

The main idea, utilized for the construction of the LTSPICE [33] memristor library model, is to build a simple and fast-operating equivalent circuit, corresponding to (6). It is presented in Figure 3a for further discussion. Before realization of the current–voltage characteristics, the derivation of the state variable x is needed. The memristor state variable x is obtained by integration of the state differential equation with respect to time. This process is ensured by a capacitor, denoted as C_1 ; its current is set to be proportional to the time derivative of the memristor state variable x [26,31]. The current of the capacitor C_1 is generated by the voltage-controlled current source G_2 , according to the left-hand side of the state equation in (6). The initial voltage of the integrating capacitor C_1 is set to be equal to the starting value of the memristor state variable x_0 . The voltage across the integrating capacitor is directly proportional to the state variable in each moment in the time domain. To avoid rapid changes in the voltage across the capacitor, which could lead to convergence issues, a high-valued resistor R_1 is attached in parallel to capacitor C_1 [31,36].

The voltage-controlled current source G_1 generates the memristor current, according to the right-hand side of the first equation in (6). The initial row of LTSPICE code shown below displays the memristor model identifier, indicated as $A_{14\text{mod}}$.

```

1 .subckt A14mod te be Y
2 .params ron = 100 roff = 25e3 m0 = 300 k = 10e4 m = 5 vthr = 0.2 mm = 1e-24
3 C1 Y 0 {1}
4 .IC V(Y)={(m0-roff)/(ron-roff)}
5 R1 Y 0 100G
6 G2 0 Y value={((k*pow(V(te,be),m))*((pow(sin(pi*V(Y)),2))))*
7 (stpp((vthr-abs(V(te,be))),mm))}
8 G1 te be value={V(te,be)*((1/(ron*(V(Y))+roff*(1-V(Y))))}
8 .func stpp(x,p)={0.5*(1+(x/sqrt(pow(x,2)+p)))}
9 .ends A14mod

```

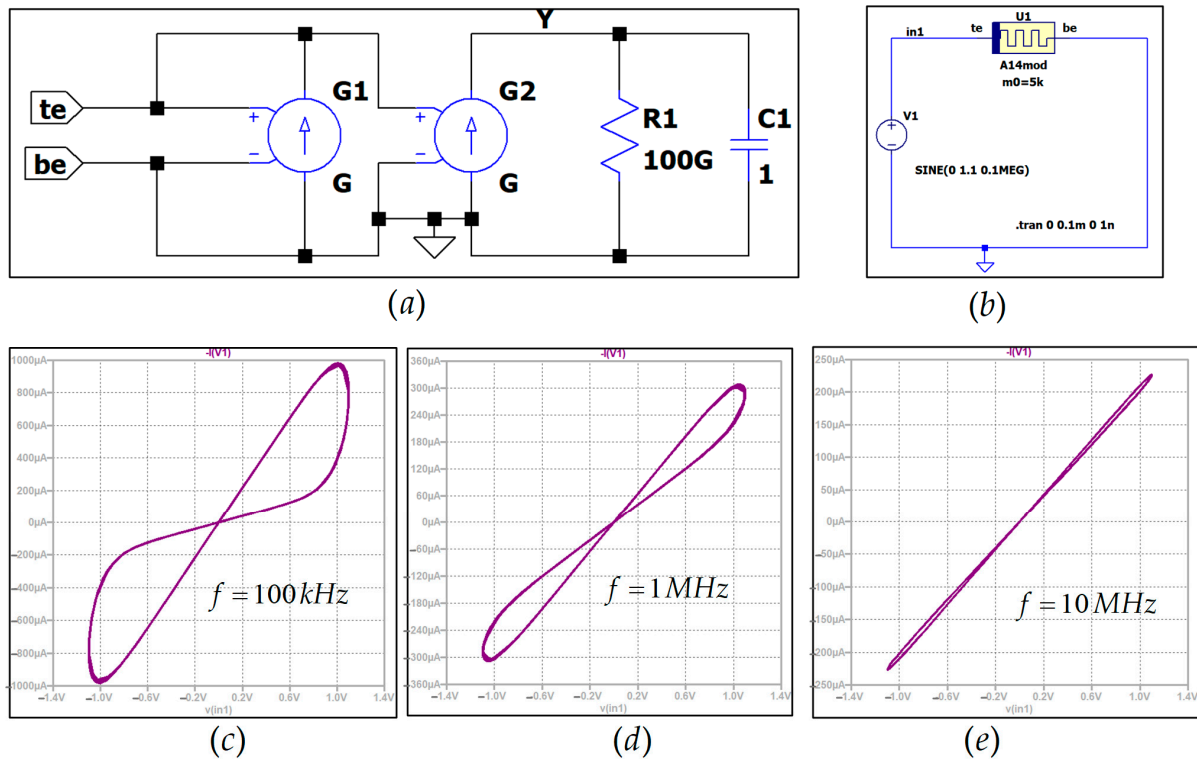


Figure 3. (a) A schematic of LTSPICE memristor model, realized in LTSPICE simulator; (b) A principal schematic of memristor model $A_{14\text{mod}}$, included in a simple electronic circuit for analysis at sinusoidal mode, supplied by a voltage source with an amplitude of 1.1 V and different frequencies; (c) Current–voltage relation of the model $A_{14\text{mod}}$ at 100 kHz; (d) i - v characteristic of model $A_{14\text{mod}}$ at 1 MHz; (e) current–voltage characteristic of the model, obtained at 10 MHz.

The memristor terminals are denoted as top electrode (te) and bottom electrode (be). The optional terminal Y is utilized for measuring the memristor state variable. The parameters of the model— R_{ON} , R_{OFF} , k , m , mm , m_0 , and v_{thr} are presented in the next row. The coefficient m_0 represents the initial value of memristance. The integrating capacitor C_1 connected between terminal Y and the ground electrode has a capacitance of 1000 mF. The initial voltage of integrating element C_1 is denoted as “IC”. It is expressed by the first equation of (6) [35]. It is inversely proportional to the starting value of the memristance. The fifth row represents the additional resistor R_1 with a value of 100 G Ω , connected in parallel to C_1 . The dependent source G_2 presented in the sixth row expresses the time derivative of state variable x . The controlled source G_1 expresses the current of the memristor element. In row 8, a differentiable and flat step-like function is presented. It is utilized for the realization of the state equation, containing the memristor activation threshold v_{thr} . The parameter mm determines the sharpness of the step-like function $stpp$. The code of applied modified model $A_{14\text{mod}}$ is finished with the command “ends” [26,33,35]. For a quick and easier tuning of the initial value of memristance m_0 , the state variable of the memristor in the initial moment is expressed as a function of m_0 . After selecting the memristor model $A_{14\text{mod}}$ in LTSPICE and placing it on the plotting field, right-click on the memristor, in the SPICE row of the appeared window, write $m_0 = 300$, then place a tick in the right by double clicking the left mouse button, to make m_0 visible, and confirm by clicking OK. The considered LTSPICE code is included in a unified memristor model library, freely available for use and download at <https://github.com/mladenovvaleri/Advanced-Memristor-Modeling-in-LTSpice> (accessed on 1 January 2024) [31]. To use this link, simply click on it or copy and paste it in the web browser. This library contains many LTSPICE models of memristors and memristor-based electronic circuits, available for use and comparisons.

The considered modified memristor model is analyzed with pulse and sine signals at a constant amplitude of 1.1 V and for different frequencies. In Figure 3b, the testing circuit for memristor analysis is presented for its visual observation. In the present case, the additional electrode Y is not included for simplification of the electronic circuitry. The initial value of the memristance is set to be 5 k Ω . After obtaining the i - v relationships of the memristor model for sinusoidal signals with amplitude of 1.1 V and different frequencies (100 kHz, 1 MHz, and 10 MHz) in the LTSPICE environment, they are presented in Figure 3c–e for their visual comparison and for confirmation of the correct operation of the modified memristor model. It is visible that, when the signal's frequency increases, then the surface of the i - v loop decreases in accordance with the basic memristors' properties and fingerprints [24,30].

Memristors, participating as synaptic weights, are tuned during the neural network training by positive or negative voltage pulses with a level, the absolute value of which is higher than the activation threshold v_{thr} [35]. An example of synaptic weight adjustment is presented in Figure 4. A simple electronic circuit, containing a pulse voltage source and a memristor, represented by the modified model A_{14mod} , is shown in Figure 3b for a visual expression of its structure and operation.

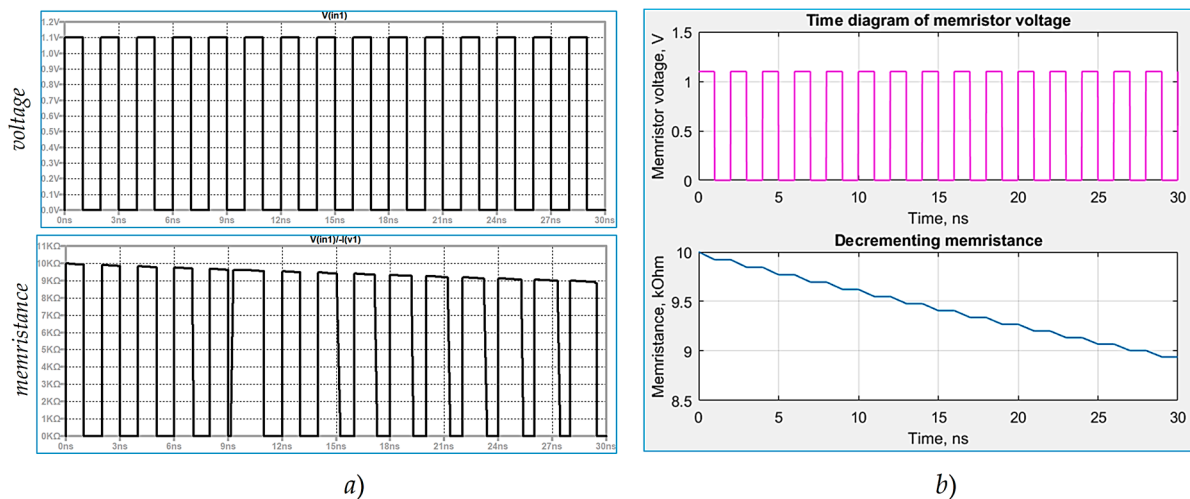


Figure 4. Memristance adjustment by positive voltage pulses; (a) time diagrams of pulse voltage signal with amplitude of 1.1 V and a frequency of 500 MHz, pulse duration of 1 ns, and corresponding memristance in LTSPICE; (b) time diagram of voltage and memristance, corresponding to the applied pulses in MATLAB.

The voltage source generates a sequence of rectangular and positive pulses with a duration of 1 ns and a duty cycle of 50%. The amplitude of the voltage is 1.1 V. Figure 4a represents the time diagrams of the pulse voltage signal and the corresponding change in the memristance M in LTSPICE. In this case, the initial value of the memristance is $M_1 = 10$ k Ω , which corresponds to a synaptic weight of $w_1 = 1$. The final value of $M_2 = 9$ k Ω corresponds to a synaptic weight of $w_2 = 1.1111$. In this case, the change in memristance with $\Delta M = 1$ k Ω corresponds to an alteration of the synaptic weight with $\Delta w = 0.1111$. The needed time interval for this memristance change, together with the pauses between the impulses, is about 28 ns. The effective duration of the voltage pulses for this time interval is about 14 ns. Figure 4b represents the results, derived by a simulation in MATLAB, which have very good coincidence with Figure 4a and confirm the tuning process. This result will be also mathematically confirmed by integration of the state differential equation of system (6), modified and represented here by (8):

$$\dot{x} = k \cdot \sin^2(\pi x) \cdot v^m \quad (8)$$

In Equation (8), because the voltage pulses used are higher than the activation threshold v_{thr} , in the right-hand side, the step function “ stp ” is missed. Equation (8) is transformed to (9), where the state variable x and the time variable t are separated one from another:

$$\frac{dx}{\sin^2(\pi x)} = k \cdot v^m dt \quad (9)$$

According to the current–voltage relationship in system (6), the state variable x is expressed as a function of the memristance M by Equation (10):

$$x = (M - R_{OFF}) \cdot (R_{ON} - R_{OFF})^{-1} \quad (10)$$

Equation (9) is integrated between $x_1 = 0.8251$, corresponding to $M_1 = 10 \text{ k}\Omega$, and $x_2 = 0.8427$, related to $M_2 = 9 \text{ k}\Omega$, using (10), and the following Equation (11) is obtained:

$$\int_{x_1}^{x_2} \frac{1}{\sin^2(\pi x)} dx = k \cdot v^m \int_0^{T_{\max}} dt \Rightarrow - \left[\frac{\cot g(\pi x_2)}{\pi} - \frac{\cot g(\pi x_1)}{\pi} \right] = k \cdot v^m \cdot T_{\max} \quad (11)$$

Then, the effective duration of the applied pulses (as a single pulse or a sequence of short impulses) is obtained as follows—Formula (12):

$$T_{\max} = \frac{- \left[\frac{\cot g(\pi x_2)}{\pi} - \frac{\cot g(\pi x_1)}{\pi} \right]}{k \cdot v^m} = \frac{- \left[\frac{\cot g(\pi \cdot 0.8427)}{\pi} - \frac{\cot g(\pi \cdot 0.8251)}{\pi} \right]}{3.10^6 \cdot 1.1^5} = 14.53 \text{ ns} \quad (12)$$

The obtained time for synaptic weight adjustment corresponds to the derived simulation result and confirms the proper tuning of the memristance by pulses. Another simulation is conducted with negative voltage pulses and the same circuit shown in Figure 3b is used. The respective time diagram of the applied negative pulses and the corresponding memristance alteration in LTSPICE is presented in Figure 5a. Figure 5b represents the results, derived from analysis in MATLAB, which is in good agreement with Figure 5a and confirms the correctness of the adjustment process. The analysis starts with the initial value of memristance, equal to $M_1 = 5 \text{ k}\Omega$, related to a synaptic weight of $w_1 = 2$ and a state variable of $x_1 = 0.9129$. The final value of memristance M_2 is $6 \text{ k}\Omega$, corresponding to a synaptic weight with a value of $w_2 = 1.6667$ and a state variable $x_2 = 0.8954$. The change in memristance with $1 \text{ k}\Omega$ in this case corresponds to an alteration of synaptic weight with $\Delta w = 1.6667 - 2 = -0.3333$. The effective duration of pulse is about $79 \text{ ns}/2 = 39.5 \text{ ns}$.

The analytical confirmation of this result is as follows—Formula (13):

$$T_{\max} = \frac{- \left[\frac{\cot g(\pi x_2)}{\pi} - \frac{\cot g(\pi x_1)}{\pi} \right]}{k \cdot v^m} = \frac{- \left[\frac{\cot g(\pi \cdot 0.8954)}{\pi} - \frac{\cot g(\pi \cdot 0.9129)}{\pi} \right]}{-3.10^6 \cdot 1.1^5} = 41.5 \text{ ns} \quad (13)$$

For future and practical implementations of memristor-based synapses, the tuning process might be related to a measurement of the achieved memristance, until reaching the needed resistance and stopping the pulse sequence. According to the requirement for a quick change in the memristance and the corresponding synaptic weight, the applied modified memristor model $A_{14\text{mod}}$ ensures a rapid alteration of the state variable and the memristance at applied voltage pulses. Its simple structure and low simulation time ensure the generation and analysis of complex neural networks with a high number of memristors in LTSPICE. Additional comparison of applied memristor models, according to simulation time, is conducted for two cases—decreasing and increasing memristance, according to analyses shown in Figures 4 and 5. The results presented in Table 2 express the faster operation of the applied modified model, according to the Lehtonen–Laiho model. Simulations were conducted on a computer with an 8 GB RAM, i5 Intel processor, and Microsoft Windows 10 Professional operating system. According to the Joglekar

and Biolek models, the modified model 14_{mod} has comparable simulation time and better switching properties.

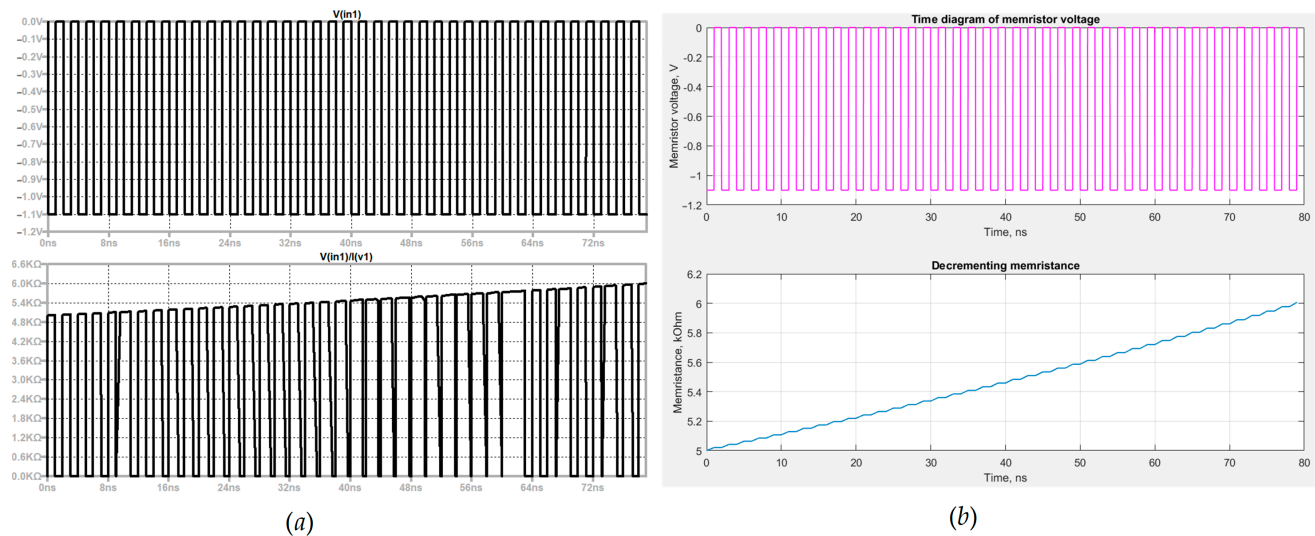


Figure 5. Memristance adjustment by negative voltage pulses; (a) time diagram of pulse voltage signal with a level of -1.1 V and a frequency of 500 MHz, and pulse duration of 1 ns in LTSPICE; (b) time diagram of voltage and the memristance, corresponding to the applied voltage pulses in MATLAB.

Table 2. Comparison of the applied memristor models K_2 , K_3 , K_5 , and $A_{14\text{mod}}$, according to simulation time.

Case	Models	Joglekar (K_2) [22]	Biolek (K_3) [23]	Lehtonen–Laiho (K_5) [21]	$A_{14\text{mod}}$ [35]
Decreasing memristance	Simulation time, ms	28.2	30.8	35.8	28.3
Increasing memristance	Simulation time, ms	28.5	31.1	37.8	24.3

4. Memristor-Based Transfer Function

The considered realization of a logarithmic-sigmoidal activation function is presented in Figure 6a for its visual expression and for further discussion. The considered transfer (activation) function is realized by two MOS transistors M_1 and M_2 , connected in anti-parallel, the memristor U_1 and a DC voltage source V_2 [35]. The MOS transistors are of N-type and in the present case, they are chosen to be *Si7540DP_N*, using the standard LTSPICE library [33]. The transistors are connected as diodes and their gate and source terminals are directly attached one to another. The anti-parallel connection of the MOS transistors ensures a symmetrical transfer function. The memristor U_1 is connected in series to the MOS transistors. Its voltage drop is the difference between the input and the output signals. The voltage source V_2 ensures shifting the output signal by 0.5 volts [35]. The dependence between the output and input voltage signals is very similar to a logarithmic-sigmoidal function, presented by the following expression (14):

$$v_{out} = [1 + \exp(-k \cdot v_{in})]^{-1} \quad (14)$$

For determination of the optimal value of the coefficient k , several different experiments and comparisons are conducted with the use of dependent voltage source B_1 shown in Figure 6b, in which voltage represents a logarithmic-sigmoidal activation function. It is established that for $k = 5$ and $m_0 = 150 \Omega$, the best proximity between the transfer functions is realized. The expression of voltage of behavioral source B_1 is $v_{ff} = 1 / (1 + \exp(-5 \cdot v_{in_ff}))$.

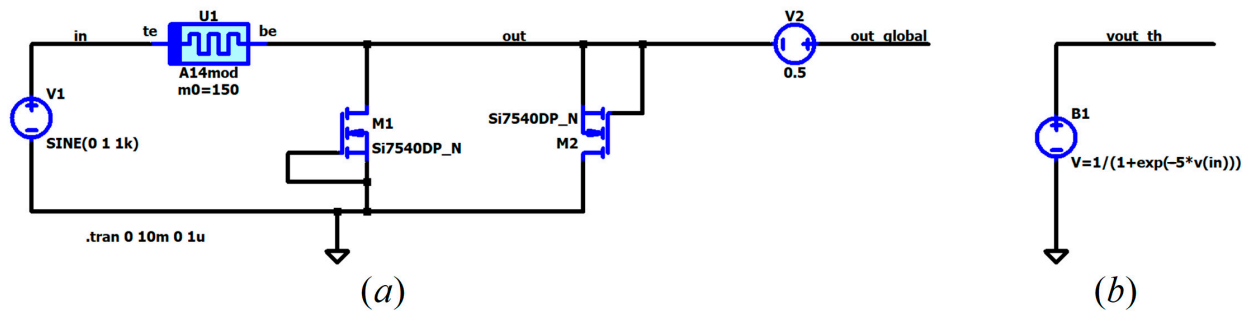


Figure 6. (a) A logarithmic-sigmoidal transfer function, realized by a metal-oxide memristor and two MOS transistors, connected in anti-parallel; (b) A transfer function, realized by a voltage-controlled voltage source.

The n -channel MOS transistor *Si7540DP_N* used in this paper has a simpler construction, lower drain-source voltage, lower drain current, and higher speed compared to the *Si4864DY* transistor used in [35]. The activation functions depicted in Figure 7 correspond to the circuits illustrated in Figure 6. These graphics are presented for their visual expression and comparison of their proximity. The transfer function of the MOS transistor-based schematic is represented by the black curve, while the theoretical activation function is depicted by the blue curve. After several experiments with different values of the coefficient k , a very good proximity between the considered transfer functions is established when the coefficient in the theoretical expression is $k = 5$.

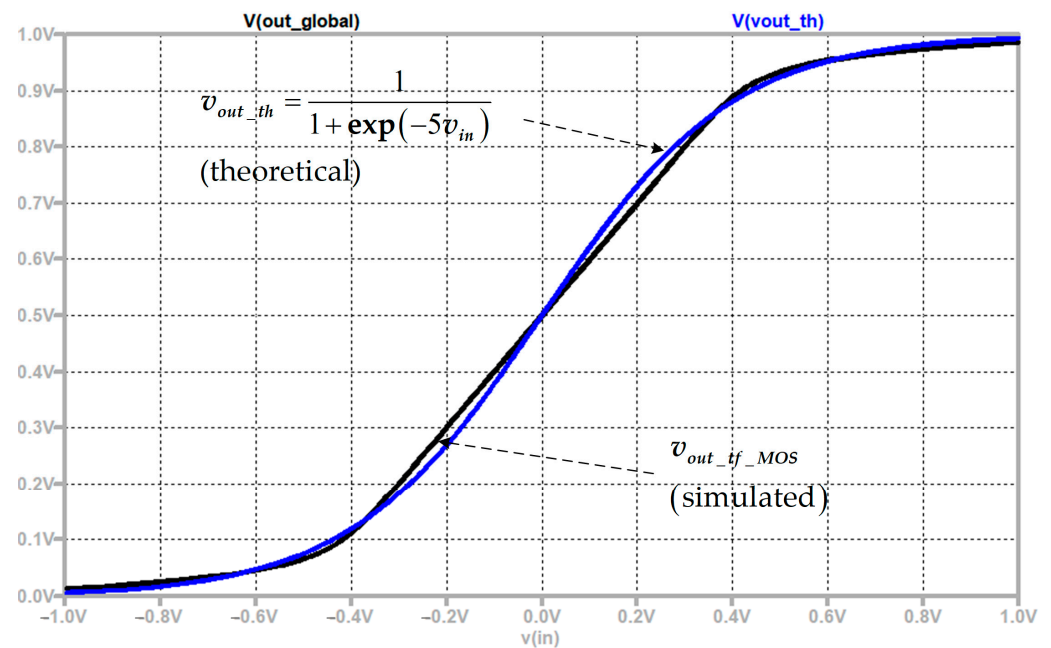


Figure 7. A comparison of the suggested transfer function, realized by MOS transistors and a memristor— $v_{out_tf_MOS}$, and the theoretical output signal, realized by a voltage-controlled source $v_{out_th} = 1 / (1 + \exp(-5v_{in}))$.

In the next section, an artificial neuron based on memristors and on the discussed memristor-based transfer function is presented.

5. Memristor-Based Neuron [35]

Figure 8 presents a block diagram of a neuron and illustrates the structure and operation of a neuron based on memristors for further explanation and discussion [2,3].

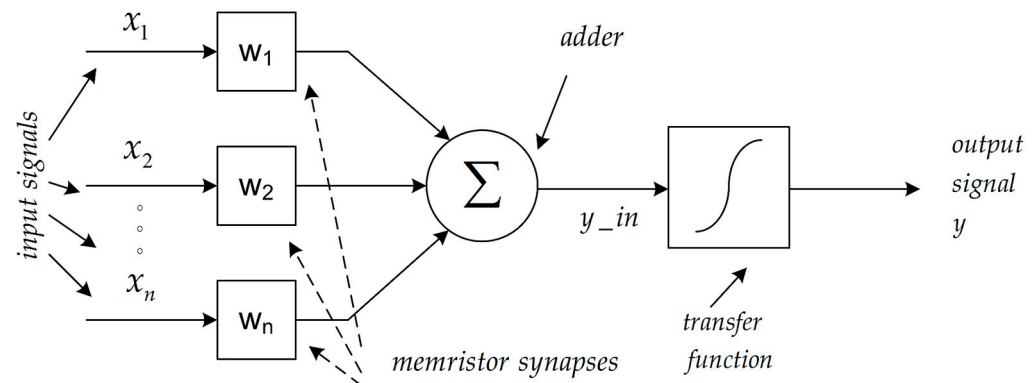


Figure 8. A block diagram of considered artificial neuron with memristor-based synapses.

The input signals of the neuron are represented as $x_1, x_2, \dots$, and x_N . The signals are directed to synapses based on memristors, with corresponding weights denoted as $w_1, w_2, \dots$, and w_N . A single memristor is employed for each synapse, and its resistance determines the corresponding synaptic weight. The change in the respective synaptic weight is realized by alteration of the corresponding memristance with externally applied voltage or current pulses. The signals obtained after the memristor synapses are directed to an adder, implemented by the use of operational amplifiers and memristors, operating as linear and constant resistors [34,35]. The signal derived after the adder is represented as y_{in} . This signal is directed to a log-sigmoidal transfer function, constructed by metal-oxide memristors and MOS transistors [35]. The output signal of the artificial neuron is indicated by y . The functioning of the memristor-based neuron under consideration is related to a feed-forward and back-error propagation learning algorithm [3]. Various activation functions could be utilized after the adder [3,7]. Commonly utilized in the neural networks are smooth and differentiable transfer functions, such as logarithmic-sigmoidal and tangent-sigmoidal ones [7]. The relay activation function, also known as the standard Heaviside function, is also utilized in artificial neural nets [3]. A gain factor can also be utilized to apply a linear activation function in artificial neurons.

Figure 9 illustrates a principal schematic of the considered artificial neuron based on memristors, providing a description of its structure and operation.

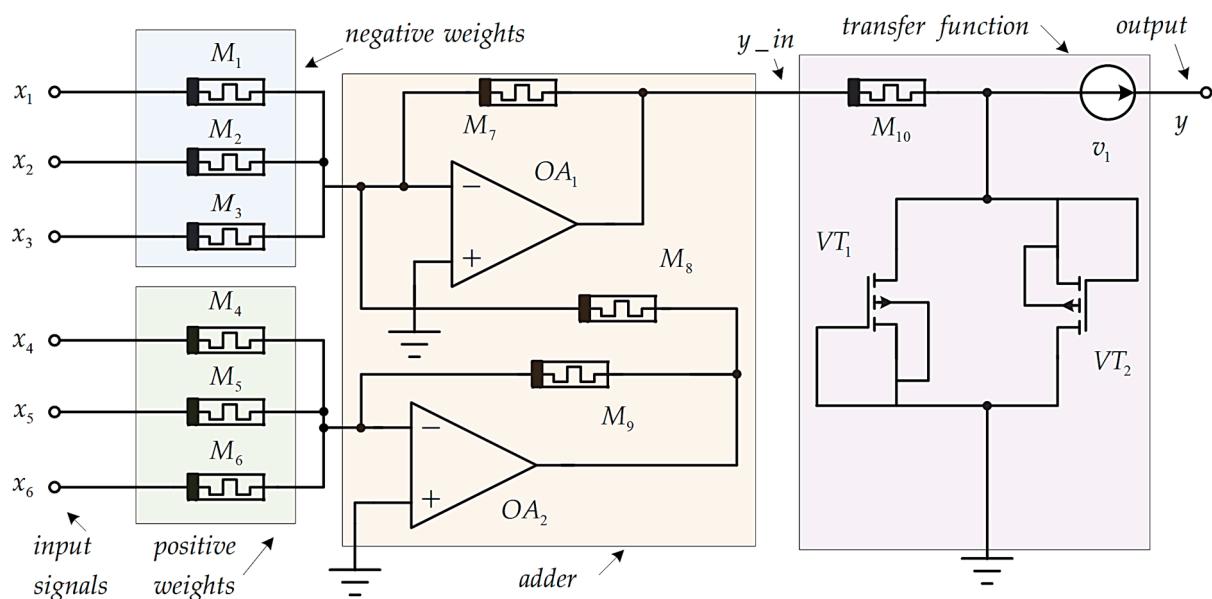


Figure 9. A principal schematic of metal-oxide memristor-based neuron with negative and positive synaptic weights.

The input signals x_1 – x_6 are normalized in the interval between -0.1 and $+0.1$ volts, in order to avoid alteration of the resistances of the synaptic weights during the application of the adjusted neural network. The number of the memristors, realizing the synaptic weights could vary, is according to the needed inputs of the artificial neuron. The synaptic weights are presented by the metal-oxide memristors M_1 – M_6 . The memristors M_1 , M_2 , and M_3 are related to negative weights, while the other memristors M_4 , M_5 , and M_6 are placed for the realization of the positive synaptic weights. The considered synapses are connected to the inverting inputs of the op-amps OA_1 and OA_2 , respectively. The memristors M_7 and M_9 are applied for the realization of voltage feedbacks of the respective op-amps. The output signal of the op-amp OA_2 is directed to the inverting input of op-amp OA_1 via the memristor element M_8 . The non-inverting inputs of the operational amplifiers are connected to the ground [34,35]. The feedback memristors M_7 , M_8 , and M_9 are adjusted to a memristance of $10\text{ k}\Omega$. Using Kirchhoff's laws, the output signal of op-amp 1, correspondent to the signal y_{in} , is derived as follows—Formula (15):

$$y_{in} = v_{out_OA1} = -\frac{M_7}{M_1}x_1 - \frac{M_7}{M_2}x_2 - \frac{M_7}{M_3}x_3 - \frac{M_7}{M_8}\left(-\frac{M_9}{M_4}x_4 - \frac{M_9}{M_5}x_5 - \frac{M_9}{M_6}x_6\right) \quad (15)$$

Having in mind that $M_7 = M_8 = M_9 = 10\text{ k}\Omega$, the next expression (16) is derived after algebraic processing of Formula (15):

$$y_{in} = v_{out_OA1} = -\frac{M_7}{M_1}x_1 - \frac{M_7}{M_2}x_2 - \frac{M_7}{M_3}x_3 + \frac{M_7}{M_4}x_4 + \frac{M_7}{M_5}x_5 + \frac{M_7}{M_6}x_6 \quad (16)$$

where the coefficients in front of the signals x_1 – x_6 are the corresponding synaptic weights—Formula (17) [34,35]:

$$w_1 = -\frac{M_7}{M_1}; \quad w_2 = -\frac{M_7}{M_2}; \quad w_3 = -\frac{M_7}{M_3}; \quad w_4 = \frac{M_7}{M_4}; \quad w_5 = \frac{M_7}{M_5}; \quad w_6 = \frac{M_7}{M_6} \quad (17)$$

Observably, synaptic weights w_1 , w_2 , and w_3 are with negative values, whereas the weights w_4 , w_5 , and w_6 are with positive values.

Figure 10 illustrates the implementation of the considered memristor neuron in the LTSPICE environment, providing further details and discussion. It is associated with the principal electronic circuit presented in Figure 9. The presented pulse voltage sources V_1 , V_2 , V_7 , V_8 , V_9 , and V_{37} correspond to the signals x_1 – x_6 and they generate the input signals v_{in1} , v_{in2} , v_{in3} , v_{in4} , v_{in5} , and v_{bias} . The adder has two inputs—for negative and for positive synaptic weights—which are denoted by *minus2* and *plus2*, respectively [35].

The output electrode of the adder is denoted by “*summ*”. The structure of the log-sigmoidal transfer function corresponds to Figure 6a. A detailed realization of the adder in the LTSPICE simulator is presented in Figure 11 for its description and further explanation.

The op-amps U_1 and U_2 are power supplied by two DC voltage sources of 3 volts. Sometimes, the voltage across the feedback memristors could exceed the activation threshold of the applied metal-oxide memristors [35]. The feedback voltage depends on the output signal y_{in} , which is proportional to the weighted sum of the input signals. To avoid the respective voltages across the feedback memristors to exceed the memristor activation threshold, the voltage feedback elements M_7 , M_8 , and M_9 are substituted by memristor-based blocks, enclosing two memristors connected in a series connection [35].

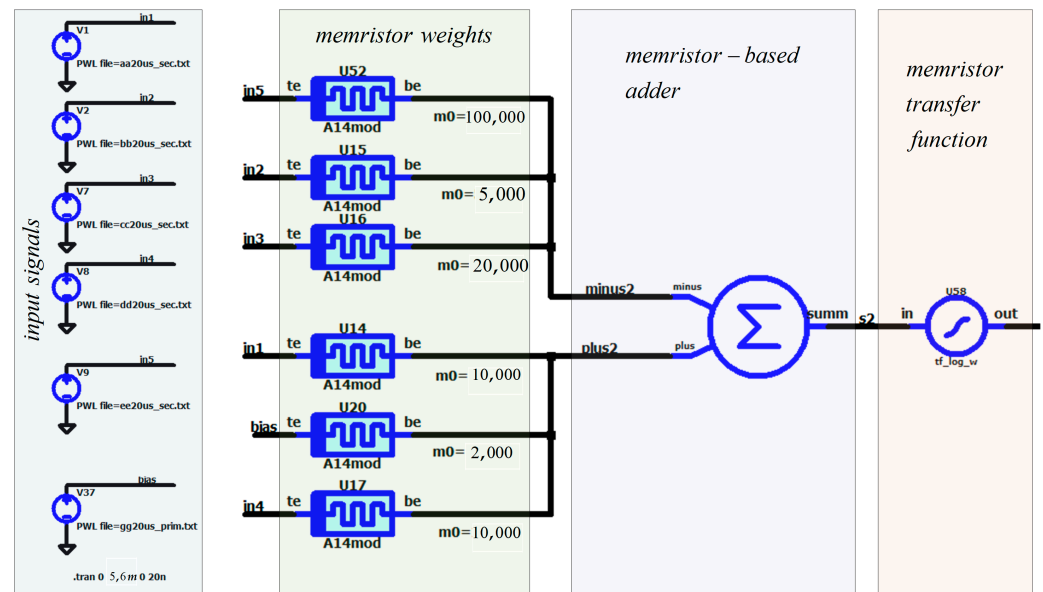


Figure 10. LTSPICE realization of a model of artificial neuron with memristor synapses and op-amps, included in the adder.

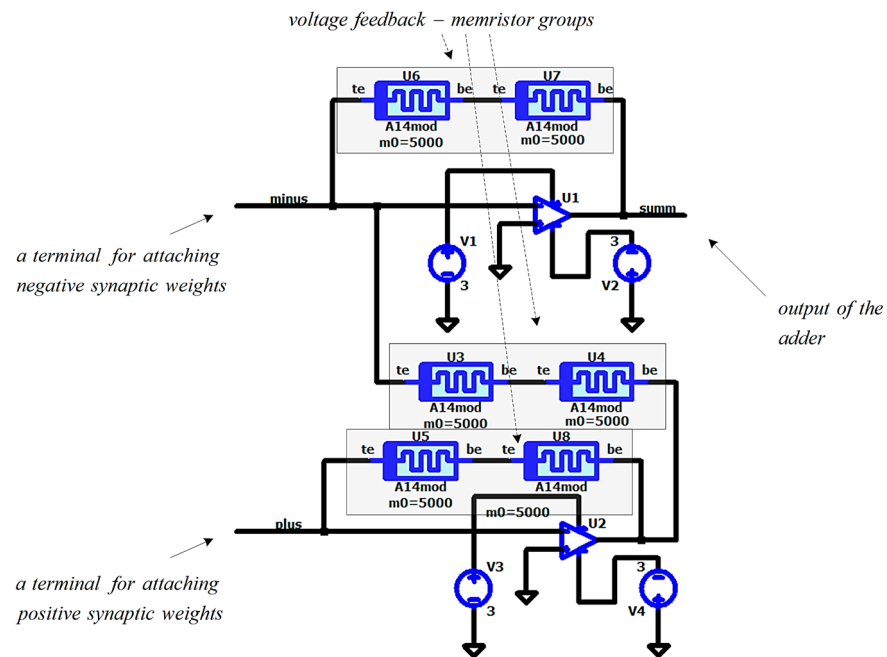


Figure 11. A realization of the considered memristor-based adder in LTSPICE.

The exact number of memristor elements n in a block, realizing a feedback element could be calculated, using the absolute value of the output signal $v_{summ} = y_{in}$ (in Figure 9), divided by the activation threshold v_{thr} , and applying the truncation function “*ceil*”, which returns the smallest integer, not less than its argument—Formula (18):

$$n = \text{ceil}\left(|v_{summ}| \cdot v_{thr}^{-1}\right) \quad (18)$$

The signal after the adder v_{summ} is connected to a memristor-based voltage divider and a buffer amplifier, presented in Figure 12. The transfer coefficient of the voltage divider is 0.1. This divider is needed for ensuring the normal operation of the memristor-based synapses of the next layer of the neural network [34,35], in which the voltage should not exceed the memristor activation threshold v_{thr} .

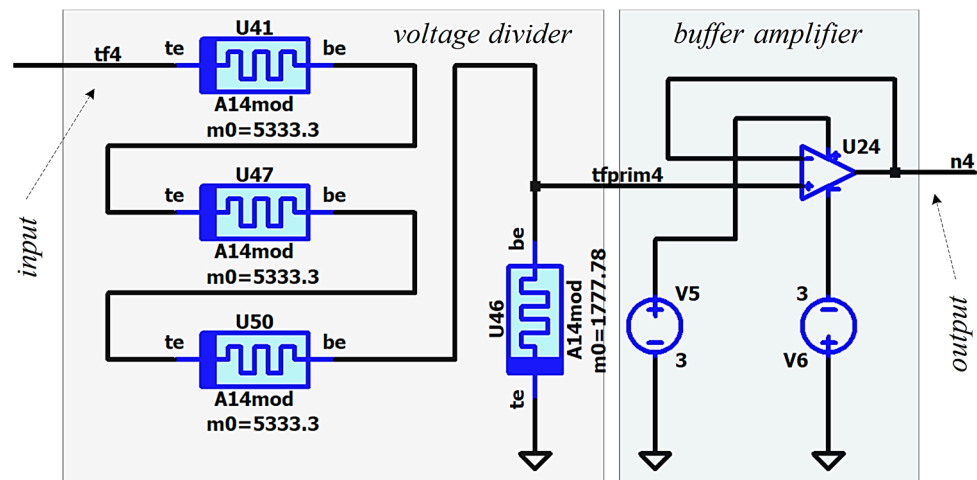


Figure 12. A voltage divider with memristors, connected to a buffer amplifier in LTSPICE environment.

The element U_{46} has a constant resistance of 1777.78Ω and is realized by only one memristor, while the number of the other memristors in the voltage divider is determined by Formula (19), to ensure that the voltage drop across the memristors does not exceed the activation threshold v_{thr} :

$$n = \text{ceil} \left(\left| v_{tfprim4} \right| \cdot v_{thr}^{-1} \right) \quad (19)$$

Sometimes, the normal operation of the memristors in the voltage divider could be violated, owing to the connected synapses of the next layer of the neural network. To ensure their normal functioning without exceeding the activation threshold and to avoid functioning in a soft-switching mode, a buffer amplifier is connected after the voltage divider. It is based on the non-inverting operational amplifier U_{24} .

6. Memristor-Based Neural Network with TF, Realized with MOS Transistors

The block diagram of the memristor-based neural network under analysis is presented in Figure 13 for its visual expression and further comments on their structure and operation. It contains five input nodes for applying the input signals x_1 – x_5 . The hidden layer of the neural network contains four neurons— N_1 , N_2 , N_3 , and N_4 . The bias signals are denoted by b_1 , b_2 , b_3 , and b_4 . The synaptic weights are denoted by w_{ij} , where $i = 1$ – 5 is the number of the neuron, and j is the number of the input signal. The adders are used for summing the weighted input signals. The output signals of the adders are denoted by s_1 , s_2 , s_3 , and s_4 . These signals are applied to the logarithmic-sigmoidal activation functions tf_1 , tf_2 , tf_3 , and tf_4 , respectively. The described elements form the hidden layer of the considered neural network. The signals tf_1 , tf_2 , tf_3 , and tf_4 are applied to the output layer of the neural network, which contains two neurons— N_{out1} and N_{out2} . The synaptic weights are denoted by v_{kp} , where k is the number of the neuron, and p is the number of the respective input signal. The bias signals of the adders are denoted by b_{out1} and b_{out2} . The output signals of the adders are denoted by s_{1out} and s_{2out} . The applied transfer functions tf_{out1} and tf_{out2} are of the type “purelin” [3]. They are realized with buffer amplifiers, using op-amps. The output signals of the neural network are denoted by y_1 and y_2 .

The operation of the presented memristor-based neural network is based on its training in the MATLAB environment, using feed-forward and back-error propagation algorithms for the adjustment of the synaptic weights [3,32]. The obtained values of the weights are used for tuning the memristances and to create the neural network in the LTSPICE environment, which will be discussed in the next section. The input signals are sequences of rectangular voltage pulses. The pulse duration is 0.8 ms. The duty cycle is 50%. The levels of the input and the bias voltage signals are shown in Table 3 for their visual evaluation. The bias signal is set to 100 mV and does not exceed the memristor activation threshold v_{thr} .

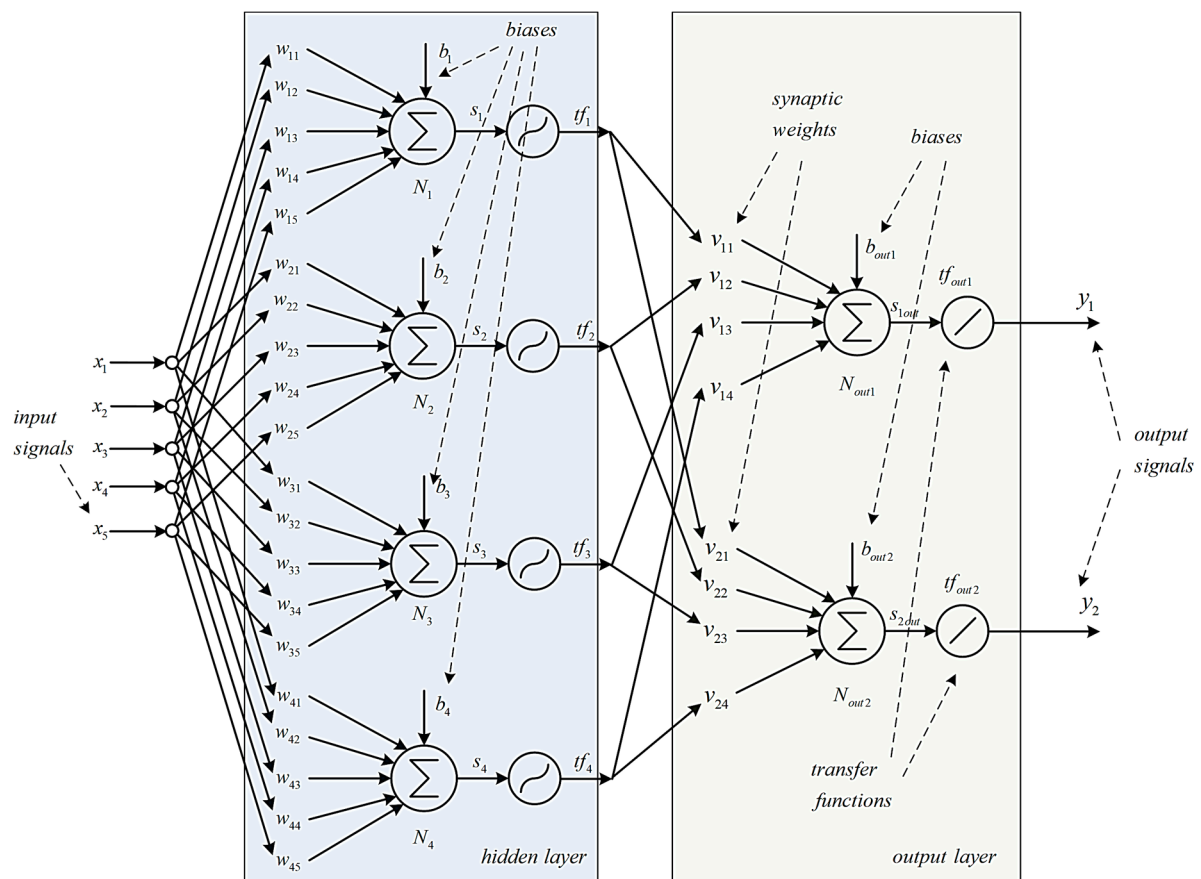


Figure 13. Block diagram of the considered memristor-based neural network, containing five input nodes, four neurons in the hidden layer and two neurons in the output layer.

Table 3. Input and bias signals for the memristor-based neural network, to be realized in LTSPICE.

Input Signals, mV					Bias Signal, mV
v_{in1}	v_{in2}	v_{in3}	v_{in4}	v_{in5}	v_{bias}
10	70	40	50	20	100

The synaptic weights for the hidden layer of the neural network are presented in Table 4 for their evaluation and for further discussion.

Table 4. Synaptic weights for the hidden layer of the considered memristor NN in LTSPICE.

Synaptic Weights for the Hidden Layer						
Neuron	w_1	w_2	w_3	w_4	w_5	w_{bias}
N_1	−0.2	1	−0.1	0.5	0.2	0.2
N_2	1	−2	−0.5	1	−0.1	0.5
N_3	0.2	0.5	−1	1	−0.2	−0.2
N_4	0.1	−0.5	1	0.1	0.2	−0.1

The corresponding resistances of the single-memristor synapses are denoted by M_1 – M_5 and are presented in Table 5. Their values are obtained using Formula (11). Having in mind that the bias signal is $v_{bias} = 0.1$ V, instead of 1 V, the value of the memristance M_{bias} is obtained by multiplication with 10.

Table 5. Memristances for the synapses in the hidden layer of the considered NN in LTSPICE.

Input Memristances, k Ω						
Neuron	M_1	M_2	M_3	M_4	M_5	M_{bias}
N_1	−50	10	−100	20	50	5
N_2	10	−5	−20	10	−100	2
N_3	50	20	−10	10	−50	−5
N_4	100	−20	10	100	50	−10

The weights and memristances for the output layer are shown in Tables 6 and 7.

Table 6. Synaptic weights for the output layer of the considered NN in LTSPICE.

Synaptic Weights for the Output Layer					
Neuron	M_{1out}	M_{2out}	M_{3out}	M_{4out}	$M_{biasout}$
N_{1out}	−0.1	0.1	0.2	−0.1	−1
N_{2out}	−0.05	0.05	0.1	−0.1	−0.5

Table 7. Memristances for the output layer of the considered NN in LTSPICE.

Output Layer Memristances, k Ω						
Neuron	M_1	M_2	M_3	M_4	M_5	M_{bias}
N_1	−10	10	5	−10	−10	10
N_2	−20	20	10	−10	−20	−10

A principal schematic of the considered memristor-based neural network is illustrated in Figure 14. The input signals are denoted by x_1 – x_5 , the bias signal is denoted by b . This schematic corresponds to the block diagram, presented in Figure 13.

The electronic realization of the considered neural network in the LTSPICE simulator, corresponding to the block diagram of Figure 13 and to the principal schematic presented in Figure 14, is given in Figure 15. It is presented for a visual observation and realization by the readers of similar SPICE simulators. The circuits are also available at: <https://github.com/mladenovvaleri/Advanced-Memristor-Modeling-in-LTSpice> (accessed on 17 November 2023) [31]—click on the hyperlink or copy and paste it in the web browser. Figure 16a illustrates the time diagrams of the input signals and the bias signal, giving their visual representation. The signals after the adders are presented in Figure 16b. Figure 16c represents the signals after the logarithmic-sigmoidal transfer functions and Figure 16d illustrates the output signals of the neural network in the LTSPICE environment. The considered neural network is also realized in the SIMULINK-MATLAB environment [32]. The SIMULINK schematic of the network is presented in Figure 17 for its visualization and further comments. The input signals are realized by pulse generators. They are forwarded to the MATLAB workspace by the standard blocks “To workspace” for further analysis. The synaptic weights correspond to gain amplifiers, in which the zoomed diagram is given in the right lower corner of the schematic. The adders are realized by standard summing blocks. The logarithmic-sigmoidal and the linear transfer function are realized by “MATLAB function” blocks. The time step is 0.001 ms. The simulation time is 6 ms. The time diagrams, obtained by the function “plot” after finishing the simulation on the MATLAB-SIMULINK environment, are presented in Figure 18. The input signals x_1 – x_5 are represented in Figure 18a. The signals after the adders and after the transfer functions of the hidden layer are illustrated in Figure 18b. Figure 18c presents the output signals of the neural network. The considered neural network is also simulated in MATLAB, using the neural network toolbox [32], for conducting a comparison of the results, derived in SIMULINK and in LTSPICE [33].

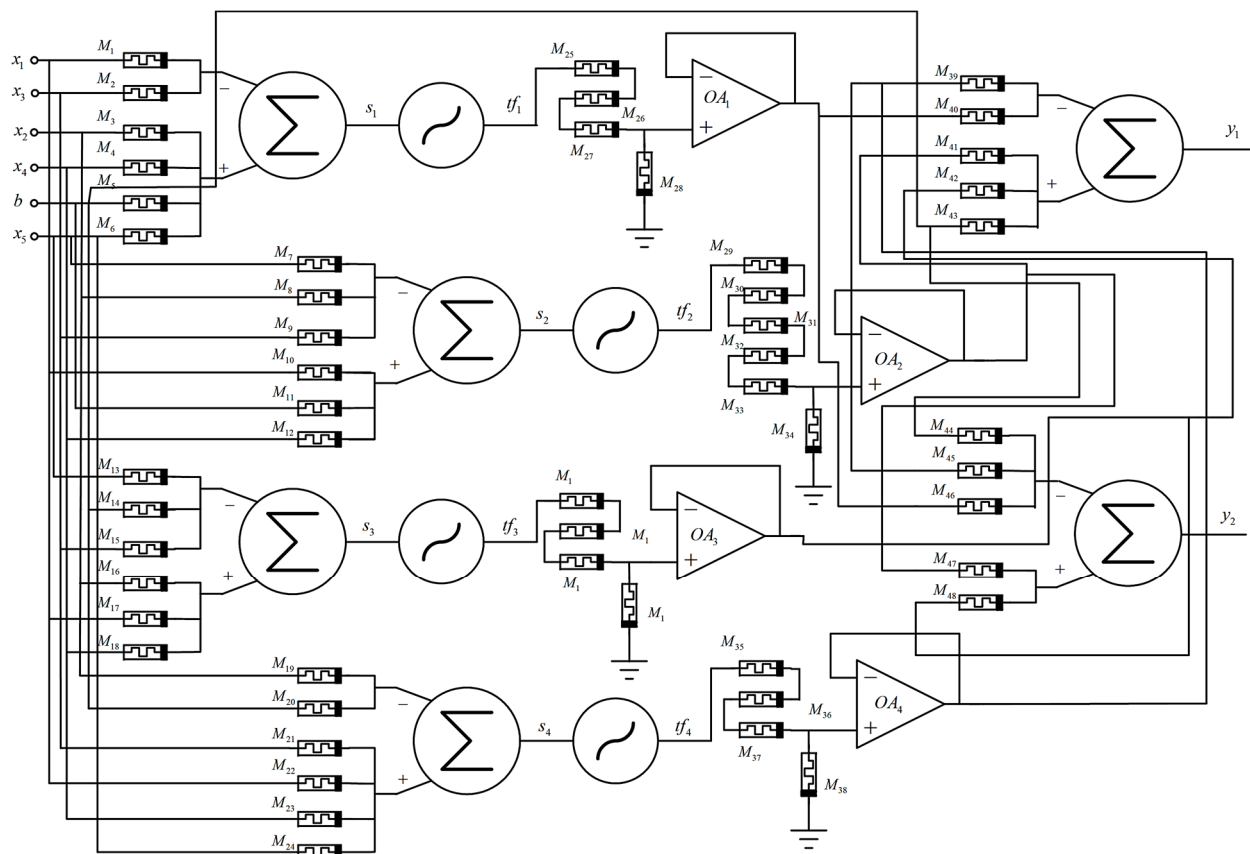


Figure 14. Principal schematic of the considered memristor-based NN under analysis.

Table 8 compares the signals' levels, derived in LTSPICE and MATLAB-Simulink [32]. Identical results are derived from the conducted analyses and simulations. This comparison confirms the capability of the considered memristor-based neural network and the activation function to properly operate with pulse input signals with levels lower than the activation threshold of the applied metal-oxide memristors [36].

Table 8. Comparison between the results, obtained after the analyses of the memristor-based neural net.

Signals	S_1	S_2	S_3	S_4	tf_1	tf_2	tf_3	tf_4	Y_1	Y_2
	mV	mV	mV	mV	mV	mV	mV	mV	mV	mV
Simulink	293	398	−157	−85	769	836	344	414	134	−104
LTSPICE	293	398	−157	−85	789	887	342	413	137	−102
MATLAB	293	398	−157	−85	769	836	344	414	134	−105

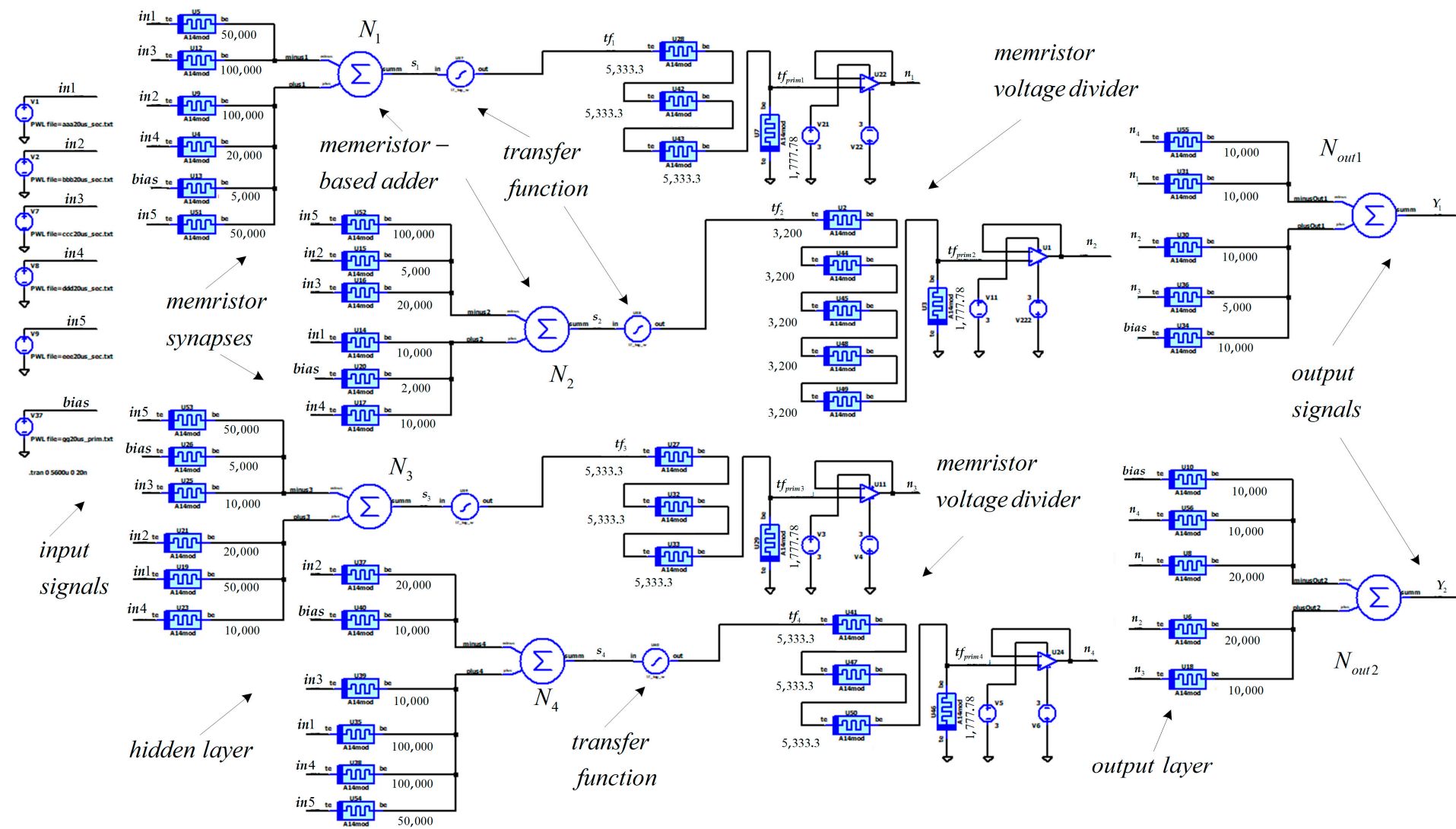


Figure 15. A realization of the considered memristor-based neural network in electronic simulator—LTSPICE, with five input signals—from in_1 to in_5 , four neurons in the hidden layer and two neurons in the output layer, with memristor-based synapses and MOS-transistor based logarithmic-sigmoidal transfer function.

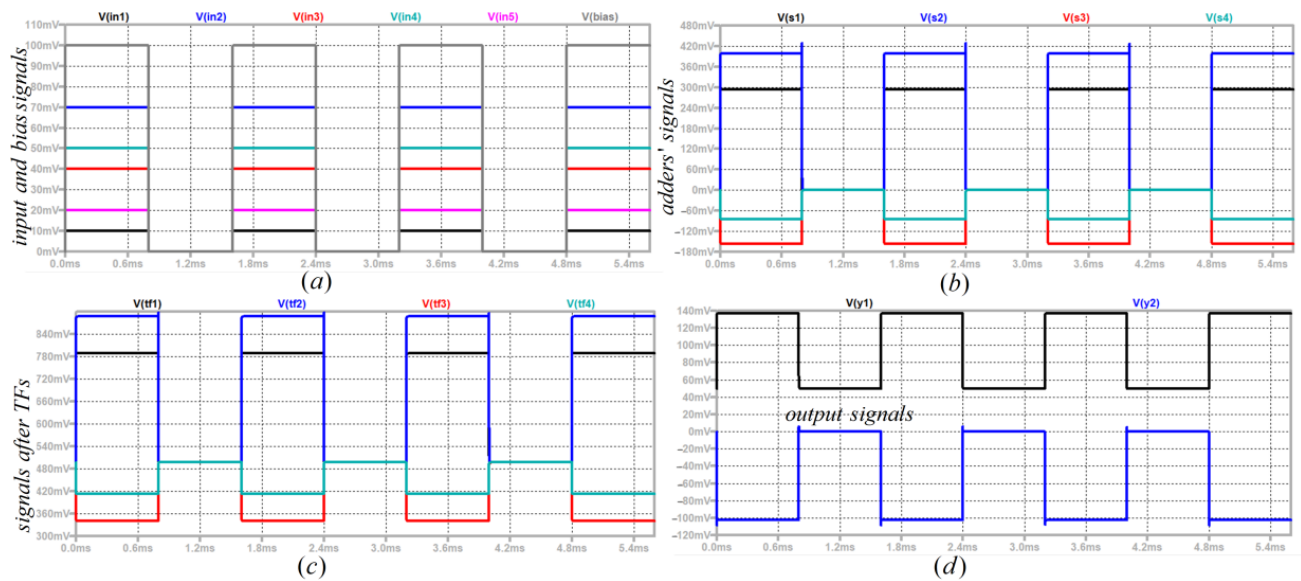


Figure 16. Analysis of considered memristor neural network in LTSPICE; (a) diagrams of input signals v_1 – v_5 and bias signal v_{bias} ; (b) graphs of signals after the adders in hidden layer— v_{s1} , v_{s2} , v_{s3} , and v_{s4} ; (c) diagrams of signals after transfer functions in hidden layer— v_{tf1} , v_{tf2} , v_{tf3} , and v_{tf4} ; (d) diagrams of output signals of neural net— v_{y1} , v_{y2} .

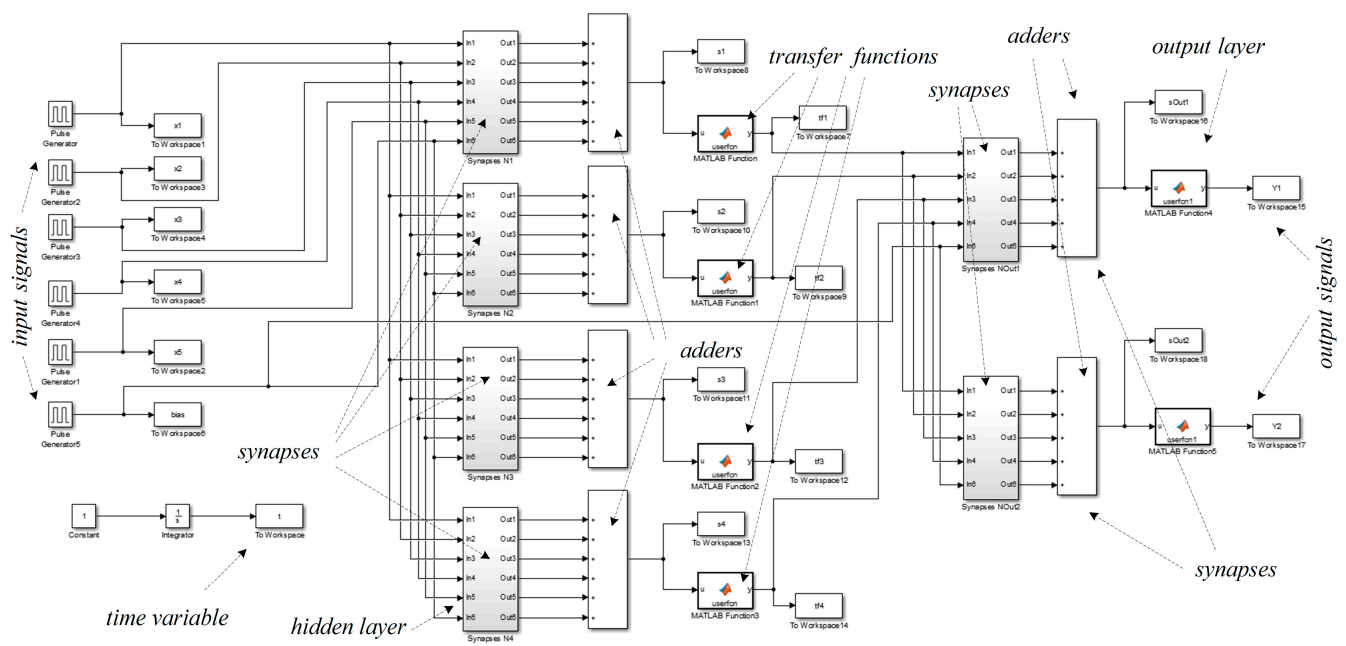


Figure 17. Simulink diagram of the analyzed memristor-based neural network.

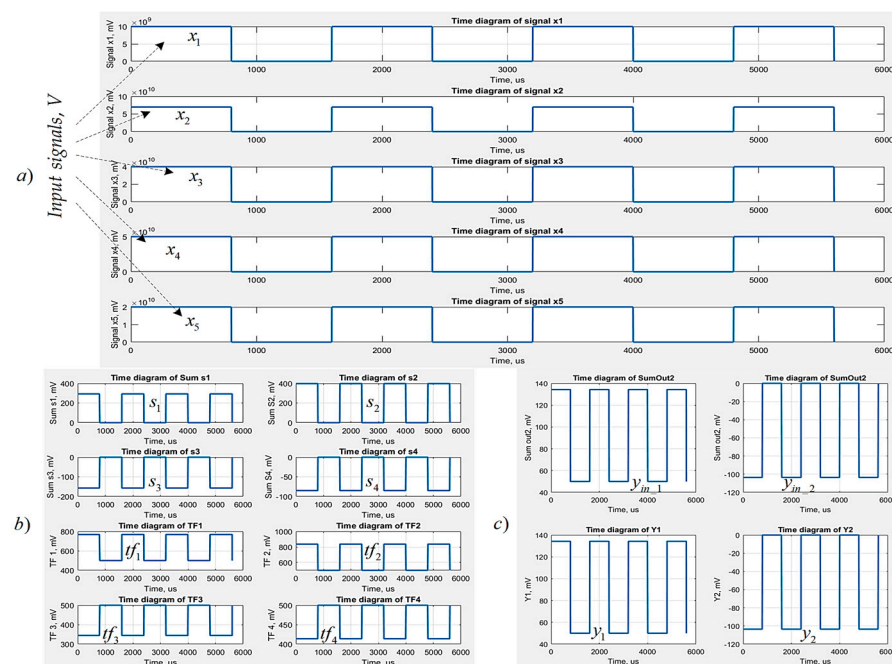


Figure 18. Diagrams, corresponding to neural net in Figure 15; (a) diagrams for input signals; (b) for the hidden layer—output signals of adders and of transfer functions; (c) time diagrams for the output layer.

7. Discussion

Comparing the simulation results of the considered neural network, realized in LT-SPIICE ver. 24.0.9, MATLAB ver. 9 (R2016a), and SIMULINK ver. 8.7 (R2016a) software, a very good matching of the signals after the adders, the transfer functions, and the outputs is established. The maximal error between the output signal's levels is less than 3%. The synapses of the neural network are implemented with single memristors and operational amplifiers for summation, utilizing their inverting inputs. The suggested synapses implement both positive and negative weights. The logarithmic-sigmoidal activation function utilized in the research relies on a combination of a metal-oxide memristor and two MOS transistors, connected in anti-parallel. The analyses demonstrate a strong correlation between the obtained results, confirming the accurate functionality of the suggested memristor-based neural network. Memristor-based neural nets are more disposed to chaotic modes, owing to the highly nonlinear behavior of memristors and fluctuations of their parameters, and additional attention to ensure their stability is needed, especially when scaled up to a large number of nodes [38,39]. Additional simulations with different input signals were conducted and the results confirmed the proper operation of the considered memristor-based neural net. Neuromorphic engineering and artificial intelligence are broad fields of science. They include many hardware and software tools, embracing different kinds of neurons, synapses, neural nets, deep neural nets, and many others. In mobile and low-power devices, having increasing application in medicine, remote measurement and control, home security, and education, intelligent devices and systems are very helpful in the present day. Their optimization and minimization are very important. Owing to this, memristors, as promising nano-sized components with memory and switching properties, are potentially applicable in artificial neurons and synaptic connections. Memristor neural nets could be applied for image recognition, audio filtration, control and monitoring of industrial processes, in medical monitoring and clustering, and in many other fields of industries and scientific applications. This work could enrich the field of artificial intelligence and neural networks, giving some possibilities to engineers for design and comparisons, and the utilization of the proposed schemes in novel neuromorphic circuits and devices.

8. Conclusions

The main purpose of this paper—the realization of a complete electronic implementation of a memristor-based neural network in the LTSPICE environment—has been achieved. The related tasks are resolved—a modified logarithmic-sigmoidal transfer function, based on memristors and MOS transistors is realized; a simple adder, founded on memristors and operational amplifiers is realized with a minimal number of electronic elements; a simple multi-layer artificial neural network, using memristors for the synaptic weights of positive and negative signs, employing a minimal number of electronic components in the LTSPICE environment, is generated and analyzed. The analyses and simulations are performed, using MATLAB, SIMULINK, and LTSPICE. We have applied the $A_{14\text{mod}}$ memristor model, which is better than the existing ones (in terms of faster operation and simulation), and helps with faster simulation when the synaptic weights are tuning, during the training of the neural network. The main advantage of the considered memristor-based artificial neural network is its simple realization with the use of very low-power and nano-sized metal-oxide memristor elements, with a very good compatibility with the present day CMOS high-density integrated electronic chips and circuits. Memristors, as novel two-terminal components, with good memory and switching properties, low power consumption and nano-dimensions, are passive elements. In some cases, for processes such as signal amplification or inversion, they are not directly applicable. The compatibility of memristors with CMOS integrated circuits allows engineers and designers to construct novel circuits with applications in neuromorphic circuitry and artificial intelligence, which is an important step in the field. This work introduces an opportunity for readers to explore implementations of neural network circuits, based on memristors. It encourages the analysis and comparison of various realizations of artificial neural networks, using SPICE simulators, together with practical measurements. The future work in this field will be related to the exploration of the synthesis and analysis of more complex multi-layer memristor-based neural networks for artificial intelligence. Additionally, it will aim to explore the practical implementation of such neural networks and compare it with other similar proposed implementations.

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

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Wang, S.; Song, L.; Chen, W.; Wang, G.; Hao, E.; Li, C.; Hu, Y.; Pan, Y.; Nathan, A.; Hu, G.; et al. Memristor-Based Intelligent Human-Like Neural Computing. *Adv. Electron. Mater.* **2023**, *9*, 2200877. [\[CrossRef\]](#)
2. Sah, M.P.; Kim, H.; Chua, L.O. Brains Are Made of Memristors. *IEEE Circ. Syst. Mag.* **2014**, *14*, 12–36. [\[CrossRef\]](#)
3. Aggarwal, C. *Neural Networks and Deep Learning*; Springer International Publishing AG: Berlin, Germany, 2018; ISBN 978-3-319-94463-0.
4. Krestinskaya, O.; Salama, K.N.; James, A.P. Learning in Memristive Neural Network Architectures Using Analog Backpropagation Circuits. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2019**, *66*, 719–732. [\[CrossRef\]](#)
5. Bradley WM, D.; Mears, R.J. Backpropagation learning using positive weights for multilayer optoelectronic neural networks. In Proceedings of the Conference Proceedings LEOS'96 9th Annual Meeting IEEE Lasers and Electro-Optics Society, Boston, MA, USA, 18–21 November 1996; Volume 1, pp. 294–295.
6. Parisien, C.; Anderson, C.H.; Eliasmith, C. Solving the problem of negative synaptic weights in cortical models. *Neural Comput.* **2008**, *20*, 1473–1494. [\[CrossRef\]](#)
7. Xu, W.; Wang, J.; Yan, X. Advances in memristor-based neural networks. *Front. Nanotechnol.* **2021**, *3*, 645995. [\[CrossRef\]](#)

8. Hong, Q.; Zhao, L.; Wang, X. Novel circuit designs of memristor synapse and neuron. *Neurocomputing* **2019**, *330*, 11–16. [CrossRef]
9. Sah, M.P.; Yang, C.; Kim, H.; Chua, L. A voltage mode memristor bridge synaptic circuit with memristor emulators. *Sensors* **2012**, *12*, 3587–3604. [CrossRef] [PubMed]
10. Dai, Y.; Feng, Z.; Wu, Z. A Novel Window Function Enables Memristor Model With High Efficiency Spiking Neural Network Applications. *IEEE Trans. Electron Devices* **2022**, *69*, 3667–3674. [CrossRef]
11. Wen, S.; Xie, X.; Yan, Z.; Huang, T.; Zeng, Z. General memristor with applications in multilayer neural networks. *Neural Netw.* **2018**, *103*, 142–149. [CrossRef]
12. Zhang, Y.; Wang, X.; Friedman, E.G. Memristor-Based Circuit Design for Multilayer Neural Networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2018**, *65*, 677–686. [CrossRef]
13. Su, B.; Cai, J.; Wang, Z.; Chu, J.; Zhang, Y. A π -Type Memristor Synapse and Neuron With Structural Plasticity. *Front. Phys.* **2022**, *9*, 798971. [CrossRef]
14. Wang, Z.; Joshi, S.; Savel'ev, S.; Song, W.; Midya, R.; Li, Y.; Rao, M.; Yan, P.; Asapu, S.; Zhuo, Y.; et al. Fully memristive neural networks for pattern classification with unsupervised learning. *Nat. Electron.* **2018**, *1*, 137–145. [CrossRef]
15. Zhang, X.; Wang, X.; Ge, Z.; Li, Z.; Wu, M.; Borah, S. A Novel Memristive Neural Network Circuit and Its Application in Character Recognition. *Micromachines* **2022**, *13*, 2074. [CrossRef]
16. Wang, Y.; Xu, H.; Wang, W.; Zhang, X.; Wu, Z.; Gu, R.; Li, Q.; Liu, Q. A Configurable Artificial Neuron Based on a Threshold-Tunable TiN/NbO_x/Pt Memristor. *IEEE Electr. Device Lett.* **2022**, *43*, 631–634. [CrossRef]
17. Li, B.; Shi, G. A CMOS rectified linear unit operating in weak inversion for memristive neuromorphic circuits. *Integration* **2022**, *87*, 24–28. [CrossRef]
18. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, S. The missing memristor found. *Nature* **2008**, *453*, 80–83. [CrossRef] [PubMed]
19. 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]
20. James, A. *Memristors—Circuits and Applications of Memristor Devices*; IntechOpen: London, UK, 2019; p. 132. [CrossRef]
21. Lehtonen, E.; Laiho, M. CNN using memristors for neighborhood connections. In Proceedings of the 2010 12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA 2010), Berkeley, CA, USA, 3–5 February 2010; pp. 1–4.
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. Ascoli, A.; Corinto, F.; Senger, V.; Tetzlaff, R. Memristor model comparison. *IEEE Circ. Syst. Mag.* **2013**, *13*, 89–105. [CrossRef]
25. Mohammad, B.; Jaoude, M.A.; Kumar, V.; Al Homouz, D.M.; Abu Nahla, H.; Al-Qutayri, M.; Christoforou, N. State of the art of metal oxide memristor devices. *Nanotechnol. Rev.* **2016**, *5*, 311–329. [CrossRef]
26. Dautovic, S.; Samardzic, N.; Juhas, A.; Ascoli, A.; Tetzlaff, R. Simscape and LTspice models of HP ideal generic memristor based on finite closed form solution for window functions. In Proceedings of the 2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Dubai, United Arab Emirates, 28 November–1 December 2021; pp. 1–6. [CrossRef]
27. Zafar, M.; Awais, M.; Shehzad, M. Computationally efficient memristor model based on Hann window function. *Microelectron. J.* **2022**, *125*, 105476. [CrossRef]
28. Solovyeva, E.B.; Azarov, V.A. Comparative Analysis of Memristor Models with a Window Function Described in LTspice. In Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), Moscow, Russia, 26–29 January 2021; pp. 1097–1101.
29. Ascoli, A.; Weiher, M.; Herzig, M.; Slesazek, S.; Mikolajick, T.; Tetzlaff, R. Graph Coloring via Locally-Active Memristor Oscillatory Networks. *J. Low Power Electr. Appl.* **2022**, *12*, 22. [CrossRef]
30. Mladenov, V. *Advanced Memristor Modeling—Memristor Circuits and Networks*; MDPI: Basel, Switzerland, 2019; p. 172, ISBN 978-3-03897-104-7. [CrossRef]
31. Mladenov, V. A Unified and Open LTSPICE Memristor Model Library. *Electronics* **2021**, *10*, 1594. [CrossRef]
32. Kim, P. *Matlab Deep Learning. With Machine Learning, Neural Networks and Artificial Intelligence*; APress: Berkeley, CA, USA, 2017; p. 151, ISBN 978-1-4842-2845-6. [CrossRef]
33. May, C. *Passive Circuit Analysis with LTspice®—An Interactive Approach*; Springer: Berlin/Heidelberg, Germany, 2020; p. 763, ISBN 978-3-030-38304-6.
34. Mladenov, V.; Tsenov, G.; Kirilov, S. Memristor-Based Neural Network Implementation with Adjustable Synaptic Weights in LTSPICE. In Proceedings of the 2023 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 5–7 October 2023; pp. 403–408. [CrossRef]
35. Mladenov, V.; Tsenov, G.; Kirilov, S. LTSPICE Memristor Neuron with MOS Transistor-Based Logarithmic-Sigmoidal Activation Function. In Proceedings of the 18th IEEE International Workshop on Cellular Nanoscale Networks and Their Applications and the 8th Memristor and Memristive Symposium, Xanthi, Greece, 28–30 September 2023. Available online: <https://cnna.duth.gr/> (accessed on 27 November 2023).
36. Campbell, K.A. Self-directed channel memristor for high temperature operation. *Microelectron. J.* **2017**, *59*, 10–14. [CrossRef]
37. Yuan, R.; Tiw, P.J.; Cai, L.; Yang, Z.; Liu, C.; Zhang, T.; Ge, C.; Huang, R.; Yang, Y. A neuromorphic physiological signal processing system based on VO₂ memristor for next-generation human-machine interface. *Nat. Com.* **2023**, *14*, 3695. [CrossRef]

38. Marco, M.; Forti, M.; Moretti, R.; Pancioni, L.; Tesi, A. Complete Stability of Neural Networks With Extended Memristors. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–15. [[CrossRef](#)]
39. Di Marco, M.; Forti, M.; Pancioni, L. New Conditions for Global Asymptotic Stability of Memristor Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, 29, 1822–1834. [[CrossRef](#)]

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.