Article

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

Valeri Mladenov and Stoyan Kirilov *

Posted Date: 16 January 2024

doi: 10.20944/preprints202401.1245.v1

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

Article

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

Valeri Mladenov and Stoyan Kirilov *

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

* Correspondence: s_kirilov@tu-sofia.bg; Tel.: +359-877-74-67-32

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 analog circuit model of 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 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

1. Introduction

Artificial neural networks have inspiration from electro-chemical communication between neural cells in 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 (TiO_2), is created in Hewlett-Packard scientific and research labs by Stanley Williams and his collaborators [21,23,24]. Different metal oxides, such as HfO_2, Ta_2O_5, Nb_2O_5, and many others substances are utilized for realizations 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 a

 doi:10.20944/preprints202401.1245.v1

good compatibility to 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 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 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 used here LTSPICE simulator is a preferred one, owing to its user-friendly interface, free license, an 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 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 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 LTSPICE simulator [34]. Subsequently a single neuron with activation function implemented with memristors and MOS transistors is also realized and proposed to the readers [35].

The goal of this paper is realization of a whole electronic implementation of a memristor neural network in LTSPICE environment. To achieve this purpose the following tasks are placed: realization of a new electronic circuit based on memristor and MOS transistors, realizing a logarithmic-sigmoidal transfer function; realization of a simple adder based on memristors and operational amplifiers with minimal number of elements; 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 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 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 synthesis and analysis of more complex multi-layer memristor-based neural networks, as well as their practical implementation and comparison with other proposed realizations.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

The 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 considered simple and fast-operating LTSPICE memristor neuron is discussed in Section 4. The applied activation function based on memristors and MOS transistors is presented 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 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

The memristors are passive one-port electronic elements with memory and switching properties. In the scientific reports, a lot of memristors based on different materials, 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, 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 of 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 electric field is quite different, in comparison to those of the classical electronic passive elements – the *resistor*, *inductor* and *capacitor*. Owing to its specifics modeling of memristors is mandatory for its 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 these of Williams-Strukov [18], Joglekar [22] and Biolek [23] is represented by the next general set of equations (1) [26]:

$$\begin{cases} i = v \cdot M(x)^{-1} \\ \dfrac{dx}{dt} = k \cdot f(x) \cdot i \end{cases} \tag{1}$$

where i and v are the 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, 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 limitation of the state variable in the interval from 0 to 1 and for introducing the boundary effects at a hard-switching mode [26]. Strukov-Williams memristor model [18] is described by the next set of equations (2):

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

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

where R_{ON} and R_{OFF} are minimal and maximal values of the memristance, and $f_{SW}(x)$ is a simple and low-order polynomial window function proposed by Williams and Strukov [18]. This classical memristor model is a very simple one, and owing to the reduced number of the elementary mathematical operations it has a very high operating speed []. Owing to the linear dependence between the time derivative of the state variable 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 is related to the so called *terminal state problems* [20,31]. This memristor model has a comparatively low accuracy in simulation of complex and non-symmetrical experimental current-voltage characteristics of metal-oxide memristors. Joglekar memristor model is one of the widely-used metal-oxide memristor models [22] and it is presented by the following system (3):

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

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 the current, 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 used polynomial window function is adjustable one and ensures different slope, according to the parameter p and the memristor state variable.

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

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

where $f_B(x,i)$ is a window function, proposed by Biolek, and p is a positive integer [23]. This classic model is 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 modeling of experimental current-voltage relationships.

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

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

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 []. Due to the highly nonlinear relation between time derivative of state variable x and the memristor voltage, this

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

standard model correctly expresses the different behavior of metal-oxide memristors for signals, higher or lower than 1 V [21]. Usually, in this memristor model, the classical Biolek window is applied [24,31]. This model has a high accuracy.

The applied in the present analyses modified memristor model, denoted by *A14mod* [35], is based on Joglekar memristor model and 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 equations (6) [35]:

$$\begin{cases} i = v \cdot \left[R_{ON}x + R_{OFF}(1-x)\right]^{-1} \\ \dfrac{dx}{dt} = k \cdot \sin^2(\pi x) \cdot v^m \cdot stp\left(|v| - v_{thr}\right) \end{cases} \tag{6}$$

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, the applied Hann window function is not related to terminal state problems. If the voltage is lower than the activation threshold v_{thr}, then the time derivative of the state variable is zero and the memristor behaves as a simple and linear resistor with a constant conductance [24,26]. This mode is used for the operation 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 a good accuracy, and it is very appropriate for application in neural networks.

2.3. Fine adjustment and parameter estimation of the applied memristor model [24,31,32]

In this work, we analyse the considered modified model of a memristor based on metal oxides [35] and it is fully allocated by equation (6). This memristor model incorporates multiple parameters that allow its precise and 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 Knowm 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.

MATLAB-SIMULINK environment is utilized to extract the 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 is altering by a small growth [32]. The RMS error between simulated and experimental i-v relations is computed. The other parameters of the model are also changing. 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 on the form of the derived i-v pinched hysteresis loop and especially on 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 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 memristor model's parameters are also obtained, using the least squares method in MATLAB, applying Optimization Toolbox [32]. The parameter estimation process could be summarized in several main steps:

 doi:10.20944/preprints202401.1245.v1

1. *Initialization* - choosing the initial values of model's coefficients, using their values, presented in the original references [36], and the parameters in the modified memristor model (6), after their evaluation in Simulink and MATLAB environment [32];
2. *Determination 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 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 MATLAB-Simulink memristor model are previously sampled, the time step is with a value of 10 μs. 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 – eq. (7) [24,32]:

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

where $N = 1000$ 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 time diagrams of simulated and experimental memristor current after the parameter extraction are presented in Figure 1 (a) for their visual observation and comparison. The normalized values of trajectories of coefficients during the estimation process are presented in Figure 1 (b) to express their alteration in the time domain. The initial values of parameters are chosen to be equal to the presented in [36]. After finishing the tuning process of the considered memristor model, the obtained optimal vales of the parameters are: $k = 3.12.10^6$; $x_0 = 0.0103$; $R_{on} = 3.37$ ohm; $R_{off} = 57.033$ kΩ; $m = 5$; $v_{thr} = 0.1583$ V. After the parameter estimation process the modified memristor model is analyzed in Simulink-MATLAB and the derived results are presented in Figure 2 for their visual comparison. The obtained time diagrams of the applied voltage signal, the corresponding experimental and simulated memristor current and state variable after the parameter extraction process are shown in Figure 2 (a) for their visual expression and comparison. The next Figure 2 (b) represents the corresponding experimental and simulated current-voltage characteristics 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, several main *criteria*, as the simulation time, the operation rate, accuracy and convergence [24,31] are considered in the next Table 1. After the conducted comparison between the memristor models, it could be stated that the applied modified memristor model *A14mod* has a little bit lower accuracy than Lehtonen-Laiho model, but it has lower complexity, good switching properties and high operating frequency [35]. According to the standard Joglekar and Biolek memristor models, the modified model *A14mod* has better characteristics and properties with a little bit higher complexity and simulation time.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

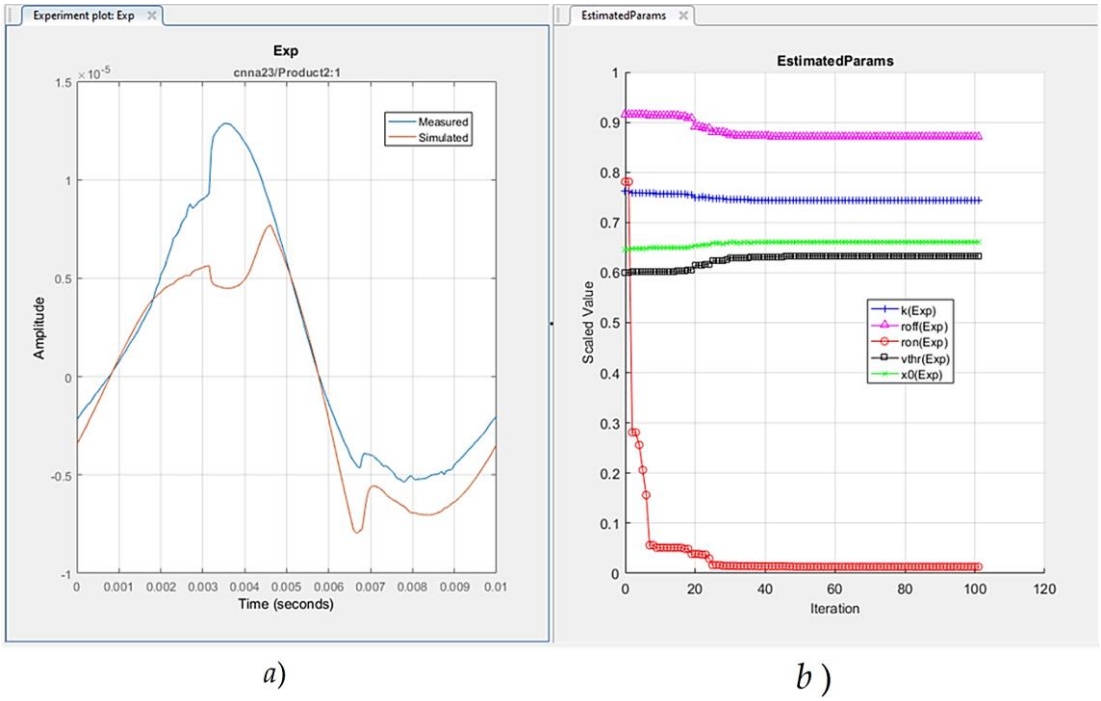

Figure 1. a) Time diagrams of the experimental and the simulated current after the parameter estimation process for the modified model *A14mod;* b) Parameter trajectories during the estimation process.

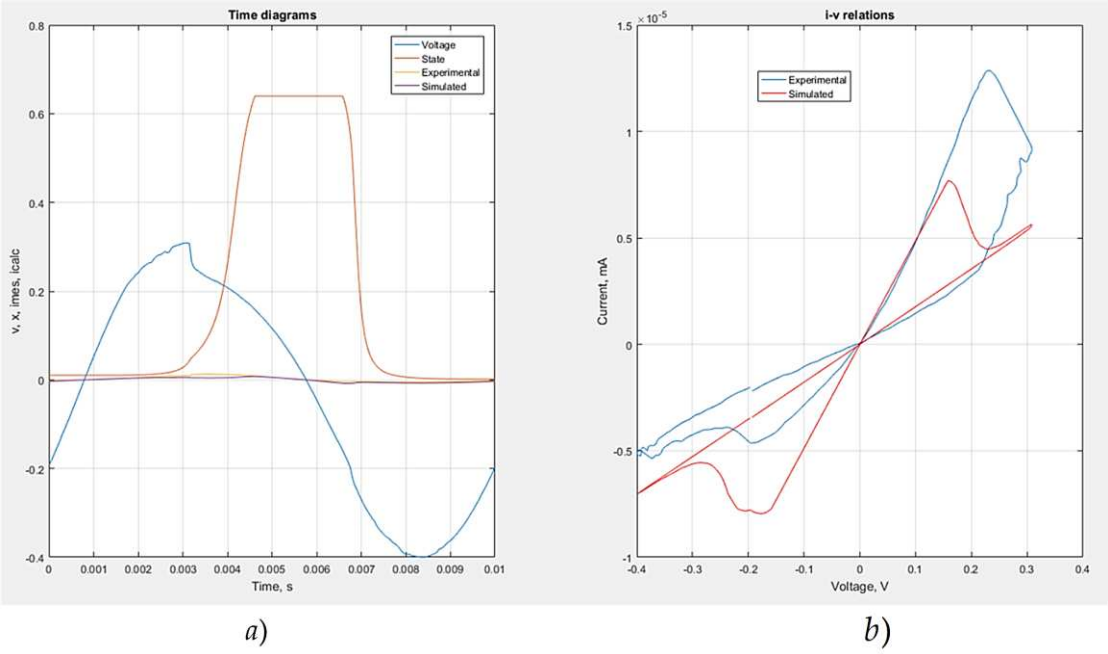

Figure 2. a) Time diagrams of the experimental and the simulated current after the parameter estimation process, and the state variable of the modified memristor model *A14mod;* b) Experimental and simulated current-voltage relationships after the parameter estimation process.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

Table 1. A short comparison of the considered metal-oxide memristor models – Joglekar, Biolek, Lehtonen-Laiho, and the applied modified memristor model *A14mod* [35].

Model	Joglekar [22]	Biolek [23]	Lehtonen-Laiho [21]	A14mod [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

The type of the applied memristor model is important for training the networks and adjustment of the synaptic weights of memristor-based neural networks by short voltage pulses. In the next section the generation and analysis of LTSPICE memristor model, corresponding to the considered modified memristor model, its main characteristics and properties are discussed.

3. LTSPICE Library Model [35] and analysis of the applied Memristor model

The main idea, utilized for construction of the LTSPICE [33] memristor library model, is to build a simple and fast-operating equivalent circuit, corresponding to (6). Before realization of the current-voltage characteristics, the derivation of the state variable x is needed. The memristor state variable is obtained by integration of the state differential equation with respect to time. This process is ensured by a capacitor; its current is set to be proportional to the time derivative of the state variable x [26,31]. The current of the capacitor is generated by a voltage-controlled current source, according to the left-hand side of the state equation in (6). The initial voltage of the integrating capacitor is set to be equal to the starting value of the memristor state variable. The voltage across the integrating capacitor is directly proportional to the state variable in each moment in the time domain. To avoid rapid changes of the voltage across the capacitor, which could lead to convergence issues, a high-valued resistor could be attached in parallel to the capacitor [31,36]. Additional voltage-controlled current source is generating the memristor current, according to the right-hand side of the first equation in (6). The initial row of the LTSPICE code displays the memristor model identifier, indicated as *A14mod*. 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, $m0$ and v_{thr} are presented in the next row. The coefficient m0 represents the initial value of the memristance. The integrating capacitor is connected between the terminal *Y* and the ground electrode. Its capacitance is 1000 mF. The initial voltage of integrating element C_1, denoted as ".IC", 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_{ad} with a value of 100 GΩ, connected in parallel to C_1 and utilized for a partial avoidance of convergence problems. The dependent source G_2 presented in the sixth row expresses the time derivative of the 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 realization of the state equation, containing the memristor activation threshold v_{thr}. The code, presented below is finished with the command "ends" [26,33,35].

```
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)))*
      (stpp((vthr-abs(V(te,be))),mm)))}
7   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
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

The parameter *mm* determines the sharpness of the step-like function *stpp*. For a quick and easier tuning of the initial value of the memristance m_0, the state variable of the memristor in the initial moment is expressed as a function of m_0 in the fourth row. After selecting the memristor model *A14mod* in LTSPICE and placing 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 left mouse button click, 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-LTSpise [31]. 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 3 a), 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 kOhms. 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 LTSPICE environment, they are presented in Figure 3 b), 3 c) and 3 d) 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].

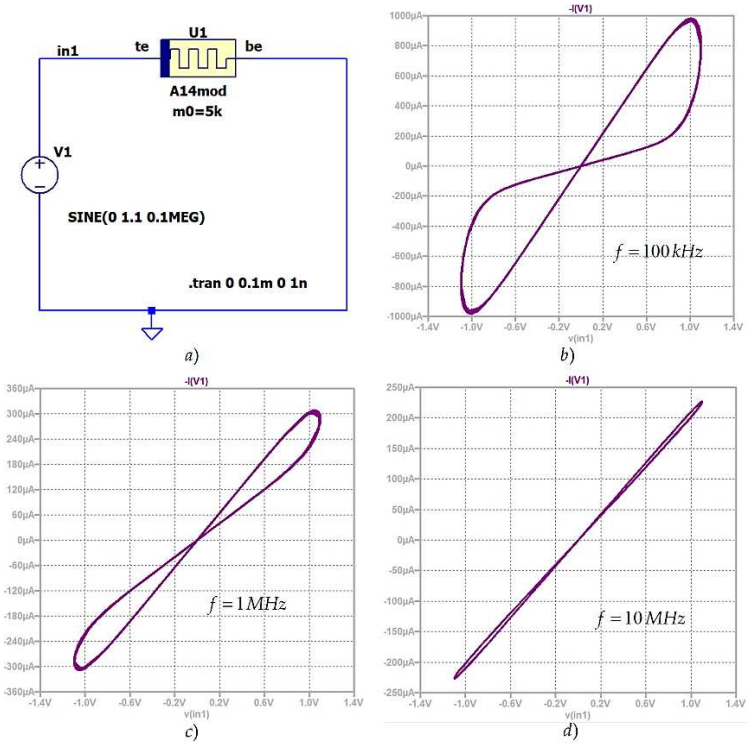

Figure 3. a) A principal schematic of a memristor model *A14mod*, included in a simple electronic circuit for analysis at sinusoidal mode, supplied by a voltage source with an amplitude of 1.1 V and different frequencies; b) Current-voltage relation of the model *A14mod* at 100 kHz; c) *i-v* characteristic of model *A14mod* at 1 MHz; d) current-voltage characteristic of the model, obtained at 10 MHz.

Memristors, participating as synaptic weights, are tuned during the neural network training by positive or negative voltage pulses with a level, which absolute value 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 *A14mod*, is shown in Figure 4 a) for a visual expression of its structure and operation. The voltage source generates a sequence of rectangular and positive pulses with a duration of 1 ns and a

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

duty cycle of 50 %. The amplitude of the voltage is 1.1 V. Figure 4 b) represents the time diagrams of the pulse voltage signal and the corresponding change of the memristance M in LTSPICE. In this case the initial value of the memristance is M_1 = 10 kOhm, which corresponds to a synaptic weight of w_1 = 1. The final value of M_2 = 9 kOhm corresponds to a synaptic weight of w_2=1.1111. In this case, the change of memristance with ΔM = 1 kOhm corresponds to alteration of the synaptic weight with Δw = 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 4 c) represents the results, derived by a simulation in MATLAB, which have a very good coincidence to Figure 4 b) and confirms the tuning process.

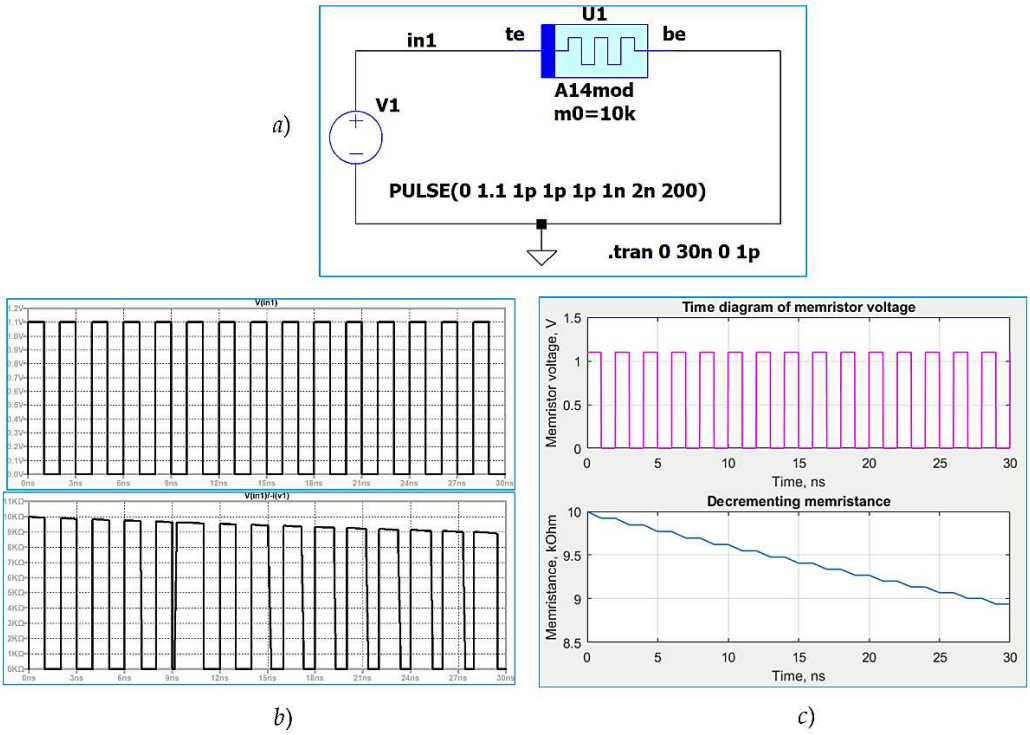

Figure 4. Memristance adjustment by positive voltage pulses; a) A simple electronic circuit with a memristor and a pulse voltage source; b) time diagrams of the pulse voltage signal with amplitude of 1.1 V and a frequency of 500 MHz, and pulse duration of 1 ns; c) time diagram of the memristance, corresponding to the applied pulses.

This result will be also mathematically confirmed by integration of the state differential equation of system (6), modified and represented here by (8):

$$\frac{dx}{dt} = k \cdot \sin^2(\pi x) \cdot v^m \qquad (8)$$

In eq. (8), because the used voltage pulses 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 to another:

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

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

$$x = \frac{M - R_{OFF}}{R_{ON} - R_{OFF}} \qquad (10)$$

Equation (9) is integrated between x_1 = 0.8251, corresponding to M_1 = 10 kOhm, and x_2 = 0.8427, related to M_2 = 9 kOhm, using (10), and the following eq. (11) is obtained:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

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

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

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

The obtained time for synaptic weight adjustment corresponds to the derived simulation result and confirms the proper tuning the memristance by pulses. Another simulation is conducted with negative voltage pulses and the same circuit shown in Figure 4 a) is used. The respective time diagram of the applied negative pulses and the corresponding memristance alteration in LTSPICE is presented in Figure 5 a). Figure 5 b) represents the results, derived from analysis in MATLAB, which is in a good agreement with Figure 5 b) and confirms the correctness of the adjustment process. The analysis starts with initial value of the memristance, equal to M_1 = 5 kOhms, related to a synaptic weight of w_1 = 2 and a state variable of x_1 = 0.9129. The final value of the memristance M_2 is 6 kOhms, corresponding to a synaptic weight with a value of w_2 = 1.6667 and a state variable x_2 = 0.8954. The change of the memristance with 1 kOhm in this case corresponds to alteration of synaptic weight with Δw = 1.6667-2= - 0.3333. The effective duration of the pulse is about 79ns/2 = 39.5 ns.

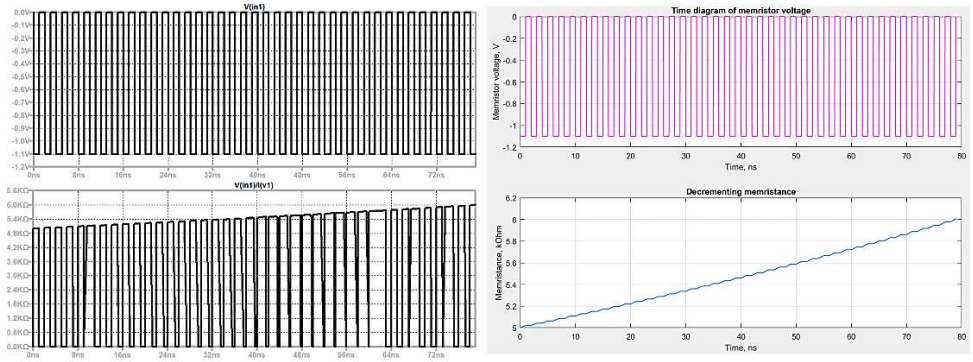

Figure 5. Memristance adjustment by negative voltage pulses; a) time diagram of the pulse voltage signal with a level of -1.1 V and a frequency of 500 MHz, and pulse duration of 1 ns; b) time diagram of the memristance, corresponding to the applied voltage pulses.

The analytical confirmation of this result is as follows – formula (13):

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

For the future practical implementation of memristor-based synapses, the tuning process will 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 of the memristance and the corresponding synaptic weight, the applied modified memristor model *A14mod* ensures a rapid alteration of the state variable and the memristance at applied voltage pulses. Its simple structure and low simulation time ensures the generation and analysis of complex neural networks with a high number of memristors in LTSPICE environment. Additional comparison of the applied memristor models, according to simulation time is conducted for two cases – decreasing and increasing of memristance, according to the analyses shown in Figure 4 and Figure 5. The results presented in Table 2 express the faster operation of the applied modified model, according to Lehtonen-Laiho model. The simulations were conducted on a computer with 8 GB RAM, i5 Intel

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

processor, and Microsoft Windows 10 Pro operating system. According to Joglekar and Biolek models, the modified model *A14mod* has comparable simulation time and better switching properties.

Table 2. Comparison of the applied memristor models, according to simulation time.

Case	Models	Joglekar [22]	Biolek [23]	Lehtonen-Laiho [21]	A14mod [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** 6 a) 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 of the output signal [35]. The dependence between the output and input voltage signals is very similar to a logarithmic-sigmoidal function, presented by the following expression (14):

$$v_{out} = \frac{1}{1+\exp(-k \cdot v_{in})} \tag{14}$$

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

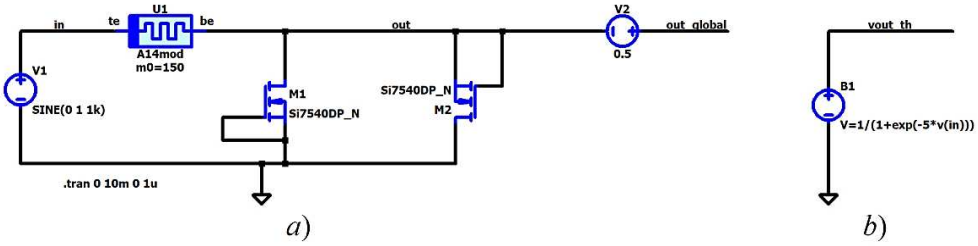

Figure 6. a) A logarithmic-sigmoidal transfer function, realized by a metal-oxide memristor and two MOS transistors, connected in anti-parallel; b) A transfer function, realized by a voltage-controlled voltage source.

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 [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 $k = 5$. In the next section, an artificial neuron, based on memristors is discussed.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

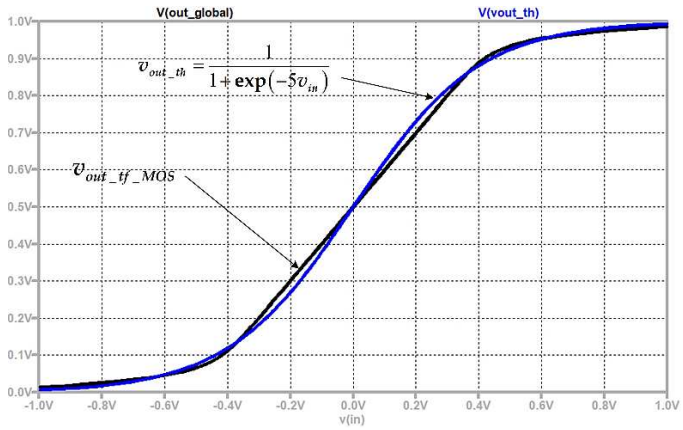

Figure 7. A comparison of the suggested transfer function, realized by MOS transistors and a memristor – $v_{out\_tf\_MOS}$ and the theoretical output signal, realized by a voltage-controlled source $v_{out\_th} = 1/(1+\exp(-5*v_{in}))$.

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 explanations and discussion [2,3]. The input signals of the neuron are represented as x_1, x_2, … and x_N. The signals are directed to synapses based on memristors, with corresponding weights denoted as w_1, w_2, ..., and w_N. A single memristor is employed for each synapse, and its resistance determines the corresponding synaptic weight. The change of 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.

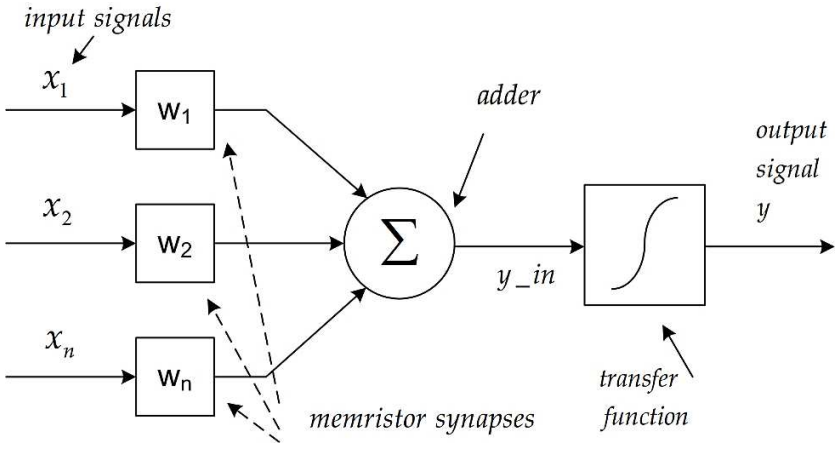

Figure 8. A block diagram of artificial neuron with memristor-based synapses.

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

description of its structure and operation. 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, 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 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 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 kOhms.

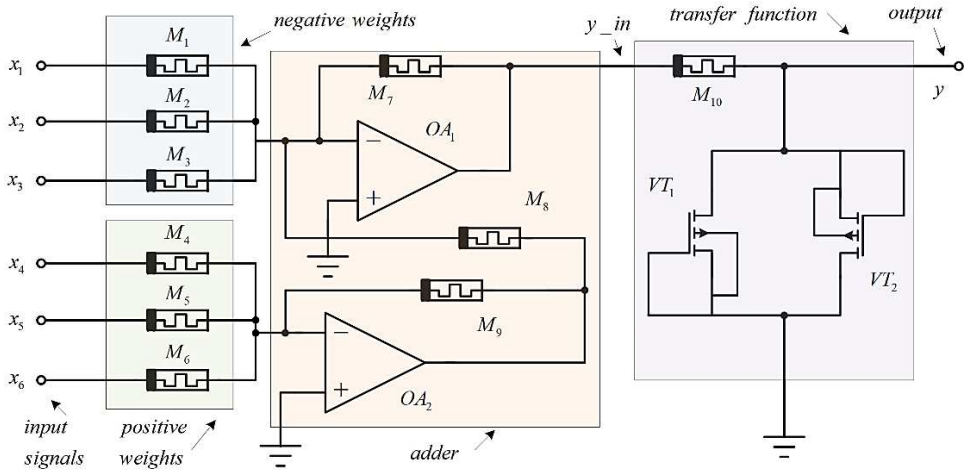

Figure 9. A principal schematic of metal-oxide memristor-based neuron with negative and positive synaptic weights.

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 $M_7 = M_8 = M_9 = 10$ kΩ, 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]:

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

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

Figure 10 illustrates the implementation of the considered memristor neuron in LTSPICE environment, providing further details and discussions. It is associated with the electronic circuit

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

presented in **Figure** 11. The presented pulse voltage sources V_1, V_2, V_7, V_8, V_9 and V_{37} are corresponding 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 *minus* and *plus*, respectively [35]. The output electrode of the adder is denoted by "*summ*". The structure of the log-sigmoidal transfer function corresponds to **Figure** 6 a). A detailed realization of the adder in LTSPICE simulator is presented in **Figure** 11 for its description and further explanations. 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].

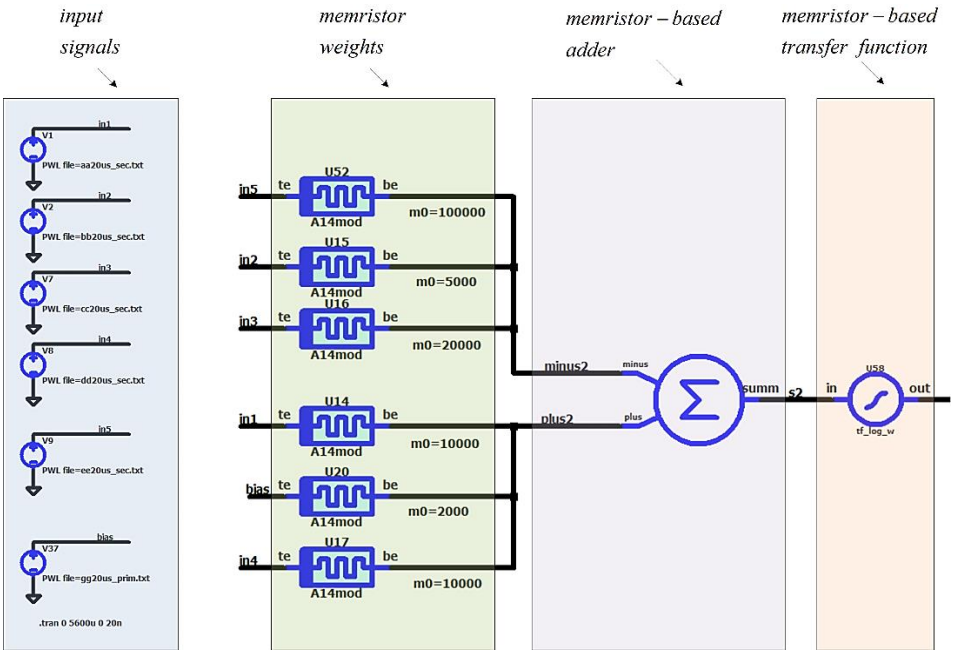

Figure 10. LTSPICE realization of a model of artificial neuron with memristor synapses and op-amps, included in the adder.

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):

$$n = ceil\left(\frac{|v_{summ}|}{v_{thr}}\right) \tag{18}$$

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], which voltage should not exceed the activation threshold v_{thr}.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

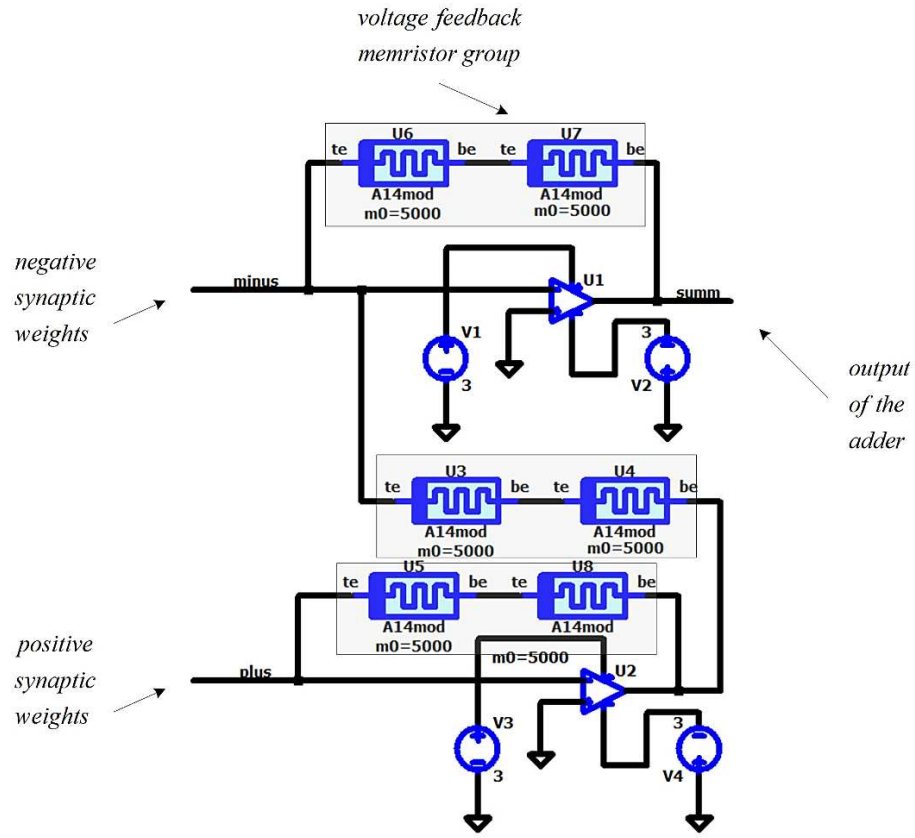

Figure 11. A realization of the considered memristor-based adder in LTSPICE.

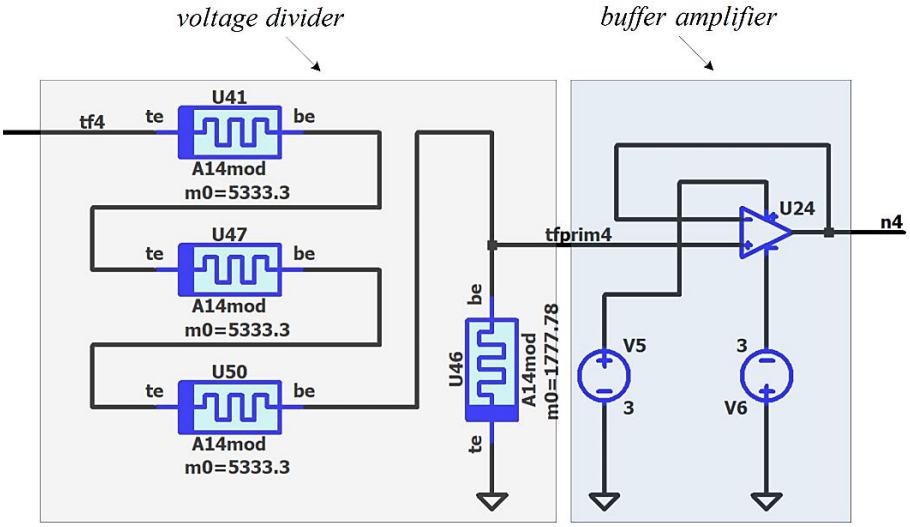

Figure 12. A voltage divider with memristor, connected to a buffer amplifier in LTSPICE environment.

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

$$n = ceil\left(\frac{\left|v_{tfprim4}\right|}{v_{thr}}\right) \tag{19}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

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 non-inverting operational amplifier.

6. Memristor-Based NN 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 contain 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 type "*purelin*" []. 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 MATLAB environment, using feed-forward and back-error propagation algorithm, for 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 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}.

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

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.

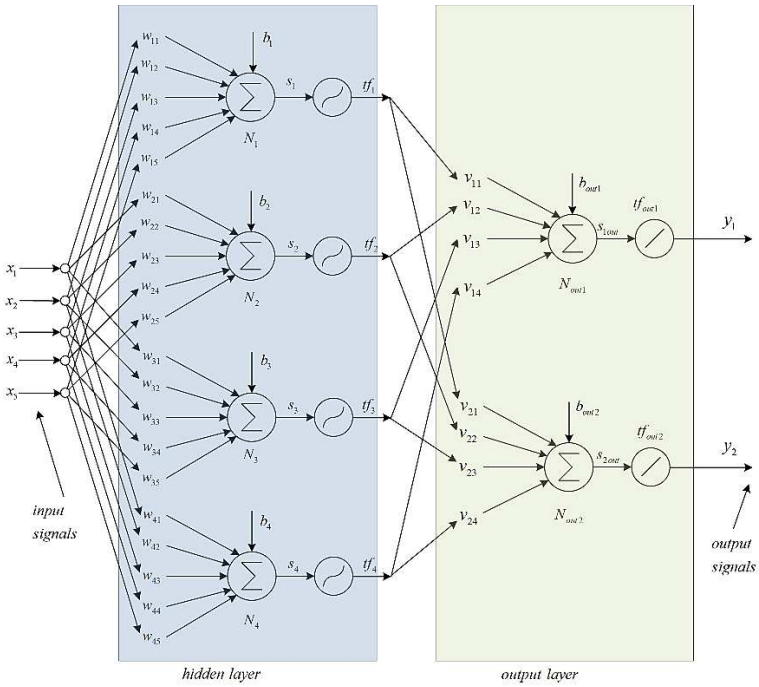

Figure 13. Block diagram of the considered memristor-based neural network, containing five input nodes, four neurons in the hidden layer and two neurons in the output layer.

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

	Input memristances, kOhms					
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 synaptic weights of the synapses in the output layer are presented in Table 6.

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

The memristances, corresponding to the synaptic weights for the output layer are represented in Table 7.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

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

	Output layer memristances, kOhms					
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 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 in similar SPICE simulators. The circuits is also available at: https://github.com/mladenovvaleri/Advanced-Memristor-Modeling-in-LTSpise [31]. Figure 16 a) 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 16 b). Figure 16 c) represents the signals after the logarithmic-sigmoidal transfer functions and Figure 16 d) illustrates the output signals of the neural network in LTSPICE environment. The considered neural network is also realized in 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 MATLAB workspace by the standard blocks "*To workspace*" for further analysis. The synaptic weights are corresponding to gain amplifiers, which zoomed diagram is given the right down corner of the schematic. The adders are realized by standard summing blocks. The logarithmic-sigmoidal and the linear transfer function is 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 MATLAB-SIMULINK environment are presented in Figure 18. The input signals $x_1 - x_5$ are represented in Figure 18 a). The signals after the adders and after the transfer functions of the hidden layer are illustrated in Figure 18 b). Figure 18 c) presents the output signals of the neural network.

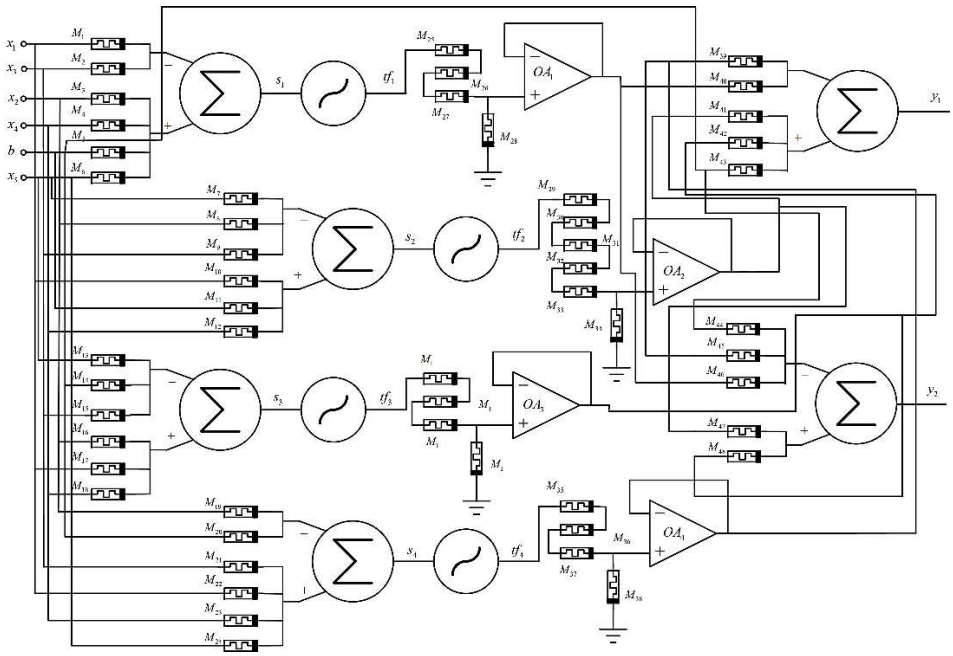

Figure 14. Principal schematic of the considered memristor-based NN.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

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]. 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 activation function to properly operate with pulse input signals with levels lower than the activation threshold of the applied metal-oxide memristors.

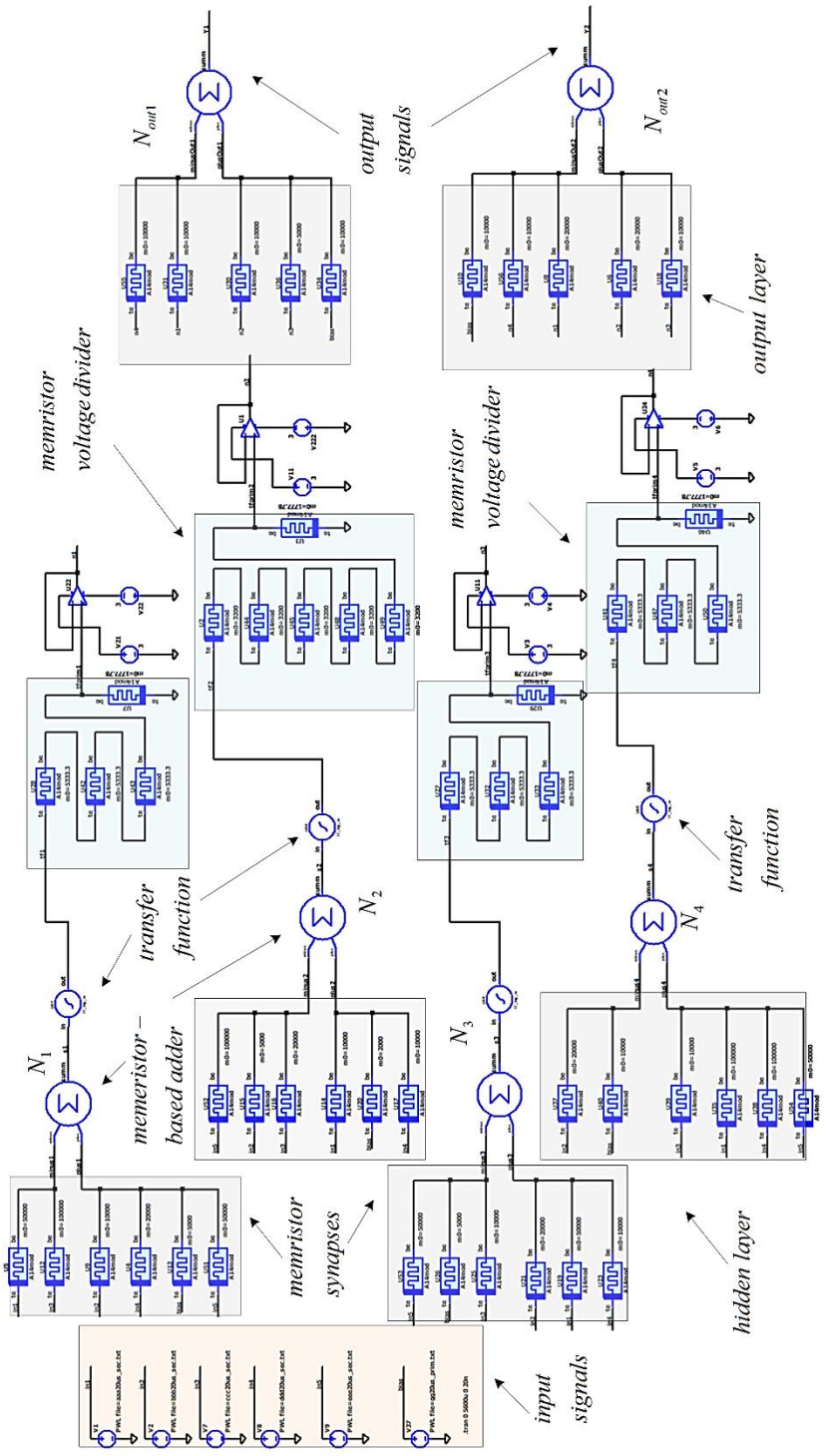

Figure 15. A realization of the considered memristor-based neural network in electronic simulator – LTSPICE, with five input signals – from in_1 to in_5, four neurons in the hidden layer and two neurons in the output layer, with memristor-based synapses and MOS-transistor based logarithmic-sigmoidal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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

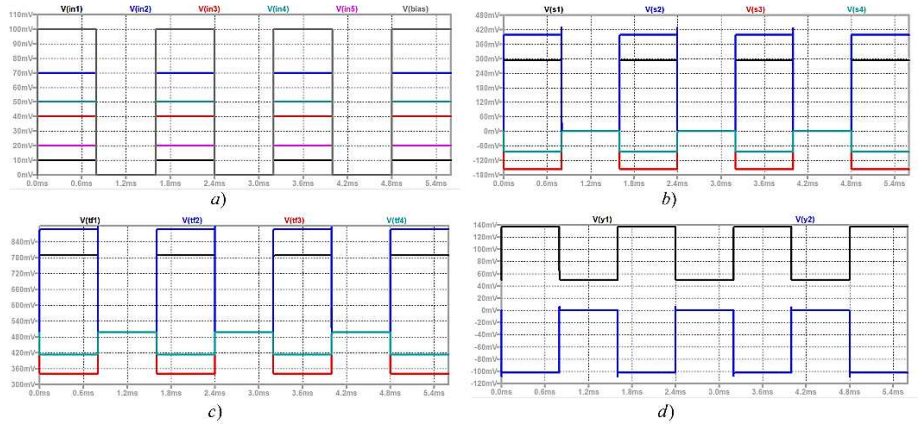

Figure 16. Analysis of the considered memristor-based neural network in LTSPICE environment; *a*) Time diagrams of the input signals v_1 – v_5 and the bias signal v_{bias}; *b*) graphs of the signals after the adders in the hidden layer - v_{s1}, v_{s2}, v_{s3} and v_{s4}; *c*) time diagrams of the signals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

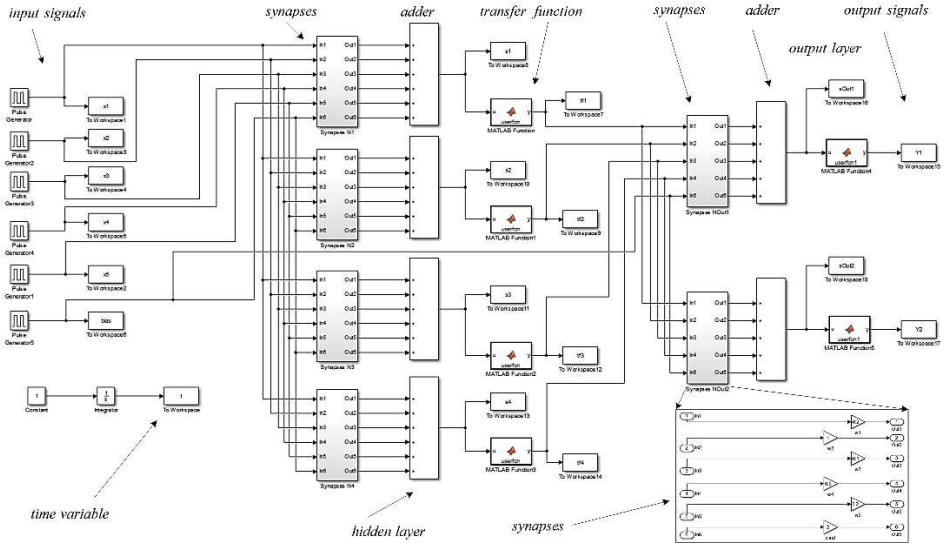

Figure 17. Simulink diagram of the analyzed memristor-based neural network**.**

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 January 2024 doi:10.20944/preprints202401.1245.v1

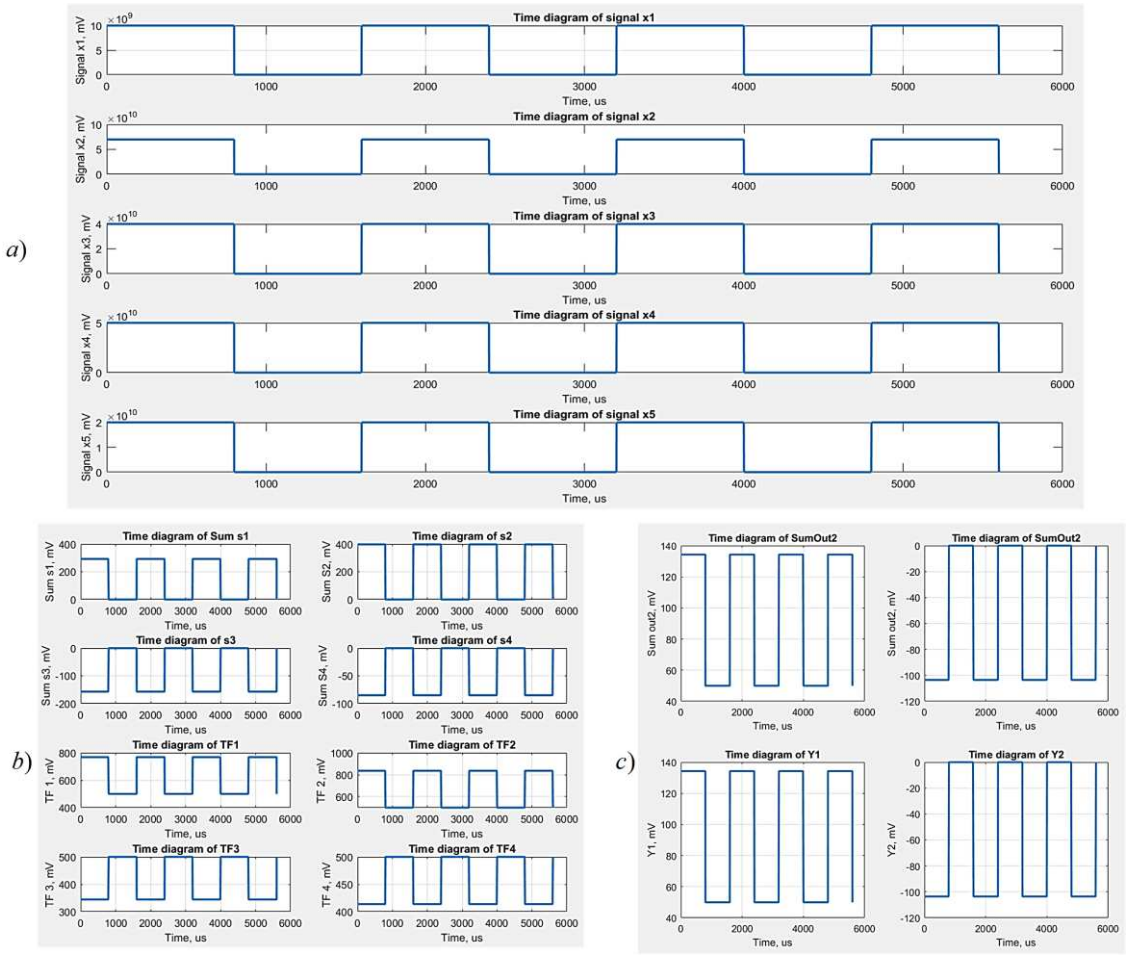

Figure 18. Time diagrams, corresponding to the neural network presented in **Figure** 15; a) time diagrams for the input signals; b) for the hidden layer – the output signals of the adders and of the transfer functions; c) time diagrams for the output layer – the output signals of the adders and these of the output transfer functions.

4. Discussion

Comparing the simulation results of the considered neural network, realized in LTSPICE, MATLAB and SIMULINK 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.

5. Conclusions

The main purpose of this paper - realization of a complete electronic implementation of a memristor-based neural network in 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 minimal number of electronic elements; a simple multi-layer artificial neural network, using memristors for the synaptic weights of positive and negative signs, employing minimal number of electronic components in LTSPICE environment is generated and analyzed. The analyses and simulations are performed, using MATLAB, SIMULINK and LTSPICE. We have applied *A14mod*

memristor model, which is better than the existing ones (in terms of faster operation and simulation), which model helps for 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 to the present-day CMOS high density integrated electronic chips and circuits. 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 exploration of synthesis and analysis of more complex multi-layer memristor-based neural networks for artificial intelligence. Additionally, it aims to explore into practical implementation of such neural networks and compare them 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: Not applicable.

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

References

1. Wang, S., Song, L., Chen, W., Wang, G., Hao, E., Li, C., Hu, Y., Pan, Y., Nathan, A., Hu, G. and Gao, S., 2023. "Memristor-Based Intelligent Human-Like Neural Computing," *Advanced Electronic Materials*, vol. 9, issue (1), pp. 1 – 39, doi: 10.1002/aelm.202200877.
2. M. P. Sah, H. Kim and L. O. Chua, "Brains Are Made of Memristors," in *IEEE Circuits and Systems Magazine*, vol. 14, no. 1, pp. 12-36, 2014.
3. Aggarwal, C., "Neural Networks and Deep Learning" *Springer Int. Publ. AG*: Berlin, Germany, 2018; ISBN 978-3-319-94463-0.
4. O. Krestinskaya, K. N. Salama and A. P. James, "Learning in Memristive Neural Network Architectures Using Analog Backpropagation Circuits," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2019, vol. 66, no. 2, pp. 719-732.
5. W. M. D. Bradley and R. J. Mears, "Backpropagation learning using positive weights for multilayer optoelectronic neural networks," *Conference Proceedings LEOS'96 9th Annual Meeting IEEE Lasers and Electro-Optics Society*, vol.1, USA, 1996, pp. 294-295.
6. Parisien, C., Anderson, C.H. and Eliasmith, C., 2008. "Solving the problem of negative synaptic weights in cortical models," *Neural computation*, vol. 20, issue (6), pp.1473-1494.
7. Xu, W., Wang, J. and Yan, X., 2021. "Advances in memristor-based neural networks," *Frontiers in Nanotechnology*, 3, p.645995.
8. Hong, Q., Zhao, L. and Wang, X., 2019. "Novel circuit designs of memristor synapse and neuron," *Neurocomputing*, 330, pp.11-16.
9. Sah, M.P., Yang, C., Kim, H. and Chua, L., 2012. "A voltage mode memristor bridge synaptic circuit with memristor emulators," *Sensors*, vol. 12, issue (3), pp.3587-3604.
10. Y. Dai, Z. Feng and Z. Wu, "A Novel Window Function Enables Memristor Model With High Efficiency Spiking Neural Network Applications," in *IEEE Transactions on Electron Devices*, vol. 69, no. 7, pp. 3667-3674, July 2022, doi: 10.1109/TED.2022.3172050.
11. Wen, S., Xie, X., Yan, Z., Huang, T. and Zeng, Z., 2018. "General memristor with applications in multilayer neural networks," *Neural Networks*, 103, pp.142-149.
12. Y. Zhang, X. Wang and E. G. Friedman, "Memristor-Based Circuit Design for Multilayer Neural Networks," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 2, pp. 677-686, Feb. 2018, doi: 10.1109/TCSI.2017.2729787.

13. Su, B., Cai, J., Wang, Z., Chu, J. and Zhang, Y., 2022. "A π-Type Memristor Synapse and Neuron With Structural Plasticity," *Frontiers in Physics*, 9, p.798971.
14. Wang, Z., Joshi, S., Savel'ev, S., Song, W., Midya, R., Li, Y., Rao, M., Yan, P., Asapu, S., Zhuo, Y. and Jiang, H., 2018. "Fully memristive neural networks for pattern classification with unsupervised learning," *Nature Electronics*, vol. 1, issue (2), pp.137-145.
15. Zhang, X., Wang, X., Ge, Z., Li, Z., Wu, M. and Borah, S., 2022. "A Novel Memristive Neural Network Circuit and Its Application in Character Recognition," *Micromachines*, vol. 13, issue (12), p.2074.
16. Y. Wang et al., "A Configurable Artificial Neuron Based on a Threshold-Tunable TiN/NbO_x/Pt Memristor," in *IEEE Electron Device Letters*, vol. 43, no. 4, pp. 631-634, April 2022, doi: 10.1109/LED.2022.3150034.
17. Li, B. and Shi, G., 2022. "A CMOS rectified linear unit operating in weak inversion for memristive neuromorphic circuits," *Integration*, vol. 87, pp.24-28.
18. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, S., „The missing memristor found," *Nature* 2008, vol. 453, pp. 80 – 83.
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, vol. 5, pp. 133–142.
20. James, A. "Memristors - Circuits and Applications of Memristor Devices," *IntechOpen*, 2019. doi:10.5772/intechopen.77562, p. 132.
21. E. Lehtonen and M. Laiho, "CNN using memristors for neighborhood connections," (*CNNA 2010*), Berkeley, CA, USA, 2010, pp. 1-4, doi: 10.1109/CNNA.2010.5430304.
22. Joglekar, Y.N. and Wolf, S.J., 2009. "The elusive memristor: properties of basic electrical circuits," *European Journal of physics*, vol. 30, issue (4), p.661.
23. Biolek, Z., Biolek, D. and Biolkova, V., 2009. "SPICE Model of Memristor with Nonlinear Dopant Drift," *Radioengineering*, vol. 18, issue (2).
24. Ascoli, A., Corinto, F., Senger, V. and Tetzlaff, R., 2013. "Memristor model comparison," *IEEE Circuits and Systems Magazine*, vol. 13, issue (2), pp.89-105.
25. Mohammad, B., Jaoude, M., Kumar, V., Al Homouz, D., Nahla, Heba Abu, Al-Qutayri, M., Christoforou, N. (2016), "State of the art of metal oxide memristor devices," *Nanotechnology Reviews*, vol. 5, No. 3, pp. 311-329.
26. S. Dautovic, N. Samardzic, A. Juhas, A. Ascoli and R. Tetzlaff, "Simscape and LTspice models of HP ideal generic memristor based on finite closed form solution for window functions," 2021 *28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, Dubai, United Arab Emirates, 2021, pp. 1-6, doi: 10.1109/ICECS53924.2021.9665488.
27. Zafar M., Awais M., Shehzad M., „Computationally efficient memristor model based on Hann window function," *Microelectronics Journal*, vol. 125, 2022, pp.1-12.
28. Solovyeva, E. B., Azarov, V. A., "Comparative Analysis of Memristor Models with a Window Function Described in LTspice," 2021 *IEEE Conf. (ElConRus*), 2021, pp. 1097-1101.
29. Ascoli, A., Weiher, M., Herzig, M., Slesazeck, S., Mikolajick, T. and Tetzlaff, R., 2022. "Graph Coloring via Locally-Active Memristor Oscillatory Networks," *J. Low Power Electr. Appl*., vol. 12, issue (2), p.22.
30. Mladenov, V., "Advanced Memristor Modeling - Memristor Circuits and Networks," *MDPI* Basel, Switzerland, ISBN 978-3-03897-104-7 (Hbk), pp. 172, 2019, https://doi.org/10.3390/books978-3-03897-103-0.
31. Mladenov, V., "A Unified and Open LTSPICE Memristor Model Library," *MDPI Electronics*, 2021; vol. 10, issue (13), No. 1594. https://doi.org/10.3390/electronics10131594, pp. 1 – 27.
32. Kim, P., 2017. "Matlab deep learning. With machine learning, neural networks and artificial intelligence," *APress*, ISBN-13 (electronic): 978-1-4842-2845-6, DOI 10.1007/978-1-4842-2845-6, pp. 151.
33. May, C., "Passive Circuit Analysis with LTspice® - An Interactive Approach," *Springer Nature* 2020, ISBN 978-3-030-38304-6, pp. 763.
34. V. Mladenov, G. Tsenov and S. Kirilov, "Memristor-Based Neural Network Implementation with Adjustable Synaptic Weights in LTSPICE," 2023 *International Conference Automatics and Informatics (ICAI), Varna, Bulgaria*, 2023, pp. 403-408, doi: 10.1109/ICAI58806.2023.10339092.
35. V. Mladenov, Georgi Tsenov, Stoyan Kirilov, "LTSPICE Memristor Neuron with MOS Transistor-Based Logarithmic-Sigmoidal Activation Function," *accepted for publishing in the Proceedings of the 18th IEEE International Workshop on Cellular Nanoscale Networks and their Applications and The 8th Memristor and Memristive Symposium, Xanthi, Greece* (https://cnna.duth.gr/)

 doi:10.20944/preprints202401.1245.v1

36. Campbell, K.A., 2017. Self-directed channel memristor for high temperature operation. *Microelectronics journal*, 59, pp.10-14.
37. Yuan, R., Tiw, P.J., Cai, L., Yang, Z., Liu, C., Zhang, T., Ge, C., Huang, R. and Yang, Y., 2023. "A neuromorphic physiological signal processing system based on VO_2 memristor for next-generation human-machine interface," *Nature Communications*, vol. 14, issue (1), p.3695.