All-memristive Spiking Neural Network Circuit Simulator

Vladimir Vincan
Dept. of Power, Electronic and Telecommunication Engineering
Faculty of Technical Sciences
University of Novi Sad
Novi Sad, Serbia
vladimirvincan@uns.ac.rs

Jovana Zoranovic
Dept. of Power, Electronic and Telecommunication Engineering
Faculty of Technical Sciences
University of Novi Sad
Novi Sad, Serbia
jzoranovic@uns.ac.rs

Natasa Samardzic
Dept. of Power, Electronic and Telecommunication Engineering
Faculty of Technical Sciences,
University of Novi Sad
Novi Sad, Serbia
nsamardzic@uns.ac.rs

Stanisa Dautovic
Dept. of Power, Electronic and Telecommunication Engineering
Faculty of Technical Sciences
University of Novi Sad
Novi Sad, Serbia
dautovic@uns.ac.rs

***Abstract*— In this paper we present a circuit-level simulation test bed for an all-memristive spiking neural network (MSNN), composed of synapses and leaky integrate-and-fire (LIF) neuron circuits. As recently proposed, an all-memristive neural network can be designed using volatile diffusion memristors as part of the LIF neuron, and non-volatile drift memristors as synaptic elements. The cognitive performances of our MSNN are demonstrated by the implementation of the spike timing dependent plasticity (STDP) learning rule. Starting from a circuit-level memristive neuron model which incorporates volatility, and a synaptic memristive array, a simple MSNN circuit simulator is designed and its performances are discussed.**

Keywords*— *spiking neural networks, memristors, volatility, LTspice, Simscape

I. Introduction

Memristive spiking neural networks (MSNNs) are a specific type of spiking neural networks (SNNs) that utilize unique properties of memristive devices: computational and memory capability, stochasticity, multi-level states, nanoscale size, etc. Various memristive devices are seen as an enabling technology for different energy efficient neuromorphic architectures, especially in the future edge-AI and 5G/6G applications.

Depending on the spiking neuron model type, memristors that model neurons have to fulfil additional requirements. For example, the Hodgkin-Huxley neuron model requires memristors to have the local activity property [1]. Concerning synapses, a common feature of the synaptic memristor is non-volatility [2], [3] which enables the realization of synaptic plasticity.

Various MSNN architectures have been presented so far [4], including all-memristive neural networks with leaky integrate-and-fire (LIF) [5] and integrate and fire (IF) [6] memristive neuron circuits. Nevertheless, the LIF neuron model remains the most popular, as it allows a satisfactory trade-off between device complexity and the ability to replicate biological neuron dynamics [7].

In this paper we present circuit-based simulations of all-memristive spiking neural networks with the memristive LIF neuron model. First, we demonstrate that the modified SPICE model of a memristor [8], [9] with four subcircuit modules and three state variables can be used to mimic synaptic activity. The spike timing dependent plasticity (STDP) learning rule has been demonstrated on a single synaptic memristor in LTspice and Simscape/Simulink simulators. Upon demonstrating that the same memristor model can be used to design the leaky integrate-and-fire neuron when operating in the volatile regime [9], we have realized simple 2×2 and 5×5 memristor synaptic arrays with the ability to perform unsupervised synaptic weight modification.

II. Simulation of Memristive Array in LTspice

Starting from the SPICE model of a memristor with three state variables [8] which was modified with a new window function $f_N(x)$ [9], [10], we have adjusted subcircuit model parameters of the memristor in order to achieve non-volatile dynamics. The memristor model has been implemented with the following equations:

$$R_{mem}(x) = \left(R_{on} - R_{off}\right) \cdot x + R_{off} = \frac{V_{mem}}{I_{mem}}; \quad (1)$$

$$C_x \frac{dx}{dt} = I_x - \frac{x - y}{R_x}; \quad (2)$$

$$C_y \frac{dy}{dt} = I_y; \quad (3)$$

$$C_z \frac{dz}{dt} = I_{mem} - \frac{z}{R_z}; \quad (4)$$

$$I_x = I_{mem} \cdot k \cdot f_N(x); \quad (5)$$

$$I_y = \begin{cases} I_{mem} \cdot k \cdot f_N(y), z > q_p \; and \; V_{mem} > 0 \\ I_{mem} \cdot k \cdot f_N(y), z < q_n \; and \; V_{mem} < 0 \\ 0, else \end{cases} \quad (6)$$

$$f_N(x) = \frac{1 - (2x-1)^2}{1 - (2x-1)^2 + (2x-1)^{2N}}; \quad (7)$$

$$k = \frac{\mu_v \cdot R_{on}}{D^2}, \quad (8)$$

where R_{mem}, V_{mem} and I_{mem} are the resistance, voltage and current passing through the memristor; x, y and z are internal state variables which can be represented as voltages of the subcircuits; μ_v is the dopant mobility; D is the thickness of the active layer; q_p and q_n are threshold parameters for non-volatile resistance switching. Parameters of the neuron and synaptic memristors are shown in Table I. A larger value of capacitance C_x of the synaptic memristor has been set to 5 F, as it provides higher values of the x-module time constant [11], enabling the non-volatile state transition. It is worth mentioning that the values of resistances and capacitors in these submodules do not have physical interpretation, they are used solely to realize coupled differential equations of memristors [11]. Additionally, large capacitance values have already been demonstrated in Biolek's memristor modeling paper [12]. The LTspice code used for modeling the memristors in neurons and memristive synapses is given in [9] and [13].

TABLE I. MEMRISTOR MODEL PARAMETERS

Parameter	Neuron model	Synapse model
R_{on}	1 Ω	1 Ω
R_{off}	100 kΩ	100 kΩ
μ_v	100 $pm^2s^{-1}V^{-1}$	100 $pm^2s^{-1}V^{-1}$
D	10 nm	10 nm
q_p	100 nV	100 nV
q_n	−80 nV	−80 nV
C_x	5 F	0.5 F
R_x	1 Ω	1 Ω
C_y	1 F	1 F
C_z	1 F	1 F
R_z	0.1 Ω	0.1 Ω

The memristor, as part of the LIF neuron, allows gradual conductance modulation due to memductance dependence on the state variable x, which cannot be realized with a simple RC circuit in the LIF model.

The STDP learning rule has been demonstrated on the synaptic memristor. The initial values of the internal state variables have been set to $x_0 = 0.5$, $y_0 = 0.5$ and $z_0 = 0$. By measuring the voltage change of the non-volatile cell state variable Δy, before and after pulse actuation, we can evaluate the overall memristance change, i.e. the overall synaptic weight change [11]. The synapse receives voltage pulse trains from presynaptic and postsynaptic neurons, where each pulse has an amplitude A = 1 V and pulse width t_{pw} = 1 ms. Both neurons emit N = 10 pulses with a t_{ipi} = 1 ms inter-pulse interval. The STDP curve has been calculated by changing the time between the beginning of the presynaptic and the postsynaptic neuron voltage pulse trains, t_{gap}, Fig. 1. As inter-pulse interval decreases, the synaptic weight modification becomes more prominent.

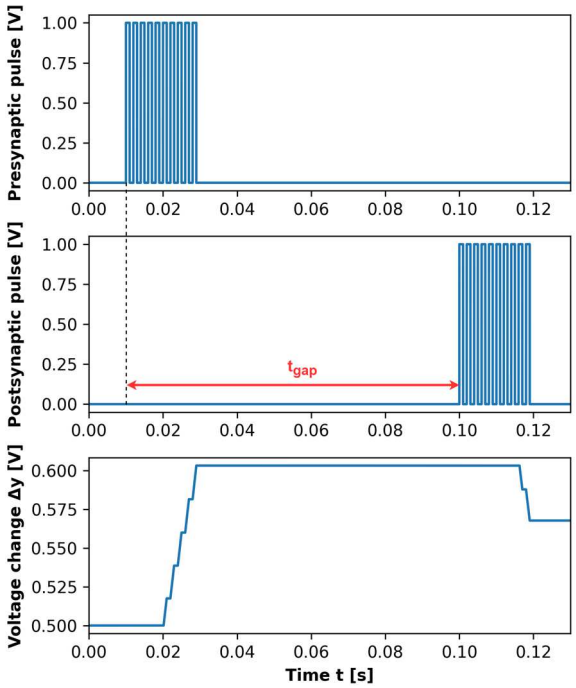

Fig. 1. Demonstration of unsupervised synaptic weight update

Additionally, pulse width t_{pw} and pulse amplitude A have been varied on a single synaptic non-volatile memristor in order to determine the changes in the STDP learning curve. Fig. 2 presents the variation of pulse width t_{pw}, where each pulse has a width of a) 2 ms (blue curve), b) 3 ms (red curve) and c) 4 ms (green curve). Fig. 3 presents the variation of pulse amplitude, where each pulse has an amplitude A of: a) 1 V (blue curve), b) 2 V (red curve) and c) 3 V (green curve).

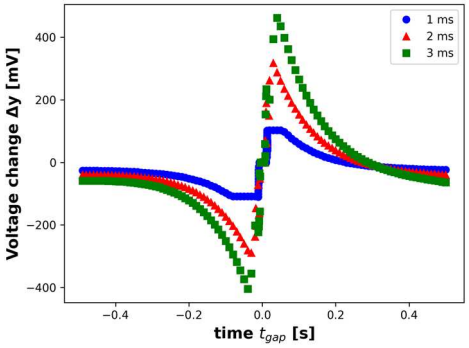

Fig. 2. STDP learning curve with varying presynaptic neuron pulse width of 2 ms (blue curve), 3 ms (red curve) and 4 ms (green curve).

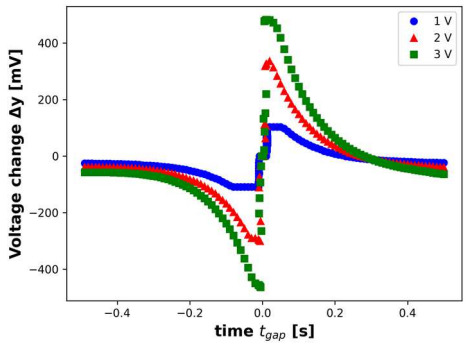

Fig. 3. STDP learning curve with varying presynaptic neuron pulse amplitude: 1 V (blue curve), 2 V (red curve) and 3 V (green curve).

Using four non-volatile memristors, and two volatile memristors [9], we have formed a simple MSNN, composed of a 2×2 memristive synaptic array, and two postsynaptic memristive LIF neuron circuits, Fig. 4. Line resistances that exist in the LIF model are substituted with memristive synapses whose resistance changes with input voltage. Source signals V_0 and V_1 represent signals of the presynaptic neurons, memristors S_{00}, S_{01}, S_{10} and S_{11} placed in the 2x2 crossbar are synapses, which are connected to LIF postsynaptic neurons N_0 and N_1.

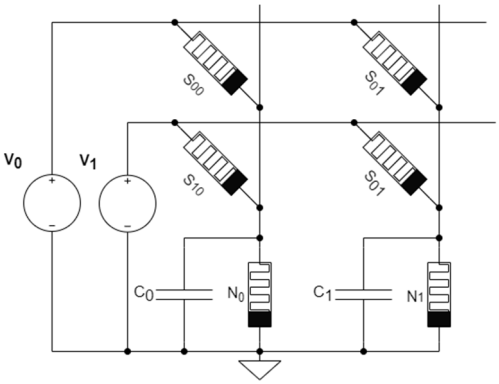

Fig. 4. Circuit used to simulate unsupervised synaptic weight update.

Fig. 5 shows simulation results of unsupervised synaptic weight modification of the memristive array. The initial values of synapses are: $R_{init}(S_{00}) = 90$ kΩ, $R_{init}(S_{01}) = 30$ kΩ, $R_{init}(S_{10}) = 90$ kΩ and $R_{init}(S_{11}) = 40$ kΩ. The waveforms in the upper plot pane, Fig. 5, show the voltages of the sources V_0 and V_1 and postsynaptic neurons N_0 and N_1, while waveforms in the middle plot pane show the voltages of synapses S_{00}, S_{01}, S_{10} and S_{11}. The bottom plot plane presents the current passing through the memristors of the postsynaptic neurons: N_0 (yellow curve) and N_1 (green curve). If synapses S_{01} and S_{11} are set to a low resistive state (LRS), while synapses S_{00} and S_{10} are set to a high resistive state (HRS) and source V_0 emits a pulse train while V_1 is inactive, N_1 will fire, see bottom plot pane in Fig. 5.

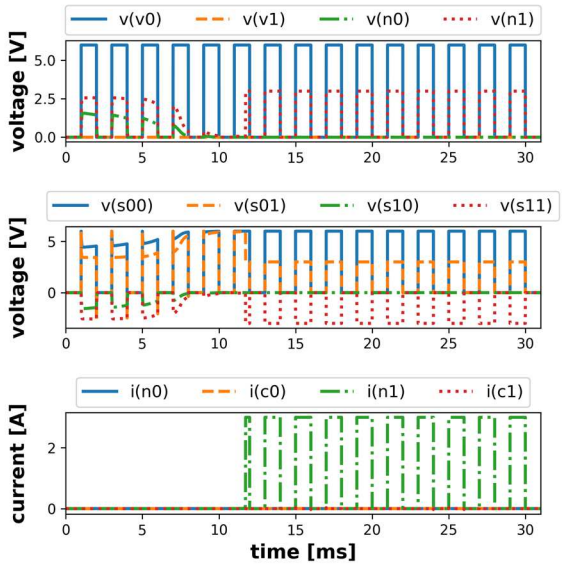

Fig. 5. Simulation results of the memristive array with S_{01} and S_{11} in LRS state while S_{00} and S_{10} are in HRS state. The presynaptic input is coming from V_0 (only the second LIF neuron fires).

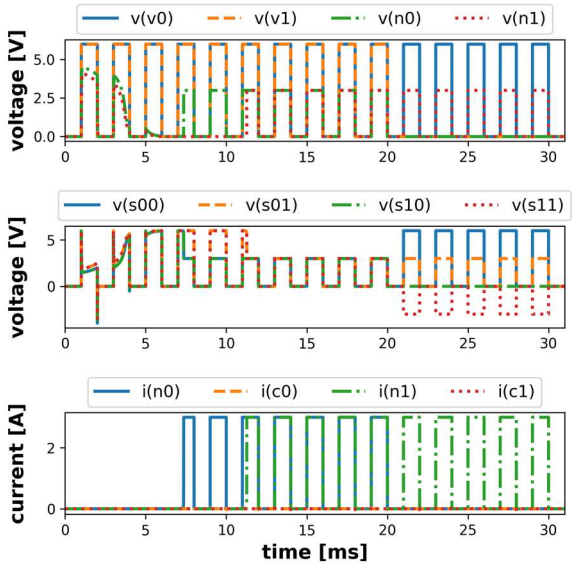

Fig. 6. Simulation results of the memristive array with S_{01} and S_{10} in LRS state while S_{00} and S_{11} are in HRS state. The presynaptic input is coming from both V_0 and V_1 (both postsynaptic LIF neurons fire).

Furthermore, simulations have been performed with two initial values of synapses S_{01} and S_{10} set to LRS ($R_{init}(S_{01}) = 30$ kΩ and $R_{init}(S_{10}) = 20$ kΩ), while synapses S_{00} and S_{11} were in HRS ($R_{init}(S_{00}) = R_{init}(S_{11}) = 90$ kΩ). Presynaptic signal comes from both sources V_0 and V_1, which induces firing behavior in neurons N_0 and N_1, (Fig. 6 bottom graph). Delay in firing event originates from different initial resistances of synaptic memristors. Namely, transition is faster when initial value $x_0=(R_{off}-R_{init})/(R_{off}-R_{on})$ of state variable x is closer to one.

The amplitudes of the currents in Fig. 5 and Fig. 6 depend on the minimal (R_{on}) and maximal (R_{off}) resistance of the memristor model. If the minimal resistance is 1 Ω, current amplitudes can be in the range of several amperes. Additionally, the described neural network has been scaled up to a 5x5 dimension, which demonstrates that the model can be used for an arbitrary size of a fully-connected all-memristive spiking neural network.

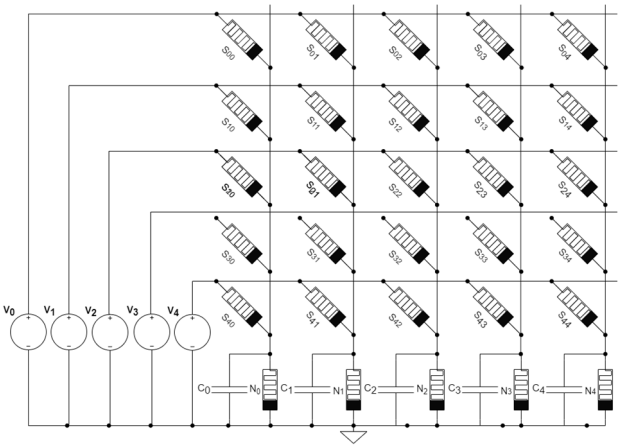

Fig. 7. 5x5 circuit used to simulate unsupervised synaptic weight update.

III. Simulation of Memristive Array in Simscape

The used memristor model and electrical circuits have been independently implemented in the Simscape/Simulink simulation environment. All the experiments described in Section II are repeated with the same sets of parameters (Table I), showing a perfect match with the results presented in Section II. As an illustration of obtained results, the STDP learning rule is shown in Fig. 9. The Simscape model of the used memristor is shown in Fig. 8.

IV. Conclusion

In the paper, a simple memristive spiking neural network simulator is presented, developed independently within LTspice and Simscape/Simulink simulators. The non-volatile memristors used for the synapses, and volatile memristors within the leaky integrate-and-fire neurons, are modeled in the same way, differing in only one parameter, which dictate volatile or non-volatile working regime of memristors. It is demonstrated how the spike timing dependent plasticity learning rule can be implemented on synaptic memristors. The simulation of the behavior of simple 2x2 and 5x5 all-memristive spiking neural networks shows how unsupervised synaptic weight modification of the memristive array controls firing events of appropriate post-synaptic neurons.

ACKNOWLEDGMENT

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under the Grant Agreement 856967, and in part by the Ministry of Education, Science and Technological Development through project no. 451-03-68/2022-14/ 200156 "Innovative scientific and artistic research from the FTS (activity) domain", 2022.


```
component memR < foundation.electrical.branch
% VOLATILE&NON-VOLATILE MEMR MODEL
% 2022 FTS UNS Research Group
parameters
    Roff = {100e3, 'Ohm'};
    Ron = {1, 'Ohm'};
    Rint={50e3, 'Ohm'};    % init res of memR
    A={1,'A'};
    AOM={1,'A*Ohm'};
    HERC={1,'1/s'};
    z0={0,'1'};% init cond for 3rd state var z
    p0={2,'1'};% exponent param. in win. fun.
    uv= {100e-12,  '1'};   % dopant mobility
    D={10e-9, '1'};        % device thickness
    qp={100e-9,'1'};
    qn={-80e-9,'1'};
    Rx={1, '1'};
    Cx={0.5, '1'};
    Cy={1, '1'};
    Rz={0.1, '1'};
    Cz={1, '1'};
end
    k=uv*Ron/D^2;
    x0={(Roff-Rint)/(Roff-Ron),'1'};
    % init cond for 1st state variable x
    y0={(Roff-Rint)/(Roff-Ron),'1'};
    % init cond for 2nd state variable y
end
variables(Access=private)
    x = {x0, '1'};   % 1st state var (vol cell)
    y = {y0, '1'};% 2nd state var, non-vol cell
    z = {z0, '1'};% 3rd state var (charge cell)
end
equations
    assert(Roff>0)
    assert(Ron>0)
    let
        M = x*Ron+(1-x)*Roff;
        My = y*Ron+(1-y)*Roff;
        fours_x = (1-(2*x-1)^2)/…
                  (1-(2*x-1)^2+(2*x-1)^(2*p0));
        fours_y = (1-(2*y-1)^2)/…
                  (1-(2*y-1)^2+(2*y-1)^(2*p0));
        i0_x = i*uv*Ron*fours_x/D^2/AOM;
        i0_y = i*uv*Ron*fours_y/D^2/AOM;
    in
    % **** module x *********
     x.der == (-(x-y)/(Rx*Cx) + i0_x/Cx)*HERC;
    %******** module y *********
        if  (v>0 && z>qp) || (v<0 && z<qn)
            y.der == 1/(Cy)*i0_y*HERC;
        else
            y.der == 0*HERC;
        end
    %******** module z ********
     z.der == 1/(Cz)*(i/A - z/Rz)*HERC;
     i == v/M;
    end
end
end
```

Fig. 8. Simscape model of memristor used in this paper.

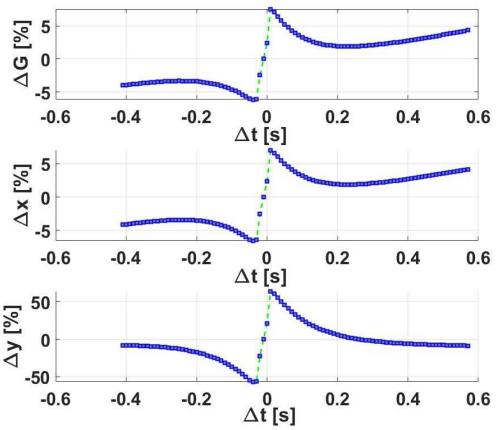

Fig. 9. STDP learning rule of the memristor working in non-volatile regime. Upper graph: relative change of the memductance $G=1/Rmem$ vs. time difference of pre- and post-synaptic neurons firing, $\Delta t=t_{pre}-t_{post}$; Middle graph: relative change of the first state variable x vs. Δt; Lower graph: relative change of the second state variable y vs. Δt.

REFERENCES

[1] Ascoli, A.; Demirkol, A.S.; Tetzlaff, R.; Slesazeck, S.; Mikolajick, T.; Chua, L.O. On local activity and edge of chaos in a NaMLab memristor. Front. Neurosci. 2021, 15, 651452. doi: 10.3389/fnins.2021.651452.

[2] Sung, C., Hwang, H., & Yoo, I.K. (2018). Perspective: A review on memristive hardware for neuromorphic computation. Journal of Applied Physics. doi: 10.1063/1.5037835

[3] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," in Proceedings of the IEEE, vol. 106, no. 2, pp. 260-285, Feb. 2018. doi: 10.1109/JPROC.2018.2790840.

[4] Krestinskaya, O.; James, A.P.; Chua, L.O. Neuromemristive circuits for edge computing: A review. IEEE Trans. Neural Netw. Learn Syst. 2020, 31, 4–23. doi: 10.1109/TNNLS.2019.2899262

[5] Wang, Z., Joshi, S., Savel'ev, S. et al. Fully memristive neural networks for pattern classification with unsupervised learning. Nat Electron 1, 137–145 (2018). doi: 10.1038/s41928-018-0023-2

[6] Pantazi, A.; Wo´zniak, S.; Tuma, T.; Eleftheriou, E. All-Memristive neuromorphic computing with level-tuned neurons. Nanotechnology 2016, 27, 355205. doi: 10.1088/0957-4484/27/35/355205

[7] Schuman, C.D., Potok, T.E., Patton, R.M., Birdwell, J.D., Dean, M.E., Rose, G.S., & Plank, J.S. (2017). A Survey of Neuromorphic Computing and Neural Networks in Hardware. ArXiv, abs/1705.06963.

[8] R. Berdan, C. Lim, A. Khiat, C. Papavassiliou and T. Prodromakis, "A Memristor SPICE Model Accounting for Volatile Characteristics of Practical ReRAM," in IEEE Electron Device Letters, vol. 35, no. 1, pp. 135-137, Jan. 2014, doi: 10.1109/LED.2013.2291158.

[9] Samardzic, N.M.; Bajic, J.S.; Sekulic, D.L.; Dautovic, S. Volatile Memristor in Leaky Integrate-and-Fire Neurons: Circuit Simulation and Experimental Study. Electronics 2022, 11, 894. doi: 10.3390/electronics11060894

[10] 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 28th IEEE International Conference on Electronics Circuits and Systems (ICECS), Dubai, United Arab Emirates, 28 November–1 December 2021. doi: 10.1109/ICECS53924.2021.9665488

[11] Li Q, Serb A, Prodromakis T, Xu H. A memristor SPICE model accounting for synaptic activity dependence. PLoS One. 2015 Mar 18;10(3):e0120506. doi: 10.1371/journal.pone.0120506. PMID: 25785597; PMCID:

[12] Biolek, Dalibor, Viera Biolková and Zdeněk Biolek. "SPICE Model of Memristor with Nonlinear Dopant Drift." *Radioengineering* 18 (2009): 210-214.

[13] Vincan V., Zoranovic J., Samardzic N., Dautovic S., "LTspice Simulations of Memristor Models and Memristive Arrays", https://github.com/VladimirVincan/memristor-models-snn (accessed: May. 15, 2022).