A Memristor Model with Concise Window Function for Spiking Brain-Inspired Computation

Jiawei Xu[1], Deyu Wang[1], Feng Li[1], Lianhao Zhang[2], Dimitrios Stathis[3], Yu Yang[3], Yi Jin[1], Anders Lansner[3], Ahmed Hemani[3], Zhuo Zou[1], Li-Rong Zheng[1]

[1] School of Information Science and Technology, State Key Laboratory of ASIC and System, Fudan University, Shanghai, China;

[2] Department of Electrical Engineering, Technical University of Denmark, Kongens Lyngby, Denmark;

[3] KTH Royal Institution of Technology, Stockholm, Sweden

E-mail: {jwxu16, zhuo, lrzheng}@fudan.edu.cn

2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS) | 978-1-6654-1913-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/AICAS51828.2021.9458424

***Abstract*—This paper proposes a concise window function to build a memristor model, simulating the widely-observed non-linear dopant drift phenomenon of the memristor. Exploiting the non-linearity, the memristor model is applied to the in-situ neuromorphic solution for a cortex-inspired spiking neural network (SNN), spike-based Bayesian Confidence Propagation Neural Network (BCPNN). The improved memristor model utilizing the proposed window function is able to retain the boundary effect and resolve the boundary lock and inflexibility problem, while it is simple in form that can facilitate large-scale neuromorphic model simulation. Compared with the state-of-the-art general memristor model, the proposed memristor model can achieve a 5.8× reduction of simulation time at a competitive fitting level in cortex-comparable large-scale software simulation. The evaluation results show an explicit similarity between the non-linear dopant drift phenomenon of the memristor and the BCPNN learning rule, and the memristor model is able to emulate the key traces of BCPNN with a correlation coefficient over 0.99.**

***Index Terms*—Memristor, window function, non-linear dopant drift, spiking neural network (SNN), Bayesian confidence propagation neural network (BCPNN)**

I. Introduction

The memristor, as an emerging device with versatility and non-volatility, has enabled applications in various fields, such as content-addressable memory [1], programmable analog circuit [2], and artificial neural networks [3]. Memorability is one of the key characteristics of the memristor, which indicates high similarity with the biological synapse. Therefore, the memristor is a promising element to build brain-inspired neural networks.

Spiking neural network (SNN) is inspired by the communication mechanism in the brain. Various memristor-based SNN implementations have been proposed, but are either limited in scale or simple in the learning rule, such as the spike-timing-dependent plasticity (STDP) learning rule. The spike-based Bayesian Confidence Propagation Neural Network (BCPNN) is a large-scale biologically plausible SNN that has been proven useful for achieving brain-like cognitive capability, such as efficient associative memory [4] and working memory [5]. Compared with STDP, the BCPNN learning rule adopts three key traces that are compatible with the signaling cascades of cellular processes underlying the induction of long-term potential [6]. The BCPNN learning rule has higher computational and memory requirements [7], and is usually implemented on high-performance supercomputers. Memristor, functioning as both the memory and the computing unit, has opened new possibilities for neuromorphic implementation of the BCPNN learning rule.

The memristor model plays a significant role in simulating SNN and guiding device fabrication. To emulate the non-linear dopant drift phenomenon of physical memristor devices, a general window function is necessary for the memristor model. However, most window functions are either facing the boundary effect, boundary lock, and inflexibility problem, or complex in form that increase the computational complexity. Though memristor models and window functions have been improved in previous works, efforts are still needed for developing a general, flexible, but concise memristor model to facilitate the effective implementation of the memristor-based BCPNN.

This paper proposes a general memristor model with a flexible but simple window function for the in-situ neuromorphic solution to the memristor-based BCPNN learning rule. The contributions of this work can be concluded as follows:

- A concise window function is proposed to build a memristor model, taking the nonlinearity, boundary effect, boundary lock issue, and flexibility all into consideration.
- As an application of the memristor model, an in-situ neuromorphic solution for the three key traces in the BCPNN learning rule is simulated based on the memristor model.

II. Preliminaries

To facilitate the effective use of the memristor in physical applications, several designer-friendly memristor models have been proposed, such as the non-linear ion drift model, the Simmons tunnel barrier model, and the VTEAM model. The VTEAM model [8] is adopted in this work, as it has the following advantages: 1) it is a voltage-controlled memristor model, and the threshold voltage phenomenon is observed in many physical memristors; 2) the VTEAM model is compatible with many window functions, demonstrating great flexibility to simulate the non-linear dopant drift phenomenon.

The window function is an essential part of a memristor model to emulate the non-linearity. Consequently, dozens of window functions have been proposed. However, most window functions are facing one or more following problems: the boundary effect, the boundary lock, and inflexibility. The Joglekar's window function [9] considers the boundary effect, but will suffer from the boundary lock problem. The Biolek's window function [10] then solves this problem, but the parameter inflexibility harms its application. Recently, Li's window function is proposed [11] to address all three problems, but the expression is complex with six controlling parameters, which may increase the effort required for simulation.

Considering the memristor characteristics and the similarity with the biological synapse, various memristor-based SNNs have been proposed. An all-memristor stochastic SNN architecture [12] is proposed to simulate the functionality of a spiking neuron. A STDP-based SNN is proposed to achieve the mechanism of lateral inhibition and homeostasis by memristor-based synapses [13]. But the majority of the state-of-the-art works are limited in scale and adopt simple learning rules. This work exploits the non-linearity of the memristor and studies the large-scale memristor-based BCPNN, adopting a more brain-like but complex learning rule.

III. Memristor Model

A. Proposed Window Function

To emulate the non-linear dopant drift phenomenon of the physical memristor with a general yet simple memristor model, a novel window function is proposed in the following form:

$$f(x) = j[\mathrm{sgn}(-i)\cdot(x-1)+\mathrm{stp}(-i)]^p \tag{1}$$

$$\mathrm{sgn}(i) = \begin{cases} 1, & i \geq 0 \\ -1, & i < 0 \end{cases} \qquad \mathrm{stp}(i) = \begin{cases} 1, & i \geq 0 \\ 0, & i < 0 \end{cases} \tag{2}$$

where j controls the magnitude, p is a positive real number and i is the memristor current. The memristor current i is considered to be positive if the internal state x is moving towards 1. The parameter p determines the decrease rate of the window function when approaching the boundaries. As p gradually approaches 0, the non-linearity effect is steadily weakening. Fig. 1 demonstrates the proposed window function against the three existing window functions.

Although the proposed window function is simple, it can address all three problems of existing window functions:

1) Boundary Effect: When $x(t)$ reaches 1 and the applied voltage is positive, or when $x(t)$ reaches 0 and the applied voltage is negative, $f(x) = 0$. It means that the x is always in the range of [0, 1]. Therefore, the proposed memristor model is free of the boundary effect.

2) Boundary Lock: Joglekar's window function $f(x) = 1-(2x-1)^{2p}$ suffers from the boundary lock problem, as the state variable x cannot escape from the state $x = 0$ or $x = 1$, because $f(x)$ is always 0 at these two states. In the proposed window function, the boundary lock problem is resolved by adding the current i to model the fact that speeds of approaching and receding from the boundaries are different.

3) Flexibility: Biolek's window function has the parameter flexibility limitation, as the parameter p in its equation $f(x) = 1-[x-\mathrm{stp}(-i)]^{2p}$ can only be a positive integer and it lacks a parameter to control the magnitude. Li's window function exhibits great flexibility by introducing six controlling parameters to its equation $f(x) = j\{1-[\alpha x^3 + a^2[x-\mathrm{stp}(-i)]^2+(1-a^2)+\beta x^2+\gamma x]^p\}$. But its complexity may hinder the large-scale simulation. The proposed window function adopts two controlling parameters, j is used to control the magnitude, and p can be any positive real number to cover almost the whole rectangular area.

B. Evaluation of the Memristor Model

In this work, the VTEAM model is combined with the proposed window function as:

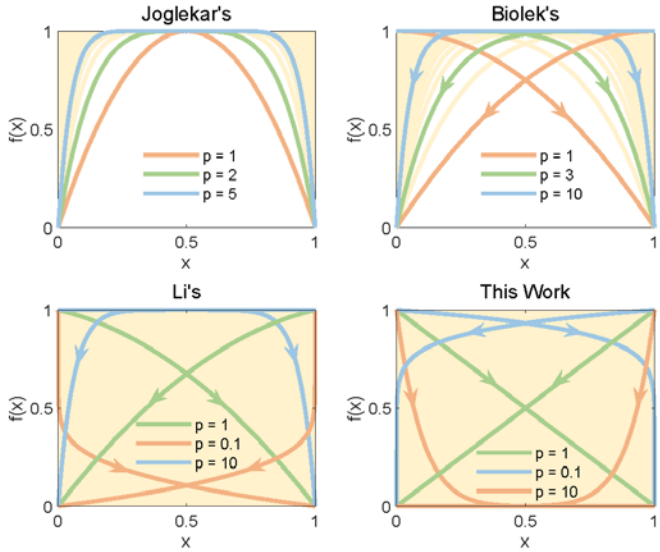

Fig. 1. Different window functions. The lines indicate window functions with different controlling parameters. The yellow regions indicate the flexibility of the window function by showing the covered area.

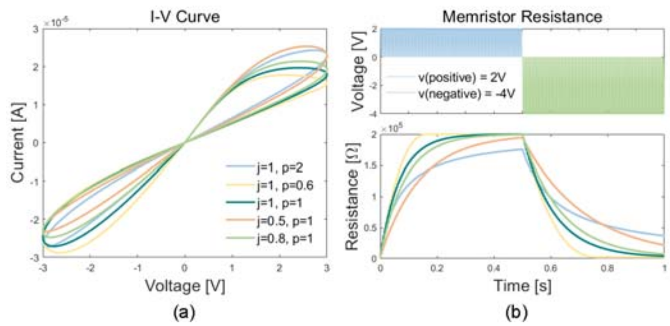

Fig. 2. The non-linear dopant drift emulated by the memristor model.

$$\frac{dw(t)}{dt} = \begin{cases} k_{\mathrm{off}}\cdot\left(\frac{v(t)}{v_{\mathrm{off}}}-1\right)^{\alpha_{\mathrm{off}}}\cdot f(x(t)), & 0 < v_{\mathrm{off}} < v \\ 0, & v_{\mathrm{on}} < v < v_{\mathrm{off}} \\ k_{\mathrm{on}}\cdot\left(\frac{v(t)}{v_{\mathrm{on}}}-1\right)^{\alpha_{\mathrm{on}}}\cdot f(x(t)), & v < v_{\mathrm{on}} < 0 \end{cases} \tag{3}$$

$$x(t) = \frac{w(t)}{W} \tag{4}$$

$$R(t) = R_{\mathrm{on}} + (R_{\mathrm{off}} - R_{\mathrm{on}})\cdot x(t) \tag{5}$$

$$v(t) = R(t)\cdot i(t) \tag{6}$$

Where $w(t)$ is an internal state variable ranging [0, W], W is the assumed physical width of the device, $x(t)$ is an internal state variable ranging [0, 1], $v(t)$ is the voltage across the memristor, $i(t)$ is the current passing through the memristor, $R(t)$ is the resistance of the memristor, and t is the time. Here, v_{on} and v_{off} are threshold voltages, R_{on} and R_{off} are the corresponding resistances of the memristor when $w(t)$ is 0 and W, respectively. The parameters k_{on}, k_{off}, α_{on}, and α_{off} are constants.

The simulated I-V characteristics of the memristor model for a periodic sinusoidal input and a pair of positive and negative pulses are shown in Fig. 2, with the parameter setting provided as follows: $\alpha_{\mathrm{on}} = \alpha_{\mathrm{off}} = 1$, v_{on} = -0.02 V, v_{off} = 0.02 V, R_{on} = 2 kΩ, R_{off} = 200 kΩ, k_{on} = -0.60 nm/s, k_{off} = 1.89 nm/s, W = 7.86 nm, w_{init} = 0 nm. Here, w_{init} represents the initial internal state of the memristor. It is observed that the improved model emulates the non-linear dopant drift phenomenon effectively. Viewing from the hysteresis loop in Fig. 2(a), it is non-linear when reaching either boundary of the memristor. In Fig.

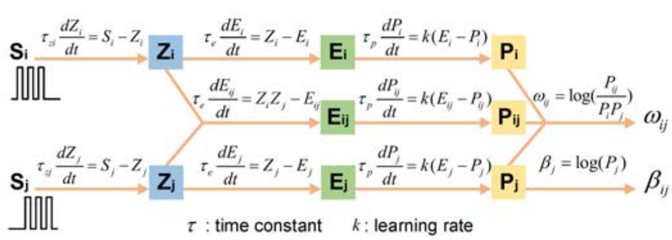

Fig. 3. The spike-based BCPNN learning model.

2(b), the resistance change curve shows that the resistance of the memristor gradually increases with positive pulses and decreases with the negative pulses in a non-linear manner. The resistance is limited between R_{on} and R_{off}, indicating that the model resolves the boundary effect problem. Furthermore, the model is free of boundary lock, as it responses differently to positive and negative pulses. Adopting control parameters to adjust the magnitude and the degree of non-linearity, the improved model demonstrates high flexibility.

IV. Memristor-based BCPNN

A. BCPNN Learning Rules

The spike-based BCPNN [6] is a type of SNN whose learning rule is derived from Bayesian rules. Mini-Column Unit (MCU) is the basic unit in the BCPNN, and MCUs are aggregated to form the Hyper Column Unit (HCU). MCUs are fully-connected with each other via synapses. The strength of the connectivity is calculated according to a set of synaptic state variables that keep track of pre-synaptic, post-synaptic, and synaptic activities. The BCPNN learning model is shown in Fig. 3, where the subscript $_i$ denotes pre-synaptic traces, $_{ij}$ denotes synaptic traces, and $_j$ denotes post-synaptic traces. In the learning model, the pre- and post-synaptic input spikes S_i and S_j are low-pass filtered into three-stage cascaded traces as Z, E, and P according to the ordinary differential equations (ODEs). The P traces represent the estimated posterior Bayesian probabilities of synaptic activities, which are used to compute the synaptic weight w_{ij} and the post-synaptic bias β_j.

The Z, E and P traces are the three key traces in the BCPNN learning rule. To facilitate the hardware implementation, the ODEs can be simplified by Euler's method [6] as:

$$Z(t) = Z(t-1)\times(1-kz)+S(t-1)\times kft \tag{7}$$

$$E(t) = E(t-1)\times(1-ke)+Z(t-1)\times ke \tag{8}$$

$$P(t) = P(t-1)\times(1-kp)+E(t-1)\times kp \tag{9}$$

The current value of each trace is calculated from the previous state. The kft, kz, ke, and kp are all constants.

B. Memristor-Based Synaptic Traces

Existing hardware implementations of the BCPNN learning rule are all based on the von Neuman architecture, and therefore are restricted by the memory wall. To pave the way towards a highly energy-efficient solution, the memristor-based neuromorphic implementation presents great potentials. Looking into the resistance response of the memristor to positive and negative pulses, its similarity to the BCPNN key trace update rule is analyzed with the the memristor model set as: j = 1, p = 1. Take the Z trace of simplified BCPNN as an example:

$$\begin{array}{ll} S=1: & Z(t)=A\cdot Z(t-1)+B \\ S=0: & Z(t)=A\cdot Z(t-1) \end{array} \tag{10}$$

Memristor resistance change to pulses:

$$\begin{array}{ll} v \text{ positive}: & w(t)=C\cdot w(t-1)+D \\ v \text{ negative}: & w(t)=E\cdot w(t-1) \end{array} \tag{11}$$

Where A, B, C, D, and E are all constants expressed as:

$$\left\{\begin{array}{l} A=1-kz \\ B=kft \end{array}\right. \quad \left\{\begin{array}{l} C=1-dt\cdot\frac{k_{\text{off}}}{W}\cdot\left(\frac{v(t)}{v_{\text{off}}}-1\right)^{\alpha_{\text{off}}} \\ D=dt\cdot k_{\text{off}}\cdot\left(\frac{v(t)}{v_{\text{off}}}-1\right)^{\alpha_{\text{off}}} \\ E=1+dt\cdot\frac{k_{\text{on}}}{W}\cdot\left(\frac{v(t)}{v_{\text{on}}}-1\right)^{\alpha_{\text{on}}} \end{array}\right. \tag{12}$$

A significant similarity is observed between the non-linear resistance change of the memristor and the simplified BCPNN traces. Therefore, the non-linear dopant drift phenomenon is utilized to emulate the learning rule of BCPNN. The proposed memristor-based architecture is shown in Fig. 4. It is the memristor-based HCU structure, with i incoming pre-synaptic spikes from MCUs in other or the same HCU and j outgoing post-synaptic spikes. Every HCU involves a $i\times j$ synaptic weight matrix and will update according to the trace cascade of the BCPNN learning model at every millisecond (ms).

V. Simulation and Evaluation

A. Simulation Results

To evaluate the emulation effect, the memristor-based BCPNN implementation is simulated in terms of Z, E, and P traces with the parameters provided as follows: kft = 5/7, kz = 1/11, ke = 1/60, kp = 1/500; j = 1, p = 1, α_{on} = 1, α_{off} = 1, v_{on} = -0.02 V, v_{off} = 0.02 V, R_{on} = 2 kΩ, R_{off} = 200 kΩ, k_{on} = -0.60 nm/s, k_{off} = 1.89 nm/s, W = 7.86 nm, w_{init} = 0 nm. The simulation lasts for 1 second (s) with the simulation step of 1 ms. The simulation results of the proposed memristor model are compared with those of models adopting other window functions, including Biolek's window function and Li's window function. For these two window function, the controlling parameters are set as follows: p = 1 (Biolek's); j = 1, p = 1, a = 1, α = 0, β = -0.3, γ = 0.3 (Li's).

Fig. 5 and Table I present the memristor-based simulation results of Z, E, and P traces. The average error, maximum error, and correlation coefficient are used as the main performance metrics and listed in Table I. As these performance metrics vary with the pre- and post-synaptic spikes, these metrics are measured by the average results of 100,000 simulation tests with random spike generation. Fig. 5 demonstrates a typical case. The input spike of the BCPNN learning rule is either 0 or 1, while the input of memristor-based implementation is either excitatory 150 mV or inhibitory -480 mV. The results show that all three models with the non-linear dopant drift phenomenon achieve a decent emulation effect with above 0.96 correlation coefficients. Due to the model flexibility, the proposed model and the model adopting Li's window function outperform the model adopting Biolek's window function.

B. Model Complexity Evaluation

To evaluate the model complexity, its impact on large-scale simulation is evaluated on the memristor-based neuromorphic implementation at a mammal cortex comparable BCPNN configuration. Taking the rat as an example, it is calculated that there are 5×10^3 HCUs in the rat cortex, and each HCU includes 10^4 incoming connections and 100 MCUs [14]. Therefore, for

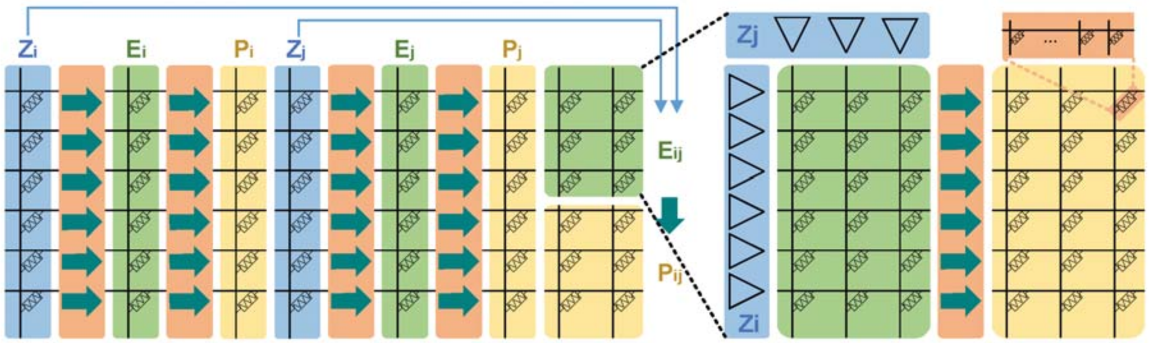

Fig. 4. The memristor-based HCU architecture for the BCPNN learning rule.

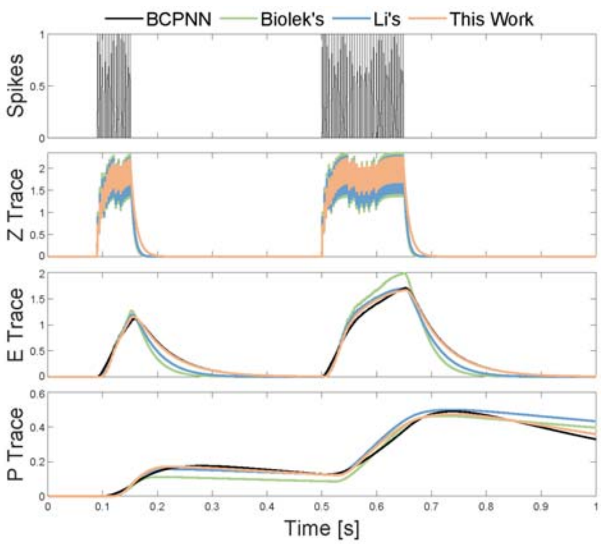

Fig. 5. The memristor-based BCPNN key traces simulation.

TABLE I
MEMRISTOR-BASED SIMULATION RESULTS OF Z, E, AND P TRACES

Trace	Performance Metrics	Biolek's	Li's	This Work
Z	Mean Error	0.059	0.041	0.000
	Max Error	0.513	0.374	0.000
	Correlation Coefficient	0.986	0.992	1.000
E	Mean Error	0.119	0.041	0.018
	Max Error	0.366	0.169	0.175
	Correlation Coefficient	0.966	0.996	0.998
P	Mean Error	0.031	0.022	0.014
	Max Error	0.076	0.065	0.041
	Correlation Coefficient	0.983	0.992	0.995

TABLE II
RAT CORTEX COMPARABLE BCPNN SIMULATION RUNTIME

	Biolek's	Li's	This Work
Simulation Time	71.2 hours	380.2 hours	65.7 hours
Normalization	1.1	5.8	1.0

1 s simulation (step: 1 ms) at the scale of typical rat cortex, it involves around 5×10^9 synaptic trace cascade computation for 10^3 times. The simulation is implemented using different memristor models by Matlab on Intel i9-9920X CPU. The results are listed in Table II. Compared with the model adopting Li's window function, the proposed memristor can provide a competitive emulation effect while achieving a 5.8× reduction of simulation time cost.

VI. CONCLUSION

In this paper, a general yet concise window function is proposed for the memristor model to emulate the non-linear dopant drift. The similarity between the non-linearity of the memristor and the BCPNN synaptic trace update is observed and analyzed. As a neuromorphic application, the simplified BCPNN learning rule is simulated using the improved memristor model. The memristor-based implementation is evaluated for emulation effect and simulation time cost.

VII. ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61876039, 62076066 and 62011530132, Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01, No. 2021SHZDZX0103), and Science and Technology Commission of Shanghai Municipality under Grant No. 19511132000 and 17DZ2260900.

REFERENCES

[1] C. Li *et al.*, "Analog content-addressable memories with memristors," *Nature Communications*, vol. 11, no. 1, pp. 1–8, 2020.

[2] J. Zha *et al.*, "A General Memristor Model and Its Applications in Programmable Analog Circuits," *Neurocomputing*, vol. 267, no. C, pp. 134–140, Dec. 2017.

[3] P. Yao *et al.*, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, 2020.

[4] D. Stathis *et al.*, "eBrainII: a 3kW Realtime Custom 3D DRAM Integrated ASIC Implementation of a Biologically Plausible Model of a Human Scale Cortex," *Journal of Signal Processing Systems*, vol. 92, no. 11, pp. 1323–1343, 2020.

[5] F. Fiebig, P. Herman, and A. Lansner, "An Indexing Theory for Working Memory based on Fast Hebbian Plasticity," *eNeuro*, vol. 7, no. 2, 2020.

[6] Y. Yang *et al.*, "Optimizing BCPNN Learning Rule for Memory Access," *Frontiers in Neuroscience*, vol. 14, pp. 1–15, 2020.

[7] L. Liu *et al.*, "A FPGA-based Hardware Accelerator for Bayesian Confidence Propagation Neural Network," in *2020 IEEE Nordic Circuits and Systems Conference (NorCAS)*, 2020, pp. 1–6.

[8] S. Kvatinsky *et al.*, "VTEAM: A General Model for Voltage-Controlled Memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786–790, 2015.

[9] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: properties of basic electrical circuits," *European Journal of Physics*, vol. 30, no. 4, pp. 661–675, May 2009.

[10] V. B. Z. Biolek, D. Biolek, "SPICE Model of Memristor with Nonlinear Dopant Drift," *Radioengineering*, vol. 18, no. 2, pp. 210–214, June 2009.

[11] J. Li *et al.*, "A novel versatile window function for memristor model with application in spiking neural network," *Neurocomputing*, vol. 405, pp. 239 – 246, 2020.

[12] P. Wijesinghe *et al.*, "An All-Memristor Deep Spiking Neural Computing System: A Step Toward Realizing the Low-Power Stochastic Brain," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 5, pp. 345–358, 2018.

[13] Z. Zhao *et al.*, "A Memristor-Based Spiking Neural Network With High Scalability and Learning Efficiency," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 931–935, 2020.

[14] C. Johansson and A. Lansner, "Towards cortex sized artificial neural systems," *Neural Networks*, vol. 20, no. 1, pp. 48–61, 2007.