ISSN 2635-1676, Nanobiotechnology Reports, 2021, Vol. 16, No. 6, pp. 767–776. © Pleiades Publishing, Ltd., 2021.
Russian Text © The Author(s), 2021, published in Rossiiskie Nanotekhnologii, 2021, Vol. 16, No. 6, pp. 799–809.

NANOELECTRONICS AND NEUROMORPHIC COMPUTER SYSTEMS

Mathematical Modeling of an Analogue Self-Learning Neural Network Based on Memristive Elements Taking into Account Stochastic Switching Dynamics

A. Yu. Morozov[a,b,*], **K. K. Abgaryan**[a,b], **and D. L. Reviznikov**[a,b]

[a] *Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, 119333 Russia*
[b] *Moscow Aviation Institute (National Research University), Moscow, 125080 Russia*
**e-mail: morozov@infway.ru*

Received July 15, 2021; revised July 28, 2021; accepted July 28, 2021

Abstract—In this paper, we consider the issues of modeling an analog self-learning pulsed neural network based on memristive elements. One of the important properties of a memristor is the stochastic switching dynamics, which is mainly due to the stochastic processes of generation/recombination and the movement of ions (or oxygen vacancies) in a dielectric film under the effect of an electric field. The stochastic features are taken into account by adding a term responsible for the additive (Gaussian) noise to the memristor-state equation. The effect of noise on the functioning of an element is demonstrated using the dynamic model of a memristor as an example. The switching of a memristor from the high-resistance state to low-resistance state and vice versa is shown to occur from cycle to cycle in different ways, which is consistent with experimental data. We formulate a stochastic model that describes the hardware analog implementation of a pulsed neural network with memristive elements as synaptic weights and a learning mechanism based on the spike timing dependent plasticity (STDP) method. The operation of two neural networks consisting of one neuron with 64 synapses and two neurons with 128 synapses, respectively, is modeled. The recognition of 8 × 8 images is performed. The stochastic component in the memristor model is shown to have an effect on the fact that the templates from implementation to implementation are distributed among the neurons in different ways and the adaptation of weights (network training) occurs at different rates. In both cases, the network successfully learns to recognize the specified images.

DOI: 10.1134/S263516762106015X

INTRODUCTION

The issue of accelerating computations is relevant at any time. At first, manufacturers increased the processor frequency, then increased the number of cores in one processor, next adapted graphics processors for general computing, and now the transition from digital to analog computing seems promising since they perform computations orders of magnitude faster [1]. In the development of analog computing devices, a relatively new electrical element, a memristor [2, 3], which is a resistor whose conductivity depends on the total electric charge flowing through it, is often used. One of the ways to use these elements to speed up computations is to combine them into a matrix (crossbar), which allows effective implementation of the analog product of a matrix by a vector [4, 5]. Besides this, due to a certain similarity of the properties of these elements with the properties of biological synapses, it is possible to use memristors to create analog self-learning pulsed neural networks (PNNs) [6].

Pulsed neural networks are the third generation of neural networks. In this type of network, information is exchanged in the form of impulses, which is most consistent with the physiology of the biological brain. PNN training is based on local rules for changing weights. There are hardware implementations of a PNN based on semiconductor elements, in particular, the TrueNorth project [7]. Memristive elements are used both in PNNs and in deep neural networks [8]. However, error backpropagation algorithms, which are nonlocal in nature and have great computational complexity, are often used in this case.

There are a number of studies devoted to the implementation of a PNN based on memristors. In particular, the study [9] is aimed at developing experimental and theoretical approaches to finding effective learning rules. An approach to modeling neural networks based on the implementation of metal-oxide heterostructures with nonvolatile memory and multilevel resistive switching is presented in [10]. A learning protocol that is insensitive to the initial state of memristors, as well as to their differences within the same network, is experimentally demonstrated in [11].

PNN learning at the hardware level is usually based on the Hebb rule and synaptic plasticity. The spike timing dependent plasticity (STDP) method is used, according to which the change in the weights of neuron synapses depends on the time difference between the input and output pulses [12–14]. Those synaptic connections that cause neuron activation are strengthened while others are weakened. From the schematic viewpoint, the STDP method is implemented due to the 1T1-R crossbar architecture, in which each memristor corresponds to one transistor responsible for changing the conductivity, and the presence of feedback for each neuron with all synapses. A mathematical model of the PNN was proposed in [15]. The presence of such a model allows the selection of PNN parameters and the simulation of network operation in various modes.

According to experimental data on memristive elements, the nature of their functioning is partially stochastic [16]. As a rule, the process of switching a memristor from the low-resistance state to the high-resistance state and vice versa occurs from cycle to cycle in different ways [17]. In particular, this is associated with random processes occurring in the crystal lattice [18–20]. Mathematical models of memristors are usually formulated in the form of dynamic systems with respect to the memristor-state parameter, which characterizes the conductivity level of the element. In practice, the probabilistic behavior of elements can be taken into account using the stochastic-state equation instead of the deterministic one. One of the options for obtaining such an equation is to introduce an additive in the form of additive white (Gaussian) noise [21].

In this work, we study the effect of the stochastic properties of memristors on the operation of a neuromorphic network. Of practical interest is the issue of using the values of the network parameters determined using the deterministic model under the conditions of the stochastic dynamics of memristive elements. In other words, is it possible to perform network configuration only using a deterministic model? Besides, it is important to understand how the noise level will affect the learning rate and at what level of additive noise the neural network will not be able to function normally.

STOCHASTIC MEMRISTOR MODEL

Several groups of mathematical models of memristors can be distinguished: models of linear [22] and nonlinear drift [23], a model based on the Simmons barrier [24], models using special window functions to limit the state variable [25–27], and models taking into account the levels of voltages, at which the switching process begins in the form of threshold conditions [28–31].

An important feature of memristors is their stochastic behavior. From cycle to cycle, the switching of a memristor from the low-resistive state to the high-resistive state can occur in different ways [16, 17]. The nature of this behavior is associated with random processes occurring at the level of movement of oxygen vacancies in the dielectric film of a memristive element [18–20]. In [32, 33], the effect of noise on the switching process is considered from the general viewpoint of nonlinear relaxation phenomena in metastable systems under the influence of noise [32, 33].

In this work, stochastic features are taken into account by introducing a stochastic addition in the form of additive white (Gaussian) noise into the differential equation of the memristor state. This approach was successfully tested in [21].

We consider a model with a nonlinear voltage dependence. In general terms, the equation describing the memristor state with additive white noise can be represented as follows:

$$dx = af(x)V^s dt + \eta dW,$$

where $x \in [0, 1]$ is the state variable, a is a constant determined by the material properties, V is the current voltage value, s is an odd integer, $f(x)$ is the window function used to approximate the nonlinear effects of ion drift and limit the boundaries, η is the coefficient characterizing the noise intensity, and W is the Wiener process. The Biolek window function is often used [26]:

$$f(x,v) = \begin{cases} 1-(x-1)^{2p}, & v \le 0, \\ 1-x^{2p}, & v > 0. \end{cases}$$

Here, we used the model of a memristor of this class [34] taking into account noise:

$$dx = \eta dW$$
$$+ a \cdot V^s \begin{cases} \left[1-(1-x)^{2round\left(\frac{b}{|V|+c}\right)}\right]dt, & V \le -v_{\text{thr}}, \\ 0, -v_{\text{thr}} < V \le v_{\text{thr}}, & \\ \left[1-(x)^{2round\left(\frac{b}{|V|+c}\right)}\right]dt, & V > v_{\text{thr}}, \end{cases} \tag{1}$$

$$I = x^n \beta \sinh(\alpha_M V) + \chi[\exp(\gamma V) - 1], \quad R = \frac{V}{I},$$

where I, V, and R are the current values of current, voltage, and resistance; v_{thr} is the threshold value of the activation voltage; n, β, α_{M}, χ, and γ are the adjustable parameters in the expression for the current, *round* is the function of obtaining an integer result; and b and c are the adjustable coefficients of the basic equation.

Memristor operation was modeled at the following parameter values: $n = 5$, $\beta = 7.069 \times 10^{-5}$ A, $\alpha_{\text{M}} = 1.8$ V^{-1}, $\chi = 1.946 \times 10^{-4}$ A, $\gamma = 0.15$ V^{-1}, $a = 1$ V^{-5}, $s = 5$,

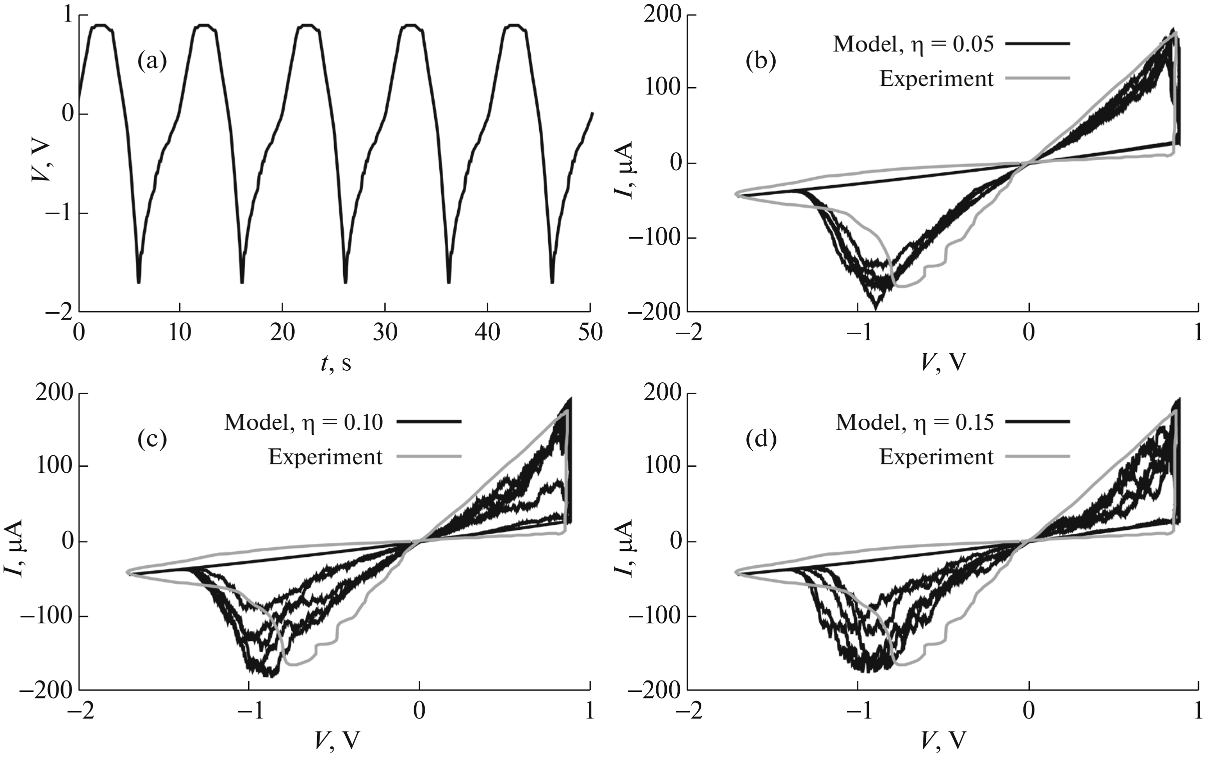

Fig. 1. (a) Voltage versus time, and (b—d) *I—V* characteristics at a noise value of η = (b) 0.05, (c) 0.10, and (d) 0.15.

$b = 15$ V, $c = 2$ V, $v_{\text{thr}} = 1$ V, $x(0) = 0$, and $V(t)$ is shown in Fig. 1a. Some of the values (α_M, γ, a, s, b, c, and v_{thr}) correspond to the values given in the previous study [34] while the part (β, χ) is selected to best fit the experimental data [34] on hafnium oxide (HfO_2) within the framework of the deterministic model ($\eta = 0$).

Figures 1b—1d show a comparison of the model current—voltage characteristic (*I—V*) at various noise values with the experimental curve from [34].

There are several random trajectories on the *I—V* characteristic plots, each of which corresponds to a certain switching cycle of the memristor. We note that the resulting spread in the *I—V* characteristics for various cycles is in agreement with the experimental data [16] for hafnium oxide.

The presence of noise in the memristor model leads to the fact that all parameters acquire stochastic properties. In particular, at $\eta = 0.05$, the estimate of the mathematical expectation of the minimal memristor resistance is $M[R_{\text{on}}] \approx 6.77$ kΩ and of the dispersion is $D[R_{\text{on}}] \approx 1$ kΩ^2. Other parameters such as the maximal resistance and switching voltage are less affected by additive noise. It is clear that at an increase in the noise value, the spread of the parameters will also increase.

Without loss of generality, we assume that the memristor model can be represented as two equations:

$$dx = F_X(x, V)dt + \eta dW, \quad R = F_R(x, V).$$

The first equation determines the dependence of the change rate of the memristor state on the applied voltage (V) and state (x) while the second one determines the value of the memristor resistance.

STOCHASTIC MATHEMATICAL MODEL OF THE NEUROMORPHIC NETWORK

The issues of modeling circuitry solutions of neuromorphic networks, including the use of the STDP learning method, are considered in [15, 35—39]. In this study, the STDP method is implemented using the 1T1R crossbar architecture and the presence of feedback in accordance with [15]. At the activation moment of neuron, two pulses of opposite sign arrive via the feedback channel with delays. If there is activity at the synapse and a positive feedback pulse arrives, then the resistance value of the corresponding memristor decreases, and if a negative feedback pulse arrives, the memristor resistance increases.

The circuitry model of a neuron represents a parallel RC circuit and an abstract pulse generator. When the value of the potential on the capacitor exceeds a certain threshold, its potential is reset, and the pulse generator produces an output signal and a feedback

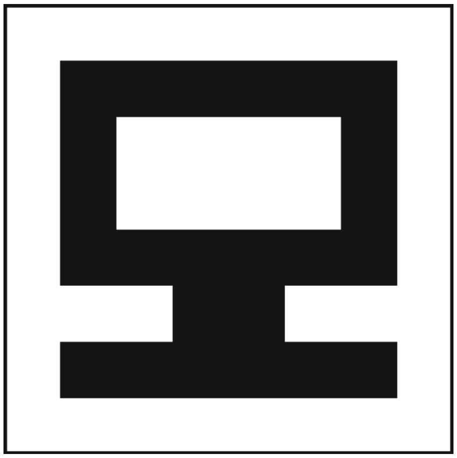

Fig. 2. Recognizable template.

signal. Additionally, at the activation moment of a neuron, other neurons are suppressed (the potential accumulated by them is forcibly reduced in proportion to a certain coefficient).

The network is trained in the following way: at the initial moment, the synaptic weights are randomly initialized, and then, with equal probability, either arbitrary noise or predetermined templates are fed to the network input multiple times. Over time, the network adapts to template recognition. One learning epoch refers to the time during which a sample or random noise is displayed to the network. Template distribution among neurons occurs during the learning process.

A previously developed mathematical model of a neuromorphic network is presented in [15] without taking into account the stochastic dynamics of memristor switching. Taking into account the corresponding properties of memristive elements consists in replacing equations (3) in [15] with:

$$dx_{i,j} = \begin{cases} F_X(x_{i,j}, V_{\text{te}}^j - V_{\text{int}}^j)dt + \eta dW_{i,j}, V_{\text{g}}^i(t) > 0, \\ 0, V_{\text{g}}^i(t) = 0, \end{cases}$$

$$R_{i,j} = F_R(x_{i,j}, V_{\text{te}}^j - V_{\text{int}}^j), \quad i = \overline{1,n}, \quad j = \overline{1,m},$$

where n is the number of inputs, m is the number of neurons, V_{g}^i is the current value of the voltage at the ith input of the neural network, V_{te}^j is the current value of the voltage in the feedback of the jth neuron, V_{int}^j is the voltage at the capacitor of the jth neuron, $R_{i,j}$ is the resistance value of the memristor of the ith synapse of the jth neuron, $x_{i,j}$ is the state of the memristor of the ith synapse of the jth neuron, η is the coefficient characterizing the noise intensity, and $W_{i,\,j}$ is the Wiener process, which corresponds to the ith memristor of the jth neuron.

As a result of using stochastic equations of state for memristors, the entire model of the neuromorphic network also becomes stochastic.

We note that all memristors and neural networks in this study are modeled, and physical implementation is the subject of future research.

MATHEMATICAL MODELING OF A NEUROMORPHOUS NETWORK TAKING INTO ACCOUNT THE STOCHASTIC FEATURES OF MEMRISTIVE ELEMENTS

We consider a network consisting of one neuron ($m = 1$ and so the j index is further omitted) with 64 synapses. Parameter values: $n = 64$, $R_{\text{int}} = 1\ \text{k}\Omega$, $C_{\text{int}} = 45\ \mu\text{F}$, $V_{\text{te}}^{+} = 1.5\ \text{V}$, $V_{\text{te}}^{-} = -1.6\ \text{V}$, $V_{\text{te}}^{0} = 10\ \text{mV}$, $V_{\text{out}}^{+} = 2\ \text{V}$, $V_{\text{th}} = 3\ \text{mV}$, $\tau_r = 20\ \text{ms}$, $\tau_s = 2\ \text{ms}$, and $\tau_{\text{out}} = 10\ \text{ms}$. These values were obtained in [15] based on a deterministic model of a neuromorphic network. Parameters V_{te}^{+}, V_{te}^{-}, V_{te}^{0}, τ_r, and τ_s determine the amplitudes, duration, and delay between pulses in the feedback, due to which learning according to the STDP method occurs.

In the simulation, at each epoch (equal to $\tau_r/2$ s), the vector of input signals $\mathbf{V}_\mathbf{g}(t)$ corresponds to the recognized template (Fig. 2) or is set randomly (the vector elements have a discrete distribution: $V_{\text{g}}^i = 0$ V with a probability of 0.81 and $V_{\text{g}}^i = 2$ V with a probability of 0.19). For clarity, we write the $\mathbf{V}_\mathbf{g}$ vector corresponding to the template as a matrix:

$$\mathbf{V}_\mathbf{g} = \begin{pmatrix} 0, 0, 0, 0, 0, 0, 0, 0, \\ 0, 2, 2, 2, 2, 2, 2, 0, \\ 0, 2, 0, 0, 0, 0, 2, 0, \\ 0, 2, 0, 0, 0, 0, 2, 0, \\ 0, 2, 2, 2, 2, 2, 2, 0, \\ 0, 0, 0, 2, 2, 0, 0, 0, \\ 0, 2, 2, 2, 2, 2, 2, 0, \\ 0, 0, 0, 0, 0, 0, 0, 0 \end{pmatrix} \text{V}.$$

Figures 3 and 4 show the graphs of the behavior of the characteristic voltages of the neural network as a function of time. The upper graphs correspond to the zero value of the coefficient characterizing the noise intensity, i.e., to the actually deterministic memristor model, while the lower plots correspond to two different realizations at a value of $\eta = 0.05$. The dashed line shows what signal was fed to the input of the neural network at each specific epoch. A value of −1 represents noise, and a value of 0 represents a recognizable template. The dotted line with a dot reflects the voltage value on the capacitor of the neuron, and the solid gray thin line reflects the voltage value in the

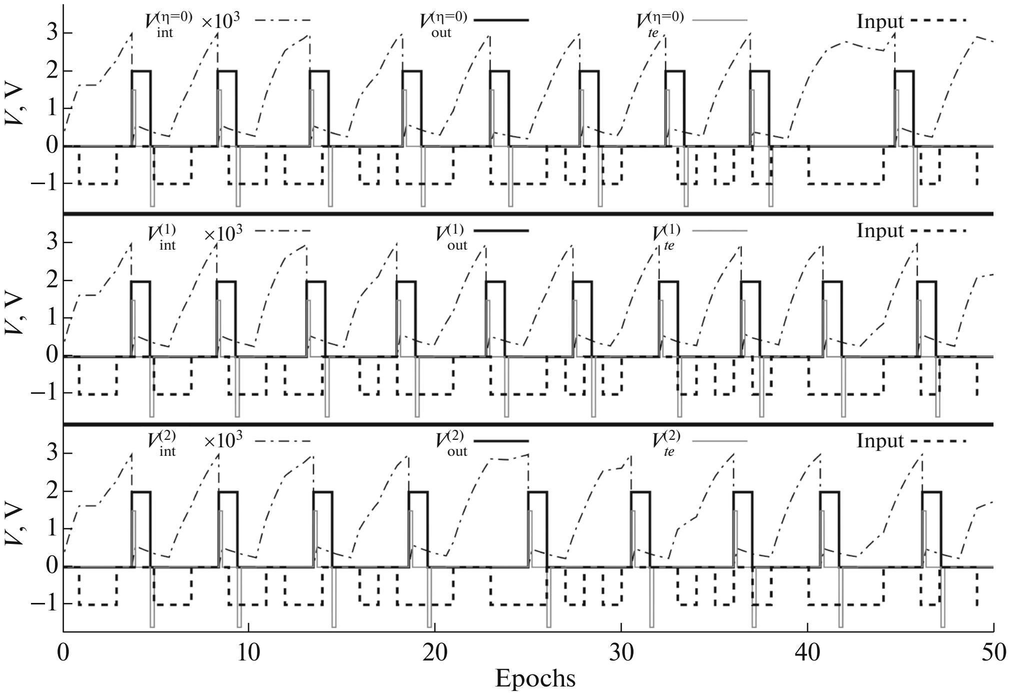

Fig. 3. Comparison of the behavior of the parameters of the neural network at the beginning of training.

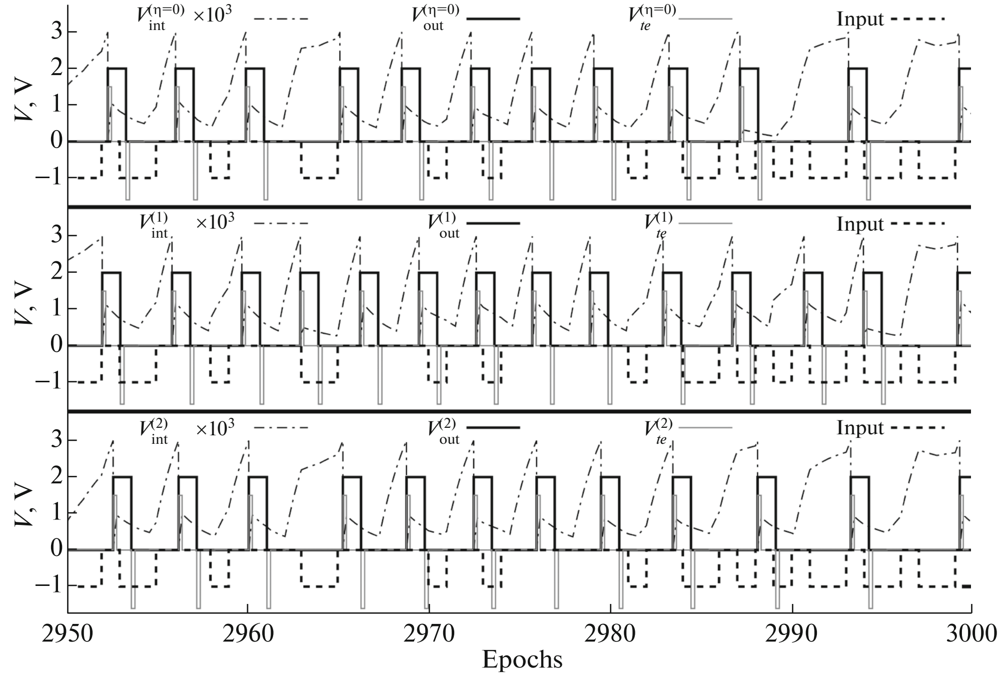

Fig. 4. Comparison of the behavior of the main parameters of the neural network at the end of training at $\eta = 0$ (upper graph) and 0.05 (lower graphs).

feedback. The solid thick line corresponds to the network-output voltage. As soon as the voltage at the capacitor reaches a certain threshold, the neuron is activated: a pulse at the output and a series of pulses in feedback appear.

In all cases, the initial conditions and the learning process were fixed (the same sequence of values was fed to the input of the network). The graphs show that the network successfully learned while, due to the presence of a stochastic term in the memristor model,

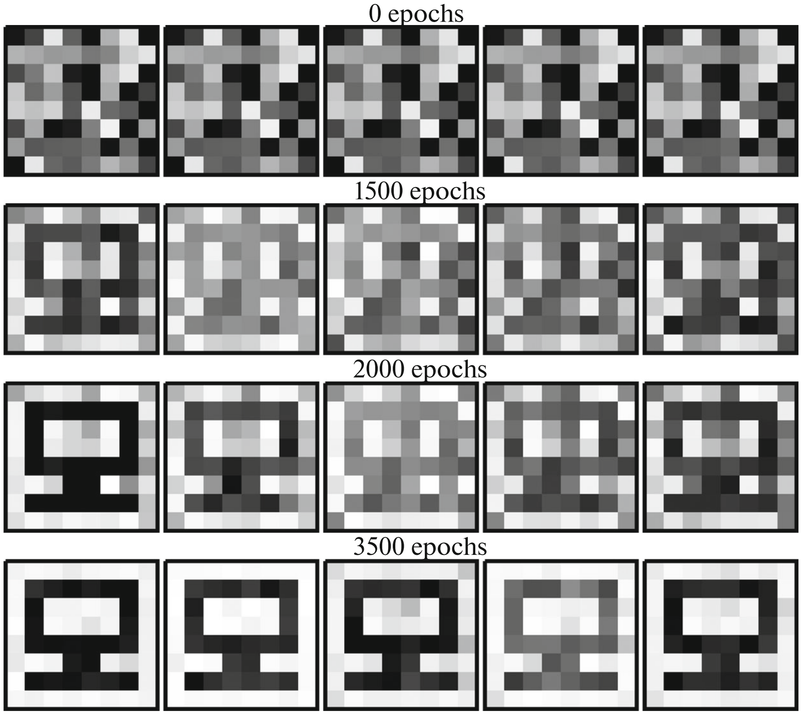

Fig. 5. Comparison of changes in the weights in various implementations.

neuron activations occur with some random shifts in time. Figure 5 shows the process of changing the state variables of memristors for five different implementations. Here, we see that a stochastic addition to the memristor model leads to a different adaptation rate of the weights: in particular, the recognized pattern begins to be seen already from the 1500th epoch in the first implementation and only from the 2000th epoch in other implementations.

Next, we consider the problem of recognizing two samples [15]. The parameters of the neuromorphic network model remain the same, except for $V_{\mathrm{th}} = 4$ mV. Since there is an interaction between neurons, the suppression coefficient of $\alpha = 0.1$ is additionally set (at the activation moment of one neuron, the potential of the other neuron is reset in accordance with the suppression coefficient).

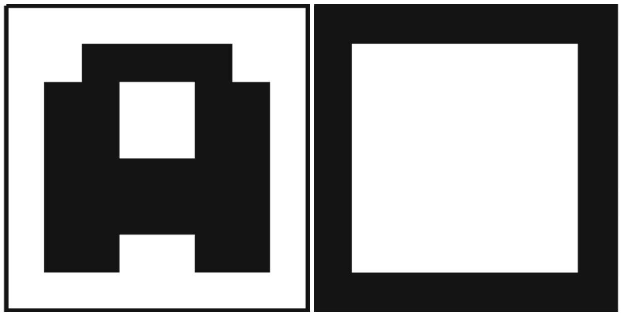

Fig. 6. Recognizable templates.

As earlier, the vector of input signals $\mathbf{V}_{\mathbf{g}}(t)$ in each epoch corresponds to the recognized patterns or is set randomly (the vector elements have a discrete distribution: $V_{\mathrm{g}}^{i} = 0$ V with a probability of 0.8 and $V_{\mathrm{g}}^{i} = 2$ V with a probability of 0.2).

Figures 7 and 8 show graphs of the voltage versus time at the beginning and at the end of the learning process for two different implementations. Figure 9 shows the change in the state variables of memristors for five implementations.

The dotted line shows what was fed to the input of the neural network at each particular epoch. A value of −1 corresponds to noise, and a value of 0 corresponds to the first recognizable template, and a value of 1 corresponds to the second recognized template.

Here, we see that the templates were distributed among the neurons in different ways while the differences in the functioning of the network are insignificant: as in the previous example, there are small time shifts in the output pulses.

In all cases, the network successfully learned to recognize the specified images. However, the stochastic component in the memristor model influenced the fact that the templates from implementation to implementation are distributed among neurons in various ways and the adaptation of weights occurs at various rates.

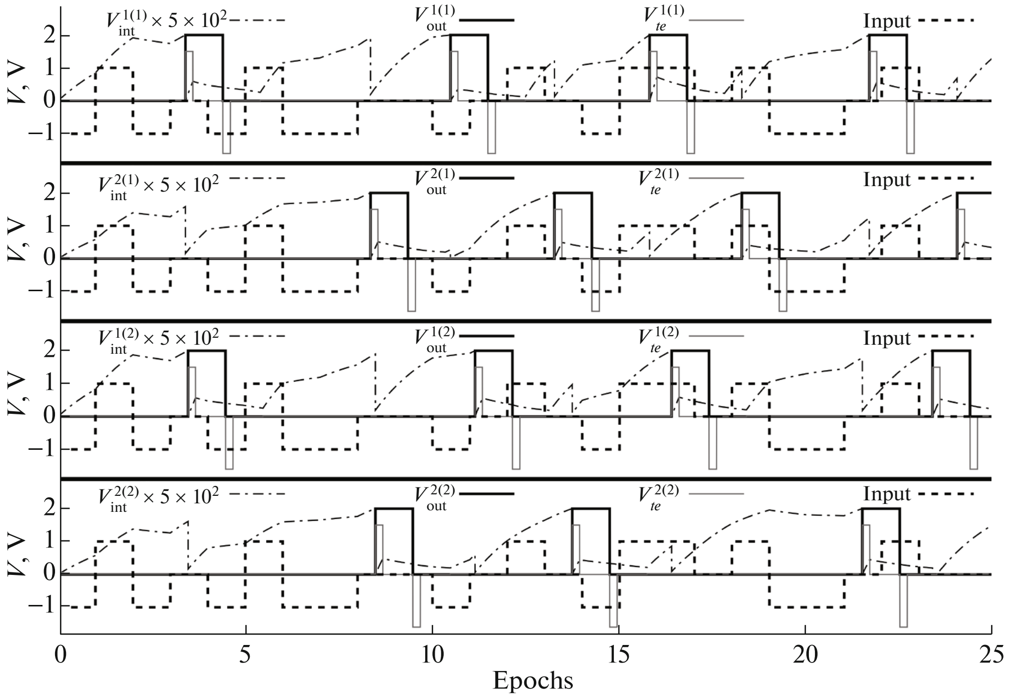

Fig. 7. Comparison of the behavior of the main parameters of the neural network at the beginning of training in two different implementations.

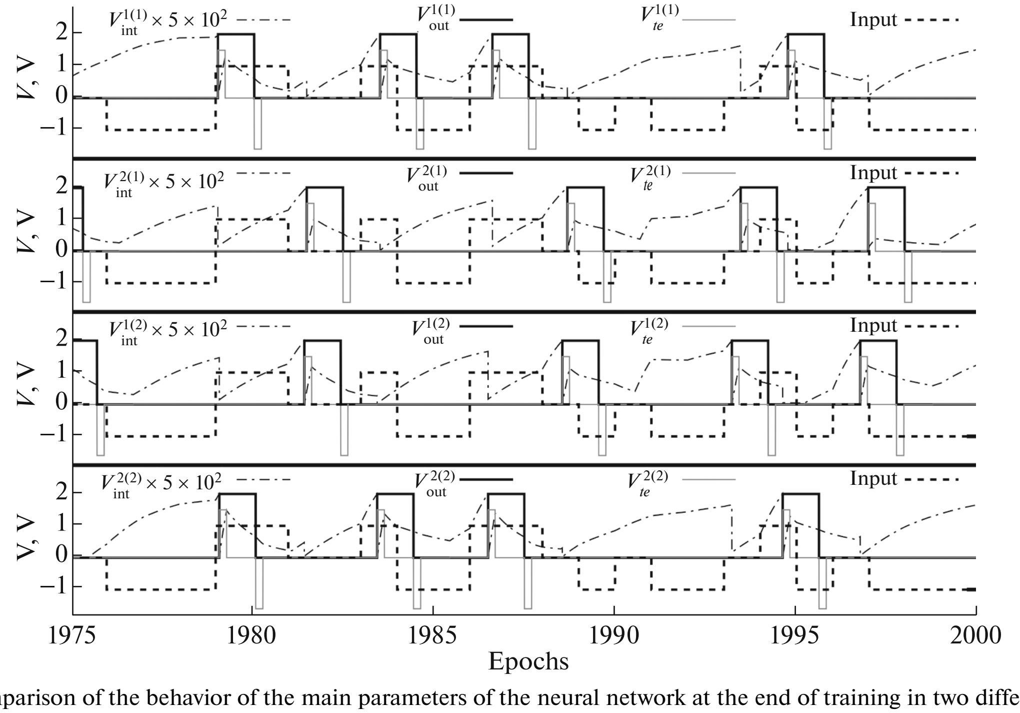

Fig. 8. Comparison of the behavior of the main parameters of the neural network at the end of training in two different implementations.

DISCUSSION

As a result of simulating the operation of a neuromorphic network, it was found that the network successfully learned to recognize the specified templates at the parameter values selected within the framework of the deterministic model. This suggests that it is possible to configure the network without taking into account the stochastic nature of memristive elements.

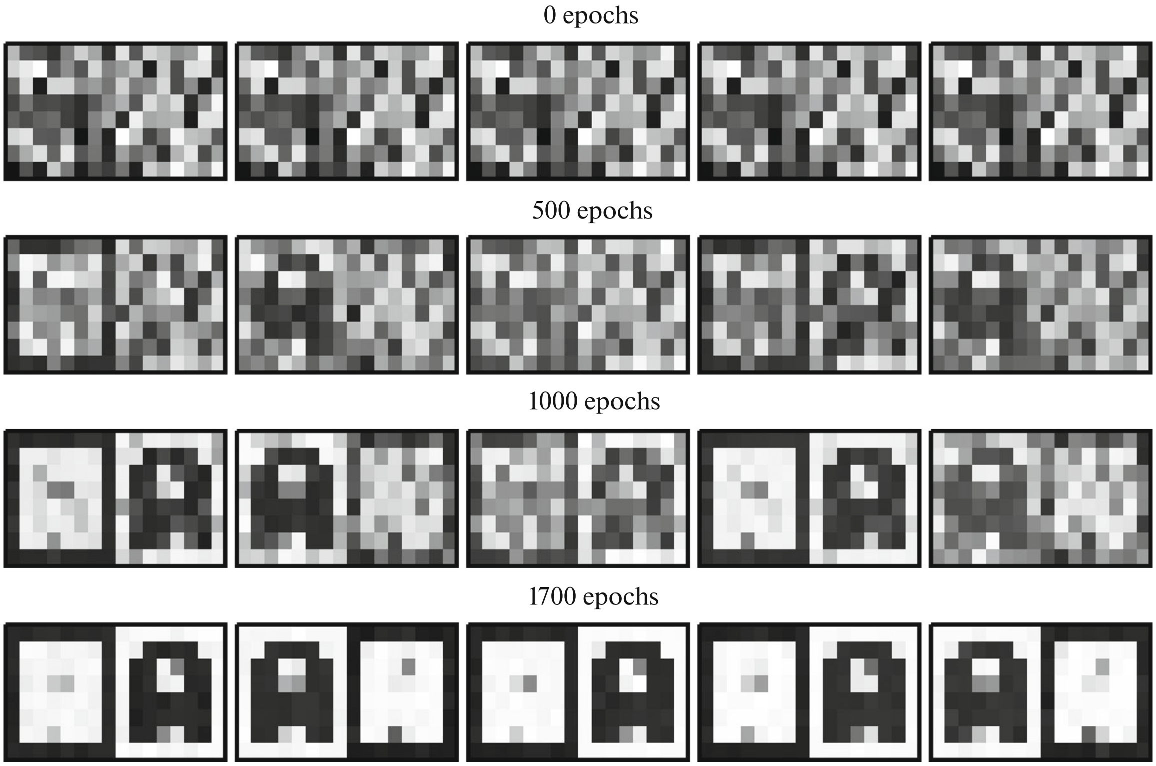

Fig. 9. Comparison of changes in the weights of two neurons in various implementations.

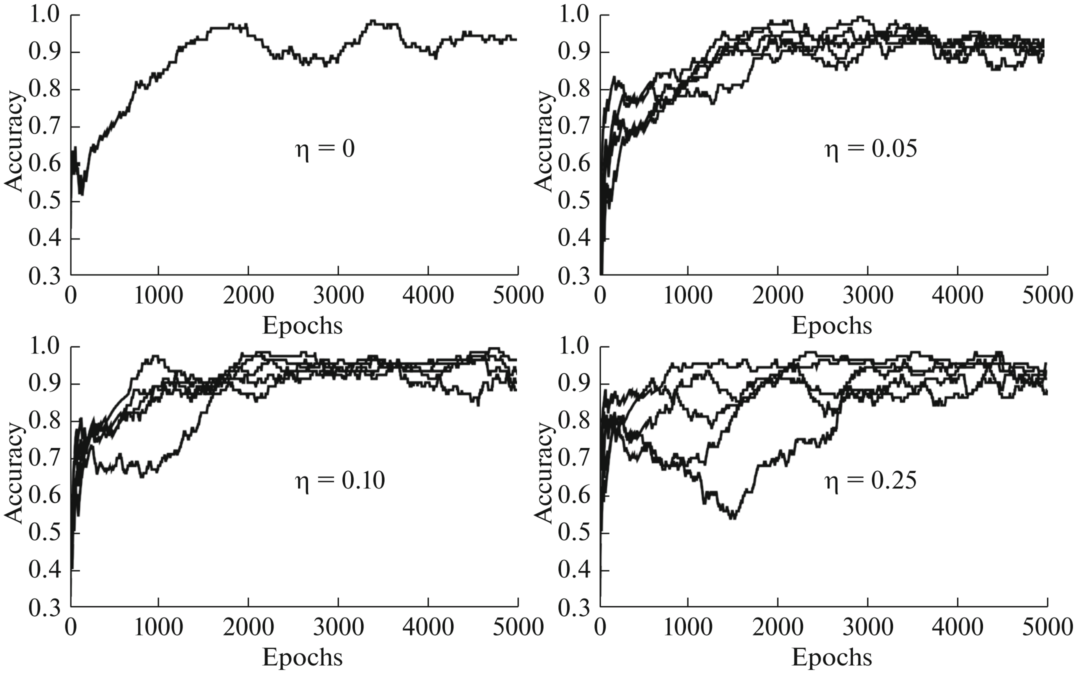

Fig. 10. Dependence of recognition accuracy on the epoch number in the case of one sample.

Additive noise first affects how the templates are distributed among neurons and, to a lesser extent, the delays in the output pulses.

The main characteristic of any neural network is accuracy. Usually, to evaluate it, a test dataset is fed into the network as input and the percentage of correct

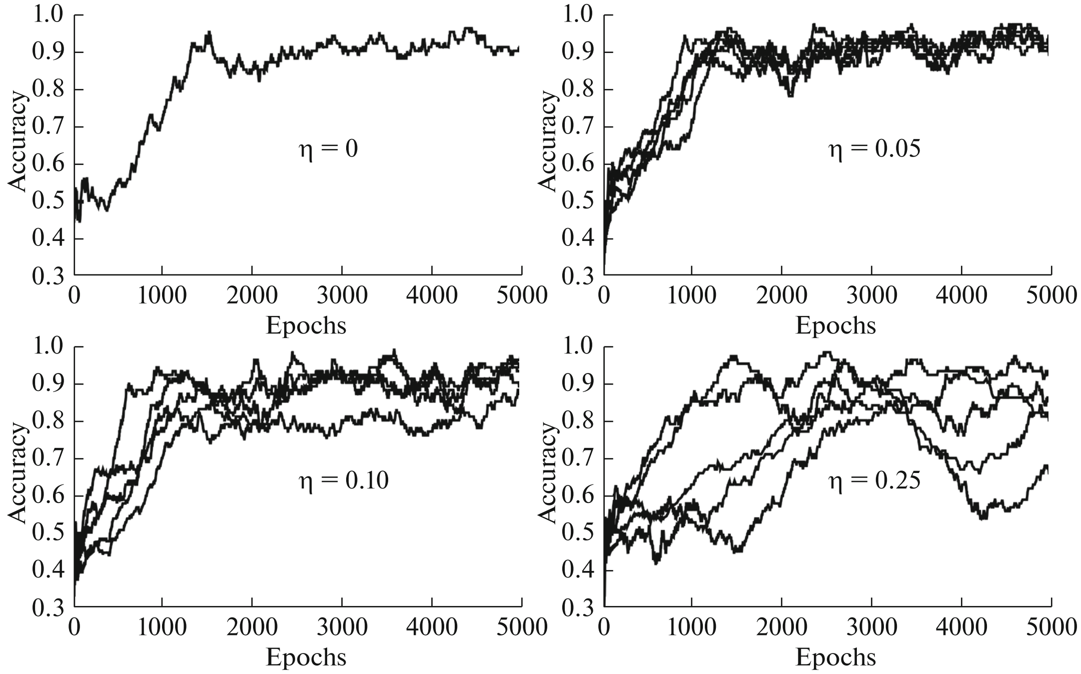

Fig. 11. Dependence of recognition accuracy on the epoch number in the case of two samples.

answers is determined. A distinctive feature of the network considered here is the presence of an internal state that changes during operation and, therefore, the classical testing approach is not strictly correct. In this regard, to estimate the accuracy, averaging over a certain time window was performed in the process of training the network: it was calculated how many times the network correctly responded to the input data. The used window size is 100 epochs. In addition, the question is whether the templates in the learning process can be randomly redistributed among neurons. Therefore, it is impossible to determine in advance which neuron will be responsible for which template before the learning process. To solve this issue, at each epoch, it is checked which neuron is most suitable for the template: the dot product between the input vector and the current vector of the synaptic-weight values of each neuron is maximized.

Figures 10 and 11 show the dependences of the accuracy on the epoch number for the two networks considered earlier. The curves corresponding to zero noise coefficient are actually obtained using the deterministic model. For nonzero values of the noise coefficient, the graphs show five different curves corresponding to different implementations. As can be seen, at a small value of η, the network in a number of implementations is trained faster than that in the absence of noise. However, at an increase in η, the learning process loses its stability and stops converging, as is evidenced by the strong scatter in the graphs at $\eta = 0.25$.

CONCLUSIONS

In the paper, we present the mathematical modeling of a self-learning neuromorphic network based on memristive elements, taking into account the stochasticity of the ongoing processes. A dynamic memristor model with additive noise was implemented. The *I*—*V* characteristics of hafnium oxide obtained as a result of simulation are in agreement with the experimental data. A complex stochastic mathematical model of pulsed neuromorphic network with a learning mechanism according to the STDP rule was formulated. Using the example of template recognition by neuromorphic networks with one and two neurons, the adaptation rate of network weights to recognized templates, as well as the template distribution over neurons, is shown to depend on the stochastic features of memristive elements. In this case, the network parameters can be configured without taking into account the stochastic dynamics of memristor switching. At a relatively low noise level, neural networks with parameters selected using a deterministic mathematical model are successfully trained to recognize specified templates.

FUNDING

This work was supported by the Russian Foundation for Basic Research, grant no. 19-29-03051 MK.

REFERENCES

1. B. N. Chatterji, IETE J. Educ. **34**, 27 (1993). https://doi.org/10.1080/09747338.1993.11436397
2. H.-S. P. Wong, H.-Y. Lee, S. Yu, et al., Proc. IEEE **100**, 1951 (2012). https://doi.org/10.1109/JPROC.2012.2190369
3. J. J. Yang, D. B. Strukov, and D. R. Stewart, Nat. Nanotechnol. **8**, 13 (2013). https://doi.org/10.1038/nnano.2012.240
4. C. Li, M. Hu, Y. Li, et al., Nat. Electron. **1**, 52 (2017). https://doi.org/10.1038/s41928-017-0002-z
5. M. Hu, C. E. Graves, C. Li, et al., Adv. Mater. **30**, 1705914 (2018). https://doi.org/10.1002/adma.201705914
6. A. Yu. Morozov, D. L. Reviznikov, and K. K. Abgaryan, Izv. Vyssh. Uchebn. Zaved., Mater. Elektron. Tekh. **22**, 272 (2019). https://doi.org/10.17073/1609-3577-2019-4-272-278
7. P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, et al., Science (Washington, DC, U. S.) **345** (6197), 668 (2014). https://doi.org/10.1126/science.1254642
8. A. V. Emelyanov, D. A. Lapkin, V. A. Demin, et al., AIP Adv. **6**, 111301 (2016). https://doi.org/10.1063/1.4966257
9. V. A. Demin, D. V. Nekhaev, I. A. Surazhevsky, et al., Neural Netw. **134**, 64 (2021). https://doi.org/10.1016/j.neunet.2020.11.005
10. N. V. Andreeva, E. A. Ryndin, and M. I. Gerasimova, BioNanoSci. **10**, 824 (2020). https://doi.org/10.1007/s12668-020-00778-2
11. A. V. Emelyanov, K. E. Nikiruy, A. V. Serenko, et al., Nanotechnology **31**, 045201 (2019). https://doi.org/10.1088/1361-6528/ab4a6d
12. P. Diehl and M. Cook, Front. Comput. Neurosci. **9**, 99 (2015). https://doi.org/10.3389/fncom.2015.00099
13. Y. Guo, H. Wu, B. Gao, and H. Qian, Front Neurosci. **13**, 812 (2019). https://doi.org/10.3389/fnins.2019.00812
14. V. Milo, D. Ielmini, and E. Chicca, in *Proceedings of the 2017 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA, 2017), p. 11.2.1. https://doi.org/10.1109/IEDM.2017.8268369
15. A. Yu. Morozov, K. K. Abgaryan, and D. L. Reviznikov, Chaos, Solitons Fractals **143**, 110548 (2021). https://doi.org/10.1016/j.chaos.2020.110548
16. A. Rodriguez-Fernez, C. Cagli, L. Perniola, et al., Microelectron. Eng. **195**, 101 (2018). https://doi.org/10.1016/j.mee.2018.04.006
17. G. S. Teplov and E. S. Gornev, Russ. Microelectron. **48**, 131 (2019). https://doi.org/10.1134/S1063739719030107
18. N. V. Agudov, A. V. Safonov, A. V. Krichigin, et al., J. Stat. Mech.: Theory Exp., **024003** (2020). https://doi.org/10.1088/1742-5468/ab684a
19. A. N. Mikhaylov, E. G. Gryaznov, A. I. Belov, et al., Phys. Status Solidi C **13**, 870 (2016). https://doi.org/10.1002/pssc.201600083
20. D. O. Filatov, D. V. Vrzheshch, O. V. Tabakov, et al., J. Stat. Mech.: Theory Exp., No. 12, **124026** (2019). https://doi.org/10.1088/1742-5468/ab5704
21. A. Vasil'ev and P. S. Chernov, Mat. Model. **26** (1), 122 (2014).
22. D. B. Strukov, G. S. Snider, D. R. Stewart, et al., Nature (London, U.K.) No. 453, **80** (2008). https://doi.org/10.1038/nature06932
23. J. J. Yang, M. D. Pickett, L. Xuema, et al., Nat. Nanotechnol. **3**, 429 (2008). https://doi.org/10.1038/nnano.2008.160
24. M. D. Pickett, D. B. Stukov, J. L. Borghetti, et al., J. Appl. Phys. **106**, 074508 (2009). https://doi.org/10.1063/1.3236506
25. Y. N. Joglekar and S. J. Wolf, Eur. J. Phys. **30**, 661 (2009). https://doi.org/10.1088/0143-0807/30/4/001
26. Z. Biolek, D. Biolek, and V. Biolkova, Radioengineering **18**, 210 (2009).
27. J. Zha, H. Huang, and Y. Liu, IEEE Trans. Circuits Syst. II: Express Briefs **63**, 423 (2015). https://doi.org/10.1109/TCSII.2015.2505959
28. S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, IEEE Trans. Circuits Syst. I: Regular Papers **60**, 211 (2012). https://doi.org/10.1109/TCSI.2012.2215714
29. S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, IEEE Trans. Circuits Syst. II: Express Briefs **62**, 786 (2015). https://doi.org/10.1109/TCSII.2015.2433536
30. C. Yakopcic, T. M. Taha, G. Subramanyam, et al., IEEE Electron Dev. Lett. **32**, 1436 (2011). https://doi.org/10.1109/LED.2011.2163292
31. G. Zheng, S. P. Mohanty, E. Kougianos, and O. Okobiah, in *Proceedings of the 2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)* (Columbus, OH, 2013), p. 916. https://doi.org/10.1109/MWSCAS.2013.6674799
32. B. Spagnolo, D. Valenti, C. Guarcello, et al., Chaos Solitons Fractals **81**, 412 (2015). https://doi.org/10.1016/j.chaos.2015.07.023
33. B. Spagnolo, C. Guarcello, L. Magazzu, et al., Entropy **19**, 20 (2017). https://doi.org/10.3390/e19010020
34. V. Mladenov, Electronics **8** (4), 16 (2019). https://doi.org/10.3390/electronics8040383
35. D. Ielmini, S. Ambrogio, V. Milo, et al., in *Proceedings of the 2016 IEEE International Symposium on Circuits and Systems ISCAS* (Montreal, QC, 2016), p. 1386. https://doi.org/10.1109/ISCAS.2016.7527508
36. V. Milo, G. Pedretti, R. Carboni, et al., in *Proceedings of the 2016 IEEE International Electron Devices Meeting IEDM* (San Francisco, CA, 2016), p. 16.8.1. https://doi.org/10.1109/IEDM.2016.7838435
37. S. Ambrogio, S. Balatti, V. Milo, et al., in *Proceedings of the 2016 IEEE Symposium on VLSI Technology* (Honolulu, HI, 2016), p. 1. https://doi.org/10.1109/VLSIT.2016.7573432
38. C. Wenger, F. Zahari, M. K. Mahadevaiah, et al., IEEE Electron Dev. Lett. **40**, 639 (2019). https://doi.org/10.1109/LED.2019.2900867
39. A. Yu. Morozov, K. K. Abgaryan, and D. L. Reviznikov, Izv. Vyssh. Uchebn. Zaved., Mater. Elektron. Tekh. **23**, 186 (2020). https://doi.org/10.17073/1609-3577-2020-3-186-195

Translated by A. Ivanov