

Analyses of a 1-layer neuromorphic network using memristive devices with non-continuous resistance levels

Cite as: APL Mater. 7, 091110 (2019); <https://doi.org/10.1063/1.5108658>

Submitted: 30 April 2019 . Accepted: 05 September 2019 . Published Online: 19 September 2019

A. Siemon , S. Ferch, A. Heitmann, R. Waser , D. J. Wouters , and S. Menzel 

COLLECTIONS

Paper published as part of the special topic on [Emerging Materials in Neuromorphic Computing](#)

Note: This paper is part of the Special Topic on Emerging Materials in Neuromorphic Computing.



View Online



Export Citation



CrossMark

ARTICLES YOU MAY BE INTERESTED IN

[Exploiting the switching dynamics of HfO₂-based ReRAM devices for reliable analog memristive behavior](#)

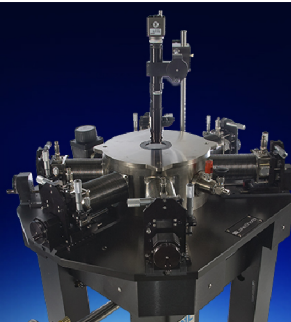
APL Materials 7, 091105 (2019); <https://doi.org/10.1063/1.5108654>

[Ferroelectric materials for neuromorphic computing](#)

APL Materials 7, 091109 (2019); <https://doi.org/10.1063/1.5108562>

[Multilevel HfO₂-based RRAM devices for low-power neuromorphic networks](#)

APL Materials 7, 081120 (2019); <https://doi.org/10.1063/1.5108650>



Cryogenic probe stations
for accurate, repeatable
material measurements

LEARN MORE 

Analyses of a 1-layer neuromorphic network using memristive devices with non-continuous resistance levels

Cite as: APL Mater. 7, 091110 (2019); doi: 10.1063/1.5108658

Submitted: 30 April 2019 • Accepted: 5 September 2019 •

Published Online: 19 September 2019



A. Siemon,^{1,2}  S. Ferch,^{1,2} A. Heitmann,^{2,3} R. Waser,^{1,2,3,4}  D. J. Wouters,^{1,2}  and S. Menzel^{2,4} 

AFFILIATIONS

¹Institut für Werkstoffe der Elektrotechnik II (IWE II), RWTH Aachen University, Sommerfeldstr. 24, 52074 Aachen, Germany

²JARA—Fundamentals for Future Information Technology, 52425 Jülich, Germany

³Peter Grünberg Institut 10 (PGI-10), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany

⁴Peter Grünberg Institut 7 (PGI-7), Forschungszentrum Jülich GmbH, 52425 Jülich, Germany

Note: This paper is part of the Special Topic on Emerging Materials in Neuromorphic Computing.

ABSTRACT

The emerging nonvolatile memory technology of redox-based resistive switching (RS) devices is not only a promising candidate for future high density memories but also for computational and neuromorphic applications. In neuromorphic as well as in memory applications, RS devices are configured in nanocrossbar arrays, which are controlled by CMOS circuits. With those hybrid systems, brain-inspired artificial neural networks can be built up and trained by using a learning algorithm. First works on hardware implementation using relatively large and high current level RS devices are already published. In this work, the influence of small and low current level devices showing noncontinuous resistance levels on neuromorphic networks is studied. To this end, a well-established physical-based Verilog A model is modified to offer continuous and discrete conduction. With this model, a simple one-layer neuromorphic network is simulated to get a first insight and understanding of this problem using a backpropagation algorithm based on the steepest descent method.

© 2019 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/1.5108658>

INTRODUCTION

Current computing systems are designed for computation purposes. However, there is an increasing need for more cognitive tasks as pattern recognition. As we, humans, are very good in solving such tasks, people developed brain-inspired artificial neural networks (ANN's) to handle these tasks. This culminated in the complex, multilayer structured Deep Learning (DL) systems of today that even achieve better-than-human performance.¹

Current DL ANN systems, however, are mainly software constructions that still run on classical von Neumann computers. While their computational performance has tremendously increased over the last decades, thanks to the advancement and scaling of CMOS technology, for implementing more cognitive tasks, they are much less efficient in terms of both system size and energy dissipation than the human brain. In addition, strong performance

improvement is no longer expected due to the ending of Dennard scaling and of Moore's law. Hence, especially for edge computation applications, there is a need for more efficient hardware to realize these ANN systems.

The new emerging resistive switching (RS) devices² (also called "memristors"³ or better "memristive devices"⁴) offer interesting possibilities for an efficient hardware implementation of these ANN's: (i) RS devices can emulate synapse functionality (adaptable, non-volatile weight) in a single device that is small and scalable compared to complex CMOS circuits that would be needed to directly emulate the same functionality; (ii) RS devices can be configured in 2D (and even 3D) crossbar arrays resulting in very dense network connectivity; and (iii) such crossbar configuration is very efficient for analog vector dot product computations.⁵ The latter feature is of major importance as DL ANN's use a backpropagation (BP) learning algorithm to determine the synapse weight values during the

training phase.⁶ The basic kernel of this BP algorithm is matrix multiplications involving a large amount of multiply and accumulate operations (MAC'S). This requires heavy computation and is the computation bottleneck. Using RS crossbar arrays, the results follow directly from a parallel current readout, and so they constitute efficient vector-dot-product engines.^{5,7,8} The BP algorithm needs 6-bit resolution of the synaptic weights during the learning phase, requiring RS devices with analog programming behavior with many programmable levels.^{9,10} Furthermore, because of energy considerations, the RS devices should operate at low currents (and low voltage).

The major RS devices investigated for these applications are Phase-Change Memory (PCM) and redox-based resistive switching RAM (ReRAM) devices.¹¹ These devices were initially developed for applications in binary switching memories but also have potential for multilevel operation. In both devices, the switching from a high ohmic resistance state (HRS) to a low ohmic resistance state (LRS) is called SET and switching from LRS to HRS is defined as RESET. If a device is set and reset with the same polarity of the device, the switching is called unipolar, otherwise it is a bipolar switching device. PCM devices have the drawback of unipolar switching while analog programming behavior is available during SET. In contrast, ReRAM devices can switch bipolar. While standard memory type ReRAM devices show abrupt SET switching, by tuning the material stack, analog programming in both SET and RESET can be obtained.^{12,13} This makes ReRAM devices of high interest for this application.

Filamentary switching ReRAM devices operate by the modulation of a conductive filament, which is induced by the motion of metal cations as Ag or Cu in electrochemical metallization (ECM) memory cells or by the motion of charged oxygen vacancies in devices based on the valence change mechanism (VCM).^{14,15} The conductivity of this filament can be controlled either by the RESET voltage or by the maximum current level attained during SET

operation. Depending on the chosen current level, however, quantization of different conduction states with large “gaps” in attainable values can be observed.^{16–23}

The aim of this work is to investigate how this quantization of possible conduction states in the ReRAM device influences the learning operation of an ANN. For a first insight, the small system presented in Prezioso's work⁷ is analyzed by means of simulation using a deterministic ECM device circuit model featuring discrete conduction steps. Here, a BP learning algorithm based on the steepest decent method is used for a one-layer neuromorphic network. It is shown that the existence of discrete conduction states separated with conductivity gaps can have a strong influence on both the training cost and the system accuracy.²⁴

SIMULATION MODEL

To create a model that features the discrete conduction state mechanism,¹⁸ two well-established models were combined. The first one is a well-established compact model for ECM cells,^{25,26} which was also fitted to different ECM devices. The second one is a kinetic Monte Carlo (kMC) ECM model, which was derived also from the compact model in order to include switching variability and quantization.^{18,27} By combining the quantization part of the kMC, which was compared to the findings in experimental data, with the compact model, a deterministic compact model featuring the quantized conduction mode is obtained. Panel (a) of Fig. 1 depicts together with panel (b) the equivalent circuitry of the continuous conduction model, whereas panels (a) and (c) show the discrete conduction model. In both cases, a filament grows from an inert electrode toward an active electrode, which consists of Ag or Cu. In the continuous conduction model, the tunneling process is modeled by a mean tunneling gap (x_{mean}) and the tunneling processes appear over the complete filamentary area. In the discrete conduction model, the

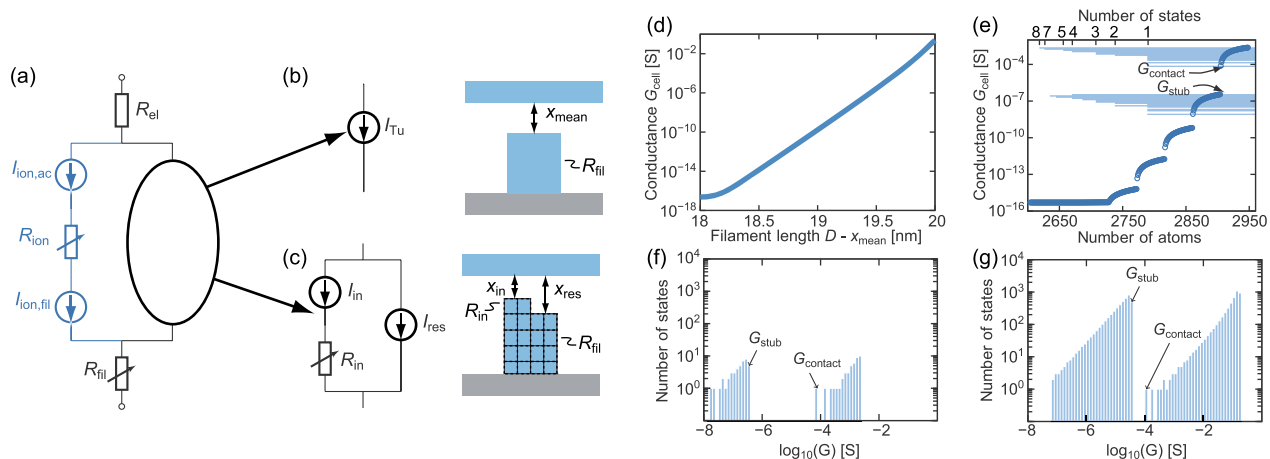


FIG. 1. (a) Equivalent circuit diagram (ECD) of the ECM model, where the tunneling branch needs to be distinguished for (b) the continuous conduction model and (c) the discrete conduction model. (b) Illustration of the tunneling gap for the continuous model including ECD. (c) Illustration of the discrete conduction model with layer-by-layer growth and ECD with all elements. (d) G vs x for the continuous model, and (e) G vs N (number of atoms in the filament) for the discrete model (open symbols) with $r_{fil} = 1$ nm. Thin lines (bar plot) indicate the number of states at the different conduction values for the last two conduction layers. Bar plots showing the number of states vs G for (f) $r_{fil} = 1$ nm and (g) for $r_{fil} = 10$ nm.

filament is built up by an atom by atom and a layer by layer growth mechanism, which results in two parallel tunneling gaps. One tunnel gap describes the tunneling process of the residual layer (x_{res}), which is the last complete layer in the filament, and the other specifies the tunnel process to the incomplete layer with a tunneling gap (x_{in}). The complete model description is found in the [supplementary material](#). In Ref. 28, we showed that the switching kinetics simulated with 1D KMC model agree with the ones simulated with a more complex 2D KMC model,²⁷ which allows for the deposition of atoms at arbitrary positions. Thus, the approximation of the growth mode in our simple 1D model gives reasonable results while reducing the computation time by orders of magnitude.

Figures 1(d) and 1(e) show the conductance of the two different models as a function of their respective state variables. In the continuous ECM model, the conductance increases exponentially if the tunneling gap decreases representing an analog device. The discrete model, however, shows regimes of quasianalog conductance separated by distinct conductance jumps. This jump occurs whenever a new layer appears. The first atom on the new layer gives the biggest conductance jump. This behavior has been reported for the Cu_2S -based ECM cell.^{29,30} With increasing number of atoms in the incomplete layer, the conductance jumps get smaller and eventually they become quasianalog. This transition can be illustrated if the logarithm of the conductance is divided into equidistant conductance bins and counting the number of states within these bins, which is done exemplary for the last two layers in Fig. 1(e). As shown in Fig. 1(f), the number of states is increasing until a complete layer forms. Between two layers, a conductance gap appears. The width of this conductance gap depends on the filament radius. The largest gap is always the gap between the first atom of the galvanic contact layer (G_{contact}), which closes the gap between the filament and the electrode, and the full previous layer (G_{stub}). It is larger for smaller filament radii and vice versa [cf. Figs. 1(f) and 1(g)]. For very large filament radii, the conductance vs state variable plot will converge to the analog exponential behavior of the continuous model.

Both models are deterministic. Thus, the weight updates in response to an excitation is determined exactly by the current state of the cell. In reality, however, the weight update might change due to the intrinsic switching variability. Nevertheless, the conductance

would show quantized levels. Using our simulation model, we are able to discriminate between the effects of switching variability and quantized conduction effects. In this study, we focus on the impact of noncontinuous resistance levels.

SYSTEM AND SETUP

Prezioso *et al.*⁷ proposed an ANN which can distinguish between the three classes “z,” “v,” and “n” using a 3×3 pixel picture, resulting in a 9 input 3 output one layer ANN, and an algorithm based on the steepest descent. Here, the ReRAM devices are used to weight the different inputs for each output and can be changed by applying write pulse of height $\pm V_{\text{write}}$ and pulse length t_{cycle} . This system is adapted and built up for simulation using the continuous and the discrete model. It is chosen as a simple way to understand the differences in the behavior between the two buildups. A complete and extended description of the system and the learning algorithm is found in the [supplementary material](#).

The simulations were performed using a combination of MATLAB and Cadence software. MATLAB acted as a control unit during the simulation, whereas the main circuit simulations were performed using the Spectre Circuit Simulator from Cadence. The training procedure starts with a read step of all possible input pictures, which is used to evaluate the network response and cost values. Afterwards, the weight updates were calculated using MATLAB. Weight update, read, and evaluation steps define one training epoch, which is repeated in a loop structure until the specified number of training epochs was reached. In one training epoch, the whole dataset of input pictures is used.

CONTINUUM MODEL SIMULATIONS

For first tests, the behavior of the conductance state depending on the applied write voltage pulses is studied. For this, the device was first read by a read voltage of $V_{\text{read}} = 0.1$ V and then a voltage pulse of different pulse heights V_{write} and $t_{\text{cycle}} = 10$ μs length is applied. Afterwards, a second read pulse is performed. In Fig. 2(a), the difference of both simulated read conductance values is plotted

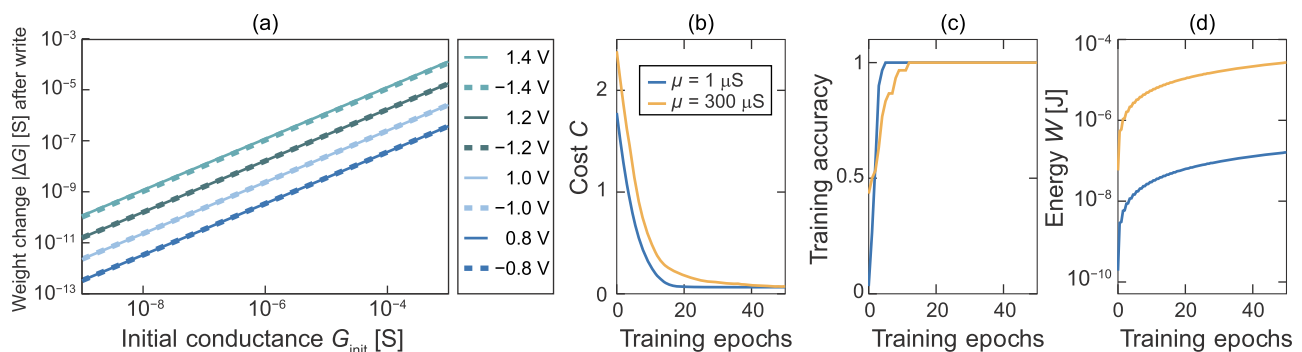


FIG. 2. (a) Conductance evolution dependent on the applied write voltage for 10 μs pulses. The used voltage is varied in a range of $|V_{\text{write}}| = 0.8$ V–1.4 V. The dashed lines indicate the negative voltages, whereas the solid lines depict the positive values. Comparison of (b) cost, (c) training accuracy, and (d) energy during training for two different weight initializations. Blue correspond to $\mu = 1$ μS , $\sigma = 0.1$ μS , and $V_{\text{write}} = 1.3$ V, whereas orange, the standard parameters were used ($\mu = 300$ μS , $\sigma = 20$ μS , and $V_{\text{write}} = 1$ V).

TABLE I. Standard simulation parameters.

Parameter	Symbol	Value
Write voltage	V_{write}	1 V
Pulse length	t_{cycle}	10 μs
Read voltage	V_{read}	0.1 V
Parameter for the range of the output	α	1.7159 V
Normalization factor	β_0	2×10^5
Mean conductance value	μ	300 μS
Standard deviation of the conductance values	σ	20 μS
Minimum conductance value	G_{min}	0 μS
Maximum conductance value	G_{max}	600 μS

against the conductance of the first read. The conductance change shows not only a dependency on the applied voltage, as was expected due to the nonlinear switching kinetics of the device, but also a state dependence. In our model, this state dependence is based on the filamentary growth mechanism in an ECM cell^{31,32} and the exponential dependence of the current on the tunneling gap.^{33,34} The state-dependence has been shown for different ReRAM devices and this basis for analog weight adaption.^{35–43} The behavior of the experiment of Fig. 2(a) is crucial for neuromorphic applications, since it gives a first hint of how to pick the write voltage for specific initialization states of the system.⁴⁴ If not stated otherwise, the standard parameters of Table I are used for simulation. The first step of all system simulations is the initialization of the devices with conductance states, which are drawn from a truncated normal distribution with the minimum and maximum value of the conductance $G_{\text{min}} = 0 \mu\text{S}$ and $G_{\text{max}} = 600 \mu\text{S}$, respectively.

To verify the correct behavior, the system was set up with the continuous model for two different initializations of the

conductance state of the 60 devices. In case 1, the standard parameters were used. In case 2, the mean conductance value and the standard deviation were chosen to be $\mu = 1 \mu\text{S}$ and $\sigma = 0.1 \mu\text{S}$, respectively. To achieve a comparable learning speed, two different write voltages for these experiments are used. In case 1, the standard write voltage was chosen, whereas the write voltage of case 2 is set to 1.3 V.

The cost and training accuracy of the system are shown in Figs. 2(b) and 2(c), respectively. As expected, the training cost is reduced and the accuracy increases with the number of training epochs, showing the correct behavior of the system. Even though the accuracy reached 100% after a few epochs, the cost function could be further minimized. The accuracy only gives an account of the right classification for the training set, but it does not identify the insecurity of the decision. The cost function illustrates how well the decision is matching the ideal output and thus indicates the security of the decision.

In Fig. 2(d), the energy consumption of training the ReRAM array is depicted. Even though the write voltage of case 2 is higher, the energy consumption is less, since the array has an overall higher impedance than in case 1. Note that this energy is only the energy needed for the ReRAM array. The overall power consumption will be higher since the whole CMOS periphery and control unit is not taken into account. Overall, the system seems to be robust against the change in the conductance initialization if the write voltage is chosen properly with respect to the initialization.

As a next step, the convergence behavior of the system is further investigated. This time the system was five times initialized with the same distribution of conductance. Thus, the impact of the update voltage can be evaluated independent of the specific initialization. For each initialization, the write voltage was varied between 1 V and 1.6 V to find the optimum voltage for a stable, fast convergence. Figures 3(a)–3(d) show the cost function and the training accuracy of these simulations. In Figs. 3(a) and 3(b), the voltages are depicted

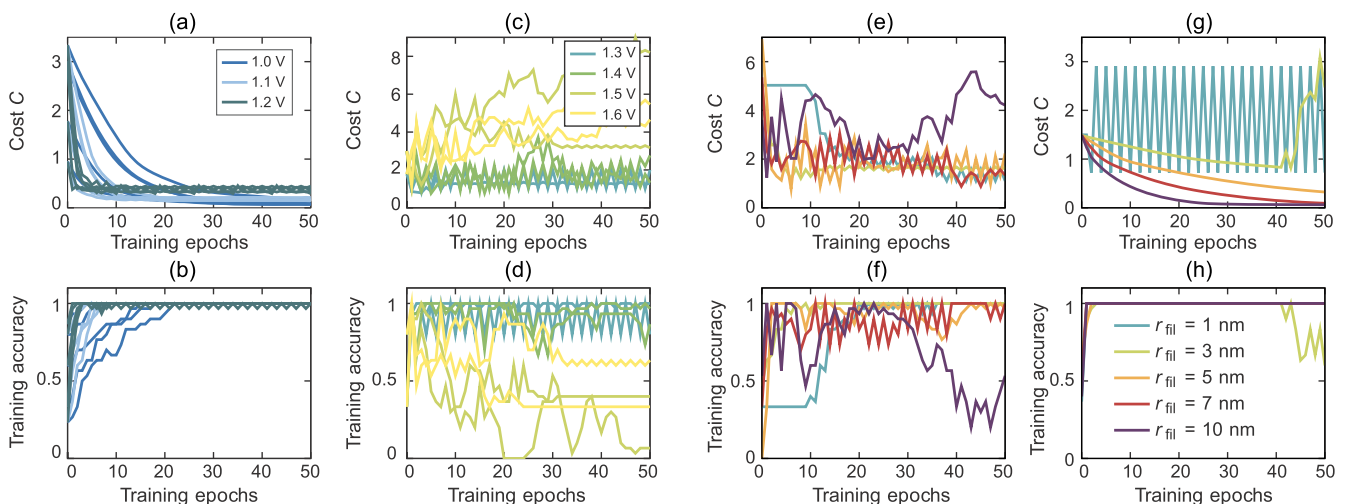


FIG. 3. [(a)–(d)] Influence of the update voltage V_{write} on the training performance. The results in the top row depict the voltages showing convergence, whereas the used voltages in the bottom row cause divergence. [(e)–(h)] Training performance of the system considering discrete conductance states for different filament radii. For (e) and (g), the conductances were initialized with $\mu = 300 \mu\text{S}$ and $\sigma = 60 \mu\text{S}$, and $V_{\text{write}} = 1 \text{ V}$ was used. (f) and (h) show the results for the parameters $\mu = 1 \mu\text{S}$, $\sigma = 0.1 \mu\text{S}$, and $V_{\text{write}} = 1.3 \text{ V}$.

for which the system mostly converges. The update voltages 1 V and 1.1 V show a fast and stable convergence. For 1.2 V, the system converges even faster, but it is not really stable and does not reach the high accuracy and low cost function values as for the other two voltages. Having a higher update voltage, the conductance change is higher in each step, which is beneficial in the beginning for the rough weight adjustment, but hindering the fine adjustment. Here, a voltage of 1.1 V is the best compromise. For this voltage, the system shows the best performance, as it converges faster than for 1 V and is more reliable than with 1.2 V. If the voltage is further increased, the problems occurring for 1.2 V will be even more present, as shown in Figs. 3(c) and 3(d). As shown before, if the system is initialized in a different conductance regime, the optimum voltage can be different, since the conductance change depends on the voltage and on the starting conductance [cf. Fig. 2(a)].

DISCRETE SIMULATION MODEL

If a discrete conduction mechanism is assumed, the system needs to be able to compensate for the nonexistent conduction states. To test the robustness of the system with regard to this effect, the discrete resistance model is used. Since the conduction gaps depend on the filament radius, the filament radii of all devices are set to 1 nm, 3 nm, 5 nm, 7 nm, or 10 nm. For the initialization, two extreme cases can be considered: initializations in a high state density region and in a low state density region. The standard initialization parameters of Table I ($\mu = 300 \mu\text{S}$ and $\sigma = 20 \mu\text{S}$) result in an initialization in the galvanic layer (low state density region). Here, the conductance state density is very low. Hence, the standard deviation was increased to $\sigma = 60 \mu\text{S}$. Figures 3(e) and 3(f) depict the cost function and training accuracy of these simulations. In Figs. 3(e) and 3(f), abrupt spikes are visible for all used filament radii. Thus, convergence does not seem feasible for the system. As a second test, the devices were initialized with $\mu = 1 \mu\text{S}$ and $\sigma = 0.1 \mu\text{S}$. Here, the targeted conductances are below G_{stubb} and thus in a higher state density region for filaments with $r_{\text{fil}} > 1 \text{ nm}$. For filament radii $r_{\text{fil}} > 3 \text{ nm}$ and thus a nearby high state density, the cost function reduces quickly [Fig. 3(g)]. For the filament radius of 1 nm, however, the targeted conduction is inside the gap between G_{stubb} and G_{contact} . Accordingly, all cells are initialized to G_{stubb} . Like before, the write voltage is set to 1.3 V to achieve comparable learning rates. The results of these

simulations are depicted in Figs. 3(g) and 3(h). Here, convergence is not achieved for filaments with 1 nm or 3 nm radii. The 1 nm case does not show any convergence from the beginning. In contrast, the 3 nm case converges until the 42nd epoch, but then starts to oscillate.

To investigate the nature of those fluctuations, the transients of the conductances of randomly selected devices of the $r_{\text{fil}} = 5 \text{ nm}$, $\mu = 300 \mu\text{S}$, and $\sigma = 60 \mu\text{S}$ case are shown in Fig. 4(a). The gray horizontal lines depict feasible conductance states. All transients start in the states of the galvanic layer and as expected the transition between the states is abrupt. Thus, the change in the conductance is abrupt, too. If one device state starts to oscillate between G_{stubb} and G_{contact} (pink and light green lines), fluctuations in the cost function result. Similar to using larger writing voltages, the weight jumps are too big to be handled properly with the algorithm. The weights cannot be adjusted fine enough due to the missing state density in the gap region. The algorithm works better for the ($\mu = 1 \mu\text{S}$, $\sigma = 0.1 \mu\text{S}$) cases as the devices were initialized in a region of high state density.

Another possibility to avoid transitions over the gap, and thus to keep the conductance changes small enough to have less convergence issues, is to change the normalization factor β [cf. supplementary material (20)], since it defines the target weights. Until now, the standard value was 2×10^5 and is kept unchanged. Due to the standard normalization factor, the targeted weights are in the range of $5 \mu\text{S}$ – $15 \mu\text{S}$. Hence, the weights of the $\mu = 300 \mu\text{S}$ cases need to be decreased, whereas the weights of the $\mu = 1 \mu\text{S}$ cases need to be increased. To achieve lower targeted weights, β is set to 2×10^6 for the $r_{\text{fil}} = 3 \text{ nm}$, $\mu = 1 \mu\text{S}$, and $\sigma = 0.1 \mu\text{S}$ case. The comparison of the overall maximum and minimum conductance states of the system with both β values is depicted in Fig. 4(b). Using the standard β value, the maximum conductance value shows jumps to one point above G_{contact} (turquoise). This jump corresponds to the start of the cost function fluctuation. In contrast, the maximum conductance value in the simulation with the increased β value stays below G_{contact} during the course of the training.

DISCRETE SIMULATION MODEL WITH VARIABLE FILAMENT

In a second study, the filament radii were no longer assumed to be equal for all cells. They were drawn from a truncated normal

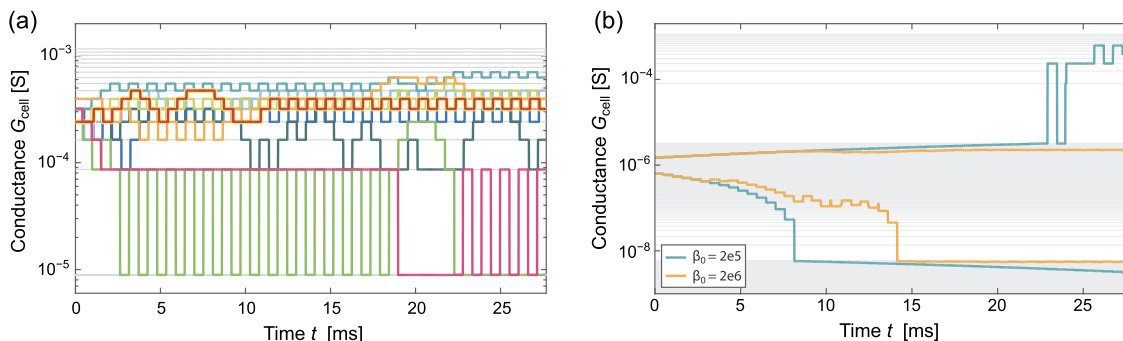


FIG. 4. (a) Transient of 10 randomly selected cell conductances for the case $r_{\text{fil}} = 5 \text{ nm}$, $\mu = 300 \mu\text{S}$, and $\sigma = 60 \mu\text{S}$. The background grid indicates feasible states. (b) Transient of minimum and maximum conductance of all cells in the network for two values of normalization parameter β_0 .

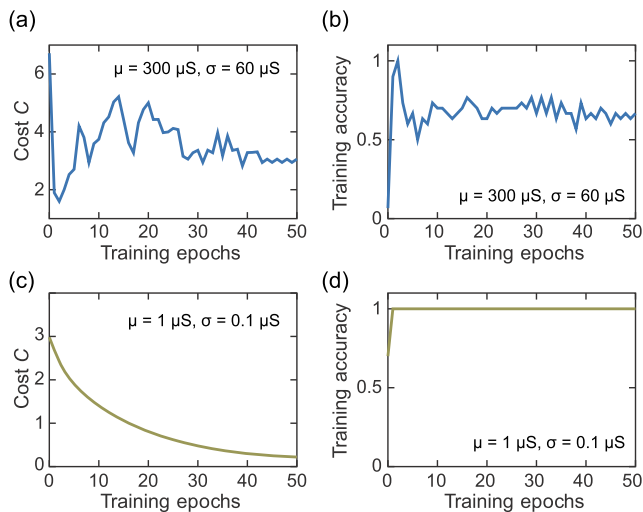


FIG. 5. [(a) and (b)] Training performance resulting from weight initialization with parameters $\mu = 300 \mu\text{S}$ and $\sigma = 60 \mu\text{S}$. [(c) and (d)] Training performance resulting from weight initialization with parameters $\mu = 1 \mu\text{S}$ and $\sigma = 0.1 \mu\text{S}$. In these simulations, the memristive synapses were initialized each with a randomly drawn filament radius r_{fil} from a truncated normal distribution with $r_{\text{fil,min}} = 1 \text{ nm}$, $r_{\text{fil,max}} = 20 \text{ nm}$, $\mu_{\text{fil}} = 7 \text{ nm}$, and $\sigma_{\text{fil}} = 2 \text{ nm}$.

distribution with $r_{\text{fil,min}} = 1 \text{ nm}$ and $r_{\text{fil,max}} = 20 \text{ nm}$ denoting the respective minimum and maximum filament radii, respectively. These choices of filament radii are consistent with experimental work.^{45–47} The upper limit was chosen in order not to exceed cell dimensions (feature size $F = 40 \text{ nm}$). The mean and standard deviation were chosen to be $\mu_{\text{fil}} = 7 \text{ nm}$ and $\sigma_{\text{fil}} = 2 \text{ nm}$, respectively. Again, simulations were performed using both previously discussed initialization regimes: the standard initialization regime and the lower conductance regime ($\mu = 1 \mu\text{S}$ and $\sigma = 0.1 \mu\text{S}$). The resulting training performance indicators are visualized in Fig. 5. The results validate our previous results. For regions of low conductance state density, i.e., the standard initialization regime, stable operation cannot be ensured [Figs. 5(a) and 5(b)]. While now cells with small diameters can be compensated with cells with a wide filament, the occurrence of resistance gaps still exists and may deteriorate the network performance. Using an initialization with $\mu = 1 \mu\text{S}$, the cells are initialized in a region with high state density. Here, comparable performance to the continuous model is achieved [Figs. 5(c) and 5(d)]. Nevertheless, it is still necessary to adjust parameter β_0 to ensure stable training operation.

CONCLUSION

In summary, a Verilog A model of an ECM ReRAM was derived by combining two well-established ECM models, which incorporates the discrete conduction steps that are observed in the experiment at low current levels. It is used to investigate the influence of discrete conduction steps on the training accuracy of a simple one-layer ECM-based ANN using a backpropagation learning algorithm based on the steepest descent method. The simulation results show that noncontinuous resistance levels will deteriorate the

network performance in comparison to devices with completely analog behavior. The convergence of the system will be either difficult or not achieved at all. Here, we studied a one-layer network. The problem of convergence is expected to become more severe for networks with multiple layers. The problem of noncontinuous resistance levels will occur when scaling down the size of the device to the $10 \times$ nanometer scale. In addition, it will appear if the current levels, and thus the filament dimensions, are getting very small. Thus, it is necessary to consider noncontinuous resistance levels when designing a neural network in ultrascaled technology nodes. The simulation results also showed how to cope with this problem. First, all the memristive devices need to be initialized in a region of a high state density. To this end, the cells need to be characterized before initialization. In addition, some algorithm parameters can be adapted to improve the convergence. The last opportunity would be to increase the amount of synapses in order to achieve quasicontinuous resistance levels, but this comes with the expense of higher area and power consumption of the network. Besides the noncontinuous resistance levels, fluctuations can occur due to the intrinsic switching variability leading to bigger or smaller conductance change during a weight update. Nevertheless, the resistance change will be due to individual jumps of atomic defects, leading to noncontinuous resistance levels for very small filament dimensions. Thus, the results should still be valid when variability is considered. The switching variability will lead to additional problems for the learning process. The impact of the variability will be addressed in a future study. Finally, additional experimental results on ultrascaled ECM devices operated in this narrow filament regime will ultimately be needed to confirm the conclusions of this paper.

SUPPLEMENTARY MATERIAL

See [supplementary material](#) for a detailed description of the used ECM models and background information on the 1-layer neural network and the learning algorithm.

ACKNOWLEDGMENTS

This work was supported in part by the Deutsche Forschungsgemeinschaft under Grant No. SFB 917 and also in part by the European Research Council through the European Union's Horizon 2020 Research and Innovation Program through the Project MNEMOSENE under Grant No. 780215.

REFERENCES

- Y. LeCun, Y. Bengio, and G. Hinton, *Nature* **521**, 436 (2015).
- D. S. Jeong, R. Thomas, R. S. Katiyar, J. F. Scott, H. Kohlstedt, A. Petraru, and C. S. Hwang, *Rep. Prog. Phys.* **75**, 76502-1 (2012).
- L. O. Chua, *Appl. Phys. A* **102**, 765 (2011).
- I. Valov, E. Linn, S. Tappertzhofen, S. Schmelzer, J. van dan Hurk, F. Lentz, and R. Waser, *Nat. Commun.* **4**, 1771 (2013).
- M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, and E. Eleftheriou, *Nat. Electron.* **1**, 246 (2018).
- S. E. Dreyfus, *J. Guid., Control, Dyn.* **13**, 926 (1990).
- M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, *Nature* **521**, 61 (2015).
- P. Gu, B. Li, T. Tang, S. Yu, Y. Cao, Y. Wang, and H. Yang, in *IEEE 20th Asia and South Pacific Design Automation Conference (ASP-DAC)* (IEEE, 2015).

- ⁹S. B. Eryilmaz, D. Kuzum, S. Yu, and H. S. P. Wong, in *Proceedings of the IEEE International Electron Device Conference (IEDM2015)* (IEEE, 2015).
- ¹⁰S. Yu, P. Y. Chen, Y. Cao, L. Xia, Y. Wang, and H. Wu, in *Proceedings of the IEEE International Electron Device Meeting (IEDM 2015)* (IEEE, 2015), p. 451.
- ¹¹D. J. Wouters, R. Waser, and M. Wuttig, *Proc. IEEE* **103**, 1274 (2015).
- ¹²S. Park, A. Sheri, J. Kim, J. Noh, J. Jang, M. Jeon, B. Lee, B. Lee, B. Lee, and H. Hwang, in *2013 IEEE International Electron Devices Meeting* (IEEE, 2013), p. 25.6.1.
- ¹³M. Prezioso, I. Kataeva, F. Merrikh-Bayat, B. Hoskins, G. Adam, T. Sota, and K. Likharev, in *Proceedings of the IEEE International Electron Device Meeting (IEDM 2015)* (IEEE, 2015).
- ¹⁴R. Waser, R. Dittmann, G. Staikov, and K. Szot, *Adv. Mater.* **21**, 2632 (2009).
- ¹⁵I. Valov, M. Luebben, A. Wedig, and R. Waser, *ECS Trans.* **75**, 27 (2016).
- ¹⁶Y. Li, S. Long, Y. Liu, C. Hu, J. Teng, Q. Liu, H. Lv, J. Sune, and M. Liu, *Nanoscale Res. Lett.* **10**(1), 420 (2015).
- ¹⁷W. Yi, S. E. Savel'ev, G. Medeiros-Ribeiro, F. Miao, M. Zhang, J. J. Yang, A. M. Bratkovsky, and R. S. Williams, *Nat. Commun.* **7**, 11142-1 (2016).
- ¹⁸S. Tappertzhofen, E. Linn, S. Menzel, A. J. Kenyon, R. Waser, and I. Valov, *IEEE Trans. Nanotechnol.* **14**, 505 (2015).
- ¹⁹S. Brivio, J. Frascaroli, E. Covi, and S. Spiga, *Sci. Rep.* **9**, 6310 (2019).
- ²⁰T. Tsuruoka, T. Hasegawa, K. Terabe, and M. Aono, *Nanotechnology* **23**, 435705 (2012).
- ²¹R. Degraeve, A. Fantini, N. Raghavan, L. Goux, S. Clima, Y. Chen, A. Belmonte, S. Cosemans, B. Govoreanu, D. Wouters, P. Roussel, G. Kar, G. Groeseneken, and M. Jurczak, in *Proceedings of International Symposium on Physical and Failure Analysis of Integrated Circuits (IPFA)* (IEEE, 2014), p. 245.
- ²²S. Long, X. Lian, C. Cagli, X. Cartoixa, R. Rurali, E. Miranda, D. Jiménez, L. Perniola, M. Liu, and J. Suñé, *Appl. Phys. Lett.* **102**, 183505 (2013).
- ²³K. Terabe, T. Hasegawa, T. Nakayama, and M. Aono, *Nature* **433**, 47 (2005).
- ²⁴A. Siemon, S. Ferch, S. Menzel, R. Waser, and E. Linn, in *15th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, Dresden, Germany, 23–25 August 2016.
- ²⁵S. Menzel, S. Tappertzhofen, R. Waser, and I. Valov, *Phys. Chem. Chem. Phys.* **5**, 6945 (2013).
- ²⁶M. Luebben, S. Menzel, S. G. Park, M. Yang, R. Waser, and I. Valov, *Nanotechnology* **28**, 135205-1 (2017).
- ²⁷S. Menzel, B. Wolf, S. Tappertzhofen, I. Valov, U. Böttger, and R. Waser, in *6th IEEE International Memory Workshop (IMW)*, Taipei, Taiwan (IEEE, 2014), p. 42.
- ²⁸S. Menzel, *J. Comput. Electron.* **16**, 1017 (2017).
- ²⁹A. Nayak, T. Tsuruoka, K. Terabe, T. Hasegawa, and M. Aono, *Nanotechnology* **22**, 235201-1 (2011).
- ³⁰S. Tappertzhofen, I. Valov, and R. Waser, *Nanotechnology* **23**, 145703 (2012).
- ³¹M. Arita, Y. Ohno, Y. Murakami, K. Takamizawa, A. Tsurumaki-Fukuchi, and Y. Takahashi, *Nanoscale* **8**, 14754 (2016).
- ³²Y. Yang, P. Gao, L. Li, X. Pan, S. Tappertzhofen, S. Choi, R. Waser, I. Valov, and W. D. Lu, *Nat. Commun.* **5**, 4232-1 (2014).
- ³³S. Menzel, U. Böttger, and R. Waser, *J. Appl. Phys.* **111**, 014501-1 (2012).
- ³⁴W. Chen, N. Chamele, Y. Gonzalez-Velo, H. J. Barnaby, and M. N. Kozicki, *IEEE Electron Device Lett.* **38**, 1244 (2017).
- ³⁵S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, and H. P. Wong, *IEEE Trans. Electron Devices* **58**, 2729 (2011).
- ³⁶S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, *Nano Lett.* **10**, 1297 (2010).
- ³⁷T. Chang, S.-H. Jo, K.-H. Kim, P. Sheridan, S. Gaba, and W. Lu, *Appl. Phys. A* **102**, 857 (2011).
- ³⁸D. Mahalanabis, H. J. Barnaby, Y. Gonzalez-Velo, M. N. Kozicki, S. Vrudhula, and P. Dandamudi, *Solid-State Electron.* **100**, 39 (2014).
- ³⁹J. Frascaroli, S. Brivio, E. Covi, and S. Spiga, *Sci. Rep.* **8**, 7178-1 (2018).
- ⁴⁰K. Moon, A. Fumarola, S. Sidler, J. Jang, P. Narayanan, R. M. Shelby, G. W. Burr, and H. Hwang, *IEEE J. Electron Devices Soc.* **6**, 146 (2018).
- ⁴¹W. Wu, H. Wu, B. Gao, N. Deng, S. Yu, and H. Qian, *IEEE Electron Device Lett.* **38**, 1019 (2017).
- ⁴²M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R. S. Williams, J. J. Yang, Q. Xia, and J. P. Strachan, *Adv. Mater.* **30**, 1705914-1 (2018).
- ⁴³D. S. Jeong and C. S. Hwang, *Adv. Mater.* **30**, 1704729-1 (2018).
- ⁴⁴I. Kataeva, F. Merrikh-Bayat, E. Zamanidoost, and D. Strukov, in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015.
- ⁴⁵U. Celano, L. Goux, A. Belmonte, K. Opsomer, R. Degraeve, C. Detavernier, M. Jurczak, and W. Vandervorst, *J. Phys. Chem. Lett.* **6**, 1919 (2015).
- ⁴⁶M. Kudo, M. Arita, Y. Ohno, and Y. Takahashi, *Appl. Phys. Lett.* **105**, 173504 (2014).
- ⁴⁷M. Arita, Y. Ohno, and Y. Takahashi, *Phys. Status Solidi A* **213**, 306 (2016).