IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

(V)TEAM for SPICE Simulation of Memristive Devices with Improved Numerical Performance

DALIBOR BIOLEK[1,2], (Senior Member, IEEE), ZDENĚK KOLKA[3], (Member, IEEE), VIERA BIOLKOVÁ[3], (Member, IEEE), ZDENĚK BIOLEK[1,2], and SHAHAR KVATINSKY[4], (Senior Member, IEEE)

[1]Department of Electrical Engineering, University of Defence, 662 10 Brno, Czech Republic
[2]Department of Microelectronics, Brno University of Technology, 616 00 Brno, Czech Republic
[3]Department of Radio Electronics, Brno University of Technology, 616 00 Brno, Czech Republic
[4] Viterbi Faculty of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel

Corresponding author: Dalibor Biolek (dalibor.biolek@unob.cz).

This work was supported in part by the Czech Science Foundation under Grant 20-26849S, and in part by the Infrastructure of K217UD, Brno, Czech Republic.

ABSTRACT The paper introduces a set of models of memristive devices for a reliable, accurate and fast analysis of large networks in the SPICE (Simulation Program with Integrated Circuit Emphasis) environment. The modeling starts from the recently introduced TEAM (ThrEshold Adaptive Memristor Model) and VTEAM (Voltage ThrEshold Adaptive Memristor Model). A number of improvements are made towards the stick effect elimination and other numerical refinements to make the analysis of large networks fast and accurate. A set of models are proposed that utilize the synergy of several techniques such as window asymmetrization, integration with saturation, state equation preprocessing, scaling, and smoothing. The performance of models is tested in Cadence PSPICE 17.2 and particularly in HSPICE v2017, the latter on a large-scale CNN (Cellular Nonlinear Network) for detecting edges of binary images. The simulations manifest the usability of developed models for fast and reliable operation in networks containing more than one million nodes.

INDEX TERMS Memristor, VTEAM, window function, stick effect, SPICE, Cellular Nonlinear Network.

I. INTRODUCTION

In 2013, a model for current-controlled memristive devices, called TEAM (ThrEshold Adaptive Memristor Model), was published in [1]. Two years later, an analogous VTEAM (Voltage ThrEshold Adaptive Memristor Model) was proposed in [2], which exhibits a voltage threshold, not current threshold.

The (V)TEAM falls into the category of phenomenological models, which are fitted to real memristive devices via tweaking a set of parameters. In other words, the "fabrication details" of the device are stamped onto experimental data which are used for setting the parameters of the universal (V)TEAM model.

The development of the TEAM model started from the Pickett model [3], also denoted as Simmons's Tunnel Barrier Model. Its port (resistive) equation consistently starts from the physical model of the tunnel effect between electrodes separated by a thin insulating film [4]. Although its state equations are phenomenological, not physical, they model accurately enough the large dynamic range of the time derivative of the tunnel barrier width of this memristive device.

The Pickett model is rightly regarded as the reference model of the TiO_2 memristor with a high predictive ability, since its physical parameters are fitted to the behavior of a real-world device. Paradoxically, this strong point has also negative implications: the numerical problems during simulations, caused by the complexity of equations, and by the impossibility of modeling memristors operating on different physical principles than the reference TiO_2 memristive device.

Developing the TEAM and consequently the VTEAM was motivated by three aims, namely: 1. To simplify the Pickett model and thus improve its robustness in the environment of numerical simulation programs. 2. To make the models more universal such that they can be fitted to various real-world memristors via a set of parameters. 3. To provide the required accuracy of the simulations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

It was shown in [1] and particularly in [2] that the TEAM and VTEAM meet the above requirements and that they can be fitted with a high accuracy to previously proposed but more complex and numerically exacting physical models of a variety of current memristive devices.

The Verilog-A and Matlab codes of the models are available in [5]. The Verilog-A codes are written with a view to achieving fast computation. It is noted in [1] that "A Verilog-A model …is more efficient in terms of computational time than a SPICE macromodel…". On the other hand, the simple Euler method with a fixed time step is used for the integration of differential equations, with the well-known impact on accuracy. The same applies to the models in MATLAB. On the other hand, the Verilog-A enables partial corrections of this way generated numerical errors via methods that cannot be directly used in SPICE, such as resetting the state variable if it overflows/underflows its limit value due to some inaccuracy. A similar approach is used in [5] for resolving a fundamental numerical problem associated with the window functions in (V)TEAM – the stick effect, but at the expense of accuracy.

This work deals with the implementation of (V)TEAM in the environment of SPICE simulation programs with the aim of providing robust and fast simulations that utilize sophisticated numerical algorithms. The results described below will make the universal models of memristive devices accessible to a wide community of SPICE users.

The paper has the following structure: In Section II the facts about the (V)TEAM are summarized and potential problems associated with their implementation in current simulation programs are identified. Several methods of effective solution of the stick effect are suggested in Section III. Section IV describes further procedures of numerical improvements. The relevant versions of the models are subjected to an exacting test for the accuracy and speed, with subsequent evaluation. The source codes of the selected models are available to potential users.

II. (VOLTAGE) THRESHOLD ADAPTIVE MEMRISTOR MODEL

The current-voltage relationship of the (V)TEAM model can be chosen arbitrarily to best accommodate the behavior of concrete modeled device. In [1] and [2], two simple models are suggested, which consider the linear or exponential dependence of the memristance R_M and the state variable w:

$$R_M(w) = R_{on} + (R_{off} - R_{on})\frac{w - w_{on}}{w_{off} - w_{on}} \tag{1}$$

or

$$R_M(w) = R_{on}\left(\frac{R_{off}}{R_{on}}\right)^{\frac{w - w_{on}}{w_{off} - w_{on}}} \tag{2}$$

where R_{on}, R_{off} and w_{on}, w_{off} are the limit values of the memristance and the state variable w in the ON and OFF regime.

Crucial for the (V)TEAM model are the state equations, which have the same form for both the TEAM and the VTEAM. Given below are the equations for the VTEAM (the equations for the TEAM are generated via the (v, v_{on}, v_{off}) and (i, i_{on}, i_{off}) replacement):

$$\frac{dw}{dt} = \begin{cases} k_{off}\left(\dfrac{v}{v_{off}} - 1\right)^{\alpha_{off}} f_{off}(w), & 0 < v_{off} < v \quad \text{(OFF)} \\ 0, & v_{on} < v < v_{off} \quad \text{(subTHR)} \\ k_{on}\left(\dfrac{v}{v_{on}} - 1\right)^{\alpha_{on}} f_{on}(w), & v < v_{on} < 0 \quad \text{(ON)} \end{cases} \tag{3}$$

Here v is the voltage across the device, v_{on}, v_{off} are the threshold voltages, k_{on}, k_{off}, α_{on}, α_{off} are the fitting parameters, and $f_{on}(w)$, $f_{off}(w)$ are the window functions whose role is to preserve the state variable within the physically realistic limits $w \in (w_{on}, w_{off})$. It is indicated on the right side of (3) that the state equation has different forms for the OFF, ON, and subTHReshold regime of operation.

It is demonstrated in [2], [5] that the above model can be fitted with high accuracy to various experimental memristive devices even for the symmetric window functions, namely for

$$f_{on}(w) = f_{off}(w) = f(w) \tag{4}$$

The Verilog-A and MATLAB codes of the TEAM and VTEAM models containing the well-known rectangular, Joglekar, Prodromakis, Biolek and Kvatinsky windows used in (3) and (4) are given in [5]. It should be noted that the last two windows are asymmetric, thus they are not governed by (4). The Kvatinsky windows, designed to approximate the complex state equations of the Pickett model, are in the form

$$f_{off}(w) = \exp\left[-\exp\left(\frac{w - a_{off}}{w_c}\right)\right]$$
$$f_{on}(w) = \exp\left[-\exp\left(\frac{a_{on} - w}{w_c}\right)\right] \tag{5}$$

where a_{off}, a_{on}, and w_c are parameters assumed from the Pickett model.

The utilization of the VTEAM model in conventional simulation programs consists in the classical numerical integration of the differential equation (3), starting from the initial condition $w(0)$, and in subsequently computing the memristance from the known state variable. The corresponding block diagram is shown in Fig. 1. The numerical integration is prone to well-known numerical errors [6], which can be intensified just via windowing. Note that the windowing is a numerically problematic operation, acting in the schematic in Fig. 1 in the feedback loop together with the block of integrator.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

It is demonstrated in [7] that the utilization of window functions in state equations of ideal generic memristors can result in fatal numerical errors. The substance of these errors consists in the co-existence of two types of memory in the real-world memristor, which is modeled by an ideal generic memristor: The memory, represented by the so-called Natural state variable (the charge or the flux), which is boundless in principle, and its physical implementation, represented by a physical state variable, for example the width of conductive channel w, which has its physical limits. If one-to-one correspondence between the natural and physical state variables exists, then the corresponding model exhibits all the fingerprints of ideal generic memristor. However, if the physical memory reaches its physical limit, its state is fixed, and its link to the natural state variable is interrupted. An interesting consequence of this phenomenon is described in [7]: The numerical solution of the simulation task is quite different from the analytic solution. It is alarming that this hidden error can be revealed only after confronting the simulation results from a physical viewpoint or with the analytic solution if it is available for simple simulation tasks.

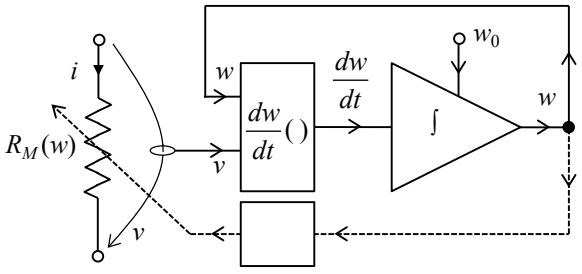

FIGURE 1. Block diagram of the original VTEAM model with Eqs. (1), (2), (3).

To avoid this problem and to speed-up the simulation of networks containing ideal generic memristors, a method for state-space transformation is developed in [8]. This method also prevents another problem, known as the stick effect, which is associated with the limit states of the physical state variable.

Luckily for the (V)TEAM, the problem of the above hidden errors does not apply, since these models are not classified as ideal generic memristors. The corresponding extended memristors cannot be described via charge-flux constitutive relations, and therefore the problem of natural versus physical state variable does not exist here. However, it is obvious from the state equation (3) and from the diagram in Fig. 1 that the stick effect in unavoidable in (V)TEAM for an arbitrary symmetric window (4), which takes zero values at boundaries (such as the rectangular, Joglekar and Prodromakis window), so the method of state space transformation can play an important role in such cases. On the contrary, the Biolek window, designed in [9] with discontinuities at boundary states or the Kvatinsky window prevents the stick effect reliably. It is obvious that such a discontinuity, frequently regarded as disadvantage, is in fact an effective tool for avoiding such a problem.

Since the VTEAM is fitted in [2] to several real-world memristive systems just by using symmetric windows (5), most frequently the rectangular window, the stick effect must be resolved. It can be read from the source codes in [5] that the solution may be via a parameter called in [5] p_window_noise. The corresponding algorithm slightly modifies the behavior of the window function depending on the direction of the memristor current. For example, if the state variable approaches its upper limit w_{off}, then, after crossing the level w_{off} – p_window_noise, the value p_window_noise is systematically subtracted in the Euler integration formula from the computed value of the state variable until w leaves the boundary region (w_{off} – p_window_noise, w_{off}). This procedure is applied accordingly also for preventing the stick effect near the lower limit w_{on}. A similar remedy is suggested in [10], [11], when a small number, added to the window function, ensures that the right side of the state equation is not multiplied by zero at the boundaries. The limit states can also be provided via controlled switches [12] or by tools of behavioral modeling [13]. An interesting solution consists in the replacement of the window function by an integrator with artificial limitation of its boundary states via diode limiters [10]. The low quality of such a limitation, caused by the specific shape of the current-voltage characteristics of diodes, is a potential disadvantage. Classical behavioral modeling of ideal diodes can lead to numerical problems.

Other potential numerical problems are hidden in the state equation (3): Model discontinuities during the transition of memristor voltage through threshold levels, multiplication of extremely large and small numbers, and the variables taking extremely large or small values that can overflow or underflow the SPICE limits. The inspiration for resolving these issues may be found in the general procedures described *e.g.* in [6].

The following Section suggests three effective methods of avoiding the stick effect: window asymmetrization, integration with saturation, and analytic pre-processing of state equations, also denoted below as the transformation of state variables.

III. AVOIDING STICK EFFECT

A. WINDOW ASYMMETRIZATION

Although the utilization of standard symmetric window functions (4) in the state equations (3) leads to simple models, all these window functions, which are zero at the boundaries of the state variables, inevitably cause the stick effect if the memristor acquires one of its limit states. This effect is perhaps most evident when trying to get a maximum simplification of the model by using the rectangular window.

On the other hand, the stick effect can be easily removed via asymmetric window functions. Experiments show that when the symmetric window is replaced by an asymmetric

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

window, for example the Biolek window function, then, after some prospective trimming of model parameters, the "small-signal" behavior of both models is identical, and the simulation of the "hard-switching" regime is robust and without the stick effect.

Asymmetrization of standard window functions is trivial. For example, the simplest rectangular window

$$f(w) = \sigma(w - w_{on}) - \sigma(w - w_{off}) \tag{6}$$

where σ is the Heaviside step function, can be made asymmetric as follows:

$$f_{off}(w) = \sigma(w_{off} - w), \quad f_{on}(w) = \sigma(w - w_{on}) \tag{7}$$

Equation (7) means that in the OFF state, when the state variable w increases towards the limit value w_{off}, the window function f_{off} is zeroing if this limit is reached, and the state variable stops moving. After changing the state to ON, the window function changes to f_{on} which attains the value 1 for the limit state w_{off}. That is why the state variable unsticks from this boundary and moves back towards the state w_{on}.

A typical implementation of (3) into the simulation program consists in the utilization of a pair of current sources i_{off} and i_{on} in parallel, which charge the integrating capacitor. The OFF or ON source is controlled by the corresponding formula (3) and is active only for the respective values of the driving voltage, thus for $v>v_{off}$ or $v<v_{on}$. It can be accomplished via multiplying the current by a logic function, which provides the value 1 for $v>v_{off}$ and 0 for the remaining cases, or the value 1 for $v<v_{on}$ and 0 otherwise. The asymmetric rectangular window can be then introduced by a similar logic function, which sets to zero the derivative of the state for $w>w_{off}$ or $w<w_{on}$:

$$i_{off} = k_{off}\left(\frac{v}{v_{off}} - 1\right)^{\alpha_{off}} (v > v_{off})\,(w < w_{off})$$
$$i_{on} = k_{on}\left(\frac{v}{v_{on}} - 1\right)^{\alpha_{on}} (v < v_{on})\,(w > w_{on}) \tag{8}$$

The prospective numerical problems caused by the discontinuities in (8) can be resolved via the smoothing techniques described in Section IV.

B. INTEGRATION WITH SATURATION

There are two methods of how to maintain the state variable within given limits: Either by interrupting the driving signal of the integrator when the limit state is reached, or via limiting the integrator states independently of its excitation. The first method is used for windowing, the second can be implemented by modelling the saturation of the integrator output.

The integration with saturation is advantageous in the sense that it avoids the stick effect and simultaneously prevents the state variable from swinging out of its natural limits due to numerical errors. This method is mentioned in several works, *e.g.* in [10]. An example of SPICE implementation is given in Fig. 2. Modeling the diodes may be an issue, since the diodes should behave as ideal switches with zero threshold voltage and zero voltage drop in ON state. The experiments described in Section V led to the model of ideal diode in Fig. 2. The model is based on the SPICE modeling of a voltage-controlled switch whose resistance R_S depends on the control voltage v_c according to a smooth function, thus with the continuous derivative dR_S/dv_c.

With reference to the dynamic range of the number representation in the program, the ratio between the switch resistances in the OFF and ON states should be chosen below 10^{12} [14]. In the case of convergence problems, the difference between the V_{on} and V_{off} values should be increased, and the corresponding voltage drop on the diodes can be compensated by the respective modifications of the levels w_{on} and w_{off}.

The questionable windowing can be avoided, if the integration with saturation is utilized. On the other hand, both methods can be combined in order to model the boundary effects more precisely, but the window function cannot be zero at boundaries [10]. Basically, various types of shaping the current-voltage characteristics of diodes might be used for such a modeling.

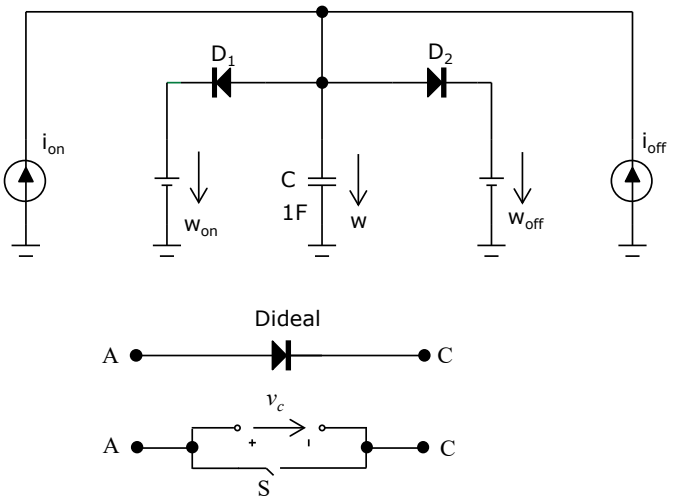

FIGURE 2. Spice implementation of integrator with saturation. The diodes should behave as ideally as possible but not causing numerical problems. Here, the diode switches between the OFF and ON states within a narrow region of voltages, from 0V (OFF) to 100mV (ON). The switch model syntax may vary in different SPICE simulators.

C. STATE EQUATION PREPROCESSING

The state equation (3) can be rewritten in the form

$$\frac{dw}{dt} = g(v)\,f(w, sign(v)) \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

where *sign*(*v*) is the sign function, and

$$g(v) = k(v)\left(\frac{v}{v_t(v)} - 1\right)^{\alpha(v)}, \tag{10}$$

$$f(w,1) = f_{off}(w), \quad f(w,-1) = f_{on}(w), \tag{11}$$

$k(v)$ *is* k_{off}, k_{on} , 0 for OFF, ON, subTHR state,

$v_t(v)$ is v_{off}, v_{on} for OFF, ON state,

$\alpha(v)$ is α_{off}, α_{on} for OFF, ON state.

The window function and the voltage-dependent parameters v_t, α can be for the subTHR state defined in an arbitrary but mathematically correct way.[1]

It is obvious from (4), (10) and (11) that the original VTEAM model can work with the classical window functions, which depend only on the state (see, for example, the Joglekar or Prodromakis window), but it generally takes windowing to be dependent on the signal direction (a typical example is the Biolek window).

Symmetric windowing

In the first step, consider the simple case of symmetric windows (4). Then the state equation (9) holds:

$$\frac{dw}{dt} = g(v)f(w). \tag{12}$$

Re-arranging and integrating both sides of (12) yield

$$\int_{w_0}^{w} \frac{dw}{f(w)} = G(w,t) = F(w) - F(w_0), \tag{13}$$

where

$$G(w,t) = \int_{t_0}^{t} g(v)dt \tag{14}$$

is the integral of $g(v)$ with respect to time, $F(w)$ is the primitive function of the function $1/f(w)$, w_0 is the initial state at time t_0, and $-F(w_0)$ is a constant of integration.

Figure 3 shows the relationship between the window function and $F(w)$. The graph of $F(w)$ is drawn for an arbitrarily chosen constant of integration. The Figure illustrates the condition where the operating point starts moving from the initial state $w = w_0$ (point ❶) towards point ❷. According to (13), after reaching this point, the F function changes by the value of the integral G (14).

[1] Since $k = 0$ for the subTHR state, the window function can be defined as dependent on the sign of voltage. Alternatively, the window function could have been defined as zero for the subTHR state, with $g(v)$ playing no role.

It can be easily verified that the F function, which corresponds to the rectangular window, is in the form of the piece-wise-linear function (dashed line in Fig. 3).

As demonstrated in [8], the general smooth window function, which is a smooth approximation of the rectangular window, is transformed into an F function which is monotonically increasing within the interval (w_{min}, w_{max}) (solid curves in Fig. 3). Then the inverse *FI*( ) of the F function from Fig. 3 exists, and (13) can be rewritten in the form

$$w = FI\left(G + F(w_0)\right) \tag{15}$$

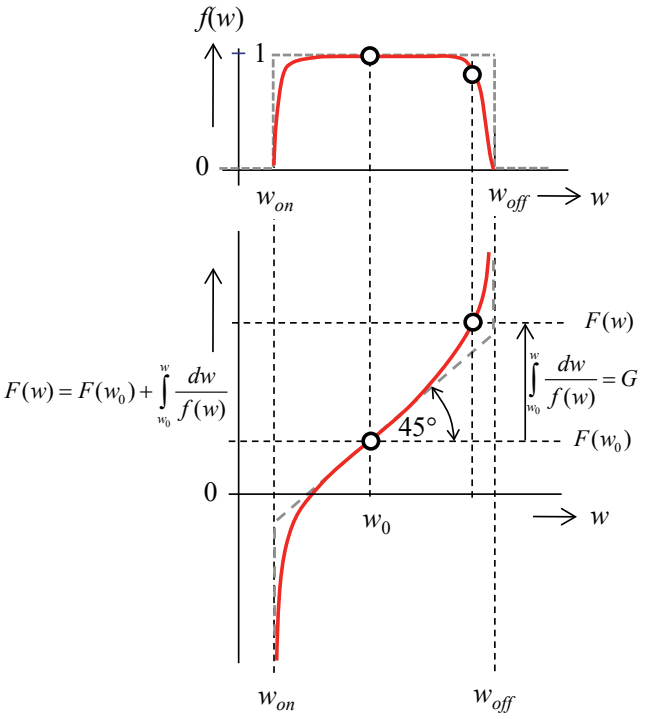

FIGURE 3. Window function *f*(*w*) and the corresponding function *F*(*w*).

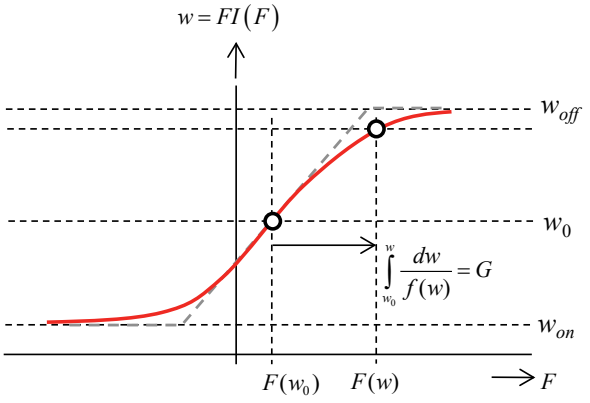

FIGURE 4. *FI* functions as an alternative to windowing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

It is obvious from Fig. 4 that, when the function *FI*( ) is known, it can be used for computing the state variable w directly from the variable G. The initial condition w_0 must be known for this computation as well as the original function $F(\,)$.

The F functions for the Joglekar and Prodromakis window functions $f(w)$ are derived in [8]. The *FI* functions for general window functions can be found via formula (13) and the subsequent inversion of the F function. In addition, a universal sigmoidal approximation of the parametric *FI* function is proposed in [8].

Analytical expressions of F functions for several windows are summarized in Table I. The symbols ${}_2F_1$ and Ei denote the hypergeometric functions and the exponential integral. The F function for the Prodromakis window cannot be derived in terms of known mathematical functions.

TABLE I
F FUNCTIONS FOR SELECTED WINDOW FUNCTIONS.

$f(w)$	$F(w)$
Joglekar $1-(2w-1)^{2p}$	$(2w-1)\,{}_2F_1\left[\frac{1}{2p},1;1+\frac{1}{2p},1-f(w)\right]$
Biolek off $1-(w-1)^{2p}$	$(w-1)\,{}_2F_1\left[\frac{1}{2p},1;1+\frac{1}{2p},1-f_{off}(w)\right]$
Biolek on $1-w^{2p}$	$w\,{}_2F_1\left[\frac{1}{2p},1;1+\frac{1}{2p},1-f_{on}(w)\right]$
Kvatinsky off $\exp(-\exp(\frac{w-a_{off}}{w_c}))$	$w_c\mathrm{Ei}\left(-\ln(f_{off}(w))\right)$
Kvatinsky on $\exp(-\exp(\frac{a_{on}-w}{w_c}))$	$-w_c\mathrm{Ei}\left(-\ln(f_{on}(w))\right)$

The concept of using *FI* functions instead of classical window functions leads to the original procedure of evaluating the VTEAM model in Fig. 1 being modified to the block diagram in Fig. 5. The voltage v is transformed into $g(v)$ via the algebraic equation (10) and then integrated with the initial condition computed as $F(w_0)$. The resulting waveform G is transformed via algebraic *FI* function (15) into the state variable w, which controls the memristance by (1) or (2). This transformation requires a corresponding shift of the G waveform, so $F(w_0)$ is added to G according to Fig. 5.

Recall the consequence of the original state equations (3): if the window function attains the value zero, the state variable w stops moving at this moment. In the modeling diagram in Fig. 5, the variable w is derived from another state variable G, which is the result of integrating an auxiliary variable g. If the memristor reaches its boundary state and the state variable w is stopped, it is necessary to interrupt the integration of variable g, thus the movement of the state G must be stopped. Otherwise, the model would behave as a memristor with an infinite depth of memory, which is a feature of ideal memristor. In combination with the given form of the differential equation (3) of VTEAM, it would imply an infinite increase in the state variable G, because the function g in the integral (14) evolves in time as asymmetric, thus with a nonzero DC component. This finding points out the unsuitability of the given method for modeling memristors with window functions of type (5), which decrease to zero asymptotically. The "stop at boundaries" operation in Fig. 5 cannot be implemented via interrupting the signal $g(v)$, which drives the integrator, since it would cause definitive sticking of the integrator state. More convenient is introducing the saturation limits of the integrator, which are computed to be in coincidence with the limit values of the state variable w.

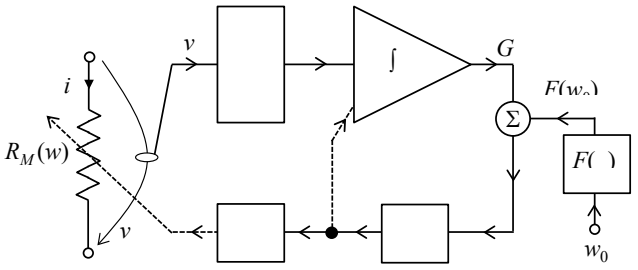

FIGURE 5. Block diagram of the simulation using the *F* and *FI* functions.

Comparing Fig. 5 with Fig. 1, we can conclude that the integration of the differential equation is now provided by the integrator in a feedforward configuration. The only feedback is of the "stop at boundaries" type, which can be provided by the integration with saturation. On the other hand, for the memristor with a simple rectangular window, the diagram in Fig. 5 can be reduced to the initial diagram in Fig. 1, because the F and *FI* functions are linear within the limits w_{on} and w_{off}, namely $F(x) = x$. The model from Fig. 5 can therefore be advantageous for stick effect elimination for window functions of general forms with zero values at boundaries.

Asymmetric windowing

Now consider a more general asymmetric windowing (11), when the window function depends on the direction of memristor voltage or current.

Fig. 6 illustrates a sketch of f_{on} and f_{off} functions that apply to positive $v > v_{off}$ and negative $v<v_{on}$, respectively. The arrows denote the corresponding directions of changing the state variable w, or, the sign of the time derivative of the state variable in (3). For positive $v > v_{off}$, the operating point moves in the (f,w) coordinates from the initial point ❶ with $w = w_0$ to the right along the $f_{off}(w)$ curve, and the state variable w increases towards its upper limit w_{off}. Simultaneously, the time-integral G according to (14) increases along the blue curve *"off"*. Fig. 6 shows that at the moment when the state variable reaches the value of w and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

the voltage decreases below v_{off} threshold the operating point stops moving in the (F, w) coordinates at point ❷, and after the voltage decreases below v_{on}, the operating point switches to the position ❸ on the f_{on} curve and moves along it back towards w_{on}. After reaching the position ❹, the memristor switches to the off state and the operating point moves to the position ❺.

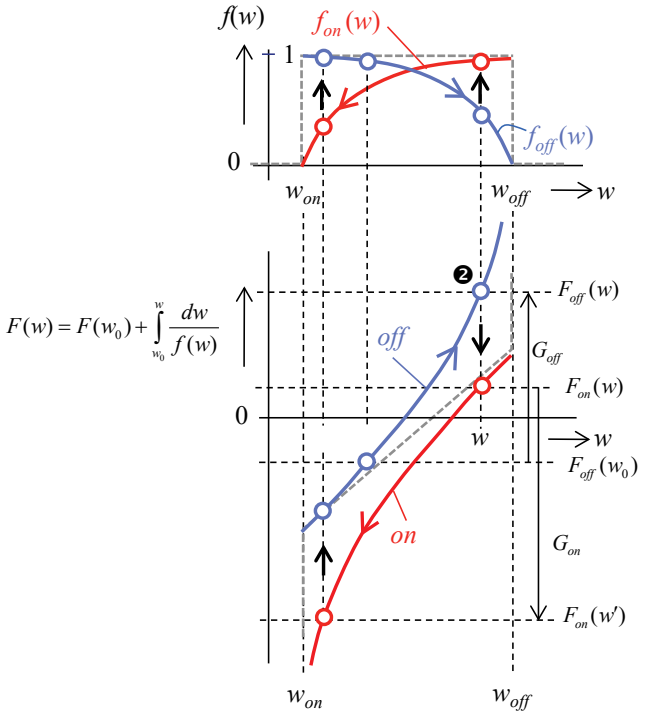

FIGURE 6. Window functions $f_{on}(w)$, $f_{off}(w)$, and the corresponding F functions of the *on* and *off* types.

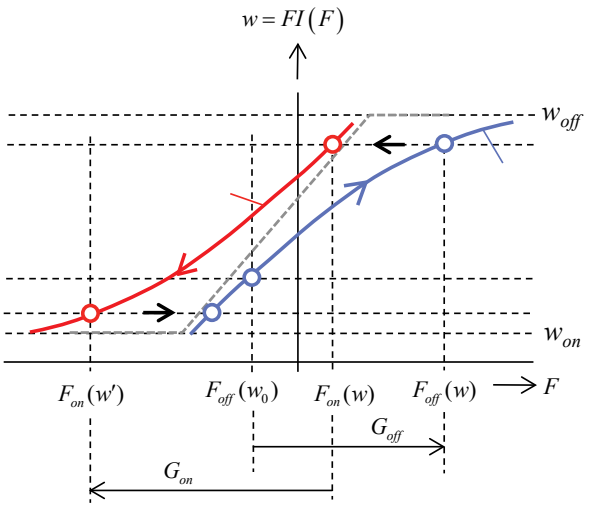

FIGURE 7. Principle of *off-on* switching: In the *off* regime, the operating point proceeds along the blue curve in the arrow direction. At the point (G_x, w_x), the voltage decreases below v_{on} and the memristor switches from the *off* to the *on* regime. The operating point continues to move back along the shifted red curve.

The trajectory of the operating point in the (F, w) coordinates (see Fig. 6) or also in the (w, F) coordinates (see Fig. 7) is characterized by discontinues changes of the F function during off-on and on-off switching. Simultaneously, however, the G variable changes continuously: the move of the operating point from position ❶ to ❷ and from position ❸ to ❹is accompanied by a change of the variable G by the values of G_{off} and G_{on}, where

$$G_{off} = \int_{w_0}^{w} \frac{dw}{f(w)}, G_{on} = \int_{w}^{w'} \frac{dw}{f(w)} \tag{16}$$

Since

$$G_{off} + G_{on} = \int_{w_0}^{w'} \frac{dw}{f(w)}, \tag{17}$$

the variable G does not change when switching from point ❷ to point ❸. The same holds for switching from the ON state to the OFF state. The continuous nature of the G variable is also obvious from the fact that it is integrated from the signal $g(v)$.

The block diagram in Fig. 5 is modified for asymmetric windowing according to Fig. 8.

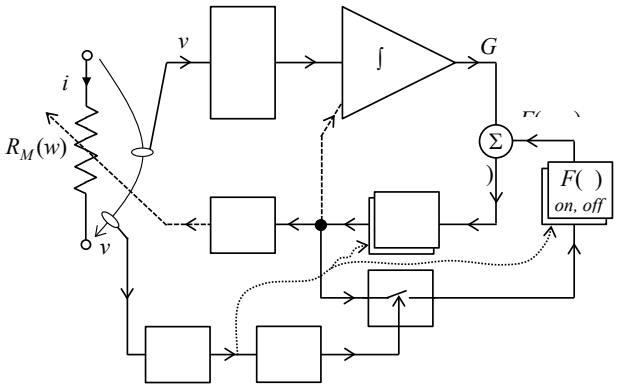

FIGURE 8. Block diagram of the simulation for the case of asymmetric windowing.

The immediate state of the memristor is evaluated via a comparison of the memristor voltage and the threshold levels v_{off} and v_{on} in the block "state on-off". The value -1 or 1 is assigned to the ON state or to the remaining OFF states and subTHR according to (3). This state then controls the utilization of the transforming functions F and FI in computation, and the value of state is also used for evaluating the time instant of the state change in the block "state change". This evaluation is crucial for computing the auxiliary variable w_{ini} (see Fig. 8), which is the last coordinate of the "changeover points" in Fig. 6 (for example w_0 for point ❶ or w for points ❷ and ❸) and which is the starting coordinate for the operating point moving along the $F(w)$ characteristic as a consequence of the evolution of the variable G. The value of w_{ini} is evaluated via the following algorithm:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

$$w_{ini} = \begin{cases} w_0 & \text{initialization} \\ w & \text{state change} \\ last\ value & \text{otherwise} \end{cases} \quad (18)$$

At the beginning of the simulation (*time* = 0), the initial condition w_0 is assigned to w_{ini}. Every change of the state from ON to OFF or from OFF to ON, which is identified in the "state change" block in Fig. 8, is accompanied by sampling the state variable w, and the "old" value w_{ini}.is rewritten by this sample. When the state is not changed, the variable w_{ini} is not changed either.

The operation of the "state change" block can be accomplished via the logic operation

$$state\_change = (state \neq last(state)) \quad (19)$$

If the values of *state* are different in the current and previous simulation steps (i.e. if the state was changed), then *state_change* =1, otherwise *state_change* = 0.

Finding the values of waveforms in the previous simulation step is not trivial in SPICE programs working with adaptive time step in the transient analysis. Starting with the version 16.5, the function STATE can be used in ORCAD PSpice. A similar function LAST is available in Micro-Cap. HSPICE does not provide such a possibility. The cross function can be used in Verilog-A. It detects the state change and enforces the time step of analog solver in the vicinity of the point of discontinuity.

IV. OTHER NUMERICAL IMPROVEMENTS

A. SCALING, LOG-ANTILOG

A commonly used format on PCs is the double-precision binary floating-point with number representation from 2^{-1022} to 2^{1023}, thus from about ca 10^{-308} to 10^{308}. This range can be violated during the execution of some mathematical operations, for example the evaluation of the window functions (5), when $\exp(-\exp(-6.6)) = 5.6521\times10^{-320}$. In reality, the layout may be even worse, because various simulation programs apply stricter limits to internal variables or voltages and currents. For example, the smallest nonzero positive number which PSPICE and HSPICE can process is 10^{-30}. The corresponding argument x of the above function $\exp(-\exp(x))$ is 4.235. For higher value of this argument, the window function returns zero.

Even worse results can be expected if the results of the computations, represented by high numbers, are evaluated as the voltages or currents of controlled sources. The PSpice limits for voltages and currents are 10^{10} V and 10^{10} A, and their overflow leads to the termination of the simulation run due to convergence problems. These limits are rather higher for HSPICE, LTspice, and Micro-Cap.

The fact that PSPICE and HSPICE handle incorrectly parameters smaller than 10^{-30} in magnitude should be taken into account when working with physical quantities that frequently appear in physical models of memristive devices, such as the electron mass (9.109×10^{-31} kg) or the Planck constant (6.626×10^{-34} Js).

For the simulation of (3), the above issues can be resolved via classical procedures known as scaling and log-antilog: The quantities which, in principle, take extremely low or high values, particularly those which are represented via voltages or currents, must be scaled in order to fit them within acceptable limits. A typical example is the width of the conductive channel of the TiO_2 memristor, which is in the range of units of nanometers. The scaling of the state variable also affects its time derivative, and therefore it must be accompanied by the corresponding scaling of the right side of the differential equation (3). Another example is the integration of the differential equation in SPICE via charging the capacitor from a current source, whose current is given by the right side of (3). The dynamic range of this current should be scaled to values that match the typical set of error criteria. Concretely for the state equation (3) it means to multiply the coefficients k_{on} and k_{off} by a proper scaling factor, and to divide the original capacitance (1 Farad) of the integrating capacitor by the same factor.

The right side of (3) is given as a product of three terms; the last two of them can significantly change during the simulation run. For example, the term $(v/v_{off}-1)^{\alpha off}$ can take extremely high values whereas the multiplying window function can produce values near zero, potentially beyond the numerical limits of the program. In such cases, it is useful to compute the logarithms of such terms, which on principle cannot be zero (for example the window functions (5)), to sum them and perform the inverse exp operation.

B. SMOOTHING

Equations (8) of VTEAM for the rectangular window can be easily generalized to arbitrary window functions as follows:

$$i_{off} = k_{off}\left(\frac{v}{v_{off}}-1\right)^{\alpha_{off}} f_{off}(w)\ (v > v_{off})$$
$$i_{on} = k_{on}\left(\frac{v}{v_{on}}-1\right)^{\alpha_{on}} f_{on}(w)\ (v < v_{on}) \quad (20)$$

The discontinuous functions modelling the threshold property introduce discontinuities into the simulated waveforms and become a potential source of numerical problems. It is therefore useful to replace them by their smooth approximations, thus by continuous functions with continuous first derivatives. Experiments described in Section V confirmed the usefulness of the smooth functions *LTH* and *HTL* (Low-To-High and High-To-Low) as the replacement of the discontinuous functions ($v>v_{off}$) and ($v<v_{on}$) from (20):

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

$$LTH(x)=\begin{cases}0, & x<-0.5\\ 2\times 4^{-(x+1)(x-0.5)^2}-1, & -0.5\le x\le 0.5\\ 1, & x>0.5\end{cases} \tag{21}$$

$$HTL(x)=\begin{cases}1, & x<-0.5\\ 2\times 4^{(x-1)(x+0.5)^2}-1, & -0.5\le x\le 0.5\\ 0, & x>0.5\end{cases}$$

where the auxiliary variable

$$x=\frac{v-\frac{v_L+v_H}{2}}{v_H-v_L} \tag{22}$$

is derived from the voltage v and the limits v_L and v_H. The functions (21) change their values continuously for the voltage v within the interval between v_L and v_H, not discontinuously at the point $v = v_{off}$ or $v = v_{on}$.

The corresponding smoothing (21), (22) can also be used as a remedy for other discontinuities described in Section III.

V. BENCHMARK TESTING

A. METHODOLOGY OF TESTING

The models were tested in two steps: First, each model was analyzed with sinusoidal driving voltage in Cadence PSpice 17.2. Then, the models successfully passing the first test were used for testing in a large CNN (Cellular Nonlinear Network) for detecting edges in the image according to [15]. The HSPICE v2017 was used for the simulation of network containing 5000 to 200,000 memristive cells.

The VTEAM parameters were fitted to the memristor used in [15]. The results are summarized in Table II.

TABLE II
VTEAM PARAMETERS OF MEMRISTOR FROM [15].

parameters	values
R_{on} [kΩ]	2
R_{off} [kΩ]	10
R_{ini} [kΩ]	5
α_{on} [-]	3
α_{off} [-]	3
v_{on} [V]	0.8
v_{off} [V]	-0.8
k_{on} [μm/s]	1
k_{off} [μm/s]	-1

Since $v_{on} > 0$, $v_{off} < 0$, it is obvious that the memristor from [15], in contrast to the original equations (1) - (3), decreases/increases its memristance for positive voltage $v > v_{on}$ / negative voltage $v < v_{off}$. This can be taken into consideration by formally arranging the state equation (3) such that the ON / OFF state corresponds to a voltage higher than v_{on} / lower than v_{off} [2].

The port equation was in the form of (1) with scaled limits of the state variable $w_{on} = 0$, $w_{off} = 1$, $w_{ini} = 0.375$.

The Joglekar and Biolek windows, used in several models, were considered with the parameter $p = 10$. The corresponding nonlinear functions $F(w)$ for these windows were computed in Matlab for their prospective implementation in Spice as table-based functions. Also their inverses $FI(F)$ were derived. However, since such functions cannot be directly imported into HSPICE, their analytical approximations were found, and the corresponding formulae are summarized in Table III. Finally, they were used for simulations in HSPICE and also in PSpice.

Simulations were run on the HP Z840 Workstation, Intel® CPU E5-2690 v3 @ 2x2.60GHz, 512 GB RAM, Windows 10 Pro.

TABLE III
F AND FI FUNCTIONS FOR MODELING SELECTED SYMMETRIC AND ASYMMETRIC WINDOW FUNCTIONS *).

$F(w)$	$FI(F)$
symmetric windows	
$\frac{1}{2a}\ln\left(\frac{\sinh(2aw)}{\sinh(2a(1-w))}\right)$	$0.5+\frac{1}{4a}\ln\left(\frac{\cosh(a(1+F)}{\cosh(a(1-F))}\right)$
asymmetric windows, off	
$\frac{1}{2a}\ln\left(\frac{\sinh(a(1+w))}{\sinh(a(1-w))}\right)$	$\frac{1}{2a}\ln\left(\frac{\cosh(a(1+F))}{\cosh(a(1-F))}\right)$
asymmetric windows, on	
$\frac{1}{2a}\ln\left(\frac{\sinh(aw)}{\sinh(a(2-w))}\right)$	$1+\frac{1}{2a}\ln\left(\frac{\cosh(a(1+F))}{\cosh(a(1-F))}\right)$

*) $a = 10.3$ for Joglekar (symmetric) and Biolek (asymmetric) window with $p = 10$.

B. TESTING UNDER SINUSOIDAL EXCITATION

The following 10 models were tested:

R:
Model employing asymmetric rectangular window (8).
R_d:
Model with symmetric rectangular window and integrator with saturation, implemented by ideal diodes with parameters according to Fig. 2.
R_s:
Modified R model with smooth approximations (21), (22). Parameters for smoothing $k(v)$ from (10) are: $v_L = v_{on}$ - 0.1V, $v_H = v_{on}$ (HTL), $v_L = v_{off}$, $v_H = v_{off}$ +0.1V (LTH), parameters for smoothing the rectangular window: $v_L = 0$V, $v_H = 1$V (HTL), $v_L = 0.9$V, $v_H = 1$V (LTH).
R_t:
Model with the rectangular window, implemented by F and FI transformations according to Fig. 5.
J:
Model employing the Joglekar window and the integrator with saturation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

J_s:
Modified J model with smoothing discontinuities in $k(v)$ from (10) and parameters from R_s model.
J_t:
Model with the Joglekar window, implemented by F and FI transformations according to Fig. 5.
B:
Model with the Biolek window and the classical integrator without saturation.
B_s:
Modified B model with smoothing discontinuities in $k(v)$ from (10) and parameters from R_s model.
B_t:
Model with the Biolek window, implemented by F and FI transformations according to Fig. 8.

All the models were driven by sinusoidal voltage with various amplitudes and repeating frequencies. Regarding a high value of the parameter $p = 10$ used in the window functions, the models exhibited a good match in the simulated responses. The results of transient analysis for the B and B_t models are shown in Fig. 9.

The performance of models was evaluated under their unified excitation via 2V/10 Hz sinusoidal voltage. Simulation time, accuracy, model sensitivity to simulation options (vntol, Gmin, step ceiling), and nonconvergence liability were compared. Table IV summarizes the results of transient runs within 10 repeating periods of the driving signal with a maximum timestep of 0.5ms and skipbp directive.

TABLE IV
MODEL COMPARISON, SINUSOIDAL EXCITATION.

VTEAM model	wall time [ms]	accuracy	sensitivity	converg. problems
R	90	low	high	no
R_d	50	*)	low	no
R_s	190	high	high	no
R_t	60	high	low	no
J	80	moderate	moderate	no
J_s	140	moderate	moderate	no
J_t	80	high	low	no
B	80	high	low	no
B_s	160	high	low	no
B_t	330	low	high	yes

*) The inaccuracy of state variable limits, caused by the threshold voltages of diodes in Fig. 2, can be compensated via adjusting the w_{on} and w_{off} values.

Concerning the simulation time, the fastest is the R_d model, closely followed by the R_t and J, J_t, and B models. A common issue of the R and R_s models with rectangular window is the difficulty of preserving the accuracy and, simultaneously, low sensitivity to simulation options. The weak point of the classical R model (asymmetric window without smoothing) is related to an inaccurate evaluation of the condition $w < w_{off}$ or $w > w_{on}$ in (8). Consequently, the limit values of the state variable are computed incorrectly. The accuracy can be increased via decreasing the step ceiling but at the cost of slowing down the simulation. The speed of the computation strongly depends on the analysis options (*e.g.* vntol=0.1m, Gmin=1e-14 lead to a forty-fold deceleration of the transient analysis in comparison to the standard settings).

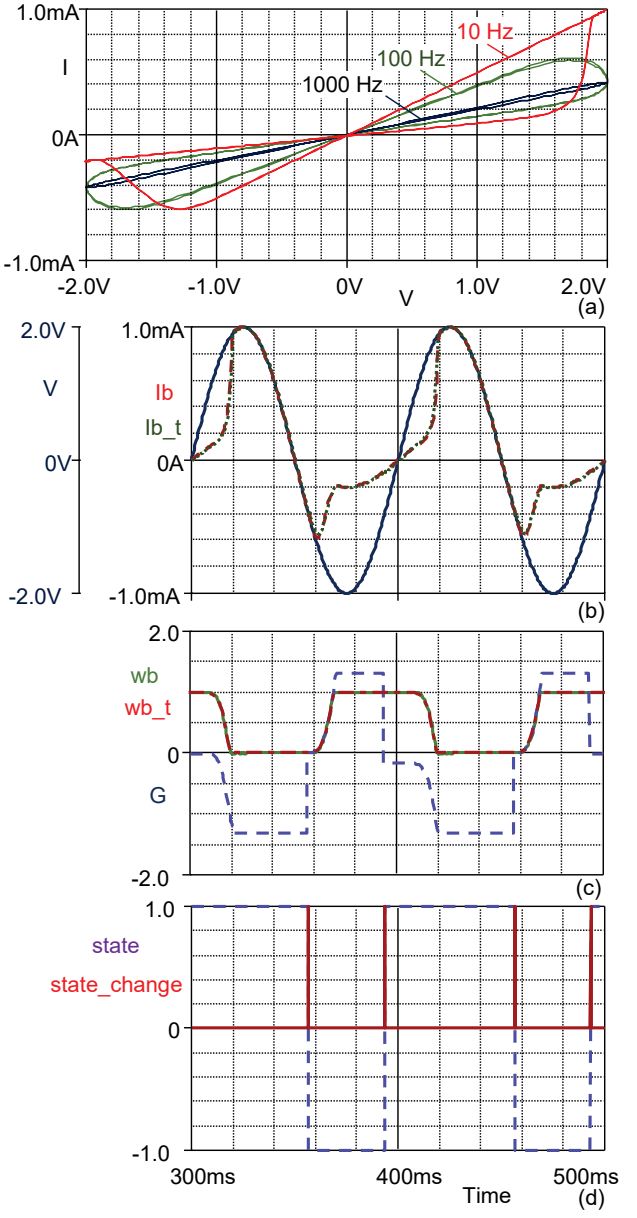

FIGURE 9. Results of the transient analysis of memristor B- and B_t models under sinusoidal voltage excitation; (a) *v-i* hysteresis loops of B model for various frequencies, (b) waveforms of memristor voltage and current, (c) waveform of state variable wb of B model and state variable wb_t and variable G (14) of B_t model, (d) waveforms of state and state_change (19) signals.

The R_d model is very fast, with a low sensitivity of the analysis speed to analysis options. The threshold voltage of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

diodes can be relatively high (ca 100mV), but then the saturation levels of the integrator should be adjusted to exact values via a modification of the voltages Vhigh and Vlow. Decreasing the diode threshold leads to a significant deceleration of the analysis, which can be eliminated in part via a proper selection of the analysis options (particularly vntol and Gmin).

For a proper operation of the R_s model, double smoothing of discontinuities in the state equation (threshold property and state variable limitations) must be performed. The model is then accurate, with a low sensitivity to the analysis options. Though the simulation times are longer than without smoothing, the behavior is robust and the precision of computation is high even for a higher difference of the levels v_H and v_L defining the width of the transition zone in the denominator of (22).

The R_t model is very accurate and fast, and it is not necessary to specially set the simulation options for its optimal behavior. From this point of view, it can be evaluated as the most suitable model for a memristor with a rectangular window.

The results of testing all the variants of models with the Joglekar window can be generalized to memristor models with different symmetric windows which are described via unambiguous polynomial functions of the state variable (such as the Prodromakis window and others): To avoid the stick effect, the classical variant requires a diode limiter of the integrator output, which slows down the simulation for diodes with a relatively low threshold voltage. Although the model based on the transformation of the state variable does not use this direct limitation, the simulation times can be extended due to evaluating the nonlinear functions F and FI, particularly if they are modeled as tables. In the end, however, the J and J_t models are comparable to the best models with a rectangular window in terms of speed and accuracy, and the J_s model performs even better than the R_s model does.

The B and B_s models can also be evaluated positively. The natural advantage of the Biolek window is that it does not cause the stick effect, so it is not necessary to take measures against it, which usually leads to slowdowns in calculations or to a reduction in accuracy.

Convergence problems were recorded in the B_t model, i.e. in the model using asymmetric windowing, implemented by the algorithm from Fig. 8. It turns out that the core of the problem lies in the need to accurately identify the moment of change of the memristor state according to (19) and subsequent sampling of the state variable w for updating the initial condition w_{ini}. The SPICE programs are not equipped with tools for an accurate identification of the time instant of occurrence of an event (*e.g.* changes in the memristor state), as in VERILOG-A, for example, and so memristor models with asymmetric windows according to Fig. 8 should be implemented on other platforms, *e.g.* HSPICE with the possibility of utilizing the VERILOG code.

Figure 9 (a) shows the simulation of the pinched hysteresis loops of a memristor with the Biolek window (B) driven by a sinusoidal signal with an amplitude of 2V and various repeating frequencies. The Figures 9 (b), (c), (d) demonstrate simulations with the B and the B_t model for a frequency of 10 Hz. The results are in full agreement with the knowledge that the state variable w cannot be changed in the subthreshold region, that is to say when the driving voltage is within the interval $(v_{off}, v_{on}) = (–0.8\text{V}, + 0.8\text{V})$. Figure 9 (d) summarizes the waveforms of the auxiliary quantities state and state_change. It is obvious that, in this case, the state of the memristor changes at the moments when the state variable w is practically constant. This contributes to alleviating the convergence problems that may occur when sampling a state variable that is changing dramatically over time. The accuracy of the simulation of the B_t model is strongly related to the setting of the saturation levels of the integrator, the output of which is the quantity G (see Fig. 8). For a correct operation of the model B_t it is necessary to carefully set the error criterion VNTOL so that it is at least one order of magnitude lower than the difference in the threshold levels of the control voltage of the switch in Fig. 2 in the diode model, which is used to limit the output of the respective integrator. Even so, we do not avoid convergence problems that can occur due to changes in the conditions outside the model, for example when the network of interacting components is changed or the driving signal parameters are modified.

Therefore, all models from Table IV apart from the B_t model advance to the subsequent stage of testing in large CNNs.

C. TESTING IN LARGE CNNs

The suitability of the designed models for the simulation of large memristive networks was tested via a CNN (Cellular Nonlinear Network), which allows a simple scaling of the simulation problem complexity. The used EDGE M-CNN from [15] can detect edges in a binary image (with black or white pixels only). The network consists of cells organized in a two-dimensional lattice corresponding to the image pixels. Each cell is connected with the nearest neighbors as shown in Fig. 10 (a) for a cell at the position (i, j).

Each cell has an input node connected to a constant-voltage source, whose voltage represents the pixel state (white ≡ -1V, black ≡ +1V). In the benchmark network, each cell is connected with inputs and outputs of its eight nearest neighbors as shown in Fig. 10 (a). The relative position of the neighboring cells will be denoted by two indices $(k, l) \in \{–1,0,1\}$, while the combination (0,0) is the cell itself.

Figure 10 (b) shows the circuit implementation of each cell. The "memcomputing core" [15] consists of a controlled current source i_x, constant current source I_z, capacitor C_x, resistor R_x, and memristor M_x. The current source $i_{x\,i,j}$ of a cell at the position (i, j) is controlled by its own input and output and by inputs and outputs of the neighboring cells

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

$$i_{x_{i,j}} = \sum_{k,l=-1}^{1} a_{k,l} v_{y_{i+k,j+l}} + \sum_{k,l=-1}^{1} b_{k,l} v_{u_{i+k,j+l}} \quad (23)$$

where v_u and v_y are the input and output voltages, respectively, and $a_{k,l}$ and $b_{k,l}$ are transconductances. The elements of the core form a dynamical system of the second order.

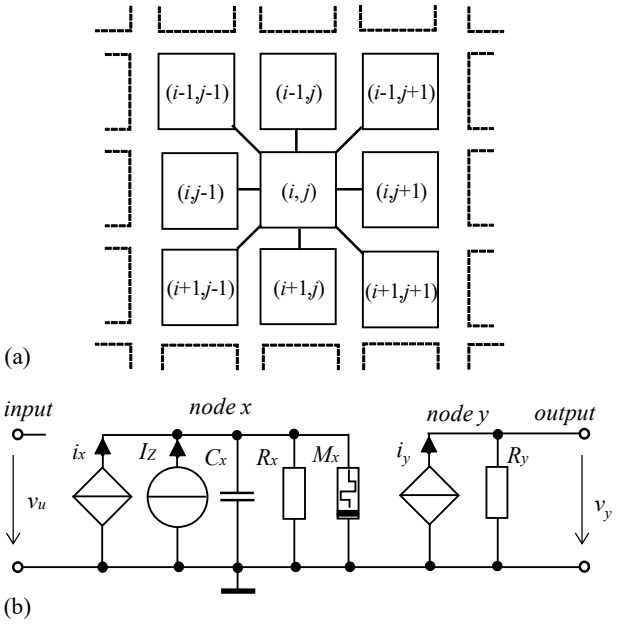

FIGURE 10. (a) Structure of CNN. A cell (*i*, *j*) with connections to its nearest neighbors is drawn with a solid line. (b) Circuit realization of the cell.

The output block consists of a resistor R_y and a current source controlled by the x-node

$$v_y = R_y g_y(v_x) \quad (24)$$

where

$$g_y(v_x) = 0.5 g_{lin}\left(|v_x + v_{sat}| - |v_x - v_{sat}|\right) \quad (25)$$

is a saturation function.

The parameters of the CNN [15] are summarized in Table V. The parameters of memristor models are the same as in Table II.

The simulations in PSPICE 17.2 have some limitations: The analysis of a CNN with 50,000 memristors takes about 1316 seconds, and an attempt to analyze the network with 100,000 cells leads to the analysis being terminated by "Aborting simulation (Symbols Table overflow)". Therefore, the simulations were performed with a more robust HPP solver of HSPICE v2017.

For automated generation of HSPICE input files, a MATLAB script was used. The test image is resampled to the desired resolution and converted to a binary image. Let the image pixel size be $M \times N$. The CNN cells are generated only for internal pixels, i.e. the number of cells will be $(M-2)(N-2)$. The pixels of the first and last rows and columns are used as "neighbor" inputs in (23). If the cell output voltage is in the negative saturation (see (24)), the cell has detected an edge, while positive saturation means "no edge" [15]. The simulator output is read back by the MATLAB script and presented as an image. Fig. 11 shows an example of the process.

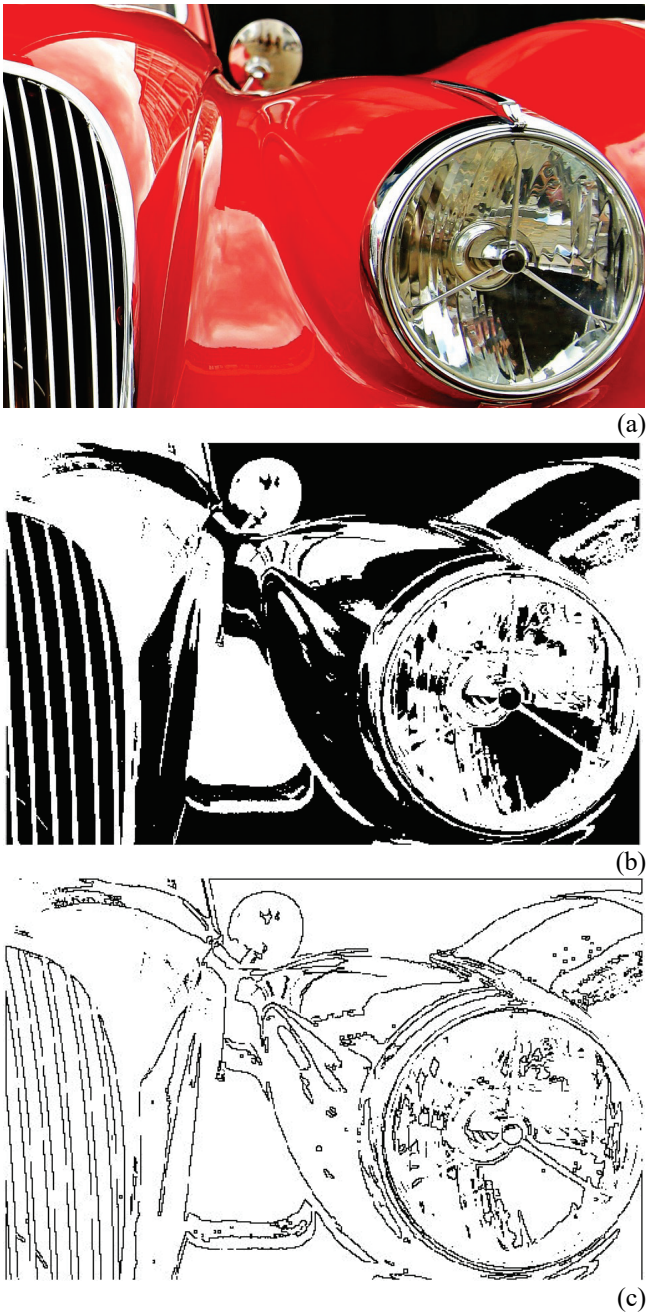

FIGURE 11. Example of CNN simulation. The original image -a vintage-car mask [16] (a) - was resampled to a resolution of 356×563 pixels and converted to binary image (b). After 10 ms transient analysis, the CNN found edges (c). The initial conditions of all the 199510 memristor cells were: $v_x(0) = 0$ V, $M_x(0) = 5$ kΩ.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

TABLE V
CNN PARAMETERS FROM [15].

parameters	values
C_x [μF]	10
R_x [kΩ]	1
I_z [mA]	-0.1
$a_{0,0}$ [mS]	1.675
$a_{k,l}$ [mS], $(k,l)\neq(0,0)$	0
$b_{0,0}$ [μS]	805
$b_{k,l}$ [mS], $(k,l)\neq(0,0)$	-0.1
R_y [kΩ]	1
g_{lin} [mS]	1
V_{sat} [V]	0.1

Figure 12 shows the dependence of the simulation times (the wall time of transient analysis) on the number of cells in the CNN for all the memristor models being tested. The transient analysis was performed with the parameter Tmax = 10ms. It is clear from the graph that the models can be divided into two groups in terms of achieved simulation times, represented by red and yellow lines: the B, B_s, and R models (fast), and the J, J_s, J_t, R_t, R_d models (longer simulation times). The R_s model can be classified as fast, except for the CNN consisting of about 50,000 cells, where the simulation times approach a slower variant of the models.

The simulation of a CNN with half a million cells is already beyond the capabilities of HSPICE, where simulation runs are terminated prematurely without any error message. Since the submodel of each memristor in the CNN cell represents 4 to 6 nodes for the simulator, a network with 500,000 cells represents a simulation network with 2-3 million nodes.

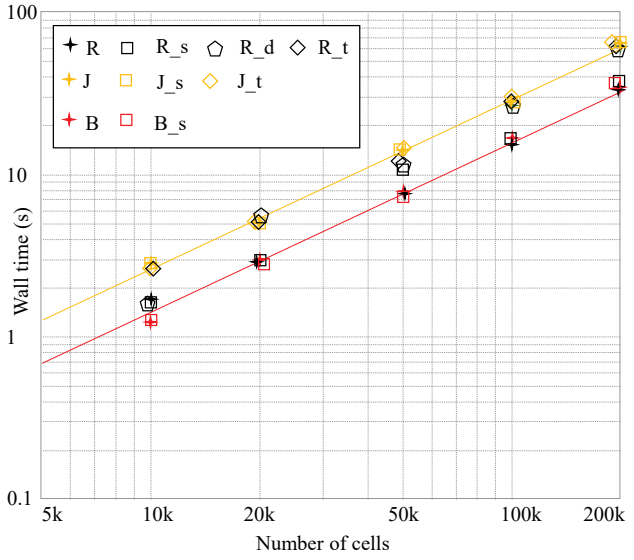

FIGURE 12. The computation time of transient analyses vs the number of cells for various memristor models.

VI. CONCUSION

The mathematical model of VTEAM [2] was optimized for implementation in SPICE simulation programs. The results are fully transferable for the dual TEAM model [1]. Attention was paid to the solution of the stick effect. Several techniques were used, including the technique of state variable transformation. These techniques were combined with methods of smoothing the discontinuities in state equations. Several variants of the model were tested for accuracy, speed, and reliability in Cadence PSpice and HSPICE. In the HSPICE environment, the models were subjected to demanding tests in extremely large memristive networks.

The results show 9 variants of models that are robust and stable when simulating networks with hundreds of thousands of CNN cells employing memristors. These models can be divided into two categories, the so-called fast and slower variants. "Fast" models use asymmetric windows (Biolek and asymmetric rectangular windows), which in principle do not need additional aids to prevent the stick effect. Smoothing techniques in themselves improve the convergence properties, but, on the other hand, they slow down the calculations. It has also been shown that the operation of smoothing algorithms is sensitive to the setting of transient analysis parameters and error criteria. The testing of models for speed, accuracy and the above-mentioned sensitivity revealed that the least advantageous models were models with classic symmetrical windows of the Joglekar type. If, due to a numerical error, there is even a "slight" overflow or underflow of the state variable outside the allowed region of the window function, the derivative of the state variable is calculated from the values of the window function outside this region, which can lead to errors. This can be partially reduced by appropriately redefining the window function outside the allowed area of the state variable, which, however, causes other complications - slowing down the calculations and a greater tendency to the stick effect. Thus, it clearly turns out that symmetric windowing should be either avoided or implemented by the method of transforming the state variables. Even after this transformation, however, the J_t model is slower than the models based on asymmetric windows (*e.g.* the B model).

Furthermore, it has become clear that models with the transformation of state variables cannot be successfully implemented in the SPICE environment for asymmetric windows, because the SPICE standard does not allow an effective identification of an event associated with a discontinuity in time (specifically the time at which this discontinuity occurs). The relevant B_t model was therefore only tested in Cadence PSpice 17.2 with the provisional use of the "state" function. In the next step, it will be interesting to implement this model as an embedded Verilog-A code in HSPICE and then perform its testing in a multi-memristor structure according to the methodology used in this work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059241, IEEE Access

The models that passed the tests with best results (B, B_s, R, R_s) work reliably and quickly in networks with 200k CNN cells (in fact, the extent of the simulated circuits is only given by the computer hardware and the limits of the HSPICE program), and their optimized SPICE codes therefore may be useful for a wide variety of researchers involved in simulations of large memristor circuits.

The procedures from this work can be applied not only to VTEAM but also to other known models of memristors. In particular, the stick effect elimination is a challenge associated with all models that use symmetric window functions in their differential equations of motion. The proposed method of state variable transformation is directly applicable to the General Hyperbolic Sine Model [17] and the Yakopcic model [18], whose state equations represent the product of nonlinear functions of the state variable and the voltage or current.

REFERENCES

[1] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: ThrEshold Adaptive Memristor Model," *IEEE Trans Circ Syst I*, vol. 60, no. 1, pp. 211–221, 2013.

[2] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "VTEAM: A general model for voltage-controlled memristors," *IEEE Trans Circ Syst II*, vol. 62, no. 8, pp. 786–790, 2015.

[3] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams, "Switching dynamics in titanium dioxide memristive devices," *J. of Applied Physics*, vol. 106, p. 074508, 2009.

[4] J. Simmons, J. "Electric tunnel effect between dissimilar electrodes separated by a thin insulating film," *Journal of Applied Physics*, vol. 34, no. 9, pp. 2581–2590, 1963.

[5] S. Kvatinsky, K. Talisveyberg, D. Fliter, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Models of memristors for SPICE simulations," *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, Eilat, Israel, pp. 1–5, 2012.

[6] D. Biolek, M. Di Ventra, and Y. V. Pershin, "Reliable SPICE simulations of memristors, memcapacitors and meminductors," *Radioengineering*, vol. 22, no. 4, pp. 945–968, 2013.

[7] D. Biolek, Z. Biolek, V. Biolkova, and Z. Kolka, "Modeling of TiO2 memristor: from analytic to numerical analyses," *Semicond. Sci. Technol*., vol. 29, 125008 (5pp), 2014.

[8] D. Biolek, Z. Biolek, V. Biolkova, and Z. Kolka, "Reliable modeling of ideal generic memristors via state-space transformation," *Radioengineering*, vol. 24, no. 2, pp. 393–407, 2015.

[9] Z. Biolek, D. Biolek, and V. Biolkova, "SPICE model of memristor with nonlinear dopant drift," *Radioengineering*, vol. 18, no. 2, pp. 210–214, 2009.

[10] S. Shin, K. Kim, and S-M. Kang, Compact models for memristors based on charge-flux constitutive relationships," *IEEE Trans Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 4, pp. 590–598, 2010.

[11] K. Eshraghian, O. Kavehei, K-R. Cho, J. M. Chappell, A. Iqbal, S. F. Al-Sarawi, and D. Abbott, "Memristive device fundamentals and modeling: Applications to circuits and systems simulation," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1991–2007, 2012.

[12] Á, Rák and G. Cserey, "Macromodeling of the memristor in SPICE," *IEEE Trans Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 4, pp. 632–636, 2010.

[13] F. García, M. López-Vallejo, and P. Ituero, "Building memristor applications: from device model to circuit design," *IEEE Transactions on Nanotechnology*, vol. 13, no. 6, pp. 1154–1162, 2014.

[14] PSpice A/D Reference Guide. Cadence, Product Version 16.0, June 2007.

[15] A. Ascoli, R. Tetzlaff, S-M. Kang and L. O. Chua, "Theoretical Foundations of Memristor Cellular Nonlinear Networks: A DRM_2-Based Method to Design Memcomputers With Dynamic Memristors," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 8, pp. 2753–2766, Aug. 2020, doi: 10.1109/TCSI.2020.2978460.

[16] https://cdn.pixabay.com/photo/2016/08/07/11/50/jaguar-1576109_960_720.jpg

[17] M. Laiho, E. Lehtonen, A. Russel, and P. Dudek, "Memristive synapses are becoming a reality," *The Neuromorphic Engineer*, College Park, MD: Inst. Neuromorphic Eng., 2010.

[18] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, "A Memristor Device Model," *IEEE Electron Device Letters*, vol. 32, no. 10, pp. 1436–1438, 2011.

DALIBOR BIOLEK (M'99-SM'15) received the M.Sc. degree in Electrical Engineering from Brno University of Technology, Czech Republic, in 1983, and the PhD. degree in Electronics from the Military Academy Brno, Czech Republic, in 1989.

He is currently with the Department of EE, University of Defence Brno (UDB), and with the Department of Microelectronics, Brno University of Technology (BUT), Czech Republic. His scientific activity is directed to the areas of general circuit theory, frequency filters, mem-systems, and computer simulation of electronic systems. At present, he is professor at BUT and UDB in the field of Theoretical Electrical Engineering.

Prof. Biolek is a member of the CAS/COM Czech National Group of IEEE. He has been a member of editorial boards of international journals including the International Journal of Electronics and Communications (AEU) and Electronics Letters.

ZDENĚK KOLKA (M'02) received the MSc degree in 1992 and the PhD degree in 1997, both in electrical engineering, from Brno University of Technology (BUT), Czech Republic. In 1995, he joined the Department of Radio Electronics, Brno University of Technology. His scientific activity is directed to the areas of general circuit theory, computer simulation of electronic systems and digital circuits. For years, he has been engaged in algorithms of the symbolic and numerical computer analysis of electronic circuits. He has published over 100 papers. At present, he is a professor at BUT in the field of radio electronics.

VIERA BIOLKOVÁ (M'12) received her MSc. degree in Electrical Engineering from Brno University of Technology, Czech Republic, in 1983. She joined the Department of Radio Electronics in 1985, and is currently working as a Research Assistant at the Department of Radio Electronics, Brno University of Technology, Czech Republic. Her research and educational interests include modeling of large-scale systems, signal theory, analog signal processing, memristors and memristive systems, optoelectronics, and digital electronics.

ZDENĚK BIOLEK received the PhD. degree in Electronics and Informatics from Brno University of Technology, Czech Republic, in 2001.

He is currently with the Department of Microelectronics, Brno University of Technology, and with the Department of EE, University of Defence Brno, Czech Republic. Until the year 1993 he worked as independent researcher in semiconductor company TESLA

Rožnov. He is the author of unique electronic instruments associated with IC production and testing. He is also the author of several papers from the area of the utilization of variational principles in electrical engineering, and also from the field of memristors and mem-systems. Dr. Z. Biolek is the co-author of two books about memristive systems and modeling and simulation of special electronic circuits including switched-capacitor filters, switched DC-DC converters, and memristors.

SHAHAR KVATINSKY (SM'18) is an Associate Professor at the Andrew and Erna Viterbi Faculty of Electrical Engineering, Technion – Israel Institute of Technology. Shahar received the B.Sc. degree in Computer Engineering and Applied Physics and an MBA degree in 2009 and 2010, respectively, both from the Hebrew University of Jerusalem, and the Ph.D. degree in Electrical Engineering from the Technion – Israel Institute of Technology in 2014. From 2006 to 2009, he worked as a circuit designer at Intel. From 2014 and 2015, he was a post-doctoral research fellow at Stanford University. Kvatinsky is an editor of Microelectronics Journal and has been the recipient of numerous awards: 2020 MDPI Electronics Young Investigator Award, 2019 Wolf Foundation's Krill Prize for Excellence in Scientific Research, 2015 IEEE Guillemin-Cauer Best Paper Award, 2015 Best Paper of Computer Architecture Letters, Viterbi Fellowship, Jacobs Fellowship, ERC starting grant, the 2017 Pazy Memorial Award, the 2014 and 2017 Hershel Rich Technion Innovation Awards, 2013 Sanford Kaplan Prize for Creative Management in High Tech, 2010 Benin prize, and seven Technion excellence teaching awards. His current research is focused on circuits and architectures with emerging memory technologies and design of energy efficient architectures.