Reliable Modeling of Ideal Generic Memristors via State-Space Transformation

Dalibor BIOLEK [1,2], Zdenek BIOLEK [2], Viera BIOLKOVA[3], Zdenek KOLKA [3]

[1] Dept. of Electrical Engineering, University of Defense, Kounicova 65, 662 10 Brno, Czech Republic
[2] Dept. of Microelectronics, Brno University of Technology, Technicka 10, 616 00 Brno, Czech Republic
[3] Dept. of Radio Electronics, Brno University of Technology, Technicka 12, 616 00 Brno, Czech Republic

dalibor.biolek@unob.cz, zdenek.biolek@gmail.com, {biolkova, kolka}@feec.vutbr.cz

Abstract. *The paper refers to problems of modeling and computer simulation of generic memristors caused by the so-called window functions, namely the stick effect, non-convergence, and finding fundamentally incorrect solutions. A profoundly different modeling approach is proposed, which is mathematically equivalent to window-based modeling. However, due to its numerical stability, it definitely smoothes the above problems away.*

Keywords

Constitutive relation, memductance, memristance, memristor

1. Introduction

The announcement [1] of the fabrication of the so-called HP memristor, which imitates the behavior of the so-called fourth fundamental circuit element, introduced into the circuit theory by L. Chua in 1971 [2], initiated the development of a lot of its various models for computer simulation [1], [3–30]. The very first one is a simple model from [1] describing the TiO_2 memristor as a resistor whose resistance is dependent on the position of the boundary between the doped and undoped TiO_2 layers. This position, dependent on the history of the current flowing through the device, is modeled there via a differential equation where the derivative of the position with respect to time is directly proportional to the electric current. In order to model the nonlinear dopant drift, the current is then multiplied by the so-called window function which takes zero values for the limiting values of the boundary positions, this way modeling the boundary slowdown near its terminal states. The frequently used Joglekar window [12] can be shaped by an integer parameter *p*. A serious drawback of the Joglekar and several other windows is called the stick effect or the terminal-state problem: if the system reaches one of its limiting states, then the derivative of the state is zero, with no possibility of leaving this state via external excitation. Some authors deal with this problem via considering non-zero values of the window at limiting states [13]. However,

this causes another problem: if the system reaches a state which represents some physical limit, it cannot continue in the motion, and thus the state derivative with respect to time must be zero. This status can be changed only if the direction of the driving quantity is changed too. This knowledge is reflected by the Biolek window [14], which depends not only on the state but also on the device current, or, more exactly, on its direction. Let us mention for the sake of completeness that also other window functions for modeling the nonlinear dopant drift can be found in the literature [15], [16], for example the Strukov [1], Prodromakis [17], TEAM or Kvatinsky [18] windows, the piecewise-linear window [19], the Tukey window [20], the trigonometric windows [21], the BCM [22] and generalized BCM model [23], etc. Some of the above techniques were developed in an effort to refine the models in order to make them conform to experimental data. The survey of selected models is given in [24], [25] under the names "Air Force Research Lab Model" [26] and "HP labs MIM Model" [27]. Today frequently utilized "Hyperbolic Sine Model" [28] was generalized in [29], and another modification of this model was published in [30]. An excellent analysis of the hyperbolic models is given in [31].

A common feature of the above hardware-correlated models as well as of models of other memristive systems such as PCM (Phase Change Memory) [32], (ITMT) Insulator-to-Metal Transition memristive system [33], TaO_x devices [34], etc., is that they gradually move away from the original definition [2] of the memristor as the ideal circuit element, which is defined by the charge (integral of current) – flux (integral of voltage) constitutive relation, whose state variable is either the charge or the flux. All the models are based on state equations, with the state variables being generally derived from the geometry of the physical implementation of the system (the position of the boundary between the layers of the TiO_2 memristor [1], the tunneling gap width in the Pickett model [35], the metallic phase fraction of ITMT systems [33], the angular spin in the spintronic memristor [36] or from the variables which characterize accumulated quantities such as the temperature – the state variable of the thermistor [37], bulb [38], [39], discharge lamp [40], or PCM system [32]. The window approach, a potential source of numerical problems in

CIRCUITS

the course of subsequent simulations [41], is frequently used therein for modeling various boundary phenomena.

Some of the above models of the memristive system fall into the class of the so-called generic memristors or ideal generic memristors [42], whose computer simulation frequently labors with substantial numerical errors. A lot of papers describe the already mentioned stick effect. It is stated in [17] that the Prodromakis window "is essentially free of the terminal-state problem". A contraexample will be given in Sec. 2. Convergence problems have also been reported, particularly when simulating hard-switching regimes or using the rectangular window via increasing the p parameter for the Joglekar window.

The stick effect and the convergence problem are in fact only the tip of the iceberg. The papers [41], [43] provide SPICE simulations of a simple model of the HP memristor whose results merely differ from its theoretical behavior. The analysis in [44] points out that memristor models employing window functions of the Joglekar type act as amplifiers of the numerical errors generated by the finite-precision number representation in the SPICE environment. One can keep a check on these errors only to some degree: Such conditions can always be induced, particularly via exciting the simulated system into the hard switching regime, when the errors exceed the acceptable level, and this cannot be put right by tightening the error criteria or by other arrangements such as selecting the method of numerical integration. This regime of absolutely incorrect behavior of the model, which can be regarded as balancing between the correct behavior and the stick effect, is rather uncomfortable because it can be detected only via a rigorous analysis, for example, by comparing the simulation results with the analytic solution, provided it is exceptionally available or, for example, by establishing that the model behavior violates the general fingerprints of the memristor.

The above facts together with the other findings that window functions of the Joglekar, Prodromakis, and mostly rectangular type are in principle potential sources of the stick effect or nonconvergence, lead unavoidably to the idea of replacing the relevant models of the ideal generic memristors by other models that might not use the window functions at all, and that would not suffer from the above shortcomings.

The paper proposes a novel method of modeling ideal generic memristors which does away with the drawbacks of the window methods and provides correct results also in extremely hard switching regimes, eliminating the stick effect. The ideological starting point of this method was published by the two of us (DB and ZB) in [43]: Every ideal generic memristor can be characterized by an algebraic transitional function between the so-called native (charge or flux) and the physical (e.g. boundary position of the TiO$_2$ memristor) state variable whose form is unambiguously related to the corresponding window function. From a more complex point of view, this transitional func-

tion was also introduced in the excellent work [45]. We have shown in [43] that, for a simple case of the transitional function of the Joglekar window with $p = 1$, an analytical formula exists in the form of sigmoidal (logistic) function. Then the model will exhibit a correct behavior if this function is used instead of the classical approach with the Joglekar window.

The paper follows this idea with the aim of extending the technique of "sigmoidal" modeling to all possible types of ideal generic memristors, regardless of the type of their window functions or their transitional characteristics, and regardless of whether the analytic form of these characteristics exists or not. The final objective is to put this method into common practice of computer modeling of ideal generic memristors wherever the unreliable window-based models are currently utilized. These objectives are achieved in four steps: 1) The terms native and physical state variables are defined, and the boundaries between ideal memristor [2], ideal generic memristor [42], and memristive system [46] are recalled. 2) The risk of the window techniques being a hidden source of fatal numerical errors, devaluing the simulation results, is illustrated. 3) A method of modeling ideal generic memristors, removing problem No 2), is suggested. 4) The respective source codes for the users of SPICE-family programs are released.

Although the paper focuses on the modeling of ideal generic memristors, the analysis of the reasons of fatal malfunction of some currently used models can become a starting point for similar analyses of the behavior of models of general memristive systems in the environment of numerical simulation programs [44].

2. Fatal Malfunction of Classical Window-based Model of TiO$_2$ Memristor: A Demonstration

The example of a fatal malfunction of the model of HP memristor described in [41] was concerned with the simple Joglekar window for $p = 1$. We demonstrate below that the breakdown can also turn up for other window functions that are considered to be resistant to numerical errors. Prodromakis window is selected in this section for such a demonstration.

Figure 1 shows the well-known simple model of the HP memristor with the modeling of the nonlinear dopant drift via the window function $f(x)$, where $0 \leq x \leq 1$ is the normalized width of the doped TiO$_2$ layer [43].

Figure 2 illustrates the results of the PSpice analysis, using the Prodromakis window with $p = 10$ [17], for an amplitude 0.25 mA of sinusoidal 1 Hz excitation, and for the model parameters specified in Appendix 1. As is obvious from the figure, the model exhibits the well-known fingerprints of the ideal generic memristor, namely: 1. The
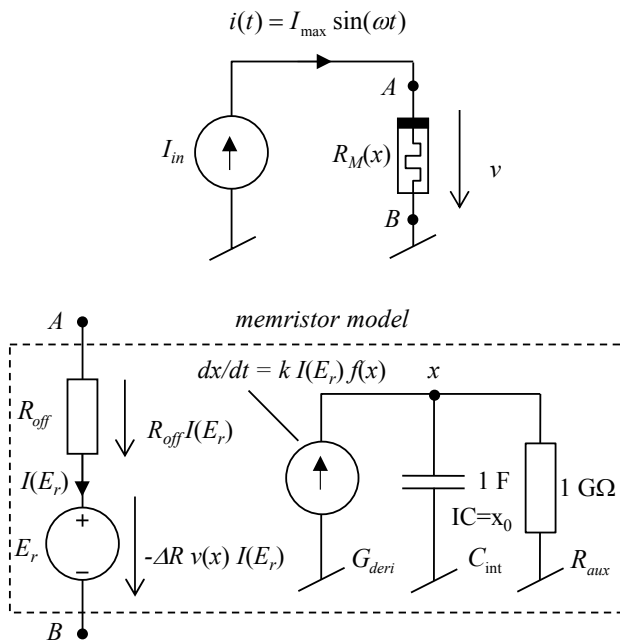
Fig. 1. Demonstration of the behavior of the memristor driven by sinusoidal current. The memristive port A-B is modeled by the equation $R_M(x) = R_{off} - (R_{off} - R_{on})x$, where x is a state variable whose derivative with respect to time is proportional to the port current and the nonlinear function of state $f(x)$. Concrete model parameters are given in SPICE code in Appendix 1.

hysteresis loop is pinched at the v-i origin and it is odd-symmetric. 2. The memristor passes immediately to the periodical steady state [47].

Figure 3 presents the simulation results for an increased amplitude of 1 mA. The violation of the fingerprint of immediate transition to periodical steady state is obvious at first glance. It follows from the waveform of the state variable the state variable does not return back to the initial state $x_0 = 0.5$ after the first repeating period of the driving signal but is instead attracted to the limiting state $x = 0$. As a consequence of this incorrect behavior of the model during the first repeating period, the waveforms at all consecutive time instants are evaluated erroneously. The periodical steady state, computed from such a model, then may have nothing to do with the correct result. In fact, the correct behavior of the model from Fig. 1 is quite different, as will be shown later in Sec. 6 (see Fig. 9).

The analysis of such cases given in [41] reveals the cause of the malfunction of the numerical model: at a certain time instant, due to numerical errors which are generated via a co-action of the rounding errors and the method of modeling the integrator according to Fig. 1, the state variable of the memristor stops moving despite the fact that the native state variables [43], the charge and the flux, vary in time. The state variable x stops near a level that fundamentally limits its motion ($x = 1$ in our case). In this way, a shift arises between the actual and the simulated operating point of the memristor. If the signal driving the memristor reaches the values that unstick the state variable x from the fixed level, then the state continues moving from

the incorrect operating point, which makes the simulated waveform of the state variable and all the signals derived from it inaccurate. This error can be affected to a certain extent via tweaking the parameters of the model and the analysis but it cannot be fully suppressed.

The prime cause of the given incorrect behavior of the model can be perceived as the split between the fundamental non-limitation of the motion of native state variables (charge and flux) of ideal memristor, and the fundamental limitation of the corresponding physical state variables [43] (x in the above case) of ideal generic memristor, which is given by actually existing limits (for example $0 \leq x \leq 1$). By virtue of the numerical errors, the causal relation between the native and the physical state variable is cut off in the proximity of the boundary states of the variable x, which harms the numerical solution. Since the window functions are directly designed to model the bounded physical state variables, their use is the main cause of the incorrect behavior of the given models, particularly under hard-switching regimes. The key to the construction of reliably working models of ideal generic memristors is giving up the window technique and eliminating the physical, bounded-value state variables from the process of the numerical integration of the differential equations of memristors.

Let us note, for the sake of completeness, that if the experiment from Fig. 3 is repeated using the rectangular window (the corresponding source code is also given in Appendix 1), the well-known stick effect appears. In other words, the model fails as well.

3. Native and Physical State Variables of Memristor

It is obvious from the classical definition of the memristor [42] controlled by the charge q

$$v = R_{mem}(q) \cdot i, \qquad (1)$$

$$\frac{dq}{dt} = i, \qquad (2)$$

or by the flux φ

$$i = G_{mem}(\varphi) \cdot v, \qquad (3)$$

$$\frac{d\varphi}{dt} = v \qquad (4)$$

that the memristor represents state-dependent Ohm's law between voltage v and current i whereas the state variable is either charge (time-domain integral of current) or flux (time-domain integral of terminal voltage). Let us recall two facts in this respect: 1) The memristance R_{mem} or memductance G_{mem} depends only on the state variables and not on instantaneous values of voltage and current. 2) The flux φ (flux linkage) has, in general, nothing to do with the magnetic flux [48].
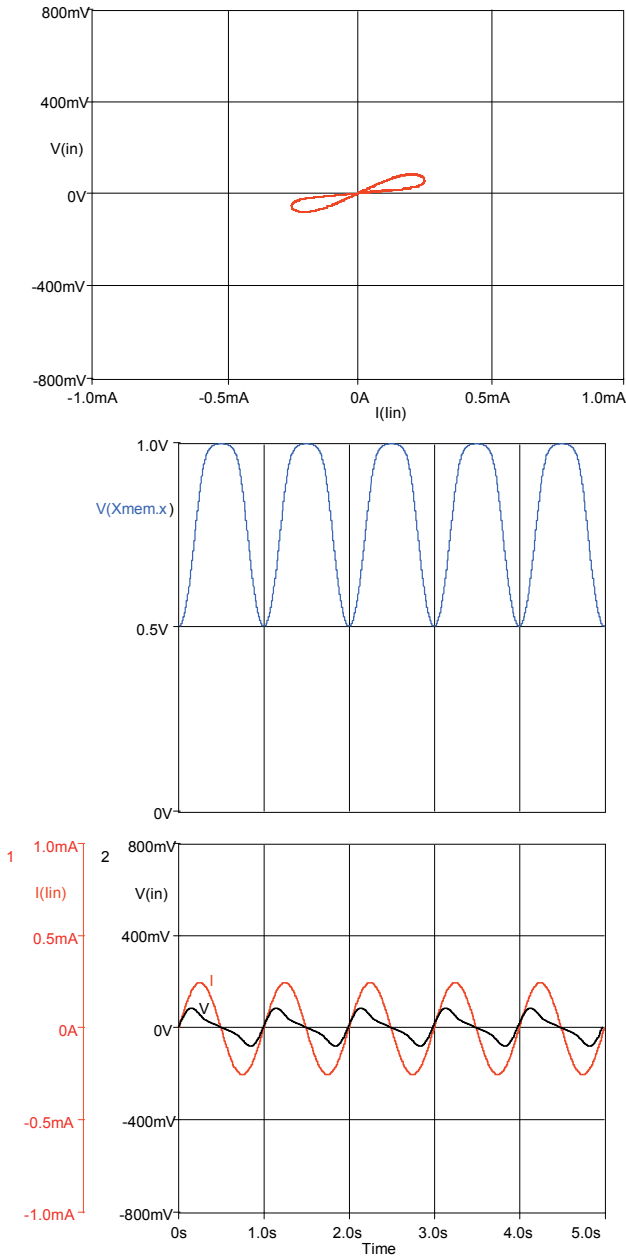
Fig. 2. Simulation results of the model from Fig. 1 for $I_{max} = 0.25$ mA: *v-i* odd-symmetric pinched hysteresis loop, periodical waveform of state variable, the corresponding voltage and current waveforms (immediate transition to periodical steady state). The model exhibits the well-known fingerprints of ideal memristor.
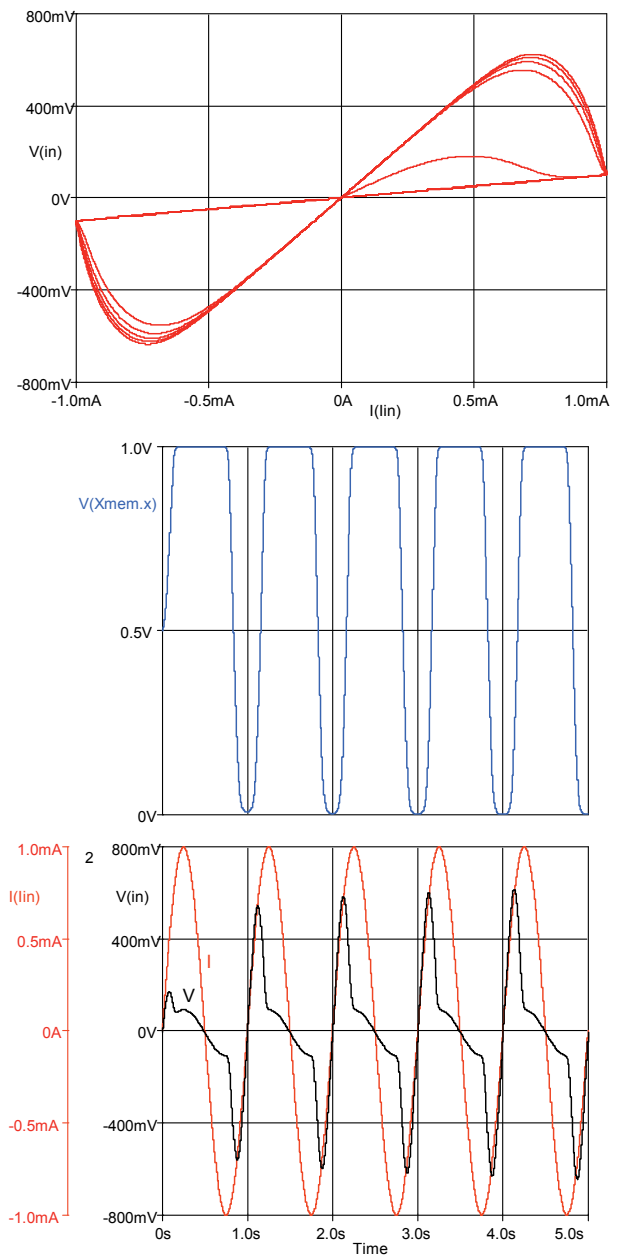
Fig. 3. Simulation results of the model from Fig. 1 for $I_{max} = 1$ mA: gradually forming *v-i* pinched hysteresis loops, waveform of state variable, voltage and current exhibit initial transients. The model violates the well-known fingerprints of ideal memristor.

A number of theoretical pieces of knowledge follow from definitions (1)–(4), for example that the memristor is unambiguously determined by the dependence of its parameter on the state variable (PSM = Parameter vs. State map $R_{mem}(q)$ or $G_{mem}(\varphi)$) or by the equivalent constitutive relation (CR) $\varphi = \varphi(q)$ or $q = q(\varphi)$, whereas its memristance or memductance is given by the flux derivative with respect to charge or the charge derivative with respect to flux. Also the well-known memristor fingerprints [42], [43], [47], [49] result from these definitions.

The memristors defined above will be thereinafter denoted as ideal memristors. From the technical point of view, the term "ideal" is redundant because memristor defined in such a way is in itself an ideal hypothetical element. On the other hand, it may contribute to a more distinct interpretation, because today the term memristor also includes more general memristive systems that need not exhibit some attributes of ideal memristors. Such memristive systems are denoted in [42] as extended memristors. The current- or voltage-controlled memristive system is defined by equations of the type [42]

$$v = R_{\text{mem}}\left(\mathbf{x},i\right)\cdot i \,,\ R_{\text{mem}}\left(\mathbf{x},0\right)\neq\infty \,, \tag{5}$$

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x},i) \,, \tag{6}$$

or

$$i = G_{\text{mem}}\left(\mathbf{x},v\right)\cdot v \,,\ G_{\text{mem}}\left(\mathbf{x},0\right)\neq\infty \,, \tag{7}$$

$$\frac{d\mathbf{x}}{dt} = g(\mathbf{x},v) \tag{8}$$

where $\mathbf{x}$ is the vector of state variables while f and g are, in general, nonlinear functions. The memristive system cannot be generally characterized by the above constitutive relations or PSMs. Also, some fingerprints of ideal memristors do not apply there, for example the fingerprints on the symmetric pinched hysteresis loops [50] or on the immediate transition to periodical steady state [47].

The term "generic memristor", introduced by L. Chua in [42], supplements the pair "ideal memristor" and "memristive system" = "extended memristor". The port and state equations of the general memristive system controlled by current (5), (6) or voltage (7), (8) are specified for generic memristors as follows [42]:

$$v = R_{\text{mem}}\left(\mathbf{x}\right)\cdot i \,, \tag{9}$$

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x},i) \,, \tag{10}$$

or

$$i = G_{\text{mem}}\left(\mathbf{x}\right)\cdot v \,, \tag{11}$$

$$\frac{d\mathbf{x}}{dt} = g(\mathbf{x},v) \,. \tag{12}$$

Ideal generic memristors form an important subset of generic memristors. Their memristance or memductance depends on a scalar state variable x, whose derivative is directly proportional to current or voltage:

$$v = R_{\text{mem}}\left(x\right)\cdot i \,, \tag{13}$$

$$\frac{dx}{dt} = f(x)i \,, \tag{14}$$

or

$$i = G_{\text{mem}}\left(x\right)\cdot v \,, \tag{15}$$

$$\frac{dx}{dt} = g(x)v \,. \tag{16}$$

In contrast to (2) or (4), this model makes use of a different state variable than the charge or flux (for example the boundary position in the TiO_2 memristor), exhibiting, none the less, all the external manifestations and fingerprints of the ideal memristor. From the mathematical point of view, the ideal generic memristor is a memristor whose port and state equation can be converted via state variable transformation into a form that would correspond to (1), (2) or (3), (4). Among other things, it means that the ideal

generic memristor and also the ideal memristor can be defined by the CR and PSM characteristics.

Distinguishing between the ideal generic memristor and the more general memristive system can be essential for the construction of the model of currently existing object, exhibiting memristive behavior, and for its effective implementation in the simulation program. In the first case, the model must exhibit all the fingerprints of ideal memristor, and every deviation from this paradigm behavior is an easily detectable failure of the model in the software environment. The connections between the ideal memristor and ideal generic memristor are summarized below.

To control its parameter (the memristance or memductance) the ideal memristor uses the state variable of the type of charge or flux, denoted in [43] as *native* state variable. Nevertheless, the axiomatic definitions (1)–(4) do not solve the question of the physical implementation of memristor memory, i.e. the way of accumulating the native state variable according to state equations (2) and (4). These equations describe an ideal integration of current into charge or voltage into flux, without modeling the currently existing limits. From this point of view, the ideal memristor is thus an element with unbounded memory.

The existing systems with memristive properties use physical-chemical processes for the accumulation of the memory. The measure of the memory effect can be quantified via a concrete state variable which is naturally bounded in value. Let us call it the physical state variable. For the simple model [1] of the HP memristor, this variable can be the width of the doped layer, which must not exceed the total width of the TiO_2 layer or, for example, it can be the normalized width with the upper limit 1. Memristance can also be used as the physical state variable if a one-to-one relation exists between the memristance and the width of the layer. The memristance bounded in values, as a state variable, is used, for example, to model the threshold-type bipolar memristive system [44], [51], [52]. The crystalline fraction, bounded within the limits 0 and 1, is one of the physical state variables for modeling the phase change memory (PCM) cells [32]. In the memristive system of the type of Insulator-to-Metal Transition [33], the measure of the memory effect is modeled by the state variable called metallic phase fraction, which is also naturally bounded in values. A similar bounded variable – volume fraction – is used in [34] for modeling the tantalum-dioxide memristors. The well-known memristive Hodgkin-Huxley model of the axon [53], [54] contains a couple of memristive systems, denoted Potassium memristor and Sodium Ion-Channel memristor. The state variable of the first one is the potassium gate-activation variable within a range from 0 to 1. The sodium channel is modeled by a two-dimensional memristive system, whose two dimensionless state variables are also bounded within the limits from 0 to 1.

For the model of a real memristive system whose internal state is modeled via a physical, not native state variable to be denoted as the model of ideal generic memristor,

it must exhibit, the same as ideal memristor, an "infinite memory depth". This can be theoretically achieved if a one-to-one relationship exists between the native (unbounded in values) and physical (bounded in values) state variables. To put it simply, the value of the physical state variable follows unambiguously from the value of the natural state variable and vice versa. In reality, this theoretically unambiguous relationship will be violated by the effect of the "finite resolution" of the physical state variable in the proximity of its limiting values (noise in the case of real systems, numerical errors in computer simulations).

In the following, an alternative way of modeling the ideal generic memristor will also be analyzed: The generator of the native state variable (the integrator of the current generating the charge or the integrator of the voltage generating the flux) can be a part of the model. This variable is then transformed via a nonlinear noninertial block into the corresponding physical state variable. In such a case, the model behaves as an ideal memristor with infinite memory depth (= ideal integrator as a part of the model) regardless of the (non)existence of a one-to-one correspondence between the native and the physical state variables.

4. Ideal Generic Memristor

Consider the following special forms of (5)–(8) of one-dimensional current- or voltage-controlled memristive systems:

$$v = R_{\mathrm{mem}}(x) \cdot i, \tag{17}$$

$$\frac{dx}{dt} = f(x)i, \tag{18}$$

or

$$i = G_{\mathrm{mem}}(x) \cdot v, \tag{19}$$

$$\frac{dx}{dt} = g(x)v. \tag{20}$$

The port equations (17) and (19) formally correspond to (1) and (3) of the ideal memristor (the parameter of the element only depends on the state variable, not on the voltage or current), while the state equations seemingly do not.

The state equation (18) can be modified to the form

$$\frac{dx}{f(x)} = idt = dq, \quad f(x) \neq 0. \tag{21}$$

The condition $f(x) \neq 0$ follows from the mathematical formalism though, but it also has its physical reasons. According to (18), the derivative of the state variable would be zero for $f(x) = 0$, which would mean that the system would have no possibility to extricate itself from this state. It is the well-known stick effect which accompanies the improper modeling of memristive systems.

It follows from (21) that

$$\frac{dx}{dq} = f(x), \frac{dq}{dx} = \frac{1}{f(x)}, \quad f(x) \neq 0. \tag{22}$$

The right-side nonlinear function $f(x)$ in (18) thus determines the slopes of the transitional functions between the physical and the native state variables for a given x.

After integrating both sides of (21) we get

$$\int \frac{dx}{f(x)} = q, \quad f(x) \neq 0. \tag{23}$$

Consider that x is a physical state variable limited within the interval $x_{\min} - x_{\max}$. Denote as $F(x)$ the primitive function of the integrand $1/f(x)$ on the left side of (23) on the interval $x_{\min} \leq x \leq x_{\max}$. Then

$$\int \frac{dx}{f(x)} = F(x) + C = q, \quad f(x) \neq 0, \tag{24}$$

where C is a constant of integration.

Equation (24) can be read with the help of Fig. 4 (a).

The curve of the graph of the primitive function can be moved vertically by changing the constant of integration C such that its point of intersection with the axis $q = 0$ determines the initial value of the physical state variable x_0. Then, based on the curve, and knowing the deflection of x from x_0, one can read the charge that passed through the memristor and thus caused this deflection.

Knowing the function $f(x)$ in the state equation (18), one can derive the primitive function $F(x)$ and perform the transformation of the physical state variable x into the native state variable q. In other words, this transformation converts the state equation (18) into the state equation (2) of ideal memristor, and the memristive system (17), (18) can be declared an ideal generic memristor. However, the necessary condition is the one-to-one transformation: different states of the native state variables $q_1 \neq q_2$ must be assigned to two different states $x_1 \neq x_2$ from the interval $[x_{\min}, x_{\max}]$. This condition also implies the existence of the one-to-one function $FI(q)$, which is the inverse of the primitive function $F(x)$. Note that $FI(q)$ reflects the causal dependence of the physical state variable on the native state variable, see Fig. 4 (b).

The dot-and-dash curves in Fig. 4 correspond to the case when the physical state variable achieves its limit value for a finite value of the native state variable. Then the function $FI(q)$ is not a one-to-one function, and the change of the native state variable does not cause any change of the physical state variable. Since in actually existing systems, in contrast to the state variable x, the native state variable is not stored in any manner, such systems lose the attribute of unlimited memory, and their models therefore do not behave as ideal generic memristors. As shown in [41], this is the reason for the fatal malfunction of some currently used memristor models

$F(x)+C=q$

x_{min} x_{max} (a)

$x=FI(q)$

x_{max}

C

x_0

x_{min}

0

q (b)
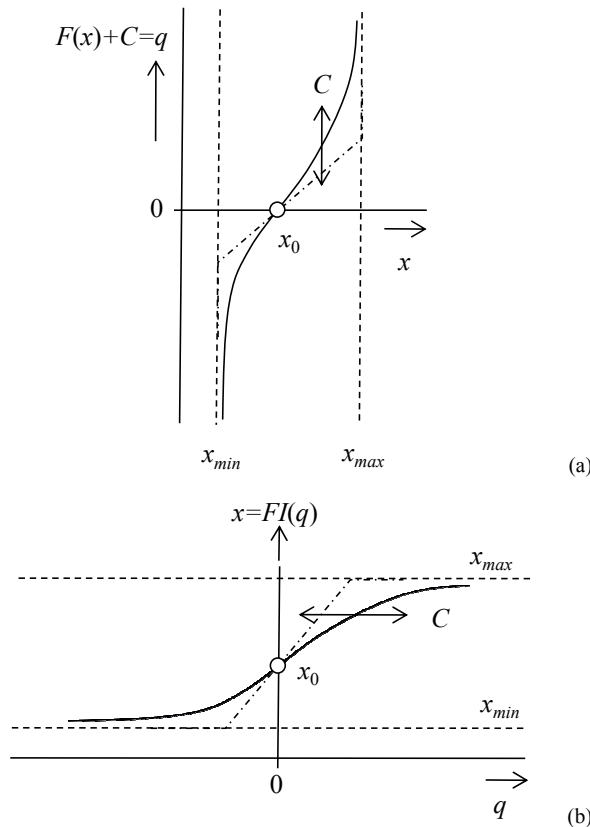
Fig. 4. The function for transforming the physical (x) to the native (q) state variable of ideal generic memristor (a), and its inverse (b) [43]. The dot-and-dash curves are not a one-to-one map and thus they do not correspond to ideal generic memristor[1].

implemented in the simulation programs, where the physical state variable is subject to a finite precision of the number representation in the program environment.

An analogous analysis can also be made for the voltage-controlled ideal generic memristor (19), (20). The main results can be summarized as follows:

Proposition 1.

Equations (17), (18), or (19), (20) are equations of current- or voltage- controlled ideal generic memristor if the primitive function to the function $1/f(x)$ or $1/g(x)$ is one-to-one on the interval $[x_{min}, x_{max}]$. Then this function and its inverse can be used for a mutual transformation between physical and native state variables of the memristor.

One of the fingerprints of ideal generic memristor (17), (18), or (19), (20), can be simply formulated as follows:

Proposition 2.

Every change of the native state variable causes a change of the physical state variable.

---

[1] With the exception specified at the end of Sec. 3.

The above stick effect is a typical consequence of violating this rule.

Recall that the above Propositions hold for the models (17), (18), or (19), (20) of ideal generic memristor, i.e. on the assumption that the computation of the native state variable from the current or voltage is not part of the model (see also the note at the end of Sec. 3).

5. Examples of Systems That Are (Not) Ideal Generic Memristors

(a) HP memristor with linear dopant drift

The well-known simple physical model of the TiO_2 memristor from [1], [14] is described as state-dependent Ohm's law (17), with the memristance R_{mem}

$$R_{mem}(x) = R_{on}x + R_{off}(1-x) \qquad (25)$$

where $x \in [0,1]$ is the normalized width of the doped TiO_2 layer, and R_{off} and R_{on} are the limiting memristance values.

The speed of the boundary between the doped and undoped layers is described by a simplified state equation [14]

$$\frac{dx}{dt} = ki \qquad (26)$$

where i is the element current, and k is a technological constant.

A comparison of (26) and (18) confirms that it is a model of ideal generic memristor. There is a linear relationship between the physical and the native state variables because $f(x) = k$ and, according to (22), $dx = k\,dq$. Equation (24) can be written as

$$q = x/k + C = (x - x_0)/k. \qquad (27)$$

According to (27), the physical state variable x is not bounded in values, which is in conflict with its fundamental limitation within the range [0,1]. It is a price to be paid for the simplification of the model of dynamics (26). The model holds only for an admissible range of the state variable.

(b) HP memristor with nonlinear dopant drift, Joglekar [12] or Prodromakis [17] window.

The state equation (26) is completed by the so-called window function $f_w(x)$, which models slow-down of the boundary motion in the proximity of the limit positions $x = 0$ and $x = 1$:

$$\frac{dx}{dt} = kif_w(x). \qquad (28)$$

It holds for the Joglekar window

$$f_w(x) = 1 - (2x - 1)^{2p}. \qquad (29)$$

Increasing the integer p from 0, the shape of the window can be modified up to the rectangular type (for $p \to \infty$). The window curve is symmetric with respect to the vertical line $x = 0.5$. For $x = 0.5$, the function (29) reaches its maximum value (one) for all p.

The Prodromakis window is defined as [17]

$$f_w(x) = j\left(1 - \left[(x - 0.5)^2 + 0.75\right]^p\right) = j\left(1 - \left[x^2 - x + 1\right]^p\right) \cdot \tag{30}$$

The parameter j can adjust an arbitrary maximum value f_{wmax} of (30). This window also provides symmetry with respect to the vertical line $x = 0.5$, with the maximum at this point. The condition $f_w(0.5) = f_{wmax}$ yields

$$j = \frac{f_{wmax}}{1 - 0.75^p}. \tag{31}$$

Comparing (28) and (18), we get

$$f(x) = k f_w(x). \tag{32}$$

For clarity, let us denote

$$\int \frac{dx}{f_w(x)} = F_w(x) + C_w = kq = \bar{q}, \ f_w(x) \neq 0, \tag{33}$$

where

$$F_w(x) = kF(x) \tag{34}$$

is a function associated with the concrete window function $f_w(x)$, and $C_w = kC$ is the constant of integration (see (24)). The function $F_w(x)$ is thus used to transform the normalized physical state variable x into the normalized native state variable

$$\bar{q} = kq. \tag{35}$$

The model (28) will be the ideal generic memristor if the function $f_w(x)$ complies with Proposition 1, namely, if the function

$$F_w(x) + C_w = \int \frac{dx}{f_w(x)}$$

is a one-to-one function for $0 \leq x \leq 1$.

It holds for the Joglekar window [22]

$$\bar{q} = F_w(x) + C_w = \int \frac{dx}{1 - (2x - 1)^{2p}} =$$
$$= (2x - 1)_2 F_1\left[\frac{1}{2p}, 1, 1 + \frac{1}{2p}, (2x - 1)^{2p}\right] = \tag{36}$$
$$= \frac{1}{2} \sum_{i=0}^{\infty} \frac{(2x - 1)^{2pi+1}}{2pi + 1}.$$

For given arguments and for $x \in (0,1)$, the Gauss hypergeometric function $_2F_1[a,b,c,d]$ can be expressed as an infinite series from (36). For $p = 1$, the result can be simplified to the form

$$\bar{q} = F_w(x) + C_w = \frac{1}{4} \ln\left(\frac{x}{1 - x}\right), x \in (0,1), \tag{37}$$

and for $p = 2$

$$\bar{q} = F_w(x) + C_w = \frac{1}{4}\left[\frac{1}{2}\ln\left(\frac{x}{1 - x}\right) + \tan^{-1}(2x - 1)\right], x \in (0,1). \tag{38}$$

For higher p, the notation of $F_w(x)$ in terms of the elemental functions becomes rather unclear. Then the numerical computation of hypergeometric function can be preferable. Examples of the curves for $C_w = 0$ in Fig. 5 correspond to one-to-one functions. The model employing the Joglekar window is therefore an ideal generic memristor.

The inverse functions $FI_w(\ )$ for a direct transformation of normalized native state variable into physical state variable can also be found. For $p = 1$ (see (37)), $FI_w(\ )$ can be expressed as a simple analytical formula:

$$x = FI_w(\bar{q}) = \frac{1}{1 + e^{-4\bar{q}}}, C_w = 0. \tag{39}$$

The corresponding graph is the well-known logistic curve [55]. The graphs of the inverse functions $FI_w(\ )$ originate from the graphs in Fig. 5 via swapping the axes, thus they are not given here.

For $p > 1$, the inverse function $FI_w(\ )$ cannot be explicitly derived via elemental functions, and numerical computation should be applied.

Note that if the physical state variable x was set exactly to one of its limit values $x = 0$ or $x = 1$, the model would lose its ability to behave as ideal generic memristor. An exact solution of the state equation (28) for the initial state $x_0 \in (0,1)$ cannot lead to such conditions though. However, this does not apply to the numerical solution which proceeds in the software environment with finite-precision arithmetic.

For the Prodromakis window, the analytic expression of the function $F_w(x)$ for a general parameter p is tedious. For $p = 1$ we get
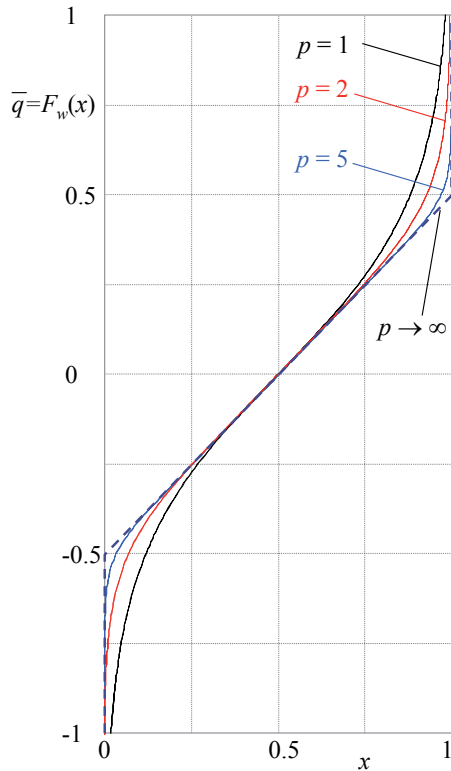
$$\bar{q} = F_w(x) + C_w = \frac{1}{j} \ln\left(\frac{x}{1 - x}\right), x \in (0,1), \tag{40}$$

and for $p = 2$

$$\bar{q} = F_w(x) + C_w = \frac{1}{j}\left[\frac{1}{2}\ln\left(\frac{x}{1 - x}\right) + \frac{\tan^{-1}\left(\frac{2x - 1}{\sqrt{7}}\right)}{\sqrt{7}}\right], x \in (0,1). \tag{41}$$

The numerical computation is useful for a general parameter p. The corresponding curves for several parameters p are shown in Fig. 6. The parameter j from (31) was selected for each value of p such that $f_{wmax} = 1$.

Fig. 5. The function $F_w(x)$, $C_w = 0$, for transforming between normalized physical and native state variables of ideal generic memristor with the Joglekar window for various parameters p. The limit case $p \to \infty$ corresponds to rectangular window, where the condition of one-to-one function is violated at points $x = 0$ and $x = 1$.

Fig. 6. The function $F_w(x)$, $C_w = 0$, for transforming between normalized physical and native state variables of ideal generic memristor with the Prodromakis window for various parameters p and for $f_{wmax} = 1$.

Comparing (40) and (37) yields that the transforming functions $F_w(x)$ for $p = 1$ are the same for the Joglekar and the Prodromakis windows if $j = 4$, thus $f_{wmax} = 1$. It is in conformity with the knowledge that both windows are identical under these conditions. The differences for $p > 1$ are already obvious. It follows from Fig. 6 that, compared to the Joglekar window, the parameter p for the Prodromakis window must be higher in order to achieve a similar behavior of $F_w(x)$. It is related to the fact that the slope of the Prodromakis window at boundary points $x = 0$ and $x = 1$ for a given p and for $f_{wmax} = 1$ is only one quarter of the slope of the Joglekar window.
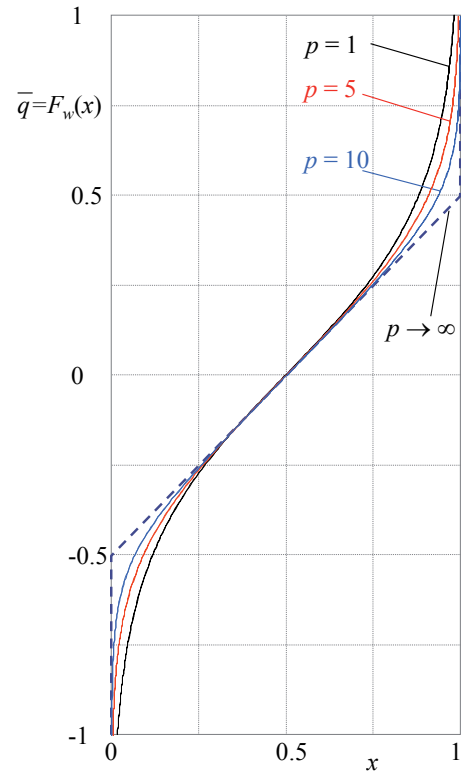
The given model of the memristive system employing the Prodromakis window with finite value of p is therefore a model of ideal generic memristor.

(c) HP memristor with nonlinear dopant drift and the Biolek window [14] or the BCM model [22].

The corresponding window functions depend not only on the state variable x but also on the current or voltage, depending on whether the memristive system is controlled by the current or voltage. The state equation for current-controlled system is then in the general form

$$\frac{dx}{dt} = kif_w(x, i).$$ (42)

Such memristive systems cannot be therefore ideal generic memristors.

(d) Some models of memristive switching devices

It follows from a comparison of the port and the state equations of several other known models of memristive systems with (17)–(20) and Propositions 1 and 2 that, for example, the Lehtonen and Laiho model [28], Pickett model [27], TEAM model [18], Ielmini [56], [57] and Miranda [58] filament model, and other hardware-correlated models [24] are not models of ideal generic memristors. Some of the above models can behave as ideal generic memristors under special conditions. For example, the memductance of the Lehtonen model [28] practically does not depend on the terminal voltage if this voltage is maintained near zero. As follows from the state equation of the model

$$\frac{dx}{dt} = af_w(x)v^q$$ (43)

where a is a fitting constant, f_w is the window function, q is a positive odd integer, the additional condition $q = 1$ must be also fulfilled in order to consider this model an ideal generic memristor.

(e) Spin-valve magnetic domain-wall memristor

An example of a long spin-valve structure with domain wall motion in the free layer is published in [36]. The resistance depends on the domain-wall position x:

$$R = R_L \frac{x}{D} + R_H \frac{D - x}{D}$$ (44)

where D is the free-layer length, R_L and R_H are the minimum and the maximum limits of the resistance. The speed of the domain wall is directly proportional to the current:

$$\frac{dx}{dt} = \Gamma i \qquad (45)$$

where Γ is a proportionality constant.

It is obvious that the model is analogous to the model of the HP memristor with linear dopant drift, and that it is a model of ideal generic memristor. This model is specified in [59] into the form of a second-order memristive system, which is no longer an ideal generic memristor.

(f) Hydraulic memristor

The hydraulic memristor according to Fig. 7 is described in [43]. A part of the liquid flowing through a conic profile tube rotates a propeller. The propeller moves a plug in the direction of the flowing liquid. The active cross-area and thus also the hydraulic resistance change depending on the plug position. It is shown in [43] that the analogy of Ohm's law holds for such a system

$$p = R_M(x)i \qquad (46)$$

where p is the difference between the pressures at the ends of the tube, i is the liquid rate, and R_M is the hydraulic resistance, which depends on the normalized position of the plug $x = w/w_{max} \in [0,1]$ (see Fig. 7) according to the formula

$$R_M(x) = \frac{1}{a + bx^3} \qquad (47)$$

The a and b parameters depend on the system geometry and on the liquid viscosity as described in [43].

The state equation is in the form

$$\frac{dx}{dt} = \frac{A}{1 + Bx^3}i \qquad (48)$$

where the parameters $A > 0$, $B > 0$ depend on the geometry according to [43]. This equation is in the form of (18). The corresponding function $F(x)$ for transforming the physical state variable x into the native state variable q (integral of the liquid speed, thus the distance) is

$$F(x) + C = \int \frac{1 + Bx^3}{A} dx = \frac{x}{A}\left(1 + \frac{B}{4}x^3\right) \qquad (49)$$
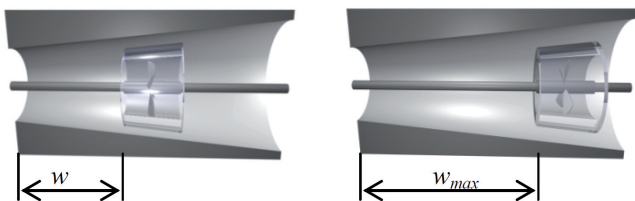
Fig. 7. Example of a hydraulic memristor [43]. Propeller is led by a threaded rod. It is driven by liquid flow and moves between terminal positions $w = 0$ and $w = w_{max}$.

Since $F(x)$ is a one-to-one function for $0 \le x \le 1$ and for all positive A, B, this system is a hydraulic ideal generic memristor.

6. Reliable Models of Ideal Generic Memristors Employing PNSM

Reliable modeling of ideal generic memristors proposed hereinafter is based on replacing the problematic window functions by another functions $x = FI(x_N)$ for a direct transformation of the native state variable x_N into the physical state variable x, or by the functions $x = FI_w(\overline{x}_N)$, where $\overline{x}_N$ is the normalized native state variable. Note that the function $FI_w(\ )$ can be derived, where appropriate, from the corresponding window function $f_w(\ )$ of ideal generic memristor.

Since the function FI serves to the unambiguous transformation of the native-to-physical state variable, it can be designated, analogously to the term PSM (Parameter vs. State Map), as PNSM (Physical vs. Natural State Map). Accordingly, if an unambiguous physical-to-native state variable backward transformation existed in the form of the function $x_N = F(x)$ according to (24), then it could be denoted as NPSM (Natural vs. Physical State Map).

The block-oriented schematic of the model of current-controlled memristor employing PNSM is shown in Fig. 8. A dual version holds for the voltage-controlled memristor. The differential equation of the model is trivial: the charge-current relationship is computed numerically via a simple integration. Such a numerical computation is not burdened with the numerical problem resulting from the limitation of the physical state variable. The physical state variable is then computed from the native variable via nonlinear algebraic transformation in the PNSM block.

As follows from Fig. 8, the function $FI(\ )$ or $FI_w(\ )$ is used for native-to-physical variable transform. The function $F(\ )$ for backward physical-to-native transform (NPSM) is not necessary. It is of practical importance because the model from Fig. 8 will reliably exhibit all the fingerprints of ideal memristor even in the limiting cases from Figs. 5 and 6 and also for $p \to \infty$. Recall that the latter case corresponds to the classical rectangular window, where the function $F(\ )$ in terminal states is not defined, but
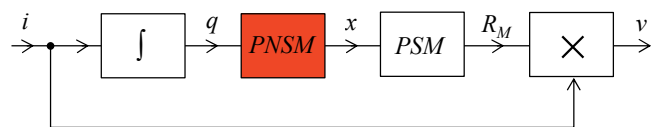
Fig. 8. Block schematic of the modeling via PNSM (Physical vs. Native State Map). The current is integrated into the charge (native state variable). The PNSM block transforms, by nonlinear function, the native state variable into the physical state variable. The latter is then used for computing the memristance in the PSM block (Parameter vs. State Map). The memristor voltage is computed as a product of the memristance and the current.
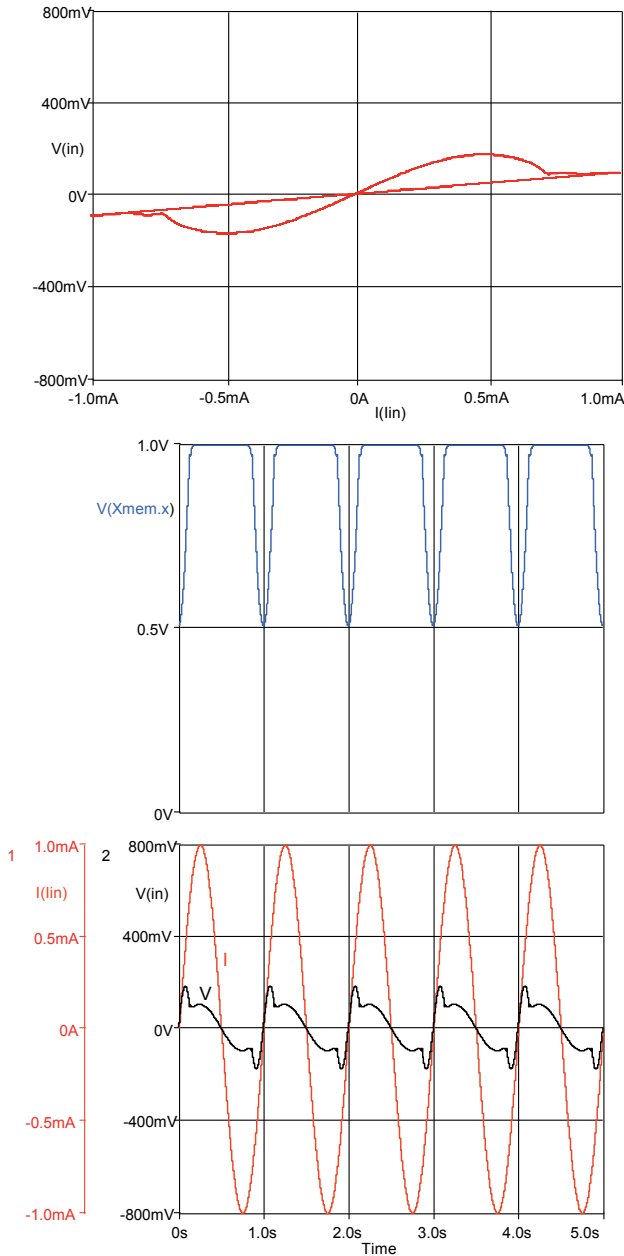
Fig. 9. Simulation results of the model from Fig. 1 for $I_{max} = 1$ mA. The nonlinear dopant drift is modeled correctly: the Prodromakis window is substituted by the equivalent nonlinear block PNSM for the native-to-physical variable transform: v-i odd-symmetric pinched hysteresis loop, periodical waveform of the state variable, the corresponding voltage and current waveforms (immediate transition to periodical steady state). The model exhibits the well-known fingerprints of ideal memristor. Compare with the incorrect behavior of the model of the same memristor in Fig. 3.

the domain of the function $FI(\ )$ is the set of all real numbers (see the note at the end of Sec. 3).

The memristor from Fig. 1 with the Prodromakis window for $p = 10$ and with the simulation parameters from Fig. 3 was modeled via the procedure from Fig. 8. The corresponding transforming function $FI_w(\ )$ for the Prodromakis window was sampled and modeled in PSpice using a look-up table (see the function TABLE in Appen-

dix 2). The source code in Appendix 2 also contains the corresponding conversion from the charge to the normalized charge which appears in the mathematical model of the function $FI_w(\ )$.

The results of the respective simulations in Fig. 9 show the signs of correct behavior of the ideal generic memristor. Comparing them with Fig. 3, which summarizes the preceding simulation of the same memristor, but with the application of the classical method of windowing, one can observe great differences. The window-based method provided correct results of the transient analysis only up to the time instant when, due to numerical problems, the state of the physical variable was erroneously fixed. The behavior of the classical model in Fig. 3 in the subsequent periodical steady state does not match the reality.

7. PNSM Approximation

It turned out that the function $FI_w(\ )$ for normalized native-to-physical state variable transform could serve as a useful replacement of the classical window function for more reliable memristor modeling. It was shown in Sec. 6 that such a function can be modeled in SPICE as a look-up table. However, in some cases it can be more advantageous to model this function by an analytic formula, analogously to the classical window functions. While the window function can be considered an approximation of the ideal rectangular window, the transforming function $FI_w(\ )$ is an approximation of the dashed piece-wise-linear function in Fig. 10. An approximation function is sought that is strictly monotonic, passes through the point $[\bar{q}, x] = [0, 0.5]$ with the slope $dx/d\bar{q} = 1$, converges to 1 or 0 for the native state variable tending to $+\infty$ or $-\infty$, and matches the approximated curve to a degree that can be controlled by an independent parameter.

Recall that the piece-wise-linear transforming function in Fig. 10 corresponds to the ideal rectangular window. It originates via integrating the difference of shifted step functions $\theta(\bar{q}+0.5) - \theta(\bar{q}-0.5)$ with respect to the independent variable $\bar{q}$. The step function $\theta(\bar{q})$ can be approximated by the sigmoidal function

$$\theta(\bar{q}) \approx \frac{1}{1 + e^{-a\bar{q}}}$$

for the parameter $a > 0$. The step function is acquired for $a \to \infty$. For the sake of a good approximation of the above difference of shifted step functions, it is necessary to select a sufficiently high value of the parameter a, ca $a > 10$.

The integration of the difference of shifted sigmoidal functions leads to the analytic formula of the normalized PNSM

$$FI_w(\bar{q}) = \frac{1}{a} \ln \frac{1 + e^{a(\bar{q}+0.5)}}{1 + e^{a(\bar{q}-0.5)}}. \qquad (50)$$
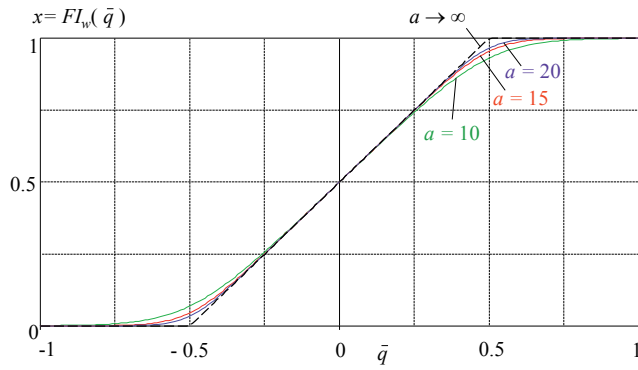
Fig. 10. Normalized PNSM for various values of the parameter
a in the approximation formula (50).

The corresponding graphs for various parameters *a* are in Fig. 10.

In terms of the numerical aspects of the computations, it is not suitable to select the parameter *a* too high (max. ca 100) in view of the risk of overflowing the numeric format (Double-precision binary floating-point, ca 10^{308}) [44]. The function for $a \to \infty$, corresponding to the rectangular window, can easily be programmed in PSpice with the help of the function TABLE, as shown in Appendix 2.

8. Treatment of Initial Conditions

The initial value of the memristance or, more generally, the initial value of the physical state variable from which the memristance can be calculated, serves as one of the input parameters of the transient analysis of circuits that contain memristors.

However, since the differential equation for the native, not physical state variable is solved numerically as part of the PNSM method, it is necessary to enter the initial value of the native variable. It would be possible to convert it from the physical initial state via the transforming function $F_w(\ )$. However, this function may not be available because the PNSM method is built on the utilization of its inverse $FI_w(\ )$.

Figure 11 illustrates a possible solution of this problem.

Knowing the initial value of the physical state variable x_0 and also the model of the function $FI_w(\ )$, then, in order to compute the corresponding initial state $\bar{q}_0$, one row must be added to the SPICE netlist which defines the controlled source Eq0 in Fig. 11. Its voltage is defined such that it is equal to the normalized native state variable $\bar{q}_0$. In this way, SPICE finds the sought value via one iteration run of the numerical solution [43]. Then the initial value $\bar{q}_0$ is used, along with the waveform of the native state variable generated by numerical integration, for computing the waveform of the physical state variable according to Fig. 11.
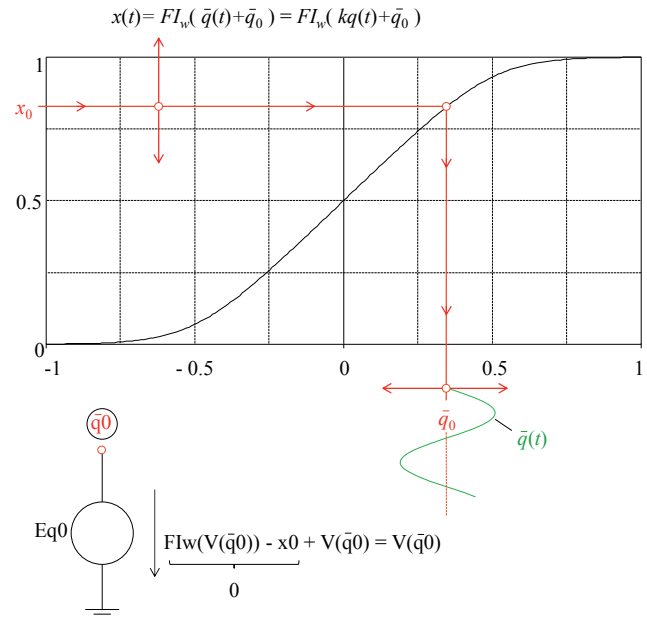

Fig. 11. Explanation as to the SPICE computation of the initial value of the native state variable from the physical state variable. The initial value of the native state variable $\bar{q}_0$ is determined by numerical computation of the voltage of auxiliary source Eq0.

9. Conclusions

The results of the paper can be summarized in two conclusions:

1) The window functions for modeling ideal generic memristors are potential sources of an unacceptable behavior of the models, particularly in the proximity of the terminal states of the memristance. Since these problems appear due to the nature of modeling the rate of the change of the physical state variable versus its distance from its terminal states, they cannot be completely removed other than by refraining from using the window functions.

2) An alternative method of modeling ideal generic memristors is described which does not use the window functions and therefore it does not suffer from the problems under 1).

In the paper it is proposed to use the nonlinear transforming functions between the native and the physical state variables of the memristor (PNSM – Physical vs. Native State Map) instead of the window functions used up to now. There is an unambiguous relation (33) between the two characteristics (excepting the additive constant of integration, which can be set, for example, to zero). Compared with the window functions, the PNSM characteristic brings significant advantages, namely eliminating the stick effect as well as the apparent (nonconvergence) or hidden (incorrect solutions) problems that frequently bother the SPICE users when simulating the circuits containing memristors.

Lists of working input files containing benchmark models of the well-known HP memristor (Appendix 1 – classical window-based modeling, Appendix 2 – the proposed method) are available for readers from the SPICE community with the prospect of quickly mastering such an effective method.

Appendix 1: SPICE Codes, Simulation Based on Window Functions

```
*HP memristor - window approach
.param Ron=100 Roff=1k uv=1e-14 D=10n
+ k={uv*Ron/D**2} deltaR={Roff-Ron} p=10
.param x0=0.5

*Definition of classical window functions
*
*Rectangular window
.func frect(x)={if(x<0,0,if(x>1,0,1))}
*Prodromakis window
.func fprod(x)={(1-(x**2-x+1)**p)/(1-0.75**p)}
*
*classical memristor models employing windowing
*
.subckt MEMRISTOR_Window 1 2
Roff 1 aux {Roff}
Er aux 2 value={-deltaR*v(x)*I(Er)}
Gderi 0 x value={k*I(Er)*fprod(v(x))}
*replace fprod by frect for rectangular window
Cint x 0 1 IC={x0}
Raux x 0 1e9
.ends MEMRISTOR_Window
*
*Sinusoidal current source
.param Imax=1mA f=1Hz
Iin 0 in SIN 0 {Imax} {f}
*memristor in parallel
* Select the appropriate memristor model
Xmem in 0 MEMRISTOR_Window
*
* Transient analysis
.tran 0 5 0 5m skipbp
.probe
.end
```

Appendix 2: SPICE Codes, Simulation Employing PNSM

```
HP memristor - PNSM approach
.param Ron=100 Roff=1k uv=1e-14 D=10n
+ k={uv*Ron/D**2} deltaR={Roff-Ron} p=10
.param x0=0.5 a=50
*
*X to q state maps
*
*Map of Prodromakis window, p = 10
.func FIp10(q)={TABLE(q,
+ -1.5, 2.390u,
+ -1.4, 6.897u,
+ -1.3, 19.911u,
+ -1.2, 57.465u,
+ -1.1, 165.908u,
+ -1,   478.047u,
+ -0.9, 1.374m,
+ -0.8, 3.912m,
```

```
+ -0.7, 10.871m,
+ -0.6, 28.404m,
+ -0.5, 65.685m,
+ -0.4, 127.779m,
+ -0.3, 209.592m,
+ -0.2,  302.423m,
+ -0.1,  400.274m,
+  0,    500.000m,
+  0.1,  599.723m,
+  0.2,  697.574m,
+  0.3,  790.405m,
+  0.4,  872.219m,
+  0.5,  974.314m,
+  0.6,  971.595m,
+  0.7,  989.129m,
+  0.8,  996.088m,
+  0.9,  998.626m,
+  1,    999.522m,
+  1.1,  999.834m,
+  1.2,  999.943m,
+  1.3,  999.980m,
+  1.4,  999.993m,
+  1.5,  999.998m)}
*
*Map of rectangular window
.func FIrect(q)={TABLE(q,-1,0,1,1)}
*
*Universal map - sigmoidal approximation
.func FIsygm(q)=
+{1/a*log((1+exp(a*(q+0.5)))/(1+exp(a*(q-0.5))))}
*
*novel memristor models employing PNSM
*
.subckt MEMRISTOR_PNSM 1 2
Roff 1 aux {Roff}
Er aux 2 value={-deltaR*v(x)*I(Er)}
Gq 0 q value={I(Er)}
Cint q 0 1
Raux q 0 1T
*replace FIp10 in the two lines below
*by FIrect or FIsygm if applicable
Eq0 q0 0 value={FIp10(v(q0))-x0+v(q0)};OP comput.
Ex x 0   value={FIp10(k*v(q)+v(q0))}
.ends MEMRISTOR_PNSM
*
*Sinusoidal current source
.param Imax=1m f=1
Iin 0 in SIN 0 {Imax} {f}
*memristor in parallel
Xmem in 0 MEMRISTOR_PNSM
*
*Transient analysis
.tran 0 5 0 5m skipbp
.probe
.end
```

Acknowledgments

References

[1] STRUKOV, D. B., SNIDER, G. S., STEWART, D. R., WILLIAMS, R. S. The missing memristor found. *Nature*, 2008, vol. 453, p. 80–83. DOI: 10.1038/nature06932

[2] CHUA, L. O. Memristor – The missing circuit element. *IEEE Transactions on Circuit Theory*, 1971, vol. CT-18, no. 5, p. 507 to 519. DOI: 10.1109/TCT.1971.1083337

[3] RÁK, A., CSEREY, G. Macromodeling of the memristor in SPICE. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2010, vol. 29, no. 4, p. 632–636. DOI: 10.1109/TCAD.2010.2042900

[4] BENDERLI, S., WEY, T. A. On SPICE macromodelling of TiO_2 memristors. *Electronics Letters*, 2009, vol. 45, no. 7, p. 377–379. DOI: 10.1049/el.2009.3511

[5] BATAS, D., FIEDLER, H. A memristor Spice implementation and a new approach for magnetic flux controlled memristor modeling. *IEEE Transactions of Nanotechnology*, 2011, vol. 10, no. 2, p. 250–255. DOI: 10.1109/TNANO.2009.2038051

[6] BIOLEK, D., BIOLEK, Z., BIOLKOVÁ, V. SPICE modeling of memristive, memcapacitative and meminductive systems. In *Proceedings of the European Conf. on Circuits Theory and Design.* Antalya (Turkey), 2009, p. 249–252. DOI: 10.1109/ECCTD.2009.5274934

[7] BIOLKOVÁ, V., KOLKA, Z., BIOLEK, D., BIOLEK, Z. Memristor modeling based on its constitutive relation. In *Proceedings of the European Conference of Circuits Technology and Devices.* Tenerife (Spain), 2010, p. 261–264. ISBN: 978-960-474-250-9

[8] RADWAN, A. G., ZIDAN, M. A., SALAMA, K. N. On the mathematical modeling of memristors. In *Proceedings of the 22nd Int. Conference on Microelectronics (ICM 2010).* Cairo (Egypt), 2010, p. 284–287. DOI: 10.1109/ICM.2010.5696139

[9] ASCOLI, A., CORINTO, F., SENGER, V., TETZLAFF, R. Memristor model comparison. *IEEE Circuits and Systems Magazine*, 2013, vol. 13, no. 2, p. 89–105. DOI: 10.1109/MCAS.2013.2256272

[10] MAHVASH, M., PARKER, A.C. A memristor SPICE model for designing memristor circuits. In *Proceedings of the 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS).* Seattle (USA), 2010, p. 989-992. DOI: 10.1109/MWSCAS.2010.5548803

[11] ZHANG, Y., ZHANG, X., YU, J. Approximated SPICE model for memristor. In *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS 2009).* Milpitas (USA), 2009, p. 928–931. DOI: 10.1109/ICCCAS.2009.5250371

[12] JOGLEKAR, Y. N., WOLF, S. J. The elusive memristor: properties of basic electrical circuits. *European Journal of Physics,* 2009, vol. 30, no. 4, p. 661–675. DOI: 10.1088/0143-0807/30/4/001

[13] SHIN, S., KIM, K., KANG, S-M. Compact models for memristors based on charge-flux constitutive relationships. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2010, vol. 29, no. 4, p. 590–598. DOI: 10.1109/TCAD.2010.2042891

[14] BIOLEK, Z., BIOLEK, D., BIOLKOVÁ, V. SPICE model of memristor with nonlinear dopant drift. *Radioengineering,* 2009, vol. 18, no. 2, p. 210–214. ISSN: 1805-9600

[15] KAVEHEI, O., IQBAL, A., KIM, Y. S., ESHRAGHIAN, K., Al-SARAWI, S. F., ABBOTT, D. The fourth element: Characteristics, modelling, and electromagnetic theory of the memristor. *Proceedings of the Royal Society*, 2010, p. 1–28. DOI: 10.1098/rspa.2009.0553 (Published online)

[16] ESHRAGHIAN, K., KAVEHEI, O., CHO, K-R., CHAPPELL, J. M., IQBAL, A., Al-SARAWI, S. F., ABBOTT, D. Memristive device fundamentals and modeling: Applications to circuits and systems simulation. *Proc. of the IEEE*, 2012, vol. 100, no. 6, p. 1991–2007. DOI: 10.1109/JPROC.2012.2188770

[17] PRODROMAKIS, T., PEH, B. P., PAPAVASSILIOU, C., TOUMAZOU, C. A versatile memristor model with nonlinear dopant kinetics. *IEEE Transactions on Electron Devices*, 2011, vol. 58, no. 9, p. 3099–3105. DOI: 10.1109/TED.2011.2158004

[18] KVATINSKY, S., FRIEDMAN, E. G., KOLODNY, A., WEISER, U. C. TEAM: ThrEshold Adaptive Memristor Model. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2013, vol. 60, no. 1, p. 211–221. DOI: 10.1109/TCSI.2012.2215714

[19] YU, J., MU, X., XI, X., WANG, S. A Memristor model with piecewise window function. *Radioengineering*, 2013, vol. 22, no. 4, p. 969–974. ISSN: 1805-9600

[20] TAKAHASHI, Y., SEKINE, T., YOKOYAMA, M. SPICE model of memristive device using Tukey window function. *IEICE Electronics Express*, 2015, vol. 12, no 5, p. 20150149. DOI: 10.1587/elex.12.20150149

[21] CHOWDHURY, J., DAS, J. K., ROUT, N. K. Trigonometric window functions for memristive device modeling. In *Proceedings of the Fifth International Conference on Advanced Computing & Communication Technologies.* Rohtak (India), 2015, p. 157–161. DOI: 10.1109/ACCT.2015.25

[22] CORINTO, F., ASCOLI, A. A boundary condition-based approach to the modeling of memristor nano-structures. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2012, vol. 59, no. 11, p. 2713–2726. DOI: 10.1109/TCSI.2012.2190563

[23] ASCOLI, A., CORINTO, F., TETZLAFF, R. Generalized boundary condition memristor model. *International Journal of Circuit Theory and Applications*, 2015, Early View. DOI: 10.1002/cta.2063

[24] YAKOPCIC, C., TAHA, T. M., SUBRAMANYAM, G., PINO, R. E. Memristor SPICE modeling. *Advances in Neuromorphic Memristor Science and Applications*. Springer Series in Cognitive and Neural Systems, 2012, p. 211–244. ISBN 978-94-007-4491-2

[25] YAKOPCIC, C., TAHA, T. M., SUBRAMANYAM, G., PINO, R. E. Generalized memristive device SPICE model and its application in circuit design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2013, vol. 32, no. 8, p. 1201–1214. DOI: 10.1109/TCAD.2013.2252057

[26] PINO, R. E., BOHL, J. W., McDONALD, N., WYSOCKI, B., ROZWOOD, P., CAMPBELL, K. A., OLBEA, A., TIMILSINA, A. Compact method for modeling and simulation of memristor devices: ion conductor chalcogenide-based memristor devices. In *Proceedings of IEEE/ACM Int. Symp. on Nanoscale Architectures.* Anaheim (USA), 2010, p. 1–4. ISBN: 978-1-4244-8020-3. DOI: 10.1109/NANOARCH.2010.5510936

[27] ABDALA, H., PICKETT, M. D. Spice modeling of memristors. In *Proceedings of Int. Symposium on Circuits and Systems ISCAS 2011.* Rio de Janeiro (Brazil), 2011, p. 1832–1835. DOI: 10.1109/ISCAS.2011.5937942

[28] LEHTONEN, E., LAIHO, M. CNN using memristors for neighborhood connections. In *Proceedings of 12th Int. Workshop on Cellular Nanoscale Networks and their Applications (CNNA).* Berkeley (USA), 2010, p. 1–4. ISBN: 978-1-4244-6679-5. DOI: 10.1109/CNNA.2010.5430304

[29] LAIHO, M., LEHTONEN, E., RUSSEL, A., DUDEK, P. Memristive synapses are becoming reality. *Inst. of Neuromorphic Engineering*, 2010. DOI: 10.2417/1201011.003396

[30] CHANG, T., JO, S-H., KIM, K-H., SHERIDAN, P., GABA, S., LU, W. D. Synaptic behaviors and modeling of a metal oxide

memristor device. *Applied Physics A*, 2011, vol. 201, p. 857–863. DOI: 10.1007/s00339-011-6296-1

[31] LINN, E., SIEMON, A., WASER, R., MENZEL, S. Applicability of well-established memristive models for simulations of resistive switching devices. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2014, vol. 61, no. 8, p. 2402–2410. DOI: 10.1109/TCSI.2014.2332261

[32] SONODA, K., SAKAI, A., MONIWA, M., ISHIKAWA, K., TSUCHIYA, O., INOUE, Y. A compact model of phase-change memory based on rate equations of crystallization and amorphization. *IEEE Transactions on Electron Devices*, 2008, vol. 55, no. 7, p. 1672–1681. DOI: 10.1109/TED.2008.923740

[33] PICKETT, M. D., WILLIAMS, R. S. Sub-100 fJ and sub-nanosecond thermally driven threshold switching in niobium oxide crosspoint nanodevices. *Nanotechnology*, 2012, vol. 23, no. 21, p. 215202. DOI: 10.1088/0957-4484/23/21/215202

[34] STRACHAN, J. P., TORREZAN, A. C., MIAO, F., PICKETT, M. D., YANG, J. J., YI, W., RIBEIRO, G. M. State dynamics and modeling of tantalum oxide memristors. *IEEE Transactions on Electron Devices*, 2013, vol. 60, no. 7, p. 2194–2202. DOI: 10.1109/TED.2013.2264476

[35] PICKETT, M. D., STRUKOV, D. B., BORGHETTI, J. L., YANG, J. J., SNIDER, G. S., STEWART, D. R., WILLIAMS, R. S. Switching dynamics in titanium dioxide memristive devices. *Journal of Applied Physics*, 2009, vol. 106, p. 074508. DOI: 10.1063/1.3236506

[36] WANG, X., CHEN, Y., XI, H., LI, H., DIMITROV, D. Spintronic memristor through spin-torque-induced magnetization motion. *IEEE Electron Device Letters*, 2009, vol. 30, no. 3, p. 294–297. DOI: 10.1109/LED.2008.2012270

[37] PERSHIN, Y. V., Di VENTRA, M. Memory effects in complex materials and nanoscale systems. *Advances in Physics*, 2011, vol. 60, p. 145–227. DOI: 10.1080/00018732.2010.544961

[38] CLAUSS, D. A., RALICH, R. M., RAMSIER, R. D. Hysteresis in a light bulb: connecting electricity and thermodynamics with simple experiments and simulations. *European Journal of Physics,* 2001, vol. 22, no. 4, p. 385–394. DOI: 10.1088/0143-0807/22/4/313

[39] BIOLEK, Z., BIOLEK, D., BIOLKOVÁ, V. (Co)content in circuits with memristive elements. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2015, vol. 62, no. 2, p. 488–496. DOI: 10.1109/TCSI.2014.2364694

[40] FRANCIS, V. J. *Fundamentals of Discharge Tube Circuits*. London (UK): Methuen, 1948.

[41] BIOLEK, D., BIOLEK, Z., BIOLKOVÁ, V., KOLKA, Z. Modeling of TiO$_2$ memristor: from analytic to numerical analyses. *Semiconductor Science Technology*, 2014, vol. 29, p. 125008. DOI: 10.1088/0268-1242/29/12/125008

[42] CHUA, L. If it's pinched it's a memristor. *Semiconductor Science Technology*, 2014, vol. 29, p. 104001. DOI: 10.1088/0268-1242/29/10/104001

[43] BIOLEK, D., BIOLEK, Z. Fourth fundamental circuit element: SPICE modeling and simulation. In *Memristors and Memristive Systems* (Tetzlaff, R. ed). New York: Springer, 2014, p. 105–162. DOI: 10.1007/978-1-4614-9068-5_4

[44] BIOLEK, D., Di VENTRA, M., PERSHIN, Y. V. Reliable SPICE simulations of memristors, memcapacitors and meminductors. *Radioengineering*, 2013, vol. 22, no. 4, p. 945–968. ISSN: 1805-9600

[45] GEORGIOU, P. S. A mathematical framework for the analysis and modelling of memristor nanodevices. *PhD Thesis*, 2013, Imperial College London, 212 p.

[46] CHUA, L. O., KANG, S-M. Memristive devices and systems. *Proceedings of the IEEE*, 1976, vol. 64, no. 2, p. 209–223. DOI: 10.1109/PROC.1976.10092

[47] BIOLEK, Z., BIOLEK, D., BIOLKOVÁ, V., KOLKA, Z. Some fingerprints of ideal memristors. In *Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS)*. Beijing (China), 2013, p. 201–204. ISBN: 978-1-4673-5760-9. DOI: 10.1109/ISCAS.2013.6571817

[48] LI, Q., XU, H., LIU, H., TIAN, X. Variable-relation criterion for analysis of the memristor. *International Science Index*, 2011, vol. 5, no. 12, waset.org/Publication/10978.

[49] ADHIKARI, S. P., SAH, M. P., CHUA, L. O., KIM, H. Three fingerprints of memristor. *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 2013, vol. 60, no. 11, p. 3008–3021. DOI: 10.1109/TCSI.2013.2256171

[50] BIOLEK, D., BIOLEK, Z., BIOLKOVÁ, V. Pinched hysteretic loops of ideal memristors, memcapacitors and meminductors must be 'self-crossing'. *Electronics Letters*, 2011, vol. 47, no. 25, p. 1385–1387. DOI: 10.1049/el.2011.2913

[51] PERSHIN, Y. V., Di VENTRA, M. SPICE model of memristive devices with threshold. *Radioengineering*, 2013, vol. 22, no. 2, p. 485-489. ISSN: 1805-9600

[52] BIOLEK, D., KOLKA, Z., BIOLKOVÁ, V. Memristive systems for analog signal processing. In *Proceedings of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. Melbourne (Australia), 2014, p. 2588–2591. DOI: 10.1109/ISCAS.2014.6865702

[53] CHUA, L., SBITNEV, V., KIM, H. Hodgkin-Huxley axon is made of memristors. *International Journal of Bifurcation and Chaos*, 2012, vol. 22, no. 3, p. 1230011-1-48. DOI: 10.1142/S021812741230011X

[54] BIOLEK, D., BIOLKOVÁ, V., KOLKA, Z. Spice models of memristive devices forming a model of Hodgkin-Huxley axon. In *Proceedings of the 18th International IEEE Conference on Digital Signal Processing (DSP)*. Santorini (Spain), 2013, p. 1-5. DOI: 10.1109/ICDSP.2013.6622743

[55] WEISSTEIN, E. W. *Logistic Equation*. From MathWorld-A Wolfram Web Resource http://mathworld.wolfram.com/LogisticEquation.html

[56] IELMINI, D., LARENTIS, S., BALATTI, S. Physical modeling of voltage-driven resistive switching in oxide RRAM. In *Proceedings of the IEEE Int. Integrated Reliability Workshop Final Report (IRW)*. South Lake Tahoe (USA), 2012, p. 9–15. DOI: 10.1109/IIRW.2012.6468905

[57] IELMINI, D. Modeling the universal set/reset characteristics of bipolar RRAM by field- and temperature-driven filament growth. *IEEE Transactions on Electron Devices*, 2011, vol. 58, no. 12, p. 4309–4317. DOI: 10.1109/TED.2011.2167513

[58] MIRANDA, E., JIMÉNEZ, D., SUNÉ, J. The quantum point-contact memristor. *IEEE Electron Device Letters*, 2012, vol. 33, no. 10, p. 1474–1476. DOI: 10.1109/LED.2012.2210185

[59] WANG, X., CHEN, Y., GU, Y., LI, H. Spintronic memristor temperature sensor. *IEEE Electron Device Letters*, 2010, vol. 31, no. 1, p. 20–22. DOI: 10.1109/LED.2009.2035643

About the Authors ...

Dalibor BIOLEK, Zdeněk BIOLEK, Viera BIOLKOVÁ – for the CV see page 377 of this Issue.

Zdeněk KOLKA – for the CV see page 383 of this Issue.