Wave Digital Emulation of Charge- or Flux-Controlled Memristors

Karlheinz Ochs
Ruhr-Universität Bochum
Institute for Digital Communication Systems
Bochum, Germany
Email: Karlheinz.Ochs@rub.de

Enver Solan
Ruhr-Universität Bochum
Institute for Digital Communication Systems
Bochum, Germany
Email: Enver.Solan@rub.de

***Abstract*—Memristors are passive elementary circuit elements, which essentially are resistors with memory. This property makes the memristor well suited for applications in neuromorphic computation circuits. In such circuits the functionality mainly depends on the memristor. Hence, different desired functionalities for the resulting circuit need appropriate devices. Normally, memristors are not commercially available. This makes the use of emulators, which mimic the memristive behavior, expedient. For this purpose, we introduce a method based on wave digital virtualization for emulating real memristors after identifying them from i-u-measurements. The so virtualized memristor can be directly integrated in a real circuit.**

I. Introduction

Memristors behave like a resistor with memory [1]. Since they are additionally compact and energy efficient, these devices are desired for different kind of applications, like nonvolatile memory [2] or reconfigurable logic circuits [3]. In view of a similar functionality of memristors and synapses they are useful for neuromorphic computation circuits [4]. Even so, the memristor was for decades a solely hypothetical device until a HP research group announced that they built a memristor by means of nanotechnology in 2008 [5]. Nevertheless, the commercial use of memristors is not common yet and consequently investigations with real devices can be costly.

Simulations, which use mathematical models of memristors, can help to understand the behavior of a single device or of complete circuits including memristors. However, in the context of real circuits, these simulations are less suited. To this end, memristor emulators are needed. These emulators mimic the memristive behavior and can be directly integrated into a real electrical circuit. Most popular emulators are analog hardware emulators based on transistors and operational amplifiers, which need an external power supply [1], [6]. More flexible than a pure analog emulator is a mixed signal emulator whose functionality can be programmed via different interfaces on a micro-controller [7]. All of these emulators mimic the behavior of memristive devices based on mathematical models. On the other hand, there are differences between mathematically modeled and real devices. These differences can affect the functionality of the overall system. Hence, for accurate investigations, there is a need of emulators, which mimic the real device.

The present paper uses the wave digital concept to virtualize memristors. This method is well suited for emulations, because of a variety of stability properties [8], [9]. Due to these benefits, this emulator is reusable, because it is not limited to a particular application. On this basis, we extend the idea to get a possibility for virtualizing real memristors from measured i-u-curves. Therefore, the resulting algorithmic model can be implemented in various ways. For example, we can produce a hardware emulator on a hard-wired chip or we can use a digital signal processor system for a software emulator. In particular, with this virtualization it is possible to make reproducible analyzes conversely to real devices.

The paper is organized as follows: First, some basic definitions are given. Sec. III shows how a real memristor can be virtualized with the wave digital concept based on the identification from measurements. In Sec. IV, simulation results show the usability and quality of the wave digital virtualization of identified memristors. In the last section a conclusion of the main results is drawn.

II. Memristor

Chua defined the memristor axiomatically by its constitutive relation between flux φ and charge q, see [1]. In this manner, the memristor can be charge- or flux-controlled. For the former, we get the charge-controlled constitutive relation and Ohm's law

$$\varphi = \hat{\varphi}(q)\,, \quad u = M(q)i\,, \quad \text{with} \quad M(q) = \frac{\partial\hat{\varphi}}{\partial q}\,, \tag{1}$$

where $M(q)$ denotes the charge-dependent memristance. Analogous to this,

$$q = \hat{q}(\varphi)\,, \quad i = W(\varphi)u\,, \quad \text{with} \quad W(\varphi) = \frac{\partial\hat{q}}{\partial\varphi} \tag{2}$$

describes a flux-dependent memristor with the flux-controlled memductance $W(\varphi)$, which is the reciprocal value of a memristance.

In this work, the memristor will be viewed as a pure mathematical model, where the quantities φ and q need not to have the physical interpretation of a magnetic flux and electrical charge, respectively. Indeed, they are the integrals of voltage u and current i in a mathematical sense.

It should be stressed that, with knowledge of the initial values of flux and charge, the constitutive relation is a unique representation of a memristor while the typical pinched hysteresis loop in the i-u-plane is not as it changes for different input signals. Therefore, the determination of the output is possible for an arbitrary input, cf. [10].

III. Wave Digital Realization of Real Memristor

Here, we recapitulate shortly the principle modeling issue for memristors in the wave domain. We use this model for realizing emulators of real memristors after identifying them from i-u-measurements.

A. Memristor in the Wave Domain

The wave digital modeling is a concept to virtualize electrical circuits based on digital signal processing and on scattering parameter theory. Hence, wave quantities a and b are used as signal parameters, which are interrelated with voltages and currents by a bijective mapping

$$a = u + Ri\,, \qquad b = u - Ri\,, \qquad \text{with} \qquad R > 0\,. \tag{3}$$

The positive constant R is the port resistance, a and b denote the incident and reflected wave at one port, respectively. Consequently, the electrical description of each device in a circuit has to be translated to the wave domain.

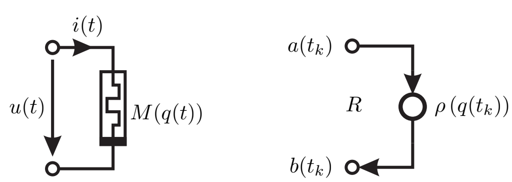

Fig. 1. A memristor (left) and its wave digital realization (right).

The electrical device and the corresponding wave digital realization of a charge-controlled memristor are shown in Fig. 1. In order to distinguish between the continuous-time domain for the analog device and discrete-time domain for the wave digital realization the time arguments are declared. The variable t_k denotes equidistant sampling instances $t_k = t_0 + kT$, with index $k \in \mathbb{N}$, sampling period T, and initial time t_0. By replacing the voltage and current with the wave quantities, in accordance with (3), equation (1) becomes

$$b(t_k) = \rho(q(t_k))a(t_k)\,, \quad \text{with} \quad \rho(q(t_k)) = \frac{M - R}{M + R}\,, \tag{4}$$

where ρ is the reflection coefficient. This equation is implicit with respect to b. That is why the signal flow graph in Fig. 1 contains an invisible algebraic loop. The missing algebraic loop of Fig. 1 can be found in the extended wave digital signal flow graph of Fig. 2 (dashed) including three units: A transformation unit, a processing unit and the computation of the reflection coefficient. The implicit part of equation (4) is visualized by a red-colored delay-free directed loop.

This implicit equation has to be solved at each instance. To this end, one can use a fix-point iteration, which requires

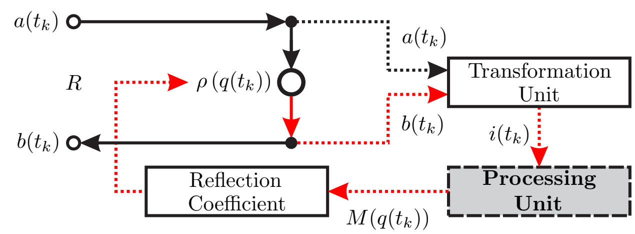

Fig. 2. Extended signal flow graph shown as dashed lines for calculating the current memristance with an iterative procedure. The red path is repeated several times during one simulation time step for an accurate calculation.

repeating the computations of the red path several times while the signals on the black paths remain constant. In case of a high sampling rate, the signals vary slowly from instance to instance and the number of iterations can be significantly reduced. Indeed, the simulation results of the next sections were accurate enough although no iteration has been used.

The transformation unit maps the waves a and b onto the current i subject to (3). The processing unit determines from this current the charge-dependent memristance $M(q)$, which is used by a computation unit to compute the reflection coefficient compliant with (4). Since the processing unit is more elaborated and includes the initialized identification procedure, a detailed explanation is separated from this section.

B. Processing Unit

The processing unit is subdivided in three units as depicted in Fig. 3. It consists of an integrator, a post-processing unit, and an initialization unit.

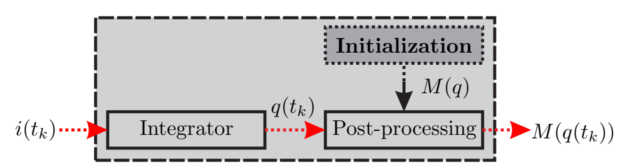

Fig. 3. Block diagram of the processing unit.

With the integrator, we get the charge from the current, which is needed for the charge-controlled memristor defined in (1). Concerning this, we use a wave digital integrator based on an ideal current source interconnected to a capacitance. Because of consistent numerical accuracy with the wave digital method, we use the trapezoidal rule for the numerical integration [8]

$$q(t_k) \approx q(t_{k-1}) + \frac{T}{2}[i(t_k) + i(t_{k-1})]\,.$$

The post-processing unit is initialized with the before identified memristance $M(q)$ as a function of charge and it calculates the current memristance sample $M(q(t_k))$ from the input charge sample $q(t_k)$. The initializing unit derives the memristance function to the post-processing unit.

C. Memristor Identification Procedure

For a unique characterization of a memristor one need to determine the constitutive relation from a measured i-u-curve.

To this end, we use the same integrator like before to integrate the current and voltage from the measured data, see Fig. 4.

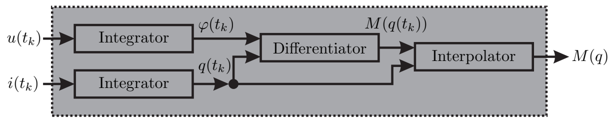

Fig. 4. Initialization unit for sampling values.

According to equation (1) and (2), a numerical differentiation of the charge and flux samples is used to achieve an approximation of the memristance samples $M(q(t_k))$. For the numerical differentiation, we used the backward differences

$$M(q(t_k)) \approx \frac{\varphi(t_k) - \varphi(t_{k-1})}{q(t_k) - q(t_{k-1})} \, .$$

In order to increase the numerical accuracy, one may use other numerical differentiation methods, like central differences, instead.

With this, the desired charge-dependent memristance $M(q)$ can be obtained after a suitable interpolation of the associated samples of charge $q(t_k)$ and memristance values $M(q(t_k))$. For the sake of simplicity, we have used a linear interpolation. Anyway, one can choose other interpolating techniques depending on particular requirements. In order to increase the efficiency of a real-time implementation, the memristance of real or simulated memristors can also be approximated by curve fitting techniques.

It is important to understand, that this method is equally suitable to identify a flux-controlled memristor. Even from a real measured charge-dependent memristor, we can generate the flux-dependent memductance function $W(\varphi)$ and use it for further investigations.

IV. Simulation

The aim of the simulation is to validate the virtualization and identification concept for real memristors. Due to lack of measurements from a real memristor, we exploit an LTSpice simulation, where the memristor is directly coupled to an ideal voltage source. For this, we used the ideal memristor model of [11]

$$M(q) = R_1 + \frac{R_0 - R_1}{1 + \gamma \mathrm{e}^{-4\alpha q}} \, , \quad \text{with} \quad \gamma = \frac{M(0) - R_0}{R_1 - M(0)} \, , \tag{5}$$

where γ is an auxiliary constant depending on initial values. The charge-dependent memristance of this memristor has the parameters: $R_0 = 100\Omega$, $R_1 = 10\mathrm{k}\Omega$, $M(q_0) = 5\mathrm{k}\Omega$ and $\alpha = 10^3$. The advantage of this memristor model is the analytical description of the memristance as a function of charge, from which (with a somewhat lengthy but elementary computation) the constitutive relation

$$\hat{\varphi}(q) = \varphi_0 + R_0[q - q_0] + [R_0 - R_1] \ln \left| \frac{1 + \gamma \mathrm{e}^{-4\alpha q}}{1 + \gamma \mathrm{e}^{-4\alpha q_0}} \right| \tag{6}$$

can be deducted. A description of the flux-dependent memductance cannot be expressed analytically, so that we use the charge-dependent memristance to identify the flux-dependent memductance in the simulations. For this purpose, a sine input voltage is used for the excitation of the memristor:

$$s(t) = E \sin(2\pi \, F \, t) \, , \ \text{ with } E \quad = 1 \text{ V and } F = 1 \text{ Hz} \, .$$

From this, we derive the flux

$$\varphi(t) = \Phi_s \sin^2(\pi F t) \, , \ \text{with } \varphi(0) = 0 \, , \ \Phi_s = \frac{E}{\pi F} \, . \tag{7}$$

An LTSpice simulation yields the i-u-curve of Fig. 5, which has the typical form of a pinched hysteresis loop, cf. [11].

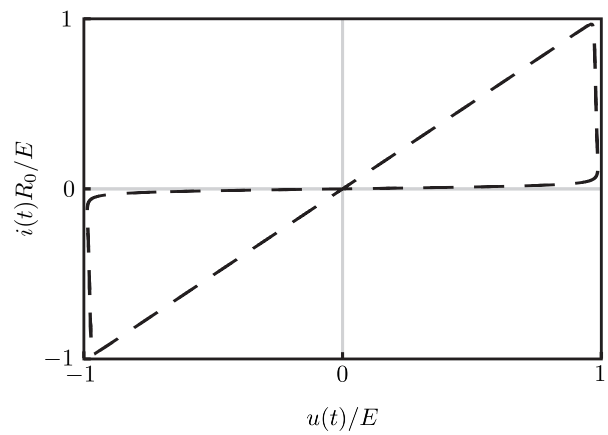

Fig. 5. LTSpice simulated i-u-curve for the sine input voltage.

For the virtualization, the memristance function $M(q)$ or alternatively the memductance $W(\varphi)$ is needed. Since we know the analytical function $M(q)$ with all parameters, we could use it directly. However, we are more interested in virtualizing such a memristor if we only know voltage and current measurements of this memristor. Accordingly, it is required to identify the flux-dependent memductance.

A. Memristor Identification

In fact, we know the voltage and current samples as well as the associated sampling instances, which are non-uniform because the LTSpice simulation uses a variable step size for the numerical integration. From this information, we can uniformly re-sample the signals and process them as explained in Fig. 4. This yields the flux-dependent memductance $W(\varphi)$ of Fig. 6.

B. Memristor Validation

Now, the wave digital model of a memristor will be validated. For this, we have to bear the dynamic range of the memristor identification in mind. Because of the excitation, the flux has limited amplitude as given in (7), which also limits the range of validity of our identification to $\varphi \in [0, \Phi_s]$, cf. Fig. 6. Considering this limitation, a periodic triangular voltage

$$v(t) = \frac{2E}{\pi} \arcsin(\sin(2\pi F t))$$

will excite the virtualized memristor, with the same amplitude and period as in the identification case. Then, one has

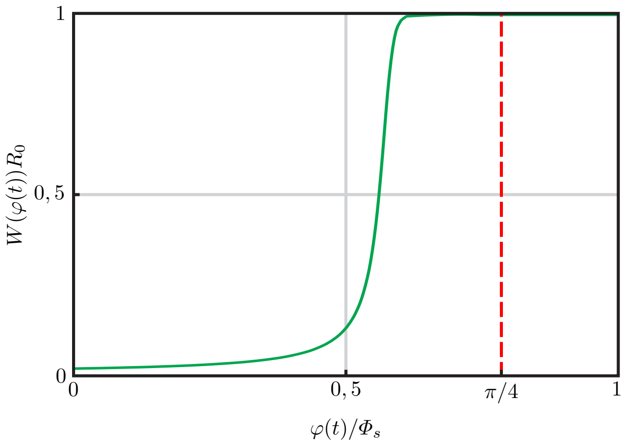

Fig. 6. Memductance vs. flux derived from the i-u-curve of Fig. 5.

$\Phi_v/\Phi_s = \pi/4 < 1$, where Φ_v denotes the amplitude of resulting flux for input voltage $v(t)$. Regarding to Fig. 6 it is guaranteed that the flux of this excitation voltage remains inside the range of validity.

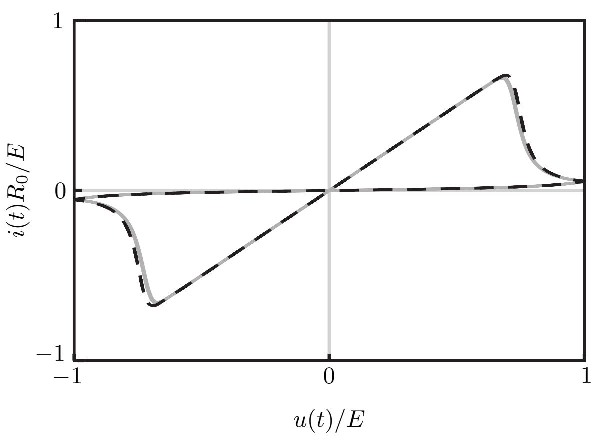

Fig. 7. Simulated i-u-curve for the triangular input voltage: Wave digital simulation (gray) and LTSpice simulation (dashed).

The simulation result is depicted in Fig. 7. The dashed curve shows the reference simulation with LTSpice while the gray curve shows the wave digital simulation results of the virtualized memristor. Since there is almost an exact coincidence between the two simulated curves, we can state that the memristor successfully has been virtualized after identifying it from the simulated voltage and current values of Fig. 5.

V. Conclusion

In this paper, an emulator for real charge- or flux-controlled memristors has been proposed. For this purpose, the wave digital concept has been used, which yields an algorithmic model for implementation on a digital signal processing system or on a hard-wired chip. Emulation of real memristors needs to extract the constitutive relation from measured i-u-curves. For this matter, we have introduced a reliable identification procedure. Simulation results has shown, that the charge-controlled memristor has been successfully identified and emulated with this method. Even, if we only have the charge-controlled constitutive relation in analytical form, we can use this approach to derive the reciprocal flux controlled constitutive relation numerically.

If a memristors is identified once, one can use this emulation method to replace it in real circuits. Reproducible analyzes with real devices are hard, because of parameter variations during the fabrication process. The introduced method can be used to overcome this problem. Thus, we can make reproducible analyzes by emulating real devices.

Acknowledgment

The financial support by the Deutsche Forschungsgemeinschaft through FOR 2093 is gratefully acknowledged.

References

[1] L. Chua: *Memristor - The Missing Circuit Element*, IEEE Transactions on Circuit Theory, Vol. CT-18, No. 5, pp. 507-509, September 1971.

[2] Y. Ho, M. G. Huang, P. Li: *Nonvolatile memristor memory: Device characteristics and design implications*, 2009 IEEE/ACM International Conference on Computer-Aided Design, pp. 485-490, November 2009.

[3] Q. Xia, W. Robinett, M. W. Cumbie, N. Banerjee, J. T. Cardinali, J. J. Yang, W. Wu, X. Li, W. M. Tong, B. D. Strukov, S. G. Snider, G. Medeiros-Ribeiro, R. S. Williams: *Memristor-CMOS Hybrid Integrated Circuits for Reconfigurable Logic*, NANO Letters, Vol.9, No.10, pp. 3640-3645, September 2009.

[4] M. Ziegler, K. Ochs, M. Hansen, H. Kohlstedt: *An electronic implementation of amoeba anticipation*, Applied Physics A, Vol. 114, No. 2, pp.1-6, February 2013.

[5] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams: *The missing memristor found*, Nature, Vol. 453, pp. 80-83, May 2008.

[6] D. M. Biolek, V. Biolkova, Z. Kolka, Z. Biolek: *Passive Fully Floating Emulator of Memristive Device for Laboratory Experiments*, Advances in Electrical and Computer Engineering, 2015.

[7] Y. V. Pershin, D. M. Ventra: *Practical Approach to Programmable Analog Circuits With Memristors*, IEEE Transactions on Circuits and Systems I: Regular Papers, Vol.57, No.8, pp. 1857-1864, February 2010.

[8] A. Fettweis: *Wave Digital Filters: Theory and Practice*, Proceedings of the IEEE, Vol. 74, No. 2, pp. 270-327, February 1986.

[9] A. Fettweis: *Pseudo-passivity, sensitivity and stability of wave digital filters*, IEEE Transactions on Circuit Theory, Vol. 19, No. 6, November 1972.

[10] L. Chua: *Resistance switching memories are memristors*, Applied Physics A (Materials Science & Processing), Vol. 102, No.4, pp. 765-783, January 2011.

[11] D. Biolek, D. M. Ventra, Y. V. Pershin: *Reliable SPICE Simulations of Memristors, Memcapacitors and Meminductors*, Radioengineering, Vol. 22, No. 4, July 2013.