Mimicking Spike-Timing-Dependent Plasticity with Emulated Memristors

A. Cisternas Ferri, A. Rapoport, P. I. Fierens, and G. A. Patterson

***Abstract*—Memristors have found application in neuromorphic circuits and it has been shown that, under certain conditions, they may mimic the behavior of neuronal synapses. Experimenting with memristor-based synapses has several problems. Indeed, memristor samples are difficult to obtain and tweaking their parameters to adapt their behavior requires a long fabrication and testing process. For this reason, simulation and emulation become attractive alternatives for the study of memristive systems. We postulate that emulation has the advantage of working with real-world circuits and not stylized simulation models. In this paper, we propose a basic memristor emulation architecture and show that it can be used to mimic certain characteristics of synapses.**

***Index Terms*—Arduino, Memristor, STDP, Synapse.**

I. Introduction

The use of memristive systems for computation and machine learning has been extensively explored in the literature (see, *e.g.*, [1]–[5] and references therein). In particular, memristors have found application in a large number of neuromorphic circuits [6]–[17]. In this paper we focus on the ability of memristors to mimic the behavior of neuronal synapses.

A synapse is a biological structure that takes place at the junction of two nerve cells, that is, between the axon of one neuron and a dendrite of another. It is currently accepted that synapses play a fundamental role in memory formation and learning by changing the strength of the connection between two cells. The Spike-Timing-Dependent Plasticity (STDP) process is a biological model that describes how the strength of synaptic connections is modulated [18]–[20]. Mainly, this process accounts for the time difference between the action potentials fired by the presynaptic and the postsynaptic neurons. While there is a strengthening of the synaptic connection when the presynaptic neuron fires immediately before the postsynaptic cell, there is a weakening of the connection if the firing order is reversed.

In recent years, the possibility of implementing memristors as synaptic connectors has been exhaustively studied because synapses can be viewed as two-terminal devices with variable conductance [6]–[10], [15]. It has been argued that the conductance of a particular family of memristors can be modulated by applying specific signals, namely action potentials, whose particular shapes are the crucial components that defines the emerging STDP behavior [8], [10]. Some researchers have focused on developing sophisticated communication protocols in order to adjust the voltage applied to the memristor as a function of the presynaptic and postsynaptic electrical signals [6]. In general, these approaches have qualitative success simulating the synaptic behavior under very particular signal conditions or resorting to very complex circuits which make them impractical when designing large-scale computing architectures. Recent results have shown that memristors with diffusive dynamics exhibit a temporal response similar to that of a synapse [21], [22], enabling to mimic synaptic functions, such as STDP, with a minimum of hardware requirements and signal conditioning. In this work, we focus on characterizing the influence of the diffusive parameters of a particular type of memristors when considering it as a synapse [23], [24]. Moreover, there is a mathematical model of this type of memristor that not only accounts for the static current-voltage relationship, but it also accurately describes the dynamic response which includes diffusive effects [25].

Since samples of memristors are not widely available, simulation and emulation become attractive alternatives for the study of memristive systems. Furthermore, both simulation and emulation allow to easily explore different parameters and configurations. Although some researchers have focused on simulation-based approaches (see, *e.g.*, [26]), we find that emulation has the added advantage of working with real-world circuits and not stylized simulation models. For this reason, in this paper we develop a versatile emulator, based on off-the-shelf hardware, that may emulate different types of memristors. We verify the correctness of the design with a simple memristor model presented by Strukov *et al*. [27] and we also implement the more complex model in Patterson *et al*. [25] which is used to mimic a STPD-like process.

The remaining of the paper is organized as follows. In Section II, we briefly present the design of a memristor emulator and we verify its correctness with two well-known memristor models. In Section III, we show that the emulator can be used to mimic the behavior of a neuronal synapse. Finally, we present some conclusions in Section IV

II. Emulator architecture

Several emulation designs have been proposed in the literature [28]–[34]. In this work, we follow the approach of Olumodeji and Gottardi [33] based on a microcontroller from the Arduino family and a digital potentiometer. The digital potentiometer acts as the memristor itself. The microcontroller senses the current that traverses the potentiometer and changes its resistance according to a given memristor model.

A. Cisternas Ferri and A. Rapoport are with the Department of Physics of the University of Buenos Aires, Argentina. P. I. Fierens and G. A. Patterson are with the Buenos Aires Institute of Technology (ITBA) and the National Scientific and Technical Research Council (CONICET), Argentina.

A. Memristor models

A memristor is a two-terminal device whose electrical resistance can be modulated by an external electric field [35]. In principle, almost any model of memristors can be emulated with the proposed architecture. However, in this paper, we focus on models where the resistance is given by

$$R(t) = R_{\text{on}} w(t) + R_{\text{off}} \left(1 - w(t)\right), \tag{1}$$

where R_{on} and R_{off} are the low and high-resistance levels of the memristor, respectively. Memristor's resistance is governed by an adimensional parameter $w(t)$ which takes values in $[0, 1]$. Actual models differ in how $w(t)$ evolves with the current $i(t)$ that traverses the memristor. One of the simplest evolution models is that proposed by Strukov *et al.* [27]

$$\frac{d}{dt} w(t) = \mu R_{\text{on}}\ i(t). \tag{2}$$

In this model, $w(t)$ represents the relative position of a barrier between a highly-doped (low resistance) and a lightly-doped (high resistance) region. The constant μ accounts for the mobility.

The microcontroller senses the current $i(t)$ (as a voltage drop on the digital potentiometer) and numerically integrates Eq. (2) (*e.g.*, using Euler's algorithm). Then, it changes the resistance of the potentiometer according to Eq. (1). Since the digital potentiometer has a limited resolution, its resistance is set to the achievable value which is closest to that prescribed by the model. Two limitations are, thus, evident even in the implementation of this simple model. First, the speed of computation of the microcontroller limits the lowest achievable integration time step. Second, the resolution of the digital potentiometer also puts a cap on the minimum resistance change that can be emulated. Both factors together represent a limitation to the acceptable input frequencies.

Although there are many variations to the simple model in Eq. (2), in this work we use a recent model which is studied in Patterson *et al.* [25] and given by

$$\frac{d}{dt} w(t) = \frac{\lambda(v(t)) - w(t)}{\tau_0} \exp\left(\frac{|v(t)|}{v_0}\right), \tag{3}$$

$$\lambda(v(t)) = \min\left\{\Gamma^{-}(v(t)), \max\left[\lambda(v(t-h)), \Gamma^{+}(v(t))\right]\right\}, \tag{4}$$

$$\Gamma^{\pm}(v) = \frac{1}{1 + e^{-\alpha(v \mp \delta)}}. \tag{5}$$

The sigmoid functions $\Gamma^{\pm}$ describe the formation and destruction of conducting filaments as a function of the voltage drop $v(t)$ on the memristor. Eq. (3) implies that the memristance does not follow input changes immediately. Indeed, τ_0 is a characteristic response time. Parameters v_0, α and δ are positive constants. The integration time step is h in Eq. (4). This memristor model is a variation of that in Refs. [36], [37], which was experimentally validated in [23], [24].

The microcontroller senses the voltage drop on the memristor $v(t)$ and numerically integrates Eq. (3). Limitations imposed by the emulator's hardware are similar to that found in the simpler model. Indeed, microcontroller's computation speed and the resolution of the potentiometer impose a cap on the input frequency.

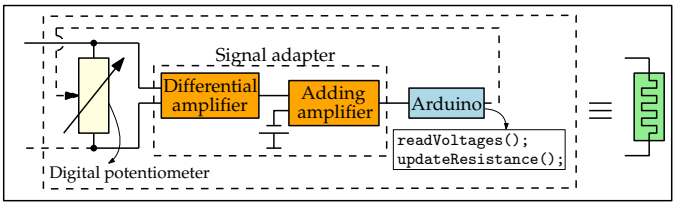

Fig. 1. Emulator's schematic. The Arduino microcontroller senses the voltage drop across the digital potentiometer and integrates the differential equations of the corresponding memristor model. The resistance of the potentiometer is changed accordingly. Voltage limitations of the analog-to-digital converters of the Arduino require the actual voltage on the potentiometer to be adapted.

B. Design details

A schematic of the emulator design is shown in Fig. 1. As explained in Section II-A, the main limitations come from the computational speed of the microcontroller and the resolution of the digital potentiometer. In order to free up the emulator from the first limitation, we used an Arduino Due which is an Arduino boards with one of the fastest processors, an Atmel ARM-based processor SAM3X running at a clock frequency of 84 MHz [38]. The minimum achievable integration time step with the Arduino Due was $\sim$ 400 µs. As a rule of thumb, using Euler's algorithm, integration steps must be much smaller than the characteristic times of the system. This implies that the input frequency needs to be $\ll 2.5$ kHz.

We used a Renesas X9C103P [39] digital potentiometer. Although this potentiometer has a low resolution (it can be set at only 100 resistance values), it has the advantage of accepting positive and negative voltages. A higher resolution potentiometer might be desirable for future implementations, but the X9C103P has shown to have a sufficient resolution for the experiments conducted in this work. The resistance of the potentiometer can be set on any of the uniformly distributed 100 values in the (measured) range $(35.0 \pm 0.8)\ \Omega$ - (9.5 ± 0.1) kΩ.

We used code developed by Timo Fager [40] to interact with the digital potentiometer. The resistance of the X9C103P must be changed in a sequential manner through its increment/decrement input. This sequential feature may lead to further delays and, hence, higher integration time steps. For this reason, other interfaces might be convenient for future developments. Actually, we implemented other versions of the emulator with potentiometers that used a Serial Peripheral Interface (SPI).

The inputs to the analog-to-digital converters (ADCs) of the Arduino Due must be limited to the range 0-3.3 V. Since the voltage drop on the memristor (or a resistor in series with it) can be negative, the sensed voltage is adapted before being measured by the microcontroller (see Fig. 1). First, a differential amplifier acts as a buffer so the circuit is not loaded

by the measurement setup and an offset voltage is added to make all input voltages non-negative.

The accuracy of the measurements may impose another limitation to the emulation design. Indeed, when the sensed voltage is small, the relative measurement error might be high and this error might propagate as the model's equations are integrated. Atmel's SAM3X has 12-bit ADCs and this resolution (LSB < 1 mV) seemed sufficient for our experiments. However, measurement errors may be higher than the nominal resolution. In this respect, a careful design of the signal adaptation circuit in Fig. 1 is needed.

In practice, measurements on the printed circuit board revealed noise sources (among them, perhaps, digital clock feedthrough) with amplitudes at least an order of magnitude higher than the ADCs resolution. In some cases, we enhanced the measurements with a low pass filter to reject noise.

C. Experimental verification of the emulator

We implemented both models described in Section II-A and compared the measurements with simulations to verify the results. A view of the experimental setup is shown in Fig. 2(a) and the circuit schematic can be seen in Fig. 2(b).

Fig. 3 shows the response curve i-v for the model described by Eq. (2) when the memristor is driven by a sinusoid signal of amplitude A and variable frequency. The voltage $v(t)$ stands for the voltage drop between emulator terminals. As it can be observed, the hysteresis disappears as the signal frequency is increased.

Fig. 4 shows results for the model described by Eqs. (3)-(5). Signal frequencies are lower than in the case of Fig. 3 because the time required to compute each integration step increases with the complexity of the model. Indeed, the maximum work frequency is conditioned by the type of model considered.

Experimental results were verified to be in complete agreement with numerical simulations. In particular, Fig. 4 shows numerical simulations for the model described in Ref. [25].

III. Synapse mimicking

Learning rules are the different processes through which connections between network elements are adjusted. Hebb postulated [41] a principle that states that the synaptic transmission efficiency is driven by correlations between pre and postsynaptic neuronal activity. One of the usual protocols to study the change in the synaptic strength is by means the so-called Spike-Timing-Dependent Plasticity (STDP) process. This protocol can be summarized as follows. Given a pair of neurons that are synaptically coupled, a test pulse is applied in the presynaptic neuron that provokes a postsynaptic potential in the other neuron. Then, both neurons are stimulated with a signal of periodic pulses where one of the signals is delayed a given time Δt with respect to the other. Whenever pre and postsynaptic pulses overlap, their net effect is that of a single driving pulse with amplitude equal to the difference of potentials between the pre and postsynaptic signals. After this stimulation, a new test pulse is injected into the presynaptic neuron measuring the level of the postsynaptic potential. The difference between the levels of postsynaptic potential when applying the different test pulses quantifies the change of the synaptic connection.

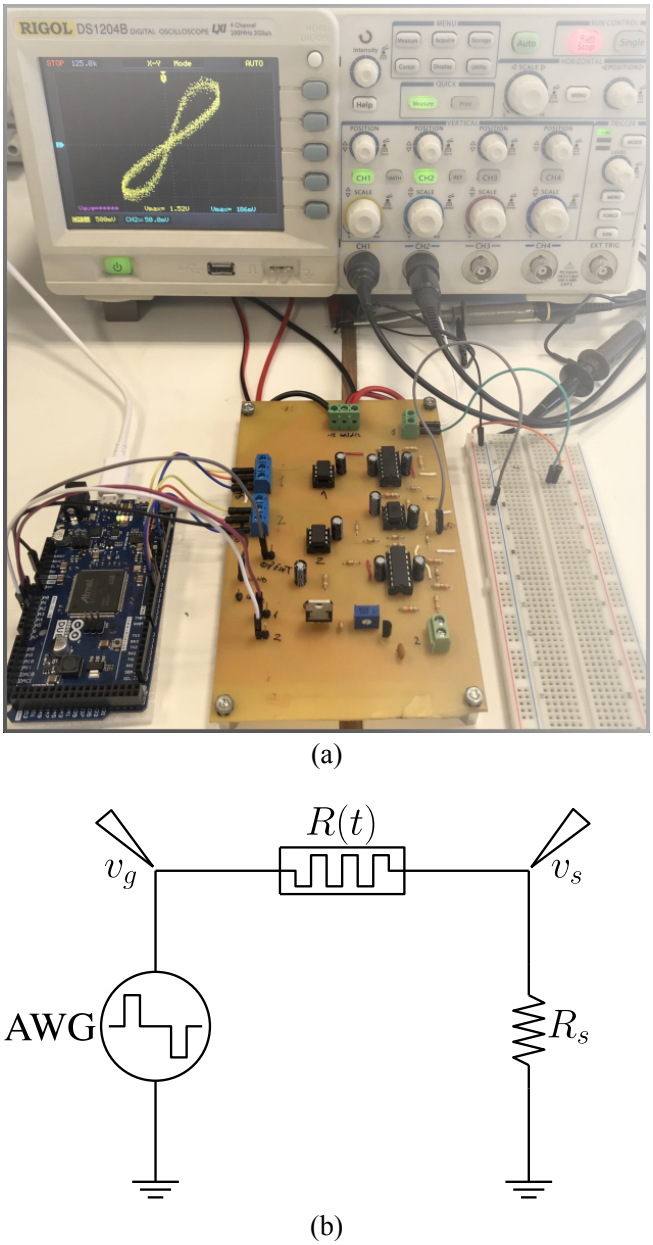

Fig. 2. Experimental setup. (a) On top: an oscilloscope showing a typical hysteresis cycle of the model by Strukov *et al.* [27]. The Arduino microcontroller (bottom left) is connected to a printed circuit board with the digital potentiometer and signal adaptation circuitry (bottom center). A sensing resistor is located on a protoboard (bottom right). (b) Circuit schematic. An arbitrary waveform generator (AWG) feeds the circuit under test. A series resistor $R_s = 1$ kΩ is place in order to measure the current flowing through the variable resistance $R(t)$ given by the memristor.

In this work, we propose a pulsing experiment which resembles the STDP process in a context where a memristive element plays the role of the synaptic junction. For this, we consider the circuit schematized in Fig. 2(b) and apply a stimulus signal with a 500 ms period. This signal consists of

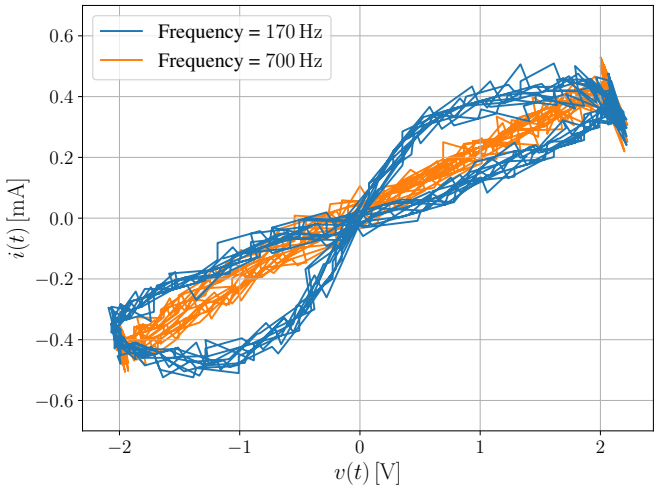

Fig. 3. Typical histeresis cycles for the model by Strukov *et al.* [27]. The input signal is a sinusoid. As the signal frequency is increased, the hysteresis disappears. Parameters were set to $A = 2.5$ V, $R_{\text{on}} = 35\ \Omega$, $R_{\text{off}} = 9500\ \Omega$, and $\mu = 10^4\ \text{V}^{-1}\ \text{s}^{-1}$.

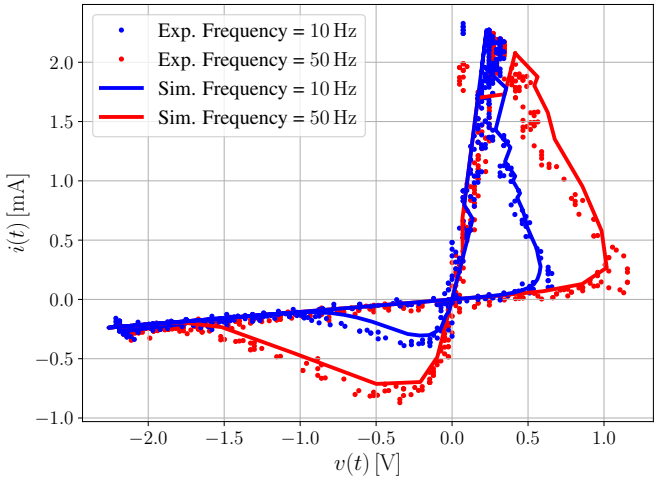

Fig. 4. Emulated (dots) and simulated (solid line) hysteresis cycles for the model described in Patterson *et al.* [25]. The input signal is a sinusoid. As the signal frequency is increased, the hysteresis disappears. Parameters were set to $A = 2.5$ V, $R_{\text{on}} = 35\ \Omega$, $R_{\text{off}} = 9500\ \Omega$, $\alpha = 15\ \text{V}^{-1}$, $\delta = 0.2$ V, $v_0 = 0.3$ V, and $\tau_0 = 0.01$ s.

two stimulus pulses (one positive and one negative) and two measurement pulses (one before the first stimulus pulse and one after the second stimulus pulse). The width of all pulses is 50 ms. Figure 5 shows the case of two overlapping pulses. The overlap is $\Delta t = 25$ ms. Since the net effect of a the pre and postsynaptic stimuli is equivalent to a single driving signal equal to the difference of pre and postsynaptic potentials, the resulting excitation is split in three different pieces: i) a first piece corresponding to the non-overlapping part of the pre-synaptic pulse (a positive 25 ms-wide pulse in Fig. 5); ii) a second piece corresponding to the overlap between pre and postsynaptic stimuli (a 25 ms-wide 'silence'); iii) a third piece corresponding to the non-overlapping part of the post-synaptic

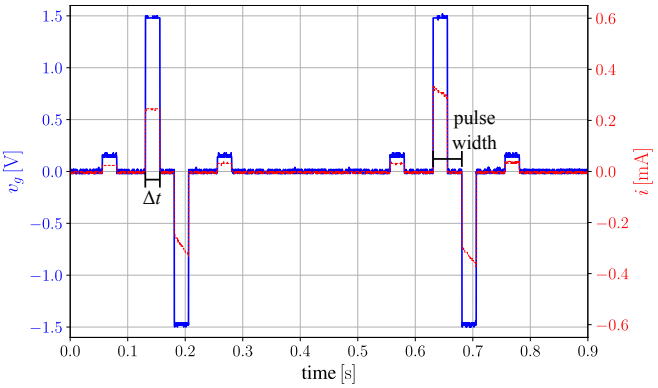

Fig. 5. Two periods of the driving signal v_g and the resulting current through the digital potentiometer i. Each period consists of two stimulus pulses of absolute amplitude 1.5 V and two measurement pulses of amplitude 200 mV. In this case, both pulses are 50 ms-wide and $\Delta t = 25$ ms, leading two a 25 ms overlap between the pre and the postsynaptic stimuli. Parameters were set to $R_{\text{on}} = 1000\ \Omega$, $R_{\text{off}} = 5000\ \Omega$, $\alpha = 30\ \text{V}^{-1}$, $\delta = 0.75$ V, $v_0 = 0.2$ V, and $\tau_0 = 10$ s.

pulse (a negative 25 ms-wide pulse). The absolute amplitude of these pulses is 1.5 V. The measurement pulses are 200 mV high and are situated 50 ms before and after the stimulus pulses. Figure 5 also shows the corresponding current. It can be seen that, during the stimulus pulses, the current varies according to the change of the resistance of the emulator.

To study the influence of Δt on the change of the resistance, we applied a number of consecutive periods of the stimulus signal for a given Δt time. Figure 6 shows the evolution of resistance during the application of 8 consecutive periods for two different Δt and two different resistance initial conditions. As it can be seen, the final value of the resistance depends on the value of Δt but not on the initial condition. We perform an exhaustive characterization applying 20 consecutive periods of the stimulus signal varying Δt. Results are presented in Fig. 7 where the final resistance state is shown as a function of Δt for three different values of τ_0. Data reveal two distinct behaviors depending on whether the value of Δt is smaller or greater than 50 ms, that is, the pulsewidth. Note that there is a destructive interference between the positive and negative stimuli, that modifies the applied signal profile, when $|\Delta t| < 50$ ms. In this case, $|\Delta t|$ has a notorious influence on the memristor final resistance. Once the value of 50 ms is exceeded, the resistance is no longer influenced by $|\Delta t|$ and its value depends only on the parameter τ_0.

Finally, Fig, 8 shows the percentage of change of the resistance, with respect to its final value, as a function of Δt. Remarkably, the relationship between the change of the resistance and Δt is reproduces the behavior experimentally observed in synapses (see, *e.g.*, Ref. [18]) indicating that this particular type of memristor model can be used to implement learning rules based on a STDP process. As expected, the parameter τ_0 affects the resistance change ratio. The larger τ_0, the lower the change of resistance.

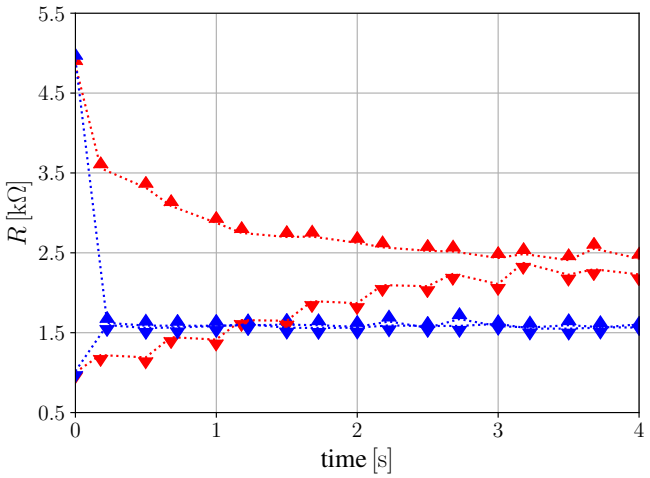

Fig. 6. Evolution of the emulator resistance as a function of time. Upside and downside triangles stand for high ($= R_{\text{off}}$) and low ($= R_{\text{on}}$) initial conditions, respectively. Red and blue colors stand for $\Delta t = 5$, 50 ms, respectively. Parameters were set as those in Fig. 5 except that $\tau_0 = 5$ s.

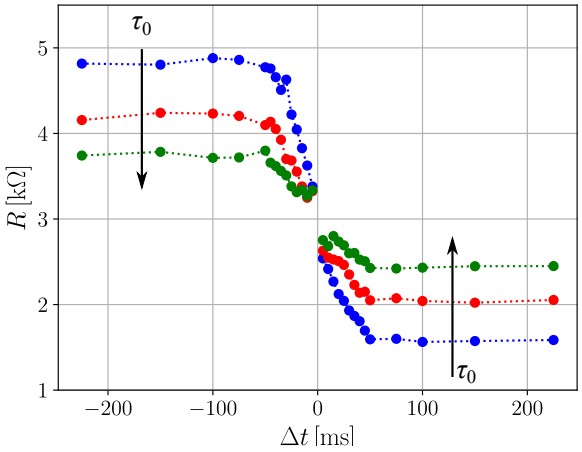

Fig. 7. Resistance of the emulator as a function of Δt for $\tau_0 = 5$ (blue), 10 (red) and 20 s (green). Initially, the resistance was set to 5000 Ω in all cases. The remaining parameters were set as those in Fig. 5.

IV. Conclusions

In this work, we presented a very simple synaptic-like circuit, not as the final word on the subject, but as a first step in a research path we propose, that is, the investigation of memristive devices as building blocks for neuromorphic systems based on emulation architectures. In particular, changes in the memristor resistance when certain spiking patterns are applied to each of its terminals appear to exhibit the same behavior as the synaptic strength under a similar excitation from pre and postsynaptic neurons. One difficulty in experimentally studying this type of operation is that memristor samples are difficult to obtain and tweaking their parameters to adapt their behavior requires a long fabrication and testing process. One alternative is to resort to numerical simulations. In spite of

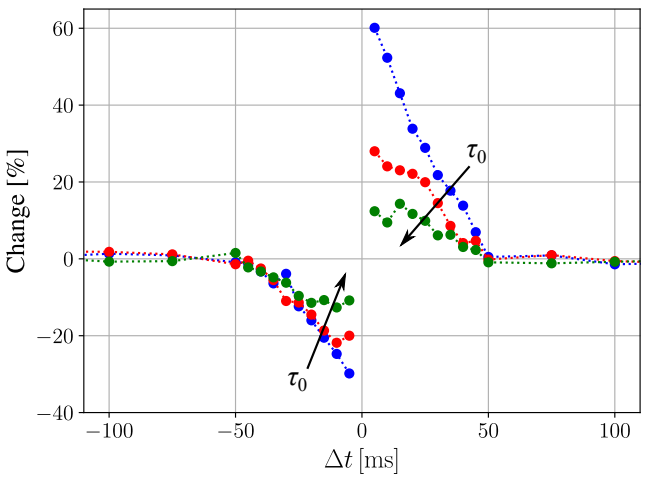

Fig. 8. Percentage of change of the resistance as a function of Δt for $\tau_0 = 5$ (blue), 10 (red) and 20 s (green).

the versatility of this approach, it inhibits the possibility to interact with real-world circuitry. Emulation, on the contrary, allows both to study the influence of different parameters and to investigate some of the problems found when memristors are connected to other circuit elements.

In this paper, we explored one possible emulator design. Following the work of other researchers, we developed an emulator consisting of a microcontroller and a digital potentiometer. Based on the current that traverses the potentiometer, the microcontroller changes its resistance value according to the equations describing a particular memristor model. Moreover, the same computing device may control several potentiometers emulating several memristors. We verified the feasibility of this approach in two different ways. First, we verified that our design can emulate two different types of memristors correctly. Then, we also showed that the emulator can be used to study how the memristor mimics certain characteristics of a synapse, in particular the spike-timing-dependent plasticity.

We also found some limitations to the proposed emulator architecture. The computation speed of the microcontroller puts a cap on the highest frequency of the signals that can be considered. The time used to change the resistance of digital potentiometer may also limit the frequency of acceptable input signals. Another limiting factor is the resolution of the digital potentiometer. Indeed, its resistance cannot change continuously and small changes predicted by a model are not reflected in the actual emulator behavior. However, all these limitations can be alleviated by changes in the hardware, *e.g.*, a higher speed microcontroller and a higher resolution potentiometer.

All in all, the emulator design used in this paper enables the study of a synaptic-like behavior. The following step into this direction is to use the same type of emulator to analyze more complex neuromorphic circuits, consisting of several memristors. In this regard, we do not envision emulated devices as actual components of large neuromorphic circuits, but they

may serve to explore which are the desirable characteristics of some of the constituent parts of those circuits. The use of memristors to mimic synapses and the basic architectural design of the emulator can be found in the literature and the work in this paper can be considered as our first step in a research path which combines both concepts. The same idea has its precedents in the literature, *e.g.*, in the work by Pershin and Di Ventra [28].

REFERENCES

[1] Y. V. Pershin, S. La Fontaine, and M. Di Ventra, "Memristive model of amoeba learning," *Phys. Rev. E*, vol. 80, p. 021926, Aug 2009.

[2] Y. V. Pershin and M. D. Ventra, "Memcomputing: A computing paradigm to store and process information on the same physical platform," in *2014 International Workshop on Computational Electronics (IWCE)*, June 2014, pp. 1–2.

[3] Y. V. Pershin, L. K. Castelano, F. Hartmann, V. Lopez-Richard, and M. D. Ventra, "A memristive pascaline," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 6, pp. 558–562, June 2016.

[4] D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi, and C. S. Hwang, "Memristors for energy-efficient new computing paradigms," *Advanced Electronic Materials*, vol. 2, no. 9, p. 1600090, 2016.

[5] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, "A Survey of Neuromorphic Computing and Neural Networks in Hardware," 2017.

[6] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[7] D. Querlioz, O. Bichler, and C. Gamrat, "Simulation of a memristor-based spiking neural network immune to device variations," in *The 2011 International Joint Conference on Neural Networks*, July 2011, pp. 1775–1781.

[8] B. Linares-Barranco, T. Serrano-Gotarredona, L. A. Camuñas-Mesa, J. A. Perez-Carrasco, C. Zamarreño-Ramos, and T. Masquelier, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Frontiers in neuroscience*, vol. 5, p. 26, 2011.

[9] S. Hu, H. Wu, Y. Liu, T. Chen, Z. Liu, Q. Yu, Y. Yin, and S. Hosaka, "Design of an electronic synapse with spike time dependent plasticity based on resistive memory device," *Journal of Applied Physics*, vol. 113, no. 11, p. 114502, 2013.

[10] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, "STDP and STDP variations with memristors for spiking neuromorphic learning systems," *Frontiers in neuroscience*, vol. 7, p. 2, 2013.

[11] J. Bill and R. Legenstein, "A compound memristive synapse model for statistical learning through stdp in spiking neural networks," *Frontiers in Neuroscience*, vol. 8, p. 412, 2014.

[12] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, pp. 61 – 64, 2015.

[13] S. Saïghi, C. G. Mayr, T. Serrano-Gotarredona, H. Schmidt, G. Lecerf, J. Tomas, J. Grollier, S. Boyn, A. F. Vincent, D. Querlioz, S. La Barbera, F. Alibart, D. Vuillaume, O. Bichler, C. Gamrat, and B. Linares-Barranco, "Plasticity in memristive devices for spiking neural networks," *Frontiers in Neuroscience*, vol. 9, p. 51, 2015.

[14] E. Covi, S. Brivio, A. Serb, T. Prodromakis, M. Fanciulli, and S. Spiga, "Analog memristive synapse in spiking networks implementing unsupervised learning," *Frontiers in Neuroscience*, vol. 10, p. 482, 2016.

[15] J.-X. Shen, D.-S. Shang, Y.-S. Chai, S.-G. Wang, B.-G. Shen, and Y. Sun, "Mimicking Synaptic Plasticity and Neural Network Using Memtranstors," *Advanced Materials*, vol. 30, no. 12, p. 1706717, 2018.

[16] Y. Kim, Y. J. Kwon, D. E. Kwon, K. J. Yoon, J. H. Yoon, S. Yoo, H. J. Kim, T. H. Park, J.-W. Han, K. M. Kim, and C. S. Hwang, "Nociceptive memristor," *Advanced Materials*, vol. 30, no. 8, p. 1704320, 2018.

[17] J. H. Yoon, Z. Wang, K. M. Kim, H. Wu, V. Ravichandran, Q. Xia, C. Hwang, and J. J. Yang, "An artificial nociceptor based on a diffusive memristor," *Nature Communications*, vol. 9, p. 417, 2018.

[18] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *Journal of Neuroscience*, vol. 18, no. 24, pp. 10 464–10 472, 1998.

[19] G.-q. Bi and M.-m. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 139–166, 2001, pMID: 11283308.

[20] Y. Dan and M.-m. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron*, vol. 44, no. 1, pp. 23 – 30, 2004.

[21] Z. Wang, S. Joshi, S. E. Savel'ev, H. Jiang, R. Midya, P. Lin, M. Hu, N. Ge, J. P. Strachan, Z. Li, Q. Wu, M. Barnell, G.-L. Li, H. L. Xin, R. S. Williams, Q. Xia, and J. J. Yang, "Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing," *Nature materials*, vol. 16, no. 1, pp. 101 – 108, 2017.

[22] J. S. Najem, G. J. Taylor, R. J. Weiss, M. S. Hasan, G. Rose, C. D. Schuman, A. Belianinov, C. P. Collier, and S. A. Sarles, "Memristive Ion Channel-Doped Biomembranes as Synaptic Mimics," *ACS nano*, vol. 12, no. 5, pp. 4702 – 4711, 2018.

[23] P. Lorenzi, R. Rao, F. Irrera, J. Su, and E. Miranda, "A thorough investigation of the progressive reset dynamics in HfO2-based resistive switching structures," *Applied Physics Letters*, vol. 107, no. 11, p. 113507, 2015.

[24] E. Miranda, B. Hudec, J. Su, and K. Frhlich, "Model for the CurrentVoltage Characteristic of Resistive Switches Based on Recursive Hysteretic Operators," *IEEE Electron Device Letters*, vol. 36, no. 9, pp. 944–946, Sept 2015.

[25] G. A. Patterson, J. Suñé, and E. Miranda, "Voltage-Driven Hysteresis Model for Resistive Switching: SPICE Modeling and Circuit Applications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 12, pp. 2044–2051, 2017.

[26] L. Xia, B. Li, T. Tang, P. Gu, P. Chen, S. Yu, Y. Cao, Y. Wang, Y. Xie, and H. Yang, "Mnsim: Simulation platform for memristor-based neuromorphic computing system," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 5, pp. 1009–1022, May 2018.

[27] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[28] Y. V. Pershin and M. D. Ventra, "Experimental demonstration of associative memory with memristive neural networks," *Neural Networks*, vol. 23, no. 7, pp. 881 – 886, 2010.

[29] H. Kim, M. P. Sah, C. Yang, S. Cho, and L. O. Chua, "Memristor emulator for memristor circuit applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 10, pp. 2422–2431, Oct 2012.

[30] A. Ascoli, F. Corinto, and R. Tetzlaff, "A class of versatile circuits, made up of standard electrical components, are memristors," *International Journal of Circuit Theory and Applications*, vol. 44, no. 1, pp. 127–146.

[31] A. Yesil, "A new grounded memristor emulator based on mosfet-c," *AEU - International Journal of Electronics and Communications*, vol. 91, pp. 143 – 149, 2018.

[32] D.-S. Yu, T.-T. Sun, C.-Y. Zheng, H. H. C. Iu, and T. Fernando, "A simpler memristor emulator based on varactor diode," *Chinese Physics Letters*, vol. 35, no. 5, p. 058401, 2018.

[33] O. A. Olumodeji and M. Gottardi, "Arduino-controlled HP memristor emulator for memristor circuit applications," *Integration*, vol. 58, pp. 438 – 445, 2017.

[34] M. A. Ermini, J. Dhanasekar, and V. Sudha, "Memristor emulator using mcp3208 and digital potentiometer," *ICTACT Journal on Microelectronics*, vol. 3, no. 4, pp. 473 – 476, 2018.

[35] L. Chua, "Memristor-The missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, September 1971.

[36] J. Blasco, N. Ghenzi, J. Su, P. Levy, and E. Miranda, "Modeling of the Hysteretic $I-V$ Characteristics of TiO_2-Based Resistive Switches Using the Generalized Diode Equation," *IEEE Electron Device Letters*, vol. 35, no. 3, pp. 390–392, March 2014.

[37] E. Miranda, "Compact Model for the Major and Minor HysereticIVLoops in Nonlinear Memristive Devices," *IEEE Transactions on Nanotechnology*, vol. 14, no. 5, pp. 787–789, Sept 2015.

[38] Atmel, "SMART ARM-based MCU SAM3X / SAM3A Series," March 2015.

[39] Renesas, "X9C102, X9C103, X9C104, X9C503. Digitally Controlled Potentiometer (XDCP)," July 2009.

[40] T. Fager, "Arduino library for managing digital potentiometers X9Cxxx," July 2011.
[41] D. O. Hebb, *The organization of behavior: A neurophysiological approach*. Wiley, 1949.