A differential memristive synapse circuit for on-line learning in neuromorphic computing systems

Nair, Manu V ; Muller, Lorenz K ; Indiveri, Giacomo

DOI: https://doi.org/10.1088/2399-1984/aa954a

Nano Futures

PAPER

A differential memristive synapse circuit for on-line learning in neuromorphic computing systems

To cite this article: Manu V Nair *et al* 2017 *Nano Futures* **1** 035003

View the article online for updates and enhancements.

NANO
FUTURES

RECEIVED
15 June 2017

REVISED
19 October 2017

ACCEPTED FOR PUBLICATION
23 October 2017

PUBLISHED
17 November 2017

PAPER

A differential memristive synapse circuit for on-line learning in neuromorphic computing systems

Manu V Nair , Lorenz K Muller and Giacomo Indiveri[1]

Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland

[1] Author to whom any correspondence should be addressed.

E-mail: mnair@ini.uzh.ch, lorenz@ini.uzh.ch and giacomo@ini.uzh.ch

Keywords: differential, memristive devices, ReRAM, synapse, on-line learning, neuromorphic, memristor

Supplementary material for this article is available online

Abstract

Spike-based learning with memristive devices in neuromorphic computing architectures typically uses learning circuits that require overlapping pulses from pre- and post-synaptic nodes. This imposes severe constraints on the length of the pulses transmitted in the network, and on the network's throughput. Furthermore, most of these circuits do not decouple the currents flowing through memristive devices from the one stimulating the target neuron. This can be a problem when using devices with high conductance values, because of the resulting large currents. In this paper, we propose a novel circuit that decouples the current produced by the memristive device from the one used to stimulate the post-synaptic neuron, by using a novel differential scheme based on the Gilbert normalizer circuit. We show how this circuit is useful for reducing the effect of variability in the memristive devices, and how it is ideally suited for spike-based learning mechanisms that do not require overlapping pre- and post-synaptic pulses. We demonstrate the features of the proposed synapse circuit with SPICE simulations, and validate its learning properties with high-level behavioral network simulations which use a stochastic gradient descent learning rule in two benchmark classification tasks.

1. Introduction

Neuromorphic computing systems typically comprise neuron and synapse circuits arranged in a massively parallel manner to support the emulation of large-scale spiking neural networks [1–8]. In these systems, the bulk of the silicon real-estate is taken up by synaptic circuits, where the memory and computational sites are co-localized [5]. Consequently, to save area and maximize density, many neuromorphic computing approaches avoid implementing complex synaptic circuits with on-chip learning mechanisms [4, 9, 10], and resort to training the network on external computers. However, these approaches lose the ability to execute on-line 'life-long' learning and require that the network parameters (such as the synaptic weights) be programmed at deployment time. In addition, if these parameters are stored in static random access memory cells or in dynamic random access memory banks, they need to be re-programmed every time the system is reset. For large networks [4, 11, 12], the time required to initialize the system with these parameters can become prohibitive.

Memristive devices can potentially address these problems by virtue of their compactness and non-volatility [13]. Given their advantages, several neuromorphic arrays that use memristive devices have been proposed [14–18]. Typically, these approaches propose to use memristive devices in dense synaptic arrays for implementing large-scale neural networks. For instance, [15, 19] describe use of 1R arrays to implement perceptrons trained by supervised learning protocols such as [20]. Similarly, in [21], the authors train a 1T-1R array to implement perceptrons classifying face images from the Yale face database [22]. In [23], the authors use the Recursive Least-Squares algorithm for training synaptic weights to perform complex tasks such as human motor control. In works such as [15, 23], the authors propose the use of two devices per synaptic element to

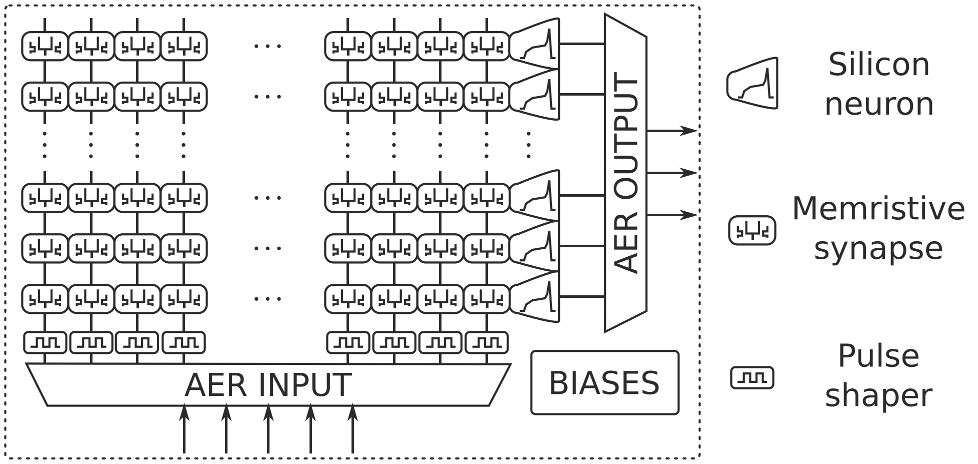

Figure 1. Multi-neuron chip architecture: input AER events are converted into read and write pulses sent to multiple memristive synapses; the currents produced by the synaptic circuits are integrated by the neuron assigned for the row; ouput spikes are converted into AER events and transmitted off-chip.

implement positive and negative weight terms. Other approaches describe synaptic arrays with a 1T-1R synapse elements that learn using classical [16, 18, 24–26] or stochastic [17, 27, 28] spike-timing dependent plasticity (STDP) learning rules. In these arrays the currents used to program the memristive devices can be very large, especially for devices in high-conductance states. This imposes severe restrictions on the power budget, capacitor sizes, and other aspects for the design of ultra-low power memristive-neuromorphic circuits. Moreover, the learning protocols employed in most of these architectures couple the length of the pulses used to transmit signals across the layers of the network with the duration of the pulses required to program the devices [18, 24]. This requirement imposes severe constraints on the maximum data throughput of the network, because each row or column in the cross-bar array has to wait for the pre- and post-synaptic pulses to finish, before a new one can be sent. In this paper, we propose a novel synaptic circuit thataddresses at the same time both the large current and overlapping pulses problems. To overcome the problem of integrating large currents in the post-synaptic neuron, we propose a novel differential-mode sub-threshold memristive synapse circuit that decouples, normalizes, and re-scales the memristive device current from the one supplied to the post-synaptic neuron. To overcome the problem of overlapping pulses in cross-bar architectures, we propose an event-based scheme that decouples the duration of the input spikes from the read and update phases of the target synapse, coupled with the use of a novel spike-based synaptic update mechanism.

In recent years, several algorithms employing spike-triggered learning based on post-synaptic neuronal activity, instead of vanilla STDP mechanisms, have been proposed in computational neuroscience literature [29–31]. Several neuromorphic implementations of these mechanisms have also been realized [6–8, 32, 33]. In this paper, we demonstrate how the proposed differential memristive synapse circuit can be incorporated in a neuromorphic system that employs a learning circuit based on such ideas. This circuit is inspired by the biologically plausible learning rule presented in [31] and gradient-descent based methods applied to memristive devices [34, 35]. We use these learning circuits to implement a randomized unregulated step descent algorithm, which has been shown to be effective for training synaptic elements with limited precision [36].

In the following section, we present the network architecture that is compatible with the proposed differential memristive synapse circuit. In section 3, we describe the techniques used for sensing and changing the memristive device conductances, and present circuit simulation results that quantify its performance figures. In section 4, we assess the features of neuromorphic architectures that make use of the proposed circuits and validate them with behavioral simulations in two classification tasks. Finally, sections 5 and 6 contain the discussion and concluding remarks.

2. Neuromorphic architectures for memristive synapses

The architecture we propose is composed of an array of synapses and neurons that receive input spikes into columns of synaptic cells, and produce output spikes from the silicon neurons arranged in rows (see figure 1). This type of architecture can be integrated within a full-custom neuromorphic very large scale integration chip, or be used as a single-core in multi-core neuromorphic systems [4, 9]. Both the input and output spikes are represented by fast digital pulses that are encoded using the address-event representation (AER) [37–40].

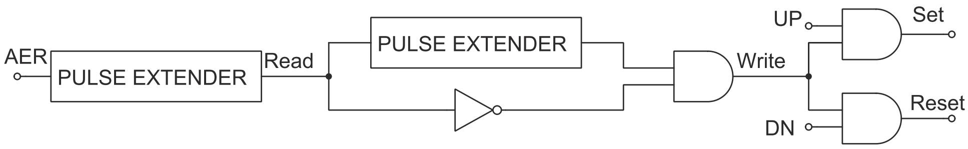

Figure 2. Pulse-shaping circuit for creating a sequence of read and write pulses, with each AER input event.

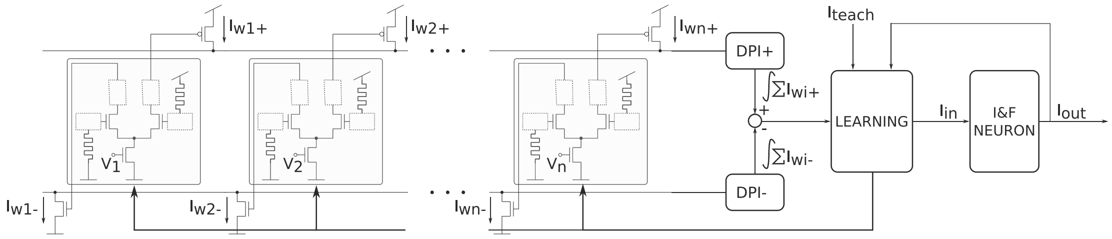

Figure 3. Memristive synapse circuits for on-line learning in a neuromorphic architecture.

On the input side, asynchronous AER circuits ensure that events are transmitted as they arrive. Upon the arrival of a pre-synaptic address-event, a pulse-shaping circuit decouples the duration of the input spikes from the read and update phases of the target synapse. This frees the communication bus to transmit spikes from sender nodes to the cross-bar array, increasing the throughput of the network by use of shared or time-multiplexed communication resources. The block diagram describing the operation of the pulse-shaping circuits is shown in figure 2. The pulse-shaping circuit consists of two pulse-extender circuits [7] and is configured to produce two pulses in quick succession on the arrival of an AER event. These pulses sequentially enable the read-mode operation, where the state of the addressed synapse is sensed, followed by the write-mode operation, where the conductance of the memristive devices are increased or decreased, in the target synapse. The change in the synapse state is directed by two control signals produced by the learning circuits in the post-synaptic neuron labeled as UP and DN, which are used for increasing and decreasing the synaptic weights, respectively.

On the output side, a 1-D arbiter circuit queues output events in case of collisions and transmits them on the shared output bus [38]. A programmable bias-generator circuit [41] provides a set of globally-shared temperature-compensated current signals for biasing the analog parameters of the neuromorphic circuits, such as time-constants, refractory periods, or learning rates.

Address-events target destination columns of the memristive array. By construction, all rows of the stimulated column will process the input event in parallel. Furthermore, the extended read and write pulses typically last longer than the input AER event duration. Therefore, sequential AER stimulation of multiple columns will produce multiple read/write operations across the full array that will overlap in time and operate in parallel. A block diagram of the circuits present in a single row of the cross-bar architecture is illustrated in figure 1 is shown in figure 3. It comprises multiple synaptic circuits that receive voltage pulses from the pulse-shaping circuits, two current-mode Differential Pair Integrator (DPI) circuits that emulate excitatory and inhibitory synapse dynamics with biologically realistic time constants [42, 43], a current-mode learning block that implements a spike-driven learning mechanism [6, 7], and an ultra-low-power adaptive Integrate-and-Fire neuron circuit that faithfully reproduces biologically realistic neural dynamics [44, 45]. In the read-phase, the synaptic circuit senses the state of the two memristive devices in it, and produces rescaled and normalized differential currents that are driven into the positive and negative DPI inputs. The DPI circuits integrate these weighted currents and generate a rescaled output current that is driven into a neuron and its learning block. The learning block uses a copy of this 'dendritic' current to compare it to the net input current, which includes contributions from the neuron and an external source. The external source could represent a teacher signal in supervised learning protocols, or contributions from other neurons in unsupervised learning protocols. Based on this comparison, the learning block evaluates an error signal and produces the UP and DN weight update control signals that are used during the write-mode phase to increase or decrease the weights of the stimulated target synapse. We demonstrate the operation of this architecture with a concrete example in section 4.

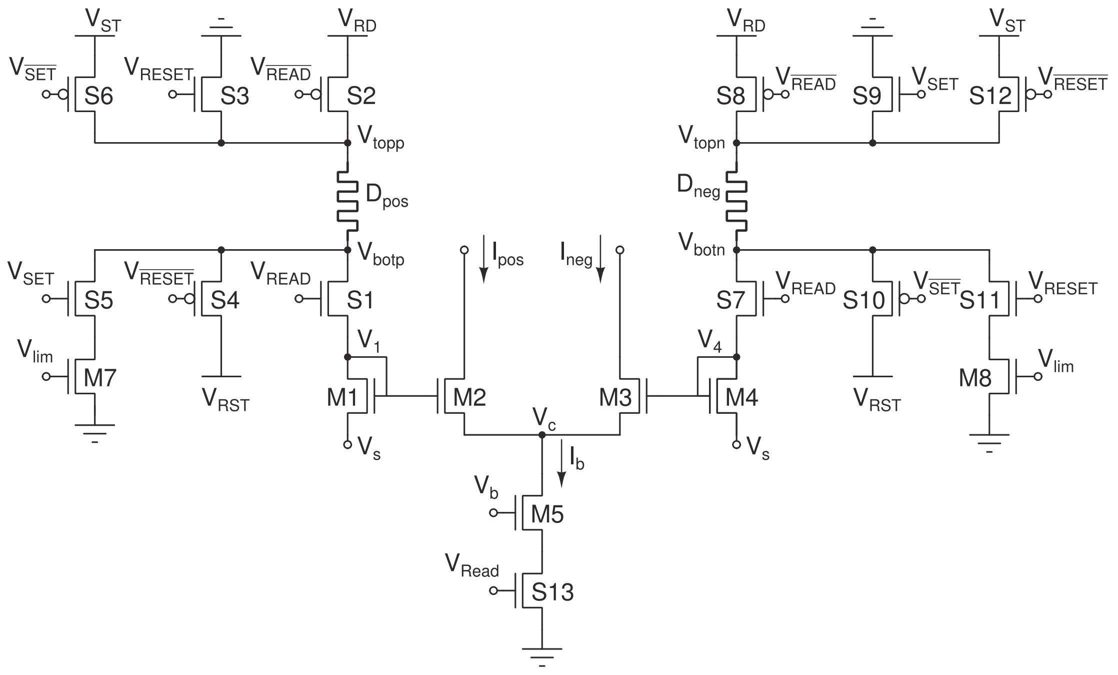

Figure 4. Differential memristive synaptic circuit.

3. The differential memristive synapse circuit

The differential memristive synapse circuit is based on the classic Gilbert-normalizer element [46, 47], originally designed for bipolar transistors, but functional also for Metal Oxide Semiconductor Field-Effect Transistors (MOSFETs) operated in the sub-threshold domain [47]. The synapse circuit stores its weight as the difference between the conductances of two memristive devices, one representing a positive term and the other representing a negative term. Programming the devices is done in a push-pull manner: to increase the synaptic weight, the conductance of the positive term is increased, and that of the negative term is decreased. The complementary operation is achieved by simultaneously decreasing and increasing the conductances of the positive and negative terms, respectively. The output current produced by this circuit, in read-mode, can be scaled to very low values (e.g., in the range of pico Amperes). This reduces the total current driven into the post-synaptic neuron, which can then be implemented with very small capacitors and ultra-low power sub-threshold circuits. The differential operation coupled with the normalizing ability of the circuit has two additional advantages. It reduces the effect of memristive device variability and implements both positive (excitatory) and negative (inhibitory) synapse contributions, effectively doubling the 'high–low' dynamic range of the synaptic weight. In write-mode, the circuit enables programming the memristive devices with programmable current limits, pulse widths, and heights. The write-mode parameters can be chosen by the user to optimize the write-mode power consumption depending on the memristive device integrated in the circuit.

The operating principles of the circuit are independent of the memristive device technology used. It works with many different resistive memory technologies and an arbitrary number of resistive stable states. In this work, we assume our Complementary Metal-Oxide-Semiconductor (CMOS) circuits can be directly interfaced to HfO_2 based devices through post-processing methods, as described in [48].

The schematic diagram of the differential memristive synapse circuit is shown in figure 4. The circuit is used in a 'read-mode' phase to measure the conductance of the two memristive devices to produce scaled output currents that are conveyed to the downstream current-mode neural processing circuits. It is then operated in a 'write-mode' to update the state of the memristive devices using the downstream learning circuit control signals. All S_i MOSFETs represent switches, with gates controlled by digital signals. Signals with an overline, such as $\overline{X}$, represent the inverted version of the signal X. The signal V_{Read} is the voltage used during the read-mode, while V_{Set} and V_{Reset} are the set and reset voltages used in the write-mode to increase/decrease the synaptic weight. The signal V_{b} is a sub-threshold bias voltage that sets the (sub-threshold) scale of the output currents. The MOSFETs Sx have dimensions $W/L = 5\ \mu\mathrm{m}/0.5\ \mu\mathrm{m}$; MOSFETs M1 & M4 have $W/L = 1\ \mu m/2\ \ \mu m$, M2 & M3 have $W/L = 0.5\ \mu\mathrm{m}/1\ \mu\mathrm{m}$, and M5 $W/L = 2\ \mu\mathrm{m}/1\ \mu\mathrm{m}$.

3.1. Read-mode operation

To operate the differential synapse circuit shown in figure 4 in read-mode, the switches S1, S2, S7, and S8 are turned on and all other switches are turned off; the digital control signals V_{set} and V_{reset} are set to logical zero. The current-mode normalizer circuit is implemented by MOSFETs M1-M6. In this mode of operation, the memristive devices D_{pos} and D_{neg} are connected to the corresponding V_{top} and V_{bot} nodes. When the V_{read} pulse is active, the currents flowing through the memristive devices are measured and the output currents, I_{pos} and I_{neg}, are sent to the excitatory and inhibitory DPI circuits, respectively.

Therefore, in this mode of operation, during a V_{read} pulse:

$$\begin{aligned} I_{D_{\text{pos}}} &= I_{M1} \quad \text{and} \\ I_{D_{\text{neg}}} &= I_{M4}, \end{aligned} \tag{1}$$

where I_{D_x} is the current through the device D_x, and I_{M_i} is the current through the MOSFET M_i.

For low-power operation, it is desirable to make I_{D_x} very small. Under this condition, we can assume that the transistors operate in sub-threshold domain. This allows us to analytically derive the relationship between the circuit parameters and the current flowing through the circuit's output branches. By writing the sub-threshold equations for a MOSFET and equating it to the currents through the resistive devices, we get:

$$\begin{aligned} (V_{\text{RD}} - V_i) &= R_x I_x \\ I_{M_i} &= I_0 \mathrm{e}^{\frac{\kappa V_i - V_{\text{s}}}{U_{\text{T}}}} \\ I_x &= I_{M_i}, \end{aligned} \tag{2}$$

where R_x represents the resistance of the memristive device D_x, V_{RD}, the supply voltage provided in 'read-mode', V_{s}, the source voltage of the input MOSFETs $M1$ and $M4$, V_i, the gate voltage of the MOSFET M_i, κ, the sub-threshold slope factor [47], and U_{T}, the thermal voltage. By solving for V_i:

$$I_x = I_0 \mathrm{e}^{\frac{-\kappa R_x I_x}{U_{\text{T}}}} \mathrm{e}^{\frac{\kappa V_{\text{RD}} - V_{\text{s}}}{U_{\text{T}}}}. \tag{3}$$

If $R_x I_x$ is sufficiently small, then

$$I_0 \mathrm{e}^{\frac{-\kappa R_x I_x}{U_{\text{T}}}} \approx I_0 \left(1 - \frac{\kappa}{U_{\text{T}}} R_x I_x\right) \tag{4}$$

so

$$I_x = I_{M_i} = I_0 \frac{1}{\mathrm{e}^{-\frac{\kappa V_{\text{RD}} - V_{\text{s}}}{U_{\text{T}}}} + \frac{\kappa}{U_{\text{T}}} R_x I_0}. \tag{5}$$

Equation (5) describes how the input current changes with the conductance of the memristive device, and with V_{RD} and V_{s} voltages. In particular, for large $V_{\text{RD}} - V_{\text{s}}$ values, the current is approximately linear with respect to the memristive device conductance, but assumes relatively large values (large values make the circuit less power-efficient). For very small $V_{\text{RD}} - V_{\text{s}}$ values, the circuit produces very small currents that change linearly, but with a small dependence on the device memristance R_x. The effect of this trade-off is highlighted in figure 5(a), which plots equation (5) for different values of V_{s}, with V_{RD} set to 1.8 V. Figure 5(b) shows circuit simulations results, carried out using a standard 130 nm CMOS process, that support the theoretical analysis. V_{RD} was set to 1.8 V, while V_{s} was swept to obtain the three different $V_{\text{RD}} - V_{\text{s}}$ values shown in the figure legend. In this mode of operation, the voltage applied across the memristive device is set low enough to prevent conductance changes. This allows us to model the device as a fixed resistor, and to characterize the circuit as a function of all resistance values between the memristive device's low and high resistance states.

The output currents of the differential memristive circuit are directly proportional to the input currents sensed from the corresponding input branch and scaled by the bias current I_{b}. Specifically, if all transistors operate in sub-threshold saturation domain:

$$\begin{aligned} I_{M1} &= I_0 \mathrm{e}^{\frac{\kappa}{U_{\text{T}}} V_1} \mathrm{e}^{-\frac{V_{\text{s}}}{U_{\text{T}}}} & \qquad I_{\text{pos}} &= I_0 \mathrm{e}^{\frac{\kappa}{U_{\text{T}}} V_1} \mathrm{e}^{-\frac{V_c}{U_{\text{T}}}} \\ I_{M4} &= I_0 \mathrm{e}^{\frac{\kappa}{U_{\text{T}}} V_4} \mathrm{e}^{-\frac{V_{\text{s}}}{U_{\text{T}}}} & \qquad I_{\text{neg}} &= I_0 \mathrm{e}^{\frac{\kappa}{U_{\text{T}}} V_4} \mathrm{e}^{-\frac{V_c}{U_{\text{T}}}}. \end{aligned} \tag{6}$$

By solving for $\mathrm{e}^{-\frac{V_c}{U_{\text{T}}}}$ using the extra condition that $I_{\text{b}} = I_{\text{pos}} + I_{\text{neg}}$, and replacing terms in equation (6), we obtain:

$$I_{\text{pos}} = I_{\text{b}} \frac{I_{M1}}{I_{M1} + I_{M4}} \qquad I_{\text{neg}} = I_{\text{b}} \frac{I_{M4}}{I_{M1} + I_{M4}}. \tag{7a}$$

This allows us to produce output currents that are scaled versions of the currents flowing through the memristive devices, and potentially much smaller, thus enabling the design of ultra low-power current-mode memristive sensing architectures. In order to ensure proper operation of the differential memristive output normalizing

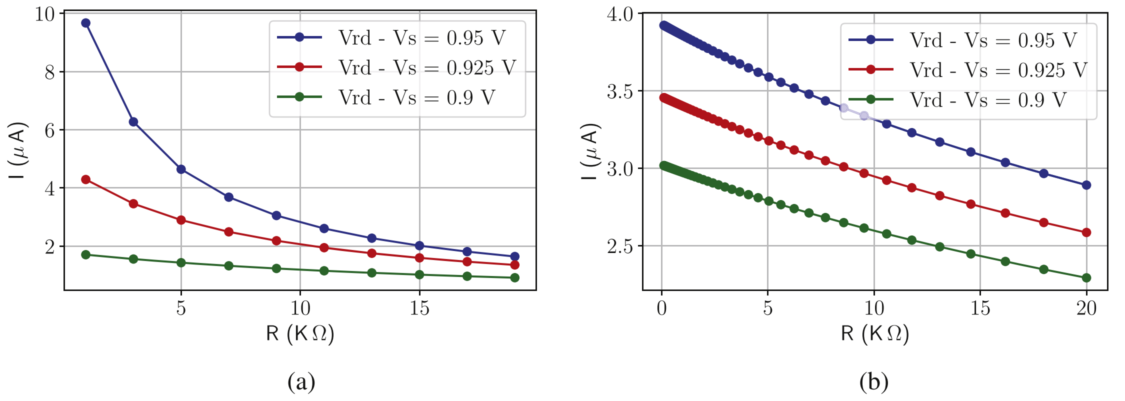

Figure 5. (a) Theoretical solution of the input current as a function of memristive device resistance, for different values of $V_{RD} - V_s$ settings, with V_{RD} set to 1.8 V. (b) SPICE circuit simulations of the same current for a 130 nm CMOS process.

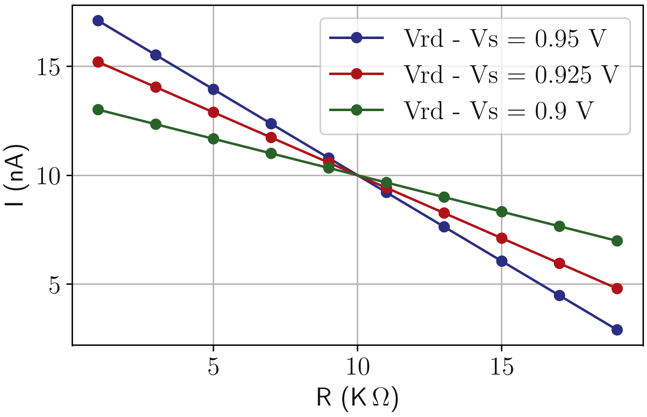

Figure 6. Theoretical normalized output current of the analytic differential memristive circuit as a function of 20 different memristive conductance values, for three different ($V_{RD} - V_s$) settings, with $I_b = 20$ nA, and $V_{rd} = 1.8$ V.

behavior, while minimizing the power dissipated in the input current sensing stage, it is important to have large V_s values, with small $V_{RD} - V_s$ values.

Figure 6 shows the theoretical normalized output current I_{pos}, for a bias current $I_b = 20$ nA, for resistance values of D_{pos} increasing from 1 to 20 KΩ, and of D_{neg} decreasing proportionally from 20 to 1 KΩ. A differential current-mode readout circuit that computes $I_{pos} - I_{neg}$ can double the resolution of the conductance/memory state sensing operation. More realistic circuit simulation results for a 130 nm CMOS processare shown in figure 7. To generate this plot, the memristive devices were modeled as resistors and the resistance of D_{pos} was swept from 100 Ω to 20 KΩ. The bias voltage V_b was set to generate a bias current, I_b, of 20 nA. The simulation results show the output of the circuit for different settings of V_s and as a function of different conductance values assumed for the memristive devices. The blue, red, and green traces are the current outputs when the resistance of D_{neg} was set as 1 KΩ, 10 KΩ, and 19 KΩ respectively. The solid and dashed lines plot I_{pos} and I_{neg} respectively. It can be seen from the plots that the cross over point shifts as the resistance values of D_{neg} change. Note how the linearity of the circuit is improved when $V_{rd} - V_s$ is reduced, at the cost of slightly reduced difference between I_{pos} and I_{neg}.

3.2. Variability reduction

The strategy of using two memristive devices programmed in a complementary fashion and connected to the current-mode normalizer circuit has the added benefit of significantly reducing the impact of memristive device variability in the output currents. To demonstrate this effect, we show in figure 8 the results of Monte Carlo simulations in which we compare the variability of the output currents versus that of the memristive devices. In these simulations we set $I_b = 20$ nA and $V_S = 0.9$ V. On the basis of HfO_2 data available from the literature [49], we used conservative figures for the distributions of the memristive device high/low states and their variance. In particular, we sampled resistance values from a Gaussian distribution with (mean, standard deviation) of (6 kΩ, 1200 kΩ) and (3 kΩ, 600 Ω) in the high and low resistance states, respectively (see samples in figure 8(a)), and

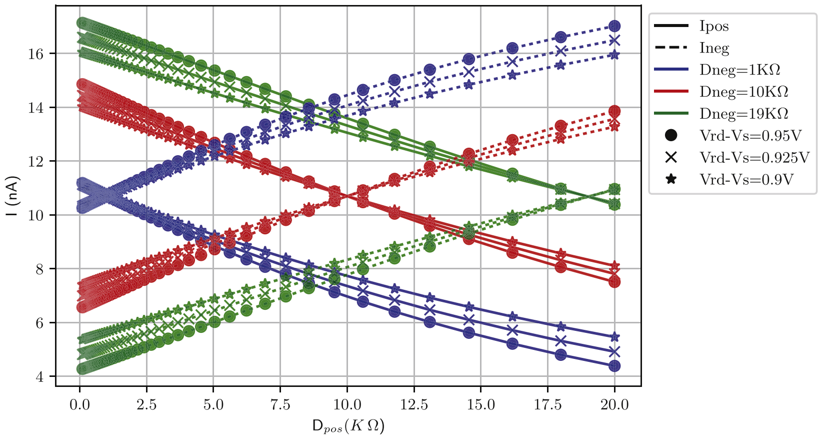

Figure 7. Circuit simulation results for the transfer characteristics of the normalizer circuit. Different colors represent circuit response for different values of the D_{neg} resistor. Solid traces represent the current I_{pos} while dashed ones represent I_{neg}. Different markers denote different $V_{\mathrm{rd}} - V_{\mathrm{s}}$ settings.

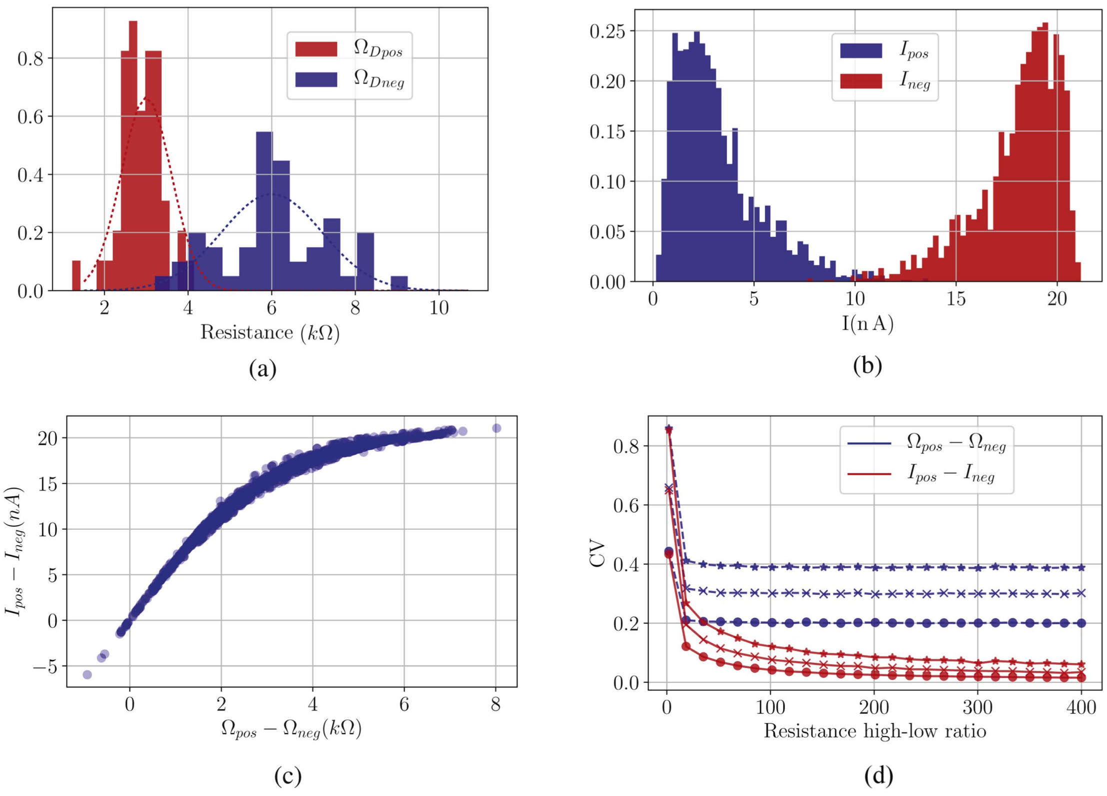

Figure 8. Effect of memristive device variability on synapse circuits outputs. (a) Distribution of device resistance values (Mean, Std. Dev.) for $\Omega_{D_{\mathrm{neg}}} = (2.87\ \mathrm{K}\Omega, 490\ \Omega)$ and $\Omega_{D_{\mathrm{pos}}} = (6.12\ \mathrm{K}\Omega, 1.3\ \mathrm{K}\Omega)$. The dashed lines represent the sampling distributions. (b) Distribution of output currents for the resistance samples derived from (a). (c) Difference of output current samples, versus difference of high–low resistance samples derived from (a). (d) Coefficient of variation of output current difference and of resistance difference as a function of different high–low ratios, and for three different values of resistance CV (0.2, 0.3, and 0.4 for the circles, crosses and stars respectively).

measured the circuit response using such values (see figure 8(b)). We observed that the histogram of the output currents I_{pos} and I_{neg} are symmetric, illustrating the effect of normalization, with a standard deviation of approximately 2.12 nA for both branches. The normalization circuit effectively compresses the error in output current for large difference between resistances and expands it for small differences as shown in figure 8(c). Even for these conservative values of resistance figures, with a very small high–low ratio, the Coefficient of Variation (CV) was reduced from 0.429 for the we show a systematic $\Omega_{\mathrm{pos}} - \Omega_{\mathrm{neg}}$ to 0.284 for $I_{\mathrm{pos}} - I_{\mathrm{neg}}$. For more typical cases, for example with high–low resistance values of 100 Ω and 10 KΩ, the same analysis shows a drastic reduction of CV from 0.219 to 0.003. In figure 8, we show a systematic comparison of the CVs between the basic resistance differences and the output current differences, for increasing ratios of high–low states. The comparison was performed by running Monte Carlo simulations in which the device high and low resistance states were sampled from a normal distribution with three different coefficients of variation (0.2, 0.3, and 0.4), and the output currents were calculated using the circuit's transfer function derived analytically in section 3.

3.3. Write-mode operation

The write-mode operation takes place immediately after the read-mode phase, as determined by the sequence of V_{read} and V_{write} pulses generated by the pulse-shaping circuit of figure 2. In this phase, V_{read} is zero, the V_{write} is high. Furthermore, the switches of two memristive devices (S4–S10) are turned on in a complementary manner, such that the resultant voltage across the memristive devices induces opposite changes in their conductance values. For example, to increase the net output current ($I_{\mathrm{pos}} - I_{\mathrm{neg}}$), the conductance of D_{pos} is increased and that of D_{neg} is decreased. This is done by enabling the switches S5, S6, S9, and S10 by programming the V_{set} signal to logical one, and V_{reset} to logical zero. This connects V_{topp} to V_{ST}, V_{botp} to ground, V_{botn} to V_{RST}, and V_{topn} to ground. Similarly, to decrease ($I_{\mathrm{pos}} - I_{\mathrm{neg}}$), the V_{reset} signal is to set logical one, and V_{set} is set to logical zero. The MOSFETs M7 and M8 are current-limiting transistors that protect the devices from damage during programming. The signal V_{lim} is a bias voltage chosen that ensure that the memristive devices are not damaged during the forming operation. To minimize power consumption all switching transistors are turned on only during a read or write pulse.

The pulse shaping circuit of figure 2 can be tuned to increase or decrease the write pulse duration. Therefore, by programming the length of these pulses and by choosing appropriate values for V_{ST} and V_{RST} voltages, it is possible to use this circuit to produce reliable binary, gradual, or stochastic changes in the memristive devices [13, 49]. The mode of operation of the memristive devices and the nature of the changes that should be induced in the memristive device conductance depend on the specific learning algorithm implemented in the learning block of figure 3.

4. Learning simulations

In this section we demonstrate examples of spike-based learning simulations using a learning rule that is ideally suited for implementation in neuromorphic architectures that comprise the proposed memristive synapses. In the first subsection, we learn a single low-dimensional pattern with varying contrast, in the second subsection we learn many overlapping high-dimensional patterns.

4.1. Single pattern binary classification

Here, we show simulation results of two neurons trained to classify an input spike-train by adjusting their synaptic weights. We study the performance of such a learning system connected by multiple binary synapses and compare it to that of a hypothetical 32-bit floating-point precision synapse in the same setting. This illustrates what the performance limitation is with an 'ideal' synaptic element when classifying a finite-rate Poisson train with a leaky integrator neuron.

In this task, the neurons a and b are connected via randomly initialized synapses to two neural populations p_1 and p_2 of size n_c; n_c is also the number of synaptic connections from p^1 and p^2 to a and b. At any point in time, the two populations fire with two different average neuron firing rates, with Poissonian statistics. The goal is for neuron a to learn to fire more than neuron b, whenever input units from population p_1 fire at a higher rate than input units in p_2. Intially both neurons are equally connected to both populations. To achieve this, we use a supervised training protocol: given the input, we provide a teacher signal to the neuron that should fire more. The teacher signal is represented by a poissonian spike train sent to the target neurons via a separate channel. The spike-based learning algorithm is a discretized version of the one presented in [31]. It performs a gradient descent procedure on the difference of the observed and desired neuron firing rates, and it can be readily implemented in mixed signal CMOS neuromorphic hardware [7, 36]. A detailed description of this learning rule and the parameter values used are provided in the supplementary material available online at stacks.iop.org/NANOF/1/035003/mmedia.

In figure 9, we compare the classifications results produced by a system using 32-bit floating-point precision synapses with the results obtained simulating the proposed binary synapses in the same setting; the colors indicate which of the two learning neurons fired more strongly; the ideal solution is a separation of red and blue at a 45° angle. In both cases the performance increases with the number of synapses. This increase for binary synapses is due to the fact that additional synapses allow the network to resolve a larger number of effective connection strengths. In both cases, higher n_c also implies a higher input firing rate, which gives the neurons a

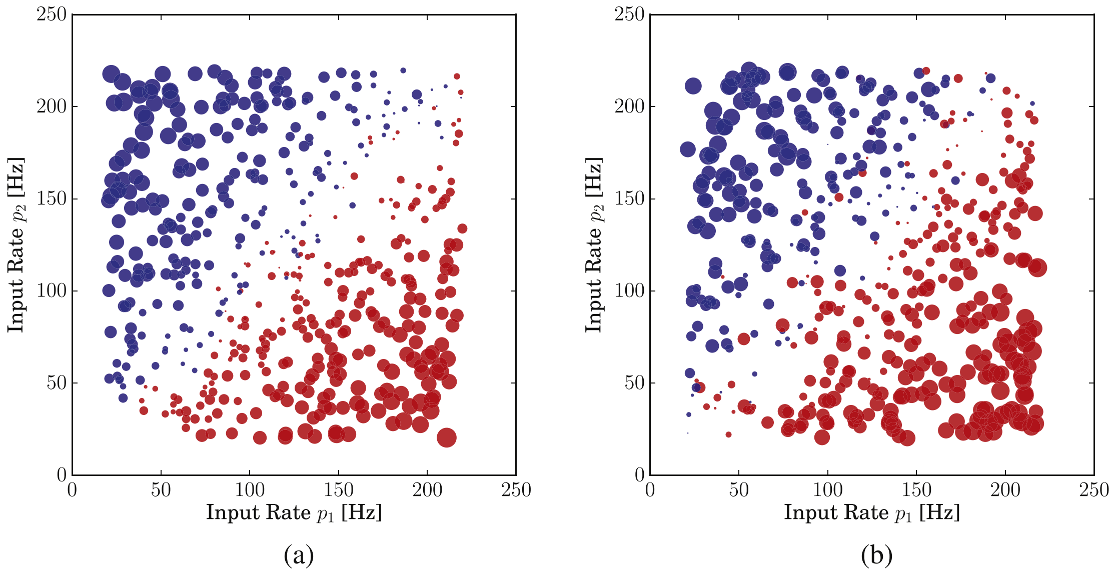

Figure 9. Comparison of the best classification output of the two neuron system with (a) binary and (b) 32-bit floating-point resolution synapses (accuracies are 93.2% and 98.6%; average case 76.8% and 96.7%). The size of the circles indicates the difference in output rates of the two neurons. The binary elements' weight distributions correspond the output of the proposed normalizer circuit incorporating memristors with resistance levels drawn from $\mathcal{N}(6\ \mathrm{K}\Omega,\ 1.2\ \mathrm{K}\Omega)$ (high) and $\mathcal{N}(0.6\ \mathrm{K}\Omega,\ 0.12\ \mathrm{K}\Omega)$ (low), correspoding to a CV of 5 and a high–low ratio of 10.

more precise estimate of the underlying rate parameter in a fixed time window. We show the best performing instances (both with $n_c = 100$).

The proposed learning rule finds a 'stable' solution to the classification problem, in that the misclassified points only elicit a slightly higher response in the wrong output neuron. The binary synapses reach a worse classification accuracy than the high resolution synapse, but the performance gap varies with the number of synapses/input channels. We observed that the gap closes in terms of the best performance, but is roughly stable for the average performance (from 14% to 5% for $n_c = \{25,\ 100\}$ in best performance, stable at ca. 18% in average performance). Since the test-time corresponds to 10^5 times the characteristic time of the individual neuron's Poisson spike trains, it is improbable that the instantiations of the poisson statistics caused this trial-to-trial difference in performance. This indicates that the binary update may sometimes get stuck in a bad configuration, and that in practice occasional restarting of the optimization procedure leads to better results.

4.2. Classifying multiple patterns

Here we show how it is possible to train a population of output neurons to classify multiple overlapping patterns in a supervised setting. For this demonstration we use the common benchmark of classifying handwritten digits from the MNIST data-set. Specifically, we test the system using MNIST digits from 0 to 4 scaled to 24 × 24 pixels, as in [28]. In the network, there is an input layer consisting of $24 \cdot 24 \cdot n_c$ Poisson neurons, whose spike rates are scaled according to the intensity of the MNIST digit image pixel, and the output layer consisting of 5 neurons that should learn to respond to the corresponding digit, and an additional layer of teacher neurons indicating which of 5 output neurons should fire in response to a given input. The index of the output neuron that fires the most in response to a test stimulus is considered the label that the network assigns to this input. During training 1000 digits, randomly drawn from the training set, are presented for 100 ms each while the learning circuits are enabled. The learning circuits are then disabled and the performance of the network is evaluated on 500 additional digits (randomly drawn from the test set). Further implementation details are given in the supplementary material.

The learning algorithm is the same one used in section 4.1. To compensate for the discretization errors, the update is made probabilistic as in [36]. Although we restrict ourselves to probabilistic signals that are independent per neuron, rather than per synapse, we achieve a performance comparable to that of [28].

In figure 10(a) we report the performance of the network as a function of the number n_c of synapses used per pixel, in terms of classification accuracy. In figure 10(b) we show two examples of the learned weight matrices.

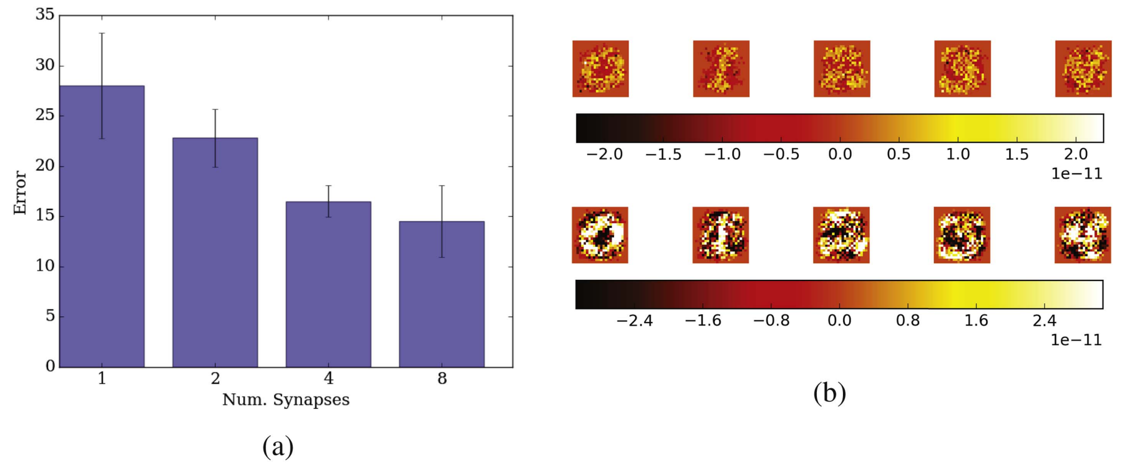

Figure 10. MNIST classification performance. (a) Error on the (reduced) MNIST classification task (test set) as a function of the number of binary synapses per input pixel for high and low CV. The CV settings correspond to figures 8(a) and (b). Errorbars indicate std. dev. on 5 repetitions. (b) Learned weight matrices with one and eight synaptic weights for neurons classifying digits 0 to 4 (left to right).

5. Discussion

The memristive synapse circuit proposed in this paper comprises two memristive devices, 20 MOSFETs, and a pulse-shaping circuit. Clearly, the area of this synaptic circuit is much larger than that of synapse elements employed in dense 1R or 1T-1R crossbar arrays [15, 19, 24]. However, this allows the system to scale to large synapse/neuron numbers, and to use all synapses in parallel. Furthermore, the currents passing through the memristive devices are contained within each synaptic element and do not diffuse to neighboring devices, eliminating sneak-path issues and enabling quick charging/discharging of high capacitive loads (in the supplementary material we discuss how similar strategies could be used for dense cross-bar arrays). The strategy of using two memristive devices per synapse allows the use of a normalizer circuit, which has the highly desirable property of minimizing the effect of variability across the memristive devices. While the strategy of using two memristive devices in a differential way also eliminates the need for a fixed reference in the normalizer circuit, and automatically provides the possibility to implement both positive and negative synaptic weights.

6. Conclusion

We proposed a differential current-mode memristive synapse circuit that decouples the current used to sense or change memristive device state from the current used to stimulate ultra low-power post-synaptic neuron circuits. We showed that the proposed circuit significantly reduces the effect of device variability, and that it is ideally suited for implementing advanced spike-based learning mechanisms that do not use overlapping pulses at the terminals of the target synapse. We argued that the strategy of using pulse extenders and Gilbert-normalizers in each synapse element maximizes throughput and minimizes power consumption in large-scale event-based neuromorphic computing platforms. Given that memory-related constraints, such as size and throughput, represent one of the major bottlenecks in conventional computing architectures [5], and given the potential of neuromorphic computing platforms to perform robust computation using variable and slow computing elements, the proposed circuit offers an attractive solution for building alternative non-von Neumann computing platforms with advanced and emerging memory technologies.

Acknowledgments

This work was supported by SNSF grant number CRSII2_160756 and by the EU ICT grant ‘NeuRAM^{3}’ (687299).

ORCID iDs

Manu V Nair https://orcid.org/0000-0002-0182-8358
Giacomo Indiveri https://orcid.org/0000-0002-7109-1689

References

[1] Chicca E, Stefanini F, Bartolozzi C and Indiveri G 2014 Neuromorphic electronic circuits for building autonomous cognitive systems *Proc. IEEE* **102** 1367–88

[2] Park J, Ha S, Yu T, Neftci E and Cauwenberghs G 2014 A 65k-neuron 73-Mevents/s 22-pJ/event asynchronous micro-pipelined integrate-and-fire array transceiver *Biomedical Circuits and Systems Conf. (BioCAS), 2014 IEEE* pp 675–8 IEEE

[3] Benjamin B V *et al* 2014 Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations *Proc. IEEE* **102** 699–716

[4] Merolla P A *et al* 2014 A million spiking-neuron integrated circuit with a scalable communication network and interface *Science* **345** 668–73

[5] Indiveri G and Liu S-C 2015 Memory and information processing in neuromorphic systems *Proc. IEEE* **103** 1379–97

[6] Mitra S, Fusi S and Indiveri G 2009 Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI *IEEE Trans. Biomed. Circuits Syst.* **3** 32–42

[7] Qiao N *et al* 2015 A re-configurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128 K synapses *Frontiers Neurosci.* **9**

[8] Huayaney F L M, Nease S and Chicca E 2016 Learning in silicon beyond STDP: a neuromorphic implementation of multi-factor synaptic plasticity with calcium-based dynamics *IEEE Trans. Circuits Syst. I: Regular Pap.* **63** 2189–99

[9] Moradi S, Indiveri G, Qiao N and Stefanini F 2015 Networks and hierarchical routing fabrics with heterogeneous memory structures for scalable event-driven computing systems *European Patent Application* EP 15/165272Filed 27.04.2015

[10] Qiao N and Indiveri G 2016 Scaling mixed-signal neuromorphic processors to 28 nm FD-SOI technologies *Biomedical Circuits and Systems Conf., (BioCAS), 2016* pp 552–5 IEEE

[11] Furber S, Galluppi F, Temple S and Plana L 2014 The SpiNNaker project *Proc. IEEE* **102** 652–65

[12] Schemmel J *et al* 2010 A wafer-scale neuromorphic hardware system for large-scale neural modeling *Proc. 2010 IEEE Int. Symp. on Circuits and Systems (ISCAS)* pp 1947–50 IEEE

[13] Ielmini D and Waser R 2015 *Resistive Switching: From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications* (New York: Wiley)

[14] Jo S H *et al* 2010 Nanoscale memristor device as synapse in neuromorphic systems *Nano Lett.* **10** 1297–301

[15] Prezioso M *et al* 2015 Training and operation of an integrated neuromorphic network based on metal-oxide memristors *Nature* **521** 61–4

[16] Suri M *et al* 2011 Phase change memory as synapse for ultra-dense neuromorphic systems: application to complex visual pattern extraction *2011 IEEE Int. Electron Devices Meeting (IEDM)* p 4 IEEE

[17] Milo V *et al* 2016 Demonstration of hybrid CMOS/RRAM neural networks with spike time/rate-dependent plasticity *2016 IEEE Int. Electron Devices Meeting (IEDM)* pp 16–8 IEEE

[18] Pedretti G *et al* 2017 Memristive neural network for on-line learning and tracking with brain-inspired spike timing dependent plasticity *Sci. Rep.* **7**

[19] Park S *et al* 2015 Electronic system with memristive synapses for pattern recognition *Sci. Rep.* **5** 10123

[20] Kelley H J 1960 Gradient theory of optimal flight paths *ARS J.* **30** 947–54

[21] Yao P *et al* 2017 Face classification using electronic synapses *Nat. Commun.* **8** 15199

[22] Georghiades A, Belhumeur P and Kriegman D 1997 Yale face database, Center for computational Vision and Control at Yale University, vol 2 http://cvc.~yale.~edu/projects/yalefaces/yalefa

[23] Deng L *et al* 2015 Complex learning in bio-plausible memristive networks *Sci. Rep.* **5** 10684

[24] Garbin D *et al* 2015 HfO_2-based OxRAM devices as synapses for convolutional neural networks *IEEE Trans. Electron Devices* **62** 2494–501

[25] Saighi S *et al* 2015 Plasticity in memristive devices *Frontiers Neurosci.* **9**

[26] Covi E *et al* 2016 Analog memristive synapse in spiking networks implementing unsupervised learning *Frontiers Neurosci.* **10**

[27] Serb A *et al* 2016 Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses *Nat. Commun.* **7** 12611

[28] Bill J and Legenstein R 2014 A compound memristive synapse model for statistical learning through STDP in spiking neural networks *Frontiers Neurosci.* **8**

[29] Brader J, Senn W and Fusi S 2007 Learning real world stimuli in a neural network with spike-driven synaptic dynamics *Neural Comput.* **19** 2881–912

[30] Graupner M and Brunel N 2012 Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location *Proc. Natl Acad. Sci.* **109** 3991–6

[31] Urbanczik R and Senn W 2014 Learning by the dendritic prediction of somatic spiking *Neuron* **81** 521–8

[32] Giulioni M *et al* 2012 Robust working memory in an asynchronously spiking neural network realized in neuromorphic VLSI *Frontiers Neurosci.* **5**

[33] Mostafa H, Mayr C and Indiveri G 2016 Beyond spike-timing dependent plasticity in memristor crossbar arrays *2016 IEEE Int. Symp. on Circuits and Systems (ISCAS)* pp 926–9 IEEE

[34] Nair M V and Dudek P 2015 Gradient-descent-based learning in memristive crossbar arrays *2015 Int. Joint Conf. on Neural Networks (IJCNN)* pp 1–7 IEEE

[35] Soudry D, Di Castro D, Gal A, Kolodny A and Kvatinsky S 2015 Memristor-based multilayer neural networks with online gradient descent training *IEEE Trans. Neural Netw. Learn. Syst.* **26** 2408–21

[36] Müller L, Nair M and Indiveri G 2017 Randomized unregulated step descent for limited precision synaptic elements *Int. Symp. on Circuits and Systems, (ISCAS), 2017* IEEE pp 1–4

[37] Deiss S, Douglas R and Whatley A 1998 A pulse-coded communications infrastructure for neuromorphic systems *Pulsed Neural Networks* ed W Maass and C M Bishop (Cambridge, MA: MIT Press) ch 6, pp 157–78

[38] Boahen K 2000 Point-to-point connectivity between neuromorphic chips using address-events *IEEE Trans. Circuits Syst.* II **47** 416–34

[39] Merolla P, Arthur J, Shi B and Boahen K 2007 Expandable networks for neuromorphic chips *IEEE Trans. Circuits Syst.* I **54** 301–11

[40] Chicca E *et al* 2007 A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity *IEEE Trans. Circuits Syst.* I **5** 981–93

[41] Delbruck T, Berner R, Lichtsteiner P and Dualibe C 2010 32-bit configurable bias current generator with sub-off-current capability *Int. Symp. on Circuits and Systems (ISCAS), 2010* pp 1647–50 Paris, France: IEEE

[42] Bartolozzi C and Indiveri G 2007 Synaptic dynamics in analog VLSI *Neural Comput.* **19** 2581–603

[43] Bartolozzi C, Mitra S and Indiveri G 2006 An ultra low power current-mode filter for neuromorphic systems and biomedical signal processing *Biomedical Circuits and Systems Conf. (BioCAS), 2006* pp 130–3 IEEE

[44] Livi P and Indiveri G 2009 A current-mode conductance-based silicon neuron for address-event neuromorphic systems *Int. Symp. on Circuits and Systems, (ISCAS), 2009* pp 2898–901 IEEE

[45] Indiveri G *et al* 2011 Neuromorphic silicon neuron circuits *Frontiers Neurosci.* **5** 1–23

[46] Gilbert B 1990 Current-mode circuits from a translinear viewpoint: a tutorial *Analogue IC Design: The Current-Mode Approach* ed C Tomazou *et al* (Herts: Stevenage) ch 2, pp 11–91

[47] Liu S-C, Kramer J, Indiveri G, Delbruck T and Douglas R 2002 *Analog VLSI: Circuits and Principles* (Cambridge, MA: MIT Press)

[48] Chen H-Y *et al* 2016 Resistive random access memory (RRAM) technology: from material, device, selector, 3D integration to bottom-up fabrication *J. Electroceram.* **109** 133504

[49] Brivio S *et al* 2016 Experimental study of gradual/abrupt dynamics of HfO_2-based memristive devices *Appl. Phys. Lett.* **109** 133504