An Adaptive Training Method for Memristive Next Generation Reservoir Computing Systems

K. Nikiruy, I. Petrenyov, A. Shkurmanov, M. Ziegler, *Chair of Energy Materials and Devices, Department of Materials Science, CAU,* Kiel, Germany, kni@tf.uni-kiel.de

J. Schneegaß, T. Ivanov, *Department of Electrical Engineering and Information Technology, Micro- and Nanoelectronic Systems, TU Ilmenau,* Ilmenau, Germany

D. Rossetti, F. Corinto, A. Ascoli, *Department of Electronics and Telecommunications, Politecnico di Torino,* Turin, Italy

A.S. Demirkol, R. Tetzlaff, *Faculty of Electrical and Computer Engineering, Institute of Circuits and Systems, TU Dresden,* Dresden, Germany

***Abstract* — Next-generation reservoir computing, combining reservoir computing with statistical prediction methods, has gained attention for its efficiency in computing time and dataset requirements for learning [1,2]. This approach has proven effective for predicting the behavior of complex dynamical systems, which is especially relevant in hardware implementations. We recently showed that memristors are promising candidates for encoding synaptic weights in hardware. However, ridge regression, typically used for training artificial neural networks, requires a precise adjustment of the memristors' resistances. Also, the memristive weights could be dynamically adapted over training, but this capability is not utilized in ridge regression. This work proposes an innovative next generation reservoir computing structure, which, subject to an adaptive learning process, becomes tolerant to the cycle-to-cycle variability inherent to memristive synapses. In particular, the network synaptic weights are conditionally modulated in steps over the learning phase through a gradient descent-based training protocol. This allows to mitigate the detrimental effects on the reservoir time series prediction accuracy due to the unavoidable stochastic errors originating in the process of writing weights into the memristive physical stacks. Additionally, the adaptive learning process may compensate for potential changes in the source data to predict.**

Keywords— Artificial Intelligence, Memristors, Machine Learning, Neural Networks, Reservoir Computing, Auto-Regression, Ridge Regression Training, Adaptive Gradient Descent Training, Learning, Inference

I. Introduction

Artificial Intelligence (AI) has been attracting considerable interest in the scientific community over the past decade, especially with the advent of nanotechnologies enabling the circuit implementation of bio-inspired data processing paradigms. In this context, the combination of *Reservoir Computing* (RC) with statistical prediction methods, also known as *Next Generation Reservoir Computing* (NGRC), enables extremely efficient performance in data processing and data set requirements for learning [1,2]. This method has proven its strengths in predicting the behavior of complex dynamical systems, particularly in hardware implementations [3,4]. As we recently demonstrated, memristors are particularly suitable for the realization of the synaptic weights in the output layer of an *artificial neural network* [4]. The standard procedure to train a reservoir uses a ridge regression algorithm [1], which employs data points from an input sequence in order to determine optimal values for the synaptic weights all at once within *a single step*. Such a learning procedure is however sensitive to the random errors in the conductances, written in the memristors, relative to the desired weights. Programming the memristors accurately in a reproducible form is a formidable challenge due to the *stochastic cycle-to-cycle variability* affecting their electrical response to the periodic stimuli employed to store synaptic weights into their physical stacks [5,6]. Moreover, the main advantageous reason for using a memristor as a synaptic emulator, namely its plasticity, cannot be used in a ridge regression-based learning protocol, due to the high variability it suffers from [4]. Introducing a novel learning algorithm, exploiting a continuous adjustment of the memristive weights, would enable the resulting neural network to adapt to time-varying conditions, allowing the implementation of multi-tasking paradigms across its physical medium.

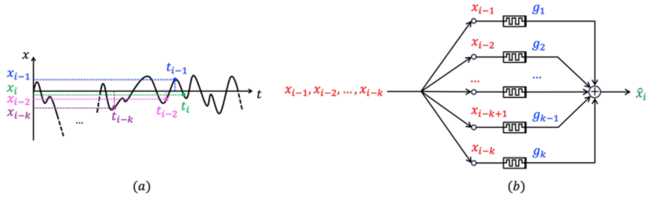

Fig. 1 Time series prediction of the i^{th} sample x_i of an input data set (a) through the mNGRC scheme (b). The input time series is not pre-processed. The memristors, encoding the synaptic weights, tuned over the *training* phase, are assumed to operate as linear resistors during the *inference* phase, which allows to test how well did the reservoir learn the computing task [1].

Motivated by the aim to overcome the open problem discussed above, in this work we introduce an innovative *Memristive Next-Generation Reservoir Computing* (mNGRC) system, which, by undergoing an adaptive learning process, acquires a more robust inference functionality, relative to the case, where it is merely subject to a ridge regression training protocol, especially against the intrinsic stochasticity of the memristive synapses. In particular, the synaptic weights of the network, and thus also the memristances, encoding them, are gradually adjusted during the learning phase, under suitable conditions, through an adaptive procedure based upon a well-known learning rule, referred to as gradient descent. Importantly, as demonstrated via numerical simulations, this approach helps reducing the negative impact, which stochastic errors, occurring unavoidably while mapping synaptic weights onto memristive physical stacks, have on the reservoir performance, and can also compensate for unexpected changes in the input data under processing.

II. Adaptive Memristive Next Generation Reservoir Computing

Fig. 1 shows a sketch of the mNGRC scheme for time series prediction. The reservoir employs an *autoregression model* for inferring the next sample of an input dataset. The synaptic coefficients that weight the past input samples, which are added in the polynomial expression for the current output sample, are encoded in the conductances of memristive devices [1]. More in detail, the time waveform of a state variable x of a nonlinear dynamic system (see the black trace in Fig. 1(a)) is sampled at regular time intervals. The resulting time series of length n is then fed into the network, k samples per step, for a *one-step-ahead prediction*, as shown in Fig. 1(b). For this purpose, at the time instant t_i, the mNGRC system employs the k data points, which directly precede the sample x_i, to be predicted, specifically x_{i-1}, …, and x_{i-k} ($i \in \{k+1, \dots, n\}, k << n$). The predicted value for x_i, called $\hat{x}_i$, is thus estimated via

$$\hat{x}_i = \sum_{j=1}^{k} w_j \cdot x_{i-j}, \quad (1)$$

where the real-valued polynomial coefficients[1] w_1,…, w_k, also known as *synaptic weights*, are encoded in the conductances g_1,…, g_k of k memristive devices through a suitable programming step, which implements the mapping operation $w_j \rightarrow g_j$ for any index j in the set $\{1, \dots, k\}$.

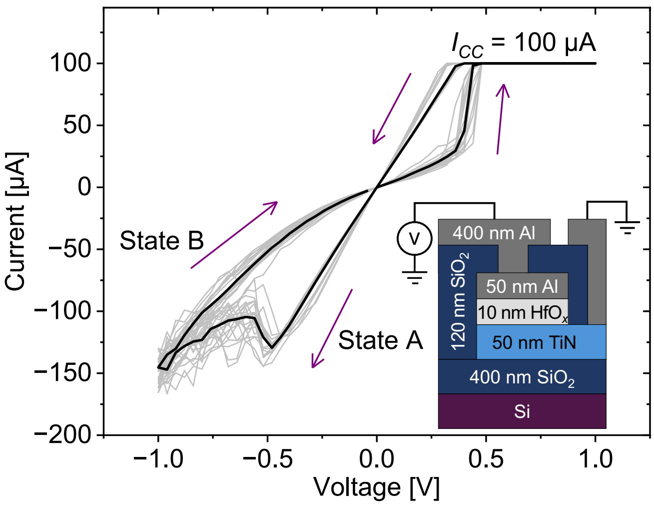

Fig. 2. Quasi-static current-voltage locus of a device sample under 20 consecutive resistance switching cycles of a purely-AC periodic triangular voltage signal of amplitude 1V and frequency 0.1Hz applied between its top and bottom electrodes. The grey traces reveal the cycle-to-cycle variability in the RESET and SET resistance switching transitions, which the non-volatile memristor undergoes in each cycle during the negative and positive excursions of the voltage stimulus, respectively. Extracting the median data point of the currents, recorded for each stimulus value across all the 20 cycles, results in the black locus. The modulus of the current is constrained to keep smaller than a RESET (SET) compliance level, fixed here to 1mA (100 μA), to prevent an irreversible damage to the physical stack. The SET compliance level also allows to re-program the device SET and RESET conductance levels, also referred to as states A and B in the remainder of the paper, respectively. The inset shows a sketch of the device physical stack, where the resistance switching medium is a sub-stoichiometric Hafnium oxide (HfO_x) layer.

Thus, the proposed energy-efficient hardware implementation requires just one column of k memristive devices to apply a k-th order one-step-ahead prediction of a sequence of $n-k$ input data points. Critically, for the mapping operation to work ideally, the conductances of the memristors should be linearly tunable under suitable stimulation. Moreover, the memristors should exhibit a large off-to-on resistance ratio and their operating principles should be robust against variability sources. The following section describes the methodology of fabrication adopted in house to produce memristive devices, showcasing a good yet non-ideal performance in these critical figures of merit. These devices shall be later used in the reservoir from Fig. 1(b) for testing the proposed adaptable learning paradigm.

III. Device Fabrication and Electrical Characteristics

A. Device Fabrication

A thin-film technology was employed for the fabrication of the memristive devices (the Al/HfO_x/TiN layer sequence of one device sample is shown in the inset of Fig. 2). In order to make a statistically sound statement about the electrical properties of the memristors, taking into account their intrinsic variability, a wafer-scale fabrication process was pursued. This resulted in the production of a 4-inch wafer hosting more than 30000 individual memristors. At the beginning of the fabrication, a 4-inch Si wafer was subject to a wet oxidation process at a temperature T of 1050 °C, which turned it into a 400 nm SiO_2 layer. Then, 50 nm thick TiN bottom electrodes were deposited on the SiO_2 layer through a 90 s DC magnetron sputtering process at a power P of 800 W injecting a gaseous mixture of nitrogen and argon in the chamber. Afterwards, a 10 nm thick HfO_x layer was deposited on an area of 50×50 μm^2 by using RF sputtering at a power of 500 W, injecting in the chamber, kept at a pressure P of 0.005 mbar, the same gaseous mixture, mentioned above, with a Ar:O_2 flow ratio of 1:1. Then DC sputtering at a power of 500 W was pursued to deposit 50 nm thick Al top electrodes. To passivate the area, a 130 nm SiO_2 layer was evaporated by means of an inductively coupled plasma chemical vapor deposition process. Finally, to allow connections to external circuitry, a 400 nm thick Al layer was sputtered on top of the overall physical structure under DC conditions. All the layers were photolithographically patterned through the use of a mask aligner and a subsequent etching process.

B. Device Electrical Characteristics

In order to endow the memristor samples with reversible resistance switching capability, the fabrication process was followed by an electroforming step performed through a probe station equipped with a "Keysight B2911B" source-measurement unit. During this step, the sample was subject to a constant forming voltage of 4V, and, concurrently, the current, flowing through it, was limited to a compliance level of 100 μA to prevent a hard breakdown of the physical stack in the resulting SET transition. The electroforming process induced the formation of a conductive filament through the HfO_x "active" layer. Unfortunately, this step is one of the major factors at the origin of the stochasticity characterizing the memristor conductance modulation process via periodic excitation, as explained in detail next. A typical post-electroforming current-voltage locus of a memristive sample, as a periodic quasi-static triangular voltage waveform of amplitude 1V, zero time average, and frequency 0.1Hz was let fall between its terminals, is shown in Fig. 2 for 20

[1] While in this study the synaptic weights are allowed to assume both positive and negative values, their mapping onto the strictly-positive conductance levels, which may be stored in a passive resistance switching memory, calls for the development of ad hoc circuit design strategies in the hardware realization of the mNGRC engine.

consecutive switching cycles. Here, the compliance levels of the current in the positive and negative excursion of the voltage stimulus were respectively set to 100 µA and to 1 mA. The device was found to exhibit a bipolar resistive switching behavior, undergoing a SET (RESET) process, for positive (negative) values of the voltage stimulus. Despite, in the experiment, Fig. 2 refers to, the memristor switches essentially between two resistance states, its conductance can be re-programmed to different levels by varying the current compliance during the SET process, as demonstrated in Fig. 3. In the underlying experiment, one device sample was reprogrammed into as many as 34 different resistive levels, 17 for state A and 17 for state B, by modulating the current compliance for the SET process in the purely-AC periodic quasi-static stress test illustrated in Fig. 2. This enables to tune the device conductance across a range of values spanning nearly one order of magnitude. In particular, the higher conductance state A was discretely modulated from 270 µS to 660 µS, whereas the lower conductance state B was stepped from 60 µS to 330 µS. As may be evinced by inspecting the red (black) dataset in Fig. 3, however, the read measurements of each conductance level from state A (B), following each of the 20 consecutive SET (RESET) transitions, provide proof of evidence for the non-negligible cycle-to-cycle variability the memristive device suffers from. As may be evinced from the figure inset, the distribution of the relative standard deviation across the entire conductance measurement data set reveals that the fluctuations of a resistance level, relative to the mean between the corresponding 20 read measurements, may be as large as 40%. In the worst-case scenario the ridge regression algorithm, typically employed to train the memristive synaptic weights of the reservoir of Fig. 1(b), would be inadequate, as will be demonstrated shortly. Reducing the error in the conductance programming phase to a bare minimum would require additional error correction cycles, which however increase the energy consumption and reduce the time efficiency of the process. On the basis of this analysis, as outlined in the section to follow, we propose a novel training methodology, inspired to the gradient descent algorithm, which allows to adapt conditionally the memristive synaptic weights of the reservoir in Fig. 1(b) during the learning phase, even while adopting the non-ideal yet time- and energy-efficient programming strategy described above.

IV. Reservoir Training Methods

State-of-the-art training method

For reservoir computing a *ridge regression* training protocol is commonly used to train the j^{th} synaptic weight w_j, for each index $j \in \{1, \dots, k\}$, as described in [1]-[3]. Here, before the computation of the weight vector $\boldsymbol{w}$, whose components are $w_1, \dots, w_k$, a matrix $\boldsymbol{S}$ is formed from the training data points according to

$$\boldsymbol{S} = \begin{pmatrix} x_k & \cdots & x_1 \\ \vdots & \ddots & \vdots \\ x_{h-1} & \cdots & x_{h-k} \end{pmatrix}, \qquad . \tag{2}$$

where the m^{th} row contains in fact the set of input samples $x_{k+m-1}, \dots, x_m$ necessary to provide an estimate, referred to as $\hat{x}_{k+m}$, for the input data point x_{k+m} ($m \in \{1, \dots, h-k\}$). The weight vector $\boldsymbol{w}$ is then calculated via

$$\boldsymbol{w} = (\boldsymbol{S}^T \cdot \boldsymbol{S} + \gamma \cdot \boldsymbol{I})^{-1} \cdot \boldsymbol{S}^T \cdot \boldsymbol{x_T}, \tag{3}$$

where γ is the so-called regression coefficient, set here to 10^{-8}, $\boldsymbol{I}$ stands for the identity matrix, and $\boldsymbol{x_T} \triangleq [x_{k+1}, \dots, x_h]^{\mathrm{T}}$ denotes the $(h-k)$-long *training target output vector*, composed of the part of the n-point time series, which is predicted during the training phase (the remaining $n-h$ samples of the overall input sequence $\boldsymbol{x} = [x_1, \dots, x_n]^{\mathrm{T}}$, collected in the *inference target output vector* $\boldsymbol{x_I} \triangleq [x_{h+1}, \dots, x_n]^{\mathrm{T}}$, are then used for testing). This training protocol stands out for the time efficiency, yet its accuracy demands an error-free weight update. However, the variability in the conductance of any memristor sample from programming cycle to programming cycle results in significant inaccuracies during the reservoir training phase.

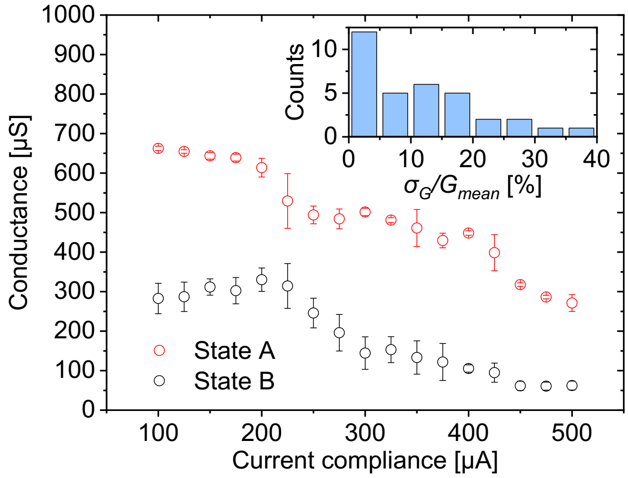

Fig. 3. Dependence of the conductance of one of the device samples upon the SET current compliance for each of the 17 levels the state A and the state B are separately programmed onto through the quasi-static test illustrated in Fig. 2. For each compliance current the spread in the data point, associated to state A (B), is computed from the 20 conductance read measurements carried out after each of the 20 consecutive SET (RESET) resistance switching processes triggered by the purely-AC periodic triangular voltage stimulus. Denoting the mean value and the standard deviation for the set of 20 conductance readings, directly following the i^{th} SET (RESET) process, under the i^{th} SET compliance current $I^{(i)}_{C,SET}$, as $\bar{G}^{(i)}_{SET(RESET)}$ and as $\sigma^{(i)}_{G,SET(RESET)}$, respectively, the histogram on the inset of Fig. 3 shows the relative standard deviation $\sigma^{(i)}_{G,SET(RESET)}/\bar{G}^{(i)}_{SET(RESET)}$ for $i \in \{1,\dots,17\}$. Each of the 17 runs allows to measure the conductance of the device after both the SET and RESET transitions for 20 consecutive cycles. Looking at its statistical distribution, the relative standard deviation keeps below 20% for most of the 34 estimates, but it increases up to 40% in the worst-case scenario.

Adaptive training method

To compensate for the inherent cycle-to-cycle variability, affecting the electrical behavior of a memristive sample, we propose an innovative adaptive procedure, that conditionally tunes the synaptic weights, according to the gradient descent learning rule, during the entire training phase. For each index $j \in \{1, \dots, k\}$, we first initialize the j^{th} synaptic weight w_j with a value randomly drawn from a normal distribution with mean 0 and a standard deviation, indicated as σ_0, and treated as a tunable parameter later on (see especially Fig. 5(b), where σ_0 is named *initial weight variation*). The initialization is employed to make the first one-step-ahead prediction, specifically to estimate the data point x_{k+1} at $t = t_{k+1}$. Over the course of the training phase, the j^{th} synaptic weight w_j is updated at any given time instant t_i, if and only if the prediction error $\varepsilon_i \triangleq |\hat{x}_i - x_i|$ for the i^{th} input sample x_i exceeds a threshold ε_{th} , which is set to 0.1 in this study ($i \in \{k+1, \ldots, h\}$). At that time instant, the cost function C_i, employed for the gradient descent rule, is expressed as

$$C_i = \frac{1}{q} \sum_{r=1}^{q} (\hat{x}_{i-r} - x_{i-r})^2, \tag{4}$$

where q, set here to 5, denotes the number of input data points, directly preceding the i^{th} sample x_i, for which the calculation of the cost function at time t_i is made. The r^{th} component in the sum from (4) is the square of the error between the sample x_{i-r} in the training target output vector $\boldsymbol{x_T} = [x_{k+1}, \ldots, x_h]^T$ at time instant t_{i-r} and the corresponding prediction value $\hat{x}_{i-r}$. From equation (1), taking into account the adaptability of the k synaptic weights in the reservoir of Fig. 1(b) during the training phase, the prediction for x_i can be recast as

$$\hat{x}_i = \sum_{j=1}^{k} w_j^{(i-1)} \cdot x_{i-j}. \tag{5}$$

where $w_j^{(i-1)}$ denotes the j^{th} synaptic weight at $t = t_{i-1}$. The cost function at $t = t_i$ can be rewritten as

$$C_i = \frac{1}{q} \cdot \sum_{r=1}^{q} \left(\left(\sum_{j=1}^{k} w_j^{(i-r-1)} \cdot x_{i-r-j} \right) - x_{i-r} \right)^2. \tag{6}$$

This form allows a direct calculation of the partial derivative of C_i with respect to $w_j^{(i-1)}$ through

$$\frac{\partial C_i}{\partial w_j^{(i-1)}} = \frac{2}{q} \cdot \sum_{r=1}^{q} \left(\left(\sum_{j=1}^{k} w_j^{(i-r-1)} \cdot x_{i-r-j} \right) - x_{i-r} \right) \cdot x_{i-r-j}$$
$$= \frac{2}{q} \cdot \sum_{r=1}^{q} (\hat{x}_{i-r} - x_{i-r}) \cdot x_{i-r-j}. \tag{7}$$

Thus, if the error ε_i in the prediction of the input sample x_i at $t = t_i$ exceeds the predefined threshold ε_{th}, for each index $j \in \{1, \ldots, k\}$, the j^{th} synaptic weight $w_j^{(i-1)}$ at the previous time step is updated by a quantity $\Delta w_j^{(i-1)}$, which is proportional to the partial derivative of the cost function C_i with respect to $w_j^{(i-1)}$, up to a negative scaling factor, in accordance with the gradient descent rule. Specifically, the j^{th} weight correction factor is computed via

$$\Delta w_j^{(i-1)} = -\gamma' \cdot \frac{\partial C_i}{\partial w_j^{(i-1)}}, \tag{8}$$

where γ', set here to 0.5, denotes the learning rate. In order to model the non-ideal operating principles of our memristive synapses, expected to encode the synaptic weights in the hardware implementation of the reservoir, for each index $j \in \{1, \ldots, k\}$, the updated value for the j^{th} synaptic weight at $t = t_i$, i.e.

$$w_j^{(i)} = w_j^{(i-1)} + \Delta w_j^{(i-1)}, \tag{9}$$

with $\Delta w_j^{(i-1)}$ computed deterministically via equation (7), is further perturbed by an additive stochastic component sampled from a normal distribution with mean value $w_j^{(i)}$ and standard deviation $\sigma_{w_j^{(i)}}$. Importantly, in the analysis to follow, the relative standard deviation $c^{(i)} = \sigma_{w_j^{(i)}} / w_j^{(i)}$ will be treated as a tunable parameter independent of i (see especially plots (a) and (b) of Fig. 5, where such constant, indicated as c, is called *weight variation*). At a certain time instant t_i, the k weights attain values, which sets the cost function C_i so close to its global minimum that the prediction error keeps consistently below the threshold ε_{th}, thereafter. Beyond this point, further modifications to the weights would necessitate alterations to the training sequence under prediction.

V. RESULTS AND DISCUSSION

To demonstrate the extent, to which the adaptive training method, introduced in the previous section, can compensate for the inherent memristor cycle-to-cycle variability, a mNGRC network, falling in the class illustrated in Fig. 1(b) of Section II, and hosting as few as 5 memristors, is trained using both ridge regression and the adaptive training method proposed here. The task involves performing a one-step-ahead prediction of the temporal solution of a classical nonlinear ordinary differential equation (ODE) system, referred to as *Duffing oscillator* in the literature. The ODE, describing the nonlinear dynamics of a damped spring with nonlinear restoring force under the effects of a periodic stimulus, reads as

$$\ddot{x} + \delta \dot{x} - F(x) = F_S(t), \tag{10}$$

where x is the displacement of the spring from the rest position, δ is the damping parameter, while

$$F(x) = -\alpha \cdot x - \beta \cdot x^3, \text{ and} \tag{11}$$

$$F_S(t) = \gamma\, (\cos(\omega_1 \mathrm{t}) + \cos(\omega_2 t)) \tag{12}$$

are the nonlinear restoring force, where α and β are real coefficients, and the periodic stimulus, composed of the sum of two cosine waves of angular frequencies ω_1 and ω_2 and common amplitude γ, respectively. Setting the model parameter setting, as reported in Table I, the oscillator exhibits a chaotic behaviour.

Table I. Model parameter setting for the ODE (10)-(12). The physical units are omitted for simplicity.

δ	α	β	γ	ω_1	ω_2
0.02	1	0.1	4	1	1.1

The two training protocols under comparison are applied to the mNGRC network of Fig. 1(b) for $k = 5$, assuming, as a worst-case scenario, a 40% stochastic deviation of the synaptic weights from their nominal values, choosing a normal distribution for the selection of the random perturbations. For the ridge regression algorithm, the displacements of the synaptic weights occurs only once after the computation of their nominal values via equation (3). In regard to the proposed adaptive training paradigm, for each index $j \in \{1, \ldots, k\}$, a stochastic perturbation is added up to the j^{th} synaptic weight at the beginning of the training phase for the prediction of the input sample x_{k+1}, as well as to its updated value $w_j^{(i)}$, computed through equation (9), at each subsequent time instant t_i, when the error ε_i between the i^{th} sample $\hat{x}_i$ in the prediction time series and the i^{th} sample x_i in the input data sequence were found to exceed the threshold ε_{th}. As shown in Fig. 4(a), the time series prediction capability of the mNGRC network, trained via ridge regression, is jeopardized by the detrimental effects originating from the poor reproducibility of the memristor programming phase. On the other hand, Fig. 4(b) demonstrates how well does the very same reservoir learn how to make a one-step-ahead prediction of the chaotic solution to the Duffing oscillator ODE (10)-(12), when it undergoes the proposed adaptive training, despite the cycle-to-cycle variability in each of the 34 resistance levels, 17 for the higher conductance state A and 17 for the lower conductance state B, which may be stored in the resistance switching memory through the purely-AC periodic stress test illustrated in Fig. 3.

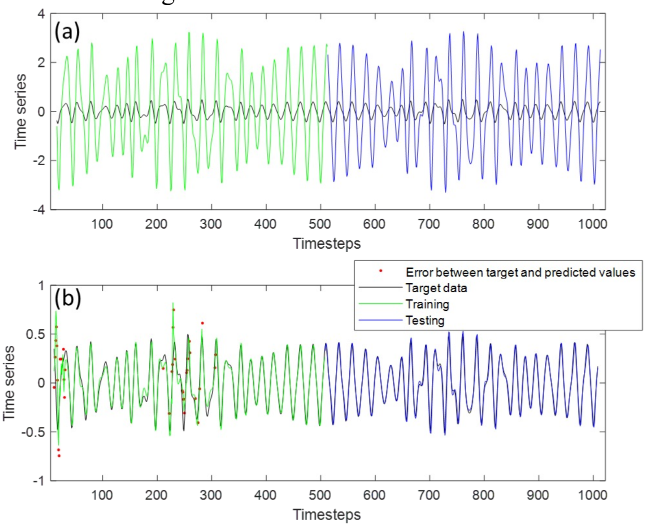

Fig. 4 One-step-ahead prediction of a chaotic time series, solving the Duffing oscillator ODE model (10)-(12) with parameter setting from Table I, through the mNGRC system of Fig. 1(b). The ridge regression algorithm, plot (a) refers to, was preliminary applied to the dataset, enabling the reservoir to learn the computing task of interest, and collected in the training target output vector $\boldsymbol{x_T} = [x_{k+1}, \dots, x_h]^{\mathrm{T}}$, to determine nominal values for the k synaptic weights. Then, after displacing the j^{th} weight w_j out of its nominal value through an additive random perturbation drawn from a normal distribution with mean value w_j and relative standard deviation $c = \sigma_{w_j}/w_j = 0.4$, for each index $j \in \{1, \dots, k\}$, the entire input time series $\boldsymbol{x} = [x_1, \dots, x_n]^{\mathrm{T}}$, where the testing dataset, collected in the inference target output vector $\boldsymbol{x_I} = [x_{h+1}, \dots, x_n]^{\mathrm{T}}$ directly follows the training sequence, was injected into the reservoir. The green (blue) trace shows the response of the network over the training (testing) phase. The black trace shows the input sequence, the first (second) half of which is used for training (testing). Clearly, the mNGRC network is unable to learn how to perform a one-step-ahead prediction of the chaotic time series under the detrimental effects originating from the cycle-to-cycle variability affecting the electrical behaviour of its memristive synapses. (b) Response of the mNGRC network of Fig. 1(b) to the same input dataset, employed in (a), under the assumption that, in the first half of the simulation, and for $j \in \{1, \dots, k\}$, at each time instant t_i, with $i \in \{k + 1, \dots, h\}$, where the gradient descent algorithm, employed here for training the reservoir, identifies an error ε_i between network output $\hat{x}_i$ and respective training input data point x_i beyond a predefined threshold ε_{th}, set to 0.1, the j^{th} weight $w_j^{(i-1)}$ from the previous time step is first updated according to equation (9), and then displaced from the updated value $w_j^{(i)}$ through a perturbation drawn from a normal distribution with mean value $w_j^{(i)}$ and constant relative standard deviation $c^{(i)} = \sigma_{w_j^{(i)}}/w_j^{(i)}$ assumed to equal a constant c, taken equal to 0.4 for a fair comparison with the ridge regression protocol. In order to allow the prediction of the first sample in the training target output vector $\boldsymbol{x_T}$, specifically x_{k+1}, each synaptic weight was randomly initialized from a normal distribution with mean 0 and standard deviation σ_0 taken equal to 0.4. The green, blue, and black traces have the same significance as in plot (a). Each red point on the green trace appears at a time instant t_i, at which $\varepsilon_i > \varepsilon_{th}$, which triggers an update to the k synaptic weights. Clearly, our adaptive weight update paradigm successfully trains the reservoir, compensating for the inherent stochasticity in the cycle-to-cycle response of each memristor to the periodic quasi-static stimulus employed to reprogram its conductance state. In both simulations, plots (a) and (b) refer to, k, h and n were respectively set to 5, 500 and 1000.

To gain a deeper insight into the impact of the cycle-to-cycle variability, affecting negatively the reproducibility of the memristor conductance update procedure, illustrated in Fig. 2, on the efficacy of the ridge regression (adaptive gradient descent-based) training method, the constant c, indicating the relative standard deviation of the normal distribution, from which a random perturbation to each synaptic weight is sampled, each time where its update is necessary (once and for all after its optimal value is extracted from the k-long vector computed via equation (3)), was incrementally increased in 40 steps from 1% to 40%. Moreover, in the proposed training protocol, for each value, assigned to c, the standard deviation σ_0 of the normal distribution, from which the initial value of each weight was sampled, was also incrementally increased in 40 steps from 1% to 40%. For each value assigned to the parameter c (pair of values assigned to the two parameters in the pair (c, σ_0)), the ridge regression (adaptive gradient descent) training protocol was run as many as 1000 times. The mean squared error (MSE) in the one-step-ahead prediction of the $(n-h)$-long inference target output vector $\boldsymbol{x_I} = [x_{h+1}, \dots, x_n]^{\mathrm{T}}$, extracted from the n-point Duffing cell chaotic time series $\boldsymbol{x} = [x_1, \dots, x_n]^{\mathrm{T}}$, for the l^{th} simulation run, indicated as $\mathrm{MSE}^{(l)}$, is computed via

$$\mathrm{MSE}^{(l)} = \frac{1}{n-h} \cdot \sum_{i=h+1}^{n} \varepsilon_{i,l}^2, \tag{13}$$

where, for $i \in \{h+1, \dots, n\}$, $\varepsilon_{i,l} = |\hat{x}_{i,l} - x_i|$ denotes the error between the i^{th} element x_i in the testing sequence of the input time series and the respective prediction value $\hat{x}_{i,l}$ for the l^{th} realization of the numerical simulation, l ranging in the set $\{1, \dots, 1000\}$. The median level across the MSE numbers, collected in the 1000 simulation runs for each parameter c (for each parameter pair (c, σ_0)) during the testing phase, following the application of the ridge regression (of the adaptive gradient descent) training algorithm, is then plotted as a blue (black) point marker in Fig. 5(a). In regard to the inference results, resulting from the proposed training protocol, in correspondence to each value, assigned to the parameter c, a green point marker is also reported in Fig. 5(a) to show the median level across all the median MSE numbers, computed, as explained above, for each of the 40 levels σ_0 was set to, while keeping c to the predefined level, and expressed by the ordinates of the vertically-aligned black point markers featuring c as abscissa. Except for those scenarios, where the relative standard deviation c is set to values below 0.03, training the mNGRC engine through the proposed adaptive algorithm improves its capability to predict the testing part of the input sequence over the case when it learns the computing task of interest according to the ridge regression protocol. The performance mismatch becomes more and more significant as c increases toward its worst-case 0.4 upper bound, since increments in the relative standard deviation truly weaken the time series prediction capability of the reservoir over the testing phase only when the synaptic weights are programmed through the ridge regression paradigm during the training process. The green trace increases only mildly with increments in c, assuming values below $3.1 \cdot 10^{-3}$. Last but not least, referring to Fig. 5 once again, plot (b), combining the information inferable from the coordinates of the 1600 black markers, reported in plot (a), shows a color map, which illustrates in a compact form the time series prediction performance of the reservoir during the inference phase, following the proposed adaptive training process, on the parameter plane, spanned by σ_0 on the horizontal axis and by c on the vertical axis. Here, importantly, for a given value, assigned to c, the time series prediction performance of the mNGR core, as the inference phase is under way, appears to be rather insensitive to increments in σ_0 up to the worst-case 0.4 upper bound. The robustness of the inference performance of the reservoir of Fig. 1(b) against the stochastic cycle-to-cycle variability, affecting inevitably the electrical behaviour of each of the memristors, it hosts, in the hypothesis the proposed adaptive training process is executed to set appropriate values for the synaptic weights, is of fundamental importance in view of a future hardware realization of the computing system.

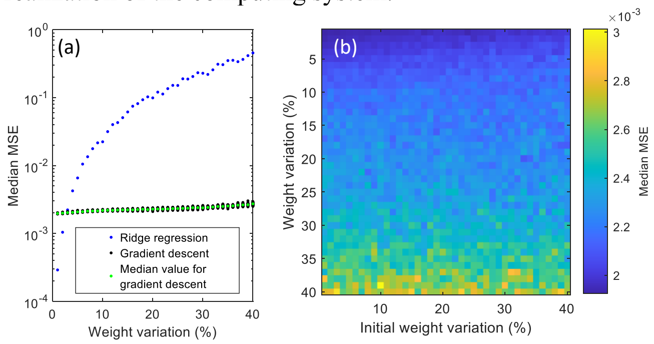

Fig. 5. (a) Illustration allowing to evaluate the capability of the mNGRC system from Fig. 1(b) to predict the $(n-h)$-long testing part of the n-point Duffing oscillator chaotic time series $\boldsymbol{x} = [x_1, \dots, x_n]^T$ after its synaptic weights are subject to a learning process based upon the ridge regression protocol and upon the proposed adaptive gradient descent paradigm, respectively. Each blue (black) point marker shows, for one particular choice of the parameter c (of the parameter pair (c, σ_0)), the median value between the numbers, computed for the MSE between the inference input vector $\boldsymbol{x_i} = [x_{h+1}, \dots, x_n]^T$ and the respective prediction vector $\widehat{\boldsymbol{x}}_i = [\hat{x}_{h+1}, \dots, \hat{x}_n]^T$ across 1000 realizations of a numerical simulation, in which the reservoir was let learn the computing task of interest according to the ridge regression (proposed adaptive gradient descent) protocol before the inference phase. The ordinate of the green point marker, in correspondence to each value, assigned to the parameter c, is the median of the ordinates of the 40 vertically-aligned black markers sharing the abscissa c. Clearly, the adaptive gradient descent training protocol outperforms the ridge regression counterpart. (b) Color map revealing how well does the mNGRC engine predict the Duffing oscillator chaotic time series, despite the stochasticity of the memristor programming and initialization process, when it undergoes the proposed adaptive learning phase. Each median MSE value, corresponding to one of the ordinates of the 1600 black point markers from plot (a), appears on the c versus σ_0 parameter plane as a point marker, whose hue is appropriately chosen according to the median MSE-color coding map, illustrated in the legend.

VI. Conclusion

This paper introduces a novel adaptive method to train the synaptic weights in memristive next-generation reservoir computing networks. The classical ridge regression training process employs a well-defined computational protocol to find optimal weights for the reservoir in a single step. Here, however, the practical impossibility for encoding weights onto memristive conductance states in a precise and reproducible form results in an inevitable and unacceptable degradation in the reservoir one-step-ahead time series prediction performance. On the other hand, by inducing suitable step-wise changes to the synaptic weights, so as to allow the trajectory point, moving, as a result, along the multi-dimensional surface of a history-dependent cost function, to converge progressively toward its global minimum, the proposed adaptive gradient descent-based learning algorithm is able to endow the predictive capability of the reservoir with robustness against the detrimental perturbation effects originating from the stochastic cycle-to-cycle variability characterizing switching transitions in memristive devices. Together with other recent studies on non-volatile ([4], [8]-[10]) and volatile memristors [11]-[16], this work sheds light on the promising potential of resistance switching memories for the design of energy-efficient bio-inspired circuits and systems for the electronics of the future.

Acknowledgment

The research study, presented in this paper, has been funded by the Deutsche Forschungsgemeinschaft, DFG for short (German Research Foundation) – Project 536063366, RECOMMEND - DFG Schwerpunktprogramm (Priority Program) 2262, Memristive Devices Toward Smart Technical Systems, MemrisTec for short.

References

[1] D.J. Gauthier, E. Bollt, A. Griffith, and W.A. Barbosa, “Next generation reservoir computing,” *Nat. Commun.*, vol. 12, no. 1, pp. 1-8, 2021

[2] W.A. Barbosa, and D.J. Gauthier, “Learning spatiotemporal chaos using next-generation reservoir computing,” *Chaos*, vol. 32, no. 9, 093137 (11pp.), 2022

[3] R.M Kent, W.A.S. Barbosa, and D.J Gauthier, “Controlling chaos using edge computing hardware,” *Nat. Commun.*, vol. 15, 3886, 2024.

[4] K. Nikiruy, T. Ivanov, M. Ziegler, D. Rossetti, F. Corinto, A. Ascoli, R. Tetzlaff, A.S. Demirkol, N. Schmitt, “Next Generation Memristor Reservoir Computing,” *IEEE MetroXRAINE*, pp. 912-917, 2024

[5] T. Gokmen and Y. Vlasov, ”Acceleration of Deep Neural Network Training with Resistive Cross-Point Devices: Design Considerations,” *Frontiers in Neuroscience*, vol. 10, 333, 2016.

[6] G.W. Burr, P. Narayanan, R.M. Shelby, S. Sidler, I. Boybat, C. di Nolfo, Y. Leblebici, ” Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: comparative performance analysis (accuracy, speed, and power),” *IEEE IEDM*, pp. 4.4.1-4.4.4, 2015, DOI: 10.1109/IEDM.2015.7409625.

[7] S. Houri, M. Asano, H. Yamaguchi, N. Yoshimura, Y. Koike, and L. Minati, “Generic Rotating-Frame-Based Approach to Chaos Generation in Nonlinear Micro- and Nanoelectromechanical System Resonators”, *Phys. Rev. Lett.*, vol. 125, no. 17, 174301 (6pp.), 2020

[8] C. Lenk, P. Hövel, K. Ved, S. Durstewitz, T. Meurer, T. Fritsch, A. Männchen, J. Küller, D. Beer, T. Ivanov, and M. Ziegler, “Neuromorphic acoustic sensing using an adaptive microelectromechanical cochlea with integrated feedback,” *Nature Electronics*, vol. 6, no. 5, pp. 370-380, May 2023

[9] A. Ascoli, et al., “A Deep Study of Resistance Switching Phenomena in TaOx ReRAM cells: System-Theoretic Dynamic Route Map Analysis and Experimental Verification,” *Adv. Ele. Mat.*, vol. 8, 2200182(31pp.), 2022

[10] I. Messaris, A. Ascoli, A.S. Demirkol, and R. Tetzlaff, “High Frequency Response of Non-Volatile Memristors,” *IEEE Trans. Circuits and Systems-I (TCAS-I): Regular Papers*, vol. 70, no. 2, pp. 566-578, 2023, DOI: 10.1109/TCSI.2022.3219368

[11] A. Ascoli, A.S. Demirkol, R. Tetzlaff, and L.O. Chua, “Edge of Chaos is Sine Qua Non for Turing Instability,” *IEEE TCAS-I*, vol. 69, no. 11, pp. 4596-4609, Nov. 2022

[12] A. Ascoli, A.S. Demirkol, R. Tetzlaff, and L.O. Chua, “Edge of Chaos Theory Resolves Smale Paradox,” *IEEE TCAS-I*, vol. 69, no. 3, pp. 1252-1265, 2022

[13] A. Ascoli, et al., “Edge of Chaos Theory Unveils the First and Simplest Ever Reported Hodgkin-Huxley Neuristor,” *Adv. Ele. Mat.*, 2025, DOI:10.1002/aelm.202400789

[14] A.S. Demirkol, A. Ascoli, I. Messaris, V. Ntinas, D. Prousalis, and R. Tetzlaff, “A Fast and Compact Threshold Switch Based Cellular Nonlinear Network Cell,” *IEEE TCAS-I*, 2025, DOI: 10.1109/TCSI.2025.3583799

[15] A. Ascoli, , et al., “NDR Effects in a Locally-Active Memristor Induce Small-Signal Amplification in a Simple Cell,” *MOCAST*, Dresden, 2025

[16] A. Ascoli, A.S. Demirkol, R. Tetzlaff, and L.O. Chua, “Analysis and Design of Bio-Inspired Circuits with Locally-Active Memristors,” *IEEE Trans. Circuits and Systems-II (TCAS-II): Express Briefs*, vol. 71, no. 3, pp. 1721-1726, March 2024