Experimental Assessment of Multilevel RRAM-Based Vector-Matrix Multiplication Operations for In-Memory Computing

Emilio Perez-Bosch Quesada, Mamathamba Kalishettyhalli Mahadevaiah, Tommaso Rizzi, Jianan Wen, Markus Ulbricht, Milos Krstic, Christian Wenger, and Eduardo Perez

***Abstract*—Resistive random access memory (RRAM)-based hardware accelerators are playing an important role in the implementation of in-memory computing (IMC) systems for artificial intelligence applications. The latter heavily rely on vector-matrix multiplication (VMM) operations that can be efficiently boosted by RRAM devices. However, the stochastic nature of the RRAM technology is still challenging real hardware implementations. To study the accuracy degradation of consecutive VMM operations, in this work we programed two RRAM subarrays composed of 8 $\times$ 8 one-transistor-one-resistor (1T1R) cells following two different distributions of conductive levels. We analyze their robustness against 1000 identical consecutive VMM operations and monitor the inherent devices' nonidealities along the test. We finally quantize the accuracy loss of the operations in the digital domain and consider the trade-offs between linearly distributing the resistive states of the RRAM cells and their robustness against nonidealities for future implementation of IMC hardware systems.**

***Index Terms*—In-memory computing (IMC), multilevel, resistive random access memory (RRAM), vector-matrix multiplication (VMM).**

I. Introduction

TACKLING the "von Neumann bottleneck" limitation in conventional computing architectures is one of the biggest challenges that the computer science community is facing nowadays [1]. This problem arises when a substantial amount of computational resources are usually invested into data transportation between the memory units and processing units, commonly physically separated. Systems running applications based on deep learning, like image classification, object detection, biometric pattern recognition, etc., are steadily more limited in terms of power consumption and latency due to the constant data exchange traffic between the processing and memory units. These applications usually collect significant amounts of data ready to be processed in deep neural networks (DNNs) and in addition, impose design constraints such as low power consumption and low latency, constraints that traditional architectures are struggling to meet. The von Neumann limitation indeed hampers the implementation of such accurate and fast, yet low-energy demanding systems for artificial intelligence purposes. Although increasing separately the performance of both processing and storage units seems to be a short-term solution, it is not actually alleviating the bottleneck in the long run [2], [3]. New computer paradigms such as in-memory computing (IMC) assisted by emerging nonvolatile memories (NVMs) are becoming potential solutions to overcome the traditional architectures' limitations [4], [5], [6]. IMC brings the possibility to perform computing operations in situ, that is directly within memory subarrays avoiding data exchange between processing and storage units. IMC architectures become especially attractive in artificial neural network (ANN) applications where most of the workload of the inference phase relies on multiply-accumulate (MAC) operations, more precisely in vector-matrix multiplications (VMMs). In this context, NVMs such as resistive random access memories (RRAMs), phase change memories (PCMs), magnetoresistive random access memories (MRAMs) and ferroelectric random access memories (FeRAMs) [7], [8], [9], [10] can boost the performance and efficiency of MAC operations computed directly within the memory units, utilizing basic circuit laws like Ohm's law and Kirchhoff's law [11], [12]. A simplified example is exposed in Fig. 1, where the matrix elements correspond to the conductance values of four RRAM cells programed in four conductive levels (a, b, c, and d) and the two elements of the input vector are represented by V_1 and V_2 voltages. The operation results in a two-element output current vector represented by I_1 and I_2.

Manuscript received 11 January 2023; revised 2 February 2023; accepted 7 February 2023. Date of publication 22 February 2023; date of current version 24 March 2023. This work was supported in part by the Deutsche Forschungsgemeinschaft (German Research Foundation) under Project 434434223-SFB1461; and in part by the Federal Ministry of Education and Research of Germany under Grant 16ES1002, Grant 16FMD01K, Grant 16FMD02, Grant 16FMD03, and Grant 16ME0092. The review of this article was arranged by Editor J. Kang. *(Corresponding author: Emilio Perez-Bosch Quesada.)*

Emilio Perez-Bosch Quesada, Mamathamba Kalishettyhalli Mahadevaiah, Tommaso Rizzi, Jianan Wen, Markus Ulbricht, and Eduardo Perez are with the IHP-Leibniz-Institut fuer innovative Mikroelektronik, 15230 Frankfurt (Oder), Germany (e-mail: quesada@ihp-microelectronics.com).

Milos Krstic is with the IHP-Leibniz-Institut fuer innovative Mikroelektronik, 15230 Frankfurt (Oder), Germany, and also with the Institute for Informatics and Computational Science, University of Potsdam, 14476 Potsdam, Germany (e-mail: krstic@ihp-microelectronics.com).

Christian Wenger is with the IHP-Leibniz-Institut fuer innovative Mikroelektronik, 15230 Frankfurt (Oder), Germany, and also with BTU Cottbus-Senftenberg, 01968 Cottbus, Germany (e-mail: wenger@ihp-microelectronics.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TED.2023.3244509.

Digital Object Identifier 10.1109/TED.2023.3244509

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

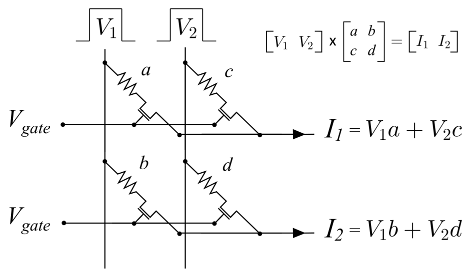

Fig. 1. MAC operation example performed in a memory subarray composed by a 2 × 2 1T1R matrix. Following Ohm's law (individual current flowing through the memristors) and Kirchhoff's current law (summation of the individual currents flowing into the nodes) a simple VMM operation is performed in situ.

Recently, the research community has intensively studied and optimized the performance of the RRAM technology within in-memory MAC-based systems/circuits, specially in simulation environments for early-stage design exploration [13], [14], [15], [16]. Among others, Mehonic et al. [17] reported hand-written digit recognition with up to 97% accuracy ratio using an RRAM-based ANN. However, to our best knowledge, the overall assessment of the RRAM MAC operation in a simulation environment is not yet fully performed and includes several simplifications due to the limits of the underlying hardware model [18], [19], [20], [21], [22]. Recently, Bengel et al. [23] experimentally analyzed the impact of binary RRAM nonidealities in VMM operations highlighting the fact that the low resistive state (LRS) variability plays a major role compared to the high resistive state (HRS) variability when performing MAC operations. The results were supported by circuit-level simulations using the physics-based Jülich Aachen Resistive Switching Tools-valence change mechanism (JART VCM) model considering ZrO_2/Ta-based devices [24].

Our work experimentally assesses the accuracy loss of 1000 identical consecutive VMM operations due to the resistive state degradation of HfO-based RRAM devices integrated into 4-kbit arrays, both in the digital and analog domain. The trade offs of linearly distributing the multiple resistive states of the cells will be discussed taking into account two different programming schemes (linear and quasilinear). To properly disseminate the effects of RRAM nonidealities over the VMM operations, we consider 8 × 8 matrices embedded in the 4-kbit arrays.

II. Experimental Methodology

The RRAM devices under study are fabricated as a metal-insulator-metal (MIM) structure located on the metal line 2 of the CMOS process and it consists of a TiN/Al:HfO_2/Ti/TiN stack. The bottom and top TiN layers were deposited by magnetron sputtering with a thickness of 150 nm. The 7 nm Ti layer acts as an oxygen scavenging layer that enables the resistive switching properties of the insulator layer. The Al-doped HfO_2 layer was grown with a thickness of 6 nm and an Al content of about 10% by atomic layer deposition (ALD). The MIM stacks were patterned with an area of about 0.4 μm^2. Their resistive state can be electrically controlled by the creation or disruption of conductive filaments (CFs) consisting of oxygen vacancies within the insulator layer.

To protect the RRAM device against current overshoots during the programming operations, it is built in series to a nMOS transistor fabricated in the 250 nm CMOS technology, resulting in a one-transistor-one-resistor (1T1R) structure. Its main functionality is to limit the current through the cell by setting a compliance current controlled by the voltage applied to its gate terminal (V_{gate}). Such 1T1R structures are embedded into 4-kbit memory arrays manufactured in a 64 × 64 manner, where the transistor also acts as selector device preventing the so-called sneak path currents among adjacent cells. Moreover, thanks to the tuning capability of V_{gate}, multiple intermediate resistive states can be achieved by means of the multilevel-cell (MLC) approach. Setting the transistor's V_{gate} in certain values during the Set operation can move the individual RRAM cells into different LRSs. The reader is referred to [25] for further details concerning the array structure. To fine-tune the process, the multilevel incremental step pulse with verify algorithm (M-ISPVA) [26] was chosen to set the devices to the desired resistive states. With this algorithm, we can target at least 4 resistive levels with excellent switching properties that can be efficiently arranged following different distributions.

To enable multi-level behavior of the RRAM cells in an optimum way, we followed the methodology proposed in [27]. In this work, two sets of 64 samples (8 × 8) were programed by following two different distributions of resistive states: the "quasilinear" distribution (already explored in the mentioned article) and the "linear" distribution of just LRSs. The latter distribution targets four linearly spaced LRSs in which the HRS is only a "pivot" state to transit between them and it is not considered a valid state for the VMM operations. Distributing linearly the resistive states programed on the individual RRAM cells loosens up the analog-to-digital converter (ADC) parametrization constraints regarding the output currents detection and conversion to the digital domain. On the other hand, the former considers three linearly spaced LRSs and a single nonlinearly spaced HRS as valid states.

Regarding both distributions, Table I indicates the target current (I_{trg}) and V_{gate} parameters chosen to program the devices to the various resistive levels using the M-ISPVA. To illustrate both distributions of resistive states, we experimentally programed 128 RRAM devices into all the resistive levels shown in Table I and represented the cumulative distribution functions (cdfs) associated with their read-out currents in Fig. 2. Observing the cdf steepness of both distributions of the resistive states, one can appreciate that programming the LRSs with slightly higher V_{gate} values, which imposes higher compliance limits during the set operations, leads to broader cdfs for the quasilinear distribution. On the contrary, the lower V_{gate} values used in the linear distribution allow to target more accurately the desired current levels (I_{trg}) at expense of making them more vulnerable to the RRAM nonidealities, as it will be demonstrated below. For further details concerning the programming phase of the samples and the specific electrical parameters described in Table I, the interested reader is referred to [26] and [27].

In order to reproduce the theoretical VMM scenario represented in Fig. 3, first of all, the devices under study have to be switched to the resistive states illustrated with different colors.

TABLE I
MAIN M-ISPVA PROGRAMMING PARAMETERS USED TO PROGRAM THE FOUR CONDUCTIVE LEVELS OF QUASILINEAR AND LINEAR DISTRIBUTIONS, RESPECTIVELY

	Quasilinear		Linear	
Sate	I_{trg} (μA)	V_{gate} (V)	I_{trg} (μA)	V_{gate} (V)
HRS	5	2.7	5	2.7
LRS0	-	-	10	0.9
LRS1	20	1.2	20	1.1
LRS2	30	1.4	30	1.3
LRS3	40	1.6	40	1.5

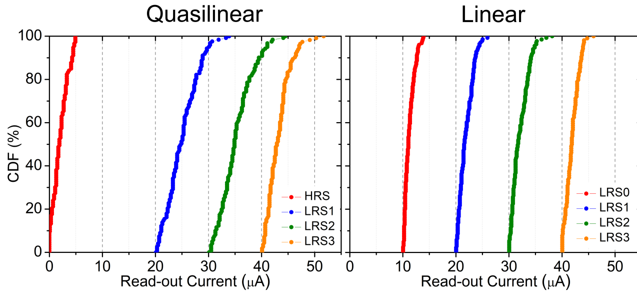

Fig. 2. CDFs of the read-out currents experimentally measured for the quasilinear (three LRSs and one HRS) and linear (four LRSs) distributions.

The allowed voltage amplitudes that feature the input vector of the multiplication are 0.125, 0.25, 0.375, and 0.5 V, in order to minimize the read-disturb phenomenon [28]. The elements of the input vector and the resistive matrix were randomly selected using a standard uniform distribution. Specific matrix configurations resembling concrete weight distributions within trained DNNs will follow in future work. The corresponding theoretical output vectors for the two distributions are displayed in Fig. 3 as well. Every output current element is theoretically calculated by summing the ideal read-out currents of each RRAM device located in the corresponding row of the matrix at a certain voltage (dictated by the input vector). In this way, output element number i is the result of the summation of the read-out currents of row number i when the read-out voltages of the input vector are applied to the corresponding RRAM devices within the mentioned row. This methodology applies to all the output elements of the VMM.

A total number of 1000 identical VMM operations are executed over every subarray. To monitor the evolution of the resistive state of every cell, we perform individual read-out operations ($V_{TE} = 0.2$ V and $V_{gate} = 1.7$ V) to all the RRAM cells within the subarrays right after every VMM operation. The whole process (programming plus VMM computation and monitoring) was executed using an in-house made set-up composed by a Keithley 4200A-SCS semiconductor parameter analyzer and an Arduino Mega, both coordinated using a LabVIEW virtual instrument hosted in a PC (see Fig. 4). The addressing of the individual 1T1R cells is performed using 13 digital signals generated by the Arduino Mega. On the other hand, all the analog signals required to perform M-ISPVA and the VMM operations over the cells, e.g., V_{TE} and V_{gate}, are generated by the pulse measurement unit (PMU) integrated into the Keithley system. Such PMU is able

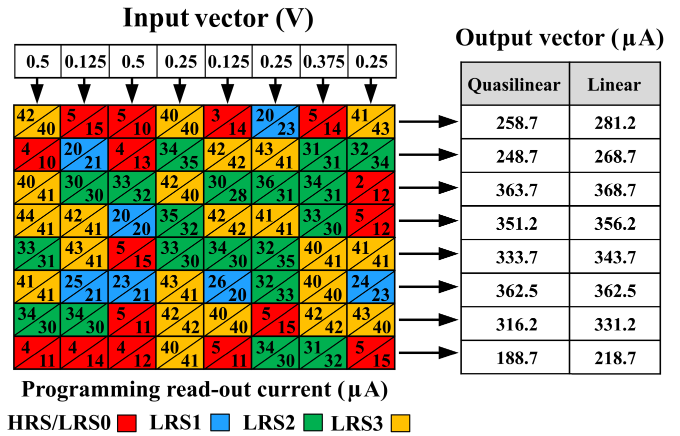

Fig. 3. Proposed VMM scenario where the output vector (right) contains the expected current results for both resistive states distributions considering the input voltage vector (top) and the ideal current values for each state indicated in Table I. Within the resistive matrix, it is indicated the read-out currents measured after programming considering the quasilinear (top-left corner) and the linear (bottom-right corner) distributions. The numeric values were rounded to no decimals for simplicity.

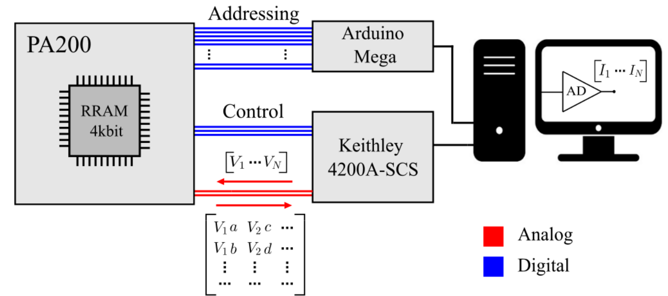

Fig. 4. Schematic of the in-house made set-up designed to program and execute VMM operations over 1T1R samples embedded in 4-kbit arrays. To do so, 16 digital plus two analog signals are generated as described by the blue and red colors, respectively. The red arrows make reference to the voltage and current elements described in Fig. 3. Among the "control" signals, V_{gate} is considered.

to generate arbitrary voltage waveforms and simultaneously measure current. The electrical signals feature a pulsewidth of 10 μs. By means of three additional source measurement units (SMUs), Keithley also generates three digital control signals to enable the peripheral circuitry of the 4-kbit array to perform the required operations. The analog read-out values measured in each individual cell are sent to the control PC where the computation of the output vector is performed in software. Regarding the conversion of the analog output currents to digital values, the ADC parametrization was carried out following the methodology described in [29] considering a linear distribution of the resistive states of the matrix elements. Thus, we defined a 7-bit ADC with a dynamic range of 50–800 μA with 6.25 μA interval length. In order to avoid masking the RRAM variability influence over the results, the ADC is considered to be ideal and thus, its parasitic effects are neglected.

III. RESISTANCE MONITORING DURING VMM OPERATIONS

The RRAM cells involved in the computation of the VMM operations may suffer from the device's nonidealities which

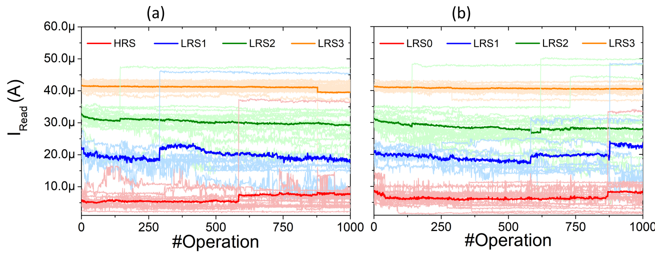

Fig. 5. Read-out currents measured at $V_{TE} = 0.2$ V and $V_{gate} = 1.7$ V over both subarrays under test right after every consecutive VMM operation. The lighter lines denote the individual read-out currents of every RRAM cell programed in their respective level, whereas the darker ones represent the averaged values over each resistive level. (a) Quasilinear. (b) Linear.

lead to fluctuations in their resistive values. Such nonidealities are a big concern in the state of the art of this technology and therefore, this study focuses on the impact of the following phenomena over consecutive VMM operations: read-out noise [30], [31], conductance relaxation [11], [32], conductance drift caused by read disturb effects [28], [31] and programming errors [26], [27]. Although intensive work has been devoted to reduce the device-to-device (DTD) variability of the RRAM cells at the device and algorithm levels [33], it also impacts the accuracy of the RRAM operations. Nevertheless, the previously mentioned nonidealities may mask the influence of the D2D variability over the VMM results.

Considering the MLC approach and its implementation in VMM operations, the robustness of the RRAM's conductive states against the nonidealities plays a major role in the conservation of the accuracy during the execution of a large number of consecutive VMM operations. As previously mentioned, right after the execution of every VMM operation we performed standard read-out operations to monitor the resistance fluctuations of the RRAM cells involved. Fig. 5 depicts the read-out current measured in every cell programed.

At a first sight, Fig. 5 clearly exposes the superior robustness of LRS3 against fluctuations during the execution of VMM operations compared to the rest of the resistive states. Followed by HRS in the quasilinear distribution, the most resistive state also denotes larger robustness against the consecutive VMM operations compared to the rest of the intermediate states, namely LRS0, LRS1, and LRS2. This is denoted by the more stable average LRS3 and HRS lines depicted in the figure, compared to those obtained from the intermediate levels. In our study, the later levels tend to suffer more from the device's nonidealities e.g., undesired resistance drifts and state relaxations, in agreement with [23] and [34]. Hence they are more prone to cause accuracy degradation in the output vector calculations. This phenomenon is especially visible in LRS1. This behavior is mainly caused by the degradation of the CFs within specific cells due to oxygen ions/vacancies recombination [35] along the execution of consecutive operations, leading to variations in their resistive state. Moreover, read-disturb effects may produce additional resistance variations. VMM operations performed with input voltages above 0.2 V induce the so-called "read stress" to certain cells which may fall into undesired oxygen vacancies movement in the filament and thus, modification of the resistive state of the cells [28], [31]. Commonly, this effect can gradually affect the resistive state of the cells (see a gradual current reduction of the average values in Fig. 5) or in more rare cases, can suddenly drift their resistive state toward LRS3 or even more conductive states, as it can be appreciated in Fig. 5. Devices whose resistive state suddenly drifts toward more conductive levels can be easily spotted as step-like light-colored lines in Fig. 5. As we will observe in the following analysis, these sudden drifts induce larger errors in the VMM calculations as a consequence of the higher resistive state fluctuation. Despite the sporadic nature of this behavior, we could observe that single RRAMs programed in LRS1 and LRS2 are more susceptible to suddenly drift toward more conductive levels, usually after 500 to 600 VMM operations. To minimize the impact of read-disturb effects over the VMM operations, lower input voltage levels will be tested in future work.

Read-out noise is the result of low current variations (of about $\pm 2\%$) measured on the involved RRAM cells. This effect is also amplified when applying read-out voltages above 0.2 V. Samples programed into intermediate levels are once again generally more vulnerable to such fluctuations [31]. However, this phenomenon is observable in all the samples to a certain extent and also influences the final VMM operations results. Additional contributions to the read-out noise might accentuate its impact into the final results, such as noise derived from imperfections in the measurement equipment and array's peripheral circuitry among others. This phenomenon is observed in Fig. 5 as "burst noise" in the consecutive read-out values.

Further, errors during the programming phase of the RRAM cells induce accuracy issues in the execution of VMM operations. Although a specific I_{trg} is defined for each resistive level in the execution of the programming algorithm (see Table I-I_{trg}), the distribution of the resistive levels and their intrinsic variability influence the achieved level. This final resistive state may vary among RRAM cells and also differ from the intended value to a certain degree. The linear scheme demonstrated a lower programming error (narrower cdf curves) compared to the quasilinear one (see Fig. 2). Nonetheless, the RRAM cells programed following a linear scheme tend to gradually relax their resistance levels toward more resistive values throughout the execution of the VMM operations. Therefore, they tend to settle their resistive value below their threshold currents [see Fig. 5(b)]. On the other hand, those cells programed following a quasilinear scheme, on average they keep steady within their I_{th}. The most acute case is the one observed in LRS0 (linear distribution), where most of the samples' states relax toward HRS after a few VMM operations, settling down their average read-out current at around 6 μA.

IV. Nonidealities Impact on VMM Operations

The RRAM nonidealities listed above have different impacts on the resulting output vector computation. The resistive value of the RRAM cells fluctuates throughout the execution of consecutive VMM operations, inducing errors in the output current, later on, translated into digit errors during the AD conversion. For the purpose of simplification, Fig. 6 exposes

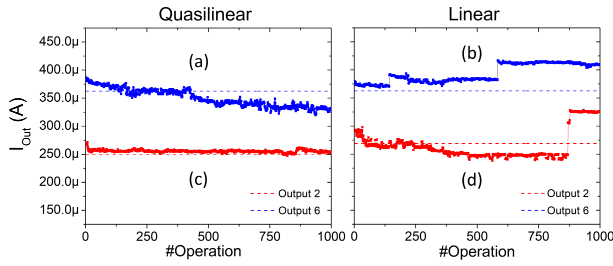

Fig. 6. Current evolution of outputs number 2 and 6 throughout 1000 VMM consecutive operations considering the quasilinear and linear distributions. The dashed lines represent the ideal calculated values of the respective output. Different predominant fluctuations are associated with different current outputs being (a) gradual degradation of the output current level, (b) sudden increase of the output current, (c) noise-like output fluctuations, and (d) combination of various types of fluctuations.

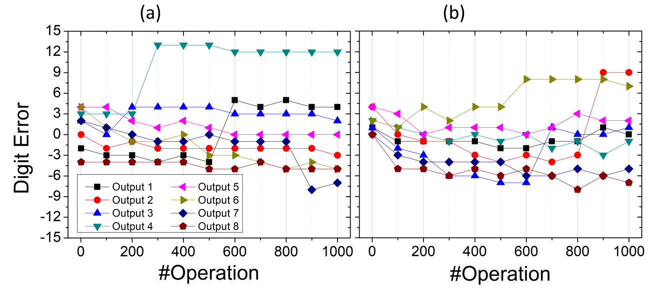

Fig. 7. Digit error computed after the AD conversion for (a) quasilinear and (b) linear distributions.

the current evolution of two out of eight output elements within both subarrays prior to the AD conversion along 1000 operations, where the dashed lines denote the theoretical current level expected for each output element (see Fig. 3). We can observe that the main current fluctuations measured among all output elements within the two subarrays follow similar behaviors as those exposed Fig. 6.

In Fig. 6(a), the predominant fluctuation gradually degrades the output current level progressively moving the experimental value below the theoretical level. This can be mainly attributed to the resistance relaxation of the RRAM cells involved in the calculation of certain outputs. In Fig. 6(b), the dominant fluctuation implies a sudden increase of the current value during specific VMM operations, which occurs due to immediate resistance drifts of specific RRAM cells toward more conductive levels due to read disturb effects. As explained before, these drifts may happen more frequently after 600 VMM operations and they induce abrupt output variations. Fig. 6(c) closely reproduces the expected theoretical values although we can observe a constant current noise along the execution of the operations. Such noise is the result of the accumulation of individual read-out noise effects among the involved RRAM cells. All the output elements are affected by this fluctuation. In the worst case scenario, the consecutive computations of certain output elements are affected by a combination of several of the previously mentioned current fluctuations. This can be observed in Fig. 6(d). As expected, these combinations degrade the accuracy of the operations in a more significant way.

Going one step further, we studied the accuracy impact on the digital domain of the current variations described above. To do so, we calculated the digit conversion error taking into account the theoretical and experimental values obtained at the output of the 7-bit ADC described in Section II. Fig. 7 represents the digit error computed for every output element considering both current distributions of the resistive states. To have a clearer view of the results, we only displayed the computed error every 100 operations, being the negative (positive) errors the result of converted digits below (above) the expected digital value. It can be observed that both subarrays present an initial digit error during the first VMM operation. In absolute terms, such error is higher in the quasilinear case. This may be the result of two factors: first of all, the mismatch between the quasilinear distribution of the resistive levels and the linear ADC parametrization; second, the higher programming error reported by this distribution compared to the linear one. Considering further VMM operations, on average the RRAM cells programed following a linear scheme denote a gradual degradation of their resistive state toward less conductive values, as concluded in Section III. This is translated into a gradual reduction of the output current level and thus, a predominant growth of the negative digit error. Consequently, we also observe a reduction of the positive error [see Fig. 7(b)]. After approximately 200 operations, the measured error is mainly negative for most of the output elements. On the contrary, the subarray programed following a quasilinear distribution [see Fig. 7(a)] reports relatively constant digit error along the consecutive operations compared to its counterpart. This is due mainly to its higher resilience against read-disturbance effects. Ultimately, the degradation of the intermediate resistance levels after 100 VMM operations implies a gradual growth of the digit error after the AD conversion. Particularly, most samples programed to LRS0 tend to be driven toward HRS during the consecutive VMM operations, which vanishes the main advantage of using a linear distribution to reduce the ADC parametrization constraints. More robust programming techniques are required to ensure the stability of this specific resistive state. Finally, for both distributions, the dominant error impact is always measured when the analog output suddenly increases due to individual acute RRAM drifts [see Fig. 6(b) and (d)]. According to this study, these are the most harmful nonidealities in terms of digit error thus, partial or complete re-programming of the resistive matrix might be required to solve this issue. Moreover, He et al. [34] proposed to reduce the number of intermediate states if, previously during the training period of the ANN, a magnification factor is introduced. Thus, the weights are pushed toward higher absolute values and the resultant resistance matrix would be more polarized into extreme and more stable resistive states, such as LRS3 and HRS.

V. Conclusion

Within the framework of RRAM-based MAC calculations, VMM operations performed by using linearly spaced resistive states of the RRAM cells may alleviate critical constraints such

as ADC design parameters. In addition, the linear distribution presents a lower programming error with respect to a quasilinear scheme. However, the former shows a higher vulnerability to resistive state fluctuations, which degenerates the accuracy of the results after approximately 200 VMM operations. Thus, linear distributed resistive states of the RRAM cells may be a good choice when high accuracy of the operations and high refreshment rate of the resistive states is required (e.g., in situ training of ANNs). On the other hand, the quasilinear scheme demonstrated higher robustness against nonidealities among consecutive VMM operations in terms of digit error of the AD conversion phase. This work serves as an initial step toward the implementation and optimization of larger array sizes in RRAM-based VMM operations in a hardware fashion. Thus, the impact of larger matrix sizes (e.g., 16 × 16) will be explored in future work.

References

[1] F. Zahoor, T. Z. A. Zulkifli, and F. A. Khanday, "Resistive random access memory (RRAM): An overview of materials, switching mechanism, performance, multilevel cell (MLC) storage, modeling, and applications," *Nanosc. Res. Lett.*, vol. 15, no. 1, pp. 1–26, Dec. 2020.

[2] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 10–14.

[3] F. Staudigl, F. Merchant, and R. Leupers, "A survey of neuromorphic computing-in-memory: Architectures, simulators, and security," *IEEE Design Test*, vol. 39, no. 2, pp. 90–99, Apr. 2022.

[4] D. Ielmini and H. S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electron.*, vol. 1, no. 6, pp. 333–343, 2018.

[5] E. Miranda and J. Suñé, "Memristors for neuromorphic circuits and artificial intelligence applications," *Materials*, vol. 13, no. 4, p. 938, Feb. 2020.

[6] W. Zhang et al., "Neuro-inspired computing chips," *Nature Electron.*, vol. 3, no. 7, pp. 371–382, 2020.

[7] D. Kuzum, S. Yu, and H.-S. P. Wong, "Synaptic electronics: Materials, devices and applications," *Nanotechnology*, vol. 24, no. 38, Sep. 2013, Art. no. 382001.

[8] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proc. IEEE*, vol. 106, no. 2, pp. 260–285, Feb. 2018.

[9] A. Sebastian, M. L. Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnol.*, vol. 15, no. 7, pp. 529–544, 2020.

[10] Z. Wang et al., "Resistive switching materials for information processing," *Nature Rev. Mater.*, vol. 5, no. 3, pp. 173–195, 2020.

[11] S. Yu, H. Jiang, S. Huang, X. Peng, and A. Lu, "Compute-in-memory chips for deep learning: Recent trends and prospects," *IEEE Circuits Syst. Mag.*, vol. 21, no. 3, pp. 31–56, 3rd Quart., 2021.

[12] A. Singh et al., "Low-power memristor-based computing for edge-AI applications," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[13] T. Rizzi, E. P.-B. Quesada, C. Wenger, C. Zambelli, and D. Bertozzi, "Comparative analysis and optimization of the SystemC-AMS analog simulation efficiency of resistive crossbar arrays," in *Proc. 36th Conf. Design Circuits Integr. Syst. (DCIS)*, Nov. 2021, pp. 1–6.

[14] M. Bavandpour, S. Sahay, M. R. Mahmoodi, and D. Strukov, "Efficient mixed-signal neurocomputing via successive integration and rescaling," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 3, pp. 823–827, Mar. 2020.

[15] S. Pechmann et al., "A low-power RRAM memory block for embedded, multi-level weight and bias storage in artificial neural networks," *Micromachines*, vol. 12, no. 11, p. 1277, Oct. 2021.

[16] A. J. Perez-Avila, E. Perez, J. B. Roldan, C. Wenger, and F. Jimenez-Molinos, "Multilevel memristor based matrix-vector multiplication: Influence of the discretization method," in *Proc. 13th Spanish Conf. Electron Devices (CDE)*, Jun. 2021, pp. 66–69.

[17] A. Mehonic, D. Joksas, W. H. Ng, M. Buckwell, and A. J. Kenyon, "Simulation of inference accuracy using realistic RRAM devices," *Frontiers Neurosci.*, vol. 13, p. 593, Jun. 2019.

[18] W. Wan et al., "33.1 A 74 TMACS/W CMOS-RRAM neurosynaptic core with dynamically reconfigurable dataflow and in-situ transposable weights for probabilistic graphical models," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 498–500.

[19] C. Xue et al., "Embedded 1-mb ReRam-based computing-in-memory macro with multibit input and weight for CNN-based AI edge processors," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 203–215, Jan. 2020.

[20] P. Yao et al., "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, 2020.

[21] S. K. Kingra et al., "Methodology for realizing VMM with binary RRAM arrays: Experimental demonstration of binarized-ADALINE using OxRAM crossbar," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.

[22] S. Shchanikov et al., "Fault tolerance of memristor-based perceptron network for neural interface," *BioNanoScience*, vol. 11, no. 1, pp. 84–90, Mar. 2021.

[23] C. Bengel et al., "Reliability aspects of binary vector-matrix-multiplications using ReRAM devices," *Neuromorphic Comput. Eng.*, vol. 2, no. 3, Sep. 2022, Art. no. 034001.

[24] C. Bengel et al., "Variability-aware modeling of filamentary oxide-based bipolar resistive switching cells using SPICE level compact models," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4618–4630, Dec. 2020.

[25] A. Grossi et al., "An automated test equipment for characterization of emerging MRAM and RRAM arrays," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 2, pp. 269–277, Apr. 2018.

[26] E. Perez, C. Zambelli, M. K. Mahadevaiah, P. Olivo, and C. Wenger, "Toward reliable multi-level operation in RRAM arrays: Improving post-algorithm stability and assessing endurance/data retention," *IEEE J. Electron Devices Soc.*, vol. 7, pp. 740–747, 2019.

[27] E. Pérez et al., "Optimization of multi-level operation in RRAM arrays for in-memory computing," *Electronics*, vol. 10, no. 9, p. 1084, May 2021.

[28] W. Shim, Y. Luo, J. S. Seo, and S. Yu, "Investigation of read disturb and bipolar read scheme on multilevel RRAM-based deep learning inference engine," *IEEE Trans. Electron Devices*, vol. 67, no. 6, pp. 2318–2323, Apr. 2020.

[29] A. Shafiee et al., "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *ACM SIGARCH Comput. Archit. News*, vol. 44, no. 3, pp. 14–26, 2016.

[30] N. Raghavan et al., "Microscopic origin of random telegraph noise fluctuations in aggressively scaled RRAM and its impact on read disturb variability," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2013, p. 5E.

[31] W. Shim, J.-S. Seo, and S. Yu, "Two-step write–verify scheme and impact of the read noise in multilevel RRAM-based inference engine," *Semicond. Sci. Technol.*, vol. 35, no. 11, Nov. 2020, Art. no. 115026.

[32] M. Lanza et al., "Standards for the characterization of endurance in resistive switching devices," *ACS Nano*, vol. 15, no. 11, pp. 17214–17231, Nov. 2021.

[33] E. Perez, M. K. Mahadevaiah, E. P.-B. Quesada, and C. Wenger, "Variability and energy consumption tradeoffs in multilevel programming of RRAM arrays," *IEEE Trans. Electron Devices*, vol. 68, no. 6, pp. 2693–2698, Jun. 2021.

[34] W. He et al., "Characterization and mitigation of relaxation effects on multi-level RRAM based in-memory computing," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Mar. 2021, pp. 1–7.

[35] Y. Lin, "Performance impacts of analog ReRAM non-ideality on neuromorphic computing," *IEEE Trans. Electron Devices*, vol. 66, no. 3, pp. 1289–1295, Mar. 2019.