

A Data-Driven Stochastic Memristor Model for Integrated Circuit Simulation

Lijie Xie, Jiawei Shen, Peilong Feng, Andrea Mifsud, Adil Malik, Amir Nassibi, Vichaya Manatchinapisit, Christos Papavassiliou

Department of Electrical and Electronic Engineering, Imperial College London, UK

Abstract—Memristors have emerged as promising candidates for multi-level data storage, in-memory processing, and neural networks since their intrinsic programmability of resistance states under applied stimuli has been well revealed in memristor modelling. However, the programming uncertainty arising from the inherently stochastic nature of the device itself has been overlooked in previous modelling approaches. This omission hinders the incorporation of memristor stochasticity into time-domain circuit simulation. To address this issue, we propose a behaviour model that incorporates real-time programming stochasticity. Our model stands out for several attributes: 1) programming stochasticity is included and exhibited in its resistance change over time; 2) Stochasticity occurring at time steps (or step-to-step stochasticity) is depicted by an optimised stochastic differential equation (SDE); 3) The SDE is determined by sufficient characterisation data of our in-house TiO_2 devices in a data-driven method. Consequently, our model is validated as highly matched to the characterised memristor device in terms of time-domain resistance evolution. Additionally, the modelling process can be adapted to different memristors with significant device variations. Furthermore, the model is transformed into the Verilog-A style for in-circuit simulation. To demonstrate its compatibility with system-level circuit simulation, a mixed-signal CMOS circuit is designed. This circuit explores the feasibility of storing multi-bit data within a single memristor, whilst also considering its programming stochasticity.

Index Terms—memristor, stochasticity, data-driven, SDE, Verilog-A model, multi-bit memory, circuit simulation

I. INTRODUCTION

Known for its intrinsic programmability and non-volatility, the memristor is emerging as a potential candidate for multi-level storage [1], [2] and information processing [3]. The behaviour of reversible resistance programming under the excitation of stimuli has been extensively studied in memristor modelling. Early-stage physical models [4]–[9] investigate the physical process involved in dynamic switching, primarily based on the hysteresis I-V characteristic under periodic inputs. Generalised memristor models [10]–[14] are applicable to a wider range of devices stemming from various technologies and more extensive input schemes, such as DC sweeps or pulse trains [15]–[17]. Moreover, motivated by the compatibility of memristors with CMOS foundries, memristor models have been adapted to SPICE simulators [18]–[20] to address the need for time-domain circuit simulation.

However, programming stochasticity, stemming from the stochastic nature of filament formation/rupture during memristor programming and resulting in uncertain memristive effects [21], is excessively ignored from the modelling perspec-

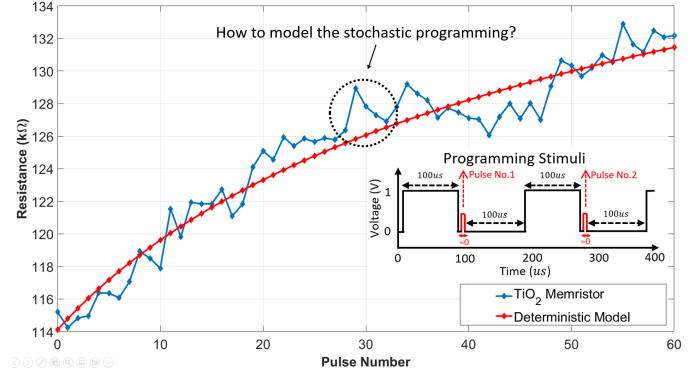


Fig. 1: The discrepancy between the measured TiO_2 memristor data and its deterministic model. The deterministic model simplifies the resistance trajectory as a smooth curve, ignoring the step-to-step stochasticity. Our model aims to incorporate this stochasticity. Inset: input voltage pulses. Each high-amplitude program pulse is followed by a low-amplitude read pulse. In this paper, all plotted resistances are acquired through read pulses ($R = \frac{v_{read}}{i_{read}}$), with the x-axis denoting the number of read pulses.

tive. The stochastic behaviour of temporal memristor evolution responding to pulsed stimuli has been widely observed in previous research [3], [7], [15], [16], [22]–[24] and our work. One such example depicted in Fig. 1 is the resistance trajectory of our in-house TiO_2 memristor (blue) [25] under consecutive pulsed stimuli. This trajectory increases with significant fluctuations during operation due to stochasticity occurring at each pulsing step. However, conventional deterministic models are usually presented as a smooth fitting curve (red), a monotonically increasing curve that reaches a higher intermediate state at each subsequent programming step. Since programming stochasticity is the intrinsic property of memristor devices and thus cannot be overlooked in practical implementations, deterministic models are not convincing in specific practical applications, such as the multi-bit data storage [26]–[31], where programming stochasticity greatly limits their multi-state capability. Therefore, it is imperative to incorporate programming stochasticity into memristor modelling to benefit circuit designs and system simulations.

In general, a memristor model without stochasticity denotes its state derivative over time as a function of the applied voltage $v(t)$ and the state $w(t)$ [4]:

$$\frac{dw(t)}{dt} = f(v(t), w(t)) \quad (1)$$

As Fig. 1 shows that the stochastic resistance trajectory behaves as a directionless random fluctuation added to a de-

terminated increment directed by external stimuli, Eq. 1 can be replaced by a stochastic differential equation (SDE) to depict the stochastic evolution of resistance [32]. The SDE consists of a deterministic term and a stochastic term, presented as follows:

$$\frac{dw(t)}{dt} = \underbrace{F(w(t), v(t))}_{\text{deterministic term}} + \underbrace{g(w(t), v(t))\xi(t)}_{\text{stochastic term}} \quad (2)$$

In our work, the functions $F(w(t), v(t))$, $g(w(t), v(t))$, and the random sequence $\xi(t)$ are described by simple mathematical expressions determined by sufficient data characterised from the modelled memristor device.

The rest of this paper is organised as follows: Section II provides a brief overview of the evolution from published memristor models to our proposed model. Section III details the modelling process, including the underlying theory, the data-driven method, and the result. Also, it includes discussions on the model's generality across different devices and its comparison to other stochastic models. Section IV depicts the model's compatibility with complex circuit simulation. It demonstrates the application of the model in a mixed-signal CMOS system to assess the feasibility of storing multi-bit data within a single memristor while considering the programming stochasticity. Section V concludes this work.

II. EVOLUTION OF MEMRISTOR MODELS

This section briefly reviews the development of memristor models and establishes a link between the underlying theory of published models to our model's theory. Initially rooted in Chua's definition of memristors [4], early-stage models [5], [7]–[9] employed physical theories and complicated mathematics to elucidate the bipolar switching behaviours. These bipolar behaviours were subsequently approximated by simplified mathematics that are applicable to generalised devices [12], [13]. To achieve a more comprehensive and precise representation of memristor multi-level dynamics under pulsed stimuli, new behaviour models [14] emerged with the data-driven method. However, all the above models lacked the incorporation of programming stochasticity inherent in memristor devices. Therefore, stochastic compact models [32]–[44] were proposed. Some employed SDE to reproduce programming stochasticity in time steps but the comprehensive validation of their outcomes remained incomplete.

A. Basic definition

In general, the I-V characteristic of a voltage-controlled memristor device can be represented by a pair of coupled equations [4]:

$$\begin{aligned} i(t) &= G(w, v) \cdot v(t) \\ \frac{dw}{dt} &= f(w, v) \end{aligned} \quad (3)$$

where $i(t)$ represents the device current, $v(t)$ is the device voltage, $G(w, i)$ stands for the device conductance, and w is the internal state variable. The rate of change of w , denoted as $\frac{dw}{dt}$, primarily depends upon the applied voltage v and the

internal state w , which can be simply explained: 1) higher voltage has a greater capability to drive state changes, and 2) under a constant voltage, state changes slow down as they approach the boundary of the effective region.

B. Physical models

Physical models correlate switching mechanisms with established physical theories, although the underlying switching mechanisms are still a topic of debate. In these correlations, the internal state w can be physically represented as the dynamic length of the doped region in a dopant drift model [5], [8], the tunnel barrier width in tunnel barrier models [7], or the number of filaments in filamentary device models [9]. One notable model, the Simmons tunnel barrier model, applies the function of electric tunnel effect theory to memristor modelling [7]:

$$\frac{dw(t)}{dt} = \begin{cases} c_{off} \sinh\left(\frac{i}{i_{off}}\right) \exp\left[-\exp\left(\frac{w-a_{off}}{w_c} - \frac{|i|}{b}\right) - \frac{w}{w_c}\right], & i > 0 \\ c_{on} \sinh\left(\frac{i}{i_{on}}\right) \exp\left[-\exp\left(\frac{w-a_{on}}{w_c} - \frac{|i|}{b}\right) - \frac{w}{w_c}\right], & i < 0 \end{cases} \quad (4)$$

where all except w and i are fitting parameters derived from experiments. However, this formula can be challenging to adapt to various memristor devices and computationally expensive in circuit simulators due to its mathematical complexity.

C. Generalised models

Generalised models [12], [13] offer flexibility in fitting various memristor devices with high accuracy. In [13], the I-V property is provided below:

$$I(t) = \begin{cases} a_1 w(t) \sinh(bV(t)), & V(t) \geq 0 \\ a_2 w(t) \sinh(bV(t)), & V(t) < 0 \end{cases} \quad (5)$$

To simplify the mathematics, $\frac{dw}{dt}$ is assumed as the product of $g(v)$ and $f(w)$:

$$\frac{dw}{dt} = g(v) \cdot f(w) \quad (6)$$

where $g(v)$ introduces the non-linear voltage dependence, and $f(w)$ is the window function, expressed in various algebraic forms to match the model to experimental data. Here, both $g(v)$ and $f(w)$ can be depicted in exponential forms is given:

$$\begin{aligned} g(v(t)) &= \begin{cases} A_p (e^{v(t)} - e^{V_p}), & v(t) > V_p \\ -A_n (e^{-v(t)} - e^{V_n}), & v(t) < -V_n \\ 0, & -V_n \leq v(t) \leq V_p \end{cases} \\ f(w) &= \begin{cases} e^{-\alpha_p(w-w_p)} \cdot \left(\frac{1-w}{1-w_p}\right), & w \geq w_p \\ e^{\alpha_n(w+w_n-1)} \cdot \left(\frac{w}{1-w_n}\right), & w \leq 1-w_n \\ 1, & 1-w_n < w < w_p \end{cases} \end{aligned} \quad (7)$$

where $V_{p,n}$ are the positive/negative threshold voltage, $w_{p,n}$ are the state boundaries, and $A_{p,n}$ and $\alpha_{p,n}$ are fitting parameters. $g(v)$ defines the exponential dependence of $\frac{dw}{dt}$ on applied voltage amplitude when it surpasses the threshold; $f(w)$ describes an exponential state rate decay when the internal state w is beyond the boundary $w_{p,n}$.

However, these models are only characterised by hysteresis I-V plots under periodic inputs. It is quite off to validate their reproduced behaviours under pulsed stimuli.

D. Behaviour models

Behaviour models provide dynamic memristor responses that closely match measured data under DC sweeps or pulsed inputs. One data-driven behaviour model [14] is introduced here, featured for: 1) employing repeated squared voltage pulses with fixed periods to gather sufficient real-world data, 2) the internal state w is replaced by observable R which is updated at each pulsing step, and 3) assuming the state boundary of the effective region is voltage-dependent. The optimised state equation is as follows:

$$\frac{dR}{dt} = g(v) \cdot f(R, v) \quad (8)$$

$g(v)$, $f(R, v)$ both takes the exponential form:

$$g(v) = \begin{cases} A_p (e^{t_p |v|} - 1), & v > 0 \\ A_n (e^{t_n |v|} - 1), & v < 0 \\ \text{else} & 0 \end{cases} \quad (9)$$

$$f(R, v) = \begin{cases} e^{\eta k_p (r_p(v) - R)} - 1, & R < \eta \cdot r_p(v), & v > 0 \\ e^{\eta k_n (R - r_n(v))} - 1, & R > \eta \cdot r_n(v), & v < 0 \\ \text{else} & 0. \end{cases}$$

where $A_{p,n}$, $t_{p,n}$ and $k_{p,n}$ are fitting parameters. $r_p(v)$ and $r_n(v)$ are boundary functions fitted using the experiment data. η is either 1 or -1 depending on the polarity of resistance change to the applied bias.

E. Stochastic compact models

Stochastic compact models [32]–[44] consider memristor variations, including device-to-device variation resulting from fabrication processes, cycle-to-cycle variation occurring between repeated switching processes, and step-to-step (or pulse-to-pulse) variation attributed to time-evolution programming uncertainty. One widely adopted approach to quantify device-to-device variation or cycle-cycle variation is the Monte Carlo method, which extracts the statistical distribution of critical fitting parameters, as reported in [34], [35], [41]–[44]. However, it does not capture step-to-step variation observed as an uncertain resistance response to the applied voltage pulse. To depict the time-evolution trajectory of stochastic resistance changes under pulsed stimuli, a stochastic differential equation (SDE) has been employed [32], [36]. For example, [32] incorporates the temporal noise in its SDE and simulates the resistance fluctuation when external pulses heat the device.

However, these SDE models lack comprehensive validation in two crucial aspects: 1) Memristor behaviours can vary significantly across repetitive experiments, as can the behaviour of the stochastic model in repeated simulations. Validating such a model requires sufficient data from repetitive experiments and simulations. 2) The model's predictability under different programming scenarios, such as different voltages, remains unverified.

III. PROPOSED MODEL

A. Model theory

In generalised memristor modelling, the state variable $w(t)$, is employed in model equations to describe the hysteresis I-V property under periodical inputs. This is because the memristor resistance, $R(t)$, varies during the pulsing period and is thus inaccessible. However, as described in the above Eq. 5, the instantaneous $R(t) = \frac{V_{read}}{I_{read}}$ can be determined by a specific $w(t)$, when a constant low voltage V_{read} is applied.

In our experiments, $R(t)$ can be accessed via the instrument **ArC ONE** (ArC Instruments, UK¹), which applies a non-destructive read voltage pulse and measures the resulting current. In other words, the memristor resistance is assumed to be a constant (or with negligible changes) if only read pulses are applied. Therefore, similar to the aforementioned behaviour model [14], $R(t)$ can represent the hidden variable in the model equation:

$$dR(t) = \underbrace{F(R(t), v(t))dt}_{\text{deterministic term}} + \underbrace{g(R(t), v(t))\xi(t)dt}_{\text{stochastic term}} \quad (10)$$

However, the deterministic and stochastic terms can not be fitted directly from the resistance trajectories, as both terms depend on $R(t)$. To address that, a statistical data-driven method [37] is employed, as sketched in Fig. 2(a). The input of this experiment is consecutive squared voltage pulses with constant pulse width t_{pulse} . Once a pulse with an amplitude v is applied to the memristor of a resistance state R_N , the resistance is updated as R_{N+1} , resulting in a resistance change ΔR derived as the output. Similar to the forward Euler method, we approximate the time derivative of the resistance at time t as the average resistance change rate within a single pulse duration, $\frac{dR(t)}{dt}$ can be derived directly from the characterisation data as follows:

$$\frac{dR(t)}{dt} = \frac{R_{t+t_{pulse}} - R_t}{t_{pulse}} = \frac{\Delta R}{t_{pulse}} \quad (11)$$

In our data-driven process, we have observed significant variations in ΔR , even with the same v and R_N , as shown in Fig. 2(b). Since the variation probed on our memristors in experiments follows a normal distribution, ΔR can be depicted by an average term ΔR_μ and a deviation term ΔR_σ , for each pair of specific R_N and v . Given data from different scenarios of R_N and v , the function $\Delta R_\mu(v, R(t))$ and $\Delta R_\sigma(v, R(t))$ can be statistically derived. Consequently, a stochastic state change can be expressed in the following form:

$$\Delta R = \underbrace{\Delta R_\mu(v, R(t))}_{\text{deterministic term}} + \underbrace{\Delta R_\sigma(v, R(t)) \bullet \xi(t)}_{\text{stochastic term}} \quad (12)$$

However, an evident discrepancy in the resistance fluctuation of the evolution trajectory still occurs between the experimental data and the proposed SDE model, as seen in Fig. 3(a). The SDE curves have significantly larger fluctuations than the experimental data because the unit-variance random term $\xi(t)$ in the SDE is by default time-independent.

¹<https://www.arc-instruments.co.uk/products/arc-one/>

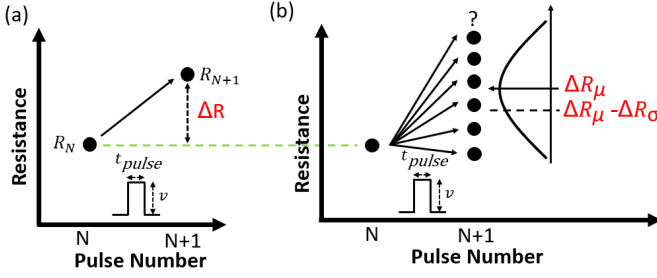


Fig. 2: Concept of the data-driven method: (a) ΔR is experimentally derived when a voltage pulse is applied. (b) The derived ΔR varies due to programming stochasticity and follows a normal distribution. Thus ΔR is decomposed into a deterministic (average) value ΔR_μ and a deviation value ΔR_σ . Both ΔR_μ and ΔR_σ are data-driven.

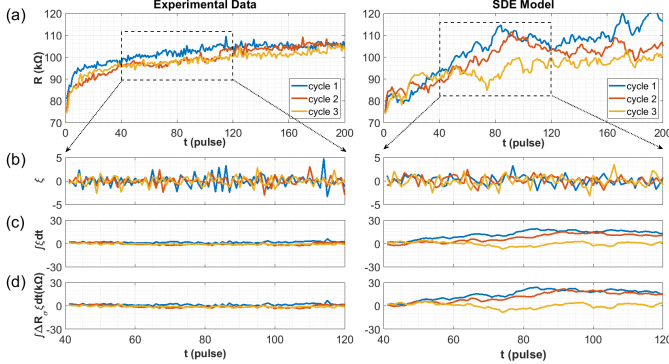


Fig. 3: The discrepancy between the experimental and SDE model curves exposes the time-dependence of unit-variance random term $\xi(t)$. (a) Resistance trajectories under consecutive programming voltage pulses (amplitude: 0.7 V, pulse width/interval: 100 μ s), each programming pulse is immediately followed by a read pulse. (b) $\xi(t)$ extracted from experimental data ($\xi(t) = \frac{\Delta R - \Delta R_\mu(v, R(t))}{\Delta R_\sigma(v, R(t))}$) shows consistency to a randomly generated unit-variance Gaussian sequence ($\mathcal{N}(0, 1)$). However, the integrals of $\xi(t)$ (c), and $\Delta R_\sigma \xi(t)$ (d) in memristor devices exhibit a time-dependent and self-convergent performance.

Specifically, $\xi(t)$ is independent of its past value $\xi(t')$ ($t' < t$), and the power of $\xi(t)$ integrated over time can be substantial due to independent increment. To investigate whether the programming stochasticity in memristor devices exhibits time dependence, $\xi(t)$, time integral of $\xi(t)$, and time integral of $\Delta R_\sigma \xi(t)$ during the 40th and 120th pulsing steps are extracted and illustrated in Fig. 3(b), (c) and (d) respectively. As observed, the experimentally derived deviation $\xi(t)$ (computed as $\xi(t) = \frac{\Delta R - \Delta R_\mu(v, R(t))}{\Delta R_\sigma(v, R(t))}$ at each time step) is highly consistent with the model's $\xi(t)$ (a randomly generated sequence of $\mathcal{N}(0, 1)$). However, the integrals $\int \xi(t) dt$ and $\int \Delta R_\sigma \xi(t) dt$ of experimental data tend to converge toward the x-axis, while those of the SDE model present a non-convergent change and significantly larger amplitudes. Therefore, $\xi(t)$ in memristor devices is not independently random but presents a conditioned randomness depending on past-time integrals. To solve this discrepancy, we replace $\xi(t)$ with a conditioned function, $m(\xi(t))$, to introduce the time dependence of the programming stochasticity into our model. The function is defined as follows:

$$m(\xi(t)) = \begin{cases} -|\xi(t)|, & \int_{t_0}^t \Delta R_\sigma m(\xi(t)) dt > 0 \\ |\xi(t)|, & \int_{t_0}^t \Delta R_\sigma m(\xi(t)) dt < 0 \end{cases} \quad (13)$$

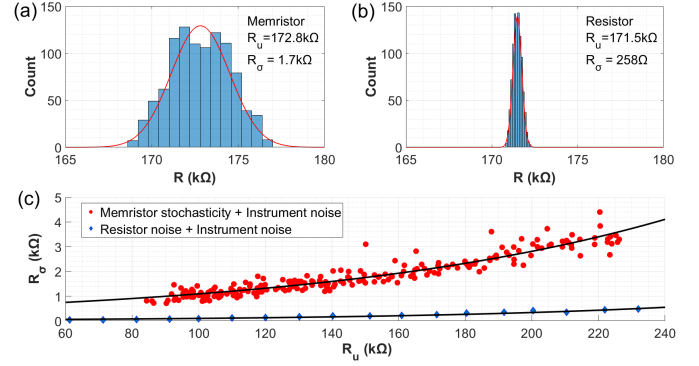


Fig. 4: Sanity checks on the measurement system (Arc ONE). (a, b) Measurement results of a stabilised memristor/a passive resistor at around 170 k Ω using Arc ONE (only read pulses applied). (c) Reading deviation of the memristor/resistors ranging from 60 k Ω to 240 k Ω . Each dot is derived from 50 times of reading. The sanity check confirms that the Arc ONE noise is a negligible contributor to programming stochasticity.

where $m(\xi(t))$ is a randomly generated standard Gaussian value, and its sign depends on the integral of $\Delta R_\sigma m(\xi(t))$ over past time duration from t_0 to t . As a result, $m(\xi(t))$ forces the entire resistance evolution in our model to converge around the deterministic resistance trajectory. The resulting SDE is as follows:

$$dR(t) = \underbrace{F(R(t), v(t))dt}_{\text{deterministic term}} + \underbrace{g(R(t), v(t))m(\xi(t))dt}_{\text{stochastic term}} \quad (14)$$

B. Data Collection

The experiments for data collection were conducted on in-house Pt/TiO_{2-x}/Pt memristors whose recipe is detailed in [25]. The measurement instrument, Arc ONE, was employed due to its ability to implement programmable pulsing schemes. Sanity checks conducted on the instrument, which involved measuring the modelled memristor devices and passive resistors, are presented in Fig. 4, demonstrating that the most measured stochasticity arises from the memristor rather than the instrument.

The benchmark pulsing protocol and the measured memristor response are presented in Fig. 5(a). The pulsing scheme that initialises the resistance at a consistently low level and then programs it to multiple higher states has been widely exploited in memristor characterisations and applications [15], [17], [22], [45], [46]. The erase pulses, with large negative amplitude, rapidly program the memristor to a consistently low resistance level, rendering it ready and responsive to subsequent write pulses. The write pulses, with positive amplitude, raise the resistance to a higher level. However, due to step-to-step stochasticity, the resistance fluctuates up and down (instead of monotonic rising), approaching a saturation region where the effect of the write pulses becomes negligible. The saturation levels vary with different write voltages, reflecting the voltage dependence of state boundaries. A read stage follows the write stage, where non-destructive read pulses of 0.5 V are applied. During the reading stage, the programmed resistance might bounce back to a stable equilibrium due to relaxation. This relaxation effect could be included in our

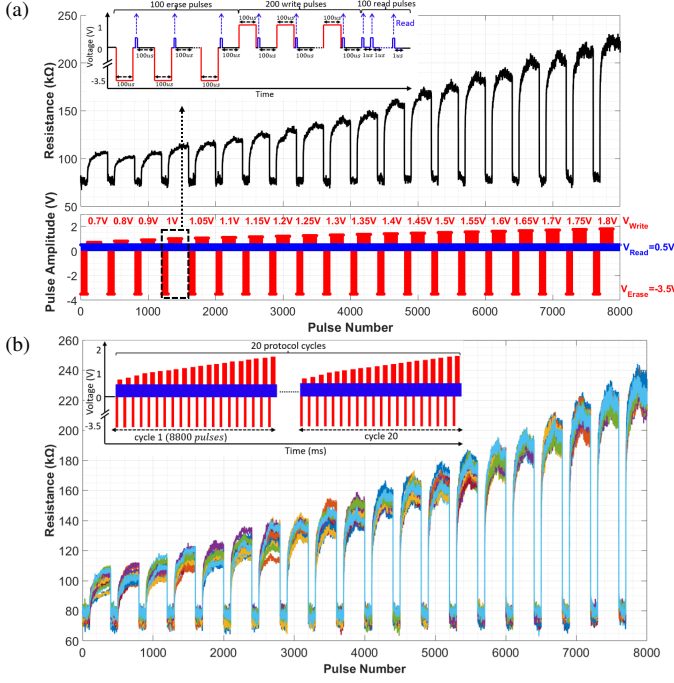


Fig. 5: (a) Transient resistance response (upper) to a cycle of input pulsing protocol (lower). The pulsing protocol consists of multiple iterations of pulse trains, upon them the write voltage increases from 0.7 V to 1.8 V. Inset: an iteration of “erase/read, write/read, read-only” pulse trains. Each iteration sequentially provides 100 erase/read pulses, 200 write/read pulses, and 100 read pulses. In ArC ONE, each erase/write pulse (100 μ s pulse width and pulse interval) is automatically followed by a read pulse (0.5 V, ≈ 0 pulse width) to update the resistance and display it to users. (b) The resistance response of a single device to consecutive 20 cycles of the protocol pulses, showing cycle-to-cycle variations. The results of 20 cycles are merged on the time scale of a single cycle, with different colours indicating different cycles. Inset: input pulses.

model as demonstrated in [47] if needed. To gather sufficient data required by the data-driven method, the experiment needs to be repetitively implemented, as shown in Fig. 5(b).

Three key considerations are outlined for deploying this pulsing protocol. First, from the experimental perspective, a large number of consecutive erase/write pulses is required. This is because the resistance change under a small number of pulses can be unpredictable and directionless, due to programming stochasticity or short-term relaxation. In contrast, the overall resistance trajectory over a large number of pulses is directional and can be reliably reproduced across repeated experiments, ensuring that the collected data is suitable for data-driven modelling. Second, from the modelling perspective, write pulses are employed with a constant amplitude in each iteration, which simplifies exploring the dependence of ΔR on v and R . By maintaining v as a constant, the $\Delta R - R$ relation can be directly derived. After determining the $\Delta R - R$ relation, the $\Delta R - v$ relation can be explored by varying the write pulse amplitude from 0.7V to 1.8V in different iterations. Third, from the perspective of application, this protocol depicts two widely observed programming phenomena, which are multi-bit programmability and step-to-step variation. Specifically, the memristor can be repeatedly reprogrammed to multiple resistance states. However, even under the same

pulsing conditions, each programming trajectory differs from the others. This step-to-step variation jeopardises the multi-state programming capability of the memristor.

C. Data processing and fitting

The proposed data-driven method utilises training data from 50 repeated protocol experiments (200k data points in total) to determine ΔR_μ and ΔR_σ . The initial step involves capturing the pulse-triggered $\Delta R - R$ relationship for each write voltage scenario. Scenarios at 1V, 1.3V and 1.6V out of a total of 20 write voltages are illustrated in Fig. 6(a) as examples (30k data points presented). The second step aims to extract meaningful patterns from extensive sets of raw data. To achieve this, the entire resistance range is divided into multiple resistance bins (the bin width is 500 Ω), assuming that resistance within each bin is at the same level. Within each bin, the distributed ΔR follows a normal distribution related to ΔR_μ and ΔR_σ . In the final step, ΔR_μ and ΔR_σ across the entire resistance spectrum are extracted for each resistance bin, as depicted in Fig. 6(b) and Fig. 6(c). These processing steps transform a substantial amount of unordered data into a format suitable for fitting.

In Fig. 6(b), ΔR_μ exhibits exponential decay as a function of R . It starts with a significant magnitude at low resistance and gradually decreases as resistance increases. When resistance approaches the boundary, ΔR_μ becomes negligible. We also observe that the resistance occasionally exceeds the boundary and often falls back within the boundary in the next pulsing step. Therefore, the fitting function of ΔR_μ can be depicted:

$$\Delta R_\mu = \begin{cases} A_1(v) \cdot \exp(B_1(v) \cdot R) \cdot \tanh\left(\frac{R_B(v) - R}{C_r}\right), & (R < R_B(v)) \\ 0, & (R > R_B(v)) \end{cases} \quad (15)$$

here $A_1(v)$ and $B_1(v)$ are voltage-dependent fitting functions that describe the decay of ΔR_μ in an exponential-like manner. $A_1(v)$ specifies the amplitude of ΔR_μ , while $B_1(v)$ specifies the shape (slope) of the exponential decay. Their voltage dependence can be revealed by data captured under different voltages. The hyperbolic tangent function is employed to improve the fitting accuracy and maintain the continuity of the exponential decay at the resistance boundary $R_B(v)$, with C_r defining the resistance region affected by the hyperbolic function.

In Figure 6(c), ΔR_σ exhibits irregular dispersion. Although it is challenging to fit the entire dispersion precisely, this model aims to capture two key properties of ΔR_σ : 1) Drift stochasticity. When the memristor undergoes a significant state change triggered by a powerful pulse, the programming uncertainty primarily depends on the external pulse. 2) Diffusion stochasticity. When the impact of the input stimulus is negligible, ΔR_σ mainly depends only on the resistance state. As seen in Fig. 6(c), states at high resistance levels exhibit more programming uncertainty. According to the data, This phenomenon can be represented in a linear relationship.

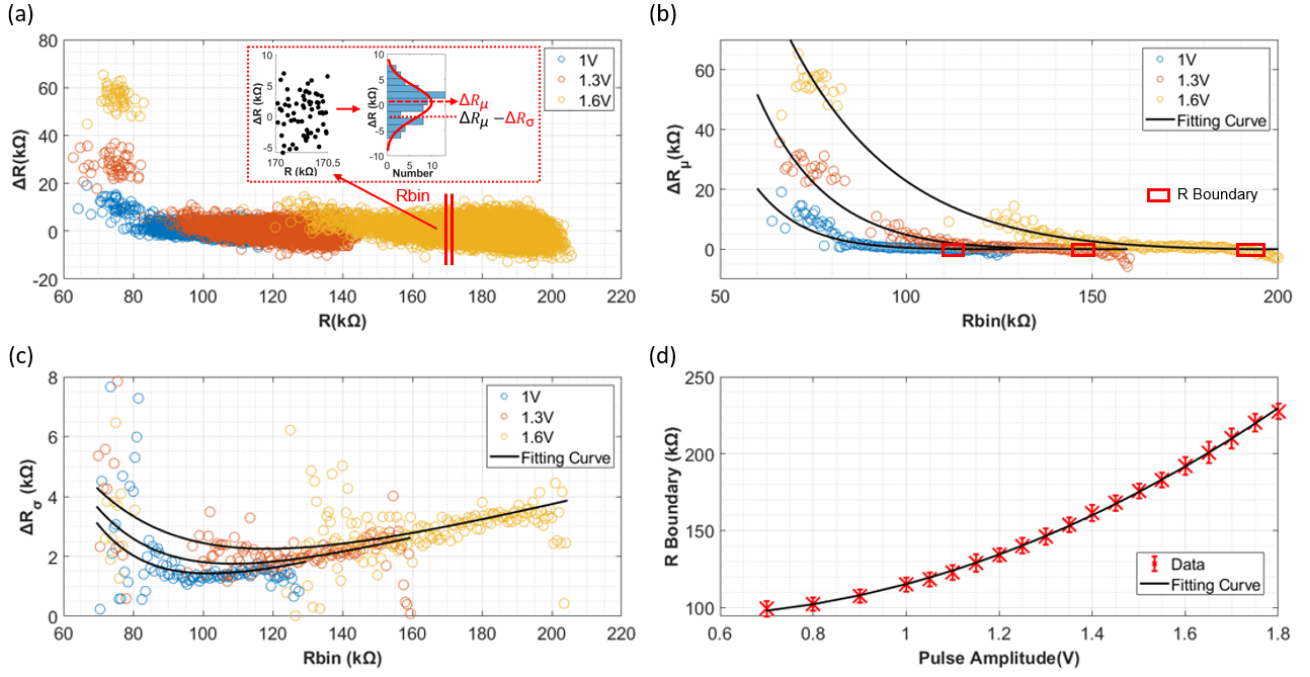


Fig. 6: Processing and fitting: (a) $\Delta R - R$ relation collected from 50 cycles of characterisation data. Out of 20 voltages, 1V, 1.3V, and 1.6V are shown as examples. All data spanning the entire range is clustered into 500 Ω -width bins to extract ΔR_μ and ΔR_σ which are shown in (b) and (c) separately. (d) Voltage-dependent resistance boundary $R_B(v)$ and fitting.

Consequently, the fitting function for ΔR_σ is as follows:

$$\Delta R_\sigma = \underbrace{A_2(v) \cdot \exp(B_2(v) \cdot R)}_{\text{drift stochasticity}} + \underbrace{k_1 \cdot R + k_2}_{\text{diffusion stochasticity}} \quad (16)$$

where $A_2(v)$, $B_2(v)$ are voltage-dependent fitting functions and $k_{1,2}$ are fitting constants applicable to all voltage scenarios.

Fig. 6(d) reveals the voltage-dependent resistance boundaries $R_B(v)$. Due to the cycle-to-cycle variation, the boundary value varies across different resistance trajectories. Consequently, instead of specifying a fixed boundary value, we define a probable range. This range begins at the resistance level where the first $|\Delta R_\mu|$ falls below 300 Ω and extends to the point where the last $|\Delta R_\mu|$ is found below 300 Ω (three exemplified boundary ranges are highlighted with the red boxes in Fig. 6(b)). The resistance boundaries in all 20 voltage scenarios are presented as error bars in the graph and have deviations around 3k Ω .

D. Results and Validation

The fitting results for the measured memristor are listed in Table. I. These fitting functions and parameters can be adjusted when applying the model to other memristor devices under different pulsing schemes, as the proposed modelling process is data-driven.

Firstly, the fitting results for both terms ΔR_μ and ΔR_σ in different R and v conditions are validated. When v is set to a constant, their 2D fitting on R are previously presented in Fig. 6(b and c). When all 20 voltage scenarios (from 0.7V to 1.8V) are covered, the fitting results in 3D planes, as depicted in Fig.

TABLE I: The information on fitting parameters

Parameter	Expression	Unit
A_1	$-0.4 \cdot v + 1.2$	$M\Omega$
A_2	$1.54 \cdot \exp(-2.25 \cdot v)$	$M\Omega$
B_1	$37.3 \cdot v - 92.9$	μS
B_2	$40 \cdot v - 97$	μS
R_B	$77.7 \cdot v^2 - 74.4 \cdot v + 112$	$k\Omega$
C_r	0.1	$M\Omega$
k_1	0.028	—
k_2	-1.9	$k\Omega$

7. In Fig. 7(a), the ΔR_μ plane highly matches the measured data, as the determined (averaged) programming effect is quite predictable through the data-driven method. In Fig. 7(b), although the 3D plane of ΔR_σ does not fit all data points as accurately due to the scattered nature of programming stochasticity, it still encompasses most data points.

Secondly, the effect of conditioned random term $m(\xi(t))$ in our SDE is validated. The random term $m(\xi(t))$ has replaced $\mathcal{N}(0, 1)$ to enable the self-convergence of programming stochasticity in this model because the practical memristor stochasticity is time-dependent and thus convergent. To validate the effect of $m(\xi(t))$, its time integral is compared with the integral of $\xi(t)$ extracted from experimental data, as shown in Fig. 8. As compared, the integral of $m(\xi(t))$ shows a self convergence.

Finally, to validate the model's capability of reproducing stochastic resistance trajectories that are highly matched to practical data in the time-domain process, the model was

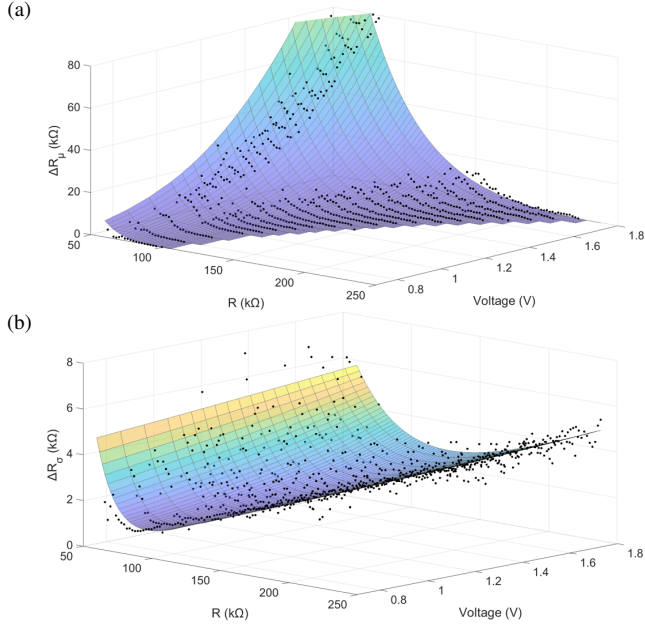


Fig. 7: Complete fitting results for (a) ΔR_μ , and (b) ΔR_σ . 3D plane: modelling surface. Black scatter: experimental data.

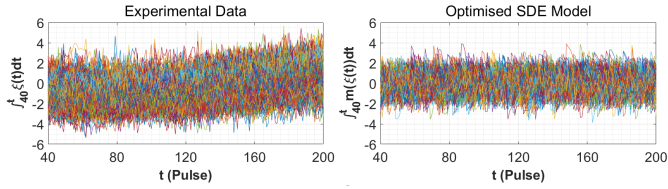


Fig. 8: Time integral of zero-mean, unit-variance $\xi(t)$ and $m(\xi(t))$ begins at the 40th time step of programming pulses. Left: integral of $\xi(t)$ extracted from experimental data, including 1000 resistance trajectories (50 cycles/voltage $\times$ 20 voltage scenarios). Right: integral of $m(\xi(t))$ from 1000 resistance trajectories generated by the model using the optimised SDE.

subjected to consecutive write pulses with a constant amplitude. As illustrated in Fig. 9, the model starts at a predefined initial resistance $74\text{k}\Omega$, then sequentially generates ΔR and updates R at each time step. In Fig. 9(a) where stochasticity is not considered, the model exhibits monotonic increments, like conventional deterministic models. In Fig. 9(b) where the stochastic behaviour is included in computing, the model generates diverse memristor evolution trajectories even under identical voltage stimuli. The produced curves exhibit fluctuations during the increment, and these fluctuations become more pronounced when the model operates at higher resistance states or with higher pulse amplitudes, consistent with the testing data.

E. Generality across Different Memristor Devices

Device-to-device variations that arise from not only the process mismatch but also the initial electroforming process [48], are a more challenging roadblock in memristor research. Although our modelling protocol does not directly account for device-to-device variation, it works for devices exhibiting distinct variations in operative resistance ranges, tolerable voltage ranges, step-to-step stochasticity, and overall memristive behaviours.

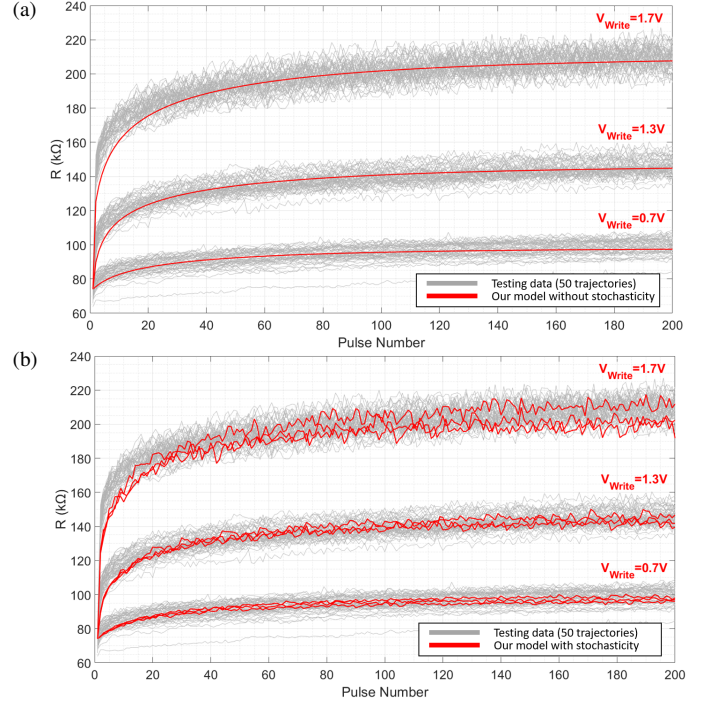


Fig. 9: The time evolution of resistance trajectories of the model (red) and the measured testing data (grey) under consecutive voltage pulses with a constant amplitude. 3 voltage scenarios are provided. Testing data are from another 50 experiments. Behaviour of: (a) stochasticity not included; (b) stochasticity included.

To showcase the generality of our modelling process across different memristor devices, it is applied to devices with significant device-to-device variations. Device 2, as exposed in Fig. 10(a), has a wide resistance range from $50\text{k}\Omega$ to $500\text{k}\Omega$, exhibiting sensitive resistance response to voltages up to 1.5V . In contrast, device 3 in Fig. 10(b) has a narrower resistance range ($90\text{k}\Omega$ to $170\text{k}\Omega$) and less sensitivity to voltages up to 2.5V . Training data for both devices is gathered from “erase-write” pulse trains similar to the experiment presented before. To test if the model can predict voltage inputs beyond the training data set, only 5 voltage scenarios out of all measured data are used to construct the model’s SDE. In consequence, the predicted/reproduced resistance trajectories are presented in the figure with testing data.

Although the previous modelling processes only addressed positive stimuli, the model also accommodates negative stimuli. In Fig. 10(c), device 4 is programmed by alternating voltage pulses ranging from $|1\text{V}|$ to $|1.5\text{V}|$. To effectively model its behaviour under negative stimuli, the same modelling process utilised for addressing positive stimuli needs to be applied again for negative stimuli.

F. Model Comparison and Discussion

The comparison between our model and other stochastic models is outlined in Table. II. Some considerations for building up a stochastic model from the perspective of circuit designs are discussed below:

- 1) The stochasticity should be reasonably quantified and represented. In this work, the stochastic component is

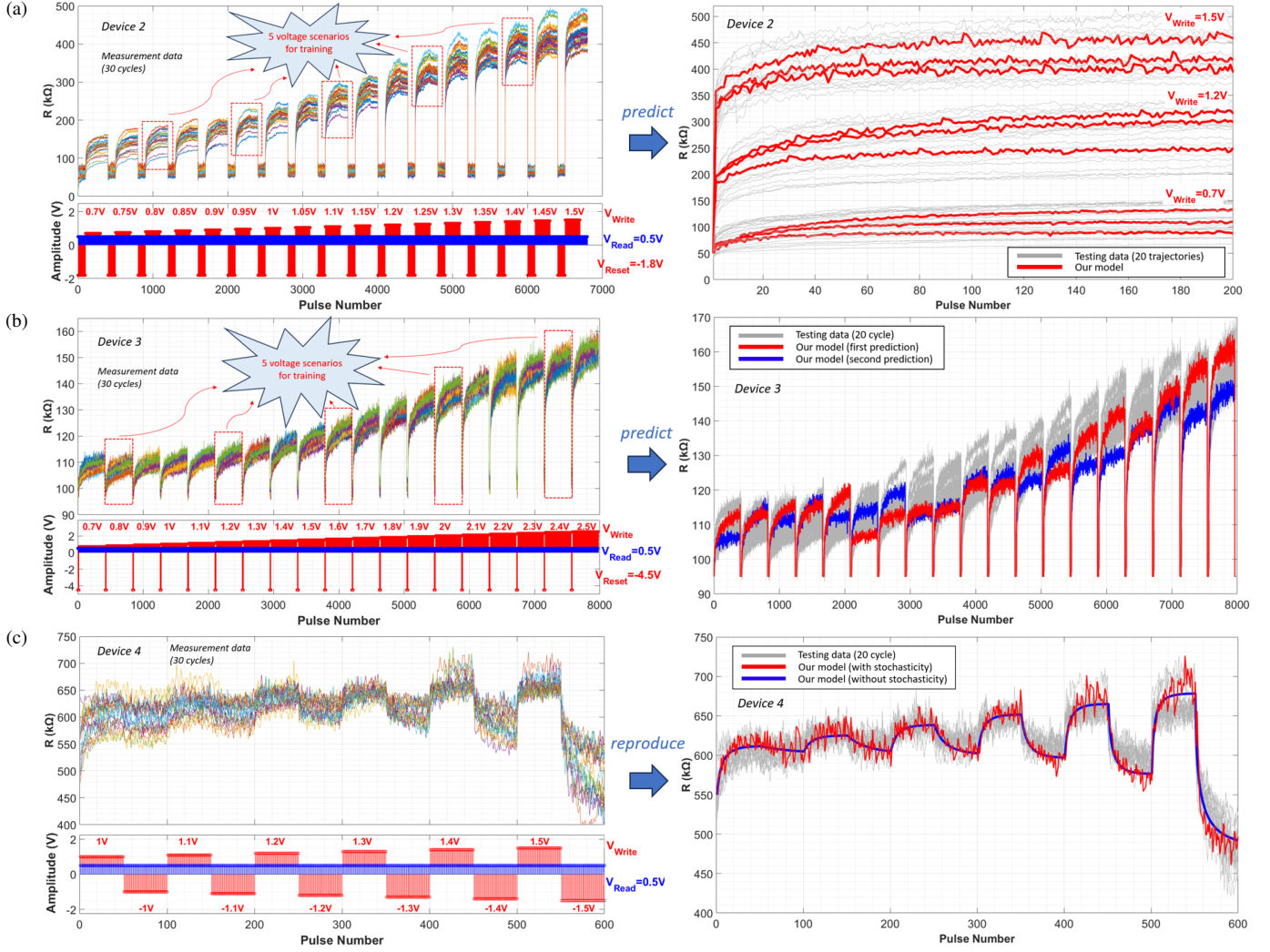


Fig. 10: Our modelling method applied to: (a) device 2; (b) device 3; (c) device 4. In these modelling processes, the model is trained using 30 cycles of measurement data (left) and its modelling performance is tested using another 20 cycles (right). These results showcase the generality of our modelling method across various memristor devices subjected to different pulsing schemes.

TABLE II: Comparison between existing stochastic models

Features	Our model	[32]	[40]	[37]	[36], [39]	[34], [35], [41]–[44]
Variable regarded as stochastic	ΔR	T	R	ΔR	V_{TH}	Fitting parameters
Computing method	SDE	SDE	Markov	-	SDE	Monte Carlo
Stochasticity evolves with time	Yes	Yes	Yes	Yes	Yes	No
Stochasticity validated in time domain	Yes	Yes	Yes	No	No	-
Predictability across different stimuli	Yes	No	No	Yes	No	-
Consistency across repetitive stimuli	Yes	-	-	-	-	-
Adapted in SPICE	Yes	Yes	No	No	Yes	-
Distribution type of stochasticity	Gaussian	Gaussian	-	Gaussian	Poisson	Gaussian (mostly)
Mathematics of stochasticity amplitude	Exponential/linear	Exponential	Matrix	linear	Constant	Constant

extracted from the measured behaviours and quantified through a data-driven approach. This is then modelled and represented by an optimised SDE to depict how stochasticity evolves over time.

- 2) The model should demonstrate predictability. For example, a memristor model trained using data from stimuli at 2 V and 4 V should make a reasonable estimation for the

stochastic responses at 3 V. A model constructed solely by fitting directly to an experimental data set might not apply to other pulsing conditions and thus be useless. However, this model utilises valuable patterns learned from the training data to investigate the relationship between the derivative of resistance and key impact parameters (v , R) and finally construct the SDE. Con-

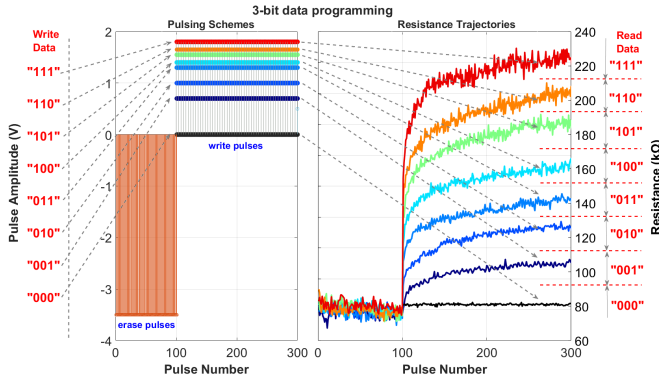


Fig. 11: Conceptual methodology for multi-level data storage by linking the write/read data to the memristor programming behaviours.

sequently, it reasonably predicts resistance trajectories across various simulation scenarios.

- 3) The experimental/modelling data should exhibit repeatability. We have observed completely irregular and non-reproducible memristor behaviours in experiments, especially during the initial electroforming process or when subjected to extremely high voltages. Therefore, the experimental resistance response to external pulses should display repeatability or, at the very least, a reasonable level of consistency before undergoing modelling. When employing such a behaviour model in circuit simulations, it is assumed that the modelled device will still exhibit similar performance once applied in circuits.
- 4) The model should be adaptable to SPICE simulators. Details are described in the following section.

IV. MULTI-BIT MEMORY SIMULATION

This section has two main objectives: 1) demonstrating the model's compatibility with system-level circuit simulations and depicting its transient performance; 2) investigating the feasibility of storing multi-bit data in a single memristor using peripheral CMOS circuits, whilst accounting for programming stochasticity.

A. Methodology

In general, research on multi-bit memristor memory uses multiple predefined resistance levels to represent different digits. To achieve multi-bit resistance states, an “erase-write” method is commonly exploited. It has two steps: 1) initialise the memristor to a consistent resistance level, and 2) program it to a specific level, as reported in previous works [26]–[31].

Building upon this scheme, the aforementioned “erase/read, write/read, read-only” pulsing scheme for memristor characterisation is repurposed for this application. Consequently, we present a straightforward open-loop methodology that consistently links input and output digits to the experimental analogue behaviours, as depicted in Fig. 11. The general process can be summarised as follows:

- 1) binary input data is mapped to a corresponding write voltage,
- 2) the write voltage is then mapped to a resistance state through a write operation on the device,

- 3) Once stored in the device, the resistance state can be read back and mapped back to the data in binary.

Although such an open-loop method is not as accurate as adaptable pulsing methods that incorporate feedback mechanisms, it is more straightforward and has been executed in our previous experiments. Therefore, the previous experimental results can verify the model's behaviours and provide a reasonable estimation of the data storage simulation.

However, two significant challenges in the multi-bit memristor memory research remain. First, there are yet no memristor models applied in the multi-bit memory simulation that inform the step-to-step variations which are crucial for evaluating the multi-bit capabilities of memristors. In this section, we have proposed a stochastic Verilog-A model to address this information gap. Second, due to substantial variations, it is difficult to define both the resistance regions that correspond to output digits and the specific pulsing schemes associated with writing input digits. In this section, a peripheral CMOS system is designed to automatically and dynamically determine the input pulsing schemes and the output resistance ranges.

B. Optimised Verilog-A Model

The translation of our SDE model into a Verilog-A format for transient simulation is challenging due to the random nature of SDE. To address this, the prior work [49] treats the SDE as an ideal circuit component combined with a noise source. For example, a resistor's thermal noise is represented as a current source in parallel with an ideal resistor. The current source amplitude randomly evolves over time according to a white noise process. Additionally, numerical schemes for transient noise analysis were adapted to stochastic differential-algebraic equations within circuit simulation [50]. However, in our model, the memristor stochasticity depends on its resistance and thus cannot be regarded as a separate noise source. To address that, another method is employed which calculates the time integral of the state derivative using the “abstime” function, as demonstrated in [51]. Our stochastic model can be readily adapted to it.

To enhance our model's computing speed, a few optimisation strategies have been employed as follows:

- 1) The model allows toggling the stochasticity on or off to accelerate simulations when only deterministic behaviour is required.
- 2) Computationally expensive exponential and hyperbolic tangent functions have been replaced by their third-order Taylor series approximations.
- 3) In this specific circuit simulation, the model's response to negative stimuli is idealised as the simulation's focus lies primarily on positive stimuli.
- 4) The stochastic equation retains only the diffusion stochasticity, as it holds more significance in influencing the simulation outcome.

Key codes of this Verilog-A model used in the following application are provided in Listing. 1.

```

analog begin
  @ (initial_step) begin
    R_value=74000;
    M=0;
    R_random=0;
    //switch1 for cycle-to-cycle variation
    if (switch1==1)
      RV=$rdist_normal(seed1,0,3000);
    if (switch1==0)
      RV=0;
    end
    dt=$abstime-it;
    v=V(p,n);
    //programming under positive inputs
    if (v > 0 ) begin
      A1=-400000*v+1200000;
      B1=0.0000373*v-0.000093;
      k1=0.028;
      k2=-1900;
      Cr=0.00001;
      RB=77660*v*v-74440*v+112000+RV;
      //delta R median
      if (R_value <= RB) begin
        fexp=(1+(B1*R/8)+(B1*R/8)**2/2+(B1*R/8)**3/6)**8;
        z=1+(Cr*(RB-R))+(Cr*(RB-R)**2/2+(Cr*(RB-R)**3/6);
        ftanh=1-2/(1+z**2);
        R_mean=A1*fexp*ftanh*(dt/0.0001);
      end
      if (R > RB)
        R_mean=0;
      //switch2 for step-to-step variation
      if (switch2==1) begin
        sigma=k1*R+k2;
        xi=$rdist_normal(seed2,0,1)
        if (M>0)
          xi=-abs(xi);
        if (M<0)
          xi=abs(xi);
        R_random=xi*sigma;
        M=M+R_random;
      end
      //update R
      R=R+R_mean+R_random;
    end
    //idealised initialisation progress
    if (v<=-2) begin
      R=74000;
      M=0;
      if (switch1==1)
        RV=$rdist_normal(seed1,0,3000);
      if (switch1==0)
        RV=0;
    end
    it=$abstime;
    IV=v/R;
    I(p,n)<+ IV;
  end
end

```

Listing 1: Key Verilog-A code of the model

C. Monolithic CMOS-memristor system

The mixed-signal CMOS-memristor system designed in TSMC 180nm technology for 2-bit data storage is presented in Fig. 12(a). The system interfaces with external user commands (characterise, write, and read) to write/read data. The digital controller plays a central role in this process. It operates an 8-bit DAC to generate square pulses with controllable amplitudes and fixed pulse width for programming the device under test (DUT). Additionally, the logical unit generates the necessary timing signals for converters and analogue switches. When the analogue switch S_1 turns on, negative programming pulses are applied to erase the memristor's resistance to an initial level. When S_2 turns on, positive programming pulses are

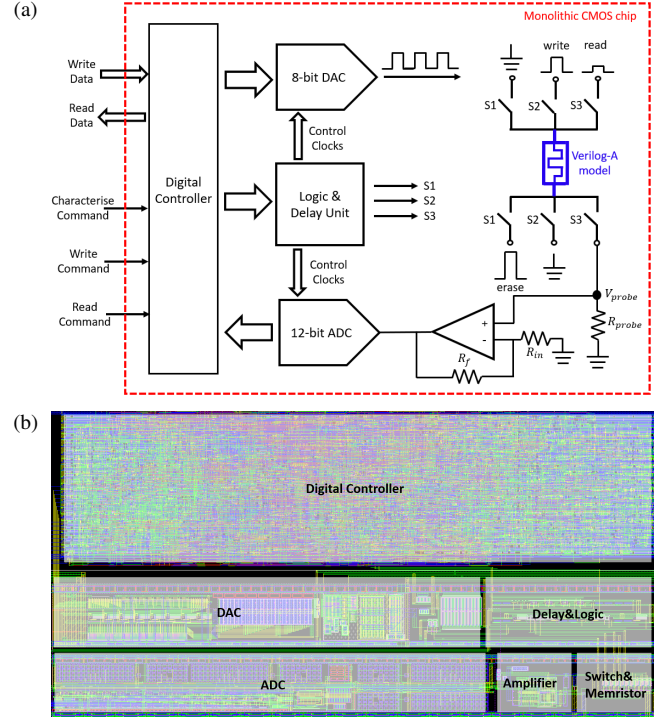


Fig. 12: Monolithic CMOS-memristor system. (a) Circuit overview. (b) Layout design of the integrated circuit.

applied to write specific data into the memristor. Both the programming voltage and the ground signal are directly applied to the electrodes of the memristor, guaranteeing sufficient voltage for memristor programming. When S_3 switches on, a low-amplitude read pulse (0.05V) that causes a negligible resistance change is applied to the top electrode. The bottom electrode is connected to a probe resistor ($10k\Omega$) much smaller than the memristor. The small voltage, V_{probe} , at the bottom electrode is amplified by a non-inverting amplifier to fall within the input range of a 12-bit ADC. The resulting digital value is then sent via an external interface for data acquisition. All these electronic components are physically implemented and placed in a monolithic microchip, as shown in Fig. 12(b). In the layout design, we have allocated enough space for memristors, which are intended to be integrated on top of the CMOS circuits in the fabrication procedures [52].

The monolithic system performs three operations triggered by user commands. Each operation runs to completion and cannot be interrupted by other user commands. During the characterise operation, the system determines the resistance regions of the four output states, along with their corresponding write voltages. During the write operation, the write digits are converted to the amplitude of a pulse train which programs the memristor to a resistance level. During the read operation, the programmed resistance level is converted back as the read digits.

D. Simulation results

The post-layout simulation of the designed 2-bit memory system was conducted using the Spectre AMS Designer simulator. Fig. 13(a) displays the resistance of the memristor

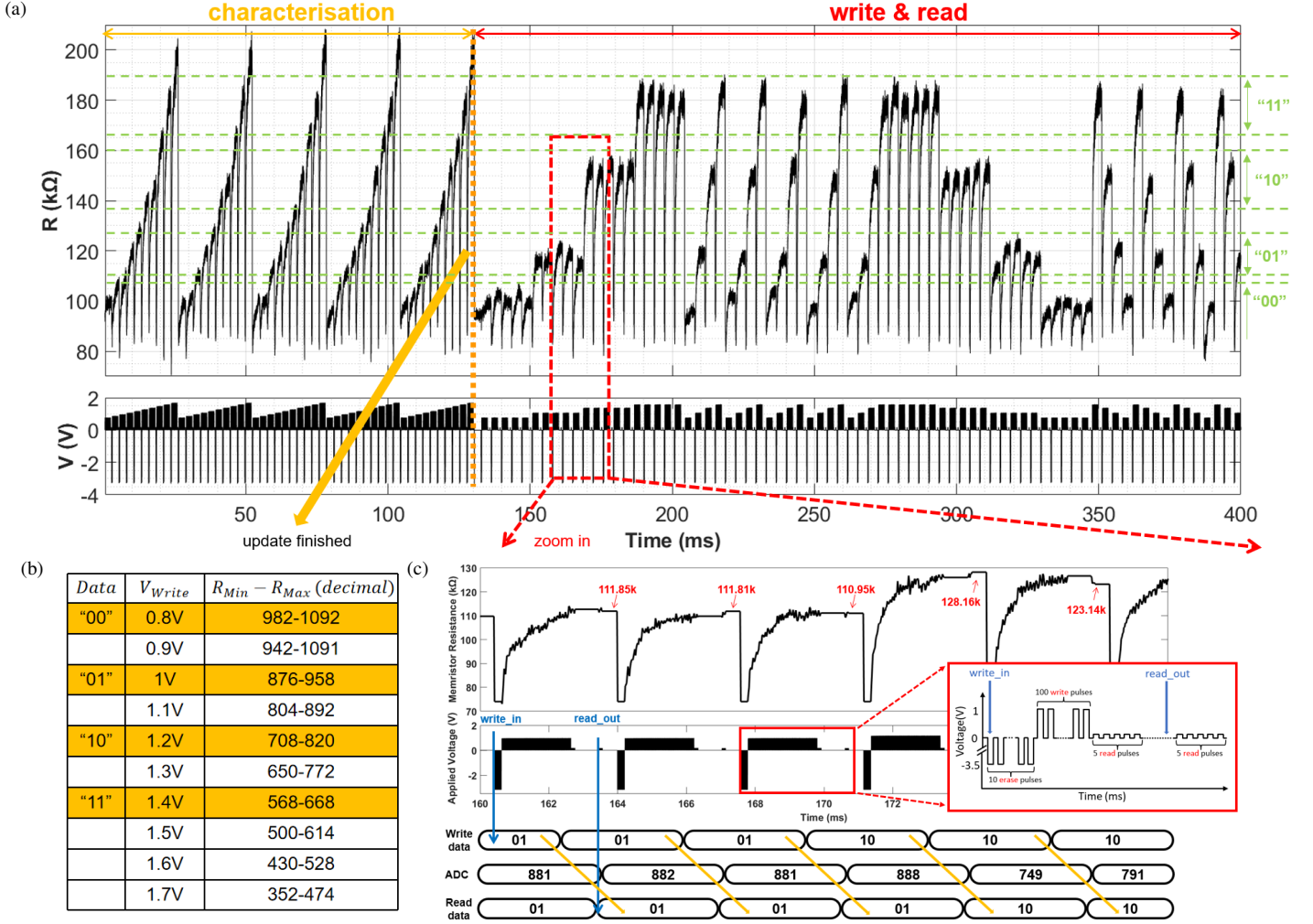


Fig. 13: Simulation results for the 2-bit storage application. (a) The stochastic resistance evolution of the memristor model (upper), and the voltage across the memristor (lower). The simulation contains a characterise operation and consecutive write/read operations. (b) Once the characterisation is finished, write voltages and resistance ranges, corresponding to the data, are determined. (c) Enlarged write/read operations, with the time sequence of write data, ADC outputs, and read data. Inset: detailed information on applied pulses.

model and the applied voltages across the model in a transient simulation. As observed, our model can introduce the programming stochasticity of the characterised memristor into the simulation. The transient simulation involves a characterise operation followed by consecutive write/read operations. In the characterise operation, the pulse generator delivers 5 cycles of protocol pulses to the memristor. Each protocol cycle consists of 10 iterations of “erase-write-read” pulse trains with the write voltage increasing from 0.8V to 1.7V in 0.1V increments. During this process, starting from the initialised 74kΩ, the resistance follows different evolution trajectories. The final resistance is probed by read pulses and recorded as digits in registers. Similar to the experimental behaviours, the memristor model produces variable resistance trajectories across different iterations, even under the same write voltage pulses. Upon completing the characterise operation, the probed ranges of final resistance for all 10 write voltages are converted into decimal digits. Because larger resistance variations might

occur when more cycles of operations are implemented, we extend the probed ranges by 10 (in decimal) at both the upper and lower boundaries, resulting in final R_{Min} and R_{Max} as displayed in Fig. 13(b). Among these 10 resistance ranges, the system automatically selects 4 distinct ranges for output digits and determines the corresponding voltages to represent input digits. In the given example, the write voltages 0.8V, 1V, 1.2V, and 1.4V map to the input digits “00”, “01”, “10”, and “11”. If the read resistance falls in these effective regions, a 2-bit digit can be read out. Otherwise, if it falls out of effective regions, an error signal is output. In the write/read operations, write/read commands alternatively instruct the system to encode/decode data according to the predefined voltage amplitudes or resistance regions. Part of the write/read results are enlarged in Fig. 13(c) as an example. When the first write command is issued, the system captures the write data “01”, and generates “erase-write-read” pulses with 1V write amplitude to program the memristor to

a higher resistance level; When a read pulse is released by the read command, the captured resistance (111.85k Ω) is digitised by the ADC (882) and compared with the predefined resistance regions. As the ADC output falls within the region 876-958, the stored information is revealed as "01". In the first three write/read operations, when the same write digit "01" results in different programmed resistances and ADC outputs due to the variation introduced by the model, the system still correctly retrieves the corresponding read digit.

V. CONCLUSION

In this paper, a data-driven stochastic compact memristor model was introduced. It allows step-to-step programming stochasticity of memristors observed in experiments to be evaluated in time-domain circuit simulation.

In our modelling process, the stochastic trajectory of memristor resistance under voltage pulses is described by a stochastic differential equation (SDE). The SDE comprises both deterministic and stochastic terms presented in simple mathematical expressions. They are statistically derived from a data-driven method using a large data set of the resistance responses to applied pulses. As a result, our model reproduces a random resistance trajectory that closely aligns with the experimental behaviour of the modelled memristor. Also, this modelling method applies to different memristors with significant device-to-device variations. More importantly, the model is adapted to the Verilog-A style, enabling its simulation in the Spectre AMS Designer simulator when incorporated into a mixed-signal CMOS system. The system explores the programming performance of storing two-bit data in a single memristor.

Our memristor model has manifested itself as a pragmatic behaviour model in circuit simulations, providing valuable insights into stochasticity across diverse programming scenarios. Designers can leverage this model to consider programming stochasticity in circuit design or develop more robust systems for achieving efficient memristive programming.

REFERENCES

- [1] G. Cao, C. Gao, J. J. Wang, J. L. Lan, and X. B. Yan, "Memristor based on two-dimensional titanic nanosheets for multi-level storage and information processing," *Nano Research*, vol. 15, no. 9, pp. 8419–8427, 2022.
- [2] Y. Wang, Q. Liu, S. B. Long, W. Wang, Q. Wang, M. H. Zhang, S. Zhang, Y. T. Li, Q. Y. Zuo, J. H. Yang, and M. Liu, "Investigation of resistive switching in Cu-doped HfO₂ thin film for multilevel non-volatile memory applications," *Nanotechnology*, vol. 21, no. 4, 2010.
- [3] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Letters*, vol. 10, no. 4, pp. 1297–1301, 2010.
- [4] L. O. Chua and S. M. Kang, "Memristive devices and systems," *Proceedings of the IEEE*, vol. 64, no. 2, pp. 209–223, 1976.
- [5] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
- [6] J. J. Yang, M. D. Pickett, X. Li, D. A. A. Ohlberg, D. R. Stewart, and R. S. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevices," *Nature Nanotechnology*, vol. 3, no. 7, pp. 429–433, 2008.
- [7] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams, "Switching dynamics in titanium dioxide memristive devices," *Journal of Applied Physics*, vol. 106, no. 7, 2009.
- [8] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: properties of basic electrical circuits," *European Journal of Physics*, vol. 30, no. 4, pp. 661–675, 2009.
- [9] T. Chang, S. H. Jo, K. H. Kim, P. Sheridan, S. Gaba, and W. Lu, "Synaptic behaviors and modeling of a metal oxide memristive device," *Applied Physics a-Materials Science Processing*, vol. 102, no. 4, pp. 857–863, 2011.
- [10] T. Prodromakis, B. P. Peh, C. Papavassiliou, and C. Toumazou, "A versatile memristor model with nonlinear dopant kinetics," *IEEE Transactions on Electron Devices*, vol. 58, no. 9, pp. 3099–3105, 2011.
- [11] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, "A memristor device model," *IEEE Electron Device Letters*, vol. 32, no. 10, pp. 1436–1438, 2011.
- [12] S. Kvaterny, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: Threshold adaptive memristor model," *IEEE Transactions on Circuits and Systems I-Regular Papers*, vol. 60, no. 1, pp. 211–221, 2013.
- [13] C. Yakopcic, T. M. Taha, G. Subramanyam, and R. E. Pino, "Generalized memristive device SPICE model and its application in circuit design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 8, pp. 1201–1214, 2013.
- [14] I. Messaris, A. Serb, S. Stathopoulos, A. Khia, S. Nikolaidis, and T. Prodromakis, "A data-driven Verilog-A ReRAM model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 12, pp. 3151–3162, 2018.
- [15] F. Alibart, L. G. Gao, B. D. Hoskins, and D. B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," *Nanotechnology*, vol. 23, no. 7, 2012.
- [16] L. Zhao, H. Y. Chen, S. C. Wu, Z. Jiang, S. Yu, T. H. Hou, H. S. P. Wong, and Y. Nishi, "Multi-level control of conductive nano-filament evolution in HfO₂ ReRAM by pulse-train operations," *Nanoscale*, vol. 6, no. 11, pp. 5698–5702, 2014.
- [17] L. Wu, H. X. Liu, J. B. Li, S. L. Wang, and X. Wang, "A multi-level memristor based on Al-doped HfO₂ thin film," *Nanoscale Research Letters*, vol. 14, 2019.
- [18] Z. Biolek, D. Biolek, and V. Biolkova, "Spice model of memristor with nonlinear dopant drift," *Radioengineering*, vol. 18, no. 2, pp. 210–214, 2009.
- [19] H. Abdalla and M. D. Pickett, "Spice modeling of memristors," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*, Conference Proceedings, pp. 1832–1835.
- [20] R. Berdan, C. Lim, A. Khia, C. Papavassiliou, and T. Prodromakis, "A memristor spice model accounting for volatile characteristics of practical ReRAM," *IEEE Electron Device Letters*, vol. 35, no. 1, pp. 135–137, 2014.
- [21] S. Gaba, P. Sheridan, J. T. Zhou, S. Choi, and W. Lu, "Stochastic memristive devices for computing and neuromorphic applications," *Nanoscale*, vol. 5, no. 13, pp. 5872–5878, 2013.
- [22] D. Wust, D. Fey, and J. Knödtel, "A programmable ternary cpu using hybrid cmos/memristor circuits," *International Journal of Parallel Emergent and Distributed Systems*, vol. 33, no. 4, pp. 387–407, 2018.
- [23] C. Mahata, M. Kang, and S. Kim, "Multi-level analog resistive switching characteristics in tri-layer HfO based memristor on ITO electrode," *Nanomaterials*, vol. 10, no. 10, 2020.
- [24] S. Lee, J. Jeon, K. Eom, C. Jeong, Y. Yang, J. Y. Park, C. B. Eom, and H. Lee, "Variance-aware weight quantization of multi-level resistive switching devices based on Pt/LaAlO₃/SrTiO₃ heterostructures," *Scientific Reports*, vol. 12, no. 1, 2022.
- [25] L. Michalas, S. Stathopoulos, A. Khia, and T. Prodromakis, "Conduction mechanisms at distinct resistive levels of Pt/TiO₂-x/Pt memristors," *Applied Physics Letters*, vol. 113, no. 14, Oct. 2018.
- [26] X. Zhu, C. Q. Wu, Y. H. Tang, J. J. Wu, and X. Yi, "Multi-level programming of memristor in nanocrossbar," *Ieice Electronics Express*, vol. 10, no. 5, 2013.
- [27] P. Rabbani, R. Dehghani, and N. Shahpari, "A multilevel memristor-cmos memory cell as a ReRAM," *Microelectronics Journal*, vol. 46, no. 12, pp. 1283–1290, 2015.
- [28] N. Taherinejad, M. P. Sai, M. Rathmair, and A. Jantsch, "Fully digital write-in scheme for multi-bit memristive storage," in *2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2016, pp. 1–6.
- [29] Y. Yilmaz and P. Mazumder, "A drift-tolerant read/write scheme for multilevel memristor memory," *IEEE Transactions on Nanotechnology*, vol. 16, no. 6, pp. 1016–1027, 2017.

- [30] M. Ramadan, N. Wainstein, R. Ginosar, and S. Kvatinsky, "Adaptive programming in multi-level cell ReRAM," *Microelectronics Journal*, vol. 90, pp. 169–180, 2019.
- [31] M. M. Adnan, S. Amer, S. Sayyaparaju, M. S. Hasan, and G. S. Rose, "A scan register based access scheme for multilevel non-volatile memristor memory," in *26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 630–633.
- [32] X. M. Guan, S. Y. Yu, and H. S. P. Wong, "A SPICE compact model of metal oxide resistive switching memory with variations," *Ieee Electron Device Letters*, vol. 33, no. 10, pp. 1405–1407, 2012.
- [33] S. Gaba, P. Knag, Z. Y. Zhang, and W. Lu, "Memristive devices for stochastic computing," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014, pp. 2592–2595.
- [34] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, and D. Ielmini, "Statistical fluctuations in HfOx resistive-switching memory: Part I - set/reset variability," *IEEE Transactions on Electron Devices*, vol. 61, no. 8, pp. 2912–2919, 2014.
- [35] P. Y. Chen and S. M. Yu, "Compact modeling of RRAM devices and its applications in 1T1R and 1S1R array design," *IEEE Transactions on Electron Devices*, vol. 62, no. 12, pp. 4022–4028, 2015.
- [36] R. Naoos, M. Al-Shedivat, and K. N. Salama, "Stochasticity modeling in memristors," *IEEE Transactions on Nanotechnology*, vol. 15, no. 1, pp. 15–28, 2016.
- [37] S. Stathopoulos, A. Serb, A. Khiat, M. Ogorzalek, and T. Prodromakis, "A memristive switching uncertainty model," *IEEE Transactions on Electron Devices*, vol. 66, no. 7, pp. 2946–2953, 2019.
- [38] A. Malik, C. Papavassiliou, and S. Stathopoulos, "A stochastic compact model describing memristor plasticity and volatility," in *28th IEEE International Conference on Electronics, Circuits, and Systems (IEEE ICECS)*, 2021.
- [39] J. Cayo, I. Vourkas, and A. Rubio, "A circuit-level spice modeling strategy for the simulation of behavioral variability in ReRAM," in *30th IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2022, Conference Proceedings.
- [40] A. Malik, C. Papavassiliou, and S. Stathopoulos, "An absorbing markov chain model for stochastic memristive devices," in *11th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2022, Conference Proceedings.
- [41] M. Rathore, R. J. Weiss, M. S. A. Shawkat, G. S. Rose, M. Liehr, M. Abedin, S. Rafiq, and N. C. Cady, "A compact model for the variable switching dynamics of HfO2 memristors," in *2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2022, pp. 1–4.
- [42] F. Vaccaro, S. Brivio, S. Perotto, A. G. Mauri, and S. Spiga, "Physics-based compact modelling of the analog dynamics of HfOx resistive memories," *Neuromorphic Computing and Engineering*, vol. 2, no. 2, 2022.
- [43] C. Bischoff, J. Leise, E. P. B. Quesada, E. Perez, C. Wenger, and A. Kloes, "Implementation of device-to-device and cycle-to-cycle variability of memristive devices in circuit simulations," *Solid-State Electronics*, vol. 194, 2022.
- [44] J. B. Roldán, E. Miranda, D. Maldonado, A. N. Mikhaylov, N. Agudov, A. A. Dubkov, M. N. Koryazhkina, M. B. González, M. A. Villena, S. Poblador, M. Saludes-Tapia, R. Picos, F. Jiménez-Molinos, S. G. Stavrínides, E. Salvador, F. J. Alonso, F. Campabadal, B. Spagnolo, M. Lanza, and L. O. Chua, "Variability in resistive memories," *Advanced Intelligent Systems*, vol. 5, no. 6, 2023.
- [45] S. Stathopoulos, A. Khiat, M. Trapatseli, S. Cortese, A. Serb, I. Valov, and T. Prodromakis, "Multibit memory operation of metal-oxide bi-layer memristors," *Scientific Reports*, vol. 7, no. 1, p. 17532, 2017.
- [46] P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H.-S. P. Wong *et al.*, "Face classification using electronic synapses," *Nature communications*, vol. 8, p. 15199, 2017.
- [47] R. Berdan, C. Lim, A. Khiat, C. Papavassiliou, and T. Prodromakis, "A memristor spice model accounting for volatile characteristics of practical rram," *IEEE Electron Device Letters*, vol. 35, no. 1, pp. 135–137, 2014.
- [48] A. Marchewka, R. Waser, and S. Menzel, "Physical modeling of the electroforming process in resistive-switching devices," in *2017 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, 2017, pp. 133–136.
- [49] O. Schein and G. Denk, "Numerical solution of stochastic differential-algebraic equations with applications to transient noise simulation of microelectronic circuits," *Journal of Computational and Applied Mathematics*, vol. 100, no. 1, pp. 77–92, 1998.
- [50] G. Denk and R. Winkler, "Modelling and simulation of transient noise in circuit simulation," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 13, no. 4, pp. 383–394, 2007.
- [51] S. Maheshwari, S. Stathopoulos, J. Wang, A. Serb, Y. Pan, A. Mifsud, L. B. Leene, J. Shen, C. Papavassiliou, T. G. Constandinou, and T. Prodromakis, "Design flow for hybrid CMOS/memristor systems—part i: Modeling and verification steps," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 12, pp. 4862–4875, 2021.
- [52] A. Mifsud, J. Shen, P. Feng, L. Xie, C. Wang, Y. Pan, S. Maheshwari, S. Agwa, S. Stathopoulos, S. Wang, A. Serb, C. Papavassiliou, T. Prodromakis, and T. G. Constandinou, "A CMOS-based characterisation platform for emerging RRAM technologies," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 75–79.