Optimization Schemes for In-Memory Linear Regression Circuit with Memristor Arrays

Shiqing Wang, Zhong Sun, *Member, IEEE*, Yuheng Liu, Shengyu Bao, Yimao Cai, Daniele Ielmini, *Fellow, IEEE*, and Ru Huang, *Fellow, IEEE*

***Abstract*—Recently, an in-memory analog circuit based on crosspoint memristor arrays was reported, which enables solving linear regression problems in one step and can be used to train many other machine learning algorithms. To explore its potential for computing accelerator applications, it is of fundamental importance to improve the computing speed of the circuit, namely to reduce its response time towards correct outputs. In this work, we comprehensively studied the transfer function of this circuit, resulting in a quadratic eigenvalue problem that describes the distribution of poles of the circuit. The minimal real part of non-zero eigenvalues defines the dominant pole, which in turn dominates the response time. Simulations for multiple linear regression solutions with different datasets evidence that, the computing time does not necessarily increase with problem size, rather it is solely determined by the minimal eigenvalue. The dominant pole is related to variables in the circuit, including feedback conductance, and gain bandwidth products of amplifiers. By optimizing these parameters synergisitically, the dominant pole shifts to higher frequencies and the computing speed is consequently optimized. Our results provide a guideline for design and optimization of in-memory machine learning accelerators with analog memristor arrays. Also, issues including power consumption and noise impact are investigated in terms of the circuit variable, thus offering a comprehensive evaluation of its impact.**

***Index Terms*—analog computing, in-memory computing, linear regression, machine learning, memristor.**

I. Introduction

NOWADAYS machine learning (ML) systems are facing a severe energy problem [1]. In-memory computing is a promising solution, thanks to the elimination of data movement between the physically separated computing and memory units in conventional computers [2], also to the intrinsically massive parallelism in emerging memory architectures that can be exploited for computing [3]. Crosspoint memristor arrays enable naturally calculation of matrix-vector multiplication (MVM) in one step in the analog domain, providing an acceleration scheme for many important algorithms such as neural networks [4], signal and image processing [5]. By adopting an appropriate setup, the time complexity of analog MVM computation can be as low as $O(1)$ [6]. On the other hand, by configuring crosspoint feedback loops, basic linear algebraic problems including systems of linear equations and matrix eigenvectors can be solved in one step as well [7]-[10]. Especially, by connecting two identical crosspoint memristor arrays to form a negative feedback loop, linear regression can be conveniently solved in the closed form with the circuit, facilitating a one-step solution to the training of many ML algorithms, such as logistic regression and neural network [11]. To support its potential for ML accelerator applications, a critical point is to investigate the computing time of the circuit and develop optimization schemes correspondingly, which will ultimately lead to an improved energy efficiency and the overall performance.

In this work, we intensively studied the transfer characteristics of the in-memory linear regression circuit. The transfer function analysis results in a quadratic eigenvalue problem (QEP), where the minimum of absolute real part of non-zero solutions dictates the dominant pole of the circuit, which in turn dominates the computing time. Such analysis is validated by a bunch of SPICE (Simulation Program with Integrated Circuit Emphasis) simulations with real-world datasets. As the problem size, *i.e.* the number of rows in the resulting matrix increases, the circuit response is found not necessarily to delay. According to the QEP as obtained, the dominant pole is determined by variables in the circuit, including feedback conductance (c), and gain bandwidth products (GBWPs p_1 and p_2) of the two sets of operational amplifiers (OAs). By optimizing these parameters, the computing speed can be improved substantially. It is found that they need to be correlatively adjusted to achieve the best performance. For instance, the optimal c always increases with the ratio of p_2/p_1 assuming amplifiers with different GBWPs are available. Additionally, a large c value contributes to reducing the energy consumption by the circuit, while amplifying the computation error and the noise impact. These results provide a practical viewpoint for the circuit evaluation, thus representing a guideline for design and improvement of the in-memory linear regression circuit for ML applications, especially in the edge computing scenarios where light ML models are involved.

The rest of this paper is organized as follows. Section II elaborates the transfer characteristics of in-memory linear

This work was supported in part by the National Key R&D Program of China under Grant 2020YFB2206001, in part by NSFC under Grant 62004002 and Grant 92064004, and in part by the 111 Project under Grant B18001.

S. Wang, Z. Sun, Y. Liu, S. Bao, Y. Cai and R. Huang are with Peking University, Beijing 100871, China (e-mail: zhong.sun@pku.edu.cn).

S. Wang is with Nanjing University, Nanjing 210023, China.

D. Ielmini is with Politecnico di Milano, Milano 20133, Italy.

regression circuit and demonstrates the deterministic role of the minimal eigenvalue in the QEP, namely the dominant pole in the temporal behavior of the circuit. Section III introduces optimization schemes to reduce the computing time of the circuit, by adopting the optimal combination of circuit variables that enables the highest frequency for the dominant pole. In Section IV, issues including power consumption and circuit noises are discussed in terms of circuit variables. Finally, this work is concluded in Section V.

II. Computing Time Analysis of In-Memory Linear Regression Circuit

A. Transfer Characteristics

Linear regression is an overdetermined linear system problem, which can be formulated as a matrix equation, that is

$$\boldsymbol{X\omega} = \boldsymbol{y}, \tag{1}$$

where $\boldsymbol{X}$ is an $n \times m$ $(n > m)$ matrix containing a column of ones and $m-1$ columns of independent variable values of n samples, $\boldsymbol{y}$ is an $n \times 1$ vector reporting values of the dependent variable, and $\boldsymbol{\omega}$ is the $m \times 1$ weight vector of fitting coefficients to be solved [11]. As matrix $\boldsymbol{X}$ is non-square and hence non-invertible, Eq. (1) is solved with the pseudoinverse concept, namely

$$\boldsymbol{\omega} = \boldsymbol{X}^{+}\boldsymbol{y} = (\boldsymbol{X}^{T}\boldsymbol{X})^{-1}\boldsymbol{X}^{T}\boldsymbol{y}, \tag{2}$$

where $\boldsymbol{X}^{+}$ is the pseudoinverse (or Moore-Penrose inverse [12]) of $\boldsymbol{X}$, $\boldsymbol{X}^{T}$ is the transpose. To facilitate the closed form of Eq. (2), it is mapped in a feedback circuit constituted by two identical crosspoint memristor arrays, one set of transimpedance amplifiers (TIAs), and one set of positive feedback amplifiers (PFAs). In Fig. 1, matrix $\boldsymbol{X}$ is mapped by the conductance matrices $\boldsymbol{G_X}$ of two identical crosspoint memristor arrays. During the computing operation, the left and right arrays play the role of $\boldsymbol{X}$ and $\boldsymbol{X}^{T}$, respectively. Vectors

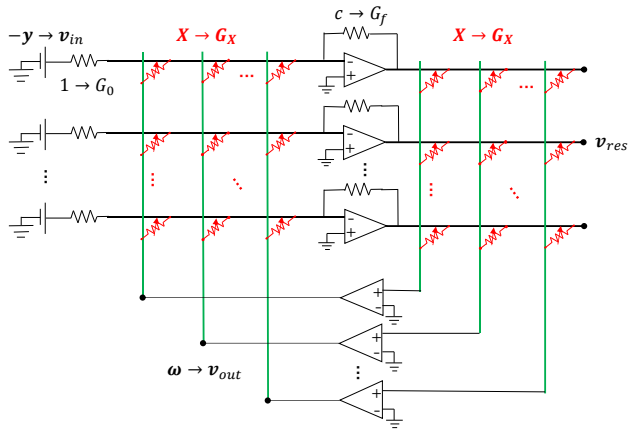

Fig. 1. The in-memory linear regression circuit based on two identical $n \times m$ crosspoint memristor arrays and feedback loops that enabled by one set of TIAs and one set of PFAs. The parameters in a linear regression problem, namely the known matrix $\boldsymbol{X}$, the known vector $\boldsymbol{y}$ and the unknown $\boldsymbol{\omega}$ are mapped to $\boldsymbol{G_X}$, $\boldsymbol{v_{in}}$ and $\boldsymbol{v_{out}}$ in the circuit, respectively. The input resistors are of a unit conductance $G_0 = 10\mu S$, which is 10 μS in this work. The feedback conductance of TIAs is assumed as $G_f = cG_0$. The output voltages of TIAs are termed as a column vector $\boldsymbol{v}_{res}$, as it represents the least squares error in linear regression.

$-\boldsymbol{y}$ and $\boldsymbol{\omega}$ are mapped by the input and output voltage vectors ($\boldsymbol{v}_{in}$ and $\boldsymbol{v}_{out}$), respectively. Once the known vector $\boldsymbol{y}$ is provided in the circuit, the static outputs of PFAs define the fitting coefficients of the corresponding linear regression model, namely the unknown vector $\boldsymbol{\omega}$ is solved. Notably, though the feedback conductance G_f (representing a scalar constant c) of TIAs is not involved in the static result, it plays an important role in the dynamic behavior as will be revealed later. Also, the output voltages $\boldsymbol{v}_{res}$ of TIAs represent essentially the residuals of data fitting divided by the constant c, namely $\boldsymbol{v}_{res} \propto (\boldsymbol{y} - \boldsymbol{X\omega})/c$, the sum of squares of which is termed the least squares. As c modulates output voltages of TIAs, it is anticipated to play an important role in issues of power consumption and computation accuracy of the circuit.

To study the computing time of the circuit, its transfer function, *i.e.*, the relationship between the output vector $\boldsymbol{v}_{out}$ and the input vector $\boldsymbol{v}_{in}$ shall be established. In Fig. 1, according to Kirchhoff's law and amplifier theory, the input-output equations of the ith $(1 \le i \le n)$ TIA and the jth $(1 \le j \le m)$ PFA are written respectively as follows:

$$-\frac{G_0 v_{in,i}(s) + G_f v_{res,i}(s) + \sum_j G_{X,ij} v_{out,j}(s)}{G_0 + G_f + \sum_j G_{X,ij}} L_1(s) = v_{res,i}(s), \tag{3}$$

$$\frac{\sum_i G_{X,ij} v_{res,i}(s)}{\sum_i G_{X,ij}} L_2(s) = v_{out,j}(s), \tag{4}$$

where s is the complex frequency, $L_1(s)$ and $L_2(s)$ are open-loop gains of OAs of TIAs and PFAs, respectively. Combining n equations of TIAs or m equations of PFAs results in the following two matrix equations in frequency domain:

$$-\boldsymbol{U}_n[\boldsymbol{v}_{in}(s) + c\boldsymbol{v}_{res}(s) + \boldsymbol{X}\boldsymbol{v}_{out}(s)]L_1(s) = \boldsymbol{v}_{res}(s), \tag{5}$$

$$\boldsymbol{U}_m\boldsymbol{X}^{T}\boldsymbol{v}_{res}(s)L_2(s) = \boldsymbol{v}_{out}(s), \tag{6}$$

where the dimensionless matrix $\boldsymbol{X}$ and constant c have replaced $\boldsymbol{G_X}$ and G_f, respectively. Matrices $\boldsymbol{U}_n$ and $\boldsymbol{U}_m$ are dimensionless as well. Specifically, $\boldsymbol{U}_n$ is an $n \times n$ diagonal matrix with diagonal entries of $\frac{1}{1+c+\sum_j X_{ij}}$ $(1 \le i \le n)$, $\boldsymbol{U}_m$ is an $m \times m$ diagonal matrix with diagonal entries of $\frac{1}{\sum_i X_{ij}}$ $(1 \le j \le m)$. Incorporating Eq. (6) in Eq. (5), and considering the single-pole model for both sets of OAs [13], namely $L_1(s) = \frac{L_{01}}{1+s/\omega_{01}}$ for TIAs and $L_2(s) = \frac{L_{02}}{1+s/\omega_{02}}$ for PFAs, where L_{01} and L_{02} are the DC open-loop gains, ω_{01} and ω_{02} are the 3-dB bandwidths, the matrix equation becomes

$$[L_{01}\omega_{01}L_{02}\omega_{02}\boldsymbol{U}_n\boldsymbol{X}\boldsymbol{U}_m\boldsymbol{X}^{T} + cL_{01}\omega_{01}\omega_{02}\boldsymbol{U}_n + \omega_{01}\omega_{02}\boldsymbol{I}_n + (cL_{01}\omega_{01}\boldsymbol{U}_n + \omega_{01}\boldsymbol{I}_n + \omega_{02}\boldsymbol{I}_n)s + \boldsymbol{I}_n s^2]\boldsymbol{v}_{res}(s) = -L_{01}\omega_{01}(s + \omega_{02})\boldsymbol{U}_n\boldsymbol{v}_{in}(s), \tag{7}$$

where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. As L_{01} and L_{02} are usually much greater than 1, *e.g.*, 10^5, minor terms in the coefficient matrix for each order of s in Eq. (7) can be omitted, resulting in the following equation

$$\boldsymbol{v}_{res}(s) = -\left(L_{02}\omega_{02}\boldsymbol{U}_n\boldsymbol{X}\boldsymbol{U}_m\boldsymbol{X}^{T} + c\boldsymbol{U}_n s + \frac{1}{L_{01}\omega_{01}}\boldsymbol{I}_n s^2\right)^{-1} \cdot (s + \omega_{02})\boldsymbol{U}_n\boldsymbol{v}_{in}(s), \tag{8}$$

which tells the relationship between $\boldsymbol{v}_{res}$ and $\boldsymbol{v}_{in}$. To obtain

the expression of $\boldsymbol{v}_{out}$, Eq. (8) is substituted backwards in Eq. (6), which finally becomes

$$\boldsymbol{v}_{out}(s) = -\boldsymbol{U}_m\boldsymbol{X}^T \cdot \left(\boldsymbol{U}_n\boldsymbol{X}\boldsymbol{U}_m\boldsymbol{X}^T + \frac{c}{L_{02}\omega_{02}}\boldsymbol{U}_n s + \frac{1}{L_{01}\omega_{01}L_{02}\omega_{02}}\boldsymbol{I}_n s^2\right)^{-1} \cdot \boldsymbol{U}_n\boldsymbol{v}_{in}(s). \quad (9)$$

Eq. (9) identifies explicitly the transfer characteristics of the circuit, involving both matrix multiplication and inversion terms. Let $p_1 = L_{01}\omega_{01}$ and $p_2 = L_{02}\omega_{02}$, which are the GBWPs of two sets of OAs, respectively, the transfer function is expressed as

$$\boldsymbol{F}(s) = -\boldsymbol{U}_m\boldsymbol{X}^T\left(\boldsymbol{U}_n\boldsymbol{X}\boldsymbol{U}_m\boldsymbol{X}^T + \frac{c}{p_2}\boldsymbol{U}_n s + \frac{\boldsymbol{I}_n}{p_1 p_2}s^2\right)^{-1}\boldsymbol{U}_n. \quad (10)$$

Based on the transfer function, the stability and response time of the circuit can be addressed. In Eq. (10), the matrix inversion part determines the poles of the circuit, that is, letting the determinant of the matrix be zero defines all pole positions in the complex plane, namely

$$\det\left(\boldsymbol{U}_n\boldsymbol{X}\boldsymbol{U}_m\boldsymbol{X}^T + \frac{c}{p_2}\boldsymbol{U}_n s + \frac{\boldsymbol{I}_n}{p_1 p_2}s^2\right) = 0. \quad (11)$$

Let $\lambda = \frac{s}{p_1}$, Eq. (11) becomes a QEP, that is $\det\left(\frac{p_2}{p_1}\boldsymbol{U}_n\boldsymbol{X}\boldsymbol{U}_m\boldsymbol{X}^T + c\lambda\boldsymbol{U}_n + \lambda^2\boldsymbol{I}_n\right) = 0$, or

$$\det\left(\frac{p_2}{p_1}\boldsymbol{X}\boldsymbol{U}_m\boldsymbol{X}^T + c\lambda\boldsymbol{I}_n + \lambda^2\boldsymbol{U}_n^{-1}\right) = 0, \quad (12)$$

with the factor matrix $\boldsymbol{U}_n$ dropped out for all three terms.

In Eq. (12), it is easily recognized that matrix $\boldsymbol{X}\boldsymbol{U}_m\boldsymbol{X}^T$ whose size is $n \times n$ is positive semi-definite, with m positive eigenvalues and $n - m$ zero eigenvalues. Also, matrices $\boldsymbol{I}_n$ and $\boldsymbol{U}_n^{-1}$ are positive definite, with all n eigenvalues being positive in both cases. According to the QEP theory [14], the solution to Eq. (12) satisfies $Re(\lambda) \leq 0$, suggesting the circuit is always stable for any concerned matrix $\boldsymbol{X}$, which in turn determines $\boldsymbol{U}_n$ and $\boldsymbol{U}_m$.

To solve Eq. (12) thus obtaining explicitly the eigenvalues, it is converted into an eigenvalue problem of a composed matrix, that is

$$\boldsymbol{M}\boldsymbol{\xi} = \lambda\boldsymbol{\xi},\ \boldsymbol{M} = \begin{bmatrix} -c\boldsymbol{U}_n & -\frac{p_2}{p_1}\boldsymbol{U}_n\boldsymbol{X}\boldsymbol{U}_m\boldsymbol{X}^T \\ \boldsymbol{I}_n & \boldsymbol{O}_n \end{bmatrix}, \quad (13)$$

where $\boldsymbol{O}_n$ is the $n \times n$ zero matrix, $\boldsymbol{\xi} = \begin{bmatrix}\lambda\boldsymbol{u} \\ \boldsymbol{u}\end{bmatrix}$, and $\boldsymbol{u}$ is the corresponding eigenvector for Eq. (12). The resulting matrix $\boldsymbol{M}$ is of size $2n \times 2n$ and has $2n$ eigenvalues in total, among which $n + m$ eigenvalues show $Re(\lambda) < 0$, and the remaining $n - m$ eigenvalues are zero, as will be indicated with an example later. The eigenvalue solutions can also be understood in analogy to the solution of a quadratic equation with one unknown [14]. From the hardware viewpoint, the $n + m$ non-zero eigenvalues (hence poles) are related to the $n + m$ single-pole OAs in the circuit, while the $n - m$ zero eigenvalues are due to the count mismatch of two sets of OAs. Thanks to its special structure, matrix $\boldsymbol{M}$ is diagonalizable, which means the Jordan blocks corresponding to each zero eigenvalue are scalar 1×1 blocks, thus guaranteeing the circuit stability in the sense of Lyapunov [15].

B. Computing Time Analysis with Real-World Datasets

The non-zero eigenvalues of matrix $\boldsymbol{M}$ determine $n + m$ poles ($s = \lambda p_1$) of the circuit in the left half complex plane. The response time of a system is often characterized by the dominant pole (or pair) s_d, corresponding to the minimal absolute eigenvalue $|Re(\lambda)|_{min}$ (or minimal real part of a complex conjugate pair of eigenvalues) of matrix $\boldsymbol{M}$ [15]. The larger the $|Re(\lambda)|_{min}$, the faster the circuit response.

To study the computing time of the in-memory linear regression circuit, especially its dependence on $|Re(\lambda)|_{min}$ (or $|Re(s_d)|$), we consider a multiple linear regression problem for PM2.5 prediction in Beijing. The dataset includes 6 attributes (concentrations of PM10, SO_2, NO_2, CO and O_3, temperature), and 1 dependent variable (concentration of PM2.5), in the time period from March 2013 to March 2017 [16]. Different time segments from 1 month to 6 months were selected to test the response time of the circuit.

Fig. 2a shows the attribute matrix $\boldsymbol{X}$ of one month (March 2014) in Eq. (1), which is of size 30×7, representing 30 samples (30 rows), 6 attributes (6 columns), and 1 additional column of ones for linear regression problem formulization [11]. The matrix has been pre-processed to shift all attribute values positive and ensure they are in a reasonable range to be ready for mapping within the conductance limits of memristors, *e.g.*, conductance ratio of 10^2 or 10^3 [17]-[19]. The input vector $\boldsymbol{y}$ representing the PM2.5 concentration is shown as well. It is scaled to generate moderate output voltages for $\boldsymbol{\omega}$, *e.g.*, in the range of ± 0.5 V [20], protecting the crosspoint devices from electrical destruction or degradation.

In the linear regression circuit (Fig. 1), matrix $\boldsymbol{X}$ is mapped in the two crosspoint memristor arrays, with 10 μS being the conductance unit [21]. It is assumed with an 8-bit precision to accurately program the memristor states according to matrix elements [22]-[24]. The vector $-\boldsymbol{y}$ is provided as input voltages. We studied the transient behavior of the circuit in

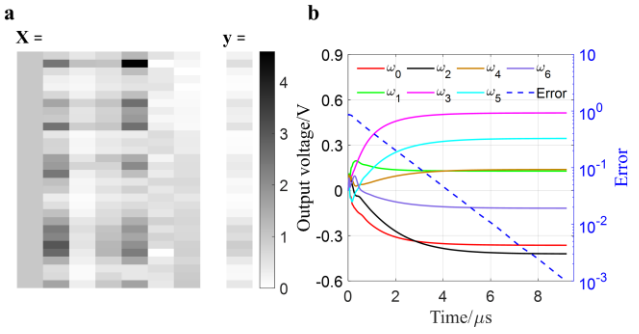

Fig. 2. Linear regression of a real-world dataset. (a) The pretreated attribute matrix $\boldsymbol{X}$ and dependent variable vector $\boldsymbol{y}$ for linear regression calculation. The size of $\boldsymbol{X}$ is 30×7, including the first column of ones and other 6 columns for 6 contributes respectively. The size of $\boldsymbol{y}$ is 30×1 correspondingly. $\boldsymbol{X}$ and $\boldsymbol{y}$ are mapped to the in-memory linear regression circuit for SPICE simulation. (b) Simulated transient output voltages of PFAs in the circuit, representing the 7 linear regression weights (left y-axis), and the corresponding computation error at each moment (right y-axis). Around 9.2 μs, it reaches the desired computing accuracy. During the simulation, a real-world OA model was used for both TIAs and PFAs. The feedback conductance of TIAs was assumed as G_0, namely $c = 1$.

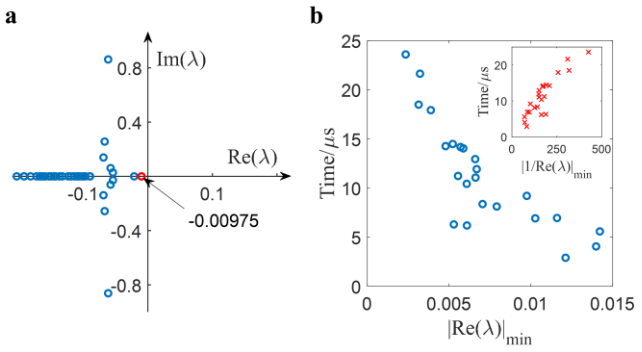

Fig. 3. (a) Non-zero eigenvalues of Eq. (13) for the matrix $\boldsymbol{X}$ in Fig. 2a. There are 37 eigenvalues in total, including 29 real eigenvalues and 4 complex conjugate pair of eigenvalues, all of which satisfy $Re(\lambda) < 0$. The minimal (absolute) eigenvalue is real, and it is labeled in the plane. (b) Computing time of the in-memory linear regression circuit as a function of the minimal eigenvalue $|Re(\lambda)|_{min}$, surveyed with 23 datasets of 1 month for PM2.5 prediction in Beijing. The computing time shows a reciprocal dependence on $|Re(\lambda)|_{min}$, which is also illustrated as a linear dependence on $1/|Re(\lambda)|_{min}$ in the inset. For all the simulations, the same circuit conditions including feedback conductance of TIAs and OA models were used.

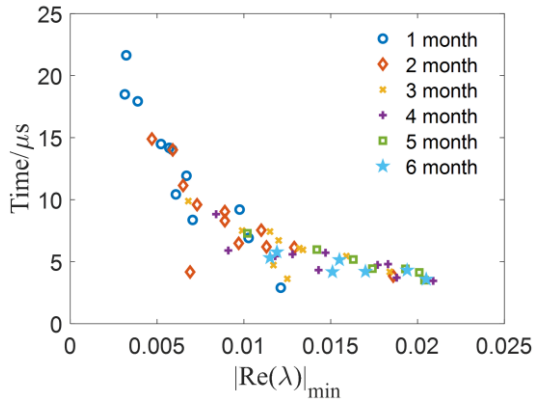

Fig. 4. Computing time dependence on $|Re(\lambda)|_{min}$, for various dataset sizes, including 12 datasets of 1 month, 12 of 2 months, 10 of 3 months, 10 of 4 months, 7 of 5 months, 7 of 6 months. The data of 12 datasets of 1 month are withdrew from Fig. 3b, following the same distribution. For all the simulations, except for the size difference of crosspoint memristor arrays, the circuit conditions were kept the same.

SPICE, and its result is shown in Fig. 2b. By defining the computation error as the Euclidean distance between the static result $\boldsymbol{v}_{out}^{*}$ and the dynamic output $\boldsymbol{v}_{out}(t)$, it is shown that the circuit computation takes around 9.2 μs to achieve the desired precision, i.e., the error falls below 10^{-3}.

We solved Eq. (13) for the matrix in Fig. 2a, to obtain the eigenvalues for the QEP describing the circuit, especially the minimal eigenvalue $|Re(\lambda)|_{min}$. As shown in Fig. 3a, there are 37 non-zero eigenvalues for this matrix, which is consistent with precious analysis in terms of matrix $\boldsymbol{M}$. The $|Re(\lambda)|_{min}$ corresponding to the dominant pole is 9.75×10^{-3}. To support the key role of $|Re(\lambda)|_{min}$ in the circuit speed, we tested different time periods of 1 month, whose linear regression matrices as in Fig. 2a are different as a result. Hence, the circuit for each time period would have a different distribution of eigenvalues, *i.e.*, a different $|Re(\lambda)|_{min}$. 23 time periods in total were used in simulation. All the resulting circuits were simulated in SPICE, and the computing times were recorded in Fig. 3b, as a function of $|Re(\lambda)|_{min}$. It clearly shows a reciprocal relationship between computing time and $|Re(\lambda)|_{min}$, thus demonstrating the dominant role of s_d in the computing time of the circuit. A proportional relationship between time and $1/|Re(\lambda)|_{min}$ is shown in the inset of Fig. 3b as well.

To test the dependence of computing time on the problem size *N*, we have considered the linear regression of other time periods of more than 1 month, namely 60 days, 90 days, 120 days, 150 days or 180 days, resulting in linear regression problems with different matrix sizes. The computing time in each simulation was recorded. Meanwhile, the $|Re(\lambda)|_{min}$ was calculated by solving the eigenvalues in Eq. (13). Fig. 4 shows the relationship between computing time and $|Re(\lambda)|_{min}$ for different matrix sizes. It evidences the dominant role of s_d in computing time holds. Due to the tighter distribution of $|Re(\lambda)|_{min}$ for larger matrices, the computing time is accordingly more concentrated, while it does not necessarily increase with the problem size, supporting the feasibility of in-memory linear regression computation for different datasets in a constant time. It is worth mentioning that for all the simulation cases, the number of system poles is equal to the number of OAs in the circuit, and the resulting matrix $\boldsymbol{M}$ in Eq. (13) can be diagonalized, thus guaranteeing the stable outputs of the circuit .

III. Optimization Scheme For Computing Time

In the above simulations, a real OA model (AD823, Analog Devices) with a GBWP of 16 MHz was used for both TIAs and PFAs, and the feedback conductance of TIAs was intuitively considered as the unit conductance, *i.e.*, $G_f = 10$ μS (or $c = 1$). In Eq. (13), matrix $\boldsymbol{M}$ is defined based on parameters including c, p_1 and p_2, suggesting their deterministic role for eigenvalues of $\boldsymbol{M}$. Consequently, variables in the circuit including feedback conductance and GBWPs shall play a critical role for the dominant pole. We studied on these parameters and optimized accordingly the computing time of the circuit by maximizing the $|Re(\lambda)|_{min}$.

As an initial attempt, we investigated solely the impact of c, whose sweep range was assumed ideally as $[10^{-2}, 10^{2}]$. The real OA model was retained for both sets of amplifiers, namely $p_1 = p_2 = 16$ MHz, based on which the $|Re(\lambda)|_{min}$ was solved for each c. The result is shown in Fig. 5a, where the maximum of $|Re(s_d)|$ arises at $c = 0.56$, with a specific frequency of 0.48 MHz corresponding to $|Re(\lambda)|_{min} = 0.03$. On both sides of the optimal point, there is a dramatic degradation of $|Re(s_d)|$, which will ultimately translate to a significant delay of circuit response. We mapped the optimal c in the circuit for simulation. The transient result is shown in Fig. 5b, from which the computing time is calculated to be 3.9 μs, according to the criterion defined before. Compared to the $c = 1$ case in Fig. 3a, the circuit response is improved by more than twice. Such a speed-up is also consistent with the optimization of $|Re(\lambda)|_{min}$. Therefore, by simply optimizing the feedback conductance of TIAs, it is possible to accelerate

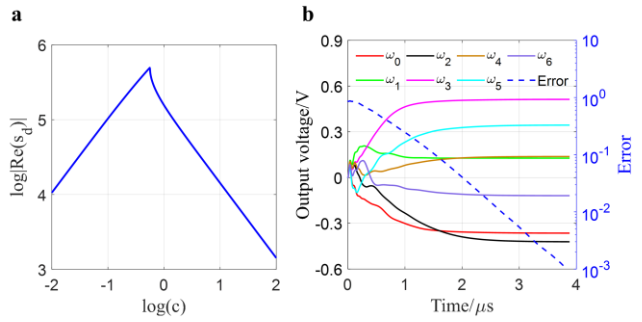

Fig. 5. (a) The calculated dominant pole s_d a function of parameter c, assuming $p_1 = p_2 = 16$ MHz. s_d was calculated by solving Eq. (13) and multiplying $|Re(\lambda)|_{min}$ with p_1. (b) Simulated transient output voltages of PFAs in the circuit with the optimal c in (a), namely $c = 0.56$, representing the 7 linear regression weights (left y-axis), and the corresponding computation error at each moment (right y-axis). Around 3.9 μs, it reaches the desired computing accuracy. During the simulation, the same OA model was retained.

remarkably the computing speed of the circuit.

To further improve the dominant pole and hence the circuit speed, we considered optimizing the GBWP of one set of amplifiers, while the other one remaining fixed. Specifically, $p_1 = 16$ MHz of TIAs was unchanged, p_2 of PFAs and c were optimized simultaneously. The bandwidth range of p_2 is reasonably considered as [10^{-1}, 10^3] MHz [25]-[27]. The result is shown in Fig. 6a, indicating that there is a line of optimal combination of c and p_2 to maximize the dominant pole, as represented by the red/black dots in the figure. Fig. 6b shows 5 situations of fixed p_2 values, where in each case, the dominant pole peaks at a specific c value, and decreases obviously on both sizes of the peak, representing the best value of c for each given p_2. Moreover, such an optimal c

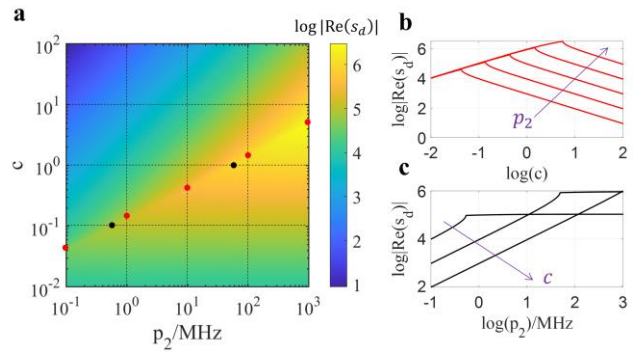

Fig. 6. Optimized pairs of p_2 and c, for a given $p_1 = 16$ MHz. (a) Distribution of dominant pole for p_2 in the range [10^{-1}, 10^3] MHz and c in the range of [10^{-2}, 10^2]. The red dash lines are for tracing the trend of dominant pole for a given p_2, meanwhile the red dots illustrate the maximum of dominant pole for varying c. The black dash lines are for tracing the trend of dominant pole for a given c, meanwhile the black dots illustrate the turning point of dominant pole for varying p_2. (b) The behavior of dominant pole as c increases (both in logarithmic scale), for 5 situations with p_2 fixed, namely $p_2 = 0.1, 1, 10, 100$ or 1000 MHz. For each p_2, there is an optimal c that maximizes the dominant pole, and such an optimal c increases with p_2. (c) The behavior of dominant pole as p_2 increases (both in logarithmic scale), for 3 situations with c fixed, namely $c = 0.1, 1$ or 10. For $c = 0.1$ or 1, there is a critical p_2 after which the dominant pole is almost saturated, and it increases with c. Due to the limited p_2 range, the critical point for $c = 10$ is not observed.

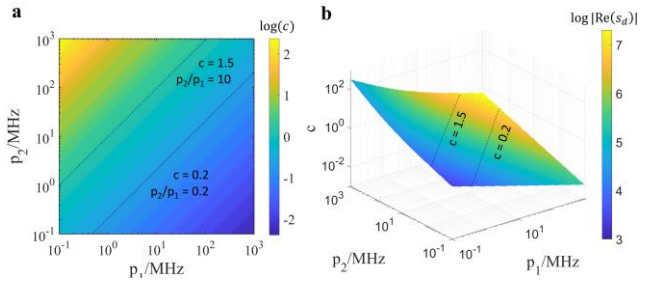

Fig. 7. Optimized combinations of three parameters. (a) Optical c for pairs of p_1 and p_2, both in the range of [10^{-1}, 10^3] MHz. Skew diagonal stripes in different colors appear in the plane, indicating that there is a (roughly) monotonic relationship between p_2/p_1 and c. Two stripes are labeled for illustrations. (b) Dominant poles corresponding to the optimal combination of p_1, p_2 and c. For a specific p_2/p_1 and the corresponding optimal c, the larger the GBWP, the greater the dominant pole.

increases monotonically with p_2. On the other hand, for a given c, the dominant pole does not peak, rather it reaches a plateau when p_2 is greater than a critical value, which is helpful to relaxing the constraint on OA parameters for circuit implementation. Again, such a critical p_2 increases with c, cross-validating the synergisitic effect between p_2 and c for dominant pole optimization.

We explored the full space of all the three parameters c, p_1 and p_2 for circuit optimization, where the bandwidth range of [10^{-1}, 10^3] MHz was swept for both p_1 and p_2, and each combination the optimal c was searched. The results are shown in Fig. 7a, which describes the optimal c for every combination of p_1 and p_2. It is shown that the optimal c increases with p_2 for any given p_1, which is a generalized conclusion of the results in Fig. 6. On the other hand, for a given p_2, the optimal c decreases with p_1 in contrast. The underlying reason is that there is an optimal c for a p_2/p_1 ratio, regardless of the specific values of p_1 or p_2, as indicated by the skew diagonal stripes in the plane. It can be recognized that the optimal c increases with the ratio of p_2/p_1, as exemplified with two situations where $c = 1.5$ and $c = 0.2$ correspond to p_2/p_1 ratio of 10 and 0.2, respectively. For the optimized combination of all three parameters, the resulting dominant pole is shown in Fig. 7b. While the optimal c increases with the ratio of p_2/p_1, the corresponding dominant pole does not necessarily increase, rather it remains less affected along the diagonal of the horizontal plane. The dominant pole is more affected by the choice of p_1 or p_2, due to the fundamental impact of GBWP parameters on poles, as shown by Eq. (11).

There is a monotonic relationship between the optimal c and p_2/p_1, which can be figured in the context of the eigenvalue problem in Eq. (13), where p_2/p_1 plays an individual role to determine the eigenvalues of matrix $\boldsymbol{M}$. Fig. 8a shows the optimal p_2/p_1 as a function of c and p_2. For a given c, the p_2/p_1 ratio is almost constant regardless of the p_2 value, as evidenced explicitly by three trajectories of different c values in Fig. 8b. Again, the optimal p_2/p_1 increases with c, as have been shown in the above analysis.

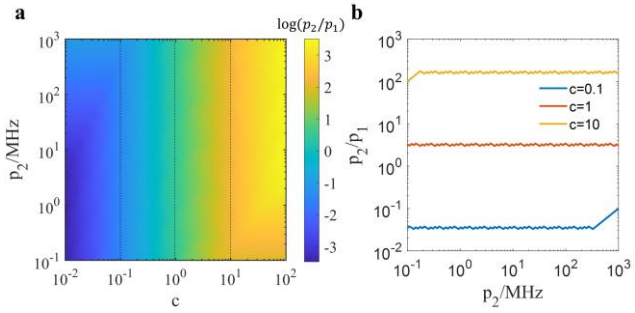

Fig. 8. (a) Heat map of $\log(p_2/p_1)$ as a function of c, p_2, and the corresponding optimal p_1. The dash lines are for indicating the constant p_2/p_1 for each given c, which is 0.1, 1, or 10, respectively. (b) Optimal p_2/p_1 values for the three c values. During the search for an optimal p_1, it was constrained in the range of [10^{-3}, 10^{4}] MHz for realistic considerations. As a result, it was the upper (or lower) boundary instead of the optimal p_1 was obtained in some cases for $c = 0.1$ (or for $c = 10$).

These results provide a guidance for computing speed optimization for the in-memory linear regression circuit. The optimal c (feedback conductance of TIAs) increases with the ratio of p_2/p_1 (GBWP ratio of two sets of OAs), or verse vice. Under this constraint, increasing either GBWP shifts the dominant pole of the circuit to higher frequency, achieving a faster response. In the above analysis, the results have been limited to a single 1-month dataset. To support the generality of the conclusion, we have also tested more datasets with different time lengths regarding the optimization schemes of circuit speed, and the results in terms of these parameters in the circuit are similar.

IV. Power Dissipation and Noise Analysis

In Fig. 1, feedback conductance of TIAs, *i.e.*, parameter c scales the residuals of data fitting, which are represented by the output voltages of TIAs. If c is less than 1, the output voltages will be amplified, causing an increasing power consumption by the right crosspoint memristor array. On the other hand, if c is large, the output voltages will be reduced. Though it may be good for power consumption, the impact of noise in the circuit is magnified in this case, due to the low signal-to-noise ratio. As a result, there is a tradeoff

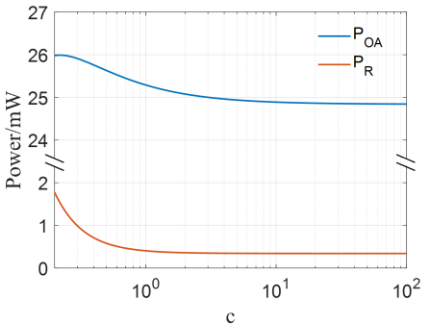

Fig. 9. Impact of parameter c on the power consumption of the circuit. It consists of two parts P_R and P_{OA}, consumed by resistive devices (crosspoint memristor arrays and input resistors) and OAs (TIAs and PFAs) in the circuit, respectively. c was swept in the range of [0.2, 100]. The estimation is based on a dataset of 3 months, namely the crosspoint array is of size 89×7.

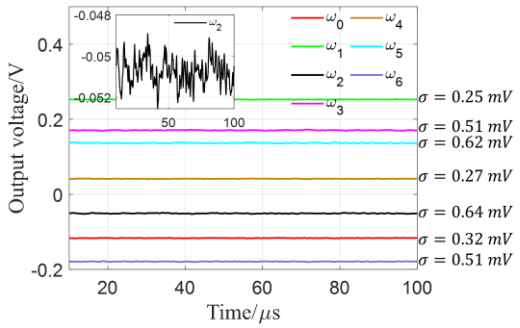

Fig. 10. DC output voltages of PFAs in the circuit with noise sources added. The circuit simulation is based on a 89×7 linear regression matrix of a 3-month dataset. The standard deviation σ of the fluctuation is labeled for each output voltage.

consideration for c in terms of power consumption, noise issue as well as the above dominant pole analysis.

To analyze the impact of c on power consumption and circuit noise, we considered a relatively large dataset, *i.e.*, a 3-month dataset, resulting in two 89×7 crosspoint memristor arrays in the circuit. Also, we constrain the c value in a more practical range, as the memory devices in the right array might be destructed by amplified output voltages of TIAs. By assuming the maximal output voltage is 1 V, the lower bound of c is calculated to be 0.2 in this case, which means the amplification of TIA output voltages will be less than 5. The power consumption of the circuit consists of two parts, bared by the resistive devices (P_R) and OAs (P_{OA}), respectively. P_R is calculated as

$$P_R = \sum_{i=1}^{n} v_{in,i}^2 G_0 + \sum_{j=1}^{m}[v_{out,j}^2 \sum_{i=1}^{n} G_{X,ij}] + \sum_{i=1}^{n}[v_{res,i}^2 (G_f + \sum_{j=1}^{m} G_{X,ij})], \quad (14)$$

where the three terms are contributed by input resistors, left crosspoint memristor array, and right crosspoint memristor array together with feedback resistors, respectively. P_{OA} is calculated as [28]

$$P_{OA} = \sum_{i=1}^{n}\left[V_s I_q + |v_{res,i}|\left(V_{CC} - |v_{res,i}|\right)\left(G_f + \sum_{j=1}^{m} G_{X,ij}\right)\right] + \sum_{j=1}^{m}\left[V_s I_q + |v_{out,j}|\left(V_{CC} - |v_{out,j}|\right)\sum_{i=1}^{n} G_{X,ij}\right], \quad (15)$$

where the two terms are contributed by TIAs and PFAs, respectively. $V_S = V_{CC} - V_{EE}$ is the source voltage of OAs, I_q is the quiescent current of OAs, which is typically assumed as 100 μA [29]. As the transient dynamics are much complicated, *e.g.*, the one in Fig. 5b involving different curves, the power consumption is characterized with static output voltages. The calculation result is shown in Fig. 9. As c increases, both parts of power dissipation consumed by crosspoint memristor arrays and OAs decline, as they both rely on the output voltages of TIAs. For a relatively large c, the power consumption by memristors is around 0.34 mW, while its counterpart by OAs is greater by two orders of magnitude, which might be optimized with more advanced circuit design and process technology for OAs [30], [31].

We have also simulated the circuit with a comprehensive consideration of all noise sources in the circuit components. Specifically, thermal noise and shot noise were included for all resistive devices [32], [33], while white noise and flicker noise were considered for OAs [34], [35]. Fig. 10a shows the

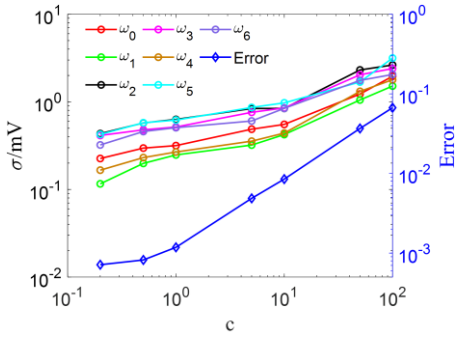

Fig. 11. Impact of c on the σ of output voltage fluctuations (left y-axis) and the overall computing error of the circuit (right y-axis). The computing error is obtained through extracting the mean values of the fluctuated outputs and calculating the Euclidean distance between them and the ideal weights.

static outputs with noise fluctuations for the case of $c = 1$, where the standard deviation for each weight output is remarked. The computation error calculated with the mean values is evaluated to be 0.002 with respect to the analytical solution. As c increases, the output voltages of TIAs are lowered down, thus the noise impact is magnified, which is evidenced in Fig. 10b. As c is increased from 0.2 to 100, the standard deviation of voltage fluctuations is amplified by one order of magnitude, while the solution error increases by two orders of magnitude. Therefore, there is a tradeoff consideration for the parameter c optimization to achieve the best circuit performance in terms of computation speed, power consumption as well as the noise impact. From the viewpoint of power consumption, $c > 1$ is favored. On the other hand, the computation error is acceptable for $c < 10$, *i.e.*, $\sigma < 1\ mV$ and error < 10^{-2}. In the overlap range, the optimal ratio of p_2/p_1 is suggested to be greater than, namely the PFAs have a larger GBWP than the OAs of TIAs. $c < 1$ can be adopted for the purpose of precise computation, at the cost of more power dissipation, together with the consideration of a reduced p_2/p_1 ratio.

V. Conclusion

In this work, we investigated the transfer characteristics of an in-memory analog computing circuit for linear regression calculation with crosspoint memristor arrays. Based on the transfer function of the circuit, a QEP was obtained, where the minimal eigenvalue (or real part of complex conjugate eigenvalues) represents the dominant pole, which in turn dominates the response time of the circuit. In contrast, the problem size does not play an active role in affecting the circuit response. According to the QEP in the matrix form, the minimal eigenvalue is related to parameters in the circuit including feedback conductance (c) of TIAs and GBWPs (p_1 and p_2) of two sets of OAs. By setting up appropriate parameters, the computing speed of the cicuit can be remarkably improved, *e.g.*, by several times faster. Optimization of these parameters should be operated synergisitically to speed up the circuit computation, for instance, the optimal c increases monotonically with the ratio of p_2/p_1. Besides, the parameter c plays also an important role in determining the power consumption and computation error of the circuit, thus suggesting a comprehensive consideration for evaluating the circuit performance. This work provides a guideline for optimizing the computing time, power dissipation, accuracy and noise of this circuit, which is crucial to promote the development of in-memory ML accelerator applications.

References

[1] E. Strubell, A. Ganesh, A. McCallum, "Energy and policy considerations for modern deep learning research," *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, no. 09, pp. 13693-13696, Apr. 2020.

[2] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nat. Nanotechnol.*, vol. 15, pp. 529-544, Mar. 2020.

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

[4] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, H. Qian, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, pp. 641-646, Jan. 2020.

[5] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, J. P.Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, Q. Xia, "Analogue signal and image processing with large memristor crossbars," *Nat. Electron*, vol. 1, pp. 52-59, Dec. 2017.

[6] Z. Sun and R. Huang, "Time complexity of in memory matrix vector multiplication," *IEEE Trans. Circuits Syst. II Express Briefs*, to be published. doi: 10.1109/TCSII.2021.3068764.

[7] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, D. Ielmini, "Solving matrix equations in one step with cross-point resistive arrays," *Proc. Natl. Acad. Sci. USA*, vol. 116, no. 10, pp. 4123-4128, Mar. 2019.

[8] Z. Sun, E. Ambrosi, G. Pedretti, A. Bricalli, D. Ielmini, "In-Memory PageRank Accelerator With a Cross-Point Array of Resistive Memories," *IEEE Trans. Electron Devices*, vol. 67, no. 4, pp. 1466-1470, Mar. 2019.

[9] Z. Sun, G. Pedretti, P. Mannocci, E. Ambrosi, A. Bricalli, D. Ielmini, "Time Complexity of In-Memory Solution of Linear Systems," *IEEE Trans. Electron Devices*, vol. 67, no. 7, pp. 2945-2951, May 2020.

[10] Z. Sun, G. Pedretti, E. Ambrosi, A. Bricalli, D. Ielmini, "In - Memory Eigenvector Computation in Time *O*(1)," *Adv. Intell. Syst.*, vol. 2, no. 8, pp. 2000042, Aug. 2020.

[11] Z. Sun, G. Pedretti, A. Bricalli, D. Ielmini, "One-step regression and classification with cross-point resistive memory arrays," *Sci. Adv.*, vol. 6, pp. eaay2378, Feb. 2020.

[12] R. Penrose, "A generalized inverse for matrices," *Math. Proc. Cambridge Philos*, vol. 51, no. 3, pp. 406-413, Jan. 1955.

[13] B. Razavi, *Design of Analog CMOS Integrated Circuits*. New York, NY, USA: McGraw-Hill, 2001.

[14] F. Tisseur and K. Meerbergen, "The Quadratic Eigenvalue Problem," *SIAM Rev.*, vol. 43, no. 2, pp. 235-286, Jun. 2001.

[15] K. J. Astrom, R. M. Murray, *Feedback systems: an introduction for scientists and engineers* 1th ed., Princeton, NJ, USA: Univ. Princeton Press, 2008.

[16] S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, S. X. Chen, "Cautionary tales on air-quality improvement in Beijing," *Proc. R Soc. A*, vol. 473, no. 2205, pp. 20170457, Sep. 2017.

[17] T.-C. Chang, K.-C. Chang, T.-M. Tsai, T.-J. Chu, S. M. Sze, "Resistance random access memory," *Mater. Today*, vol. 19, no. 5, pp. 254-264, Dec. 2015.

[18] A. Mehonic, A. L. Shluger, D. Gao, I. Valov, E. Miranda, D. Ielmini, A. Bricalli, E. Ambrosi, C. Li, J. J. Yang, Q. Xia, A. J. Kenyon, "Silicon Oxide (SiOx): A Promising Material for resistance switching?," *Adv. Mater*, vol. 30, no. 43, pp. 1801187, Jun. 2018.

[19] Z. Sun, E. Ambrosi, A. Bricalli, and D. Ielmini, "Logic computing with stateful neural networks of resistive switches," *Adv. Mater*, vol.30, no. 38, pp. 1802554, Sep. 2018.

[20] D. Ielmini and G. Pedretti, "Device and Circuit Architectures for In-Memory Computing," *Adv. Intell. Syst.*, vol. 2, no. 7, pp. 2000040, Jul. 2020.

[21] X. Yang, Y. Fang, Z. Yu, Z. Wang, T. Zhang, M. Yin, M. Lin, Y. Yang, Y. Cai, R. Huang, "Nonassociative learning implementation by a single memristor-based multi-terminal synaptic device," *Nanoscale*, vol. 8, pp. 18897-18904, Aug. 2016.

[22] J. Park, M. Kwak, K. Moon, J. Woo, D. Lee, H. Hwang, "TiO_x-based RRAM Synapse with 64-levels of Conductance and Symmetric Conductance Change by Adopting a Hybrid Pulse Scheme for Neuromorphic Computing," *IEEE Electron Device Lett.*, vol. 37, pp. 1559-1562, Dec. 2016.

[23] J. Tang, D. Bishop, S. Kim, M. Copel, T. Gokmen, T. Todorov, S. Shin, K.-T. Lee, P. Solomon, K. Chan, W. Haensch, J. Rozen, "ECRAM as Scalable Synaptic Cell for High-Speed, Low-Power Neuromorphic Computing," *IEEE International Electron Devices Meeting*, pp. 13.1.1-13.1.4, Dec. 2018. doi: 10.1109/IEDM.2018.8614551.

[24] I. Giannopoulos, A. Sebastian, M. Le Gallo, V. P. Jonnalagadda, M. Sousa, M. N. Boon, E. Eleftheriou, "8-bit Precision In-Memory Multiplication with Projected Phase-Change Memory," *IEEE International Electron Devices Meeting (IEDM)*, pp. 2-4, Dec. 2018. doi: 10.1109/IEDM.2018.8614558.

[25] J. F. Cox, *Fundamentals of linear electronics: integrated and discrete*, Albany, USA: Delmar Thomson Learning, 2002.

[26] F. Esparza-Alfaro, S. Pennisi, G. Palumbo, A. J. Lopez-Martin, "Low-Power Class-AB CMOS Voltage Feedback Current Operational Amplifier With Tunable Gain and Bandwidth," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 61, no. 8, pp. 574-578, May 2014.

[27] A. Rahaman, H. Jeong, J. Jang, "A High-Gain CMOS Operational Amplifier Using Low-Temperature Poly-Si Oxide TFTs," *IEEE Trans. Electron Devices*, vol. 67, no. 2, pp. 524-528, Feb. 2020.

[28] B. Carter and R. Mancini, *Op Amps for Everyone*, Amsterdam, The Netherlands: Elsevier, 2009.

[29] H. Zumbahlen, *Linear Circuit Design Handbook*, Amsterdam, The Netherlands: Elsevier, 2008.

[30] R. Hogervorst and J. Huijsing, *Design of Low-Voltage Low-Power Operational Amplifier Cells*, New York, NY, USA: Springer, 2010.

[31] L. Magnelli, F. A. Amoroso, F. Crupi, G. Cappuccino, G. Iannaccone, "Design of a 75-nW, 0.5-V subthreshold complementary metal–oxide–semiconductor operational amplifier," *Int. J. Circuit Theor. Appl.*, vol. 42, no. 9, pp. 967-977, Sep. 2014.

[32] M. Hu *et al.*, "Dot-Product Engine for Neuromorphic Computing: Programming 1T1M Crossbar to Accelerate Matrix-Vector Multiplication," in *Proc. ACM/IEEE Design Automat. Conf. (DAC)*, pp. 1-6, Jun. 2016. doi: 10.1145/2897937.2898010.

[33] Z. He, J. Lin, R. Ewetz, J.-S. Yuan, D. Fan, "Noise injection adaption: End-to-end ReRAM crossbar non-ideal effect adaption for neural network mapping," *Proc. 56th Annu. Design Automat. Conf.*, pp. 1-6, Jun. 2019.

[34] A. Kay, *Operational Amplifier Noise: Techniques and Tips for Analyzing and Reducing Noise*, Amsterdam, The Netherlands: Elsevier, 2012.

[35] G. Giusi, G. Cannatà, G. Scandurra, C. Ciofi, "Ultra-low-noise large-bandwidth transimpedance amplifier," *Int. J. Circuit Theor. Appl.*, vol. 43, no. 10, pp. 1455-1473, Oct. 2015.