IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

Received 21 December 2022; accepted 5 April 2023. Date of publication 10 April 2023; date of current version 9 June 2023.

Digital Object Identifier 10.1109/JXCDC.2023.3265803

A Generalized Block-Matrix Circuit for Closed-Loop Analog In-Memory Computing

PIERGIULIO MANNOCCI[1] (Graduate Student Member, IEEE), and DANIELE IELMINI[1] (Fellow, IEEE)

[1]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano and IU.NET, 20133 Milan, Italy.

CORRESPONDING AUTHORS: P. MANNOCCI; D. IELMINI (piergiulio.mannocci@polimi.it; daniele.ielmini@polimi.it)

This project has received funding from the European Research Council under grant no. 101054098.

This article has supplementary downloadable material available at https://doi.org/10.1109/JXCDC.2023.3265803, provided by the authors.

ABSTRACT Matrix-based computing is ubiquitous in an increasing number of present-day machine learning applications such as neural networks, regression, and 5G communications. Conventional systems based on von-Neumann architecture are limited by the energy and latency bottleneck induced by the physical separation of the processing and memory units. In-memory computing (IMC) is a novel paradigm where computation is performed directly within the memory, thus eliminating the need for constant data transfer. IMC has shown exceptional throughput and energy efficiency when coupled with crosspoint arrays of resistive memory devices in open-loop matrix-vector-multiplication and closed-loop inverse-matrix-vector multiplication (IMVM) accelerators. However, each application results in a different circuit topology, thus complicating the development of reconfigurable, general-purpose IMC systems. In this article, we present a generalized closed-loop IMVM circuit capable of performing any linear matrix operation by proper memory remapping. We derive closed-form equations for the ideal input-output transfer functions, static error, and dynamic behavior, introducing a novel continuous-time analytical model allowing for orders-of-magnitude simulation speedup with respect to SPICE-based solvers. The proposed circuit represents an ideal candidate for general-purpose accelerators of machine learning.

INDEX TERMS Hardware accelerator, in-memory computing (IMC), linear algebra, linear regression, machine learning, resistive memory.

I. INTRODUCTION

IN-MEMORY computing (IMC) has gained traction as a promising candidate to overcome the von-Neumann bottleneck by eliminating the separation between the memory and processing units [1]. IMC executes algebraic operations by physical laws in crosspoint memory arrays, thus allowing for low-power, high-density, and high-throughput computation [2]. Machine learning [3], [4], image processing [5], combinatorial optimization [6], and baseband processing [7] are some of the representative examples demonstrating the IMC potential as next-generation computing architecture [8]. Experimental demonstrations of IMC accelerators for matrix–vector multiplication (MVM) have been reported in the latest years [9] improving the computational complexity toward the attractive $\mathcal{O}(1)$ limit [10]. MVM alone is, however, insufficient to build a comprehensive IMC algebraic accelerator. In a growing number of applications, inverse-matrix-vector multiplication (IMVM) is needed alongside MVM to carry out tasks of increasing complexity. To relieve the dependence of iterative solvers [11] exploiting open-loop MVM on external coprocessors, several memory-agnostic IMC-IMVM accelerators have been proposed [7], [12], [13], [14], [15], [16] exploiting closed-loop, feedback-based topologies to implement the inverse operation. Closed-loop IMVM allows to vastly reduce the computation complexity of inverse computation from $\mathcal{O}(n^3)$ to $\mathcal{O}(1)$ [17]. Nonetheless, the different requirements of inverse problems, ranging from simple matrix inversion [12], to linear regression [4], [13] and regularized regressions [7], required the design and implementation of ad hoc topologies, limiting the generality of the proposed solutions.

In this article, we introduce a general-purpose, reconfigurable closed-loop IMVM universal circuit, namely the *block circuit*, capable of solving any linear matrix operation.

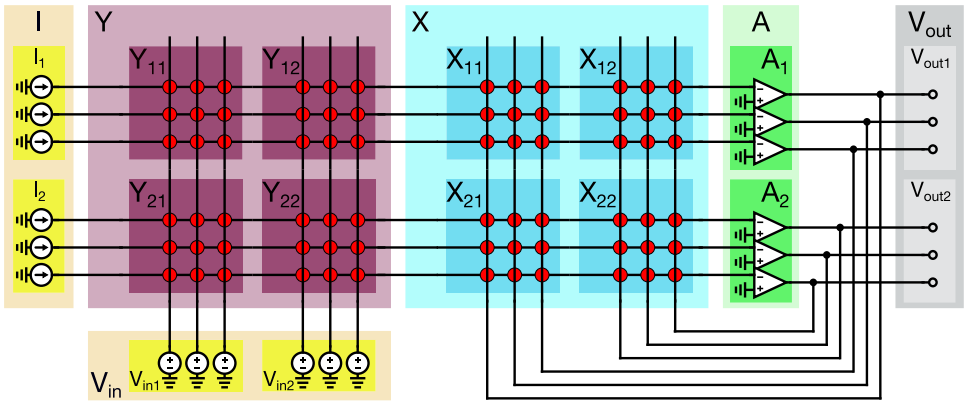

FIGURE 1. **Block-matrix circuit for general-purpose solution of linear algebra problems. Input crosspoint Y is used in the presence of voltage inputs $\mathbf{v}_{in}$, providing an open-loop MVM primitive. Feedback crosspoint X provides IMVM capability thanks to the closed-loop configuration with OA arrays A_i. Current inputs are applied directly to the shared rows of the circuit, providing direct signal injection into the feedback array.**

In Section II, we introduce the block circuit and demonstrate its capability to implement all the previously reported open-loop MVM and closed-loop IMVM circuit topologies. In Section III, we provide a model for the static operation of the circuit, deriving ideal input-output transfer functions and assessing the impact of common error sources. In Section IV, we study the dynamic behavior of the circuit, providing stability criteria and deriving closed-form equations for the time evolution of voltages and currents. The proposed dynamic model has the same accuracy as a SPICE simulation while allowing orders-of-magnitude improvement in wall-clock simulation time.

In the following, we adopt the Householder notation [18], where bold capital letters $\mathbf{A}, \mathbf{B}$ denote matrices, bold lowercase letters $\mathbf{a}, \mathbf{b}$ denote vectors and lowercase letters a, b denote scalars. $\times$ denotes matrix–vector multiplication, $\mathbf{A}^T$ is the transpose of $\mathbf{A}$, $\|\cdot\|_p$ is the vector p-norm and $|||\cdot|||_p$ the induced operator p-norm. A Hermitian positive (negative) semidefinite matrix satisfies $\mathbf{A} \succeq 0$ ($\mathbf{A} \preceq 0$) and its singular values are $\sigma_1(\mathbf{A}) \geq \ldots \geq \sigma_n(\mathbf{A})$. The condition number of $\mathbf{A}$ is $\kappa_\mathbf{A} = \sigma_1(\mathbf{A})/\sigma_n(\mathbf{A})$. $\mathbf{I_n}$ is the identity matrix of size $n \times n$, $\mathbf{0_n}$ is the matrix of all zeros of size $n \times n$. Size subscripts are omitted whenever the size is deducible from context.

II. BLOCK-MATRIX CIRCUIT

Fig. 1 shows the universal circuit for general-purpose solutions of linear algebra problems, such as matrix inversion, matrix-vector multiplication, and regression. The circuit includes two crosspoint arrays, namely the input crosspoint array $\mathbf{Y}$ and the feedback crosspoint array $\mathbf{X}$, sharing connections on rows. Operational amplifiers (OAs) are used to realize a closed-loop feedback configuration around $\mathbf{X}$, by connecting their inputs to the shared rows, and their outputs to columns of matrix $\mathbf{X}$. Input signals are applied to the circuit either through current generators $\mathbf{i}_{in}$ connected to the shared rows, or voltage generators $\mathbf{v}_{in}$ connected to columns of matrix $\mathbf{Y}$. Matrix $\mathbf{Y}$ thus acts as an MVM pre-processing with respect to $\mathbf{v}_{in}$, whereas $\mathbf{X}$ provides IMVM capability as in [12] and [15].

Matrices $\mathbf{X}$ and $\mathbf{Y}$ are further arbitrarily partitioned in blocks in Fig. 1 where, for the sake of simplicity, we have considered a 2×2 partitioning. The OA array is similarly split into blocks, with each block collecting as many OAs as the rows of the corresponding feedback block. In this sense, block $\mathbf{X_{ij}}$ represents the matrix connection between the outputs of OA array A_j and inputs of OA array A_i. A larger number of blocks are possible, provided splitting is performed consistently across all constituent matrices and vectors. Similarly, row connection to the OAs may either be realized to the inverting node, as shown in Fig. 1, or to the non-inverting node, provided all OAs belonging to the same block share the same input connection.

The proposed circuit can perform all previously demonstrated operations using analog in-memory matrix computing by suitably mapping either the input or feedback matrices. The mapping operation can be seen as a circuit rearrangement, thus preserving the same nodal equations. In the following, we report the block matrices corresponding to previously presented circuits of the IMC framework, namely MVM [10], positive-definite linear system solver [12], and linear [13], generalized [15], and ridge regression [7].

A. MATRIX-VECTOR MULTIPLICATION

Typical MVM circuits [10] perform read-out of the currents induced on a target matrix $\mathbf{Y}$ by an array of voltage inputs $\mathbf{v}_{in}$ through an array of transimpedance amplifiers (TIAs) with transconductance k as shown in Fig. 2(a). The equivalent block-matrix circuit is shown in Fig. 2(d), where the input array is used to map $\mathbf{Y}$ and the feedback array is used to map the transimpedance configuration of each amplifier, resulting in a diagonal feedback matrix $k\mathbf{I}$.

An alternative mapping, relying on the feedback array only and current inputs, is shown in Fig. 2(e), corresponding to the circuit in Fig. 2(b). Here, current inputs are converted to

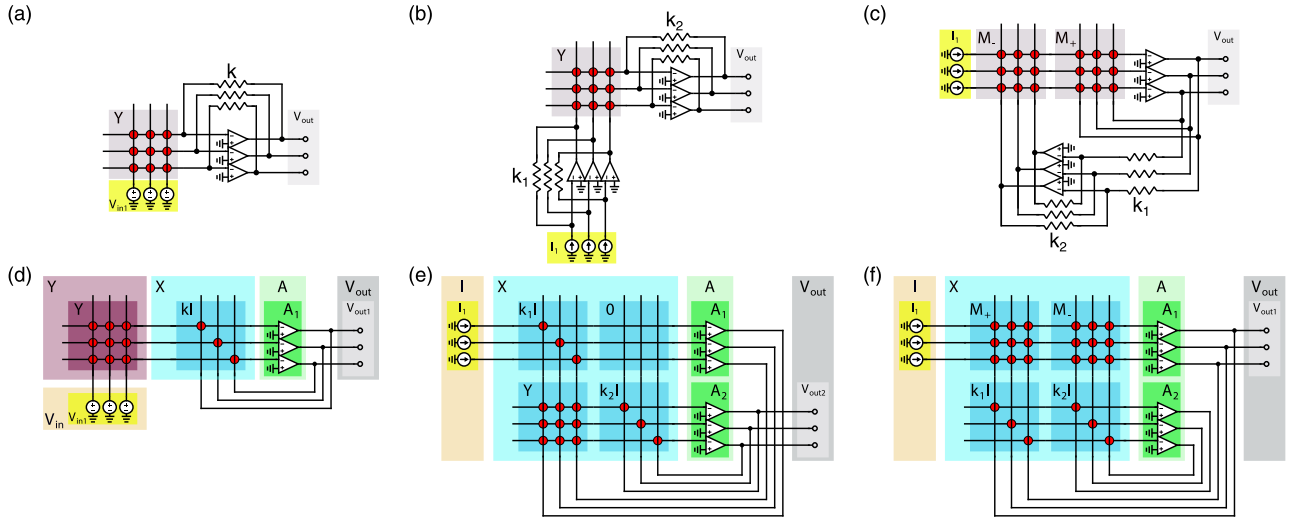

FIGURE 2. MVM and IMVM and their equivalent block-matrix circuits. (a) Standard MVM circuit, where an array of voltage generators is connected to the columns of a conductance matrix Y whose rows are fed to an array of TIAs with gain k, allowing for current collection and conversion to voltage. (b) An alternative scheme for matrix-vector multiplication employing current inputs, converted to a voltage by a first array of TIAs with conductance k_1 and applied to the target matrix Y. MVM currents are collected and converted by a second array of TIAs with tunable gain k_2. (c) Positive-definite linear system solver. Input currents are applied to the shared rows between matrices $\mathbf{M}_+$, $\mathbf{M}_-$. Conductances k_1, k_2 allow tuning the inverting gain applied to matrix $\mathbf{M}_-$, which is generally set to −1. (d) Block-circuit of (a), exploiting both the input and feedback matrix to map Y and the k-conductance TIAs, respectively. (e) Block-circuit of (b), exploiting the feedback array only to perform MVM, at the cost of increased matrix size. (f) Block-circuit of (c).

a voltage by a first TIA array with gain k_1. The outputs of the first TIA array are applied to matrix $\mathbf{Y}$, inducing currents that are converted to a voltage by a second TIA array with gain k_2. The equivalent block-matrix mapping of the circuit consists of a 2×2 partitioning, where local feedback blocks $k_1\mathbf{I}$, $k_2\mathbf{I}$ describe the TIA connection of the first and second OA array, respectively. As $\mathbf{Y}$ is connected between the outputs of A_1 and the inputs of A_2, its corresponding block in the feedback matrix is $(2, 1)$. On the other hand, since there is no direct connection between outputs of A_2 and inputs of A_1, block $(1, 2)$ is set to $\mathbf{0}$. Finally, outputs are probed on amplifier set A_2 as in the original circuit. Additional inputs $\mathbf{i_2}$ and outputs $\mathbf{v}_{\text{out},\mathbf{1}}$ of the block-matrix circuit are unused in this configuration.

B. LINEAR SYSTEM SOLVER

Fig. 2(c) shows the linear system solver [12], which is composed of two amplifier sets in the inverting configuration, A_1 and A_2 both of n OAs, and two $n \times n$ feedback matrices $\mathbf{M}_+ = (1/2)(|\mathbf{M}| + \mathbf{M})$, $\mathbf{M}_- = (1/2)(|\mathbf{M}| - \mathbf{M})$. The circuit can operate on positive-definite matrices only [17]. Analog inverting couplers, realized by means of additional OAs with trimmable transconductances k_1, k_2, provide intermediate voltage inversion and scaling for matrix $\mathbf{M}_-$ [19]. The main circuit equation is

$$\mathbf{v}_{\text{out}} = -\left(\mathbf{M}_+ - \mathbf{M}_-\right)^{-1}\mathbf{i}_\mathbf{1}. \tag{1}$$

The corresponding 2×2 block-matrix circuit is shown in Fig. 2(f). Block $(1, 1)$ maps connection from A_1 output to its own input, corresponding to matrix $\mathbf{M}_+$. Block $(1, 2)$ maps connection from A_2 output to A_1 input, corresponding to matrix $\mathbf{M}_-$. Block $(2, 1)$ maps connection from A_1 output to A_2 input, corresponding to $k_1\mathbf{I}$. Finally, block $(2, 2)$ maps connection from A_2 output to its own input, described by $k_2\mathbf{I}$. All blocks have equal size $n \times n$, such that the overall feedback matrix has size $2n \times 2n$. Since no voltage generators are used, the input array is set entirely to $\mathbf{0}$ and thus neglected. Similarly, additional current inputs $\mathbf{i_2}$ and outputs $\mathbf{v}_{\text{out},\mathbf{2}}$ are unused in this configuration.

C. LINEAR REGRESSION CIRCUIT

Fig. 3(a) shows the linear regression circuit [13], which is composed of two amplifier sets, A_1 of m OAs in inverting configuration and A_2 of n OAs in non-inverting configuration. Two feedback matrices, both mapping $m \times n$ matrix $\mathbf{M}$, are connected between the outputs of A_1 and inputs of A_2 and vice versa. Local feedback on A_1 is provided by conductances k_f, whereas no local feedback connection is set on A_2. The main circuit equations are [15]

$$\mathbf{v_1} = -\frac{1}{k_f}(\mathbf{I} - \mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T)\mathbf{i_1} \tag{2}$$

$$\mathbf{v_2} = -(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{i_1}. \tag{3}$$

Fig. 3(d) shows the corresponding 2×2 block-matrix circuit. Blocks $(1, 1)$ of size $m \times m$ and $(2, 2)$ of size $n \times n$ map local feedback connections on A_1 and A_2, amounting to $k_f\mathbf{I}$ and $\mathbf{0}$ respectively. Block $(1, 2)$ of size $m \times n$ maps connection from A_2 output to A_1 input, corresponding to $\mathbf{M}$. Due to A_1 outputs being connected on rows of matrix

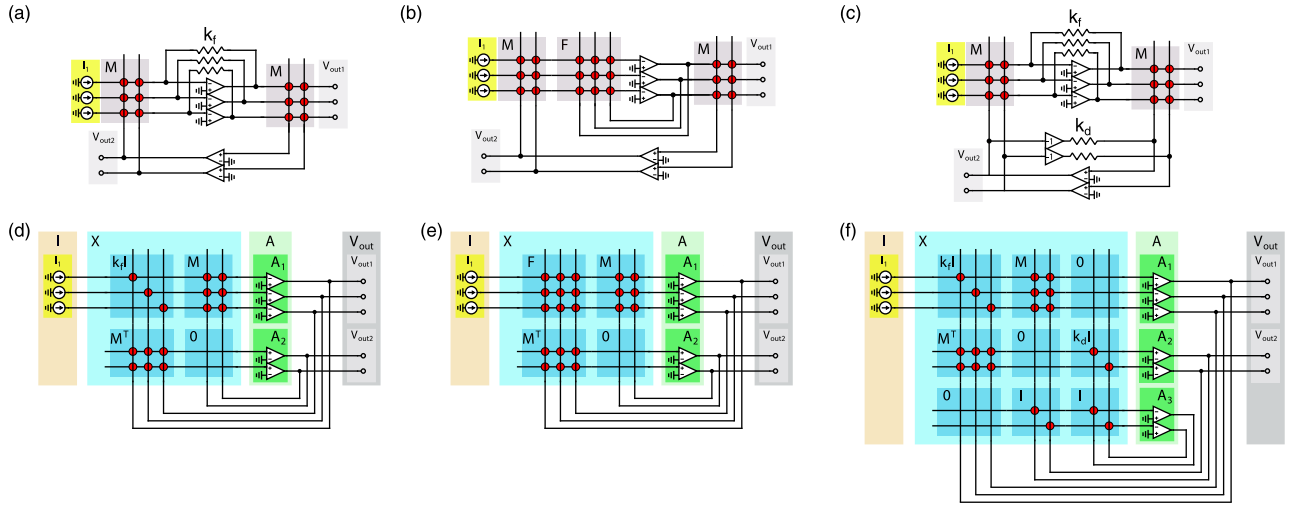

FIGURE 3. Regression circuits and their equivalent block-matrix circuits. (a) Linear regression or pseudoinverse circuit, consisting of two crosspoint arrays both mapping matrix M, an array of TIAs with feedback conductance k_f and an array of non-inverting OAs closing the global feedback. (b) Generalized regression circuit, sharing the same structure as (a). Matrix F allows encoding additional information on data covariance in linear regression operation or acts as a preconditioner when the circuit is used to solve the linear system. (c) Ridge regression circuit, consisting of a linear regression circuit with an additional negative feedback branch on the noninverting OAs. (d) Block-circuit equivalent of the linear regression circuit, using a 2 × 2 non-square partitioning. (e) Block-circuit equivalent of the generalized regression circuit. (f) Block-circuit equivalent of the ridge regression circuit, using a 3 × 3 non-square partitioning.

$\mathbf{M}$, and A_2 inputs being connected on columns, the equivalent matrix to be mapped in block (2, 1) is $\mathbf{M}^T$ with a block size $n \times m$. The overall feedback matrix has thus size $(m + n) \times (m + n)$. Finally, A_2 is set in a noninverting configuration, whereas additional inputs $\mathbf{i_2}$ are unused in this configuration.

D. GENERALIZED REGRESSION CIRCUIT

Fig. 3(b) shows the generalized regression circuit [15]. The main difference with respect to the linear regression circuit is represented by matrix $\mathbf{F}$, placed in feedback configuration on amplifiers A_1. Note that, when matrix $\mathbf{F}$ is diagonal, a weighted linear regression is obtained, which becomes equal to the linear regression in Fig. 3(a) for equal feedback conductance values. The input-output circuit equations for the circuit of Fig. 3(b) are [15]

$$\mathbf{v_1} = -\mathbf{F}^{-1}(\mathbf{I} - \mathbf{M}(\mathbf{M}^T\mathbf{F}^{-1}\mathbf{M})^{-1}\mathbf{M}^T\mathbf{F}^{-1})\mathbf{i_1} \quad (4)$$

$$\mathbf{v_2} = -(\mathbf{M}^T\mathbf{F}^{-1}\mathbf{M})^{-1}\mathbf{M}^T\mathbf{F}^{-1}\mathbf{i_1}. \quad (5)$$

The corresponding block-matrix circuit is shown in Fig. 3(e), which is similar to the linear regression block-matrix circuit except for block (1, 1), which now maps matrix $\mathbf{F}$, retaining the same $m \times m$ block size. Consequently, the overall size of the feedback matrix is still $(m + n) \times (m + n)$.

E. RIDGE REGRESSION CIRCUIT

Fig. 3(c) shows the ridge regression circuit [7]. Starting from a linear regression circuit, a local feedback branch is added on A_2 by means of an inverting analog buffer and an array of conductances k_d. The main circuit equations are thus

$$\mathbf{v_1} = -\frac{1}{k_f}(\mathbf{I} - \mathbf{M}(\mathbf{M}^T\mathbf{M} + k_f k_d \mathbf{I})^{-1}\mathbf{M}^T)\mathbf{i_1} \quad (6)$$

$$\mathbf{v_2} = -(\mathbf{M}^T\mathbf{M} + k_f k_d \mathbf{I})^{-1}\mathbf{M}^T\mathbf{i_1}. \quad (7)$$

The corresponding block-matrix circuit is shown in Fig. 3(f). The block matrix is organized in a 3 × 3 configuration, where blocks (1, 1), (1, 2), (2, 1), and (2, 2) are the same as the linear regression circuit since the corresponding connections are unchanged. Similar to the linear system solver case, analog inverting buffers are realized by inverting stages with −1 gain requiring additional amplifiers A_3. Block (3, 1) thus maps direct connections from A_1 output to A_3 input, corresponding to a zero matrix $\mathbf{0}$. Block (3, 2) corresponds to the connection from A_2 output to A_3 input. For the sake of simplicity, we consider unitary connections, thus resulting in a $n \times n$ block $\mathbf{I}$. To preserve the unitary gain, the same matrix describes the local feedback connection around A_3. The n outputs of A_3 are connected to the n A_2 inputs through conductances k_d. Consequently, the $n \times n$ block (2, 3) is $k_d\mathbf{I}$. Finally, as no direct connection is present between the n buffer outputs and the m A_1 inputs, the $m \times n$ block (1, 3) is $\mathbf{0}$. The overall feedback matrix size is thus $(m + 2n) \times (m + 2n)$. Additional inputs $\mathbf{i_2}$, $\mathbf{i_3}$ are unused in this configuration.

III. STATIC OPERATION MODEL

A. CIRCUIT MATRICES DEFINITION

To study the operation of the circuit, we define block-matrices $\mathbf{Y}$ and $\mathbf{X}$ for the input and feedback crosspoint array conductances, and block-vectors $\mathbf{i}$, $\mathbf{v}_{\text{in}}$ and $\mathbf{v}_{\text{out}}$ corresponding to

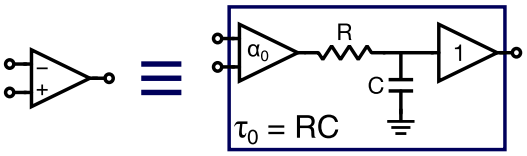

FIGURE 4. Prototype single-pole OA model. An α_0 gain stage models DC gain, whereas an RC network emulates the dominant pole of the OA with time constant $\tau_0 = RC$. A unitary output buffer provides decoupling from load resistance. The effective DC gain sign depends on whether the input is applied to the inverting input ($-\alpha_0$) or to the non-inverting input ($+\alpha_0$), grounding the other terminal.

input currents, input voltages, and output voltages, respectively. For instance, in the case of the 2×2 partitioning of Fig. 1, the block matrices and block vectors read

$$\mathbf{X} = \begin{bmatrix} \mathbf{X_{11}} & \mathbf{X_{12}} \\ \mathbf{X_{21}} & \mathbf{X_{22}} \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y_{11}} & \mathbf{Y_{12}} \\ \mathbf{Y_{21}} & \mathbf{Y_{22}} \end{bmatrix} \tag{8}$$

$$\mathbf{i} = \begin{bmatrix} \mathbf{i_1} \\ \mathbf{i_2} \end{bmatrix} \quad \mathbf{v_{in}} = \begin{bmatrix} \mathbf{v_{in,1}} \\ \mathbf{v_{in,2}} \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \end{bmatrix}. \tag{9}$$

OAs are modeled assuming the single-pole amplifier structure of Fig. 4, consisting of a first amplifying block with gain $\hat{s}\alpha_0$, an RC network whose time constant is $\tau_0 = RC$, and a unity gain buffer. The transfer between the input and output voltages $\hat{v}_i$, $v_{\text{out},i}$ of the i-th amplifier is thus described by

$$v_{\text{out},i}(s) = \alpha_i(s)\,\hat{v}_i(s) \tag{10}$$

with

$$\alpha_i(s) = \hat{s}_i \frac{\alpha_{0,i}}{1 + s\tau_{0,i}} \tag{11}$$

where $s = j\omega$ is the Laplace frequency, and $\hat{s}_i$, $\alpha_{0,i}$, and $\tau_{0,i}$ are the sign, open-loop DC gain, and intrinsic time-constant of the i-th OA, respectively. Correspondingly, the i-th OA dynamics in the time domain are described by the differential equation

$$\tau_{0,i}\frac{dv_{\text{out},i}}{dt} + v_{\text{out},i} = \hat{s}_i\alpha_{0,i}\hat{v}_i. \tag{12}$$

The entire set of amplifiers may thus be described in the frequency domain by a diagonal matrix $\mathbf{A}$

$$\mathbf{A} = (\mathbf{I} + s\mathbf{T_0})^{-1}\,\mathbf{SA_0}, \tag{13}$$

where $\mathbf{S}$ is a diagonal matrix mapping the sign of the corresponding row amplifier, i.e., $S_{ii} = -1$ if the i-th amplifier is in the inverting configuration, or $S_{ii} = +1$ if the i-th amplifier is in noninverting configuration, $\mathbf{A_0}$ is the diagonal matrix of the absolute dc gain of the OAs, i.e., $A_{0,ii} = \alpha_{0,i}$, and similarly $\mathbf{T_0}$ is the diagonal matrix of OAs time constants, i.e., $T_{0,ii} = \tau_{0,i}$, such that $A_{ii} = \alpha_i(s)$. The transfer between the OA input voltage vector $\hat{\mathbf{v}}$ and OA output voltage vector $\mathbf{v}_{\text{out}}$ is thus given by

$$\mathbf{v}_{\text{out}} = \mathbf{A}\hat{\mathbf{v}}. \tag{14}$$

Similarly, the entire set of amplifiers is described in the time domain by the corresponding system of differential equations

$$\mathbf{T_0}\frac{d\mathbf{v}_{\text{out}}}{dt} + \mathbf{v}_{\text{out}} = \mathbf{SA_0}\tilde{\mathbf{v}} \tag{15}$$

where $\hat{\mathbf{v}}$ and $\mathbf{v}_{\text{out}}$ are the OA input and output voltage vectors, respectively.

B. IDEAL STEADY-STATE TRANSFER FUNCTION

To compute the transfer functions in ideal conditions, i.e., assuming infinite DC gain α_0 of the OA, we consider Kirchhoff's law at the input nodes of the OAs in Fig. 1, from which

$$\mathbf{X}\mathbf{v}_{\text{out}} + \mathbf{Y}\mathbf{v}_{\text{in}} + \mathbf{i} = 0. \tag{16}$$

By applying the superposition principle, we first consider the case $\mathbf{v}_{\text{in}} = \mathbf{0}$. In this case, the ideal output voltage is given by

$$\begin{bmatrix} \mathbf{v_{out,i,1}} \\ \mathbf{v_{out,i,2}} \end{bmatrix} = -\mathbf{X}^{-1}\mathbf{i} = -\begin{bmatrix} (\mathbf{X}^{-1})_{11} & (\mathbf{X}^{-1})_{12} \\ (\mathbf{X}^{-1})_{21} & (\mathbf{X}^{-1})_{22} \end{bmatrix}\begin{bmatrix} \mathbf{i_1} \\ \mathbf{i_2} \end{bmatrix} \tag{17}$$

where blocks of the inverse matrix $\mathbf{X}^{-1}$ may be written in terms of blocks of matrix $\mathbf{X}$ by using the block-inversion lemma [18]. We then consider the case with $\mathbf{i} = \mathbf{0}$, for which we write

$$\begin{bmatrix} \mathbf{v_{out,v,1}} \\ \mathbf{v_{out,v,2}} \end{bmatrix} = \begin{bmatrix} (\mathbf{X}^{-1})_{11} & (\mathbf{X}^{-1})_{12} \\ (\mathbf{X}^{-1})_{21} & (\mathbf{X}^{-1})_{22} \end{bmatrix} \times \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} \end{bmatrix}\begin{bmatrix} \mathbf{v_{in,1}} \\ \mathbf{v_{in,2}} \end{bmatrix}. \tag{18}$$

The overall output voltage is thus given by the summation of both contributions, namely

$$\mathbf{v}_{\text{out}} = \mathbf{v}_{\text{out,i}} + \mathbf{v}_{\text{out,v}}. \tag{19}$$

C. OUTPUT STATIC ERROR

The block-matrix model can be used to evaluate the output error arising from various sources. Perturbations of the feedback matrix $\delta\mathbf{X}$, input matrix $\delta\mathbf{Y}$, input currents $\delta\mathbf{i}$, and input voltages $\delta\mathbf{v}_{\text{in}}$ inevitably introduce deviations $\delta\mathbf{v}_{\text{out}}$ of the output voltages from ideal state equations. To quantify the impact of these perturbations, we define the relative error ε

$$\varepsilon = \frac{\|\delta\mathbf{v}_{\text{out}}\|_2}{\|\mathbf{v}_{\text{out}}\|_2}. \tag{20}$$

Perturbations may arise from many different sources, such as quantization, finite amplifier gain, and device variability. For simplicity, we study each perturbation individually. For the i-th perturbation, we derive a maximum relative error $\bar{\varepsilon}_i$ and define upper ($\bar{\varepsilon}_i^{\uparrow}$) and lower bounds ($\bar{\varepsilon}_i^{\downarrow}$). The overall maximum relative error is then bounded by

$$\sqrt{\textstyle\sum_i(\bar{\varepsilon}_i^{\downarrow})^2} \lesssim \bar{\varepsilon}_i \lesssim \sqrt{\textstyle\sum_i(\bar{\varepsilon}_i^{\uparrow})^2}. \tag{21}$$

Table 1 summarizes the main dependences of the upper and lower bounds for the considered sources of perturbation. Additional details on bounds computation are provided in

TABLE 1. Error bounds for various sources of perturbation.

Perturbation	Affects	Exact bound	Lower bound	Upper bound
ε_{α_0}	$\delta\mathbf{X}$	$\frac{\\|\mathbf{X}^{-1}\delta\mathbf{X}\\|_2}{1-\\|\mathbf{X}^{-1}\delta\mathbf{X}\\|_2}$	$1/(\alpha_0\sqrt{n})$	$\sim(\kappa_\mathbf{X}/\alpha_0)\sqrt{n}$
$\varepsilon_{q,\mathbf{X}}$			$\sim 2^{-N_{bit,\mathbf{X}}}$	$\sim \kappa_\mathbf{X} 2^{-N_{bit,\mathbf{X}}}\sqrt{n}$
$\varepsilon_{p,\mathbf{X}}$			$\sim \sigma_{G/G_0}$	$\sim \kappa_\mathbf{X}\sigma_{G/G_0}\sqrt{n}$
$\varepsilon_{q,\mathbf{i}}$	$\delta\mathbf{i}$	$\kappa_\mathbf{X}\frac{\\|\delta\mathbf{i}\\|_2}{\\|\mathbf{i}\\|_2}$	$\kappa_\mathbf{X} 2^{-N_{bit,\mathbf{i}}}/\sqrt{n}$	$\kappa_\mathbf{X} 2^{-N_{bit,\mathbf{i}}}\sqrt{n}$
$\varepsilon_{q,\mathbf{Y}}$	$\delta\mathbf{Y}$	$\kappa_\mathbf{X}\frac{\\|\delta\mathbf{Y}\mathbf{v_{in}}\\|_2}{\\|\mathbf{Y}\mathbf{v_{in}}\\|_2}$	$\sim \kappa_\mathbf{X} 2^{-N_{bit,\mathbf{Y}}}$	$\sim \kappa_\mathbf{X} 2^{-N_{bit,\mathbf{X}}}\sqrt{n}$
$\varepsilon_{p,\mathbf{Y}}$			$\sim \kappa_\mathbf{X}\sigma_{G/G_0}$	$\sim \kappa_\mathbf{X}\sigma_{G/G_0}\sqrt{n}$
$\varepsilon_{q,\mathbf{v_{in}}}$	$\delta\mathbf{v_{in}}$	$\kappa_\mathbf{X}\frac{\\|\mathbf{Y}\delta\mathbf{v_{in}}\\|_2}{\\|\mathbf{Y}\mathbf{v_{in}}\\|_2}$	$\kappa_\mathbf{X}\kappa_\mathbf{Y} 2^{-N_{bit,\mathbf{v_{in}}}}/\sqrt{n}$	$\kappa_\mathbf{X}\kappa_\mathbf{Y} 2^{-N_{bit,\mathbf{v_{in}}}}\sqrt{n}$

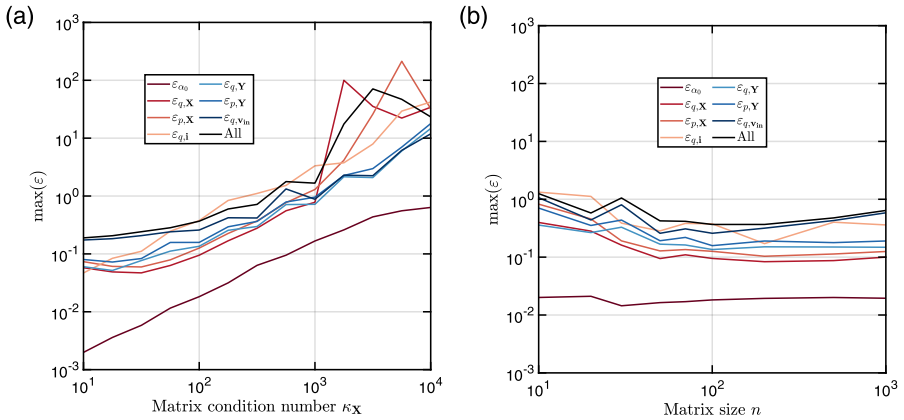

FIGURE 5. (a) Maximum relative error as a function of the condition number $\kappa_\mathbf{X}$ for a 100 × 100 feedback matrix X and different nonidealities, showing a linear dependence on $\kappa_\mathbf{X}$. The error was computed by testing 1000 different input vectors with SPICE simulations. (b) Maximum relative error as a function of the matrix size *n* for a feedback matrix X with condition number $\kappa_\mathbf{X} = 100$. The error was computed by testing 10 × *n* different input vectors with SPICE simulations and is mostly independent of the matrix size.

Appendix A. Upper bounds for all sources depend on the feedback matrix condition number $\kappa_\mathbf{X}$, which therefore serves as the primary sensitivity index of the system. Fig. 5(a) shows simulation results for matrices with increasing condition number $\kappa_\mathbf{X}$ and fixed size $n = 100$. Each line traces the maximum relative error obtained while simulating the corresponding perturbation. Aside from a multiplicative factor dictated by the perturbation nature, all errors linearly increase with $\kappa_\mathbf{X}$. On the other hand, the size dependence of upper bounds might be an excessive overestimation. Fig. 5(b) reports maximum relative errors for each perturbation as a function of the matrix size, for a fixed condition number $\kappa_\mathbf{X} = 10$. As the matrix size increases, maximum errors tend to lose any dependence on the matrix size n, suggesting that lower bounds of Table 1 may prove more helpful in analyzing the system from a scaling standpoint.

IV. DYNAMIC OPERATION MODEL

A. FREQUENCY-DOMAIN MODEL

We begin the analysis of the circuit dynamics by studying its operation in the frequency domain to evaluate the circuit poles. To this aim, we perform the loop gain analysis in Fig. 6 by removing all voltage/current generators and cutting the loop at the output of the OAs in Fig. 1. The loop-gain transfer is the one between the test voltage vector $\tilde{\mathbf{v}}$ and the output voltage vector $\mathbf{v}$.

We first consider the transfer between $\tilde{\mathbf{v}}$ and the intermediate voltage $\hat{\mathbf{v}}$, which corresponds to the input voltage of the OAs. Considering for instance $\hat{v}_1$, the application of Kirchhoff's law yields

$$\sum_{j=1}^{n} Y_{1j}\hat{v}_1 = \sum_{j=1}^{n} X_{1j}\tilde{v}_j - \sum_{j=1}^{n} X_{1j}\hat{v}_1. \tag{22}$$

Equation (22) can be rewritten in matrix/vector form as

$$\mathbf{U_Y}\hat{\mathbf{v}} = \mathbf{X}\tilde{\mathbf{v}} - \mathbf{U_X}\hat{\mathbf{v}} \tag{23}$$

where $\mathbf{U_Y}$ and $\mathbf{U_X}$ are diagonal matrices whose i-th diagonal element contains the sum of the i-th row of $\mathbf{Y}$ and $\mathbf{X}$, respectively. Equation (23) can be rewritten to obtain the closed-form relation between $\hat{\mathbf{v}}$ and $\tilde{\mathbf{v}}$

$$\hat{\mathbf{v}} = (\mathbf{U_Y} + \mathbf{U_X})^{-1}\mathbf{X}\tilde{\mathbf{v}} = \mathbf{U}^{-1}\mathbf{X}\tilde{\mathbf{v}} = \hat{\mathbf{X}}\tilde{\mathbf{v}} \tag{24}$$

where $\hat{\mathbf{X}}$ is the *voltage divider matrix* between $\tilde{\mathbf{v}}$ and $\hat{\mathbf{v}}$.

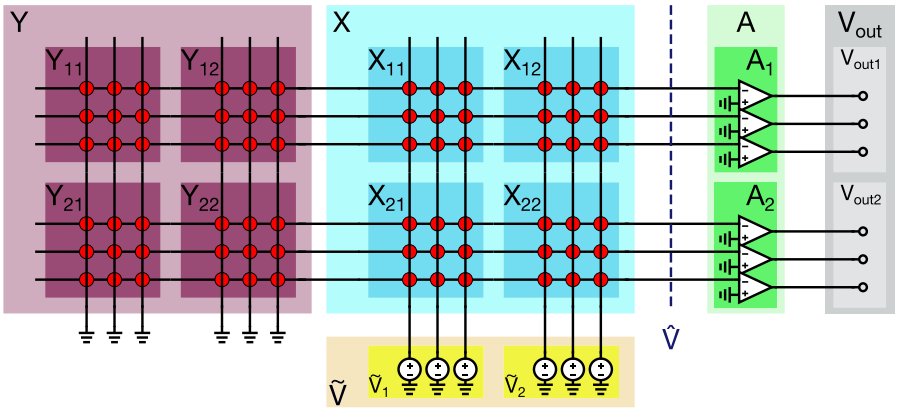

FIGURE 6. **Loop gain test schematic for the block-circuit in Fig. 1. Current and voltage sources are removed and replaced with open- and short-circuits, respectively. The feedback loop is cut at the output of the OAs, and a test voltage vector $\tilde{\mathbf{v}}$ is correspondingly applied on the columns of the feedback array X. The OA output voltage $\mathbf{v_{out}}$ is then probed to assess the loop gain.**

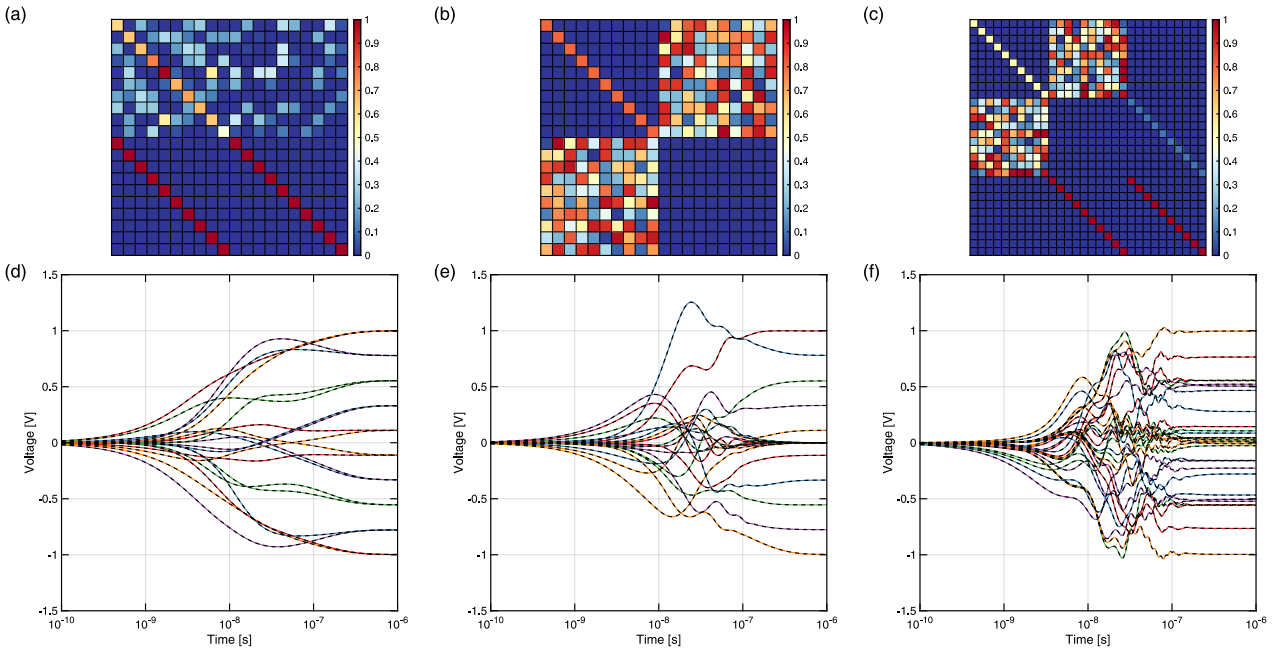

FIGURE 7. **Feedback matrices for (a) linear system solver, (b) linear regression, and (c) ridge regression, and (d)–(f) corresponding output transients, computed by SPICE simulation (colored lines) and by Eq. (31) in MATLAB (dashed lines). The proposed model accurately follows the SPICE-based solution, with reduced simulation overhead.**

Consequently, the overall transfer between the test vector $\tilde{\mathbf{v}}$ and the OA outputs $\mathbf{v}_{\text{out}}$ is given by

$$\mathbf{v}_{\text{out}} = \mathbf{A}\hat{\mathbf{X}}\tilde{\mathbf{v}} = \mathbf{G}_{\text{loop}}(s)\,\tilde{\mathbf{v}} \tag{25}$$

where $\mathbf{G}_{\text{loop}}(s)$ is the *frequency-dependent loop-gain matrix*. Poles of the closed-loop system are then found at the frequencies p for which the test vector $\tilde{\mathbf{v}}$ is identically mapped onto itself, namely

$$\mathbf{G}_{\text{loop}}(p)\,\tilde{\mathbf{v}} = \mathbf{I}\tilde{\mathbf{v}}. \tag{26}$$

By expanding $\mathbf{A}$, the previous equation may be rewritten as

$$\mathbf{S}\mathbf{A}_0\hat{\mathbf{X}}\tilde{\mathbf{v}} = (\mathbf{I} + p\mathbf{T_0})\,\tilde{\mathbf{v}} \tag{27}$$

so that circuit poles can be found by solving

$$\mathbf{T_0}^{-1}(\mathbf{S}\mathbf{A}_0\hat{\mathbf{X}} - \mathbf{I})\tilde{\mathbf{v}} = p\tilde{\mathbf{v}}. \tag{28}$$

Poles p are thus the eigenvalues of $\mathbf{T_0}^{-1}(\mathbf{S}\mathbf{A}_0\hat{\mathbf{X}} - \mathbf{I})$, where matrix $\mathbf{S}\mathbf{A}_0\hat{\mathbf{X}} = \mathbf{G}_{\text{loop}}(0)$ represents the DC loop gain matrix. For stability, it must hold $\text{Re}(p) < 0$, i.e., matrix $\mathbf{T_0}^{-1}(\mathbf{S}\mathbf{A}_0\hat{\mathbf{X}} - \mathbf{I})$ must be Hurwitz-stable [20], in accordance with stability criteria for linear time-invariant systems.

In the particular case of all OAs having the same time constant τ_0, (inverting) sign, and DC open-loop gain α_0,

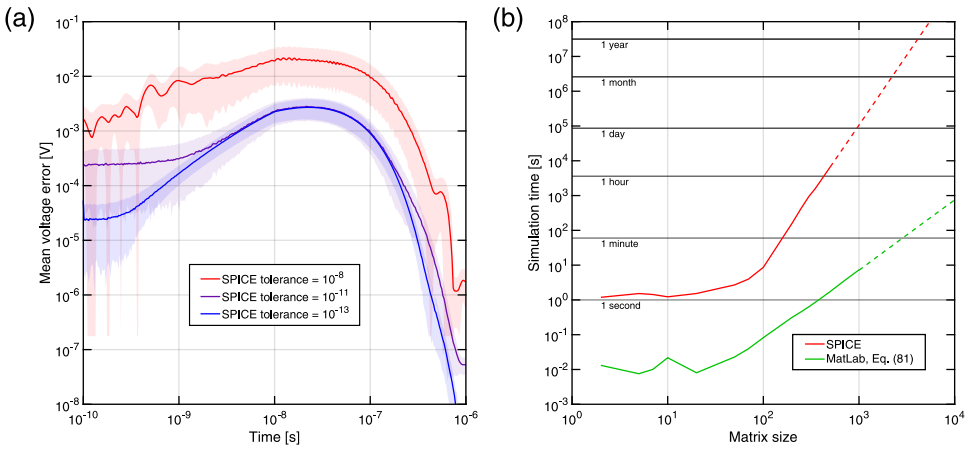

FIGURE 8. **Comparison between SPICE solver and analytical continuous-time model. (a) Mean voltage error between SPICE-based and model-based transient voltages for 1000 linear system solutions, for increasingly conservative relative SPICE tolerances, namely 10^{-8} (blue), 10^{-11} (purple), 10^{-13} (red). Shading denotes 1 standard deviation. As the accuracy increases, the SPICE-based solution approaches the model-based. (b) Simulation time comparison between SPICE and the analytical model implemented in MATLAB as a function of the matrix size n. The analytical model improves both the simulation time and complexity from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^2)$.**

Eq. (28) reduces to

$$p = -\frac{1}{\tau_0}\left(1 + \alpha_0 \lambda_{\hat{\mathbf{X}}}\right) \tag{29}$$

consistently with [17].

Inferring the stability of $\mathbf{S}\hat{\mathbf{X}}$ from the spectral characteristics of the feedback matrix $\mathbf{X}$, which are generally known, is not trivial [21]. As a general criterion, negative-definiteness of matrix $\mathbf{SX}$ is sufficient to determine the stability of $\mathbf{S}\hat{\mathbf{X}}$. Additional details are provided in Appendix B.

B. CONTINUOUS-TIME MODEL

The last step in dynamic modeling is to study the circuit in the continuous-time domain. Once again, we assume that all OAs share the same structure of Fig. 4, although each OA may have a distinct time constant $\tau_{0,i}$, gain $\alpha_{0,i}$, and sign $\hat{s}_i$.

The time-continuous Kirchhoff equation at the OAs inputs for the closed-loop system of Fig. 1 reads

$$\mathbf{i}\,(t) + \mathbf{Y}\mathbf{v}_{\text{in}}\,(t) = \mathbf{U}\hat{\mathbf{v}}\,(t) - \mathbf{X}\mathbf{v}_{\text{out}}\,(t) \tag{30}$$

where $\hat{\mathbf{v}}(t)$ is the time-continuous voltage vector at the OAs inputs. A complete analytical derivation for arbitrary initial conditions and inputs is provided in Appendix C. For current and voltage step inputs, and considering outputs to be initially at rest, then the output voltage reads

$$\mathbf{v}_{\text{out}}\,(t) = -\left(\mathbf{I} - e^{\mathbf{T_0}^{-1}\left(\mathbf{S}\mathbf{A}_0\hat{\mathbf{X}} - \mathbf{I}\right)t}\right) \times \left(\mathbf{X} + \delta\mathbf{X}_\alpha\right)^{-1}\left(\mathbf{i} + \mathbf{Y}\mathbf{v}_{\text{in}}\right). \tag{31}$$

Fig. 7 shows examples of solution transient for different circuits, namely (a) the linear system solution circuit in Fig. 2(f), (b) the linear regression circuit in Fig. 3(f), and (c) the ridge regression circuit in Fig. 3(f), with colored and dashed lines representing the SPICE-computed and model-computed voltages, respectively, highlighting the accuracy of Eq. (31). In particular, Fig. 8(a) shows absolute voltage errors between the model-based and SPICE-based output voltage transients for the circuit shown in Fig. 1, for increasingly conservative SPICE tolerance settings. As SPICE is forced to be more accurate, the solution grows closer to the one computed by the model, once again demonstrating the accuracy of Eq. (31). Finally, Fig. 8(b) reports a comparison of simulation times for increasing matrix size for SPICE (red lines) and model in MATLAB (blue lines), showing up to three orders of magnitude reduction. The MATLAB implementation also scales more favorably with size as $\mathcal{O}(n^2)$ with respect to the $\mathcal{O}(n^4)$ dependence of SPICE-based solvers.

V. CONCLUSION

We present a universal core primitive for analog crossbar-based IMC. The proposed block circuit is capable of implementing any linear matrix operation, including but not limited to MVM, IMVM, and linear and regularized regression. We derive closed-form equations for the ideal input-output transfer functions and static error, together with error bounds that can serve as guidelines for practical implementations. We provide stability criteria and an analytical model for the voltage transient in the presence of step inputs, outperforming SPICE-based solvers by several orders of magnitude. The proposed model can be retroactively applied to previous feedback-based circuit implementations, both open-loop and closed-loop, and can serve as a generalized framework for the study of analog-IMC topologies. Owing to its highly scalable structure, the circuit can represent a complete macro for matrix-based operations in novel analog processing units under the IMC paradigm.

REFERENCES

[1] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electron.*, vol. 1, no. 6, pp. 333–343, 2018, doi: 10.1038/s41928-018-0092-2.

[2] D. Ielmini and G. Pedretti, "Device and circuit architectures for in-memory computing," *Adv. Intell. Syst.*, vol. 2, no. 7, Jul. 2020, Art. no. 2000040, doi: 10.1002/aisy.202000040.

[3] M. Prezioso et al., "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61–64, May 2015, doi: 10.1038/nature14441.

[4] P. Mannocci et al., "In-memory principal component analysis by crosspoint array of resistive switching memory: A new hardware approach for energy-efficient data analysis in edge computing," *IEEE Nanotechnol. Mag.*, vol. 16, no. 2, pp. 4–13, Apr. 2022, doi: 10.1109/MNANO.2022.3141515.

[5] C. Li et al., "Analogue signal and image processing with large memristor crossbars," *Nature Electron.*, vol. 1, no. 1, pp. 52–59, 2018, doi: 10.1038/s41928-017-0002-z.

[6] F. Cai et al., "Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks," *Nature Electron.*, vol. 3, no. 7, pp. 409–418, Jul. 2020, doi: 10.1038/s41928-020-0436-6.

[7] P. Mannocci et al., "An analogue in-memory ridge regression circuit with application to massive MIMO acceleration," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 12, no. 4, pp. 952–962, Nov. 2022, doi: 10.1109/JETCAS.2022.3221284.

[8] M. A. Zidan et al., "The future of electronics based on memristive systems," *Nature Electron.*, vol. 1, no. 1, pp. 22–29, 2018, doi: 10.1038/s41928-017-0006-8.

[9] M. Hu et al., "Memristor-based analog computation and neural network classification with a dot product engine," *Adv. Mater.*, vol. 30, no. 9, Mar. 2018, Art. no. 1705914, doi: 10.1002/adma.201705914.

[10] Z. Sun and R. Huang, "Time complexity of in-memory matrix-vector multiplication," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 8, pp. 2785–2789, Aug. 2021, doi: 10.1109/TCSII.2021.3068764.

[11] M. Le Gallo et al., "Mixed-precision in-memory computing," *Nature Electron.*, vol. 1, no. 4, pp. 246–253, Apr. 2018, doi: 10.1038/s41928-018-0054-8.

[12] Z. Sun et al., "Solving matrix equations in one step with cross-point resistive arrays," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 10, pp. 4123–4128, Mar. 2019, doi: 10.1073/pnas.1815682116.

[13] Z. Sun et al., "One-step regression and classification with cross-point resistive memory arrays," *Sci. Adv.*, vol. 6, no. 5, Jan. 2020, Art. no. eaay2378, doi: 10.1126/sciadv.aay2378.

[14] Z. Sun et al., "In-memory pagerank accelerator with a cross-point array of resistive memories," *IEEE Trans. Electron Devices*, vol. 67, no. 4, pp. 1466–1470, Apr. 2020, doi: 10.1109/TED.2020.2966908.

[15] P. Mannocci et al., "A universal, analog, in-memory computing primitive for linear algebra using memristors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 12, pp. 4889–4899, Nov. 2021, doi: 10.1109/TCSI.2021.3122278.

[16] B. Feinberg et al., "An analog preconditioner for solving linear systems," in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA)*. Seoul, South Korea: IEEE, Feb. 2021, pp. 761–774, doi: 10.1109/HPCA51647.2021.00069.

[17] Z. Sun et al., "Time complexity of in-memory solution of linear systems," *IEEE Trans. Electron Devices*, vol. 67, no. 7, pp. 2945–2951, Jul. 2020, doi: 10.1109/TED.2020.2992435.

[18] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge, NY, USA: Cambridge Univ. Press, 2012.

[19] G. Pedretti et al., "Redundancy and analog slicing for precise in-memory machine learning—Part II: Applications and benchmark," *IEEE Trans. Electron Devices*, vol. 68, no. 9, pp. 4379–4383, Sep. 2021, doi: 10.1109/TED.2021.3095430.

[20] A. Hurwitz, "Ueber die Bedingungen, unter welchen eine Gleichung nur Wurzeln mit negativen reellen Theilen besitzt," *Mathematische Annalen*, vol. 46, no. 2, pp. 273–284, Jun. 1895, doi: 10.1007/BF01446812.

[21] O. Y. Kushel, "Unifying matrix stability concepts with a view to applications," *SIAM Rev.*, vol. 61, no. 3, pp. 643–729, Jan. 2019, doi: 10.1137/18M119241X.

[22] P. C. Hansen, "The 2-norm of random matrices," *J. Comput. Appl. Math.*, vol. 23, no. 1, pp. 117–120, Jul. 1988, doi: 10.1016/0377-0427(88)90336-6.

Open Access funding provided by 'Politecnico di Milano' within the CRUI CARE Agreement