An analogue in-memory ridge regression circuit with application to massive MIMO acceleration

Piergiulio Mannocci, *Student Member, IEEE,* Enrico Melacarne, and Daniele Ielmini, *Fellow, IEEE*

***Abstract*—In-memory computing (IMC) has emerged as one of the most promising candidates for distributed computing frameworks such as edge computing, owing to its unrivalled energy efficiency and high throughput. By leveraging arrays of emerging devices, such as resistive random access memories (RRAM), to implement massive parallel computation, IMC overcomes the main limitations of classic von Neumann architectures. Meanwhile, next generation telecommunication networks are bound to rely ever more intensively on matrix computations to allow simultaneous transmission and reception over multiple spatial channels, an approach known as Massive Multiple-Input Multiple-Output (MIMO). Here, we propose a closed-loop in-memory computing circuit for the acceleration of Ridge Regression, an algebraic prior that finds application in all phases of a typical massive MIMO transaction, namely channel estimation, uplink and downlink. Particularly, we show the circuit's capability to perform Zero-Forcing (ZF) and Regularized Zero-Forcing (RZF) detection and beamforming, benchmarking its performance in a realistic framework and comparing results with a commercial graphic processing unit (GPU). Our results indicate a 4 orders-of-magnitude increase in energy efficiency and a 3 orders-of-magnitude increase in area efficiency for the same throughput of a digital solution, supporting IMC for energy efficient pre- and post-processing in next-generation B5G and 6G networks.**

***Index Terms*—In-memory computing, resistive random access memory, hardware accelerator, ridge regression, massive MIMO**

I. Introduction

The increasing requirements for communication in modern society have driven the advancements in broadband cellular networks during the last decades. Since the introduction of digital telecommunications in 1991, the capacity and throughput of communication channels have been steadily increased. In fifth-generation (5G) and beyond-fifth-generation (B5G) mobile communication systems, massive MIMO (multiple-input and multiple-output) has gained momentum as the most promising technique to improve the spectral efficiency by means of large arrays of transmitter and receiver antennas. In fact, massive MIMO exhibits tremendous improvements in terms of data rate and energy efficiency with respect to other technologies [1]. On the other hand, the data-intensive computation in massive MIMO, mostly consisting of matrix-vector multiplication (MVM) and matrix inversion, represents a significant overhead in limiting the available throughput and energy efficiency.

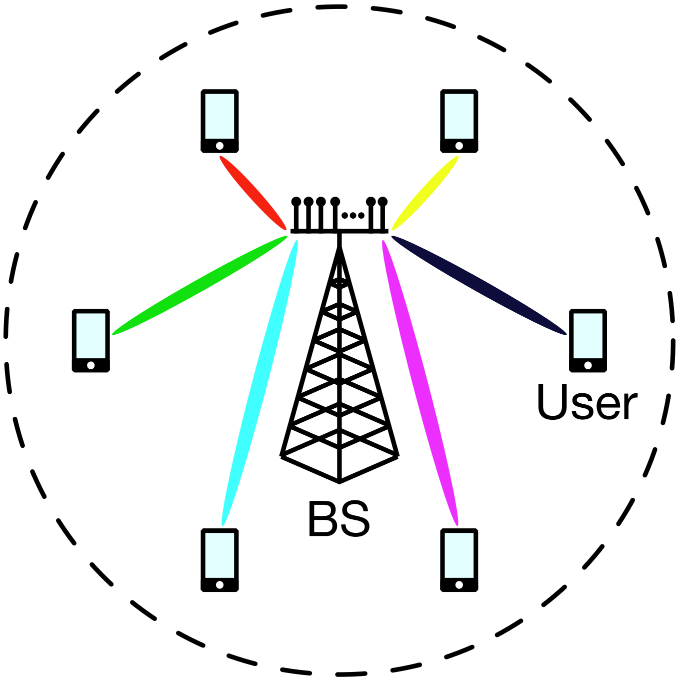

Fig. 1. Typical massive MIMO cell architecture, in which a base station (BS) equipped with M antennas must serve K single-antenna users.

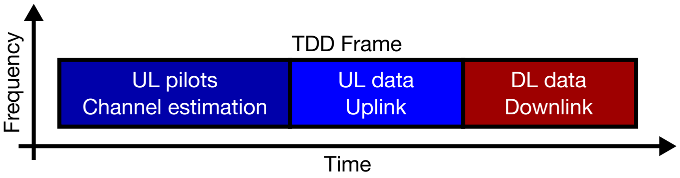

Fig. 2. Time Division Duplex (TDD) protocol. Each transaction between the users and the BS is composed of three phases. In the first phase, the users send pilot signal to the BS to allow for channel estimation. After, the same users transmit their uplink data. After estimating the channel, the BS decodes the uplink data and precodes the downlink data that is transmitted to the users in the last transaction phase.

Recently, in-memory computing (IMC) has shown strong performance in terms of increased throughput and reduced energy consumption with respect to conventional von-Neumann computers. In IMC, computation is performed *in situ* within the memory, thus eliminating the need for data movement which is generally responsible for the largest part of computing time and energy [2], [3]. Usually implemented with crosspoint arrays of resistive switching memories, IMC shows a high computation parallelism due to the inherent matrix-like structure of the memory array and the possibility to perform computation in the analogue domain via fundamental physical laws,

P. Mannocci, E. Melacarne and D. Ielmini are with Politecnico di Milano, Milano, Italy.

Correspondence should be addressed to piergiulio.mannocci@polimi.it and daniele.ielmini@polimi.it.

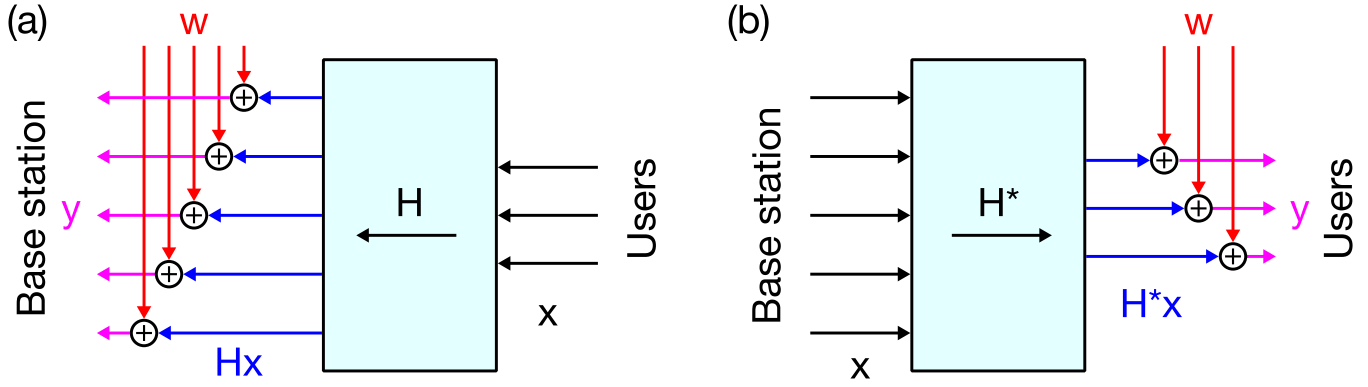

Fig. 3. (a) Uplink transmission model. The transmitted signal $\mathbf{x}$ of N_t users is propagated along the uplink channel $\mathbf{H}$, and mixes with additive Gaussian noise $\mathbf{w}$ at the base station side, where it is detected by $N_r \geq N_t$ antennas. (b) Downlink transmission model. The transmitted signal $\mathbf{x}$ of the N_r antennas is propagated along the downlink channel $\mathbf{H}^*$, and mixes with additive Gaussian noise $\mathbf{w}$ at the user side, which comprises $N_t \leq N_r$ terminals.

such as the Kirchhoff's law for summation and the Ohm's law for multiplication. Recently, IMC has been demonstrated for accelerating MVM in various scenario, such as image processing [4], sparse coding [5], deep neural networks [4] and solution of equations [6]. MVM execution within a closed-loop circuit was shown to accelerate computing tasks of higher complexity, such as matrix inversion [7], eigenvector calculation [8], linear regression [9] and generalized linear regression [10] with significant increase of throughput and energy efficiency [10], [11]. Given the relevance of matrix inversion and regression in modern machine learning (ML), closed-loop inverse-matrix-vector multiplication (IMVM) circuits appear promising for accelerating a number of real-life applications, including massive MIMO in wireless communications.

Previously proposed RRAM accelerators for baseband processing in massive MIMO relied on crosspoint arrays for the sole acceleration of MVM operations within both zero-forcing (ZF) and regularized zero-forcing (RZF), while the computationally intensive inverse-matrix calculation was carried out by an external digital processor [12], [13]. In this work, we present a fully-IMC solution of massive MIMO exploiting closed-loop IMVM circuits, thus removing the need for a separate computation unit to perform matrix inversion. After presenting the general form of MIMO technique, we introduce a new IMC circuit for ridge regression which can be used for the regularized zero-forcing encoding to address channel estimation, uplink (receive combining) and downlink (beamforming/precoding) in massive MIMO.

In the following, we adopt the Householder notation [14], 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. $\cdot$ denotes matrix-vector multiplication, and is generally omitted if the resulting expression is not ambiguous. $\mathbf{A}^T$ and $\mathbf{A}^*$ are the real and conjugate transpose of $\mathbf{A}$ respectively, $\|\cdot\|_p$ 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 trace operator is $\mathrm{Tr}\{\mathbf{A}\} = \sum_{i=1}^{n} A_{ii}$.

II. MIMO BACKGROUND

Fig. 1 shows a typical massive MIMO cell architecture where a base station (BS) with N_r antennas communicates with N_t single-antenna users or terminals. The characteristics of the environment between antennas and terminals are generally summarized by a channel propagation matrix $\mathbf{H}$, where each element h_{ij} describes the transfer between the j-th user and the i-th antenna. A typical model for $\mathbf{H}$ is the Gaussian channel, where each matrix element is a complex random variable given by:

$$h_{ij} \sim \mathcal{CN}(0,1). \tag{1}$$

A more advanced model for $\mathbf{H}$, taking into account the possible correlation at either receiver or transmitter side or both, is the Kronecker channel, given by:

$$\mathbf{H} = \mathbf{R_R}^{1/2}\mathbf{K}\mathbf{R_T}^{1/2} \tag{2}$$

where $\mathbf{K}$ is a Gaussian channel matrix with elements $k_{ij} \sim \mathcal{CN}(0,1)$, while $\mathbf{R_R}$ and $\mathbf{R_T}$ are the spatial correlation matrices at the receiver and transmitter side, respectively. The elements of $\mathbf{R_R}$ and $\mathbf{R_T}$ are given by [15]:

$$r_{ij} = \begin{cases} \rho^{j-i} & i \leq j \\ \rho^*_{ji} & i > j \end{cases} \qquad |\rho| \leq 1 \tag{3}$$

where different values of ρ may be used to differentiate correlation between receivers and transmitters. The Gaussian channel model may be regarded as a special case of the Kronecker channel with $\rho = 0$. Communication between the users and the BS typically follows the time division duplex (TDD) protocol shown in Fig. 2 [16]. The users first send some pre-defined messages, termed *pilots*, to the BS, which enable estimating the channel matrix $\mathbf{H}$. Then, the users send their uplink data, which is decoded by the BS using the recently estimated channel matrix. Finally, the BS pre-codes the data to be transmitted to the users in the downlink phase.

A. Uplink transmission

Fig. 3a illustrates the uplink operations, where each of the N_t terminals sends a signal x_j. The latter is generally encoded following a modulation scheme such as Quadrature Amplitude

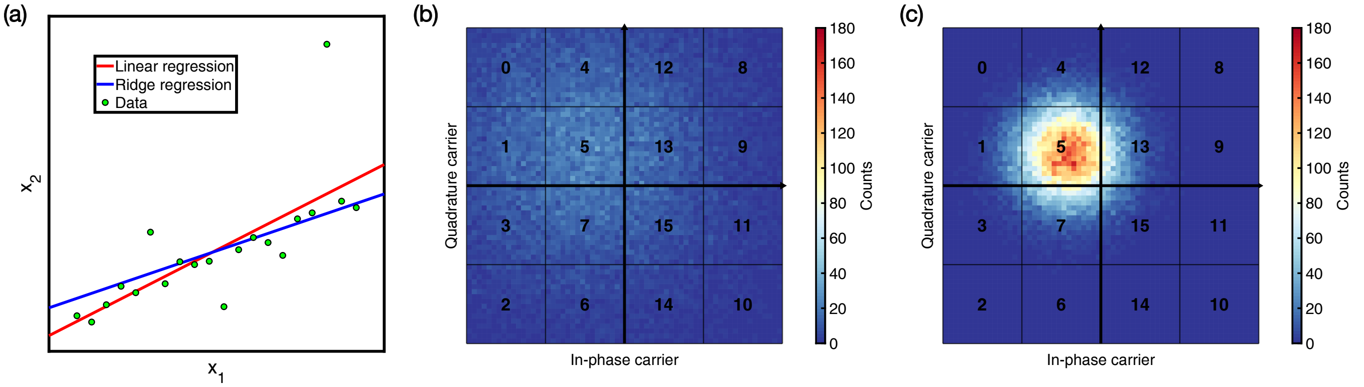

Fig. 4. (a) Regularized regression. Linear regression (red) tends to overfit data in presence of outliers, whereas ridge regression (blue) avoids overfitting by constraining the regression coefficients. (b) Decoded symbol count for 400 uplink experiments on 64 × 64 Kronecker channels with $\rho = 0.6$ and SNR = 20 dB, targeting transmission of symbol 5. Noise amplification leads to high dispersion of decoded values, resulting in almost equal decoding probability for 0, 1, 3, 4, 5 and 7. (c) Decoded symbol count for the same 400 uplink experiments of (b), using RZF. Thanks to regularization, probability of correctly decoding symbol 5 increases.

Modulation (QAM), where a given binary word or *symbol* (*e.g*, 0b101 or 5) is mapped onto in-phase and in-quadrature components of a transmitted sine wave (*e.g.*, $-1 + j1$) [17]. The overall signal from all users is thus given by the complex vector $\mathbf{x}$. The signal is propagated across the environment between the users and the BS, suffering from attenuation and phase distortion. The incoming signal at the i-th antenna is thus given by:

$$y_i = \sum_{j=1}^{N_t} h_{ij} x_j + w_i, \tag{4}$$

where w_i is the receiver noise seen at the i-th antenna, which is generally modeled by a complex-Gaussian distribution with zero mean and σ_n^2 variance, *i.e.* $w_i \sim \mathcal{CN}(0, \sigma_n^2)$. The received vector signal $\mathbf{y}$ then satisfies:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}. \tag{5}$$

To estimate the originally transmitted message, the BS may use the zero-forcing detection technique [1], where the estimated signal reads:

$$\tilde{\mathbf{x}} = \mathbf{H}^{+}\mathbf{y} = \mathbf{x} + \mathbf{H}^{+}\mathbf{w} \tag{6}$$

The retrieved signal $\tilde{\mathbf{x}}$ thus consists of the original signal $\mathbf{x}$ plus a modified noise term $\mathbf{H}^{+}\mathbf{w}$. The main limitation of the ZF approach is that the noise term may be relatively large for high condition numbers of $\mathbf{H}$, which is typically the case for channels with some degree of correlation. In such cases, the noise term may overshadow the original signal, thus leading to an unacceptable symbol error rate (SER), defined as:

$$\text{SER} = \frac{\text{\# of correctly decoded symbols}}{\text{\# of received symbols}} \tag{7}$$

where the symbol is assumed to be correctly decoded when both extracted real and imaginary parts are closer to those of the correct symbol compared to all other symbols.

To prevent such degradation, a regularized zero-forcing technique may be used [1], where the received signal $\mathbf{y}$ is decoded according to:

$$\begin{aligned} \tilde{\mathbf{x}} &= (\mathbf{H}^*\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^*\mathbf{y} \qquad (8) \\ &= (\mathbf{H}^*\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^*\mathbf{H}\mathbf{x} + (\mathbf{H}^*\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^*\mathbf{w} \qquad (9) \end{aligned}$$

Thanks to the regularization term $\lambda\mathbf{I}$, the effective condition number of the matrix to be inverted is lowered, thus preventing a possible amplification of the noise term $\mathbf{w}$. On the other hand, regularization may affect the original message, which may be irreversibly lost unless the λ parameter is properly tuned. It has been shown [18] that the optimal value for the regularization parameter, allowing for an ideal trade-off between noise amplification and signal distortion, is given by $\lambda = \sigma_w^2/\sigma_x^2$. An equivalent formulation for λ can be given in terms of the signal-to-noise ratio, defined as:

$$\text{SNR} = \frac{\mathbb{E}[\|\mathbf{H}\mathbf{x}\|_2^2]}{\mathbb{E}[\|\mathbf{w}\|_2^2]} \tag{10}$$

by noting that when $\mathbf{x}$ is drawn from the unit-average power constellation, *i.e.* $\sigma_x^2 = 1$, and for sufficiently large scale systems:

$$\mathbb{E}[\|\mathbf{H}\mathbf{x}\|_2^2] = N_r N_t \tag{11}$$

$$\mathbb{E}[\|\mathbf{w}\|_2^2] = N_r \sigma_w^2 \tag{12}$$

The optimal regulation parameter can therefore be equivalently rewritten as:

$$\lambda = \frac{N_t}{\text{SNR}} \tag{13}$$

The mathematical analogous for regularized zero-forcing detection is ridge regression [19], a regularization technique allowing to avoid overestimation by introducing an ℓ_2-penalty on the regression coefficients. Fig. 4a shows the results of ridge regression and linear regression for a set of data in two dimensions, supporting the higher accuracy of ridge regression in the presence of outliers. Correspondingly, the inherent improvement in decoding accuracy of RZF with respect to ZF is shown in Figs. 4b-c, where 400 uplink experiments were conducted on Kronecker channels with $\rho = 0.6$ and SNR = 20 dB. In each experiment, a random vector including symbol 5 was transmitted from users to BS and decoded either with ZF (Fig. 4b) or with RZF (Fig. 4c). When ZF was used, decoding of symbol 5 was generally erroneous, leading to interpretation as 0, 1, 3, 4, 5 or 7 with almost equal probability. On the other hand, when RZF was used, the symbol was correctly decoded most of the time thanks to the filtering action of ridge regression.

B. Downlink transmission

Fig. 3b illustrates the downlink operation, where each of the N_r BS antennas sends a signal x_j which is then propagated across the channel in the downlink direction and detected at the i-th user side as:

$$y_i = \sum_{i=1}^{N_r} h_{DL,ij} x_j + w_i \tag{14}$$

where w_i is the receiver noise seen at the i-th terminal. Thanks to the TDD protocol, the uplink and downlink channel are reciprocal, *i.e.* $\mathbf{H_{DL}} = \mathbf{H}^*$. Consequently, the received vector at the user side satisfies the expression:

$$\mathbf{y} = \mathbf{H}^* \mathbf{x} + \mathbf{w}. \tag{15}$$

In a typical massive MIMO case, the user-side is not equipped to perform decoding operations on the incoming signal. Therefore, the received vector $\mathbf{y}$ should ideally match the original message $\mathbf{s}$, which can be achieved by precoding techniques at the BS side to account for the propagation across the channel. To this purpose, the transmitted signal $\mathbf{x}$ is generally computed as a transformation of the original message $\mathbf{s}$ by a precoding matrix $\mathbf{B}$:

$$\mathbf{x} = \mathbf{B}\mathbf{s} \tag{16}$$

According to the ZF technique, $\mathbf{B}$ is simply equal to the pseudo-inverse of $\mathbf{H}^*$ [20]. The signal is also normalized by a scaling factor γ to satisfy a given power constraint $E[\mathbf{x}^*\mathbf{x}] = P_{tr}$, from which γ can be obtained from the relationship:

$$\gamma^2 = \frac{P_{tr}}{\text{Tr}\{(\mathbf{H}^*\mathbf{H})^{-1}\}} \tag{17}$$

The received signal at the user-side is then obtained by:

$$\begin{aligned} \mathbf{y} &= \mathbf{H}^* \gamma \mathbf{B} \mathbf{s} + \mathbf{w} \qquad (18) \\ &= \gamma \mathbf{s} + \mathbf{w} \qquad (19) \end{aligned}$$

Similar to the uplink problem, the scaling factor γ may become relatively small as the condition number of $\mathbf{H}$ increases, due to an increase of the term $\text{Tr}\{(\mathbf{H}^*\mathbf{H})^{-1}\}$. In this case, the noise term $\mathbf{w}$ is relatively large compared to the signal $\mathbf{s}$, thus resulting in an increased SER. This issue can be overcome by RZF [20], where precoding allows to strengthen the signal reaching each user, thus increasing the corresponding scaling factor γ via interference between the BS antennas. According to RZF, the pre-coding matrix reads:

$$\mathbf{B} = \mathbf{H}(\mathbf{H}^*\mathbf{H} + \lambda \mathbf{I})^{-1}, \tag{20}$$

while γ can be obtained from the relationship:

$$\gamma^2 = \frac{P_{tr}}{\text{Tr}\{(\mathbf{H}^*\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^*\mathbf{H}(\mathbf{H}^*\mathbf{H} + \lambda\mathbf{I})^{-1}\}}, \tag{21}$$

from which the signal received by the users is given by:

$$\mathbf{y} = \gamma \mathbf{H}^*\mathbf{H}(\mathbf{H}^*\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{s} + \mathbf{w}. \tag{22}$$

The parameter λ must be properly tuned to avoid excessive interference between the antennas, while at the same time effectively improving the scaling factor γ. It has been shown that, similar to the uplink problem, the optimal value for the interference parameter is given by Eq. (13) [20], [21].

C. Channel estimation

Both uplink and downlink operations require an estimate of $\mathbf{H}$ to describe the propagation channel between the BS and the users. To estimate the channel, the users synchronously send a pre-defined pilot message to the BS. Pilot vectors are generally extracted from a given *pilot book* $\mathbf{P}$ [22] given by:

$$\mathbf{P} = [\mathbf{p_1}|\mathbf{p_2}|\cdots|\mathbf{p_m}], \tag{23}$$

where each pilot vector is transmitted to the BS by the array of user terminals, such that the received matrix $\mathbf{Y}$ reads:

$$\mathbf{Y} = [\mathbf{y_1}|\mathbf{y_2}|\cdots|\mathbf{y_m}] = \mathbf{HP} + \mathbf{W}, \tag{24}$$

where $\mathbf{W} = [\mathbf{w_1}|\mathbf{w_2}|\cdots|\mathbf{w_m}]$ is the noise matrix. It has been shown that, similar to the uplink and downlink problems, the optimal estimation is given by RZF according to [23]:

$$\hat{\mathbf{H}} = \mathbf{Y}\mathbf{P}^*(\mathbf{P}\mathbf{P}^* + \lambda\mathbf{I})^{-1} \tag{25}$$

where λ is given by $\lambda = \sigma_n^2/\sigma_h^2$. In the particular case of orthogonal pilot vectors, the corresponding pilot book satisfies $\mathbf{P}\mathbf{P}^* = k\mathbf{I}$, thus leading to:

$$\hat{\mathbf{H}} = \frac{1}{k+\lambda}\mathbf{Y}\mathbf{P}^*. \tag{26}$$

Instead of estimating $\hat{\mathbf{H}}$, the RZF can be aimed at estimating $\hat{\mathbf{H}}^*$ which reads:

$$\hat{\mathbf{H}}^* = (\mathbf{P}\mathbf{P}^* + \lambda\mathbf{I})^{-1}\mathbf{P}\mathbf{Y}^*. \tag{27}$$

The matrix $\hat{\mathbf{H}}^*$ can thus be computed column-by-column (corresponding to a row-by-row computation of $\hat{\mathbf{H}}$) by applying the estimator on columns of $\mathbf{Y}^*$, *i.e.* rows of $\mathbf{Y}$.

III. IMC CIRCUIT FOR RIDGE REGRESSION

The RZF operation for solving the problems of uplink/downlink transmissions and channel estimation is a data-intensive ML algorithm that is inefficiently carried out by digital computers with von Neumann architecture. Fig. 5 shows an IMC circuit with closed-loop architecture that is derived from the closed-loop linear regression circuit [9] to efficiently accelerate the RZF operation. The circuit, which is referred to as ridge regression circuit (RRC) in the following, is composed of two crosspoint arrays each mapping the same matrix $\mathbf{M}$, two input current vectors $\mathbf{i_1}$, $\mathbf{i_2}$, and two sets of transimpedance amplifiers (TIAs) with feedback conductance t and $-\delta$, where the negative conductance is achieved by an analogue inverting buffer in the second set of TIAs.

We first consider the case in which the first input current is applied while the second one is not, *i.e.* for $\mathbf{i_2} = 0$. Assuming ideal operational amplifiers, the Kirchhoff's laws at the input terminals of the operational amplifiers read:

$$\mathbf{i_1} + \mathbf{M}\mathbf{v_1} + t\mathbf{v_2} = 0 \tag{28}$$

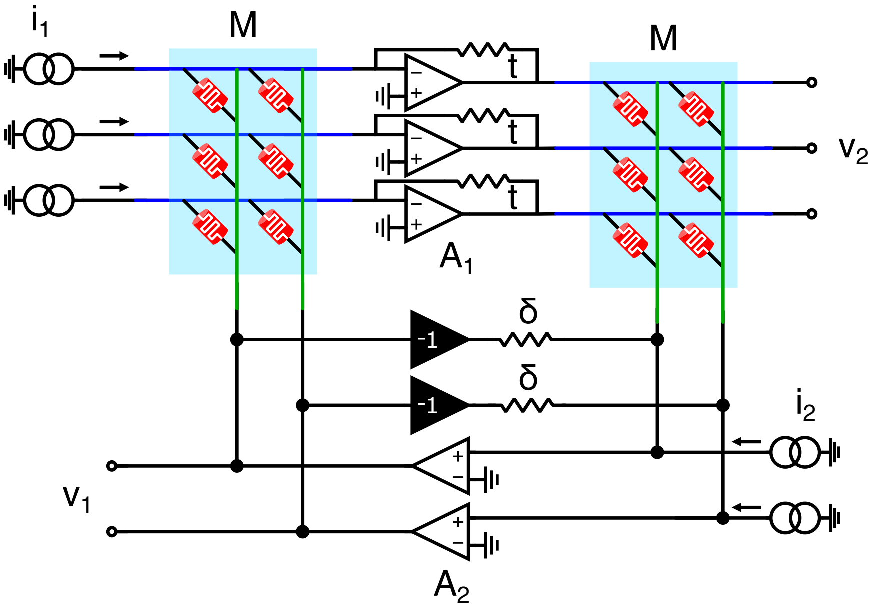

Fig. 5. In-memory ridge regression circuit. Two memory arrays, each mapping the same matrix $\mathbf{M}$, are connected in a feedback loop using two sets of programmable transimpedance amplifiers. For the A_1 set, the transimpedance resistance t is positive, whereas for the A_2 set the transimpedance resistance δ is negative and realized by means of an additional inverting analogue buffer. In uplink operation, input currents are applied at the $\mathbf{i_1}$ input, and outputs are read at the $\mathbf{v_1}$ output. In downlink operation, inputs are applied at the $\mathbf{i_2}$ input, and outputs are read at the $\mathbf{v_2}$ output.

$$-\delta\mathbf{v_1} + \mathbf{M}^T\mathbf{v_2} = 0 \tag{29}$$

from which one can extract the voltages at the first output as:

$$\mathbf{v_1} = -(\mathbf{M}^T\mathbf{M} + t\delta\mathbf{I})^{-1}\mathbf{M}^T\mathbf{i_1} \tag{30}$$

which corresponds to the ridge regression of $\mathbf{i_1}$ over $\mathbf{M}$ with $\lambda = t\delta$. Eq. (30) is the electrical equivalent of the RZF decoder of Eq. (8) for $\mathbf{M} = \mathbf{H}$, $\mathbf{i_1} = \mathbf{y}$ and $\mathbf{v_1} = \mathbf{x}$, for the case of real $\mathbf{H}$, $\mathbf{x}$ and $\mathbf{y}$. Also note that the same circuit can be used for channel matrix estimation according to Eq. (27), assuming $\mathbf{M} = \mathbf{P}$ and providing rows of $\mathbf{Y}$ as input current $\mathbf{i_1}$. Trivial steps can then be performed on the estimated matrix $\hat{\mathbf{H}}^*$ to recover $\hat{\mathbf{H}}$.

To apply the circuit to the general case of complex channels, the complex matrix $\mathbf{H}$ can be mapped in the real-valued matrix $\mathbf{H}_R$ according to:

$$\mathbf{H}_R = \begin{bmatrix} \mathrm{Re}(\mathbf{H}) & -\mathrm{Im}(\mathbf{H}) \\ \mathrm{Im}(\mathbf{H}) & \mathrm{Re}(\mathbf{H}) \end{bmatrix} \tag{31}$$

Similarly, complex $\mathbf{x}$ is mapped into the real-values $\mathbf{x}_R$ by:

$$\mathbf{x}_R = \begin{bmatrix} \mathrm{Re}(\mathbf{x}) \\ \mathrm{Im}(\mathbf{x}) \end{bmatrix} \tag{32}$$

Consequently, one can obtain:

$$\begin{bmatrix} \mathrm{Re}(\mathbf{y}) \\ \mathrm{Im}(\mathbf{y}) \end{bmatrix} = \mathbf{y}_R = \mathbf{H}_R\mathbf{x}_R \tag{33}$$

The complex-transpose operation on $\mathbf{H}$ is now equivalent to the real-transpose operation on $\mathbf{H}_R$, namely:

$$(\mathbf{H}^*)_R = \begin{bmatrix} \mathrm{Re}(\mathbf{H}^*) & -\mathrm{Im}(\mathbf{H}^*) \\ \mathrm{Im}(\mathbf{H}^*) & \mathrm{Re}(\mathbf{H}^*) \end{bmatrix} \tag{34}$$

$$= \begin{bmatrix} \mathrm{Re}(\mathbf{H})^T & \mathrm{Im}(\mathbf{H})^T \\ -\mathrm{Im}(\mathbf{H})^T & \mathrm{Re}(\mathbf{H})^T \end{bmatrix} = \mathbf{H}_R^T \tag{35}$$

When the second input is applied, *i.e.* for $\mathbf{i_1} = 0$, the state equations for ideal operational amplifiers read:

$$\mathbf{M}\mathbf{v_1} + t\mathbf{v_2} = 0 \tag{36}$$

$$\mathbf{i_2} - \delta\mathbf{v_1} + \mathbf{M}^T\mathbf{v_2} = 0. \tag{37}$$

The output voltage $\mathbf{v_2}$ then reads:

$$\mathbf{v_2} = -(\mathbf{M}\mathbf{M}^T + t\delta\mathbf{I})^{-1}\mathbf{M}\mathbf{i_2} \tag{38}$$

Remembering that, for $t\delta \neq 0$, the following equality holds:

$$(\mathbf{M}\mathbf{M}^T + t\delta\mathbf{I})^{-1}\mathbf{M} = \mathbf{M}(\mathbf{M}^T\mathbf{M} + t\delta\mathbf{I})^{-1} \tag{39}$$

it is possible to reformulate Eq. (38) as:

$$\mathbf{v_2} = -\mathbf{M}(\mathbf{M}^T\mathbf{M} + t\delta\mathbf{I})^{-1}\mathbf{i_2} \tag{40}$$

which is the electrical equivalent of the RZF precoder of Eq. (20) for real $\mathbf{H}$.

By embedding the required algebraic computation in the transfer function between input and output pairs, significant energy and time saving is expected from both the programming and computation standpoint with respect to the use of external, inverse-matrix-dedicated computing units.

IV. Simulation Results

The circuit and the corresponding steady-state equations were validated for both uplink and downlink operating modes by extensive SPICE simulations. Operational amplifiers (OAs) were assumed to have an open-loop gain of $80\,\mathrm{dB}$ and a gain-bandwidth product (GBWP) of $100\,\mathrm{MHz}$, whereas memory elements were initially considered to have 64-bit floating point precision.

Fig. 6a shows an example of an uplink transient for a 64×32 Gaussian channel. The circuit's convergence time

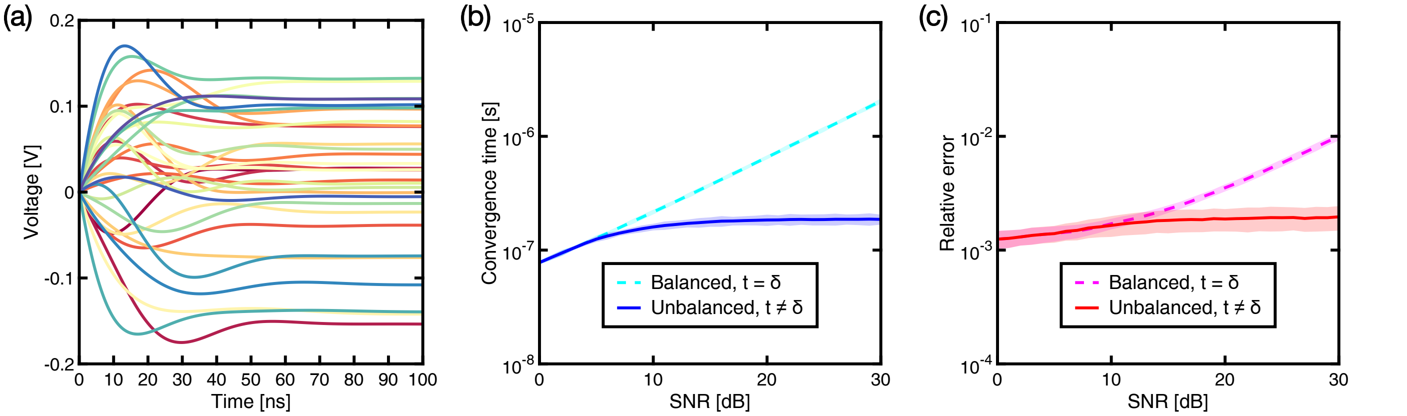

Fig. 6. (a) Transient example of IMC-RRC in uplink decoding for a 64×32 channel matrix. The circuit converges to the solution in less than 100 ns. (b) Computing time and (c) relative error as a function of the signal-to-noise ratio, for a balanced configuration (dashed line, $t = \delta = \sqrt{\lambda}$), and unbalanced configuration (continuous line, $t = 1$, $\delta = \lambda$) on a 64×32 channel. Unbalancing the resistor values of t and δ allows to contain the degradation of both computing time and error, resulting in an almost $\mathcal{O}(1)$ dependence with respect to SNR. For each line, the shaded area denotes the standard deviation of 1000 simulations.

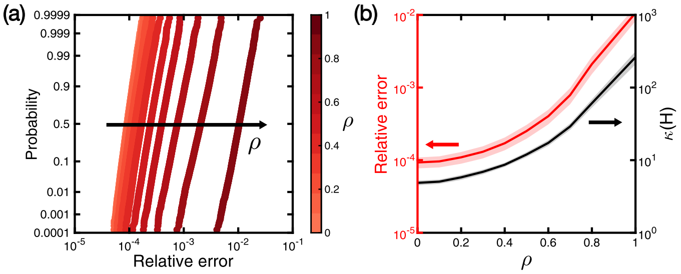

Fig. 7. (a) CDF of static error of IMC-RRC, for 100 uplink experiments on 100 64×32 channels with variable channel correlation parameter ρ. Even at high $\rho \simeq 1$, the median error is still below 1%. (b) Relative error and channel matrix condition number $\kappa_{\mathbf{H}}$ as a function of the channel's correlation parameter ρ. As the latter increases, both metrics suffer a similar degradation.

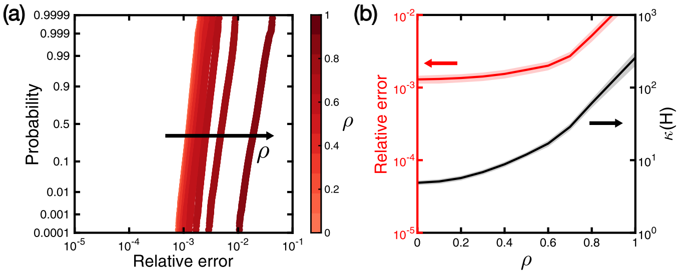

Fig. 8. (a) CDF of static error of IMC-RRC, for 100 downlink experiments on 100 64×32 channels with variable channel correlation parameter ρ. With respect to uplink experiments, the median relative error increases. (b) Relative error and channel matrix condition number $\kappa_{\mathbf{H}}$ as a function of the channel's correlation parameter ρ. For $\rho < 0.5$, the relative error degradation is mostly independent on the condition number.

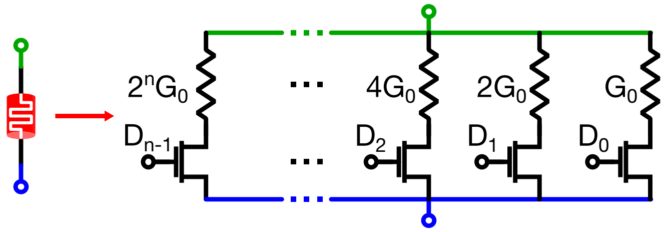

Fig. 9. Fully CMOS implementation of n-bits programmable conductive element, using a nTnR structure. By toggling the selection bits D_i, the overall conductance between the two terminals can be linearly modulated between G_0 and $(2^n - 1)G_0$.

is in the order of hundreds of nanoseconds, which may be further enhanced by increasing the GBWP of OAs [10], [11]. When the regularization term is mapped in a balanced configuration, *i.e.* $t = \delta = \sqrt{\lambda}$, both the convergence time (Fig. 6b, dashed line) and relative error (Fig. 6c, dashed line) show a dependence on the signal-to-noise ratio which tends to worsen the circuit response. In order to retrieve the desired behavior, an unbalanced configuration may be used, where the transimpedance resistor t is kept constant (*e.g.*, $t = 1$) and the regulation resistor δ is swept to match the problem requirements (*e.g.*, $\delta = \lambda$). The unbalanced configuration allows to avoid degradation of both computing time and relative error (Figs. 6b-c, continuous line). As the SNR increases, δ's impact on both metrics tends to cancel out as $\delta = \lambda \rightarrow 0$, leaving only the transimpedance resistor t to determine the circuit's performance [10].

Fig. 7a shows the cumulative distribution functions (CDFs) for 100 uplink transmissions on 100 different 64×32 Kronecker channels with increasing correlation parameter ρ. The relative error increases at increasing ρ due to the corresponding increase of the condition number $\kappa_{\mathbf{H}}$ of the channel matrix, as shown in Fig. 7b [10].

Similarly, Fig. 8a shows the CDFs for 100 downlink transmission on 100 different 64×32 Kronecker channels with increasing correlation parameter ρ. While the error is generally higher with respect to the uplink problem, it can still be explained by an increase of the condition number $\kappa_{\mathbf{H}}$ for increasing ρ, as shown in Fig. 8b.

Given the intrinsically discrete nature of the problem nonetheless, a more reliable metric for circuit performance is SER. We first evaluated the SER for the uplink problem for an amplifier gain $\alpha_0 = 80\,\text{dB}$ and variable equivalent precision of the memory cell. The latter may be a memristive cell such as a resistive switching random access memory (RRAM) [24] or a phase change memory (PCM) [25]. For an improved precision

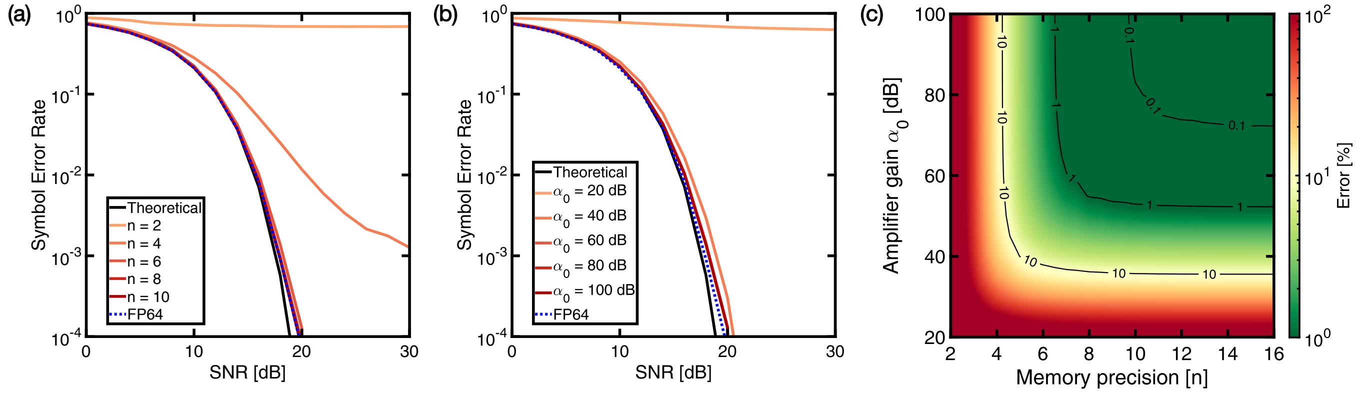

Fig. 10. Symbol error rate (SER) as a function of signal to noise ratio (SNR) in uplink decoding for (a) $\alpha_0 = 80\,\mathrm{dB}$ and variable number of bits of the memory device, (b) 6-bits memory device and variable amplifier gain α_0. In (c), colormap of the error with respect to a floating-point implementation for different combinations of α_0 and n. All lines represent the average of 10000 experiments on 64×32 Gaussian channels ($\rho = 0$), using 16-QAM modulation. Owing to the discrete nature of 16-QAM encoding, the low-precision analogue solution is comparable with the floating-point one even with relatively low α_0 and n.

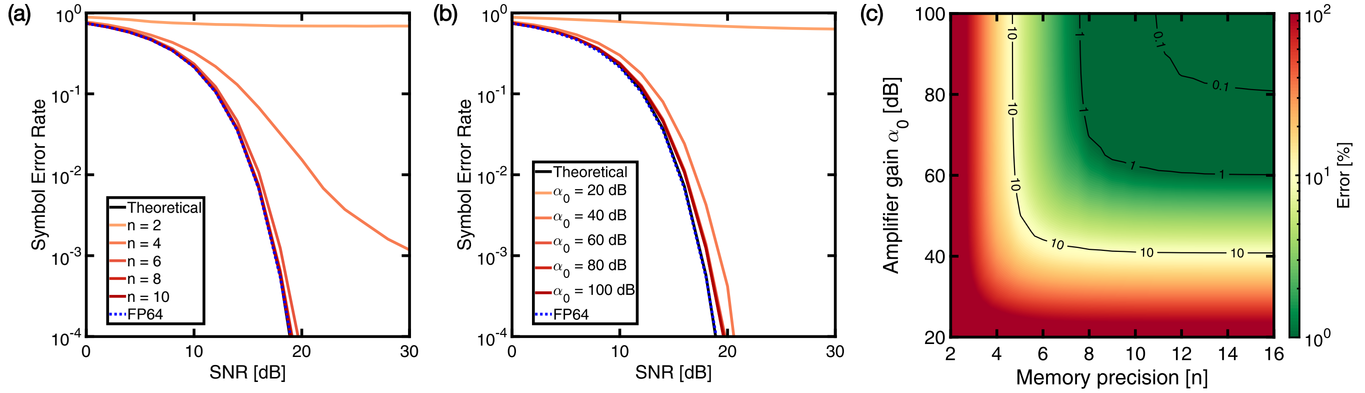

Fig. 11. Symbol error rate (SER) as a function of signal to noise ratio (SNR) in downlink precoding for (a) $\alpha_0 = 80\,\mathrm{dB}$ and variable number of bits of the memory device, (b) 6-bits memory device and variable amplifier gain α_0. In (c), colormap of the error with respect to a floating-point implementation for different combinations of α_0 and n. All lines represent the average of 10000 experiments on 64×32 Gaussian channels ($\rho = 0$), using 16-QAM modulation. Owing to the discrete nature of 16-QAM encoding, the low-precision analogue solution is comparable with the floating-point one even with relatively low α_0 and n.

of the conductive element, the n-transistor/n-resistor (nTnR) structure in Fig. 9 can be adopted. In the nTnR structure of Fig. 9, where $n = 5$ was assumed, the conductance of the i-th resistor is $G_i = G_0 2^i$ with i ranging from 0 to $n-1$ and $G_0 = 1/R_0$ being a reference unit conductance. The resistor can be a memristive device or an integrated resistance, *e.g.* a polysilicon resistor, with fixed conductance G_i.

Fig. 10a shows the average SER as a function of SNR for a 16-QAM modulation. Even with relatively low 5-bit precision memories, the RRC is capable of achieving the same SER of a 64-bit floating point implementation, thus closely matching the theoretical SER provided by [26], [27]:

$$\mathrm{SER}_{th} = 1 - \left(1 - \frac{2(\sqrt{M}-1)}{\sqrt{M}} Q\left(\sqrt{\frac{3\mathrm{SNR}}{M-1}}\right)\right)^2 \tag{41}$$

The SER was then evaluated for a given memory precision of $n = 6$ bits and varying the open-loop gain α_0 of the operational amplifier. The limited gain α_0 results in an error signal between the input terminals of each amplifier in Fig. 5, thus affecting the state equations Eqs. (30) and (40). Fig. 10b shows the average SER as a function of SNR, highlighting that even for relatively small gains above $40\,\mathrm{dB}$, the error rate is comparable to a floating-point implementation and the theoretical SER. The trade-off between gain and precision is summarized in Fig. 10c, showing the relative error with respect to the floating-point implementation, computed as:

$$\mathrm{Error} = \frac{\|\mathrm{SER}_{\mathrm{FP64}} - \mathrm{SER}_{\mathrm{RRC}}\|_2}{\|\mathrm{SER}_{\mathrm{FP64}}\|_2} \quad [\%] \tag{42}$$

Similar results were obtained for the downlink transmission. In particular, Fig. 11a shows the calculated SER as a function of SNR for variable bit precision of the memory element for a constant gain $\alpha_0 = 80\,\mathrm{dB}$, while Fig. 11b shows the SER as a function of SNR for variable gain α_0 and fixed bit precision $n = 6$. The comparison with a digital computation in floating point double precision indicates excellent accuracy of the RRC for $n = 6$ and $\alpha_0 = 60\,\mathrm{dB}$. With respect to the uplink problem, gain and precision requirements are generally higher for the downlink case, due to both the increased circuit error and the power normalization procedure. Nonetheless, Fig. 11c highlights that an open-loop gain of $60\,\mathrm{dB}$ and a 6-bits memory precision allow to match the floating-point SER within 5% in both downlink and uplink.

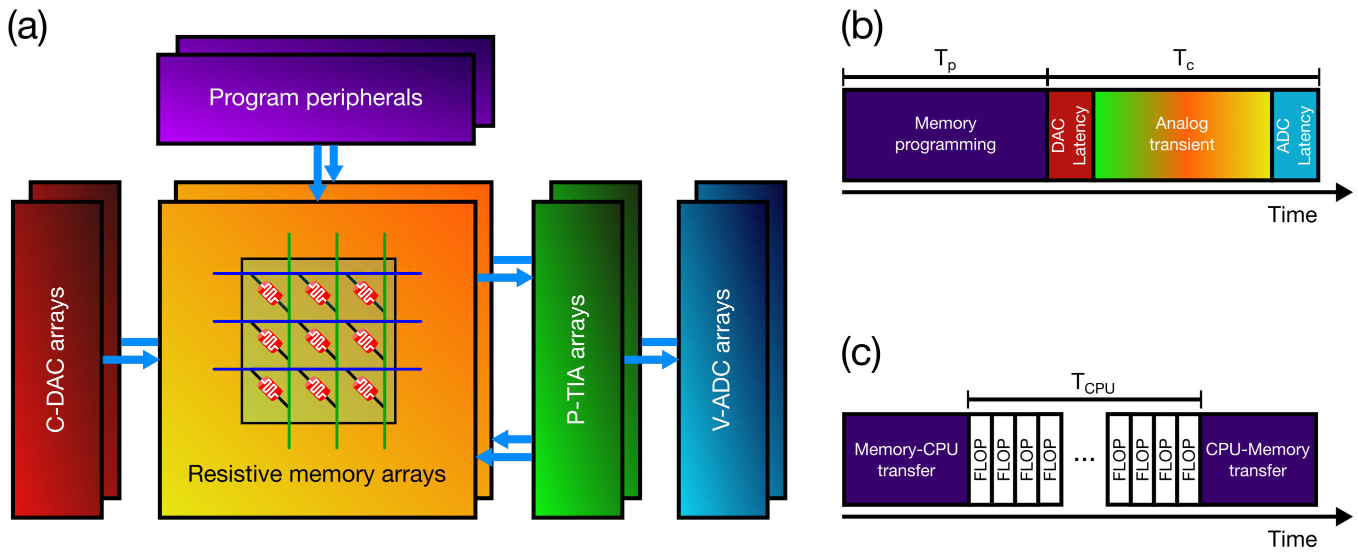

Fig. 12. (a) Benchmark framework. Crosspoint arrays with their own program peripherals are feedback-connected to programmable TIAs. Inputs are provided by means of current DACs, and outputs are read by means of voltage ADCs. (b) Timing phases of IMC-RRC. After programming devices in a T_p time, inputs are applied and outputs are sampled after circuit convergence, for an overall T_c time. (c) Timing phases of a digital computing system. Data is transferred from memory to CPU, where computation takes place in a sequence of FLOPs in T_{CPU} time. When computation is over, data is transferred back to memory.

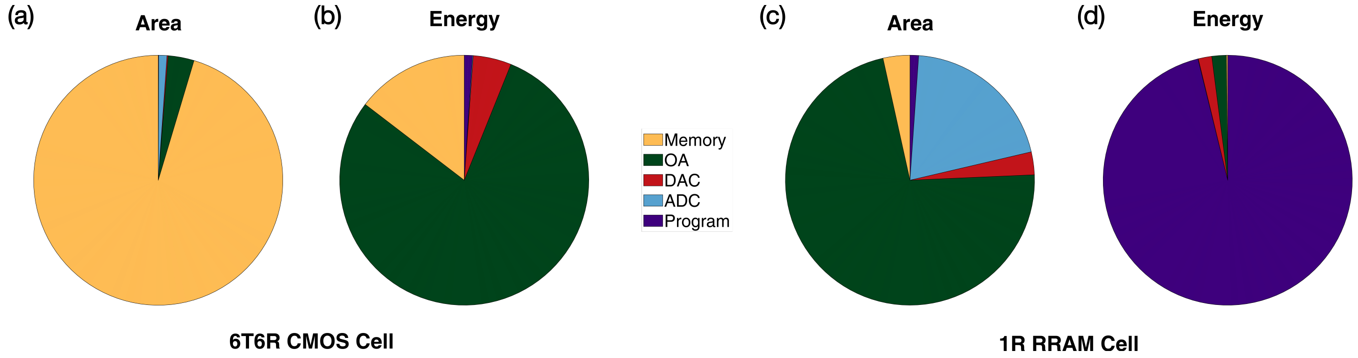

Fig. 13. Energy consumption and area breakdown of IMC-RRC, averaged on 4000 simulations of uplink and downlink in high and low SNR conditions for a 256 × 128 MIMO system. (a) Area and (b) energy breakdowns of IMC-RRC using the 6T6R-CMOS weighted-resistor of Fig. 9 as memory element. (c) Area and (d) energy breakdowns of IMC-RRC using an ideal 1R-RRAM device as memory element. 1R-RRAM outperforms 6T6R-CMOS in terms of area occupation, but consumes more energy in the programming phase.

V. Benchmark and Scaling Studies

To assess the RRC performance in realistic conditions, we considered the circuit architecture of Fig. 12a, where two resistive memory arrays are connected to two vectors of programmable transimpedance amplifiers (P-TIAs), with two vectors of current-based digital-to-analog converters (C-DACs) and two vectors of voltage-based analog-to-digital converters (V-ADCs). The program peripherals have the function of programming the memory array with the estimated channel matrix $\mathbf{H}$ for uplink/downlink transmission or the pilot matrix $\mathbf{P}$ for channel estimation.

Fig. 12b schematically shows the sequence of operations performed by the RRC during a computation. First, the memory arrays are programmed using the program peripherals in a time T_p. Analog input currents are then applied via the C-DACs, thus initiating the transient evolution of the output voltages. Once the output voltages have reached a value sufficiently close to the steady state, they are sampled and converted to the digital domain by the corresponding V-ADCs. We call T_c the total time required by the circuit to compute the results, including the C-DACs and V-ADCs latency and the analogue transient time, *i.e* $T_c = T_{DAC} + T_{tran} + T_{ADC}$ For comparison, Fig. 12c illustrates the computation in an equivalent digital system. Here, data are first transferred from the memory to the CPU, then the RZF algorithm is digitally carried out in a sequence of floating-point operations (FLOPs) within a total time T_{CPU}. Finally, the computed output is transferred back to memory. The overall time overhead for data movement is assumed equal to the computing time [29], $T_{DM} = T_{CPU}$.

Fig. 13 shows the results of 4000 benchmark simulations of the RRC, where we changed the SNR for both uplink and downlink transmissions. In all cases, we considered single-pole operational amplifiers with GBWP = $500\,\mathrm{MHz}$ and $\alpha_0 = 80\,\mathrm{dB}$. The supply voltage and currents were assumed to be $1.2\,\mathrm{V}$ and $10\,\mu\mathrm{A}$, respectively, corresponding to a standby power dissipation of $12\,\mu\mathrm{W}$ and requiring a $50\,(\mu\mathrm{m})^2$ area in a CMOS $14\,\mathrm{nm}$ technology [30]. For the C-DACs, we consid-

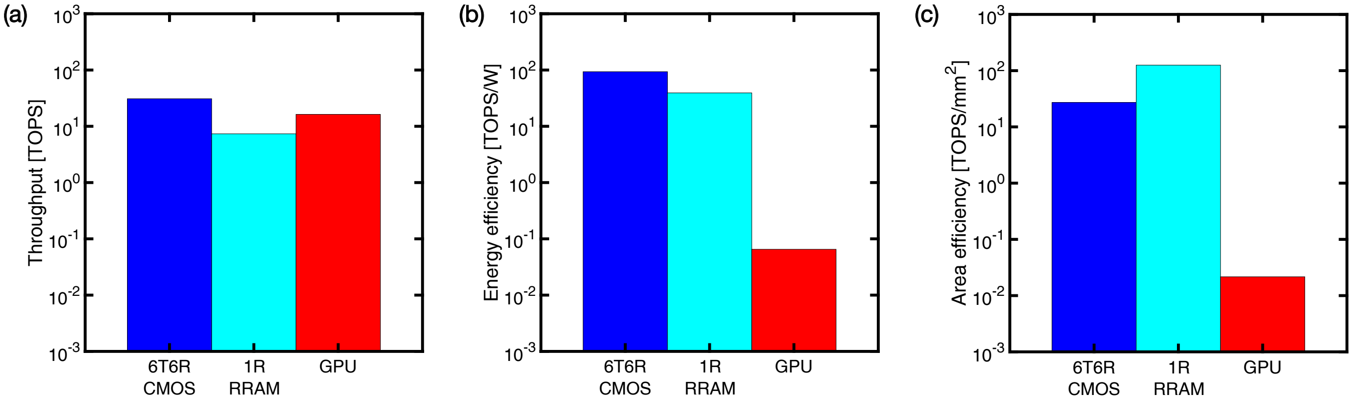

Fig. 14. Comparison of (a) throughput, (b) energy efficiency and (c) area efficiency for a 256×128 MIMO system. Three different implementations are compared, namely (i) IMC circuit with a 1R RRAM crosspoint array, (ii) IMC circuit with a 6T6R CMOS memory array, and (iii) fully-digital GPU implementation [28]. While both the 6T6R CMOS cell of Fig. 9 and a 1R RRAM cell implementation achieve the same throughput of the GPU implementation (a), the former shows a better energy efficiency figure (b), thanks to the reduced programming overhead. The 1R RRAM implementation shows a better area efficiency (c) thanks to the improved area density of the memory array.

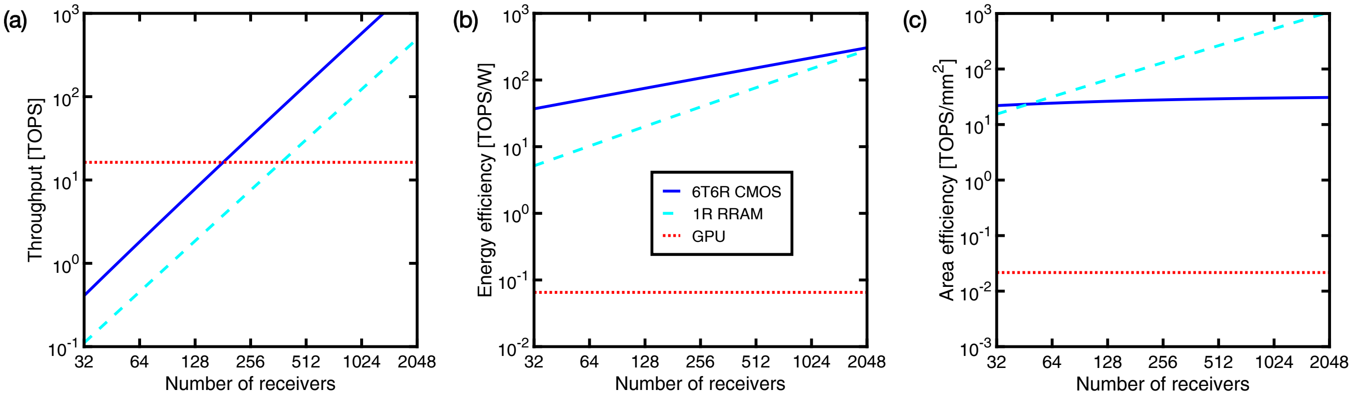

Fig. 15. Scaling projections for (a) throughput, (b) energy efficiency and (c) area efficiency. All figures of merit of RRC scale favorably with size. Particularly, the 1R-RRAM IMC implementation catches up on energy efficiency with the 6T6R-CMOS weighted resistor-based for large systems, and outperforms both 6T6R-CMOS IMC and GPU in area efficiency. The reduced programming latency for 6T6R-CMOS IMC implementation results in superior throughput with respect to both 1R-RRAM RRC and GPU.

ered an 8-bit current converter with $0.4\,\mathrm{ns}$ latency, $0.16\,\mathrm{mW}$ leakage power and $3.07\,\mathrm{\mu m}^2$ area [31]. For the V-ADCs, we considered a 10-bit converter with $0.5\,\mathrm{ns}$ latency, $12.5\,\mathrm{pJ}$ conversion energy and $0.01\,\mathrm{mm}^2$ area [32]. Fig. 13 shows the breakdown of the circuit area (a) and energy consumption (b) for the RRC. Being the circuit inherently memory agnostic, *i.e.* its operation is independent of the technology chosen to implement the elements of crosspoint arrays, we considered two possible realizations of the memory elements, namely an RRAM crosspoint or the nTnR-CMOScell in Fig. 9 at 6-bit precision (6T6R). For the 6T6R cell, we considered the configuration bits D_i to be stored in a static random access memory (SRAM) thus requiring a programming energy of $3\,\mathrm{fJ}$ [33]. Polysilicon resistors ($R_s = 400\,\Omega$ [34]) are assumed as conductive elements, resulting in an overall cell area of $\sim 0.46\,(\mathrm{\mu m})^2$ in $14\,\mathrm{nm}$ technology. The least resistive state corresponds to a resistance of $10\,\mathrm{k\Omega}$, which is obtained for all select bits D_i being at high value. Notably, the overall circuit area is dominated by the memory array, which quickly supersedes the area occupation of the operational amplifiers.

Figs. 13c and d show a similar breakdown for area and energy, respectively, for a 6-bit 1R-RRAM device with a resistance ranging from $100\,\mathrm{k\Omega}$ for the low-resistance state (LRS) to $100\,\mathrm{M\Omega}$ for the high resistance state (HRS), and an average programming energy of $0.6\,\mathrm{pJ}$ [35]. It is clear that the 6T6R-CMOS structure is beneficial from the viewpoint of programming energy, however is outperformed by 1R-RRAM devices in both area occupation and energy consumption in the computing phase, thanks to the higher density and lower conductance of the RRAM technology.

To assess the equivalent throughput, energy and area efficiency, we considered the computational complexity of RZF given by [14]:

$$\mathcal{O}(2N_t^3 + 6N_t^2(N_r+1) + 6N_rN_t + 2N_t) \tag{43}$$

for both uplink and downlink transmissions. For instance, assuming $N_r = 32$ and $N_t = 16$, the computational complexity is equivalent to 61984 FLOPs. Fig. 14 shows the calculated throughput (a), energy consumption (b) and area efficiency (c) for 6T6R-CMOS- and 1R-RRAM-based RRCs compared to a commercial FP32 GPU [28]. The throughput was calculated by:

$$\text{Throughput} = \frac{\text{FLOPs}}{\text{Latency [s]}} \tag{44}$$

and was measured in tera-FLOPs per second or TOPS. For the IMC system, the latency is given by the sum of programming time T_p and of the computing time T_c for RRC, *i.e.* $T_{IMC} = T_p + T_c$. For the GPU implementation, the latency is given by the sum of the data transfer time and computation latency, *i.e.* $T_{GPU} = T_{DM} + T_{CPU} \simeq 2T_{CPU}$. The latter was estimated considering the reported computational throughput at FP32 precision and the equivalent number of FLOPs for a 256×128 MIMO system following Eq. (43), yielding a computation latency of $T_{CPU} = 1.82\,\mu\text{s}$. The energy efficiency was calculated by:

$$\text{Energy efficiency} = \frac{\text{FLOPs}}{\text{Energy [J]}} \tag{45}$$

and was measured in TOPS/W. For the IMC system, we evaluated both static and dynamic power consumption for all circuit elements (programming peripherals, V-ADCs, C-DACs, operational amplifiers and memory arrays), totalling a mean energy consumption of $0.32\,\mu\text{J}$ for the 6T6R-CMOS and $0.76\,\mu\text{J}$ for the 1R-RRAM implementations respectively. For the GPU implementation, energy consumption was estimated considering the reported power consumption of $250\,\text{W}$ and the estimated latency T_{GPU}, for a grand total of $454.56\,\mu\text{J}$. The area efficiency was calculated as:

$$\text{Area efficiency} = \frac{\text{Throughput}}{\text{Area [m}^2\text{]}} \tag{46}$$

and measured in [TOPS/mm^2]. While the digital computation shows a similar throughput with respect to both nTnR- and RRAM-based implementations of RRC, the IMC approach exhibits an increase in energy efficiency by $\sim 1500\times$ and an increase in area efficiency by $\sim 6000\times$. In particular, the 6T6R-CMOS implementation outperforms the 1R-RRAM one from the viewpoint of energy efficiency, thanks to the reduced programming energy and latency. On the other hand, the RRAM implementation outperforms the CMOS one from the viewpoint of area efficiency, thanks to the high density of RRAM devices.

To assess the scaling of the performance metric, Fig. 15 shows the calculated throughput (a), energy efficiency (b), and area efficiency (c) as a function of N_r for IMC and digital implementations. All figures of merit show a positive scaling behavior with size of the MIMO system, where the numbers of receivers and transmitters were increased at constant ratio N_r/N_t. Scaling projections reveal that, for sufficiently large system, the IMC approach overcomes the GPU performance in all aspects, including throughput, owing to the intrinsic high parallelism of IMC. In particular, IMC shows an increase of throughput with $\mathcal{O}(N_r^2)$, while energy efficiency increases with $\mathcal{O}(N_r^{0.5\div1})$, with 6T6R-CMOS and 1R-RRAM implementations featuring similar efficiency at large N_r. The area efficiency remains constant for the CMOS-based 6T6R implementation due to both area and throughput scaling as $\mathcal{O}(N_r^2)$. On the other hand, the RRAM-based IMC scales favorably also in area efficiency owing to the reduced area consumption of the memory array and the resulting dominant role of the OAs area scaling trend as $\mathcal{O}(N_t)$. These results support IMC for energy efficient solution of MIMO, especially for aggressively massive implementations with large numbers of antennas and users.

VI. Conclusion

We present a novel IMC-based ridge regression circuit, capable of accelerating all typical operations of a massive MIMO transaction, including channel estimation, uplink and downlink transmissions. We study its static error with respect to floating-point precision computers and assess the design space for amplifiers and memory cells by evaluating the SER for various memory bit precisions. Our results indicate that, even with relatively low gains and memory precision, IMC-RRC is capable of closely matching the results of a digital computer with FP64 precision. The circuit is then benchmarked in realistic operating conditions in a custom simulation framework, showing an improvement in energy and area efficiency by $1500\times$ and $6000\times$, respectively, for medium-scale system sizes. Scaling projections to large scale systems allow to estimate improvements in throughput, energy and area efficiency by 2 to 4 orders of magnitude. These results support IMC as a strong candidate for highly-efficient, low-power, compact MIMO accelerators in next-generation architectures.

Acknowledgment

This work was supported in part by the European Research Council (grant ERC-2014-CoG-648635-RESCUE).

References

[1] M. A. Albreem, M. Juntti *et al.*, "Massive MIMO Detection Techniques: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3109–3132, 2019. doi: 10.1109/COMST.2019.2935810

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

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

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

[5] X. Ji, X. Hu *et al.*, "Adaptive sparse coding based on memristive neural network with applications," *Cognitive Neurodynamics*, vol. 13, no. 5, pp. 475–488, Oct. 2019. doi: 10.1007/s11571-019-09537-w

[6] M. Le Gallo, A. Sebastian *et al.*, "Mixed-Precision In-Memory Computing," *Nature Electronics*, vol. 1, no. 4, pp. 246–253, Apr. 2018. doi: 10.1038/s41928-018-0054-8

[7] Z. Sun, G. Pedretti *et al.*, "Solving matrix equations in one step with cross-point resistive arrays," *Proceedings of the National Academy of Sciences*, vol. 116, no. 10, pp. 4123–4128, Mar. 2019. doi: 10.1073/pnas.1815682116

[8] Z. Sun, E. Ambrosi *et al.*, "In-Memory PageRank Accelerator With a Cross-Point Array of Resistive Memories," *IEEE Transactions on Electron Devices*, vol. 67, no. 4, pp. 1466–1470, Apr. 2020. doi: 10.1109/TED.2020.2966908

[9] Z. Sun, G. Pedretti *et al.*, "One-step regression and classification with cross-point resistive memory arrays," *Science Advances*, vol. 6, no. 5, p. eaay2378, Jan. 2020. doi: 10.1126/sciadv.aay2378

[10] P. Mannocci, G. Pedretti *et al.*, "A Universal, Analog, In-Memory Computing Primitive for Linear Algebra Using Memristors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–11, 2021. doi: 10.1109/TCSI.2021.3122278

[11] Z. Sun, G. Pedretti *et al.*, "Time Complexity of In-Memory Solution of Linear Systems," *IEEE Transactions on Electron Devices*, vol. 67, no. 7, pp. 2945–2951, Jul. 2020. doi: 10.1109/TED.2020.2992435

[12] R. Cai, A. Ren *et al.*, "Memristor-Based Discrete Fourier Transform for Improving Performance and Energy Efficiency," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. Pittsburgh, PA, USA: IEEE, Jul. 2016, pp. 643–648. doi: 10.1109/ISVLSI.2016.124

[13] G. Yuan, C. Ding *et al.*, "Memristor crossbar-based ultra-efficient next-generation baseband processors," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. Boston, MA, USA: IEEE, Aug. 2017, pp. 1121–1124. doi: 10.1109/MWSCAS.2017.8053125

[14] R. A. Horn and C. R. Johnson, *Matrix analysis*, 2nd ed. Cambridge ; New York: Cambridge University Press, 2012.

[15] S. Loyka, "Channel capacity of MIMO architecture using the exponential correlation matrix," *IEEE Communications Letters*, vol. 5, no. 9, pp. 369–371, Sep. 2001. doi: 10.1109/4234.951380

[16] R. Chataut and R. Akl, "Massive MIMO Systems for 5G and beyond Networks—Overview, Recent Trends, Challenges, and Future Research Direction," *Sensors*, vol. 20, no. 10, p. 2753, May 2020. doi: 10.3390/s20102753

[17] W. Webb and L. Hanzo, *Modern quadrature amplitude modulation: principles and applications for fixed and wireless communications*. London : New York: Pentech ; IEEE Press, 1994.

[18] M. N. Boroujerdi, S. Haghighatshoar *et al.*, "Low-Complexity Statistically Robust Precoder/Detector Computation for Massive MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6516–6530, Oct. 2018. doi: 10.1109/TWC.2018.2860951

[19] A. N. Tikhonov and V. I. Arsenin, *Solutions of ill-posed problems*, ser. Scripta series in mathematics. Washington : New York: Winston ; distributed solely by Halsted Press, 1977.

[20] C. Peel, B. Hochwald *et al.*, "A Vector-Perturbation Technique for Near-Capacity Multiantenna Multiuser Communication—Part I: Channel Inversion and Regularization," *IEEE Transactions on Communications*, vol. 53, no. 1, pp. 195–202, Jan. 2005. doi: 10.1109/TCOMM.2004.840638

[21] H. Yang and T. L. Marzetta, "Performance of Conjugate and Zero-Forcing Beamforming in Large-Scale Antenna Systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 172–179, Feb. 2013. doi: 10.1109/JSAC.2013.130206

[22] E. Björnson, J. Hoydis *et al.*, "Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017. doi: 10.1561/2000000093

[23] H. Yin, D. Gesbert *et al.*, "A Coordinated Approach to Channel Estimation in Large-Scale Multiple-Antenna Systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 264–273, Feb. 2013. doi: 10.1109/JSAC.2013.130214

[24] D. Ielmini, "Resistive switching memories based on metal oxides: mechanisms, reliability and scaling," *Semiconductor Science and Technology*, vol. 31, no. 6, p. 063002, Jun. 2016. doi: 10.1088/0268-1242/31/6/063002

[25] H.-S. P. Wong, S. Raoux *et al.*, "Phase Change Memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201–2227, Dec. 2010. doi: 10.1109/JPROC.2010.2070050

[26] A. Goldsmith, *Wireless Communications*, 1st ed. Cambridge University Press, Aug. 2005.

[27] Dongweon Yoon, Kyongkuk Cho *et al.*, "Bit error probability of M-ary quadrature amplitude modulation," in *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No.00CH37152)*, vol. 5. Boston, MA, USA: IEEE, 2000, pp. 2422–2427. doi: 10.1109/VETECF.2000.883298

[28] "NVIDIA Quadro RTX 8000." Available at: www.nvidia.com/en-us/design-visualization/quadro/rtx-8000/

[29] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. San Francisco, CA, USA: IEEE, Feb. 2014, pp. 10–14. doi: 10.1109/ISSCC.2014.6757323

[30] B. Feinberg, R. Wong *et al.*, "An Analog Preconditioner for Solving Linear Systems," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. Seoul, Korea (South): IEEE, Feb. 2021, pp. 761–774. doi: 10.1109/HPCA51647.2021.00069

[31] S. Ishraqul Huq, S. Islam *et al.*, "Design of Low Power 8-Bit DAC Using PTM-LP Technology," in *2017 International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT)*. Warangal: IEEE, Jul. 2017, pp. 64–69. doi: 10.1109/ICRTEECT.2017.30

[32] A. Wang and C.-J. R. Shi, "A 10-bit 50-MS/s SAR ADC with 1 fJ/Conversion in 14 nm SOI FinFET CMOS," *Integration*, vol. 62, pp. 246–257, Jun. 2018. doi: 10.1016/j.vlsi.2018.03.010

[33] I. Arsovski, T. Hebig *et al.*, "A 32 nm 0.58-fJ/Bit/Search 1-GHz Ternary Content Addressable Memory Compiler Using Silicon-Aware Early-Predict Late-Correct Sensing With Embedded Deep-Trench Capacitor Noise Mitigation," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 4, pp. 932–939, Apr. 2013. doi: 10.1109/JSSC.2013.2239092

[34] "International Technology Roadmap for Semiconductors (ITRS) 2013," 2013. Available at: https://www.semiconductors.org/wp-content/uploads/2018/08/2013ExecutiveSummary.pdf

[35] F. Zahoor, T. Z. Azni Zulkifli *et al.*, "Resistive Random Access Memory (RRAM): an Overview of Materials, Switching Mechanism, Performance, Multilevel Cell (mlc) Storage, Modeling, and Applications," *Nanoscale Research Letters*, vol. 15, no. 1, p. 90, Dec. 2020. doi: 10.1186/s11671-020-03299-9

Piergiulio Mannocci (GS'21) is a PhD candidate at Politecnico di Milano, Milan, Italy. He received the B.Sc. and M.Sc. degrees at Politecnico di Milano, Milan, Italy in 2016 and 2020, respectively. His research interests include the design of analogue and mixed-signal circuits with emerging memories for in-memory linear algebra accelerators and optimization.

Enrico Melacarne received the B.Sc. and M.Sc. degrees at Politecnico di Milano, Milan, Italy in 2017 and 2021, respectively. His research interests include the design of in-memory analogue circuits for linear algebra and machine learning applications.

Daniele Ielmini is a Professor at the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy. He received the Ph.D. from Politecnico di Milano in 2000. He held visiting positions at Intel Corporation and Stanford University in 2006. His research interests include memory devices and in-memory computing circuit design. He authored/coauthored more than 300 papers in international journals and conferences. He is Associate Editor of IEEE Trans. Nanotechnology and Semiconductor Science and Technology (IOP). He received the Intel Outstanding Researcher Award in 2013, the ERC Consolidator Grant in 2014, and the IEEE-EDS Paul Rappaport Award in 2015. He is a Fellow of the IEEE.