Analysis and Designs of Analog ECC

Anxiao (Andrew) Jiang, *Senior Member, IEEE* and Xiangwu Zuo

***Abstract*—Nonvolatile resistive memories (e.g., memristors and phase-change memories) have become an important part of neuromorphic computing. In particular, crossbars of such nanoscale memories have been essential for realizing vector-matrix multiplications in analog circuits, which are widely used operations in deep neural networks. Analog error-correcting codes (Analog ECCs) have been proposed to make the operations more reliable.**

This paper explores the analysis and constructions of Analog ECCs in multiple ways. It presents a linear-programming based algorithm that computes the m-heights of Analog ECCs efficiently, which can be used to determine the error correction/detection capabilities of the codes. It presents a family of Analog ECCs based on permutations, and proves that the time complexity for determining the m-heights of such codes can be further reduced substantially. It then presents a number of newly discovered codes, which achieve state-of-the-art performance.

I. Introduction

Nonvolatile memories (NVMs) have become essential for implementing deep neural networks (DNNs) in analog circuits through in-memory computing. In recent years, DNNs have been realized in analog chips based on resistive NVMs, including phase-change memories (PCMs) and memristors [1], [5]. Such DNNs can run with substantially higher speed and power efficiency compared to digital circuits. The improvements are largely due to the efficient realization of vector-matrix multiplications, which are widely used operations in DNNs, in the crossbar architecture of NVMs cells. The challenge, however, is that the computing can be less reliable than its digital counterpart due to the various noise mechanisms in NVM cells and analog circuits. To make the vector-matrix multiplication more reliable, Analog error-correcting codes (Analog ECC) has been proposed recently [4].

Let $\mathcal{C}$ be a linear $[n, k]$ Analog ECC over $\mathbb{R}$. Let $\mathbf{c} = (c_0, c_1, \cdots, c_{n-1}) \in \mathbb{R}^n$ denote a generic codeword in $\mathcal{C}$. There are two types of additive errors that can be added to a codeword by the channel: a type of *limited-magnitude errors* (LMEs), and a type of *unlimited-magnitude errors* (UMEs), defined as follows. Let $[n\rangle$ denote the integer set $\{0, 1, \cdots, n-1\}$. Let δ and Δ be two positive real thresholds, where $\Delta > \delta > 0$. An error vector $\boldsymbol{\varepsilon} = (\varepsilon_0, \varepsilon_1, \cdots, \varepsilon_{n-1}) \in \mathbb{R}^n$ is called a *limited-magnitude error vector* (i.e., LME vector) if $\varepsilon_i \in [-\delta, \delta]$ for all $i \in [n\rangle$. Given a vector $\mathbf{e} = (e_0, e_1, \cdots, e_{n-1}) \in \mathbb{R}^n$, define its support with respective to Δ as $\text{Supp}_\Delta(\mathbf{e}) = \{i \in [n\rangle : |e_i| > \Delta\}$. The above definition can be extended to $\Delta = 0$. Then by this definition, the ordinary support of $\mathbf{e}$ is $\text{Supp}_0(\mathbf{e})$. And the Hamming weight of $\mathbf{e}$, denoted by $\text{w}_H(\mathbf{e})$, is $|\text{Supp}_0(\mathbf{e})|$. An error vector $\mathbf{e} = (e_0, e_1, \cdots, e_{n-1}) \in \mathbb{R}^n$ is called an *unlimited-magnitude error vector* (i.e., UME vector) of Hamming weight w if $\text{w}_H(\mathbf{e}) = w$. A noisy codeword $\mathbf{y} = (y_0, y_1, \cdots, y_{n-1}) \in \mathbb{R}^n$ is the sum of the codeword $\mathbf{c} \in \mathcal{C}$ and the two error vectors $\boldsymbol{\varepsilon}$ and $\mathbf{e}$, namely, $\mathbf{y} = \mathbf{c}+\boldsymbol{\varepsilon}+\mathbf{e}$.

DNNs often naturally have some tolerance of small noise (e.g, limited-magnitude errors) in the data they use, including noise in their intermediate computational results [2]. So we may consider LMEs as *tolerable* (as long as δ is small), and focus on the detection and correction of the UMEs, especially those UMEs whose magnitudes exceed the threshold Δ. For the NVM crossbar that implements the vector-matrix multiplication, the LMEs can be due to random noise in the NVM cell levels and circuits, while the UMEs can be due to more significant events such as stuck cells or short cells in the crossbar architecture. For more details on the implementation and its performance, please refer to [3] [4].

Given the above considerations, the decoding objective of Analog ECC is set as follows. The decoder for a linear $[n, k]$ Analog ECC $\mathcal{C}$ is a function $\mathcal{D} : \mathbb{R}^n \to 2^{[n\rangle}$ that returns a set of locations of UMEs. Let $\delta, \Delta \in R^+$ be positive thresholds with $\delta < \Delta$ as mentioned earlier, and let t be a nonnegative integer. We say that "the decoder $\mathcal{D}$ corrects t UMEs (with respect to the threshold pair (δ, Δ))" if for every possible vector $\mathbf{y} = \mathbf{c}+\boldsymbol{\varepsilon}+\mathbf{e}$ with $\mathbf{c} \in \mathcal{C}$ being a codeword, $\boldsymbol{\varepsilon}$ being an LME vector and $\mathbf{e}$ being an UME vector whose Hamming weight $\text{w}_H(\mathbf{e})$ is at most t, we have $\text{Supp}_\Delta(\mathbf{e}) \subseteq \mathcal{D}(\mathbf{y}) \subseteq \text{Supp}_0(\mathbf{e})$.

In spite of the importance of Analog ECCs for machine learning, the designs of such codes are still relatively limited. Most existing codes focus on the correction or detection of only one UME [4]. In this work, we develop an efficient algorithm that finds the error-correction capabilities of Analog ECCs (specifically, a quantity called *m-height*, which is analogous to minimum distance for conventional ECCs). We present and analyze a new family of codes called *Analog Permutation Codes* that enables more efficient computing of the m-heights. We also use genetic programming to search for new codes, and present a set of codes with state-of-the-art performance.

II. Main Results

The m-height of a linear $[n, k]$ Analog ECC $\mathcal{C}$ is defined as follows [4]. Let $\mathbf{x} = (x_0, x_1, \cdots, x_{n-1}) \neq (0, 0, \cdots, 0)$ be a vector in $\mathbb{R}^n$. Let $\pi : [n\rangle \to [n\rangle$ be a permutation such that $|x_{\pi(0)}| \geq |x_{\pi(1)}| \geq \cdots \geq |x_{\pi(n-1)}|$. For any $m \in [n\rangle$, the m-height of $\mathbf{x}$ is defined as $h_m(\mathbf{x}) = \left|\frac{x_{\pi(0)}}{x_{\pi(m)}}\right|$ if $x_{\pi(m)} \neq 0$, and as $h_m(\mathbf{x}) = \infty$ if $x_{\pi(m)} = 0$. For the all-zero vector $\mathbf{0} = (0, 0, \cdots, 0)$, its m-height is defined as $h_m(\mathbf{0}) = 0$ for all m. Then, the m-height of a linear $[n, k]$ code $\mathcal{C}$ over $\mathbb{R}$ is defined as $h_m(\mathcal{C}) = \max_{\mathbf{c} \in \mathcal{C}} h_m(\mathbf{c})$. The next important result was proven in [4].

A. Jiang and X. Zuo are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, 77845 USA. E-mail: ajiang@cse.tamu.edu and dkflame@tamu.edu. A full version of this paper is under review by IEEE Transactions on Information Theory.

Theorem 1. *Let $\mathcal{C}$ be a linear $[n, k]$ code. Given δ, $\Delta \in \mathbb{R}^+$ with $\delta < \Delta$ and $t \in \mathbb{Z}^+$, there exists a decoder for $\mathcal{C}$ that corrects t UMEs if and only if $\Delta/\delta \geq 2(h_{2t}(\mathcal{C}) + 1)$.*

It is, however, challenging to compute the m-height of an Analog ECC in general. (The analogous minimum distance problem for conventional ECC is usually NP-hard.) We can show that given the generator matrix G, the m-height of $\mathcal{C}$ can be found through a baseline algorithm that solves $n! \cdot 2^n$ linear-fractional programs with k variables. We further prove that there is a more efficient algorithm (shown in Theorem 2).

Let $m \in [n\rangle - \{0\}$. Let $\Psi = \{-1, 1\}^m$. Let (a, b, X, ψ) be a tuple where $a \in [n\rangle$, $b \in [n\rangle - \{a\}$, $X \subseteq [n\rangle - \{a, b\}$, $|X| = m - 1$, and $\psi = (s_0, s_1, \cdots, s_{m-1}) \in \Psi$. Let Γ denote the set of all the $n(n-1)\binom{n-2}{m-1}2^m$ such tuples.

Given a tuple $(a, b, X, \psi) \in \Gamma$, let $x_1, x_2, \cdots, x_{m-1}$ denote the $m - 1$ integers in X such that $x_1 < x_2 < \cdots < x_{m-1}$. Define $Y \triangleq [n\rangle - X - \{a, b\}$, and let $x_{m+1}, x_{m+2}, \cdots, x_{n-1}$ denote the $n-m-1$ integers in Y such that $x_{m+1} < x_{m+2} < \cdots < x_{n-1}$. Let $x_0 = a$ and $x_m = b$. Then $x_0, x_1, \cdots, x_{n-1}$ are the n distinct integers in $[n\rangle$. Let τ denote the permutation on $[n\rangle$ such that $\tau(j) = x_j$ for $j \in [n\rangle$. We call τ the *quasi-sorted permutation given* (a, b, X, ψ).

Theorem 2. *Let $\mathcal{C}$ be a linear $[n, k]$ code over $\mathbb{R}$. Let $G = (g_{i,j})_{k \times n} \in \mathbb{R}^{k \times n}$ be a generator matrix of $\mathcal{C}$ where no column is $\mathbf{0}$. Let $d(\mathcal{C})$ be the minimum distance of $\mathcal{C}$, and let $m \in \{1, 2, \cdots, \min\{d(\mathcal{C}), n-1\}\}$. Let $(a, b, X, \psi) \in \Gamma$, where $\psi = (s_0, s_1, \cdots, s_{m-1})$. Define $Y \triangleq [n\rangle - X - \{a, b\}$, and let τ be the quasi-sorted permutation given (a, b, X, ψ). Let $LP_{a,b,X,\psi}$ denote the following linear program with k real-valued variables u_0, u_1, $\cdots$, u_{k-1}: Maximize $\sum_{i \in [k\rangle}(s_0 g_{i,a}) \cdot u_i$ such that (1) $\sum_{i \in [k\rangle}(s_{\tau^{-1}(j)} g_{i,j} - s_0 g_{i,a}) \cdot u_i \leq 0$ for $j \in X$; (2) $\sum_{i \in [k\rangle}(-s_{\tau^{-1}(j)} g_{i,j}) \cdot u_i \leq -1$ for $j \in X$; (3) $\sum_{i \in [k\rangle} g_{i,b} \cdot u_i = 1$; (4) $\sum_{i \in [k\rangle} g_{i,j} \cdot u_i \leq 1$ for $j \in Y$; (5) $\sum_{i \in [k\rangle} -g_{i,j} \cdot u_i \leq 1$ for $j \in Y$. Let $z_{a,b,X,\psi}$ be the optimal objective value of $LP_{a,b,X,\psi}$ if it is feasible, and let $z_{a,b,X,\psi} = 0$ otherwise. Then $h_m(\mathcal{C}) = \max_{(a,b,X,\psi) \in \Gamma} z_{a,b,X,\psi}$.*

Theorem 2 shows how to computes the m-height by solving $n(n-1)\binom{n-2}{m-1}2^m$ linear programs, which reduces the complexity by a factor of $(m-1)! \cdot (n-m-1)! \cdot 2^{n-m}$ compared to the baseline method. Since $1 = h_0(\mathcal{C}) \leq h_1(\mathcal{C}) \leq \cdots \leq h_{n-1}(\mathcal{C})$ and $h_{d(\mathcal{C})}(\mathcal{C}) = \infty$, it also shows a way to find $d(\mathcal{C})$: compute the m-height for $m = 1, 2, 3 \cdots$ until it becomes ∞.

To further reduce the complexity of computing m-heights, we study codes with special structures. When $n = k!$, let G be a $k \times n$ matrix whose columns are distinct permutations of each other. We call a code with G as its generator matrix an *Analog Permutation Code* (APC). The next theorem shows that the time complexity of computing its m-height is lower than that of Theorem 2 by a factor of n. (There are other code constructions with special structures, such as codes concatenated with repetition codes, for which the time complexity can also be reduced. Due to space limitation we skip their details.)

Theorem 3. *Let $\mathcal{C}$ be a linear $[n, k]$ APC. Let $z_{a,b,X,\psi}$ be as defined in Theorem 2. Then $h_m(\mathcal{C}) = \max_{(0,b,X,\psi) \in \Gamma} z_{0,b,X,\psi}$.*

The algorithms for computing the m-heights of Analog ECCs enable us to construct new codes through computer search and optimization. We search for new codes based on Genetic Programming, where codes with random generator matrices (e.g., using Gaussian distributions for their elements), random Analog Permutation Codes and their punctured codes are used as the initial population. A summary of the results are shown in Table I. The table lists the m-heights of different linear $[n, k]$ Analog ECCs with state-of-the-art performance. Here the m-heights in the black color are of known codes from [4], and the m-heights of the blue color are of the new codes presented here. Notice that when $m \geq 4$, codes with finite m-heights can correct two or more UMEs; and the smaller the m-height is, the smaller the ratio Δ/δ (where Δ and δ are the two threshold values for UMEs and LMEs as described earlier) needs to be. Many of the new codes here can correct two or more errors. Some of them have finite m-heights for m as large as 8 (e.g., the $[10, 2]$ code in the table), which means they can correct up to 4 UMEs. We also note that nine of the new codes here are Analog Permutation Codes or their punctured codes.

$h_m(\mathcal{C})$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$
	[5,2]	[6,2]	[6,3]	[7,2]	[7,3]	[7,4]	[8,2]
$m=1$	1	1	1	1	1	2	1
$m=2$	1.83	1	2.87	1	2	3.15	1
$m=3$	3.25	2.28	7	1.90	3	12.56	1
$m=4$	—	4.10	—	2.78	10.14	—	2.88
$m=5$	—	—	—	4.95	—	—	3.79
$m=6$	—	—	—	—	—	—	7.16

$h_m(\mathcal{C})$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$
	[8,3]	[8,4]	[8,5]	[9,2]	[9,3]	[9,4]	[9,5]
$m=1$	1	1	2	1	1	1	2
$m=2$	2.49	3	7.18	1	1	3.38	4
$m=3$	3.71	3	22.48	1	3.05	5.42	12.51
$m=4$	9.01	20.96	—	1.92	5.76	12.32	76.49
$m=5$	20.37	—	—	3.32	12.71	99.90	—
$m=6$	—	—	—	3.88	29.92	—	—
$m=7$	—	—	—	12.76	—	—	—

$h_m(\mathcal{C})$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$	$[n,k]$
	[9,6]	[10,2]	[10,3]	[10,4]	[10,5]	[10,6]
$m=1$	2	1	1	1	1	2
$m=2$	8.56	1	1	≤ 3	4.23	4
$m=3$	48.72	1	2.12	5.00	8.90	23.06
$m=4$	—	1	4.36	7.22	29.80	273.24
$m=5$	—	1.92	5.97	21.56	334.75	—
$m=6$	—	3.88	17.18	300.92	—	—
$m=7$	—	3.88	69.44	—	—	—
$m=8$	—	28.74	—	—	—	—

TABLE I
THE m-HEIGHTS OF DIFFERENT LINEAR $[n, k]$ ANALOG ECCS

REFERENCES

[1] M. Le Gallo *et al.*, "A 64-core Mixed-signal In-memory Compute Chip Based on Phase-change Memory for Deep Neural Network Inference," in *Nature Electronics*, vol. 6, pp. 680-693, 2023.

[2] K. Huang, P. H. Siegel and A. Jiang, "Functional Error Correction for Robust Neural Networks," in *IEEE Journal on Selected Areas in Information Theory (JSAIT)*, vol. 1, no. 1, pp. 267-276, 2020.

[3] C. Li, R. M. Roth, C. Graves, X. Sheng and J. P. Strachan, "Analog Error Correcting Codes for Defect Tolerant Matrix Multiplication in Crossbars," in *Proc. IEEE International Electron Devices Meeting (IEDM)*, 2020.

[4] R. M. Roth, "Analog Error-Correcting Codes," in *IEEE Transactions on Information Theory*, vol. 66, no. 7, pp. 4075-4088, July 2020.

[5] W. Zhang *et al.*, "Edge Learning Using a Fully Integrated Neuro-inspired Memristor Chip," in *Science*, vol. 381, no. 6663, pp. 1205-1211, 2023.