Analog Error-Correcting Codes

Ron M. Roth, *Fellow, IEEE*

***Abstract*—Coding schemes are presented that provide the ability to locate computational errors above a prescribed threshold while using analog resistive devices for approximate real vector–matrix multiplication. In such devices, the matrix is programmed into the device by setting an array of resistors to have conductances proportional to the respective entries in the matrix. In the coding scheme that is considered in this work, redundancy columns are appended so that each row in the programmed matrix forms a codeword of a prescribed linear code $\mathcal{C}$ over the real field; the result of the multiplication of any input real row vector by the matrix is then also a codeword of $\mathcal{C}$. While error values within $\pm\delta$ in the entries of the result are tolerable (for some prescribed $\delta > 0$), outlying errors, with values outside the range $\pm\Delta$ (for a prescribed $\Delta \geq \delta$) should be located and corrected. As a design and analysis tool for such a setting, a certain functional is defined for the code $\mathcal{C}$, through which a characterization is obtained for the number of outlying errors that can be handled, as a function of the ratio Δ/δ. Several code constructions are then presented, primarily for the case of single outlying error handling. For this case, the coding problem is shown to be related to certain extremal problems on convex polygons.**

***Index Terms*—Analog arithmetic circuits, Approximate computation, Fault-tolerant computing, Linear codes over the real field, Vector–matrix multiplication.**

I. Introduction

Let ℓ and n be fixed positive integers and denote by $[n\rangle$ the integer set $\{j : 0 \leq j < n\}$. We consider here a computational circuit (referred to as a *dot-product engine*) which accepts as input an $\ell \times n$ matrix $A = (a_{i,j})_{i \in [\ell\rangle, j \in [n\rangle}$ over the real field $\mathbb{R}$ and a row vector $\boldsymbol{u} = (u_i)_{i \in [\ell\rangle} \in \mathbb{R}^\ell$, and computes the vector–matrix product $\boldsymbol{c} = \boldsymbol{u}A$, with addition and multiplication carried out over $\mathbb{R}$. In the applications of interest, the matrix A is modified very infrequently and, so, in effect, only $\boldsymbol{u}$ is seen as input.

A similar computational circuit was the focus of our recent work [18], except that $\boldsymbol{u}$ and A therein were assumed to have (nonnegative) integer entries in the range $\{0, 1, \ldots, q{-}1\}$, for a prescribed q. In current proposals for nanoscale implementations of such a circuit, the matrix A is realized as a crossbar array consisting of ℓ row conductors, n column conductors, and programmable nanoscale resistive components at the junctions, with the conductance of the resistor at each junction (i, j) being proportional to $a_{i,j}$. Each entry u_i of $\boldsymbol{u}$ is fed into a digital-to-analog converter (DAC) to produce a voltage level that is proportional to u_i. The product, $\boldsymbol{u}A$, is then computed by reading the currents at the (grounded) column conductors, after being fed into analog-to-digital converters (ADCs); see Figure 1 in [18]. In case A contains negative entries, we can write $A = A^+ - A^-$, where A^+ and A^- are both nonnegative, and use two circuits to compute $\boldsymbol{c}^+ = \boldsymbol{u}A^+$ and $\boldsymbol{c}^- = \boldsymbol{u}A^-$; the sought result is then the difference $\boldsymbol{c} = \boldsymbol{c}^+ - \boldsymbol{c}^-$. For implementations and applications of such devices, see, for example, [5], [11], [13], [15], and [19].

The scenario considered in [18] was that of *exact* integer vector–matrix multiplication. Accordingly, we presented a framework for self-protecting the computation against certain types of errors, primarily under the L_1-metric or the Hamming metric. Such a scenario is suitable for applications where the multiplication circuit is employed as an accelerator in computations carried out by ordinary ALUs. To attain the targeted precision, however (even with error-correction schemes), the computation may need to be broken into small integer ranges, e.g., arithmetic of 32-bit integers (corresponding to $q = 2^{32}$) broken into eight 4-bit multiplications ($q = 16$) that are shifted and added.

Here, on the other hand, we consider the *approximate* computation model where the input row ℓ-vector $\boldsymbol{u}$ and the programmed $\ell \times n$ matrix A are over $\mathbb{R}$, and so are the ideal computation $\boldsymbol{c} = \boldsymbol{u}A$ and the actually read output row n-vector $\boldsymbol{y}$; namely, we remain in the analog domain.[1] In addition, entries of $\boldsymbol{y}$ will still be regarded as "correct" if they are sufficiently close to the respective entries in $\boldsymbol{c}$; on the other hand, entries in $\boldsymbol{y}$ which significantly differ from those in $\boldsymbol{c}$ must be located and corrected (or, alternatively, detected). This model is suitable for applications that are insensitive to controlled inaccuracies, e.g., computations that are based on a model which only *estimates* a true behavior; this is likely to occur in learning applications.

Specifically, we assume that the read vector $\boldsymbol{y} \in \mathbb{R}^n$ may differ from the ideal computation $\boldsymbol{c} = \boldsymbol{u}A \in \mathbb{R}^n$ due to the effect of two events:

$$\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{\varepsilon} + \boldsymbol{e} \,, \qquad (1)$$

where $\boldsymbol{\varepsilon} = (\varepsilon_j)_{j \in [n\rangle}$ and $\boldsymbol{e} = (e_j)_{j \in [n\rangle}$ are error vectors in $\mathbb{R}^n$. The entries of $\boldsymbol{\varepsilon}$ are all within the interval $[-\delta, \delta]$, for some prescribed positive δ, and stand for small computational errors (or circuit noise), which are tolerable, while the entries of $\boldsymbol{e}$ represent outlying errors, which may be caused by factors such as stuck cells or short cells in the array (note that according to the model, a given ideal result $\boldsymbol{c}$ can change into a given

Ron M. Roth is with the Computer Science Department, Technion, Haifa 3200003, Israel. This work was done in part while visiting Hewlett Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304. Email: ronny@cs.technion.ac.il.

Part of this work was presented at the IEEE Int'l Symposium on Information Theory, Paris, France (July 2019).

[1] Thus, we will not rely on the existence of peripheral ADCs and DACs that may still be present as an interface with other hardware units. Also, while we will generally need to allow A to have negative real values (which, in turn, can be realized in the manner described earlier), our coding schemes will not be limited by any further assumption that the entries of $\boldsymbol{u}$ are nonnegative.

read vector $\boldsymbol{y}$ due to different error pairs $(\boldsymbol{\varepsilon}, \boldsymbol{e})$). Our goal is to design a coding scheme that allows to locate the outlying errors—i.e., the nonzero entries of $\boldsymbol{e}$—that are outside the interval $[-\Delta, \Delta]$, for the smallest possible Δ, and to compute bounds on (or estimates of) the values of the outlying errors, provided that the number of outlying errors does not exceed a prescribed number. (Under a more general setting, we also allow to just detect whether $\boldsymbol{e}$ is nonzero, without necessarily locating its nonzero entries.) Clearly, we must have $\Delta \geq \delta$, yet we will see that except for trivial cases, a stronger inequality must hold, namely, $\Delta \geq 4\delta$.

We are unaware of prior literature on the error protection setting that we have described. Existing applications of transmission and storage that allow δ-tolerance yet require correction of outliers (and, as such, can be seen as a degenerate case of our model where $\ell = 1$ and the vector $\boldsymbol{u}$ is just the constant 1) can be handled in the following manner: first quantize the data into grid points that are 2δ apart, and then use an error-correction scheme for the L_1 metric over the integers. While the quantization step introduces errors, they are still within the tolerable range of $\pm\delta$. Upon decoding of the received/read vector $\boldsymbol{y}$, one first finds the closest grid point to each entry of $\boldsymbol{y}$, and then applies the error-correction decoder.

Such a strategy would have worked also in our setting had the vector $\boldsymbol{u}$ been constrained to be a standard unit vector (i.e., be all-zero except for one entry which is 1). Yet, for general $\boldsymbol{u}$ and general number ℓ of rows in A, the multiplication by the entries of $\boldsymbol{u}$ would amplify the quantization error, and further amplification would result due to the summation along each column. On the other hand, if we used a finer quantization with accumulated error (for a worst-case choice of $\boldsymbol{u}$) of no more than $\pm\delta$, we would end up with too-fine quantization when $\boldsymbol{u}$ is a standard unit vector: the noise errors caused by the circuit would then be much larger than the quantization error, and we would end up with too many errors that need correction.

For other (different) models of analog coding, see, for example, [21] and the recent work [12].

As was the case in [18], the first k $(< n)$ entries in $\boldsymbol{c} = \boldsymbol{u}A$ will carry the (ordinary) result of the computation of interest, while the remaining $n-k$ entries of $\boldsymbol{c}$ will contain redundancy symbols, which can be used to detect or correct computational errors. Specifically, the programmed $\ell \times n$ matrix A will have the structure

$$A = (A' \mid A'') \ ,$$

where A' is an $\ell \times k$ matrix over $\mathbb{R}$ consisting of the first k columns of A, and A'' consists of the remaining $n-k$ columns; the computed output row vector for an input vector $\boldsymbol{u} \in \mathbb{R}^{\ell}$ will then be $\boldsymbol{c} = (\boldsymbol{c}' \mid \boldsymbol{c}'')$, where the k-prefix $\boldsymbol{c}' = \boldsymbol{u}A'$ $(\in \mathbb{R}^k)$ represents the target computation while the $(n-k)$-suffix $\boldsymbol{c}'' = \boldsymbol{u}A''$ $(\in \mathbb{R}^{n-k})$ is the redundancy part.

Similarly to [18], we encode the matrix A such that each row belongs to a linear $[n, k]$ code $\mathcal{C}$ over $\mathbb{R}$. This, in turn, will guarantee that any linear combination $\boldsymbol{c} = \boldsymbol{u}A$ belongs to $\mathcal{C}$. Thus, the code $\mathcal{C}$ completely characterizes the coding scheme.

Hereafter, for $\delta \in \mathbb{R}^+$ and $n \in \mathbb{Z}^+$, we denote by $\mathcal{Q}(n, \delta)$ the set of tolerable error vectors, namely:

$$\mathcal{Q}(n, \delta) = \left\{ \boldsymbol{\varepsilon} = (\varepsilon_0 \ \varepsilon_1 \ \ldots \ \varepsilon_{n-1}) \in \mathbb{R}^n \ : \ \max_j |\varepsilon_j| \leq \delta \right\} \ .$$

For $\boldsymbol{e} = (e_j)_j \in \mathbb{R}^n$ and $\Delta \in \mathbb{R}_{\geq 0}$, define

$$\mathsf{Supp}_{\Delta}(\boldsymbol{e}) = \left\{ j \in [n\rangle \ : \ |e_j| > \Delta \right\} \ .$$

Note that $\mathsf{Supp}_0(\boldsymbol{e})$ is the ordinary support of $\boldsymbol{e}$. The Hamming weight of $\boldsymbol{e}$ (which is the size of $\mathsf{Supp}_0(\boldsymbol{e})$) will be denoted by $\mathsf{w}(\boldsymbol{e})$. The set of all vectors of Hamming weight at most w in $\mathbb{R}^n$ will be denoted by $\mathcal{B}(n, w)$.

We next present the formal requirements from a decoder for our coding model. Given a linear $[n, k]$ code $\mathcal{C}$ over $\mathbb{R}$ (to which the rows of A belong and, therefore, so does every linear combination thereof), a decoder for $\mathcal{C}$ is a function $\mathcal{D} : \mathbb{R}^n \to 2^{[n\rangle} \cup \{\text{“e”}\}$ that returns a set of locations of outlying errors or an indication “e” that errors have been detected.[2] Given $\delta, \Delta \in \mathbb{R}^+$ and prescribed nonnegative integers τ and σ, we say that $\mathcal{D}$ corrects τ errors and detects σ additional errors (with respect to the threshold pair (δ, Δ)) if the following conditions hold for every $\boldsymbol{y}$ as in (1), where $\boldsymbol{c} \in \mathcal{C}$, $\boldsymbol{\varepsilon} \in \mathcal{Q}(n, \delta)$, and $\boldsymbol{e} \in \mathcal{B}(n, \tau{+}\sigma)$.

(D1) If $\boldsymbol{e} \in \mathcal{B}(n, \tau)$ then $\mathcal{D}(\boldsymbol{y}) \neq$ “e”.

(D2) If $\mathcal{D}(\boldsymbol{y}) \neq$ “e” then

$$\mathsf{Supp}_{\Delta}(\boldsymbol{e}) \subseteq \mathcal{D}(\boldsymbol{y}) \subseteq \mathsf{Supp}_0(\boldsymbol{e}) \ . \qquad (2)$$

Condition (D1) means that the decoder must return a (possibly empty) set of outlying error locations, whenever the number of outlying errors does not exceed τ. Condition (D2) deals with the case where the decoder returns a set of outlying error locations (whether due to condition (D1) or otherwise). The first containment in (2) means that no misses are allowed: the returned set must include the locations of all outlying errors whose values are outside the interval $[-\Delta, \Delta]$. The second containment in (2) states that false alarms are not allowed: a location should not be included if it did not contain an outlying error. Thus, when the number of outlying errors is above τ yet no more than $\tau + \sigma$, the decoder is allowed to merely detect the errors (by returning “e”); however, if it does return a set, then this set must satisfy the two containments in (2).

Note that there exists a “gray area” of outlying values which is not covered by the conditions: we allow the decoder to miss an outlier if its value is within the interval $[-\Delta, \Delta]$. Hence the significance of selecting Δ to be as small as possible.

While some of the results in this work concern the above general setting, other will focus on the case of a single outlier. We start by introducing in Section II the notion of a height profile of a linear code $\mathcal{C}$. We then use the height profile to formulate a necessary and sufficient condition for the existence of a decoder for $\mathcal{C}$ that satisfies conditions (D1)–(D2). That condition will imply, in particular, that such a decoder generally exists only when $\Delta/\delta \geq 4$.

[2] For simplicity, we exclude from our basic definition of a decoder the computation of the bounds on the values of the outlying errors; thus, in this work, error correction will generally mean finding the locations of the errors. In the sequel, we will comment about how one can compute such bounds (see the discussions at the end of Section II and following Proposition 6).

In Section III, we consider several special cases, starting with coding schemes where Δ/δ equals the floor value 4: we show that the dimension of $\mathcal{C}$ in this case must satisfy a certain upper bound, and then we completely characterize the codes that attain that bound. In the remaining part of Section III, we present simple constructions for single error detecting codes (corresponding to $(\tau, \sigma) = (0, 1)$) and single error correcting codes (i.e., $(\tau, \sigma) = (1, 0)$). In both cases, the smallest attainable ratio Δ/δ is proportional to the ratio between the length n and the redundancy r of $\mathcal{C}$, except that in the error correction case, r is constrained to be greater than $\sqrt{n}$; a variant of the construction does allow one to reach redundancy which is logarithmic in n, yet then the ratio Δ/δ grows linearly with n.

In Section IV, we present a characterization of any single detecting/correcting code through its parity-check matrix. This characterization, in turn, relates our coding problem to certain extremal problems on convex polygons. In Section V, we use this characterization to analyze a particular single error correcting construction which is MDS (namely, it has the smallest possible redundancy of 2), and show that the attainable ratio Δ/δ grows quadratically with the code length n. The same characterization is then used also in Section VI to obtain an asymptotic construction of a single error correcting code that attains a ratio Δ/δ that is sub-linear in n with a redundancy that is logarithmic in n.

While most of our work focuses on single error handling, it appears that even this simple case poses interesting challenging problems which are still open. In particular, determining fully the (tight) trade-off between the length a code, its redundancy, and the smallest attainable ratio Δ/δ, is yet to be found. Some more specific open problems will be stated in Sections III–V.

II. Correction capabilities via the height profile

In this section, we present a necessary and sufficient condition for the existence of a decoder for $\mathcal{C}$ that satisfies conditions (D1)–(D2), for given τ, σ, δ, and Δ. The condition is expressed in terms of a particular functional that we define for $\mathcal{C}$. In a way, this functional can be seen as a generalization of the notion of the minimum Hamming distance of $\mathcal{C}$.

Let $\boldsymbol{x} = (x_0 \; x_1 \; \ldots \; x_{n-1})$ be a nonzero vector in $\mathbb{R}^n$ and let $\pi : [n\rangle \to [n\rangle$ be a permutation on the coordinates of $\boldsymbol{x}$ that sorts the entries according to descending absolute values:

$$|x_{\pi(0)}| \geq |x_{\pi(1)}| \geq \ldots \geq |x_{\pi(n-1)}| \;.$$

Given an integer $m \in [n\rangle$, the *m-height* of $\boldsymbol{x}$, denoted $\mathsf{h}_m(\boldsymbol{x})$, is defined by

$$\mathsf{h}_m(\boldsymbol{x}) = \left| \frac{x_{\pi(0)}}{x_{\pi(m)}} \right| \;,$$

and for convenience we formally define $\mathsf{h}_m(\boldsymbol{x}) = \infty$ when $m \geq n$ (which amounts to padding $\boldsymbol{x}$ with arbitrarily many zeroes). Thus,

$$1 = \mathsf{h}_0(\boldsymbol{x}) \leq \mathsf{h}_1(\boldsymbol{x}) \leq \ldots \leq \mathsf{h}_{n-1}(\boldsymbol{x}) \leq \mathsf{h}_n(\boldsymbol{x}) = \infty \;,$$

and $\mathsf{w}(\boldsymbol{x})$ is the smallest $m \in [n\rangle$ for which $\mathsf{h}_m(\boldsymbol{x}) = \infty$. E.g., for $\boldsymbol{x} = (-3 \; 6 \; 1 \; 0 \; 3)$ we have: $\mathsf{h}_0(\boldsymbol{x}) = 1$, $\mathsf{h}_1(\boldsymbol{x}) = \mathsf{h}_2(\boldsymbol{x}) = 2$, $\mathsf{h}_3(\boldsymbol{x}) = 6$, and $\mathsf{h}_4(\boldsymbol{x}) = \mathsf{h}_5(\boldsymbol{x}) = \infty$.

For the all-zero vector, we formally define $\mathsf{h}_m(\mathbf{0}) = 0$ for every $m \geq 0$.

The m-height of a linear $[n, k]$ code $\mathcal{C}$ over $\mathbb{R}$ is defined by

$$\mathsf{h}_m(\mathcal{C}) = \max_{\boldsymbol{c} \in \mathcal{C}} \mathsf{h}_m(\boldsymbol{c}) \;.$$

Note that the maximum—possibly ∞—indeed exists: since $\mathsf{h}_m(\boldsymbol{c}) = \mathsf{h}_m(a \cdot \boldsymbol{c})$ for every nonzero real scalar a, the maximum can be assumed to be taken over all codewords $\boldsymbol{c} \in \mathcal{C}$ in the compact space $\{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_\infty = 1\}$ (say). When $k > 0$, the minimum distance $\mathsf{d}(\mathcal{C})$ of $\mathcal{C}$ is related to its height values by

$$\mathsf{d}(\mathcal{C}) = \min \Big\{ m \in [n\rangle \;:\; \mathsf{h}_m(\mathcal{C}) = \infty \Big\} \;.$$

We mention in passing that one can also define a kind of an inverse functional of a height of a vector (and then extend the definition to codes), as follows. Given a real $\eta > 1$, the *η-width* of $\boldsymbol{x} \in \mathbb{R}^n$ equals the smallest m for which $\mathsf{h}_m(\boldsymbol{x}) \geq \eta$. In particular, the ∞-width of $\boldsymbol{x}$ equals its Hamming weight. Note, however, that for finite η, the width does not generally satisfy the triangle inequality; e.g., the η-width of each of the two n-vectors

$$(\eta \; 1 \; 1 \; \ldots \; 1) \quad \text{and} \quad (1{-}\eta \;\; 0 \; 0 \; \ldots \; 0)$$

equals 1, while that of their sum equals n.

The next theorem is the main result of this section.

Theorem 1. *Let $\mathcal{C}$ be a linear $[n, k]$ code over $\mathbb{R}$. Given $\delta, \Delta \in \mathbb{R}^+$ and nonnegative integers τ and σ at least one of which is nonzero, there exists a decoder for $\mathcal{C}$ that corrects τ errors and detects additional σ errors (with respect to the threshold pair (δ, Δ)), if and only if*

$$\Delta \geq 2(\mathsf{h}_{2\tau+\sigma}(\mathcal{C}) + 1)\delta \;.$$

In particular, such a decoder exists only if $2\tau + \sigma < \mathsf{d}(\mathcal{C})$.

For the proof of sufficiency we will make use of the following notation and lemma. Given a linear $[n, k{>}0]$ code $\mathcal{C}$ over $\mathbb{R}$, $\delta \in \mathbb{R}^+$, and nonnegative integers τ and σ, we define for every $\boldsymbol{y} \in \mathbb{R}^n$ the set:

$$\mathcal{E}(\boldsymbol{y}) = \mathcal{E}_{\mathcal{C}}(\boldsymbol{y}, \delta) = \Big\{ \boldsymbol{e} \in \mathcal{B}(n, \tau{+}\sigma) \;: \\ \boldsymbol{e} = \boldsymbol{y} - \boldsymbol{c} - \boldsymbol{\varepsilon} \text{ for some } \boldsymbol{c} \in \mathcal{C} \text{ and } \boldsymbol{\varepsilon} \in \mathcal{Q}(n, \delta) \Big\} \;,$$

namely, $\mathcal{E}(\boldsymbol{y})$ consists of all the candidates in $\mathcal{B}(n, \tau{+}\sigma)$ of an outlying error vector $\boldsymbol{e}$, given that $\boldsymbol{y}$ is the received vector.

Lemma 2. *Given a linear $[n, k]$ code $\mathcal{C}$ over $\mathbb{R}$ and nonnegative integers τ and σ, let $\delta, \Delta \in \mathbb{R}^+$ be such that*

$$\Delta \geq 2(\mathsf{h}_{2\tau+\sigma}(\mathcal{C}) + 1)\delta \;.$$

Suppose that $\boldsymbol{y} \in \mathbb{R}^n$ is such that $\mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n, \tau) \neq \emptyset$. Then for every two vectors $\boldsymbol{e}, \boldsymbol{e}' \in \mathcal{E}(\boldsymbol{y})$,

$$\mathsf{Supp}_\Delta(\boldsymbol{e}') \subseteq \mathsf{Supp}_0(\boldsymbol{e}) \;.$$

Proof of Lemma 2. Let $\boldsymbol{e}, \boldsymbol{e}', \boldsymbol{e}^* \in \mathcal{E}(\boldsymbol{y})$ be such that $\boldsymbol{e}^* \in \mathcal{B}(n, \tau)$. Then

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{c} + \boldsymbol{\varepsilon} + \boldsymbol{e} \\ &= \boldsymbol{c}' + \boldsymbol{\varepsilon}' + \boldsymbol{e}' \\ &= \boldsymbol{c}^* + \boldsymbol{\varepsilon}^* + \boldsymbol{e}^* \;, \end{aligned}$$

where $\boldsymbol{c}, \boldsymbol{c}', \boldsymbol{c}^* \in \mathcal{C}$ and $\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}', \boldsymbol{\varepsilon}^* \in \mathcal{Q}(n, \delta)$. It follows that

$$\begin{aligned} \boldsymbol{c} - \boldsymbol{c}^* &= \boldsymbol{\varepsilon}^* + \boldsymbol{e}^* - \boldsymbol{\varepsilon} - \boldsymbol{e} & (3) \\ \boldsymbol{c}^* - \boldsymbol{c}' &= \boldsymbol{\varepsilon}' + \boldsymbol{e}' - \boldsymbol{\varepsilon}^* - \boldsymbol{e}^* \; , & (4) \end{aligned}$$

i.e., the right-hand sides of (3)–(4) are codewords of $\mathcal{C}$ and, as such, their m-heights cannot exceed $\mathsf{h}_m(\mathcal{C})$, for every $m \geq 0$.

Suppose to the contrary that $\mathsf{Supp}_\Delta(\boldsymbol{e}') \not\subseteq \mathsf{Supp}_0(\boldsymbol{e})$. Then $e_j = 0$ yet $|e'_j| > \Delta$ for some position $j \in [n\rangle$; without loss of generality we assume hereafter that $e'_j > \Delta$. Writing $w = 2\tau + \sigma$, we have $\mathsf{w}(\boldsymbol{e}) + \mathsf{w}(\boldsymbol{e}^*) \leq w$ and $\mathsf{w}(\boldsymbol{e}') + \mathsf{w}(\boldsymbol{e}^*) \leq w$, and, so, from (3)–(4) we get the following two inequalities:

$$\begin{aligned} \varepsilon^*_j + e^*_j - \varepsilon_j - e_j &\leq \mathsf{h}_w(\mathcal{C}) \cdot \delta \\ \varepsilon'_j + e'_j - \varepsilon^*_j - e^*_j &\leq \mathsf{h}_w(\mathcal{C}) \cdot \delta \; . \end{aligned}$$

Summing these inequalities yields:

$$e'_j - e_j \leq 2\, \mathsf{h}_w(\mathcal{C}) \cdot \delta + \varepsilon_j - \varepsilon'_j \leq 2(\mathsf{h}_w(\mathcal{C}) + 1)\delta \leq \Delta \; ,$$

thereby contradicting the assumption that $e'_j - e_j = e'_j > \Delta$. □

Proof of Theorem 1. Let w denote again the value $2\tau + \sigma$. We start by proving necessity and assume to the contrary that a decoder exists for some nonnegative $\Delta < 2(\mathsf{h}_w(\mathcal{C}) + 1)\delta$. We first consider the case where $\mathsf{h}_w(\mathcal{C}) < \infty$, namely, $w = 2\tau + \sigma < \mathsf{d}(\mathcal{C})$. Let $\boldsymbol{c}' = (c'_j)_{j \in [n\rangle} \in \mathcal{C}$ be such that $\mathsf{h}_w(\boldsymbol{c}') = \mathsf{h}_w(\mathcal{C})$. Without loss of generality we assume that

$$c'_0 = |c'_0| \geq |c'_1| \geq \ldots \geq |c'_w| \geq \ldots \geq |c'_{n-1}|$$

and that

$$|c'_w| = \begin{cases} 2\delta & \text{if } \mathcal{C} \neq \{\boldsymbol{0}\} \\ 0 & \text{otherwise} \end{cases} .$$

In either case,

$$c'_0 = \mathsf{h}_w(\mathcal{C}) \cdot |c'_w| = 2\, \mathsf{h}_w(\mathcal{C}) \cdot \delta \; . \qquad (5)$$

Let

$$\boldsymbol{y} = \boldsymbol{\varepsilon} + \boldsymbol{e} \; ,$$

where the entries of $\boldsymbol{\varepsilon} = (\varepsilon_j)_{j \in [n\rangle}$ are defined by

$$\varepsilon_j = \begin{cases} -\delta & \text{if } j \in [w\rangle \\ c'_j/2 & \text{otherwise} \end{cases}$$

and the entries of $\boldsymbol{e} = (e_j)_{j \in [n\rangle}$ are defined by

$$e_j = \begin{cases} c'_j + 2\delta & \text{if } j \in [\tau\rangle \\ 0 & \text{otherwise} \end{cases} .$$

Note that $\boldsymbol{\varepsilon} \in \mathcal{Q}(n, \delta)$ and that $\boldsymbol{e} \in \mathcal{B}(n, \tau)$; moreover, when $\tau > 0$,

$$e_0 = c'_0 + 2\delta = 2(\mathsf{h}_w(\mathcal{C}) + 1)\delta > \Delta \; .$$

Hence, upon receiving $\boldsymbol{y} = \boldsymbol{0} + \boldsymbol{\varepsilon} + \boldsymbol{e}$, the decoder must have the following return value:

- (in case $\tau = 0$) the empty set, or—
- (in case $\tau > 0$) a set of locations that contains the index 0.

Yet $\boldsymbol{y}$ can also be written as

$$\boldsymbol{y} = \boldsymbol{c}' + \boldsymbol{\varepsilon}' + \boldsymbol{e}' \; ,$$

where the entries of $\boldsymbol{\varepsilon} = (\varepsilon_j)_{j \in [n\rangle}$ are defined by

$$\varepsilon'_j = \begin{cases} \delta & \text{if } j \in [w\rangle \\ -c'_j/2 & \text{otherwise} \end{cases}$$

and the entries of $\boldsymbol{e}' = (e'_j)_{j \in [n\rangle}$ are defined by

$$e'_j = \begin{cases} -c'_j - 2\delta & \text{if } j \in [w\rangle \setminus [\tau\rangle \\ 0 & \text{otherwise} \end{cases} .$$

Note that $\boldsymbol{\varepsilon}' \in \mathcal{Q}(n, \delta)$ and that e'_0 is either[3] $-c'_0 - 2\delta = -2(\mathsf{h}_w(\mathcal{C}) + 1)\delta < -\Delta$ (if $\tau = 0$) or 0 (if $\tau > 0$). Hence, upon receiving $\boldsymbol{y} = \boldsymbol{c}' + \boldsymbol{\varepsilon}' + \boldsymbol{e}'$, the decoder must have one of the following return values:

- "e",
- (in case $\tau = 0$) a set of locations that contains the index 0, or—
- (in case $\tau > 0$) a set of locations that does not contain the index 0.

Either way, we have exhibited conflicting requirements from the decoder, thereby reaching a contradiction.

When $\mathsf{h}_w(\mathcal{C}) = \infty$ (i.e, $w \geq \mathsf{d}(\mathcal{C})$) yet $\Delta < \infty$, our proof is still valid if we take $\boldsymbol{c}'$ to be of Hamming weight $\mathsf{w}(\boldsymbol{c}') = \mathsf{d}(\mathcal{C})$ and $c'_0 > \Delta$ (say); note that $c'_w = 0$ (and so (5) is not applicable).

We now turn to proving sufficiency: we assume that $\Delta \geq 2(\mathsf{h}_w(\mathcal{C}) + 1)\delta$ and present a decoder $\mathcal{D}$ for $\mathcal{C}$ that corrects τ errors and detects σ additional errors.

For $\boldsymbol{y} \in \mathbb{R}^n$, define the following intersection:

$$\Lambda(\boldsymbol{y}) = \cap_{\boldsymbol{e} \in \mathcal{E}(\boldsymbol{y})} \mathsf{Supp}_0(\boldsymbol{e}) \; .$$

The return value of $\mathcal{D}$ for $\boldsymbol{y}$ is defined by

$$\mathcal{D}(\boldsymbol{y}) = \begin{cases} \Lambda(\boldsymbol{y}) & \text{if } \mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n, \tau) \neq \emptyset \\ \text{“e”} & \text{otherwise} \end{cases} . \qquad (6)$$

Let $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{\varepsilon} + \boldsymbol{e}$ now be a particular received vector, where $\boldsymbol{c} \in \mathcal{C}$, $\boldsymbol{\varepsilon} \in \mathcal{Q}(n, \delta)$, and $\boldsymbol{e} \in \mathcal{B}(n, \tau{+}\sigma)$. Condition (D1) is obviously satisfied by $\mathcal{D}$: if $\boldsymbol{e} \in \mathcal{B}(n, \tau)$, then necessarily $\mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n, \tau) \neq \emptyset$ and therefore $\mathcal{D}(\boldsymbol{y}) \neq$ "e".

Next, we prove that condition (D2) holds. Clearly, $\Lambda(\boldsymbol{y}) \subseteq \mathsf{Supp}_0(\boldsymbol{e})$, thereby establishing the second containment in (2). As for the first containment, observe that $\mathcal{D}(\boldsymbol{y}) \neq$ "e" implies by our definition of $\mathcal{D}$ that $\mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n, \tau) \neq \emptyset$. Hence, by Lemma 2, $\mathsf{Supp}_\Delta(\boldsymbol{e}) \subseteq \mathsf{Supp}_0(\boldsymbol{e}')$ for every $\boldsymbol{e}' \in \mathcal{E}(\boldsymbol{y})$ and, so, $\mathsf{Supp}_\Delta(\boldsymbol{e}) \subseteq \Lambda(\boldsymbol{y})$. □

Example 1. Let $\mathcal{C}$ be the $[n, 1]$ repetition code over $\mathbb{R}$, which is generated by the all-one vector $\boldsymbol{1} \in \mathbb{R}^n$. We provide an explicit description of the decoder in (6) for $\mathcal{C}$, for $\tau \leq (n{-}1)/2$ and $\sigma = n - 1 - 2\tau$.

Given a received vector $\boldsymbol{y} = (y_j)_{\in [n\rangle}$ in $\mathbb{R}^n$, we assume for convenience of notation that its entries are permuted so that

$$y_0 \geq y_1 \geq \ldots \geq y_{n-1} \; . \qquad (7)$$

With each entry, we associate an interval $\mathcal{I}_j = [y_j - \delta, y_j + \delta]$. Figure 1 shows an example of a vector $\boldsymbol{y}$ for $n = 12$, with the respective intervals.

[3] Recall that $w > 0$ since τ and σ are not both zero.

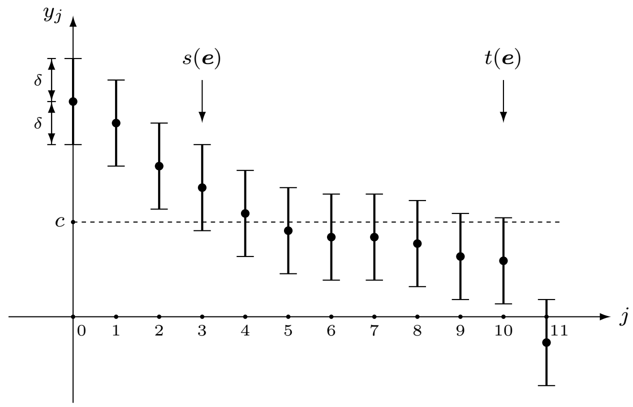

Fig. 1. Received vector $\boldsymbol{y}$ for Example 1.

A vector $\boldsymbol{e} \in \mathbb{R}^n$ belongs to $\mathcal{E}(\boldsymbol{y})$ if and only if $|\mathsf{Supp}_0(\boldsymbol{e})| \le \tau + \sigma$ and

$$\bigcap_{j \in [n\rangle \setminus \mathsf{Supp}_0(\boldsymbol{e})} \mathcal{I}_j \neq \emptyset \; . \tag{8}$$

Indeed, each element c that belongs to the intersection (8) corresponds to a codeword $\boldsymbol{c} = c \cdot \mathbf{1}$ such that $\boldsymbol{y} - \boldsymbol{c} - \boldsymbol{e} \in \mathcal{Q}(n, \delta)$. Denote by $s(\boldsymbol{e})$ and $t(\boldsymbol{e})$ the smallest and largest indexes, respectively, in $[n\rangle \setminus \mathsf{Supp}_0(\boldsymbol{e})$; note that

$$t(\boldsymbol{e}) - s(\boldsymbol{e}) \ge n - \tau - \sigma - 1 = \tau \; .$$

Due to the assumed ordering on the entries of $\boldsymbol{y}$, it follows that (8) is equivalent to

$$\mathcal{I}_{s(\boldsymbol{e})} \cap \mathcal{I}_{t(\boldsymbol{e})} \neq \emptyset$$

which, in turn, is equivalent to

$$y_{s(\boldsymbol{e})} \le y_{t(\boldsymbol{e})} + 2\delta \; .$$

E.g., for the outlying error vector $\boldsymbol{e}$ in Figure 1 we have[4] $s(\boldsymbol{e}) = 3$ and $t(\boldsymbol{e}) = 10$.

Conversely, any index $j \in [n{-}\tau\rangle$ such that

$$y_j \le y_{j+\tau} + 2\delta \tag{9}$$

defines a set $\{j, j{+}1, \ldots, j{+}\tau\}$ which does not intersect the support of a vector in $\mathcal{E}(\boldsymbol{y})$. For vectors in $\mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n, \tau)$, the condition (9) is replaced by

$$y_j \le y_{j+n-1-\tau} + 2\delta$$

(where now $j \in [\tau{+}1\rangle$). E.g., according to Figure 1, there exists an outlying error vector in $\mathcal{B}(n{=}12, \tau{=}4)$.

These observations lead to the decoding algorithm shown in Figure 2. The "if" clause checks whether $\mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n, \tau)$ is empty. E.g., for the vector $\boldsymbol{y}$ in Figure 1, the return value will be "e" if and only if $\tau < 4$. The "else" clause then computes the set $\Lambda(\boldsymbol{y})$. Notice that from the "if" clause it already follows that $y_s \le y_{s+n-\tau-1} + 2\delta$ for some $s \in [\tau{+}1\rangle$ which, in turn, implies that $y_j \le y_{j+\tau} + 2\delta$ for $j = s, s{+}1, \ldots, n{+}s{-}2\tau{-}1$.

For the vector $\boldsymbol{y}$ in Figure 1 and for $\tau = 4$, the returned set will be $\{0, 1, 11\}$, since $y_j > y_{j+4} + 2\delta$ for $j = 0, 1$ yet $y_j \le y_{j+4} + 2\delta$ for $j = 2, 3$, and $y_7 > y_{11} + 2\delta$ yet $y_6 \le y_{10} + 2\delta$. In this example $s = s(\boldsymbol{e}) = 3$ and, so, from the "if" clause we already get that $y_j \le y_{j+4} + 2\delta$ when $j = 3, 4, 5, 6$. □

[4] According to the figure, the indexes 0, 1, 2, and 11 must belong to $\mathsf{Supp}_0(\boldsymbol{e})$ (i.e., $\mathsf{w}(\boldsymbol{e}) \ge 4$). Yet, $\mathsf{Supp}_0(\boldsymbol{e})$ may also include indexes between 3 and 10. In particular, $s(\boldsymbol{e})$ could be larger than 3 and $t(\boldsymbol{e})$ could be smaller than 10.

Input: $\boldsymbol{y} = (y_j)_{j \in [n\rangle} \in \mathbb{R}^n$ such that (7) holds.
Output: $\mathcal{D}(\boldsymbol{y})$, which is either a subset of $[n\rangle$, or "e".
If $y_j > y_{j+n-1-\tau} + 2\delta$ for all $j \in [\tau{+}1\rangle$ then
  Return "e";
Else {
  Let $s \in [\tau{+}1\rangle$ be such that $y_s \le y_{s+n-1-\tau} + 2\delta$;
  $i \leftarrow \min\{j \in [s{+}1\rangle \,:\, y_j \le y_{j+\tau} + 2\delta\}$;
  $i' \leftarrow \max\{j \in [n\rangle \setminus [n{+}s{-}\tau{-}1\rangle \,:\, y_{j-\tau} \le y_j + 2\delta\}$;
  Return $[i\rangle \cup ([n\rangle \setminus [i'{+}1\rangle)$.
}

Fig. 2. Decoder $\boldsymbol{y} \mapsto \mathcal{D}(\boldsymbol{y})$ for the repetition code.

Remark 1. In the proof of Theorem 1, we can obtain an alternative proof of sufficiency by replacing $\Lambda(\boldsymbol{y})$ in the definition of $\mathcal{D}$ in (6) with the union

$$\mathrm{V}(\boldsymbol{y}) = \cup_{\boldsymbol{e} \in \mathcal{E}(\boldsymbol{y})} \mathsf{Supp}_\Delta(\boldsymbol{e}) \; .$$

Doing so, the first containment in (2) is immediate, while the second containment follows from Lemma 2. Note that Lemma 2 implies that $\mathrm{V}(\boldsymbol{y}) \subseteq \Lambda(\boldsymbol{y})$, provided that $\mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n, \tau) \neq \emptyset$. Moreover, it follows from the lemma that $\Lambda(\boldsymbol{y})$ and $\mathrm{V}(\boldsymbol{y})$ are the largest and smallest sets, respectively, that any decoder can return for any $\boldsymbol{y} \in \mathbb{R}^n$ for which $\mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n, \tau) \neq \emptyset$. □

Remark 2. The case $\tau = \sigma = 0$ is excluded from Theorem 1, since a decoder that always returns the empty set satisfies conditions (D1)–(D2) for any $\Delta \ge 0$. □

Remark 3. The proof of Theorem 1 covers also the trivial code $\mathcal{C} = \{\mathbf{0}\}$, in which case the condition in the theorem becomes $\Delta \ge 2\delta$. For nontrivial codes we must have $\Delta \ge 4\delta$. In Section III-A, we prove an upper bound of $n/(2\tau{+}\sigma{+}1)$ on the dimension of codes that attain the floor value of 4δ and show that this upper bound is attained only by a Cartesian power of the repetition code. □

Remark 4. When a coordinate is added to a linear $[n, k]$ code to form a linear $[n{+}1, k]$ code, the minimum distance of the code cannot decrease. On the other hand, the m-heights may increase, thereby adversely affecting the error correction capabilities of the code. □

Not too surprisingly, we see from Theorem 1 that it is the ratio Δ/δ (rather than the individual values of δ and Δ) which determines whether there exists a decoder satisfying (D1)–(D2) exists. Therefore, unless otherwise stated, we will assume throughout that $\delta = 1$. Accordingly, we introduce the notation

$$\Gamma_w(\mathcal{C}) = 2\,\mathsf{h}_w(\mathcal{C}) + 2 \; ,$$

so that $\Gamma_{2\tau+\sigma}(\mathcal{C})$ is the smallest Δ for which there exists a decoder for $\mathcal{C}$ that corrects τ errors and detects σ additional errors, with respect to the threshold pair $(1, \Delta)$.

Once outlying error locations have been found, one can, in principle, compute lower and upper bounds on the error values, using linear programming. Specifically, let $\mathcal{C}$ be a linear $[n, k]$ code over $\mathbb{R}$ with a generator matrix G. Given τ, σ, and $\boldsymbol{y}$ (and assuming that $\delta = 1$), suppose that the return value of the decoder is a set $\mathcal{D}(\boldsymbol{y}) = \mathcal{J}$ of size[5] $|\mathcal{J}| \le \tau$. For every superset $\mathcal{J}^* \supseteq \mathcal{J}$ of size τ and every index $t \in \mathcal{J}$, one can solve the following two linear programming problems in the real variables $\boldsymbol{x} = (x_i)_{i \in [k\rangle} \in \mathbb{R}^k$, $\boldsymbol{\varepsilon} = (\varepsilon)_{j \in [n\rangle} \in \mathbb{R}^n$, and $(e_j)_{j \in \mathcal{J}^*} \in \mathbb{R}^\tau$, where $e_j \equiv 0$ for $j \notin \mathcal{J}^*$:

$$\left.\begin{array}{l} \textit{Minimize (respectively, maximize) } e_t\textit{, subject to:} \\ \qquad \begin{array}{rcl} \boldsymbol{y} & = & \boldsymbol{x}G + \boldsymbol{e} + \boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} & \le & \mathbf{1} \\ \boldsymbol{\varepsilon} & \ge & -\mathbf{1} \end{array} \end{array}\right\} \tag{10}$$

with the inequalities holding componentwise. The largest (respectively, smallest) value of e_t obtained by enumerating over all supersets $\mathcal{J}^* \supseteq \mathcal{J}$ of size τ yields an upper (respectively, lower) bound on e_t. In general, this procedure is only a conceptual method for calculating bounds on the error values, as the enumeration over $\mathcal{J}^*$ may result in an exponential algorithm. Evaluating codes not just by the number of errors that they can locate, but also by the proximity between the lower and upper bounds that their decoders can provide on the error values (as well as by the complexity of such decoders), is left for future work.

III. Several special cases

In this section, we discuss constructions for several special cases. First, in Section III-A, we characterize the codes for which the value $\Gamma_w(\cdot)$ is minimized. Then, in Section III-B, we describe a construction for single error detection, followed in Section III-C by a construction for single error correction.

A. The floor case $\Gamma_w(\mathcal{C}) = 4$

As pointed out in Remark 3, when $\mathcal{C} \ne \{\mathbf{0}\}$ and $w = 2\tau + \sigma > 0$, we must have $\Gamma_w(\mathcal{C}) \ge 4$, with equality holding if and only if $\mathsf{h}_w(\mathcal{C}) = 1$. The next theorem provides an upper bound on the dimension of such codes.

Theorem 3. *Let $\mathcal{C}$ be a linear $[n, k]$ code over $\mathbb{R}$ and let $w \in [n\rangle$ be such that $\mathsf{h}_w(\mathcal{C}) = 1$. Then*

$$k \le \frac{n}{w+1} , \tag{11}$$

and equality is attained if and only if $w + 1$ divides n and $\mathcal{C}$ is the $(n/(w+1))$-fold Cartesian power of the $[w+1, 1]$ repetition code.

Proof. By possibly increasing w (thereby obtaining a tighter inequality in (11)), we can assume that w is the largest in $[n\rangle$ for which $\mathsf{h}_w(\mathcal{C}) = 1$. Our proof is by induction on k, for every $n \ge k$ and $w \in [n\rangle$. If $k = 1$ then (11) is immediate, and equality holds if and only if $w = n-1$, in which case $\mathsf{d}(\mathcal{C}) = n$ and therefore $\mathcal{C}$ is the $[w+1, 1]$ repetition code.

Turning to the induction step, we assume that $k > 1$ (which implies that $w < n-1$). Let $\boldsymbol{c} = (c_j)_{j \in [n\rangle}$ be a codeword in $\mathcal{C}$ such that $\mathsf{h}_w(\boldsymbol{c}) = 1$ and $\mathsf{h}_{w+1}(\boldsymbol{c}) > 1$; without loss of generality we can assume that the $(w+1)$-prefix of $\boldsymbol{c}$ is the all-one vector $\mathbf{1} \in \mathbb{R}^{w+1}$ and that the remaining entries in $\boldsymbol{c}$ are all in the open interval $(-1, 1)$. We now claim that the $(w+1)$-prefix of any $\boldsymbol{c}' \in \mathcal{C}$ must be a multiple of $\mathbf{1}$. Otherwise, for sufficiently small positive a, the codeword $\boldsymbol{c}'' = \boldsymbol{c} + a \cdot \boldsymbol{c}'$ would satisfy the following three conditions for some $\epsilon \in [0, 1)$:

- the $(w+1)$-prefix of $\boldsymbol{c}''$ would not be a multiple of $\mathbf{1}$, i.e., it would contain at least two distinct entries,
- the entries of that prefix would all be at least $1-\epsilon$, and—
- the remaining entries of $\boldsymbol{c}''$ would all be in the interval $[-1+\epsilon, 1-\epsilon]$.

Yet these three conditions would imply that $\mathsf{h}_w(\boldsymbol{c}'') > 1 = \mathsf{h}_w(\mathcal{C})$, which is impossible.

It follows that $\mathcal{C}$ has a $k \times n$ generator matrix of the form

$$G = \left(\begin{array}{cccc|cccc} 1 & 1 & \ldots & 1 & c_{w+1} & c_{w+2} & \ldots & c_{n-1} \\ \hline & & & & & & & \\ & & 0 & & & G^* & & \\ & & & & & & & \end{array} \right) , \tag{12}$$

for some $(k-1) \times (n-w-1)$ matrix G^*. Denote by $\mathcal{C}^*$ the linear $[n-w-1, k-1]$ code that is generated by G^*. Clearly, $\mathsf{h}_w(\mathcal{C}^*) \le \mathsf{h}_w(\mathcal{C}) = 1$, and since $k - 1 > 0$ we in fact have $\mathsf{h}_w(\mathcal{C}^*) = 1$. Applying the induction hypothesis to $\mathcal{C}^*$ yields

$$k - 1 \le \frac{n - w - 1}{w + 1} = \frac{n}{w + 1} - 1 , \tag{13}$$

which proves (11). Now, equality in (11) implies equality in (13); so, by the induction hypothesis, $\mathcal{C}^*$ is then the $(k-1)$-fold Cartesian power of the $[w+1, 1]$ repetition code. This means that for $i = 1, 2, \ldots, k-1$, the following vectors are codewords of $\mathcal{C}$:

$$\boldsymbol{c}_i^* = (\underbrace{\mathbf{0}\ \mathbf{0}\ \ldots\ \mathbf{0}}_{i \text{ times}}\ \mathbf{1}\ \underbrace{\mathbf{0}\ \mathbf{0}\ \ldots\ \mathbf{0}}_{k-1-i \text{ times}})$$

where $\mathbf{0}, \mathbf{1} \in \mathbb{R}^{w+1}$. As such, these codewords can be taken as the last $k-1$ rows in the matrix G in (12). Moreover, each of these codewords can play the role of $\boldsymbol{c}$ in our induction proof. Indeed, taking $\boldsymbol{c}_i^*$ instead of $\boldsymbol{c}$ and selecting $\boldsymbol{c}'$ to be $\boldsymbol{c}$, we conclude that for each $i \in [k\rangle$, the $(w+1)$-subvector

$$(c_{i(w+1)}\ c_{i(w+1)+1}\ \ldots c_{i(w+1)}\ c_{i(w+1)+w})$$

of $\boldsymbol{c}$ must be a multiple of $\mathbf{1}$. Therefore, G generates the k-fold Cartesian power of the $[w+1, 1]$ repetition code. □

B. Single error detection

In this section, we present a simple construction for detecting a single error (i.e., $\tau = 0$ and $\sigma = 1$ and, therefore, $w = 2\tau + \sigma = 1$).

Given positive integers r and n such that $r \le n$, let H be a real $r \times n$ matrix over $\{0, 1\}$ that satisfies the following properties:

[5] If a decoder satisfying conditions (D1)–(D2) returns a set larger than τ, we can always modify it to return "e" instead, without violating those conditions.

(P1) each column in H is a standard unit vector, i.e., it contains exactly one 1, and—

(P2) the number of 1s in each row is either $\lfloor n/r \rfloor$ or $\lceil n/r \rceil$.

Proposition 4. *Let $\mathcal{C}$ be a linear $[n, k{=}n{-}r]$ code over $\mathbb{R}$ with an $r \times n$ parity-check matrix H that satisfies (P1)–(P2). Then*

$$\Gamma_1(\mathcal{C}) \le 2 \cdot \lceil n/r \rceil \; . \tag{14}$$

For the case where r divides n, the code in the proposition is—up to permutation of coordinates—the dual code of the r-fold Cartesian power of the $[n/r, 1]$ repetition code discussed in Section III-A.

Proof of Proposition 4. Write $\theta = \lceil n/r \rceil$ and consider a decoder $\mathcal{D}$ that, for a received vector $\boldsymbol{y} \in \mathbb{R}^n$, computes the syndrome $\boldsymbol{s} = H\boldsymbol{y}^\top$ and returns an empty set if $\boldsymbol{s} \in \mathcal{Q}(r, \theta)$; otherwise $\mathcal{D}$ returns "e".

Assuming that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{\varepsilon} + \boldsymbol{e}$ where $\boldsymbol{\varepsilon} \in \mathcal{Q}(n, 1)$ and $\boldsymbol{e} = (e_j)_{j \in [n\rangle} \in \mathcal{B}(n, 1)$, the following observations are easily obtained from $\boldsymbol{s} = H\boldsymbol{y}^\top = H\boldsymbol{\varepsilon}^\top + H\boldsymbol{e}^\top$.

- If $\boldsymbol{e} = \boldsymbol{0}$ then $\boldsymbol{s} = H\boldsymbol{\varepsilon}^\top \in \mathcal{Q}(r, \theta)$.
- If $|e_t| > 2\theta$ for some $t \in [n\rangle$, then there must be (exactly) one entry in $\boldsymbol{s}$ outside $[-\theta, \theta]$.

These observations imply that the decoder $\mathcal{D}$ satisfies (D1)–(D2) for $(\tau, \sigma) = (0, 1)$ with respect to the threshold pair $(1, \Delta{=}2\theta)$. □

Noting that $n/r = n/(n-k) + 1 = k/(n-k) + 1$, by Theorem 1 we can rewrite (14) as

$$\mathsf{h}_1(\mathcal{C}) \le \left\lceil \frac{k}{n-k} \right\rceil \; .$$

It is still an open problem whether the simple construction of Proposition 4 attains, in general, the smallest possible value of $\mathsf{h}_1(\cdot)$, among all linear codes of the same length and dimension over $\mathbb{R}$. We can state the problem as follows.

Problem 1. *Identify the values of k and n for which every linear $[n, k]$ code $\mathcal{C}$ over $\mathbb{R}$ satisfies:*

$$\mathsf{h}_1(\mathcal{C}) \ge \left\lceil \frac{k}{n-k} \right\rceil \; .$$

The inequality obviously holds when $k \le n/2$, and the next proposition states it for $k = n - 1$.

Proposition 5. *Let $\mathcal{C}$ be a linear $[n, n{-}1]$ code over $\mathbb{R}$. Then*

$$\mathsf{h}_1(\mathcal{C}) \ge n - 1 \; .$$

Proof. Let $H = (h_0 \; h_1 \; \ldots \; h_{n-1})$ be a $1 \times n$ parity-check matrix of $\mathcal{C}$. If H contains a zero entry then $\mathsf{d}(\mathcal{C}) = 1$, in which case $\mathsf{h}_1(\mathcal{C}) = \infty$. Otherwise, suppose without loss of generality that

$$1 = h_0 \le |h_1| \le |h_2| \le \ldots \le |h_{n-1}|$$

and define the vector $\boldsymbol{c} = (c_j)_{j \in [n\rangle}$ by

$$c_j = \begin{cases} n-1 & \text{if } j = 0 \\ -1/h_j & \text{otherwise} \end{cases} \; .$$

Then $\boldsymbol{c} \in \mathcal{C}$ since $H\boldsymbol{c}^\top = 0$ and, so,

$$\mathsf{h}_1(\mathcal{C}) \ge \mathsf{h}_1(\boldsymbol{c}) = |c_0|/|c_{n-1}| = (n-1) \cdot |h_{n-1}| \ge n - 1 \; .$$

□

We are unaware of any study of Problem 1 even for the case $k = n - 2$ and, in particular, whether the answer to the following question is known.

Problem 2. *Must every $(n{-}2)$-dimensional subspace of $\mathbb{R}^n$, n even, contain a nonzero vector in which the ratio between the largest and second largest absolute values of its entries is at least $(n/2) - 1$?*

C. Single error correction

In this section, we present a construction for $w = 2\tau + \sigma = 2$. This scheme can therefore correct a single error (i.e., $\tau = 1$ and $\sigma = 0$); alternatively, it can detect two errors (i.e., $\tau = 0$ and $\sigma = 2$).

Given positive integers r and n such that $r \le n \le r(r{-}1)$, let H be a real $r \times n$ matrix over $\{-1, 0, 1\}$ that satisfies the following properties:

(H1) all columns of H are distinct,

(H2) each column in H contains exactly two nonzero entries, the first of which being a 1, and—

(H3) the number of nonzero entries in each row is either $\lfloor 2n/r \rfloor$ or $\lceil 2n/r \rceil$.

Clearly, the condition $n \le r(r{-}1)$ is necessary for having such a matrix. Conversely, when r is even, such an H can be constructed for every $n \le r(r{-}1)$: see [14] and references therein. E.g., for $r = 4$ and $n = 12$:

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 \end{pmatrix} .$$

We assume hereafter that r is even.

For easy encoding, we would prefer H to be systematic, yet requirement (H2) disallows it (unless $\mathrm{rank}(H) < r$). Instead, we can construct H so that its last r columns are as in the above example, namely, they form the $r \times r$ matrix

$$T = I_{r/2} \otimes \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} ,$$

where $\otimes$ stands for Kronecker product and $I_{r/2}$ is the identity matrix of order $r/2$. It is easy to see that $T^2 = 2\, I_r$ (and, in particular, $\mathrm{rank}(H) = r$ in this case). Hence, $(1/2)\, T \cdot H$ is a systematic matrix, and for every even $m \in [r\rangle$, its mth and $(m{+}1)$st rows are obtained by summing and subtracting the respective rows in $(1/2)\, H$.

Proposition 6. *Let $\mathcal{C}$ be a linear $[n, k{\ge}n{-}r]$ code over $\mathbb{R}$ with an $r \times n$ parity-check matrix H that satisfies (H1)–(H3). Then*

$$\Gamma_2(\mathcal{C}) \le 2 \cdot \lceil 2n/r \rceil \; .$$

Proof. Write $\theta = \lceil 2n/r \rceil$ and $H = (\boldsymbol{h}_j)_{j \in [n\rangle}$, where $\boldsymbol{h}_j = (h_{m,j})_{m \in [r\rangle}$ denotes column j in H, and consider a decoder $\mathcal{D}$ that, for a received vector $\boldsymbol{y} \in \mathbb{R}^n$, computes the syndrome

$\boldsymbol{s} = (s_m)_{m\in[r\rangle} = H\boldsymbol{y}^\top$ and returns a set containing one index t if the following three conditions hold:

(U1) $s_m \notin [-\theta, \theta]$ for exactly two values of m, say m_0 and $m_1 > m_0$,
(U2) $h_{m_0,t} = 1$, and—
(U3) $h_{m_1,t} = \mathrm{sgn}(s_{m_0} \cdot s_{m_1})$.

Otherwise $\mathcal{D}$ returns the empty set.

Assuming that $\boldsymbol{\varepsilon} \in \mathcal{Q}(n, 1)$ and that $\boldsymbol{e} = (e_j)_{j\in[n\rangle} \in \mathcal{B}(n, 1)$, the following observations are easily verified.

- If $\boldsymbol{e} = \mathbf{0}$ then $\boldsymbol{s} = H\boldsymbol{\varepsilon}^\top \in \mathcal{Q}(r, \theta)$.
- The syndrome $\boldsymbol{s}$ can have at most two nonzero entries outside $[-\theta, \theta]$.
- If (exactly) two entries in $\boldsymbol{s}$ are outside $[-\theta, \theta]$, then their positions, say m_0 and m_1 ($> m_0$), uniquely determine by (U2)–(U3) an index t for which $e_t \neq 0$.
- If $|e_t| > 2\theta$ for some $t \in [n\rangle$, then there must be two entries in $\boldsymbol{s}$ outside $[-\theta, \theta]$.

These observations imply that the decoder $\mathcal{D}$ satisfies (D1)–(D2) for $(\tau, \sigma) = (1, 0)$ with respect to the threshold pair $(1, \Delta{=}2\theta)$. ☐

Once an index t is found according to (U1)–(U3), we can compute lower and upper bounds on the outlying error value $e = e_t$ by solving the two linear programming problems in (10), with $\mathcal{J}^*$ therein taken as $\mathcal{J} = \{t\}$. Alternatively, we can solve the following two linear programming problems in the real variables e and $\boldsymbol{\varepsilon} = (\varepsilon_j)_{j\in[n\rangle}$:

$$\left.\begin{array}{c}\text{\textit{Minimize (respectively, maximize) e, subject to:}}\\ \begin{array}{rcr} e\cdot \boldsymbol{h}_t + H\boldsymbol{\varepsilon}^\top &=& \boldsymbol{s}\\ \boldsymbol{\varepsilon} &\le& \mathbf{1}\\ \boldsymbol{\varepsilon} &\ge& -\mathbf{1}\end{array}\end{array}\right\}\qquad(15)$$

where we are assuming that $\boldsymbol{y}$ (and therefore the respective syndrome $\boldsymbol{s} = H\boldsymbol{y}^\top$) is normalized so that $\delta = 1$. Since the solution of the linear programming problem yields feasible values for both $\boldsymbol{\varepsilon}$ and $\boldsymbol{e}$, one can actually decode the read vector $\boldsymbol{y}$ into a (candidate) codeword $\hat{\boldsymbol{c}} = \boldsymbol{y} - \boldsymbol{\varepsilon} - \boldsymbol{e}$ of $\mathcal{C}$ that is consistent with $\boldsymbol{y}$ and the error model.

Using the notation in the proof of Proposition 6, simpler (yet generally cruder) bounds are given by:

$$\max_{i\in[2\rangle}\left\{s_{m_i}\cdot h_{m_i,t} - \|H_{m_i}\|^2\right\} \le e \le \min_{i\in[2\rangle}\left\{s_{m_i}\cdot h_{m_i,t} + \|H_{m_i}\|^2\right\},$$

where H_m denotes row m in H and $\|\cdot\|$ stands for the L_2-norm (and we assume that $\delta = 1$). In particular, the correct value of e is determined up to a precision of $\pm\theta$.

Note that if there is only *one* entry in $\boldsymbol{s}$ outside the interval $[-\theta, \theta]$ we can tell that $\boldsymbol{e} \neq \mathbf{0}$, yet generally there is insufficient information to locate it.

While Proposition 6 yields an upper bound, $\Delta = 2\cdot\lceil 2n/r\rceil$, on $\Gamma_2(\mathcal{C})$ in terms of n and r, it may be more practical to express the redundancy r as a function of k (namely, the prescribed dimension of the code) and Δ. We do this in the next proposition.

Proposition 7. *Given a positive integer k and a real value $\Delta \ge 6$, let r be the smallest <u>even</u> integer not smaller than*

$$\max\left\{\frac{2k}{\lfloor\Delta/2\rfloor - 2}, \sqrt{k{+}1} + 1\right\}.$$

Let $\mathcal{C}$ be as in Proposition 6, where $n = r + k$. There is a decoder for $\mathcal{C}$ that satisfies (D1)–(D2) for $(\tau, \sigma) = (1, 0)$ with respect to the threshold pair $(1, \Delta)$.

Proof. First, since $r \ge \sqrt{k{+}1} + 1$ we have

$$\begin{aligned}(r-1)^2 \ge k+1 \quad&\Longrightarrow\quad r(r-1) \ge k + r\\ &\Longrightarrow\quad r(r-1) \ge n,\end{aligned}$$

namely, the code $\mathcal{C}$ indeed exists.

Secondly, since $r \ge 2k/(\lfloor\Delta/2\rfloor - 2) > 0$, we have

$$\begin{aligned}\left\lfloor\frac{\Delta}{2}\right\rfloor - 2 \ge \frac{2k}{r} \quad&\Longrightarrow\quad \left\lfloor\frac{\Delta}{2}\right\rfloor \ge \frac{2n}{r}\\ &\Longrightarrow\quad \left\lfloor\frac{\Delta}{2}\right\rfloor \ge \left\lceil\frac{2n}{r}\right\rceil\\ &\Longrightarrow\quad \Delta \ge 2\cdot\left\lceil\frac{2n}{r}\right\rceil.\end{aligned}$$

The result follows from Proposition 6. ☐

Remark 5. For $\Delta = 6$ and $k > 1$, the value of r in Proposition 7 is $2k$, corresponding to a linear $[n{=}3k, k]$ code. This code has therefore the same length and dimension as the k-fold Cartesian power of the $[3, 1]$ repetition code. However, by Theorem 3 it follows that for the latter code we have $\Gamma_2(\cdot) = 4$. Moreover, for $k = 1$, the repetition code yields a redundancy of 2, while Proposition 7 requires that r be greater than 2 for this dimension. Hence, there is a parameter range for which the construction of this section is not optimal. ☐

Comparing the construction in this section with the one in Section III-B, we see that for given n and r, we can handle there ratios Δ/δ that are twice as small, at the price of only detecting an outlying error rather than locating it. In addition, in the construction of Section III-B we can have an arbitrarily large n for any given r, while in this section we have the limitation $n \le r(r-1)$.

The above limitation can be relaxed, at the expense of increasing the value $\Gamma_2(\cdot)$, in two (not necessarily exclusive) ways. We can allow the entries of H to be taken from $\{0, \pm1, \ldots, \pm q\}$, for some positive integer q, or we can set the weight of each column to some prescribed integer $b \ge 2$. It turns out that, in general, we will get better results if we adopt only the latter strategy, namely, increasing b is advantageous over increasing q. Doing so, we can get a linear $[n, {\ge}n{-}r]$ code $\mathcal{C}$ with

$$n = \binom{r}{b}\cdot 2^{b-1}$$

and

$$\Gamma_2(\mathcal{C}) \le 2\left\lceil\frac{b\cdot n}{r}\right\rceil.$$

The proof of the latter inequality follows from a straightforward generalization of the proof of Proposition 6, where the value of θ is now taken to be $\lceil b\cdot n/r\rceil$, and conditions (U1)–(U3) are modified so that the syndrome $\boldsymbol{s}$ needs to have exactly b entries outside the range $[-\theta, \theta]$ in order to locate an outlying error.

For fixed b, this construction allows us to attain a redundancy of $r = O(n^{1/b})$ while still having $\Gamma_2(\mathcal{C}) = O(n/r)$. However, if we wish the redundancy to scale logarithmically with n, then b needs to scale linearly with r, in which case $\Gamma_2(\mathcal{C}) = \Omega(n)$ (see Section VI below).

IV. Alternative framework for single error handling

In this section, we generalize the discussion in Sections III-B and III-C to any linear code $\mathcal{C}$ and present a characterization of $\Gamma_1(\mathcal{C})$ and $\Gamma_2(\mathcal{C})$ through the parity-check matrix of $\mathcal{C}$. We will handle $\Gamma_2(\mathcal{C})$ first, and $\Gamma_1(\mathcal{C})$ will become a special (degenerate) case of our analysis. Hereafter, the notation $\mathcal{U}_j(n)$ will stand for the set of nonzero vectors in $\mathcal{B}(n, 1)$ whose nonzero entry is at position j.

Let $\mathcal{C}$ be a linear $[n, k]$ code $\mathcal{C}$ over $\mathbb{R}$ and let $H = (\boldsymbol{h}_0 \ \boldsymbol{h}_1 \ \ldots \ \boldsymbol{h}_{n-1})$ be an $(n-k) \times n$ parity-check matrix of $\mathcal{C}$ over $\mathbb{R}$. Recall that the syndrome $\boldsymbol{s}$ (with respect to H) of any read vector $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{\varepsilon} + \boldsymbol{e}$ equals

$$(H\boldsymbol{y}^\top =) \ \ \boldsymbol{s} = H\boldsymbol{\varepsilon}^\top + H\boldsymbol{e}^\top \ . \tag{16}$$

The decoding problem can be cast as locating the outlying errors in $\boldsymbol{e}$ given $\boldsymbol{s}$ and the containment $\boldsymbol{\varepsilon} \in \mathcal{Q}(n, \delta)$.

Remark 6. This formulation of the decoding problem bears similarity to the setting in compressed sensing with noise (see, for example, [1], [7], [9], [10], [20]), yet the scenarios are still different. First, in sparse recovery and compressed sensing, the vector $\boldsymbol{\varepsilon}$ in (16) is not multiplied by H, namely, it is in $\mathbb{R}^{n-k}$ rather than in $\mathbb{R}^n$.[6] Secondly, it is the L_2-norm of $\boldsymbol{\varepsilon}$, rather than its L_∞-norm, which is assumed to be bounded from above by δ. Thirdly, the requirements from the estimate for $\boldsymbol{e}$ that is returned by the decoder are not as strong as condition (D2): generally, the returned vector is required only to be sufficiently close to $\boldsymbol{e}$ under the L_2-metric.[7] □

Let

$$\mathcal{S} = \mathcal{S}_H = \left\{ H\boldsymbol{\varepsilon}^\top \ : \ \boldsymbol{\varepsilon} \in \mathcal{Q}(n, 1) \right\}$$

be the set of all syndrome vectors (with respect to H) that can be obtained if there are no outlying errors (and assuming that $\delta = 1$). It can be readily verified that $\mathcal{S}$ is a closed convex polytope in $\mathbb{R}^{n-k}$. In the presence of an outlying error vector $\boldsymbol{e}$, the set of syndrome values that can be obtained is

$$H\boldsymbol{e}^\top + \mathcal{S} \ ,$$

which is the translation of $\mathcal{S}$ by the vector $H\boldsymbol{e}^\top$. When $\boldsymbol{e} \in \mathcal{U}_t(n)$ this translation can be written as

$$e_t \cdot \boldsymbol{h}_t + \mathcal{S} \ .$$

We also define the set $2\mathcal{S}$ by

$$2\mathcal{S} = \mathcal{S} + \mathcal{S} = \left\{ H(\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}')^\top \ : \ \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}' \in \mathcal{Q}(n, 1) \right\} \ .$$

[6] In [1], the vector $\boldsymbol{\varepsilon}$ can also be multiplied by H, yet the model in that paper is probabilistic, where the entries of $\boldsymbol{\varepsilon}$ are uncorrelated, with a zero mean and a given variance.

[7] One exception is [9], where the authors also identify a region where (a counterpart of) the second containment in (2) holds.

It is easily seen that

$$2\mathcal{S} = \left\{ H\boldsymbol{\varepsilon}^\top \ : \ \boldsymbol{\varepsilon} \in \mathcal{Q}(n, 2) \right\} \ .$$

Proposition 8. *Given a linear $[n, k, d{\geq}3]$ code $\mathcal{C}$ over $\mathbb{R}$, let $H = (\boldsymbol{h}_j)_{j \in [n\rangle}$ be any $(n-k) \times n$ parity-check matrix of $\mathcal{C}$ and write $\mathcal{S} = \mathcal{S}_H$. Then $\Gamma_2(\mathcal{C})$ equals the smallest $\Delta \in \mathbb{R}^+$ such that for every distinct $j, j' \in [n\rangle$ and every pair $(e, e') \in \mathbb{R}^2$ such that $|e| > \Delta$, the translations*

$$e \cdot \boldsymbol{h}_j + \mathcal{S} \quad \textit{and} \quad e' \cdot \boldsymbol{h}_{j'} + \mathcal{S} \tag{17}$$

are disjoint; equivalently,

$$e' \cdot \boldsymbol{h}_{j'} \notin e \cdot \boldsymbol{h}_j + 2\mathcal{S} \ . \tag{18}$$

We first present a geometric interpretation of Proposition 8 and then proceed to its proof. For any $j \in [n\rangle$, the set

$$\{ e \cdot \boldsymbol{h}_j + \mathcal{S} \ : \ e \in \mathbb{R} \} \tag{19}$$

forms an infinite hyper-prism in $\mathbb{R}^{n-k}$, obtained by "sliding" the polytope $\mathcal{S}$ along a line in $\mathbb{R}^{n-k}$ that passes through the origin at the direction of the vector $\boldsymbol{h}_j$. The value Δ should be selected so that the subset

$$\{ e \cdot \boldsymbol{h}_j + \mathcal{S} \ : \ e \notin [-\Delta, \Delta] \}$$

of the jth hyper-prism does not intersect any of the other hyper-prisms, for any $j \in [n\rangle$. The decoder commits to an index $t = j$ if the syndrome $\boldsymbol{s} = H\boldsymbol{y}^\top$ belongs to that subset.

Figure 3(a) depicts the set $\mathcal{S}$ for a particular $2 \times n$ parity-check matrix of a linear $[n, k{=}n{-}2, d{=}3]$ code $\mathcal{C}$ that will be presented in Section V. In this example, the possible syndrome vectors $\boldsymbol{s} = (s_0 \ s_1)^\top$ form the plane $\mathbb{R}^2$, and $\mathcal{S}$ is a regular $2n$-polygon in $\mathbb{R}^2$. The pairs of parallel lines—one marked by solid lines and two by dashed lines—are (the infinite hyper-prisms that in two dimensions become) the infinite rectangles (19), for $j \in \{t{-}1, t, t{+}1\}$. A copy of $\mathcal{S}$ that is translated by $\Delta \cdot \boldsymbol{h}_t$ is drawn in between the two solid parallel lines; when Δ is the smallest for which (17) holds (for every $|e| > \Delta$ and $e' \in \mathbb{R}$), that copy of $\mathcal{S}$ touches (with a zero intersection area) the two adjacent infinite rectangles delimited by the two pairs of dashed parallel lines. Hence, if $\boldsymbol{e} \in \mathcal{U}_t(n)$, then the sets in (17) are disjoint for $j = t$, $j' = t \pm 1$, $|e| > \Delta$, and any $e' \in \mathbb{R}$.

Figure 3(b) shows an alternative geometric interpretation of Proposition 8. A translated copy of $2\mathcal{S}$ is centered in Figure 3(b) at the point $\Delta \cdot \boldsymbol{h}_t$. The dashed lines in Figure 3(b) represent lines through the origin at the directions of $\boldsymbol{h}_{t-1}$ and $\boldsymbol{h}_{t+1}$; when Δ is the smallest for which (18) holds (for every $|e| > \Delta$ and $e' \in \mathbb{R}$), these lines only touch the set $\Delta \cdot \boldsymbol{h}_t + 2\mathcal{S}$.

Proof of Proposition 8. Denote by Δ^* the smallest Δ that satisfies the conditions of the proposition (since $\mathcal{S}$ is a closed region in $\mathbb{R}^{n-k}$, the value Δ^* is well defined). We show that there exists a decoder for $\mathcal{C}$ that satisfies conditions (D1)–(D2) (for $(\tau, \sigma) = (1, 0)$ with respect to the threshold pair $(1, \Delta)$), if and only if $\Delta \geq \Delta^*$. The result will then follow from Theorem 1.

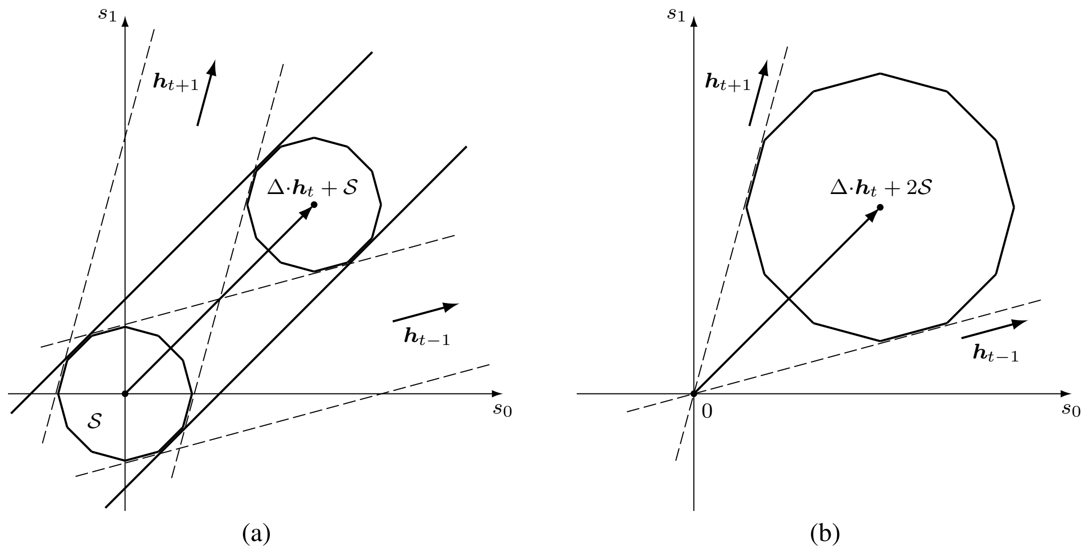

Fig. 3. Geometric interpretation of Proposition 8.

Suppose that $\Delta \geq \Delta^*$, and consider a decoder $\mathcal{D}$ which computes the syndrome $\boldsymbol{s} = H\boldsymbol{y}^\top$ and returns a set containing one index $t \in [n\rangle$ if there exists an $e \in \mathbb{R} \setminus [-\Delta, \Delta]$ such that

$$\boldsymbol{s} \in e \cdot \boldsymbol{h}_t + \mathcal{S} \tag{20}$$

(by the definition of Δ^*, there can be at most one index t for which (20) holds for some $e \notin [-\Delta, \Delta]$); otherwise $\mathcal{D}$ returns the empty set. We show that such a decoder satisfies (D1)–(D2) (for $(\tau, \sigma) = (1, 0)$ with respect to the threshold pair $(1, \Delta)$).

If $\boldsymbol{e} = \boldsymbol{0}$ then $\boldsymbol{s} \in \mathcal{S}$ and, so, $\boldsymbol{s} \notin e \cdot \boldsymbol{h}_j + \mathcal{S}$ for any $j \in [n\rangle$ and $|e| > \Delta$. Hence no index will be returned in this case.

Suppose now that $\boldsymbol{e} \in \mathcal{U}_t(n)$ for some $t \in [n\rangle$. Clearly, $\boldsymbol{s} \in e_t \cdot \boldsymbol{h}_t + \mathcal{S}$. If $|e_t| > \Delta$, then condition (20) holds for $j = t$ and $e = e_t$ and, therefore, $\mathcal{D}$ will return the (correct) index t. Otherwise, it follows from the conditions of the proposition that $\boldsymbol{s} \notin e \cdot \boldsymbol{h}_j + \mathcal{S}$ for any $j \neq t$ and $e \notin [-\Delta, \Delta]$. Hence, if an index j is returned by $\mathcal{D}$, it must be t.

Next, we show that when $\Delta < \Delta^*$, there is no decoder that satisfies (D1)–(D2). By the definition of Δ^* (and the fact that $\mathcal{S}$ is closed in $\mathbb{R}^{n-k}$) it follows that there exist distinct $t, t' \in [n\rangle$ and a pair $(e, e') \in \mathbb{R}^2$ such that $|e| = \Delta^*$ and

$$(e \cdot \boldsymbol{h}_t + \mathcal{S}) \cap (e' \cdot \boldsymbol{h}_{t'} + \mathcal{S}) \neq \emptyset .$$

This means that there exist $\boldsymbol{e} = (e_j)_{j\in[n\rangle} \in \mathcal{U}_t(n)$ and $\boldsymbol{e}' = (e'_j)_{j\in[n\rangle} \in \mathcal{U}_{t'}(n) \cup \{\boldsymbol{0}\}$ such that $|e_t| = \Delta^*$ and

$$H(\boldsymbol{\varepsilon} + \boldsymbol{e})^\top = H(\boldsymbol{\varepsilon}' + \boldsymbol{e}')^\top$$

for some $\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}' \in \mathcal{Q}(n, 1)$; we can then write

$$\boldsymbol{y} = \boldsymbol{\varepsilon} + \boldsymbol{e} = \boldsymbol{c}' + \boldsymbol{\varepsilon}' + \boldsymbol{e}' ,$$

for some $\boldsymbol{c}' \in \mathcal{C}$. Now, if there were a decoder that satisfies (D1)–(D2) for $\Delta < \Delta^*$ then, by the first containment in (2), such a decoder should return the index t for the input $\boldsymbol{y}$ (since $e_t = \Delta^* > \Delta$), yet by the second containment in (2) it should not. □

Remark 7. The proof of Proposition 8 suggests a (conceptual yet still concrete) decoder $\mathcal{D}$ for any given linear $[n, k, d{\geq}3]$ code $\mathcal{C}$ over $\mathbb{R}$ for locating a single error with respect to the threshold pair $(1, \Gamma_2(\mathcal{C}))$. Finding whether there exists an $e \in \mathbb{R} \setminus [-\Delta, \Delta]$ such that (20) holds is in effect a linear programming problem, for any examined $t \in [n\rangle$. □

Once the position t of an outlying error has been determined, finding lower and upper bounds on e can be cast as two linear programming problems as in (15).

Next, we turn to a characterization of $\Gamma_1(\mathcal{C})$ in terms of the parity-check matrix of $\mathcal{C}$.

Proposition 9. *Given a linear $[n, k, d{\geq}2]$ code $\mathcal{C}$ over $\mathbb{R}$, let $H = (\boldsymbol{h}_j)_{j\in[n\rangle}$ be any $(n-k) \times n$ parity-check matrix of $\mathcal{C}$ and write $\mathcal{S} = \mathcal{S}_H$. Then $\Gamma_1(\mathcal{C})$ equals the smallest $\Delta \in \mathbb{R}^+$ such that for every $j \in [n\rangle$ and every $e \in \mathbb{R}$ such that $|e| > \Delta$, the translations*

$$e \cdot \boldsymbol{h}_j + \mathcal{S} \quad \textit{and} \quad \mathcal{S} \tag{21}$$

are disjoint; equivalently,

$$e \cdot \boldsymbol{h}_t \notin 2\mathcal{S} .$$

Proof sketch. The proof is in effect a trimmed-down version of the proof of Proposition 8. For the "if" part, we consider a decoder $\mathcal{D}$ which computes the syndrome $\boldsymbol{s}$ and returns an empty set if $\boldsymbol{s} \in \mathcal{S}$; otherwise $\mathcal{D}$ returns "e". In the "only if" part, take $\boldsymbol{e}'$ to be the all-zero vector. □

Figures 4(a)–(b) present geometric interpretations of Proposition 9 (which are counterparts of Figures 3(a)–(b)). Specifically, Figure 4(a) shows two copies of $\mathcal{S}$, where one copy is translated by $\Delta \cdot \boldsymbol{h}_t$, for the smallest Δ for which the sets (21) are disjoint whenever $|e| > \Delta$. Figure 4(b) shows an equivalent characterization of this Δ: a translation of $2\mathcal{S}$ by $\Delta \cdot \boldsymbol{h}_t$ brings a face (edge) of $2\mathcal{S}$ to contain the origin (equivalently, the point $\Delta \cdot \boldsymbol{h}_t$ is on a face of $2\mathcal{S}$).

We end this section by presenting a geometric formulation of (a somewhat weaker version of) Problem 2, in view of Proposition 9.

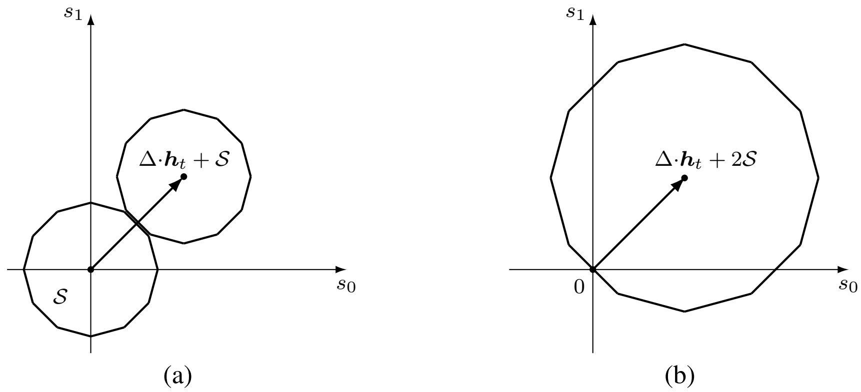

Fig. 4. Geometric interpretation of Proposition 9.

Let n be a positive even integer and let $\mathcal{P}$ be a $2n$-polygon in $\mathbb{R}^2$ with vertices $\boldsymbol{v}_m$, $m \in [2n\rangle$, and edges connecting $\boldsymbol{v}_m$ with $\boldsymbol{v}_{m+1}$ (where the index $2n$ is read as 0), such that:

(i) $\mathcal{P}$ is convex,
(ii) the length of each edge is two units, namely, $\mathcal{P}$ is equilateral (but not necessarily regular), and—
(iii) $\mathcal{P}$ has a 180° rotational symmetry, namely, $\boldsymbol{v}_{m+n} = -\boldsymbol{v}_m$, for all $m \in [n\rangle$.

For $m \in [n\rangle$, define $\boldsymbol{h}_m = (\boldsymbol{v}_{m+1} - \boldsymbol{v}_m)/2$ (namely, $\boldsymbol{h}_m$ is a vector of unit length that is parallel to the mth and $(n+m)$th edges of $\mathcal{P}$), and let e_m be the smallest nonnegative real such that the point $(1/2)\, e_m \cdot \boldsymbol{h}_m$ lies on some edge of $\mathcal{P}$; in other words, e_m is the diameter of $\mathcal{P}$ when measured along a line that passes through the origin and is parallel to the mth edge. For example, if $\mathcal{P}$ is the polygon $\mathcal{S}$ shown in Figure 4(a), then the vector that connects the centers of $\mathcal{S}$ and its translation $\Delta \cdot \boldsymbol{h}_t + \mathcal{S}$ in that figure forms one of the vectors $e_m \cdot \boldsymbol{h}_m$ (the midpoint between those centers lies on an edge of $\mathcal{P}$).

Let $\Delta(\mathcal{P}) = \max_{m\in[n\rangle} e_m$ be the maximum among those n directional diameters.

Problem 3. *What is the smallest attainable value of $\Delta(\mathcal{P})$, among all $2n$-polygons $\mathcal{P}$ that satisfy properties (i)–(iii)? Is this minimum attained by a rhombus (regarded as a degenerate $2n$-polygon), namely, does that minimum equal n?*

Relating now Problem 3 to Problem 2, the n vectors $\boldsymbol{h}_m$ are the columns of a parity-check matrix H of the $(n-2)$-dimensional linear space in the latter problem—which, in turn, is the code $\mathcal{C}$ in Problem 1—where we have added the assumption that each column in H has unit length. The (interior and boundary of the) polygon $\mathcal{P}$ forms then the set $\mathcal{S}_H$ and, so, by Proposition 9 it follows that $\Delta(\mathcal{P}) = \Gamma_1(\mathcal{C})$. A rhombus corresponds to the construction in Proposition 4, for the case $r = 2$.

There are quite a few isoperimetric extremal problems related to polygons that have been studied in the literature (and many of these problems are still open), yet none so far seems to be equivalent to Problem 3; see, for example, [2], [3], [6].

V. MDS CONSTRUCTION

Noting that $\Gamma_2(\mathcal{C}) < \infty$ only if $\mathsf{d}(\mathcal{C}) \ge 3$, by the Singleton bound it follows that $\Gamma_2(\mathcal{C}) < \infty$ only if the redundancy of $\mathcal{C}$ is at least 2. In this section, we present a construction whose redundancy is exactly 2 (namely, the code is MDS). For this construction, $\Gamma_2(\cdot)$ scales quadratically with n.

Given an integer $n > 2$, let $\alpha = \pi/n$ and let ω denote the complex primitive $2n$-th root of unity $e^{\imath\alpha}$, where $\imath = \sqrt{-1}$. Let $\mathcal{C}(n)$ be the linear $[n, n-2]$ code over $\mathbb{R}$ defined by

$$\mathcal{C}(n) = \left\{ (c_0\, c_1\, \ldots\, c_{n-1}) \in \mathbb{R}^n \;:\; \textstyle\sum_{j\in[n\rangle} c_j \omega^j = 0 \right\} .$$

Since $\omega^n = -1$, it follows that $\mathcal{C}(n)$ is a negacyclic code over $\mathbb{R}$ satisfying the closure:

$$(c_0\, c_1\, \ldots\, c_{n-2}\, c_{n-1}) \in \mathcal{C}(n) \implies (-c_{n-1}\, c_0\, c_1\, \ldots\, c_{n-2}) \in \mathcal{C}(n)$$

(see [4, §9.3]). The generator polynomial of $\mathcal{C}(n)$ is the minimal polynomial of ω with respect to $\mathbb{R}$:

$$g(x) = 1 - 2\cos(\alpha)\, x + x^2 .$$

For $j \in [n\rangle$, define

$$\boldsymbol{h}_j = \begin{pmatrix} h_{j,0} \\ h_{j,1} \end{pmatrix} = 2\sin(\alpha/2) \begin{pmatrix} \sin((j+1/2)\alpha) \\ -\cos((j+1/2)\alpha) \end{pmatrix} = \begin{pmatrix} \cos(j\alpha) - \cos((j+1)\alpha) \\ \sin(j\alpha) - \sin((j+1)\alpha) \end{pmatrix} . \quad (22)$$

It is easy to see that

$$\omega^j \cdot e^{\imath\alpha/2} \cdot 2\sin(\alpha/2) = (\imath \;\; -1) \cdot \boldsymbol{h}_j$$

and, therefore, $H = (\boldsymbol{h}_j)_{j\in[n\rangle}$ is a parity-check matrix of $\mathcal{C}(n)$. Hereafter, we compute all syndromes with respect to this parity-check matrix H.

Proposition 10. *The set $\mathcal{S} = \mathcal{S}_H$ is a regular $2n$-polygon in $\mathbb{R}^2$ whose set of vertices is given by*

$$\left\{ \boldsymbol{v}_m = 2 \cdot \begin{pmatrix} \cos(m\alpha) \\ \sin(m\alpha) \end{pmatrix} \;:\; m \in [2n\rangle \right\} . \quad (23)$$

Figure 3(a) depicts the set $\mathcal{S}$ for $n = 6$. Notice (from (22)) that

$$\boldsymbol{v}_{m+1} = \boldsymbol{v}_m - 2\boldsymbol{h}_m \quad \text{for } m \in [n\rangle \quad (24)$$

and that

$$\boldsymbol{v}_{m+n} = -\boldsymbol{v}_m \quad \text{for } m \in [n\rangle \ . \tag{25}$$

Proof of Proposition 10. For $m \in [n\rangle$, let $\boldsymbol{\varepsilon}_m = (\varepsilon_{m,j})_{j \in [n\rangle}$ be defined as follows:

$$\varepsilon_{m,j} = \begin{cases} -1 & \text{if } j < m \\ 1 & \text{if } j \geq m \end{cases} .$$

For $\boldsymbol{\varepsilon} = (\varepsilon_j)_{j \in [n\rangle} \in \mathcal{Q}(n,1)$, write

$$\begin{pmatrix} s_0(\boldsymbol{\varepsilon}) \\ s_1(\boldsymbol{\varepsilon}) \end{pmatrix} = H\boldsymbol{\varepsilon}^\top = \sum_{j \in [n\rangle} \varepsilon_j \boldsymbol{h}_j \ .$$

For $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_0$ we have

$$\begin{aligned} s_0(\boldsymbol{\varepsilon}_0) &= \sum_{j \in [n\rangle} (\cos(j\alpha) - \cos((j{+}1)\alpha)) \\ &= \cos 0 - \cos \pi = 2 \end{aligned}$$

and

$$\begin{aligned} s_1(\boldsymbol{\varepsilon}_0) &= \sum_{j \in [n\rangle} (\sin(j\alpha) - \sin((j{+}1)\alpha)) \\ &= \sin 0 - \sin \pi = 0 \ , \end{aligned}$$

i.e., $H\boldsymbol{\varepsilon}_0^\top = \boldsymbol{v}_0$. From (24) we obtain

$$\boldsymbol{v}_{m+1} - \boldsymbol{v}_m = -2\boldsymbol{h}_m = H(\boldsymbol{\varepsilon}_{m+1} - \boldsymbol{\varepsilon}_m)^\top \ , \quad m \in [n\rangle \ ,$$

and, therefore, by induction on m,

$$\boldsymbol{v}_m = H\boldsymbol{\varepsilon}_m^\top \ , \quad m \in [n\rangle \ .$$

By (22) it follows that $h_{j,0} > 0$ for all $j \in [n\rangle$, which means that $s_0(\boldsymbol{\varepsilon})$ is maximized when $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_0$; at that point $s_1(\boldsymbol{\varepsilon}_0) = 0$. For any other $\boldsymbol{\varepsilon} \in \mathcal{Q}(n,1)$ we get $s_0(\boldsymbol{\varepsilon}) < s_0(\boldsymbol{\varepsilon}_0) = 2$. As we allow $s_0(\boldsymbol{\varepsilon})$ to decrease, the largest value of $s_1(\boldsymbol{\varepsilon})$ will be attained by decreasing $\boldsymbol{\varepsilon}_0$ at the position $m \in [n\rangle$ for which the slope $h_{m,1}/h_{m,0} = \cot((m{+}1)/2)\alpha)$ is the smallest. Since $x \mapsto \cot(x)$ is decreasing for $x \in (0, \pi/2)$, this means selecting the smallest possible m, namely, $m = 0$. As $\boldsymbol{\varepsilon}$ changes from $\boldsymbol{\varepsilon}_0$ to $\boldsymbol{\varepsilon}_1$, the syndrome $H\boldsymbol{\varepsilon}^\top$ moves along the straight line segment that connects the points $\boldsymbol{v}_0 = H\boldsymbol{\varepsilon}_0^\top$ and $\boldsymbol{v}_1 = H\boldsymbol{\varepsilon}_1^\top$.

We can now repeat our arguments with $\boldsymbol{\varepsilon}$ equaling $\boldsymbol{\varepsilon}_1$. As the first entry in $\boldsymbol{\varepsilon}$ has hit the bottom value -1, the next position in $\boldsymbol{\varepsilon}$ to be decreased in order to maximize $s_1(\boldsymbol{\varepsilon})$ is the index m for which $h_{m,1}/h_{m,0} = \cot((m{+}1)/2)\alpha)$ is the second smallest, namely, $m = 1$. Thus, when $\boldsymbol{\varepsilon}$ ranges from $\boldsymbol{\varepsilon}_1$ to $\boldsymbol{\varepsilon}_2$, the respective syndrome $H\boldsymbol{\varepsilon}^\top$ moves along the line segment that connects $\boldsymbol{v}_1 = H\boldsymbol{\varepsilon}_1^\top$ and $\boldsymbol{v}_2 = H\boldsymbol{\varepsilon}_2^\top$.

We can continue this process inductively until reaching $m = n$, at which point $\boldsymbol{\varepsilon} = -\boldsymbol{\varepsilon}_0$ and

$$H\boldsymbol{\varepsilon}^\top = -\boldsymbol{v}_0 \stackrel{(25)}{=} \boldsymbol{v}_n \ .$$

The proof that $\boldsymbol{v}_{n+1}, \boldsymbol{v}_{n+2}, \ldots, \boldsymbol{v}_{2n-1}$ are the remaining vertices of $\mathcal{S}$ follows from similar arguments, except that we now seek to minimize (as opposed to maximize) $s_1(\boldsymbol{\varepsilon})$ for any given $s_0(\boldsymbol{\varepsilon})$. □

Proposition 11. *For the code $\mathcal{C}(n)$,*

$$\Gamma = \Gamma_2(\mathcal{C}(n)) = \frac{1}{\sin^2(\pi/(2n))} \ .$$

In particular, the ratio Γ/n^2 approaches the constant $4/\pi^2$ as $n \to \infty$.

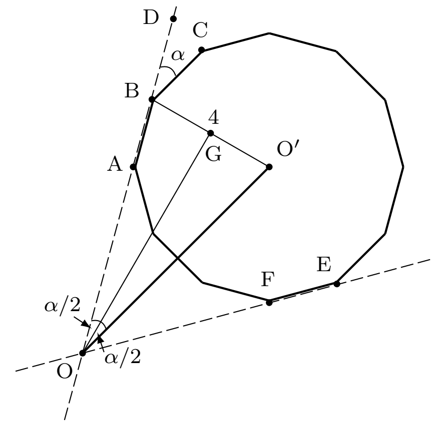

Fig. 5. Illustration for the proof of Proposition 11.

Proof. We refer to Figure 3(b), which depicts the set $\Delta \cdot \boldsymbol{h}_t + 2\mathcal{S}$. The value Δ is set to be (the smallest) so that the dashed lines, at the directions of $\boldsymbol{h}_{t-1}$ and $\boldsymbol{h}_{t+1}$, touch—but do not contain any interior points of—the set $\Delta \cdot \boldsymbol{h}_t + 2\mathcal{S}$. This value Δ equals Γ (due to the $2n$-fold rotational symmetry of $2\mathcal{S}$, the index t can be arbitrary).

We now determine Γ geometrically using Figure 5. The point O stands for the origin and O′ is the center of $\Gamma \cdot \boldsymbol{h}_t + \mathcal{S}$; the points A, B, C, E, and F are translations by $\Gamma \cdot \boldsymbol{h}_t$ of the vertices $2\boldsymbol{v}_{t+2}$, $2\boldsymbol{v}_{t+1}$, $2\boldsymbol{v}_t$, $2\boldsymbol{v}_{n+t}$ $(= -2\boldsymbol{v}_t)$, and $2\boldsymbol{v}_{n+t-1}$, respectively, of $2\mathcal{S}$. The points O, A, B, and D are colinear (with D being an arbitrary point so that the line segment $\overline{\mathrm{OB}}$ is strictly contained in $\overline{\mathrm{OD}}$), and so are the points O, F, and E. The line segment $\overline{\mathrm{BC}}$ is parallel to $\overline{\mathrm{OO'}}$ and is, therefore, at the direction of $\boldsymbol{h}_t$, and the segments $\overline{\mathrm{OB}}$ and $\overline{\mathrm{OE}}$ are at the directions of $\boldsymbol{h}_{t+1}$ and $\boldsymbol{h}_{t-1}$, respectively. Finally, the segment $\overline{\mathrm{OG}}$ bisects the angle $\angle\mathrm{BOO'}$.

We have

$$\left\| \overline{\mathrm{OO'}} \right\| = \Gamma \cdot \|\boldsymbol{h}_t\| \stackrel{(22)}{=} 2\,\Gamma \cdot \sin(\alpha/2) \ , \tag{26}$$

where $\|\cdot\|$ stands for the (Euclidean) length of a line segment or of a vector.

We also have

$$\angle\mathrm{BOO'} = \angle\mathrm{DBC} = \pi/n = \alpha \ , \tag{27}$$

since $\angle\mathrm{BOO'}$ and $\angle\mathrm{DBC}$ are corresponding angles (formed by the intersection of the transversal $\overline{\mathrm{OD}}$ with the parallel lines $\overline{\mathrm{BC}}$ and $\overline{\mathrm{OO'}}$), with $\angle\mathrm{DBC}$ also being an exterior angle of a regular $2n$-polygon. Hence,

$$\angle\mathrm{BOG} = \angle\mathrm{GOO'} = \alpha/2 \ . \tag{28}$$

Since $\angle\mathrm{OBO'}$ and $\angle\mathrm{CBO'}$ are equal and their sum is $\pi - \angle\mathrm{DBC}$, it also follows that $\angle\mathrm{OBO'} = (\pi - \alpha)/2$. By (27) we thus get that the third angle, $\angle\mathrm{BO'O}$, in $\triangle\mathrm{BOO'}$ equals $(\pi -$

$\alpha)/2$, which means that this triangle is an isosceles and, so, $\angle \mathrm{OGO}' = \pi/2$ and

$$\left\|\overline{\mathrm{GO}'}\right\| = \frac{1}{2} \cdot \left\|\overline{\mathrm{BO}'}\right\| = \|\boldsymbol{v}_{t+1}\| \stackrel{(23)}{=} 2 \; .$$

Hence,

$$\left\|\overline{\mathrm{OO}'}\right\| \stackrel{(28)}{=} \frac{\left\|\overline{\mathrm{GO}'}\right\|}{\sin(\alpha/2)} = \frac{2}{\sin(\alpha/2)} \; .$$

The result is now obtained by substituting the value of $\left\|\overline{\mathrm{OO}'}\right\|$ into (26). □

The following problem is yet to be settled.

Problem 4. *Does the construction $\mathcal{C}(n)$ have the smallest possible value of $\mathsf{h}_2(\cdot)$, among all linear $[n, n{-}2]$ codes over $\mathbb{R}$?*

As we pointed out in Remark 7, the proof of Proposition 8 suggests a single error locating decoder for any linear $[n, k, d{\geq}3]$ code. For the code $\mathcal{C}(n)$, such a decoder can be described geometrically through Figure 3(a). Given a received word $\boldsymbol{y} \in \mathbb{R}^n$, one first computes the syndrome $\boldsymbol{s} = H\boldsymbol{y}^\top$. Then, for each of the n pairs of parallel lines as (the instances) shown in Figure 3(a), one checks if $\boldsymbol{s}$ lies in the infinite rectangle formed by the pair. If it does so for more than one pair then the decoder returns an empty set of error locations (i.e., no outlying errors have been found). Otherwise, if $\boldsymbol{s}$ lies in exactly one infinite rectangle, the decoder returns the index t of the respective pair. Decoding failure is declared if $\boldsymbol{s}$ lies outside the area formed by the n infinite rectangles.

VI. Asymptotic bounds

The linear $[n, {\geq}n{-}r]$ code construction that we presented in Section III-C allows us to achieve $\Gamma_2(\cdot) = O(n/r)$, yet requires (the upper bound on) the redundancy r to satisfy $r(r{-}1) \geq n$, namely, the redundancy scales like $\sqrt{n}$. The construction in Section V, on the other hand, has a redundancy of 2, yet $\Gamma_2(\cdot)$ scales like n^2. It is yet to be determined how the smallest attainable $\Gamma_2(\cdot)$ depends on the redundancy, for any given code length. In this section, we show that there exist arbitrarily long linear $[n, n{-}r]$ codes $\mathcal{C}$ over $\mathbb{R}$ with redundancy r that scales logarithmically with n, while $\Gamma_2(\mathcal{C}) = O\left(n/\sqrt{r}\right)$; namely, $\Gamma_2(\mathcal{C})$ is sub-linear in n.[8]

Such a code $\mathcal{C}$ will be defined through a linear $[r, \kappa{>}2, d]$ code B over $\mathrm{GF}(2)$ that satisfies the following two properties:

(B1) B contains the all-one codeword, and—

(B2) $\mathsf{d}(B^\perp) > 2$.

Let $n = 2^{\kappa-1}$, and denote by B_0 the set of all n codewords of B_0 whose first entry equals 0 (B_0 consists of half of the codewords of B). Let $H = (\boldsymbol{h}_{\boldsymbol{x}})_{\boldsymbol{x} \in B_0}$ be an $r \times n$ matrix over $\mathbb{R}$ whose columns are indexed by the codewords of B_0, and for each $\boldsymbol{x} = (x_i)_{i \in [r\rangle} \in B_0$, the ith entry of $\boldsymbol{h}_{\boldsymbol{x}}$ equals $(-1)^{x_i}/\sqrt{r}$; hence, the entries of H are either $1/\sqrt{r}$ or $-1/\sqrt{r}$, and no column of H is a scalar multiple of the other. The code $\mathcal{C} = \mathcal{C}(B)$ is now defined as the linear $[n, k{\geq}n{-}r]$ code over $\mathbb{R}$ with the parity-check matrix H.

[8] We note, however, that this result is mainly of a theoretical nature, since it may require values of n that are much larger than those anticipated in practical use.

Our analysis of $\mathcal{C}(B)$ will be based on Proposition 8, except that we will over-estimate the set $\mathcal{S} = \mathcal{S}_H$ by its minimum bounding ball in $\mathbb{R}^r$. Since $\boldsymbol{s} \in \mathcal{S}$ if and only if $-\boldsymbol{s} \in \mathcal{S}$, it follows that the center of the minimum bounding ball of $\mathcal{S}$ is at the origin. The radius of that ball is given by the following lemma.

Lemma 12. *The radius ρ of the minimum bounding ball of $\mathcal{S}$ in $\mathbb{R}^r$ equals $n/\sqrt{r}$.*

Proof. Since $\mathsf{d}(B^\perp) > 2$, the restrictions of the codewords of B to any two distinct positions range over each pair in the set $\{00, 01, 10, 11\}$ exactly $2^{\kappa-2}$ times [16, p. 139]. Hence, any two distinct rows in H agree on exactly $2^{k-2} = n/2$ positions, which means that they are orthogonal:

$$H \cdot H^\top = \frac{n}{r} \cdot I \; .$$

The entries of the syndrome $\boldsymbol{s} = H\boldsymbol{\varepsilon}^\top$ of any $\boldsymbol{\varepsilon} \in \mathcal{Q}(n,1)$ are the projections of $\boldsymbol{\varepsilon}$ onto the rows of H (up to scaling by the L_2-norm, $\sqrt{n/r}$, of each row). Regarding the rows of H as a subset of an orthogonal basis of $\mathbb{R}^n$, we obtain:

$$\|\boldsymbol{s}\| \leq \sqrt{\frac{n}{r}} \cdot \|\boldsymbol{\varepsilon}\| \leq \frac{n}{\sqrt{r}} \; .$$

Equalities are attained when $\boldsymbol{\varepsilon}/\sqrt{r}$ is a row of H. □

We also use the following known lemma (see [8, p. 27]).

Lemma 13. *For any distinct $\boldsymbol{x}, \boldsymbol{x}' \in B_0$, the angle $\phi_{\boldsymbol{x},\boldsymbol{x}'}$ ($\in (0, \pi/2]$) between $\boldsymbol{h}_{\boldsymbol{x}}$ and $\boldsymbol{h}_{\boldsymbol{x}'}$ satisfies*

$$\left|\boldsymbol{h}_{\boldsymbol{x}}^\top \cdot \boldsymbol{h}_{\boldsymbol{x}'}\right| = \cos\phi_{\boldsymbol{x},\boldsymbol{x}'} \leq 1 - \frac{2d}{r}$$

and, so,

$$\sin\phi_{\boldsymbol{x},\boldsymbol{x}'} \geq 2\sqrt{\frac{d}{r}\left(1 - \frac{d}{r}\right)} \; .$$

Theorem 14. *For the code $\mathcal{C} = \mathcal{C}(B)$,*

$$\Gamma_2(\mathcal{C}) \leq \frac{n}{\sqrt{d(1 - (d/r))}} \; .$$

Proof. Given two distinct columns $\boldsymbol{h}_{\boldsymbol{x}}$ and $\boldsymbol{h}_{\boldsymbol{x}'}$ of H and some nonnegative real e, consider the projection of $e \cdot \boldsymbol{h}_{\boldsymbol{x}} + 2\mathcal{S}$ onto the plane that is spanned by $\boldsymbol{h}_{\boldsymbol{x}}$ and $\boldsymbol{h}_{\boldsymbol{x}'}$. Figure 6 depicts such a plane, where $\overline{\mathrm{OO}'}$ and $\overline{\mathrm{OA}}$ are at the direction of $\boldsymbol{h}_{\boldsymbol{x}}$ and $\boldsymbol{h}_{\boldsymbol{x}'}$, respectively, and the projection of $e \cdot \boldsymbol{h}_{\boldsymbol{x}} + 2\mathcal{S}$ onto the plane is depicted in the figure by the polygon and is contained in a (concentric) circle of radius 2ρ. In that figure, the center O' is selected so that the line OA is tangent to the circle at A (and, so, $\angle \mathrm{OAO}' = \pi/2$). Denoting $\Delta_{\boldsymbol{x},\boldsymbol{x}'} = \|\mathrm{OO}'\|$ and recalling that $\|\boldsymbol{h}_{\boldsymbol{x}}\| = 1$, we get that (18) holds for every $|e| > \Delta_{\boldsymbol{x},\boldsymbol{x}'}$ and $e' \in \mathbb{R}$, where we take $\boldsymbol{h}_j \equiv \boldsymbol{h}_{\boldsymbol{x}}$ and $\boldsymbol{h}_{j'} \equiv \boldsymbol{h}_{\boldsymbol{x}'}$. We see from Figure 6 that

$$\Delta_{\boldsymbol{x},\boldsymbol{x}'} = \|\mathrm{OO}'\| = \frac{2\rho}{\sin\phi_{\boldsymbol{x},\boldsymbol{x}'}} \leq \frac{n}{\sqrt{d(1 - (d/r))}} \; ,$$

where the inequality follows from Lemmas 12 and 13. The result is implied by:

$$\Gamma_2(\mathcal{C}) \leq \max_{\substack{\boldsymbol{x},\boldsymbol{x}' \in B_0 \, : \\ \boldsymbol{x} \neq \boldsymbol{x}'}} \Delta_{\boldsymbol{x},\boldsymbol{x}'} \; .$$

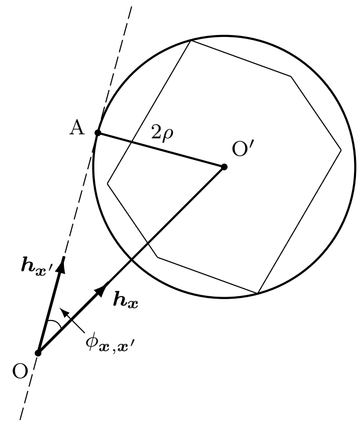

Fig. 6. Illustration for the proof of Theorem 14.

□

We now take B to be a member of a family of "good" linear codes over $\mathrm{GF}(2)$ for which both κ/r and d/r are bounded away from 0. Specifically, we take B to be a concatenated code with an outer Reed–Solomon code that contains the all-one codeword, and an inner code whose generator matrix is constructed by randomly selecting the rows except that the first row is taken to be the all-one codeword. Such a choice guarantees that B satisfies condition (B1). Now, with high probability, we will get an inner code that attains the Gilbert–Varshamov bound (this follows by a slight modification of the proof of Theorem 4.5 in [17]); moreover, with positive probability (which can be high if the inner code is taken to have rate less than $1/2$), the columns of the randomly selected generator matrix will be nonzero and distinct, thereby satisfying condition (B2) (see Problem 12.3(2) in [17]). For this construction, we get that $\Gamma_2(\mathcal{C}) = O\left(n/\sqrt{r}\right)$ while $r = O(\log n)$.

Acknowledgment

The author wishes to thank John Paul Strachan, Michael Elad, and Lutz Lampe for helpful discussions.

References

[1] E. Arias-Castro, Y.C. Eldar, "Noise folding in compressed sensing," *IEEE Signal Process. Lett.,* 18 (2011), 478–481.

[2] C. Audet, P. Hansen, F. Messine, "Extremal problems for convex polygons," *J. Glob. Optim.,* 38 (2007) 163–179.

[3] C. Audet, J. Ninin, "Maximal perimeter, diameter and area of equilateral unit-width convex polygons," *J. Glob. Optim.,* 56 (2013) 1007–1016.

[4] E.R. Berlekamp, *Algebraic Coding Theory,* Revised Edition, Aegean Park Press, Laguna Hills, California, 1984.

[5] B.E. Boser, E. Sackinger, J. Bromley, Y. Le Cun, L.D. Jackel, "An analog neural network processor with programmable topology," *IEEE J. Solid-State Circuits,* 26 (1991), 2017–2025.

[6] P. Brass, W. Moser, J. Pach, *Research Problems in Discrete Geometry,* Springer, New York, 2005.

[7] E.J. Candès, J.K. Romberg, T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.,* 49 (2006), 1207–1223.

[8] J.H. Conway, N.J.A. Sloane, *Sphere Packings, Lattices and Groups,* Springer, New York, 1988.

[9] D.L. Donoho, M. Elad, V.N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory,* 52 (2006), 6–18.

[10] J.J. Fuchs, "Recovery of exact sparse representations in the presence of bounded noise," *IEEE Trans. Inf. Theory,* 51 (2005), 3601–3608.

[11] M. Hu, J.P. Strachan, Z. Li, E.M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. Yang, R.S. Williams, "Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication," *Proc. 53rd Annual Design Automation Conference (DAC'16),* Austin, Texas (2016), Article No. 19.

[12] M. Haikin, R. Zamir, "Analog coding of a source with erasures," *Proc. IEEE Int'l Symp. Inf. Theory,* Barcelona, Spain (2016), 2074–2078.

[13] M. Hu, C.E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R.S. Williams, J.J. Yang, Q. Xia, J.P. Strachan, "Memristor-based analog computation and neural classification with a dot product engine," *Adv. Mater.,* 2018, 30, 1705914.

[14] G.O.H. Katona, Á. Seress, "Greedy construction of nearly regular graphs," *European J. Comb.,* 14 (1993), 213–229.

[15] F. Kub, K. Moon, I. Mack, F. Long, "Programmable analog vector–matrix multipliers," *IEEE J. Solid-State Circuits,* 25 (1990), 207–214.

[16] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland, Amsterdam, 1977.

[17] R.M. Roth, *Introduction to Coding Theory,* Cambridge University Press, Cambridge, UK, 2006.

[18] R.M. Roth, "Fault-tolerant dot-product engines," *IEEE Trans. Inf. Theory,* 65 (2019), 2046–2057.

[19] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J.P. Strachan, M. Hu, R.S. Williams, V. Srikumar, "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *Proc. 43rd ACM/IEEE Int'l Symp. on Computer Architecture (ISCA 2016),* Seoul, Korea (2016), pp. 14–26.

[20] J.A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inf. Theory,* 52 (2006), 1030–1051.

[21] J.K. Wolf, "Redundancy, the discrete Fourier transform, and impulse noise cancellation," *IEEE Trans. Commun.,* 31 (1983), 458–461.

Ron M. Roth (M'88–SM'97–F'03) received the B.Sc. degree in computer engineering, the M.Sc. in electrical engineering, and the D.Sc. in computer science from Technion—Israel Institute of Technology, Haifa, Israel, in 1980, 1984, and 1988, respectively. Since 1988 he has been with the Computer Science Department at Technion, where he now holds the General Yaakov Dori Chair in Engineering. During the academic years 1989–91 he was a Visiting Scientist at IBM Research Division, Almaden Research Center, San Jose, California, and during 1996–97, 2004–05, and 2011–2012 he was on sabbatical leave at Hewlett–Packard Laboratories, Palo Alto, California. He is the author of the book *Introduction to Coding Theory*, published by Cambridge University Press in 2006. Dr. Roth was an associate editor for coding theory in IEEE Transactions on Information Theory from 1998 till 2001, and he is now serving as an associate editor in *SIAM Journal on Discrete Mathematics*. His research interests include coding theory, information theory, and their application to the theory of complexity.