Corrections to "Analog Error-Correcting Codes"

Ron M. Roth, *Fellow, IEEE*

In the above article [1], Lemma 2 contained an error, which also affects the proof of Theorem 1 therein. Specifically, Theorem 1 holds only when $\sigma = 0$ or when $\tau = 0$; otherwise, the "if" part in the theorem requires a stronger condition.

Our fix includes a slight change in the requirements from a decoder (that corrects τ errors and detects σ additional errors), in forbidding false alarms only when the number of errors is at most τ. Specifically, conditions (D1) and (D2) in Section I in [1] should now read as follows.

(D1) If $\boldsymbol{e} \in \mathcal{B}(n,\tau)$ then "e$''$ $\neq \mathcal{D}(\boldsymbol{y}) \subseteq \mathsf{Supp}_0(\boldsymbol{e})$.

(D2) If $\mathcal{D}(\boldsymbol{y}) \neq$ "e$''$ then $\mathsf{Supp}_\Delta(\boldsymbol{e}) \subseteq \mathcal{D}(\boldsymbol{y})$.

Respectively, the wording of Lemma 2 should be changed as follows (the change is underlined).

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 $\boldsymbol{e} \in \mathcal{E}(\boldsymbol{y})$ and $\boldsymbol{e}' \in \mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n,\tau)$,

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

We include below the modified proofs of Lemma 2 and Theorem 1, as well as a reworked Example 1.

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

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

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

$$\boldsymbol{c}' - \boldsymbol{c} = \boldsymbol{\varepsilon} + \boldsymbol{e} - \boldsymbol{\varepsilon}' - \boldsymbol{e}' \; , \tag{1}$$

i.e., the right-hand side of (1) is a codeword of $\mathcal{C}$ and, as such, its m-height does not 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, so, from (1) we get the following inequality:

$$\varepsilon_j + e_j - \varepsilon'_j - \underbrace{e'_j}_{0} \leq 2\,\mathsf{h}_w(\mathcal{C}) \cdot \delta \; .$$

Thus,

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

thereby contradicting the assumption that $e_j > \Delta$. □

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,

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

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

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

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

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

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

- "e$''$,
- 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

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

Note that $\boldsymbol{\varepsilon}' \in \mathcal{Q}(n,\delta)$ and that $\boldsymbol{e}' \in \mathcal{B}(n,\tau)$; moreover, $e'_0 = 0$. Hence, upon receiving $\boldsymbol{y} = \boldsymbol{c}' + \boldsymbol{\varepsilon}' + \boldsymbol{e}'$, the decoder must return a set of locations that does not contain the index 0. Thus, 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 $\mathsf{w}(\boldsymbol{c}') =$

Manuscript received 27 November 2022; accepted 25 December 2022. Date of publication 4 January 2023; date of current version 19 May 2023. This work was supported by the Israel Science Foundation under Grant 1713/20.

The author is with the Computer Science Department, Technion, Haifa 3200003, Israel (e-mail: ronny@cs.technion.ac.il).

Communicated by S. Ghorpade, Associate Editor for Coding and Decoding.
Digital Object Identifier 10.1109/TIT.2023.3234208

0018-9448 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

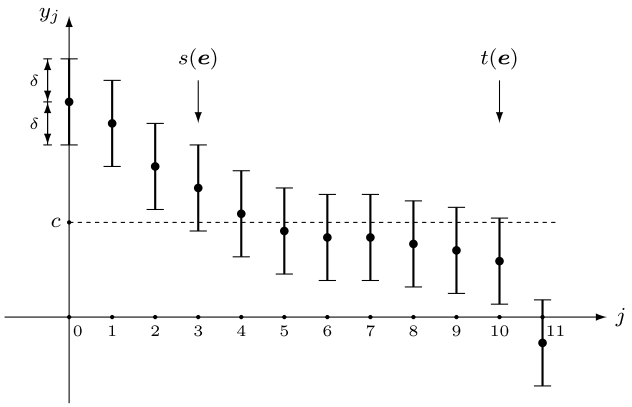

Fig. 1. Received vector $\boldsymbol{y}$ for Example 1.

$\mathsf{d}(\mathcal{C})$ and $c'_0 > \Delta$ (say); note that $c'_w = 0$ (and so (2) 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 every $\boldsymbol{y} \in \mathbb{R}^n$ such that $\mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n,\tau) \neq \emptyset$, define the following intersection:

$$\Lambda(\boldsymbol{y}) = \cap_{\boldsymbol{e}' \in \mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n,\tau)} \mathsf{Supp}_0(\boldsymbol{e}') \ .$$

The return value of $\mathcal{D}$ for every $\boldsymbol{y} \in \mathbb{R}^n$ is defined by

$$\mathcal{D}(\boldsymbol{y}) = \begin{cases} \Lambda(\boldsymbol{y}) & \text{if } \mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n,\tau) \neq \emptyset \\ \text{“e}'' & \text{otherwise} \end{cases} \ . \tag{3}$$

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 $\boldsymbol{e} \in \mathcal{B}(n,\tau+\sigma)$. 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″. Moreover, $\Lambda(\boldsymbol{y}) \subseteq \mathsf{Supp}_0(\boldsymbol{e})$ and, therefore, condition (D1) holds. As for condition (D2), 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 $\boldsymbol{e}' \in \mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n,\tau)$ 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 $\mathbf{1} \in \mathbb{R}^n$. We provide an explicit description of the decoder in (3) for $\mathcal{C}$, for $\tau \leq (n-1)/2$ and $\sigma = n-1-2\tau$.

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

$$y_0 \geq y_1 \geq \ldots \geq y_{n-1} \ . \tag{4}$$

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.

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

$$\bigcap_{j \in \overline{\mathsf{Supp}}_0(\boldsymbol{e})} \mathcal{I}_j \neq \emptyset \ , \tag{5}$$

where $\overline{\mathsf{Supp}}_0(\boldsymbol{e}) = [n\rangle \setminus \mathsf{Supp}_0(\boldsymbol{e})$. Indeed, each element c that belongs to the intersection (5) 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 $\overline{\mathsf{Supp}}_0(\boldsymbol{e})$; note that

$$t(\boldsymbol{e}) - s(\boldsymbol{e}) \geq n - \tau - 1 \ .$$

Input: $\boldsymbol{y} = (y_j)_{j\in[n\rangle} \in \mathbb{R}^n$ such that (4) 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 {
  $i \leftarrow \min\{j \in [\tau{+}1\rangle \ : \ y_j \leq y_{j+n-1-\tau} + 2\delta\}$;
  $i' \leftarrow \max\{j \in [n\rangle \setminus [n{-}1{-}\tau\rangle \ : \ y_{j-(n-1-\tau)} \leq y_j + 2\delta\}$;
  Return $[i\rangle \cup ([n\rangle \setminus [i'{+}1\rangle)$.
}

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

Due to the assumed ordering on the entries of $\boldsymbol{y}$, it follows that (5) is equivalent to

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

which, in turn, is equivalent to

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

E.g., Figure 1 was drawn for an error vector $\boldsymbol{e}$ for which $\mathsf{Supp}_0(\boldsymbol{e}) = \{0,1,2,11\}$, in which case $s(\boldsymbol{e}) = 3$ and $t(\boldsymbol{e}) = 10$.[1]

Conversely, any index $j \in [\tau{+}1\rangle$ such that

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

defines a set $\{j, j{+}1, \ldots, j{+}n{-}\tau{-}1\}$ which is contained in (and, even more so, equal to) $\overline{\mathsf{Supp}}_0(\boldsymbol{e})$ for some vector $\boldsymbol{e} \in \mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n,\tau)$.

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 $\Lambda(\boldsymbol{y})$. Specifically, among all error vectors in $\mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n,\tau)$, let $\boldsymbol{e}$ and $\boldsymbol{e}'$ be such that $s(\boldsymbol{e})$ is the smallest and $t(\boldsymbol{e}')$ is the largest; note that $i = s(\boldsymbol{e})$ and $i' = t(\boldsymbol{e}')$. Since $\tau \leq (n-1)/2$, the sets $\overline{\mathsf{Supp}}_0(\boldsymbol{e})$ and $\overline{\mathsf{Supp}}_0(\boldsymbol{e}')$ intersect and, so,

$$[n\rangle \setminus \Lambda(\boldsymbol{y}) = \bigcup_{\boldsymbol{e} \in \mathcal{E}(\boldsymbol{y}) \cap \mathcal{B}(n,\tau)} \overline{\mathsf{Supp}}_0(\boldsymbol{e}) = \{i, i{+}1, \ldots, i'\} \ .$$

For the vector $\boldsymbol{y}$ in Figure 1 and for $\tau = 4$, the returned set will be $\{0,1,11\}$, since $y_j > y_{j+7} + 2\delta$ for $j = 0,1$ yet $y_2 \leq y_9 + 2\delta$, and $y_4 > y_{11} + 2\delta$ yet $y_3 \leq y_{10} + 2\delta$. □

ACKNOWLEDGMENT

The author would like to thank Andrew Jiang who brought to his attention the error in the proof of Lemma 2.

REFERENCES

[1] R. M. Roth, “Analog error-correcting codes,” *IEEE Trans. Inf. Theory*, vol. 66, no. 7, pp. 4075–4088, Jul. 2020.

[1] However, Figure 1 is consistent with other error vectors as well. For example, we may slightly increase c so that it belongs to $\mathcal{I}_2 \cap \mathcal{I}_9$ (thus not to $\mathcal{I}_{10}$), thereby corresponding to an error vector with support $\{0,1,10,11\}$.