

Correction to “Analog Error-Correcting Codes”

Ron M. Roth

In the above paper [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 $e \in \mathcal{B}(n, \tau)$ then “e” $\neq \mathcal{D}(\mathbf{y}) \subseteq \text{Supp}_0(e)$.

(D2) If $\mathcal{D}(\mathbf{y}) \neq \text{“e”}$ then $\text{Supp}_\Delta(e) \subseteq \mathcal{D}(\mathbf{y})$.

Respectively, the wording of Lemma 2 should be changed as follows (red text marks the main changes).

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*

$$\Delta \geq 2(h_{2\tau+\sigma}(\mathcal{C}) + 1)\delta .$$

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

$$\text{Supp}_\Delta(e) \subseteq \text{Supp}_0(e') .$$

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

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

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

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

$$\mathbf{c}' - \mathbf{c} = \boldsymbol{\varepsilon} + e - \boldsymbol{\varepsilon}' - e' , \quad (1)$$

i.e., the right-hand side of (1) is a codeword of \mathcal{C} and, as such, its m -height does not exceed $h_m(\mathcal{C})$, for every $m \geq 0$.

Suppose to the contrary that $\text{Supp}_\Delta(e) \not\subseteq \text{Supp}_0(e')$. Then $e'_j = 0$ yet $|e_j| > \Delta$ for some position $j \in [n]$; without loss of generality we assume hereafter that $e_j > \Delta$. Writing $w = 2\tau + \sigma$, we have $w(e) + w(e') \leq w$ and, so, from (1) we get the following inequality:

$$\varepsilon_j + e_j - \varepsilon'_j - \underbrace{e'_j}_0 \leq 2h_w(\mathcal{C}) \cdot \delta .$$

Thus,

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

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

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 $\Delta < 2(h_w(\mathcal{C}) + 1)\delta$. We first consider the case where $h_w(\mathcal{C}) < \infty$, namely, $w = 2\tau + \sigma < d(\mathcal{C})$. Let $\mathbf{c}' = (c'_j)_{j \in [n]} \in \mathcal{C}$ be such that $h_w(\mathbf{c}') = h_w(\mathcal{C})$. Without loss of generality we assume that

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

and that

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

In either case,

$$c'_0 = h_w(\mathcal{C}) \cdot |c'_w| = 2h_w(\mathcal{C}) \cdot \delta . \quad (2)$$

Let

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

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

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

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

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

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

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

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

- “e”;
- a set of locations that contains the index 0.

Yet \mathbf{y} can also be written as

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

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

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

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

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

Note that $\boldsymbol{\varepsilon}' \in \mathcal{Q}(n, \delta)$ and that $e' \in \mathcal{B}(n, \tau)$; moreover, $e'_0 = 0$. Hence, upon receiving $\mathbf{y} = \mathbf{c}' + \boldsymbol{\varepsilon}' + e'$, 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 $h_w(\mathcal{C}) = \infty$ (i.e, $w \geq d(\mathcal{C})$) yet $\Delta < \infty$, our proof is still valid if we take \mathbf{c}' to be of Hamming weight $w(\mathbf{c}') =$

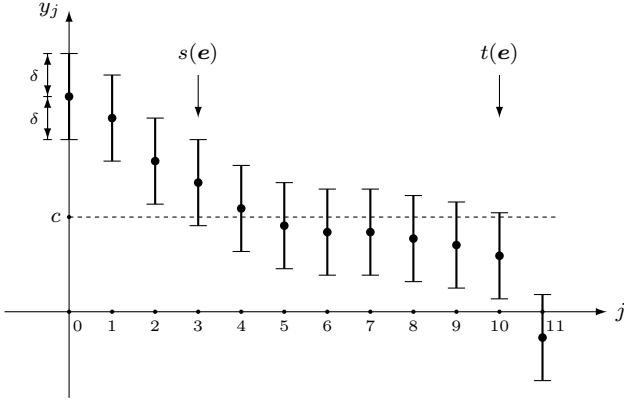


Fig. 1. Received vector \mathbf{y} for Example 1.

$d(\mathcal{C})$ 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 $\Delta \geq 2(h_w(\mathcal{C}) + 1)\delta$ and present a decoder \mathcal{D} for \mathcal{C} that corrects τ errors and detects σ additional errors.

For every $\mathbf{y} \in \mathbb{R}^n$ such that $\mathcal{E}(\mathbf{y}) \cap \mathcal{B}(n, \tau) \neq \emptyset$, define the following intersection:

$$\Lambda(\mathbf{y}) = \bigcap_{e' \in \mathcal{E}(\mathbf{y}) \cap \mathcal{B}(n, \tau)} \text{Supp}_0(e').$$

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

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

Let $\mathbf{y} = \mathbf{c} + \boldsymbol{\varepsilon} + \mathbf{e}$ now be a particular received vector, where $\mathbf{c} \in \mathcal{C}$, $\boldsymbol{\varepsilon} \in \mathcal{Q}(n, \delta)$, and $\mathbf{e} \in \mathcal{B}(n, \tau + \sigma)$. If $\mathbf{e} \in \mathcal{B}(n, \tau)$, then necessarily $\mathcal{E}(\mathbf{y}) \cap \mathcal{B}(n, \tau) \neq \emptyset$ and therefore $\mathcal{D}(\mathbf{y}) \neq \text{"e"}$. Moreover, $\Lambda(\mathbf{y}) \subseteq \text{Supp}_0(\mathbf{e})$ and, therefore, condition (D1) holds. As for condition (D2), observe that $\mathcal{D}(\mathbf{y}) \neq \text{"e"}$ implies by our definition of \mathcal{D} that $\mathcal{E}(\mathbf{y}) \cap \mathcal{B}(n, \tau) \neq \emptyset$. Hence, by Lemma 2, $\text{Supp}_\Delta(\mathbf{e}) \subseteq \text{Supp}_0(e')$ for every $e' \in \mathcal{E}(\mathbf{y}) \cap \mathcal{B}(n, \tau)$ and, so, $\text{Supp}_\Delta(\mathbf{e}) \subseteq \Lambda(\mathbf{y})$. \square

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 $\mathbf{y} = (y_j)_{j \in [n]}$ in \mathbb{R}^n , we assume for convenience of notation that its entries are permuted so that

$$y_0 \geq y_1 \geq \dots \geq y_{n-1}. \quad (4)$$

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

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

$$\bigcap_{j \in \text{Supp}_0(\mathbf{e})} \mathcal{I}_j \neq \emptyset, \quad (5)$$

where $\overline{\text{Supp}_0(\mathbf{e})} = [n] \setminus \text{Supp}_0(\mathbf{e})$. Indeed, each element c that belongs to the intersection (5) corresponds to a codeword $\mathbf{c} = c \cdot \mathbf{1}$ such that $\mathbf{y} - \mathbf{c} - \mathbf{e} \in \mathcal{Q}(n, \delta)$. Denote by $s(\mathbf{e})$ and $t(\mathbf{e})$

Input: $\mathbf{y} = (y_j)_{j \in [n]} \in \mathbb{R}^n$ such that (4) holds.

Output: $\mathcal{D}(\mathbf{y})$, which is either a subset of $[n]$, or "e".

If $y_j > y_{j+n-1-\tau} + 2\delta$ for all $j \in [\tau+1]$ then
Return "e";

Else {

$i \leftarrow \min\{j \in [\tau+1] : y_j \leq y_{j+n-1-\tau} + 2\delta\}$;

$i' \leftarrow \max\{j \in [n] \setminus [n-1-\tau] : y_{j-(n-1-\tau)} \leq y_j + 2\delta\}$;

Return $[i] \cup ([n] \setminus [i'+1])$.

}

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

the smallest and largest indexes, respectively, in $\overline{\text{Supp}_0(\mathbf{e})}$; note that

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

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

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

which, in turn, is equivalent to

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

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

Conversely, any index $j \in [\tau+1]$ such that

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

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

These observations lead to the decoding algorithm shown in Figure 2. The "if" clause checks whether $\mathcal{E}(\mathbf{y}) \cap \mathcal{B}(n, \tau)$ is empty. E.g., for the vector \mathbf{y} in Figure 1, the return value will be "e" if and only if $\tau < 4$. The "else" clause then computes the set $\Lambda(\mathbf{y})$. Specifically, among all error vectors in $\mathcal{E}(\mathbf{y}) \cap \mathcal{B}(n, \tau)$, let \mathbf{e} and \mathbf{e}' be such that $s(\mathbf{e})$ is the smallest and $t(\mathbf{e}')$ is the largest; note that $i = s(\mathbf{e})$ and $i' = t(\mathbf{e}')$. Since $\tau \leq (n-1)/2$, the sets $\overline{\text{Supp}_0(\mathbf{e})}$ and $\overline{\text{Supp}_0(\mathbf{e}')}$ intersect and, so,

$$[n] \setminus \Lambda(\mathbf{y}) = \bigcup_{\mathbf{e} \in \mathcal{E}(\mathbf{y}) \cap \mathcal{B}(n, \tau)} \overline{\text{Supp}_0(\mathbf{e})} = \{i, i+1, \dots, i'\}.$$

For the vector \mathbf{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 $y_3 \leq y_{10} + 2\delta$. \square

ACKNOWLEDGMENT

I would like to thank Andrew Jiang who brought to my attention the error in the proof of Lemma 2.

REFERENCES

- [1] R.M. Roth, "Analog error-correcting codes," *IEEE Trans. Inf. Theory*, 66 (2020), 4075–4088.

¹However, 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 $\{0, 1, 10, 11\}$.