Turbo-like Iterative Least-Squares Decoding of Analog Codes

Fangning Hu, Werner Henkel
International University Bremen
E-mail {w.henkel,f.hu}@iu-bremen.de

***Abstract*— We described a Turbo-like iterative decoding for analog product codes and prove that the iterative decoding method is an iterative projection in Euclidean space and converge to least-squares solution. Using this geometric point of view, any block analog codes can be decoded by a similar iterative method.**

I. Encoding

We first study the analog product code with parity-check component codes [1]. The encoding extends construction of a product code with binary parity check component codes to the analog (real/complex) number field. The $N = n^2$ analog symbols are first arranged by an $n \times n$ matrix and then mapped into an $(n+1) \times (n+1)$ matrix $\boldsymbol{X}$ such that the sum along each column equals to zero and the sum along each row equals to zero.

In order to simplify the analysis, we define $\boldsymbol{x}$ contains the sequence of columns of $\boldsymbol{X}$:

$$\boldsymbol{x} = \text{vec } (\boldsymbol{X})$$

A parity-check matrix $\boldsymbol{H}, \boldsymbol{H}\boldsymbol{x} = \boldsymbol{0}$ can be constructed as

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \otimes \boldsymbol{1}^T \\ \boldsymbol{1}^T \otimes \boldsymbol{I} \end{bmatrix} \tag{1}$$

where $\otimes$ denotes the Kronecker product and $\boldsymbol{1}$ is an $(n+1)$-dimensional column vector of $1'$s. $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ are corresponding to column and row constraints, respectively. It is known that the codeword space $\mathcal{X}$ is uniquely defined by the parity-check matrix as

$$\mathcal{X} = \{\boldsymbol{x} : \boldsymbol{H}\boldsymbol{x} = \boldsymbol{0}\}.$$

If we define two spaces corresponding to the column and row constraints as

$$\begin{aligned} \mathcal{H}_1 &= \{\boldsymbol{x} : \boldsymbol{H}_1\boldsymbol{x} = \boldsymbol{0}\} \\ \mathcal{H}_2 &= \{\boldsymbol{x} : \boldsymbol{H}_2\boldsymbol{x} = \boldsymbol{0}\} \end{aligned} \tag{2}$$

then $\mathcal{X} = \mathcal{H}_1 \cap \mathcal{H}_2$.

II. Decoding

Given the noise corrupted received codeword $\boldsymbol{Y}$ and knowing that the rows and columns sum to be zeros, we compute two types of information for each symbol:

$$\begin{aligned} y_{1i,j} &= -\sum_{j=1, j\neq i}^{n+1} y_{i,j} \quad i,j = 1,\ldots,n+1 \\ y_{2i,j} &= -\sum_{i=1, i\neq j}^{n+1} y_{i,j} \quad i,j = 1,\ldots,n+1 \end{aligned} \tag{3}$$

where $y_{i,j}$ denoting the components of $\boldsymbol{Y}$. The algorithm is the analog counterpart of the one for binary product codes usually used to explain Turbo decoding. Equations (3) show the extrinsic information of the rows and columns, respectively.

Let $\boldsymbol{y} = \text{vec } (\boldsymbol{Y})$, the two types of extrinsic information can also be represented by vector format:

$$(\boldsymbol{I} \otimes \bar{\boldsymbol{I}})\boldsymbol{y} \quad \text{and} \quad (\bar{\boldsymbol{I}} \otimes \boldsymbol{I})\boldsymbol{y}$$

where $\boldsymbol{I}$ is the $(n+1) \times (n+1)$ identity matrix, $\bar{\boldsymbol{I}} = \boldsymbol{E} - \boldsymbol{I}$, $\boldsymbol{E}$ is an $(n+1) \times (n+1)$ matrix of $1'$s.

The decoding method is an analog counterpart of the Turbo decoding for binary codes in [2] where the estimated vector is computed as the weighted sum of the previous vector and the extrinsic information vectors

$$\begin{aligned} \boldsymbol{y}_{(k)} &= (\boldsymbol{y}_{(k-1)} - w(\boldsymbol{I} \otimes \bar{\boldsymbol{I}})\boldsymbol{y}_{(k-1)} - w(\bar{\boldsymbol{I}} \otimes \boldsymbol{I})\boldsymbol{y}_{(k-1)})/(1 + \\ &= \boldsymbol{y}_{(k-1)} - \boldsymbol{y}^{(1)}_{(k-1)} - \boldsymbol{y}^{(2)}_{(k-1)} \end{aligned} \tag{4}$$

where w is weight parameter, k denotes the iteration step index and

$$\begin{aligned} \boldsymbol{y}^{(1)}_{(k-1)} &= \tfrac{w}{1+2w}(\boldsymbol{I} \otimes \boldsymbol{E})\boldsymbol{y}_{(k-1)} \\ \boldsymbol{y}^{(2)}_{(k-1)} &= \tfrac{w}{1+2w}(\boldsymbol{E} \otimes \boldsymbol{I})\boldsymbol{y}_{(k-1)}. \end{aligned} \tag{5}$$

We can prove that $\boldsymbol{y}^{(1)}_{(k-1)}$ is orthogonal to space $\mathcal{H}_1$ by showing that it is a linear combination of columns of $\boldsymbol{H}_1^T$ as

$$\boldsymbol{y}^{(1)}_{(k-1)} = \boldsymbol{H}_1^T \boldsymbol{\alpha}$$

where

$$\boldsymbol{\alpha} = \frac{w}{1+2w}\boldsymbol{H}_1 \boldsymbol{y}_{(k-1)}.$$

Furthermore, when $w = \frac{1}{n-1}$, $\boldsymbol{H}_1 \cdot (\boldsymbol{y}_{(k-1)} - \boldsymbol{y}^{(1)}_{(k-1)}) = \boldsymbol{0}$. This means $\boldsymbol{y}_{(k-1)} - \boldsymbol{y}^{(1)}_{(k-1)}$ lies in $\mathcal{H}_1$ by the definition of $\mathcal{H}_1$ (r.f. (2)). Thus, when $w = \frac{1}{n-1}$, $\boldsymbol{y}_{(k-1)} - \boldsymbol{y}^{(1)}_{(k-1)}$ is the projection of $\boldsymbol{y}_{(k-1)}$ onto space $\mathcal{H}_1$. Similarly, $\boldsymbol{y}_{(k-1)} - \boldsymbol{y}^{(2)}_{(k-1)}$ is the projection of $\boldsymbol{y}_{(k-1)}$ onto space $\mathcal{H}_2$, see Fig. 1.

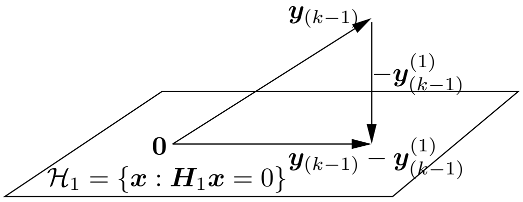

Fig. 1. The projection of $\boldsymbol{y}_{(k-1)}$ onto the space $\mathcal{H}_1$

A geometric illustration of Eqn. (4) for $w = \frac{1}{n-1}$ is given in Fig. 2. At each iteration step k, the previous vector $\boldsymbol{y}_{(k-1)}$ is projected onto $\mathcal{H}_1$ and $\mathcal{H}_2$ in parallel which deliver two projection vectors $-\boldsymbol{y}^{(1)}_{(k-1)}$ and $-\boldsymbol{y}^{(2)}_{(k-1)}$. Both projection vectors are added to $\boldsymbol{y}_{(k-1)}$ resulting the current vector $\boldsymbol{y}_{(k)}$. From Fig. 2, we see that this process makes $\boldsymbol{y}_{(\infty)}$ converge to $\mathcal{H}_1 \cap \mathcal{H}_2$ which is actually the least-squares solution. It is shown in (5) that the length

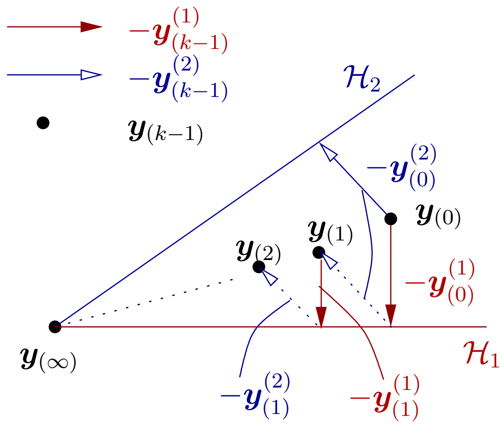

Fig. 2. A geometric illustration of the iterative decoding when $w = \frac{1}{n-1}$.

of $\boldsymbol{y}^{(1)}_{(k-1)}, \boldsymbol{y}^{(2)}_{(k-1)}$ is determined by the parameter w, the larger w is, the longer the length is.

In order to confirm $\boldsymbol{y}_{(\infty)} \in \mathcal{H}_1 \cap \mathcal{H}_2$, $\boldsymbol{y}_{(k-1)}$ should always be located in between $\mathcal{H}_1$ and $\mathcal{H}_2$ in each iterative step which means $0 < w \leq 1/(n-1)$.

Furthermore, we notice that the speed of the convergence can be improved if w is properly chosen.

III. Iterative Decoding for any Block Analog Code

For any block analog code, its parity check matrix $\boldsymbol{H}$ can be expressed as

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix}.$$

Let $\mathcal{H}_1, \mathcal{H}_2$ be orthogonal spaces to $\boldsymbol{H}_1, \boldsymbol{H}_2$, respectively. Our decoding approach is to find the projection of a vector $\boldsymbol{y}_{(k-1)}$ onto $\mathcal{H}_1, \mathcal{H}_2$.

Assume that $\boldsymbol{y}_{(k-1)} - \boldsymbol{y}^{(1)}_{(k-1)}$ is the projection of $\boldsymbol{y}_{(k-1)}$ onto space $\mathcal{H}_1$ and $\boldsymbol{y}_{(k-1)} - \boldsymbol{y}^{(2)}_{(k-1)}$ is the projection of $\boldsymbol{y}_{(k-1)}$ onto space $\mathcal{H}_2$. The iterative algorithm can be written as

$$\boldsymbol{y}_{(k)} = \boldsymbol{y}_{(k-1)} - \lambda \boldsymbol{y}^{(1)}_{(k-1)} - \lambda \boldsymbol{y}^{(2)}_{(k-1)}.$$

As long as $\lambda \leq 1$, $\boldsymbol{y}_{(\infty)}$ will converge to the least-squares solution.

According to Fig. 1, $\boldsymbol{y}^{(1)}_{(k-1)}$ is orthogonal to $\mathcal{H}_1$, thus can be expressed as a linear combination of columns of $\boldsymbol{H}_1^T$. Without loss of generality, suppose

$$\boldsymbol{y}^{(1)}_{(k-1)} = \boldsymbol{H}_1^T \boldsymbol{\alpha}.$$

Since $\boldsymbol{y}_{(k-1)} - \boldsymbol{y}^{(1)}_{(k-1)}$ lies in $\mathcal{H}_1$, we have

$$\boldsymbol{H}_1 \cdot (\boldsymbol{y}_{(k-1)} - \boldsymbol{y}^{(1)}_{(k-1)}) = \boldsymbol{H}_1 \cdot (\boldsymbol{y}_{(k-1)} - \boldsymbol{H}_1^T \boldsymbol{\alpha}) = \boldsymbol{0}.$$

The solution for $\boldsymbol{\alpha}$ is

$$\boldsymbol{\alpha} = (\boldsymbol{H}_1 \boldsymbol{H}_1^T)^{-1} \boldsymbol{H}_1 \boldsymbol{y}_{(k-1)}$$

The prerequisite for this solution is that $\boldsymbol{H}_1$ is a row full-rank matrix, otherwise the inverse of $(\boldsymbol{H}_1 \boldsymbol{H}_1^T)$ does not exit. Now, $\boldsymbol{y}^{(1)}_{(k-1)}$ can be expressed as

$$\boldsymbol{y}^{(1)}_{(k-1)} = \boldsymbol{H}_1^T (\boldsymbol{H}_1 \boldsymbol{H}_1^T)^{-1} \boldsymbol{H}_1 \boldsymbol{y}_{(k-1)} \tag{6}$$

Similarly,

$$\boldsymbol{y}^{(2)}_{(k-1)} = \boldsymbol{H}_2^T (\boldsymbol{H}_2 \boldsymbol{H}_2^T)^{-1} \boldsymbol{H}_2 \boldsymbol{y}_{(k-1)}.$$

where $\boldsymbol{H}_2$ must also be a row full-rank matrix.

IV. Conclusion

Starting from a geometric illustration of the iterative decoding of analog product codes, we found an iterative decoding procedure for arbitrary linear analog block codes by splitting its parity-check matrix in two and projecting received codewords onto the Null spaces of these two matrices in an iterative fashion.

REFERENCES

[1] M. Mura, W. Henkel and L. Cottatellucci, Iterative Least-Squares Decoding of Analog Product Codes, *IEEE Intern. Symp. on Information Theory (ISIT 2003)*, Yokohama, June 29 - July 4, 2003.

[2] J. Hagenauer, E. Offer, and L. Papke, ”iterative decoding of binary block and convolutional codes,” *IEEE Trans. Information Theory*, vol. 42, no.2, March 1996.