

Adaptive Decoding of Turbo Product Codes Using Soft Information

1st Jae-Yong Son

SK Hynix

Seongnam, Republic of Korea

mike9551@postech.ac.kr

Abstract—This paper proposes two adaptive Chase decoding algorithms for turbo product codes (TPCs) in order to improve the trade-off between error-correcting performance and decoding complexity. The key idea is to reduce the number of test sequences (TSs) in the Chase algorithm by analyzing the soft information for each row or column codeword. The first proposed algorithm utilizes the entropy of the soft decoder input vector: if the entropy of a row or column vector is below a predefined threshold, it is assumed that the decoder input vector is sufficiently reliable, and the number of least reliable bit (LRB) positions used in the Chase algorithm is reduced. The second algorithm uses a confidence-ratio metric, defined as the proportion of decoder input values whose absolute magnitude is below a certain threshold. If the confidence ratio is sufficiently low, the number of LRB positions is similarly reduced. Both algorithms can be combined with the existing syndrome-based decoding algorithm, where error-free hard-decision vectors are detected and excluded from Chase decoding. Simulation results show that the proposed algorithms achieve comparable bit error rate (BER) performance to conventional Chase-based TPCs while fairly reducing decoding complexity. In particular, when combined with the syndrome-based decoding algorithm, the total number of decoding attempts is further reduced by approximately 10% beyond what is achieved by the syndrome-based decoder alone, without any performance degradation.

adaptive decoding, Chase algorithm, entropy, log-likelihood ratio (LLR), turbo product codes (TPCs).

I. INTRODUCTION

Since the introduction of turbo product codes (TPCs) in [1], [2], they have been widely recognized as a practical solution for various applications such as wireless communications and data storage systems, due to their strong error-correcting performance and relatively low decoding complexity. The iterative decoding of TPCs typically relies on the Chase algorithm [3], which generates a list of candidate codewords by identifying the p least reliable bit (LRB) positions in the soft input

vector. From these positions, 2^p test sequences (TSs) are constructed and decoded using a hard-decision decoder (HDD), after which extrinsic information is computed from the resulting candidate list.

Although the Chase algorithm efficiently produces candidate codewords by leveraging the HDD of the component code, it consistently relies on a fixed number of p LRB positions without accounting for the reliability of the input vector. As a result, the decoding complexity of TPCs remains constant across iterations and the signal-to-noise ratio (SNR) conditions. To address this limitation, previous works [4]–[10] have independently proposed syndrome-based decoding approaches. These methods compute the syndrome of the hard-decision version of the received vector, and if the syndrome is zero, the Chase algorithm is skipped. In such cases, extrinsic information is directly computed by scaling the signed hard-decision vector. This strategy significantly reduces decoding complexity with only a negligible performance loss, making it a promising alternative to the conventional decoding process.

In this paper, two adaptive decoding algorithms are proposed that can be naturally combined with existing syndrome-based methods. Each proposed scheme first analyzes the reliability of the soft decoder input vector by calculating either the entropy or a confidence ratio. As decoding progresses or as the SNR increases, each element of the input vector tends to align more closely with the corresponding transmitted bit. Hence, if the entropy of the input vector, as defined in information theory, or the confidence ratio, defined as the proportion of low-magnitude log-likelihood ratio (LLR) values, falls below a predefined threshold, the vector is considered sufficiently reliable. In such cases, the Chase algorithm is applied with only $p - 1$ LRB positions, thereby reducing the number of TSs. This enables each row or column vector to be decoded with a variable number of TSs depending on its estimated reliability.

Simulation results show that the proposed algorithms achieve a BER loss within 0.1 dB at 10^{-5} compared to the conventional Chase-based TPCs, while fairly reducing decoding complexity. Furthermore, when combined with the syndrome-based algorithm, an additional 10% reduction in decoding attempts can be achieved without noticeable degradation in performance.

Contributions. This work proposes two *adaptive* Chase-based decoders for TPCs that (i) select only reliable rows/columns for component HDD based on soft information, and (ii) reduce the number of decoding trials using a data-driven confidence test. The key outcomes are: (1) complexity reduction measured as the average number of component HDD invocations per each row/column, (2) preserved BER across SNRs, and (3) seamless compatibility with existing syndrome-aided decoders.

Novelty vs. prior art. Unlike conventional Chase-Pyndiah style decoders with fixed iteration counts and full row/column sweeps, our approach *adapts* the number and locations of component decodings using the reliability based on the soft information, thereby shrinking the candidate search where it matters and terminating when confidence is sufficient. This data-driven control is the primary difference that translates to complexity savings at equal BER.

The rest of this paper is organized as follows. Section II reviews the conventional decoding algorithm for TPCs introduced in [1], [2]. Section III presents the proposed adaptive decoding methods. Simulation results in terms of performance and complexity are provided in Section IV. Finally, conclusions and future directions are discussed in Section V.

II. CONVENTIONAL DECODING ALGORITHM FOR TPCs

Consider a two-dimensional TPC constructed from two binary linear block codes, $\mathcal{C}_1(n_1, k_1, d_1)$ and $\mathcal{C}_2(n_2, k_2, d_2)$, where n_i , k_i , and d_i denote the code-word length, the information length, and the minimum Hamming distance of $\mathcal{C}_i$, respectively. The information bits are arranged in a $k_1 \times k_2$ array. Each column of this array is encoded using $\mathcal{C}_1$, resulting in an $n_1 \times k_2$ array. Then, each row of this intermediate array is encoded using $\mathcal{C}_2$, producing a final $n_1 \times n_2$ TPC. The resulting code is an $[n_1 n_2, k_1 k_2, d_1 d_2]$ linear block code. For simplicity, we consider the symmetric case of $\mathcal{C}(n, k, d)^2$ with $k = k_1 = k_2$, $n = n_1 = n_2$, and $d = d_1 = d_2$.

Without loss of generality, this encoding method can be naturally extended to higher-dimensional TPCs.

The encoded bits $\{0, 1\}$ are mapped to binary phase-shift keying (BPSK) modulated symbols $\{-1, +1\}$ via $\varphi(a) = 2a - 1$, and each row vector $\mathbf{X} = (x_1, x_2, \dots, x_n)$, with $x_l \in \{-1, +1\}$, is transmitted one by one over an additive white Gaussian noise (AWGN) channel.

Let $\mathbf{R} = (r_1, r_2, \dots, r_n)$ denote the received signal vector, where each element is given by $r_l = x_l + g_l$ for $l = 1, 2, \dots, n$, and g_l is a zero-mean Gaussian noise with variance σ^2 . After receiving all row vectors, the received row vectors are transformed to the received array for a TPC and then the received array is iteratively decoded in a row-wise and column-wise manner. Each row or column vector is decoded using a two-stage process [1], [2]. In the first stage, the Chase algorithm generates a list of candidate codewords using the following steps:

- 1) Obtain the hard-decision vector $\mathbf{Y} = (y_1, y_2, \dots, y_n)$ from $\mathbf{R}$, where $y_l = 1$ if $r_l > 0$ and zero, otherwise.
- 2) Identify the p LRB positions in $\mathbf{R}$ and initialize $j = 0$.
- 3) Construct the j th test pattern (TP) $\mathbf{T}^j \triangleq (t_1^j, t_2^j, \dots, t_n^j)$, where t_l^j is either 0 or 1 at the p LRB positions and zero at the other positions.
- 4) Form the j th TS $\mathbf{Z}^j = \mathbf{Y} \oplus \mathbf{T}^j$, where $\oplus$ is the component-wise modulo-2 sum operator.
- 5) Apply an algebraic HDD to $\mathbf{Z}^j$ to obtain the codeword $\mathbf{C}^j \triangleq (c_1^j, c_2^j, \dots, c_n^j)$ and add it to the candidate list.
- 6) Compute the Euclidean distance between $\varphi(\mathbf{C}^j) \triangleq (\varphi(c_1^j), \varphi(c_2^j), \dots, \varphi(c_n^j))$ and $\mathbf{R}$. Set $j = j + 1$. Repeat Steps 3)–6) until $j = 2^p$.
- 7) Select a decision codeword $\mathbf{C} \triangleq (c_1, c_2, \dots, c_n)$ with $c_l \in \{0, 1\}$ such that $\varphi(\mathbf{C})$ has minimum Euclidean distance from $\mathbf{R}$ among all candidate codewords in the list.

In the second decoding stage, the extrinsic information for each decision bit c_l , $l = 1, 2, \dots, n$, are computed for use in the iterative decoding:

- 1) Compute the soft output $\Lambda(c_l)$, $l = 1, 2, \dots, n$, as
$$\Lambda(c_l) = \frac{1}{4} \left[\|\mathbf{R} - \varphi(\mathbf{B})\|^2 - \|\mathbf{R} - \varphi(\mathbf{C})\|^2 \right] \cdot \varphi(c_l)$$

where $\mathbf{B} = (b_1, b_2, \dots, b_n)$ with $b_l \in \{0, 1\}$ is a competing codeword from the list with $b_l \neq c_l$ that has the minimum Euclidean distance from $\mathbf{R}$.

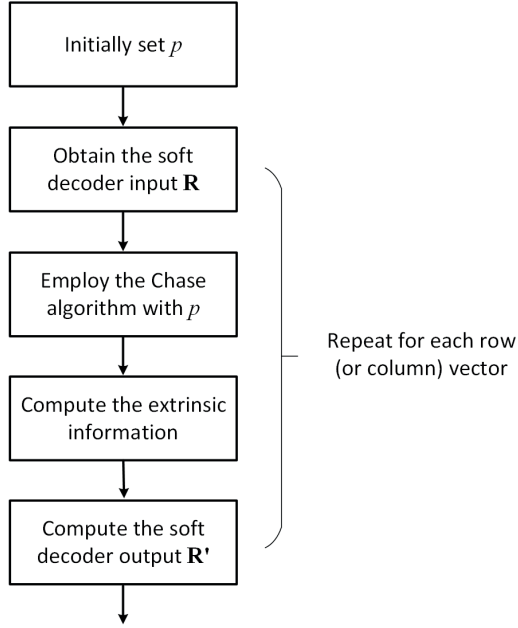


Fig. 1. Conventional decoding flow for TPCs.

- 2) If such a competing codeword exists in the list, the extrinsic information w_l , is calculated as

$$w_l = \Lambda(c_l) - r_l.$$

Otherwise, it is set to

$$w_l = \beta \cdot \varphi(c_l)$$

where $\beta \geq 0$ is a predefined reliability factor.

After decoding each row vector in the i th half-iteration, the soft output $\mathbf{R}'$ is computed by adding the scaled extrinsic information (using a weight factor $\alpha \geq 0$) to the received signal vector. This updated soft output is then passed to the decoder for the next half-iteration. The conventional decoding flow for TPCs is illustrated in Fig. 1.

III. PROPOSED DECODING ALGORITHMS

As shown in Fig. 1, the conventional decoding algorithm for TPCs initializes the parameter p for the Chase algorithm and keeps it fixed throughout the iterative decoding process. As a result, 2^p HDD operations are performed on every row or column input vector at each iteration, regardless of the SNR or the iteration number. This leads to a fixed computational complexity per iteration for a given p .

To address this limitation, we propose two simple yet effective adaptive decoding algorithms that adjust the value of p dynamically based on the reliability of each

row or column input vector. By applying a simple threshold test before executing the Chase algorithm, decoding complexity can be significantly reduced with negligible performance degradation compared to the conventional scheme. Moreover, the proposed methods are compatible with existing low-complexity decoding techniques, such as the syndrome-based methods described in [4]–[10].

The first statistical measure used to assess the reliability of each input vector is entropy. In information theory, entropy quantifies the average uncertainty of a random variable [11]. Given a discrete random variable X with probability distribution $p(x)$, the entropy is defined as

$$H(X) := - \sum_{x \in \mathcal{X}} p(x) \log p(x).$$

For decoding, the soft input is represented by LLR defined as

$$\text{LLR}(r_i) := \ln \frac{p(x_i = 1|r_i)}{p(x_i = 0|r_i)}, \quad i = 1, 2, \dots, n.$$

From this, the posterior probabilities are obtained as:

$$\begin{aligned} p(x_i = 0|r_i) &= \frac{1}{1 + \exp(\text{LLR}(r_i))}, \\ p(x_i = 1|r_i) &= 1 - p(x_i = 0|r_i). \end{aligned} \quad (1)$$

In an AWGN channel, the LLR for BPSK modulated signals is given by $\text{LLR}(r_i) = \frac{2}{\sigma^2} r_i$. Since the original TPC decoding in [2] uses the received signal directly (rather than explicitly computing the LLR), and since the proposed method only requires a threshold test, we normalize the LLR by a scaling factor of $2/\sigma^2$ for computing the probabilities in (1). The average entropy for a given input vector $\mathbf{X}$ is then approximated as

$$H(\mathbf{X}) = \mathbb{E}[H(x_i)], \quad (2)$$

where $H(x_i) = -p(x_i = 0|r_i) \ln p(x_i = 0|r_i) - p(x_i = 1|r_i) \ln p(x_i = 1|r_i)$, $i = 1, 2, \dots, n$. As the SNR or the number of decoding iterations increases, the soft input becomes more reliable, and its entropy decreases accordingly. If the average entropy in (2) falls below a predefined threshold, the input vector is deemed reliable, and the number of LRB positions is reduced from p to $p - 1$. This threshold can be determined empirically to strike a balance between decoding performance and complexity.

The second statistical measure is more intuitive. As the reliability of the soft decoder input increases, the received signal distribution becomes sharper, and the overlap between positive and negative distributions decreases. This overlap region centered around zero indicates uncertainty and is estimated using the confidence

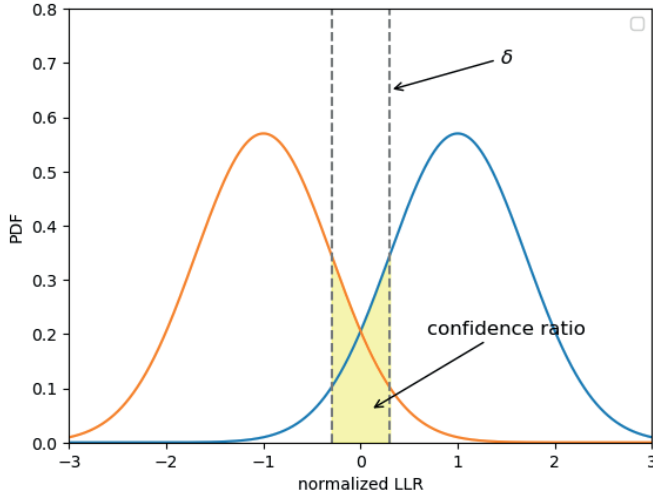


Fig. 2. LLR distribution for BPSK modulated signals over an AWGN channel. The proposed scheme triggers trials only for samples with $|r_i| < \delta$, which concentrates search where uncertainty is high.

ratio, defined as the proportion of elements in the soft input vector whose magnitudes fall below a certain threshold δ . That is:

$$\text{Confidence ratio} := \mathbb{E}[|r_i| < \delta].$$

This is visually illustrated in Fig. 2. As the reliability of the input vector increases, this confidence ratio naturally decreases. If the confidence ratio for a row or column input vector falls below a predefined threshold, it is considered reliable, and the number of LRB positions is again reduced to $p - 1$.

Each of these statistical measures is evaluated before the Chase algorithm is applied. Based on the measured reliability, the decoder dynamically adjusts the number of TPs generated. The overall process of the proposed adaptive decoding approach is summarized in Fig. 3.

If the decoder additionally employs a syndrome-based method, this step can be executed prior to the reliability check. Specifically, if the syndrome of the hard-decision version of the input vector is zero, the Chase algorithm is skipped, and the extrinsic information is computed by scaling the signed hard-decision vector. Otherwise, the proposed adaptive Chase decoding is performed using fewer TPs. This hybrid approach allows the proposed algorithms to be easily combined with syndrome-based methods, enabling further complexity reduction without sacrificing decoding performance.

IV. NUMERICAL RESULTS

A TPC constructed with a $[64, 51, 6]$ extended BCH (eBCH) code as its component code is considered for

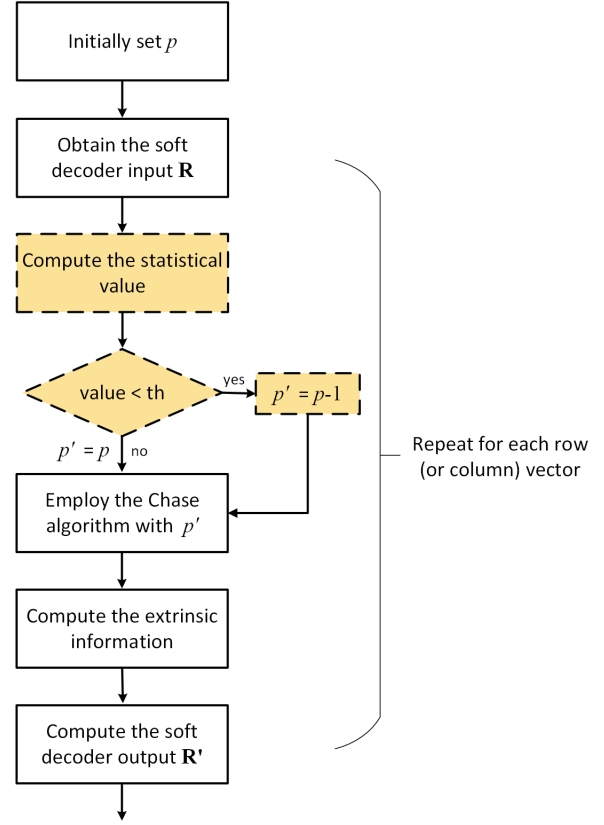


Fig. 3. Proposed adaptive decoding flow for TPCs.

both rows and columns, respectively. The Berlekamp-Massey algorithm [12], [13] is used for hard-decision decoding of each component code. Following the conventional approach in [2], the weighting factor $\alpha(m)$ and the reliability factor $\beta(m)$ for the m th half-iteration are configured as

$$\begin{aligned} \alpha(m) &= [0.0, 0.2, 0.3, 0.5, 0.7, 0.9, 1.0, 1.0, \dots], \\ \beta(m) &= [0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0, 1.0, \dots]. \end{aligned}$$

For the proposed adaptive algorithms, the threshold values for entropy and confidence ratio methods are empirically set to 0.005 and 0.01, respectively, by considering BER performance and the complexity together. The number of LRB positions is initialized with $p = 4$ and the confidence threshold for the second method is set to $\delta = 0.5$.

The BER performance of the proposed algorithms is illustrated in Fig. 4. Both proposed algorithms achieve performance comparable to that of the conventional and syndrome-based decoding algorithms. Specifically, the former shows at most a 0.1 dB performance loss at a BER of 10^{-5} compared to the conventional decoding algorithm. Furthermore, when the proposed methods are

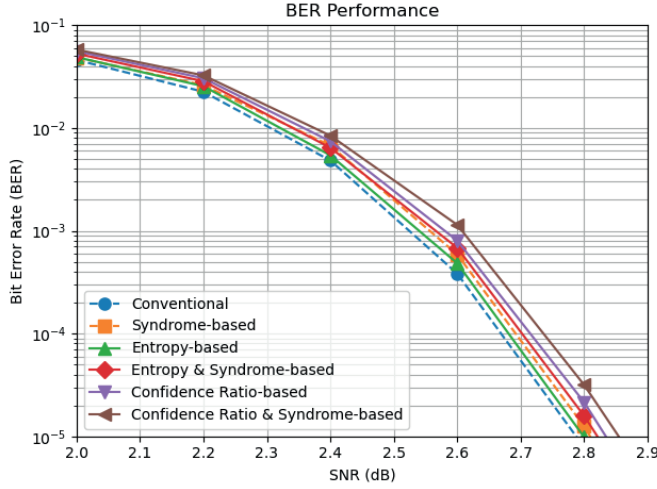


Fig. 4. BER performance of an $eBCH(64, 51, 6)^2$ code for up to 4 decoding iterations.

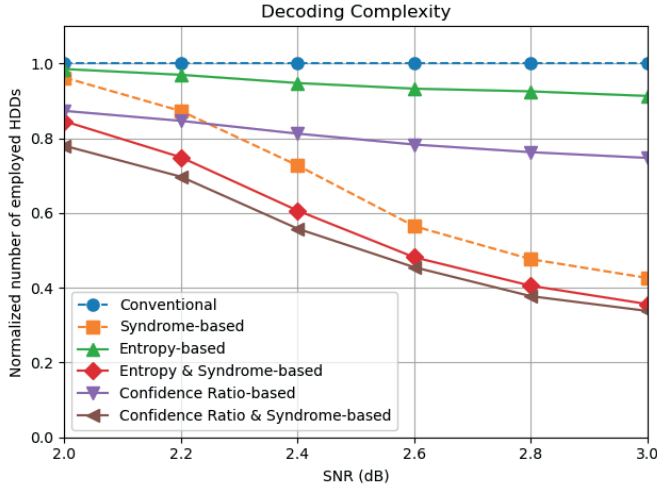


Fig. 5. Normalized number of HDD attempts for decoding an $eBCH(64, 51, 6)^2$ code over 4 iterations.

combined with syndrome-based decoding, no noticeable degradation in performance is observed.

The decoding complexity of each scheme is evaluated in terms of the normalized number of HDD attempts, as shown in Fig. 5. While the conventional decoding algorithm employs a fixed number of HDD attempts per iteration, the proposed algorithms fairly reduce the number of decoding trials by adapting p according to input reliability of the decoder. In addition, when combined with the syndrome-based decoding algorithm, the proposed methods provide an additional complexity reduction of approximately 10% without any loss in decoding performance.

V. CONCLUSIONS AND FUTURE WORKS

This paper proposed two simple adaptive decoding algorithms for TPCs. By evaluating a statistical measure of reliability of the decoder input before applying the Chase algorithm, the proposed methods effectively reduce computational complexity while incurring minimal performance degradation. Furthermore, when combined with a syndrome-based decoding strategy, the overall decoding complexity can be further reduced without sacrificing error-correcting capability.

The threshold values and parameters such as δ used in this work were selected empirically and are not yet fully optimized. Future work may focus on optimizing these parameters using AI-aided or analytical methods. Additionally, rather than applying a single-step reduction in the number of LRB positions (i.e., $p \rightarrow p-1$), a multi-level reduction strategy (e.g., $p \rightarrow p-1 \rightarrow p-2 \rightarrow \dots$) could be explored. Such approaches are expected to further reduce decoding complexity, especially under high-reliability conditions.

REFERENCES

- [1] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, "Near optimum decoding of product codes," in *Proc. IEEE GLOBECOM 1994*, vol. 1, pp. 339-343, Nov.-Dec. 1994.
- [2] R. M. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003-1010, Aug. 1998.
- [3] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. IT-18, no. 1, pp. 170-182, Aug. 1972.
- [4] E.-H. Lu and P.-Y. Lu, "A syndrome-based hybrid decoder for turbo product codes," in *Proc. IEEE 3CA 2010*, vol. 1, pp. 280-282, May 2010.
- [5] L. Li, F. Zhang, P. Zheng and Z. Yang, "Improvements for decoding algorithm of turbo product codes," in *Proc. IEEE Int. Conf. Signal Proc. Commun. Comput. (ICSPCC)*, pp. 374-378, Aug. 2014.
- [6] L. Wang, M. Wang, L. Wang, and F. Yuan, "A joint decoding algorithm for turbo product codes applied in PCM/FM telemetry system," in *Proc. IEEE CISP 2014*, pp. 826-830, Oct. 2014.
- [7] J. Son, K. Cheun, and K. Yang, "Low-complexity decoding of block turbo codes based on the Chase algorithm," *IEEE Commun. Letters*, vol. 21, no. 4, pp. 706-709, Apr. 2017.
- [8] B. Ahn, S. Yoon, and J. Heo, "Low complexity syndrome-based decoding algorithm applied to block turbo codes," *IEEE Access*, vol. 6, pp. 26693-26706, Apr. 2018.
- [9] J. Son, J. J. Kong, and K. Yang, "Efficient decoding of block turbo codes," *J. Comm. and Net.*, vol. 20, no. 4, pp. 345-353, Aug. 2018.
- [10] S. Yoon, B. Ahn, and J. Heo, "An advanced low-complexity decoding algorithm for turbo product codes based on the syndrome," *Eurasip J. Wireless Communications and Networking* 2020, 126 (2020), May 2020.
- [11] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 4, pp. 623-656, Oct. 1948.

- [12] E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, 1968.
- [13] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15, no. 1, pp. 122-127, Jan. 1969.