Blind Estimation of Code Parameters for Product Codes over Noisy Channel Conditions

Swaminathan R, A.S.Madhukumar, and Wang Guohua

Abstract

Product codes are multidimensional codes constructed using multiple component codes. In this paper, novel algorithms are proposed for the blind estimation of two-dimensional product code parameters over noisy channel conditions considering Reed-Solomon (RS) and Bose-Chaudhuri-Hocquenghem (BCH) as component codes. The performance of the algorithms in terms of probability of correct estimation is investigated for different code parameters. It is observed that the accuracy improves with decrease in modulation order and code dimension values.

Index Terms

Adaptive modulation and coding, Bose-Chaudhuri-Hocquenghem (BCH) codes, blind estimation, non-cooperative communication, product codes, reconfigurable radio, Reed-Solomon (RS) codes, and signals intelligence

I. Introduction

Forward error correction (FEC) codes and interleavers are mainly used to improve the error performance by counteracting random and burst errors, respectively, in digital communication and aeronautical telemetry systems [1] and [2]. Reconstructing an unknown channel encoder

Swaminathan R is with the Discipline of Electrical Engineering, Indian Institute of Technology (IIT) Indore, Indore - 453552, India, (e-mail: swamiramabadran@iiti.ac.in)

A.S.Madhukumar is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore - 639798, (e-mail: asmadhukumar@ntu.edu.sg).

Wang Guohua was with Temasek Laboratories, Nanyang Technological University, Singapore-639798, (e-mail: wghwood@hotmail.com).

from noisy codewords is related to cryptanalysis and is called the code reconstruction problem in the literature. Here, an observer wants to decode the information bits from a noisy data stream, where the FEC code used for transmission is unknown. In general, the receiver has the knowledge of FEC code parameters used at the transmitter for decoding the message signals in most of the applications. However, in non-cooperative scenarios [3] and [4], which exist particularly in electronic warfare, military, spectrum surveillance, signals intelligence (SIGINT), and communications intelligence (COMINT) systems, the receiver has only limited knowledge about the code parameters. Hence, it is mandatory to blindly estimate the code parameters using the intercepted sequences acquired from remote sensing through aircraft and satellite.

Apart from non-cooperative context, the code parameter estimation is also useful in applications such as adaptive modulation and coding (AMC), data storage systems, reconfigurable radio systems, etc. AMC techniques are very useful in satellite communications [5] as they improve overall spectral efficiency. Recently, the attracting service of the *Internet above the clouds*, which provides direct communication among aircraft for airborne internet access instead of satellite-based access, motivates researchers to develop high spectral efficient aeronautical communication techniques. Hence, in order to meet the demand of high throughput and high spectral efficiency, AMC-based solution has been provided in [6] for aeronautical communication. The modulation and code parameters are usually communicated using control channel to the receiver in AMC systems. Therefore, blind estimation of code parameters will result in conserving the channel resources, which in turn will further enhance the spectral efficiency of AMC-based satellite and aeronautical communication systems [7]-[9]. The wireless sensor networks (WSNs) also use AMC technique to choose different channel encoders [7]. The blind estimation techniques will reduce the energy consumption of WSNs, as it is not essential to transmit the overhead information to the sensor nodes in order to indicate the changes in code parameters. Due to advancements in digital communication systems, it is always a costly and a tedious process to design separate decoder for every application. Therefore, there is a need for an intelligent receiver system which adapts itself to any applications [10]. Hence, reconfigurable radio systems are introduced. It is also essential for an ideal reconfigurable radio to blindly estimate the code parameters in order to adapt to the variations in the channel encoders for decoding the message symbols successfully.

Blind estimation of code parameters for different FEC codes in noiseless and noisy environments are extensively investigated in the past. For example, [3] proposed parameter estimation and code classification algorithms for different types of linear block codes and convolutional codes over non-erroneous channel conditions. The algorithms for the blind estimation of convolutional code parameters were proposed in [10], [11], and [12]. In [13], the parameter estimation of convolutional codes was extended to Galois field(2^m) i.e. $GF(2^m)$ case considering noiseless environment. In [14], algorithm for the blind estimation of binary cyclic code parameters was proposed. The blind recognition algorithms for estimating punctured convolutional code parameters considering erroneous channel conditions were proposed in [15] and [16]. Recently, the code classification algorithms over noisy transmission environment were proposed in [17]. Further, the algorithms for the blind estimation of codeword length of various non-binary error correcting codes were proposed and validated using different test cases in [18] for noisy environment. Recently, the algorithms for the blind estimation of Reed-Solomon (RS) code parameters were proposed in [19] for noisy environment. The blind identification of true encoder based on the soft-decision outputs was reported in [7]-[9] and [20]-[22] for low-density parity-check (LDPC), RS, convolutional, and turbo codes. Apart from the code parameter estimation algorithms, the parameter estimation algorithms were proposed for block interleaver in [17], [19], [23], [24], and [25]. Similarly, the algorithms for the parameter estimation of convolutional interleaver were reported in [26]-[29]. Recently, novel algorithms were proposed in [4] and [28] to estimate the interleaver size (or interleaver period) by exploiting the linear dependence among the received noisy codewords. Apart from blind estimation of FEC code and interleaver parameters, the joint maximum likelihood estimators for the frequency offset, channel delays, and channel gains of alamouti-encoded aeronautical telemetry system were derived and analyzed in [30].

In a coded communication system, the receiver should be able to find the beginning of each codeword in order to decode the received coded symbols and this is called frame synchronization problem in literature. The main aim of the frame synchronization method in a FEC coded communication system is to identify the position of a codeword in the received sequence. Classical frame synchronization methods inserted few synchronization/training bits, which are known to the receiver, in the transmitted sequence apart from the message bits in order to achieve synchronization [31]. Hence, the traditional methods reduce the spectral efficiency of the transmission. For this reason, blind methods for frame synchronization without inserting the training sequence were proposed. In [32] and [33], the blind frame synchronization methods based on the soft-decision outputs were proposed for product and LDPC codes assuming that the code parameters are known. Finally, in [34], a hybrid automatic repeat request (HARQ) system was proposed using turbo product codes (TPC).

However, the existing works have not proposed algorithms for the estimation of code parameters of product codes. In addition, it is also essential to recognize the type of modulation scheme in a non-cooperative scenario before identifying the channel code parameters. Recently, cumulants-based framework was proposed in [35] for automatic modulation classification. Apart from that deep-learning techniques were also proposed in [36] and [37] to identify the type of modulation scheme. Thus, automatic modulation recognition techniques are well investigated in the literature. The current work aims to identify the incoming product code parameters assuming that the types of code and modulation scheme are known apriori. Precisely, we assume that the type of modulation scheme has been identified correctly using any of the techniques reported in the literature. It is in this context, the major motivations and contributions of the present research are as follows:

A. Motivations

The main motivations of the present research are as follows:

- It is always essential to estimate the code parameters for decoding at the receiver in a non-cooperative military and spectrum surveillance systems.
- In [32], the proposed algorithm for product codes was restricted only to blind frame synchronization assuming that the code parameters are known apriori.
- To the best of our knowledge, the algorithms for the joint estimation of code parameters of product codes and bit position adjustment parameter to attain synchronization have not been proposed in the prior works.
- The probability of correct estimation of product code parameters is not extensively reported in the literature.

B. Contributions

The main contributions are as follows:

- Proposes novel algorithms for the blind recognition of different code parameters of two-dimensional RS and Bose-Chaudhuri-Hocquenghem (BCH) product codes over erroneous channel conditions.
- The proposed algorithms also estimate bit position adjustment parameter in order to achieve frame synchronization.
- Simulation results for various test cases are given to prove the robustness of the proposed algorithms.
- The performance of the proposed algorithms in terms of probability of correct estimation is extensively analyzed for different $M-$ary modulation schemes and code parameters over additive white Gaussian noise (AWGN) channel and non-binary channel i.e. $M-$ary symmetric channel, where M denotes the modulation order.

C. Comparison with our prior works

We have proposed code classification algorithms in [17] to classify the incoming coded symbols among block and convolutional codes as well as to identify important code parameters such as codeword length and code dimension. However, the proposed algorithms in [17] were based on rank values for both non-erroneous and erroneous channel conditions considering synchronized scenario. In case of erroneous channel conditions, the rank values were evaluated based on fixing a threshold value to classify dependent and independent columns. It has been observed that the threshold value varies due to noise and hence, selecting the optimal threshold value is always a challenging process in a noisy channel conditions. In the current work, the parameter estimation algorithms proposed for noisy channel conditions are not based on rank values. Hence, the proposed algorithms do not require any threshold values to recognize the code parameters unlike the algorithms proposed in [17]. Moreover, the existing algorithms were limited only to synchronized environment. In our current work, an innovative approach to synchronize both the row and column components of product codes has been proposed and integrated with the code parameter estimation algorithms. In [17], the code dimension has been estimated based on rank values that are sensitive to noisy channel conditions and generator polynomial has not been identified using the proposed algorithms. Here, an innovative approach to identify the code dimension and the generator polynomial from the roots of code polynomials based on maximum likelihood approach has been proposed.

It is also to be noted that we have modified the proposed algorithm in [19] to estimate the code parameters of RS product codes with reduced complexity. It has been shown that the proposed algorithm in the current work requires lesser number of data symbols to identify the code parameters compared to our algorithm in [19].

D. Organization of the manuscript

The manuscript is organized as follows. In Section II, an introduction about product codes and the code construction process are given. In Section III, a generic block diagram for the parameter estimation of RS and BCH product codes is given. The blind reconstruction of RS and BCH product encoder over non-erroneous (noiseless) and synchronized scenario is discussed with greater details in Section IV. Further, in Section V, the parameter estimation algorithms are given for the blind reconstruction of RS and BCH product encoder considering erroneous (noisy) and non-synchronized scenario. In Section VI, simulation results and related discussions are given for various test cases along with the performance of the proposed algorithms. Finally, concluding remarks are given in Section VII.

II. Product codes and code construction

Product codes are powerful FEC codes which can be implemented with reasonable complexity. Product codes are constructed using inner and outer codes similar to turbo convolutional code (TCC) [38]. The main advantages of product codes are high coding gain at high code rate values and simple encoding and decoding techniques. Moreover, unlike TCC, it can support high code rates with minimal complexity using high rate component codes. Product codes are mainly useful for high speed communication systems with low BER requirement. The main applications of product codes include IEEE 802.16 for fixed and mobile broadband wireless access systems known as WiMAX, IEEE 802.20, digital video broadcasting-satellite (DVB-S) system, etc. Product codes are also being used in other applications such as satellite and aeronautical communication systems [39].

Multidimensional product codes are used to construct codes with long codeword lengths and less decoding complexity using multiple component codes with short codeword lengths and high code rate values. Therefore, instead of performing small number of decoding operations over long codeword lengths, large number of decoding operations over short codeword lengths are

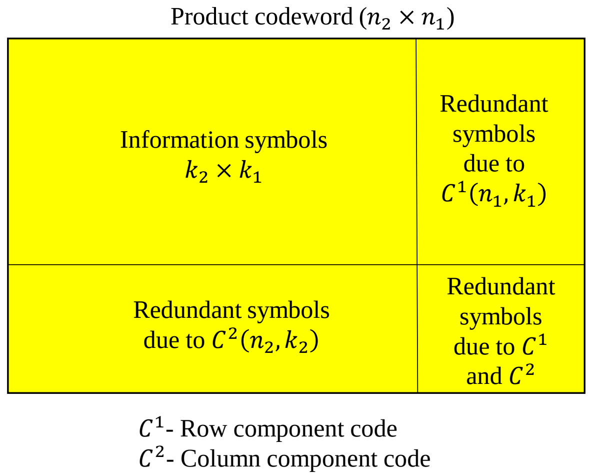


Fig. 1. Product code constructed using two linear block codes

performed to reduce the decoding complexity. Further, two-dimensional product codes perform better than multidimensional codes in terms of error performance [38]. An illustration of a two-dimensional product codeword is shown in Fig. 1. Two-dimensional product codes are constructed using two linear block codes C^i $(i = 1, 2)$ by applying serially on the two dimensions of a matrix. The two component codes C^i have the parameters $n_i, k_i, d^{(i)}_{\text{min}}$, where n_i, k_i, and $d^{(i)}_{\text{min}}$ denote the codeword length, code dimension, and minimum hamming distance in i^{th} dimension. In order to build a product code, $k_2 \times k_1$ information bits are placed in a matrix of k_2 rows and k_1 columns. Further, k_2 data blocks of length k_1 symbols are encoded by row component code C^1. As a result, k_2 codewords of length n_1 is generated and placed in a matrix of size $k_2 \times n_1$. Each column vector of length k_2 symbols is encoded using column component code C^2 and a two-dimensional codeword of size $n_2 \times n_1$ is obtained. After encoding, the parameters of the product code C are $(n_1 \times n_2, k_1 \times k_2, d^{(1)}_{\text{min}} \times d^{(2)}_{\text{min}})$. The code rate of product code is given by $r = \frac{k_1 \times k_2}{n_1 \times n_2}$.

A square product code is constructed by equating $n_1 = n_2 = n$, $k_1 = k_2 = k$ and $d^{(1)}{}_{\text{min}}$

$= d^{(2)}{}_{\text{min}} =$

d_{min} and the same is denoted as $(n, k, d_{\text{min}})^2$. Apart from square product codes, other important types include irregular product codes, non-binary product codes, etc. It is to be noted that product codes can be constructed using different component codes such as Hamming codes, BCH codes, RS codes, and LDPC codes. Irregular product codes are constructed using row/column component codes with different code rates assuming same codeword length. Non-binary product codes are constructed using non-binary FEC codes such as RS codes as component codes. In our manuscript, we restrict our discussions to two-dimensional product codes with non-binary RS and binary BCH codes as component codes and they are denoted as RS$(n_1{\times}n_2, k_1{\times}k_2, m_1{\times}m_2, p_1{\times}p_2)$ and BCH$(n_1{\times}n_2, k_1{\times}k_2, m_1{\times}m_2)$, respectively. The parameters m_i, n_i, k_i, $g_i(x)$, and p_i denote number of bits/symbol, codeword length, code dimension, generator polynomial, and the integer representation of primitive polynomial (for RS codes), respectively. The subscript $i = 1$ and 2 indicate row and column component codes, respectively. The block length or codeword length of RS and BCH codes is given by $n_i = 2^{m_i} - 1$. Let α, α^2,...,α^{2t} are the roots of the generator polynomial $g_i(x)$ of BCH and RS codes with degree $n - k$, which is given by

$$g_i(x) = \text{lcm}(\phi_1'(x), \phi_2'(x), ..., \phi_{2t}'(x)) \,, \tag{1}$$

where $\phi_i'(x)$ is the minimal polynomial of α^i. As $g_i(x)$ is defined over $GF(2^m)$ for RS code, it can be written as

$$g_i(x) = (x - \alpha)\,(x - \alpha^2)\,...(x - \alpha^{2t}) \,, \tag{2}$$

where $\phi_i'(x) = (x - \alpha^i)$ and $n - k = 2t$ for RS code.

III. Code parameter estimation process

The generic block diagram depicting the parameter estimation process of RS and BCH product codes is given in Fig. 2. In the case of RS product codes, the coded symbols are converted into binary data and then modulated using suitable modulation schemes for storage or transmission. However, the coded bits are directly modulated using appropriate modulation schemes in the case

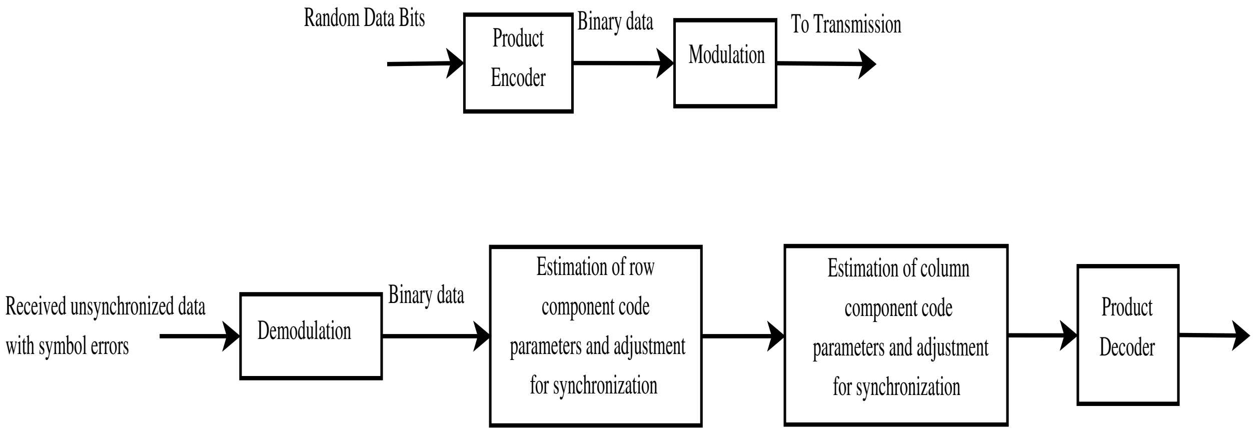


Fig. 2. Generic block diagram for parameter estimation of product codes

of BCH codes. At the receiving end, the row and column component code parameters together with the bit position adjustment parameter are estimated from non-synchronized and erroneous binary coded data symbols. The receiver knows that the incoming data is encoded using RS or BCH product codes without knowing the code parameters.

IV. Parameter estimation of product codes: Synchronized and non-erroneous scenario

The parameter estimation of two-dimensional BCH product codes for non-erroneous and synchronized case has been briefly explained in this section. The notations are given as follows: **Notations:** Let $\hat{m_i}$, $\hat{n_i}$, $\hat{k_i}$, and $\hat{g}_i(x)$ denote the estimate of m_i, n_i, k_i, and $g_i(x)$, respectively.

The incoming product encoded binary symbols are reshaped into a data matrix S of size $a_1{\times}b_1$, where a_1 and b_1 denote the number of rows and columns of S, respectively, $b_1 = 2^{m_1} - 1$, and $m_1 \in \{m_1^{\text{min}} : m_1^{\text{max}}\}$. The rank and rank ratio of S are evaluated using Gauss elimination process. It is to be noted that the Gauss elimination process converts a given matrix into a row or column echelon form by removing all the dependent rows or columns and the number of linearly independent rows/columns gives the rank of a matrix. Rank ratio is the ratio of the rank of a matrix to the number of columns. The corresponding value of m_1 that minimizes the rank ratio

$\rho_1'(m_1)$ is the estimated row component code parameter $\hat{m_1}$. The estimated codeword length of row component code is as follows: $\hat{n_1}=2^{\hat{m_1}}-1$. Finally, the rank value corresponding to $\hat{m_1}$ gives the estimate of code dimension of row component code $\hat{k_1}$. Please note that S is a full rank matrix only if rank$(S)=$min(a_1,b_1) or rank ratio is equal to unity. Further, S is a deficient rank matrix only if rank$(S)<$min(a_1,b_1) or rank ratio is less than unity.

The parameter estimation of column component code assuming synchronized and non-erroneous scenario is given as follows: Since n_1 columns are encoded using column component code C^2 to obtain two-dimensional codeword of size $n_2\times n_1$, the incoming binary data symbols are reshaped into a data matrix of size $a_2\times\hat{n_1}$, where a_2 denote the number of rows of data matrix. The obtained data matrix is reshaped further into another data matrix S' of size $a_2\times b_2$, where $b_2=2^{m_2}-1$ and $m_2\in\{m_2^{\text{min}},m_2^{\text{max}}\}$. The column component code parameters $\hat{n_2}$, $\hat{m_2}$, and $\hat{k_2}$ are obtained similar to row component code parameters.

The rank deficiency and full rank phenomena have been explained using systematic linear block encoding for better clarity. Due to inherent nature of systematic block codes, n coded symbols depend on k uncoded symbols, since $n-k$ parity symbols are generated using different linear combinations of k input symbols [25]. Therefore, $\alpha'\cdot n$ symbols depend on $\alpha'\cdot k$ symbols, where α' is an integer and $\alpha'\geq 1$. It is noted that $\alpha'\cdot k$ data and $\alpha'\cdot(n-k)$ parity symbols of α' codewords will be aligned properly in the same column across all the rows only for the case when b is a multiple of n i.e. $b=\alpha'\cdot n$ [19]. Due to proper alignment, linear relationship will exist between columns in S. This phenomenon will lead to rank deficiency and the deficient rank value is equal to $\alpha'\cdot k$ for the case when $b=\alpha'\cdot n$. It is also observed that the data and parity symbols of α' codewords will not be aligned properly in the same column across all the rows when b is not a multiple of n i.e. $b\neq\alpha'\cdot n$ [19]. Hence, the parity symbols cannot be represented as a linear combination of data symbols and the linear relationship will not be satisfied across all the rows. This will affect the linear relationship between the columns in S and the coded data matrix will resemble like a random matrix without any dependent columns, which will result in full rank.

In a nutshell, if the number of columns b is a multiple of codeword length n, then rank deficiency will be obtained. However, if b is not a multiple of n, then full rank will be obtained. In case of non-erroneous and synchronized scenario, the data and parity symbols of $\alpha' = 1$ codeword in all the rows of data matrix will be aligned properly in the same column for correct combination or value of code parameter(s) resulting in rank deficiency. Moreover, the data and parity symbols will not be aligned properly in the same column for incorrect combinations or values of code parameters resulting in full rank. Hence, the correct combination of code parameters, which minimizes the rank ratio has been chosen as the estimated code parameters of the component codes. In [17] and [19], the rank deficiency and full rank phenomena for linear block and convolutional codes are explained in detail using case studies.

V. Parameter estimation of BCH and RS product codes: Non-synchronized and erroneous scenario

The rank-based method proposed in Section IV fails for noisy scenario due to non-existence of linear relationship between the columns of deficient rank matrix because of transmission errors [18]. When the noise level is above a particular threshold value, then the deficient rank matrix will resemble like a random matrix and full rank will be obtained.

In the case of erroneous channel conditions, it is observed that the dependent columns in column echelon form F corresponding to rank deficient data matrix S will have less number of non-zero elements or more number of zero elements in comparison to independent columns in F. Thus, it is inherent that the column echelon form of rank deficient data matrix will have less number of non-zero elements [19] or more number of zero elements [17] compared to the full rank data matrix. Therefore, the rank deficient and full rank data matrices can be classified based on the number of non-zero or zero elements in F instead of rank for noisy scenario. We propose algorithms for erroneous or noisy scenario and the component code parameters of BCH and RS product codes are estimated based on non-zero-mean-ratio or zero-mean-ratio of column echelon form using Algorithm 1 to 4.

The parameter estimation of row and column component codes of two-dimensional BCH product codes considering non-synchronized and erroneous case is carried out using Algorithm 1 and 2, respectively. The notations that are common to Algorithm 1 and 2 are given as follows: **Notations:** F_j and F'_j denote the column echelon forms of data matrices S_j and S'_j, respectively, j denotes the iteration number, N refers to the number of iterations, $\omega_j(c, m_i, \phi_i)$ denotes the ratio of zero elements i.e. zero-ratio in c^{th} column of column echelon form considering i^{th} component code, where $i = 1$ and 2, $data(\cdot)$ refers to an array of incoming BCH product coded binary data symbols, and $data\_shift(\cdot)$ refers to the array of incoming coded binary symbols shifted by ϕ_i bit positions. Further, the zero-mean-ratio of i^{th} component code is denoted as $\mu(m_i, \phi_i)$. Finally, ϕ_i denotes the bit position adjustment parameter to achieve frame synchronization in i^{th} dimension.

In Algorithm 1, the incoming BCH product encoded binary data symbols are shifted by ϕ_1 bit positions, where $\phi_1 \in [0\ ,\ (2^{m_1} - 1) - 1]$ denotes the bit position adjustment parameter to achieve frame synchronization of row component code, and by simultaneously varying $m_1 \in [m_1^{\text{min}}\ ,\ m_1^{\text{max}}]$. The shifted data symbols are reshaped into a data matrix of size $a_1 \times b_1$, where $b_1 = 2^{m_1} - 1$ and the reshaped data matrix is converted into column echelon form using Gauss elimination process. The zero-mean-ratio values of column echelon form (i.e. $\mu(m_1, \phi_1)$) are calculated instead of rank ratio. To evaluate $\mu(m_1, \phi_1)$, the ratio of zero elements i.e. zero-ratio in each column of column echelon form F_j (i.e. $\omega_j(c, m_1, \phi_1)$, where $c \in \{1, 2, ..., b\}$) is evaluated and the same is repeated for N iterations. The calculated values across all N iterations are accumulated into a matrix A of size $N \times b$. Now the mean of A is evaluated and the resultant row vector of size $1 \times b_1$ is denoted by B, where $B = [\gamma(1, m_1, \phi_1)\ :\ \gamma(b_1, m_1, \phi_1)]$ and $\gamma(c, m_1, \phi_1) = \frac{\sum_{j=1}^{N} \omega_j(c, m_1, \phi_1)}{N}$. The zero-mean-ratio is calculated as follows: $\mu(m_1, \phi_1) =$

Algorithm 1: Estimation of BCH product code parameters considering erroneous and non-synchronized scenario (row component code)

Assumptions: $a_i \geq t\, b_i$, where t is a constant, $m_i \in [m_i^{\text{min}}, m_i^{\text{max}}]$, $\phi_1 \in [0\,, (2^{m_1}-1)-1]$, and the erroneous incoming bit stream is assumed to be BCH-based product codes

Input: BCH product encoded binary data symbols;

Output: $\hat{n_1}$, $\hat{m_1}$, $\hat{k_1}$, and $\hat{\phi_1}$;

1: **for** $m_1 = m_1^{min} : m_1^{max}$ **do**

  2:**for** $\phi_1 = 0 : (2^{m_1}-1)-1$ **do**

    3: Shift the erroneous incoming binary data symbols $data(\cdot)$ by ϕ_1 bit positions;

    4: **for** $j = 1 : N$ **do**

      5: $R_j = data\_shift(1+(j-1)b_1 \,:\, b_1 \cdot a_1 + (j-1)b_1)$;

      6: Reshape R_j into a matrix S_j of size $a_1 \times b_1$, where $b_1 = 2^{m_1}-1$;

      7: Convert S_j into F_j using Gauss elimination process;

      8: Compute $\omega_j(c, m_1, \phi_1)$ in each column of F_j, where $c \in \{1, 2, ..., b_1\}$;

      9: Form a row matrix $A_j = [\omega_j(1, m_1, \phi_1) \,:\, \omega_j(b_1, m_1, \phi_1)]$;

    end

    10: Accumulate all the row matrices into a single matrix A of size $N \times b_1$, where $A = [A_1\,;\, A_2\,;\, A_3\,;\, \cdots A_N]$;

    11: Compute $B = mean(A)$, where $B = [\gamma(1, m_1, \phi_1) \,:\, \gamma(b_1, m_1, \phi_1)]$ and $\gamma(c, m_1, \phi_1) = \frac{\sum_{j=1}^{N} \omega_j(c, m_1, \phi_1)}{N}$;

    12: Calculate $\mu(m_1, \phi_1)$, where $\mu(m_1, \phi_1) = \frac{\sum_{c=1}^{b_1} \gamma(c, m_1, \phi_1)}{b_1}$;

  end

end

13: Obtain $[\hat{m_1}, \hat{\phi_1}] = \underset{m_1, \phi_1}{\text{argmax}}(\mu(m_1, \phi_1))$ and $\hat{n_1} = 2^{\hat{m_1}}-1$;

14: Shift the binary data symbols by $\hat{\phi_1}$ bit positions;

15: After synchronization, obtain the possible roots of generator polynomial from code polynomials through maximum likelihood approach and identify $\hat{g_1}(x)$ from (1);

16: The highest degree of $\hat{g_1}(x)$ gives the estimate of $n_1 - k_1$;

17: Obtain $\hat{k_1}$ from $n_1 - k_1$;

$\frac{\sum_{c=1}^{b_1}\gamma(c,m_1,\phi_1)}{b_1}$.

If c^{th} column of the data matrix is dependent and if the same is not an all-zero column due to erroneous bits, then it is intuitive that the ratio of zero elements in c^{th} column of column echelon form will be higher compared to the independent columns. Hence, zero-mean-ratio $\mu(m_1,\phi_1)$ of column echelon form will be higher for rank deficient data matrix compared to full rank data matrix. Therefore, the row component code parameter $\hat{m_1}$ and synchronization parameter $\hat{\phi_1}$ are identified by observing the corresponding combination of $[m_1,\phi_1]$ which maximizes $\mu(m_1,\phi_1)$. From m_1, the codeword length of row component code n_1 is estimated as mentioned in step 13 of Algorithm 1. The maximum likelihood approach proposed in our work to estimate the roots of the generator polynomial is explained in detail as follows: After shifting $\hat{\phi_1}$ bit positions, the received M data bits are converted into $X = M/\hat{n_1}$ codewords or code polynomials. It is obvious that X is linearly increasing with M. Since all these code polynomials are independent of each other, their occurrence is equally likely. Therefore, instead of selecting the entire data sample (where M and X are very large), we limit our search to a finite set of code polynomials occurring from the first K data symbols, where K is sufficiently high and less than M. The roots of the finite set of code polynomials are evaluated from the selected set. As the code polynomials share the roots of generator polynomial, the most probable $2t$ roots of the generator polynomial $g_1(x)$ (i.e. α, α^2,...,α^{2t}) are identified using a maximum likelihood approach. The generator polynomial for BCH code is given by lcm$(\phi_1'(x),\phi_2'(x),...,\phi_{2t}'(x))$, where $\phi_i'(x)$ is the minimal polynomial of α^i (refer to (1)). After identifying the roots and minimal polynomials corresponding to the roots, the generator polynomial $g_1(x)$ is recognized. Since the generator polynomial for BCH code is a $n_1 - k_1$ degree polynomial and $\hat{n_1}$ is already known, $\hat{k_1}$ is estimated from $n_1 - k_1$.

To estimate the column component code parameters of BCH product codes using Algorithm 2, shift the incoming binary data symbols by ϕ_2 bit positions, where ϕ_2 denotes the bit position adjustment parameter to achieve frame synchronization of vertical component code, and also

Algorithm 2: Estimation of BCH product code parameters considering erroneous and non-synchronized scenario (column component code)

Input: BCH product encoded binary data symbols, $\hat{n_1}$, $\hat{m_1}$, and $\hat{\phi_1}$;

Output: $\hat{n_2}$, $\hat{m_2}$, $\hat{k_2}$, and $\hat{\phi_2}$;

1: $\phi = \hat{\phi_1} + (\hat{n_1} \cdot kk)$, where $kk > 1$;

2: **for** $m_2 = m_2^{min} : m_2^{max}$ **do**

3: $\phi_2 = [\phi_2 \;\; \phi]$, accumulate ϕ_2 until $\phi_2 > \hat{n_1} \cdot (2^{m_2} - 1)$;

4: Shift the erroneous incoming binary data symbols $data(\cdot)$ by ϕ_2 bit positions;

5: **for** $j = 1 : N$ **do**

6: Store the shifted data symbols in a row matrix R_j;

7: Reshape R_j into a data matrix of size $a_2 \times \hat{n_1}$;

8: Transpose the data matrix and reshape again into another data matrix S'_j of size $a_2 \times b_2$, where $b_2 = 2^{m_2} - 1$;

9: Convert S'_j into F'_j using Gauss elimination process;

10: Compute $\omega_j(c, m_2, \phi_2)$ in each column of F'_j, where $c \in \{1, 2, ..., b_2\}$;

11: Form a row matrix $A'_j = [\omega_j(1, m_2, \phi_2) \; : \; \omega_j(b_2, m_2, \phi_2)]$;

end

12: Accumulate all the row matrices into a single matrix A' of size $N \times b_2$, where $A' = [A'_1 \; ; \; A'_2 \; ; \; A'_3 \; ; \; \cdots A'_N]$;

13: Compute $B' = mean(A')$, where $B' = [\gamma(1, m_2, \phi_2) \; : \; \gamma(b_2, m_2, \phi_2)]$ and $\gamma(c, m_2, \phi_2) = \frac{\sum_{j=1}^{N} \omega_j(c, m_2, \phi_2)}{N}$;

14: Calculate $\mu(m_2, \phi_2)$, where $\mu(m_2, \phi_2) = \frac{\sum_{c=1}^{b_2} \gamma(c, m_2, \phi_2)}{b_2}$;

end

15: Obtain $[\hat{m_2}, \hat{\phi_2}] = \underset{m_2, \phi_2}{\text{argmax}}(\mu(m_2, \phi_2))$ and $\hat{n_2} = 2^{\hat{m_2}} - 1$;

16: Shift the binary data symbols by $\hat{\phi_2}$ bit positions and obtain the possible roots of generator polynomial from code polynomials and identify $\hat{g_2}(x)$ from (1);

17: The highest degree of $\hat{g_2}(x)$ gives the estimate of $n_2 - k_2$;

18: Obtain $\hat{k_2}$ from $n_2 - k_2$;

by simultaneously varying m_2 between m_2^{min} to m_2^{max}. Similar to non-erroneous scenario, the incoming binary data symbols are reshaped into a data matrix of size $a_2 \times \hat{n}_1$ and the resultant data matrix is reshaped further into another data matrix of size $a_2 \times b_2$, where $b_2 = 2^{m_2} - 1$. Rest of the steps for estimating the column component code parameters m_2^{est}, n_2^{est}, and synchronization parameter $\hat{\phi}_2$ are similar to Algorithm 1. After shifting $\hat{\phi}_2$ bit positions, the code dimension $\hat{k}_2$ and generator polynomial $\hat{g}_2(x)$ of column component code are identified similar to Algorithm 1 using maximum likelihood approach as mentioned in steps 16 to 18 of Algorithm 2.

For instance, if the receiver starts receiving the product coded data symbols at Δ^{th} bit position of χ'^{th} BCH code (row component code), then time synchronization can be achieved by shifting $\hat{\phi}_1 = ((\beta \cdot \chi' \cdot n_1) - \Delta) + 1$ bit positions, where β is a positive integer and $n_1 = 2^{m_1} - 1$. This is because, a new row component BCH codeword will start at every $(\beta \cdot \chi' \cdot n_1) + 1$ bit positions. Therefore, the maximum range of bit position adjustment parameter can be restricted to $(2^{m_1} - 1) - 1$ as given in Algorithm 1. It is to be noted that the column component code should also be synchronized similar to row component code in order to complete the time synchronization process. If the receiver starts receiving the product coded bits at Δ^{th} bit position of χ^{th} product code, then time synchronization can be achieved by shifting $\hat{\phi}_2 = ((\beta \cdot \chi \cdot n_1 \cdot n_2) - \Delta) + 1$ bit positions, where $n_2 = 2^{m_2} - 1$. This is because, a new BCH product codeword will start at every $(\beta \cdot \chi \cdot n_1 \cdot n_2) + 1$ bit positions. To synchronize column component code, it is not necessary to search the bit position adjustment parameter within the range of 0 to $(\hat{n}_1 \cdot (2^{m_2} - 1)) - 1$. The search space can be restricted as given in steps 1 and 2 of Algorithm 2. As row component code is already synchronized, it is essential only to accumulate the starting bit positions of row component BCH codeword in an array, which is denoted by ϕ_2, until $\phi_2 > \hat{n}_1 \cdot (2^{m_2} - 1)$. Observe the corresponding combination of $[m_2, \phi_2]$ that maximizes zero-mean-ratio $\mu(m_2, \phi_2)$ as given in step 15 of Algorithm 2.

The parameter estimation of two-dimensional RS product codes considering non-synchronized and erroneous case is explained using Algorithms 3 and 4. It is a modified version of an algorithm

Algorithm 3: Estimation of RS product code parameters considering erroneous and non-synchronized scenario (row component code)

Assumptions: $m_1 = m_2$, $a_1 \geq t\, b_1$, where t is a constant, $m_1 \in [m_1^{\text{min}}, m_1^{\text{max}}]$, $\phi_1 \in [0\,,\,((2^{m_1}-1)m_1)-1]$, and the erroneous incoming bit stream is assumed to be RS-based product codes

Input: RS product encoded data symbols **Output:** $\hat{n_1}$, $\hat{m_1}$, $\hat{p_1}$, $\hat{k_1}$, and $\hat{\phi_1}$;

1: **for** $m_1 = m_1^{min} : m_1^{max}$ **do**

2: $p_1 = \text{primpoly}(m_1, 'all')$;

3: **for** $\phi_1 = 0 : ((2^{m_1}-1)m_1) - 1$ **do**

4: Shift the erroneous incoming coded binary data symbols $data(\cdot)$ by ϕ_1 bit positions and convert it into the respective elements of GF using p_1;

5: **for** $j = 1 : N$ **do**

6: $R_j = data\_shift(1 + (j-1)b_1 \,:\, b_1 \cdot a_1 + (j-1)b_1)$;

7: Reshape the RS encoded GF array elements R_j into a data matrix S_j of size $a_1 \times b_1$, where $b_1 = 2^{m_1} - 1$;

8: Convert S_j into F_j using finite-field Gauss elimination process;

9: Compute $\Omega_j(c, m_1, p_1, \phi_1)$ in each column of F_j, where $c \in \{1, 2, ..., b_1\}$;

10: Form a row matrix $A_j = [\Omega_j(1, m_1, p_1, \phi_1) \,:\, \Omega_j(b_1, m_1, p_1, \phi_1)]$;

end

11: Accumulate all the row matrices into a single matrix A of size $N \times b_1$, where $A = [A_1\,;\,A_2\,;\,A_3\,;\,\cdots A_N]$;

12: Compute $B = mean(A)$, where $B = [\gamma(1, m_1, p_1, \phi_1) \,:\, \gamma(b_1, m_1, p_1, \phi_1)]$ and $\gamma(c, m_1, p_1, \phi_1) = \frac{\sum_{j=1}^{N} \Omega_j(c, m_1, p_1, \phi_1)}{N}$;

13: Normalize B with respect to the maximum value, where $B_{\text{norm}} = [\gamma'(1, m_1, p_1, \phi_1) \,:\, \gamma'(b_1, m_1, p_1, \phi_1)]$ and $\gamma'(c, m_1, p_1, \phi_1) = \frac{\gamma(c, m_1, p_1, \phi_1)}{\max(B)}$;

14: Calculate $\mu'(m_1, p_1, \phi_1)$, where $\mu'(m_1, p_1, \phi_1) = \frac{\sum_{c=1}^{b_1} \gamma'(c, m_1, p_1, \phi_1)}{b_1}$;

end

end

15: Obtain $[\hat{m_1}, \hat{p_1}, \hat{\phi_1}] = \underset{m_1, p_1, \phi_1}{\text{argmin}}\,(\mu'(m_1, p_1, \phi_1))$ and $\hat{n_1} = 2^{\hat{m_1}} - 1$;

16: After shifting $\hat{\phi_1}$ bit positions, convert the binary data symbols into the respective elements of GF using $\hat{p_1}$. Identify the number of roots of generator polynomial from code polynomials using maximum likelihood approach and estimate $n_1 - k_1$;

17: Obtain $\hat{k_1}$ from $n_1 - k_1$ and obtain the generator polynomial $\hat{g_1}(x)$ from (2);

proposed by us in [19] for blind reconstruction of RS encoder. Please note that the usefulness of algorithm in [19] is limited to direct RS codes alone, not to the RS product codes. In the current work, the algorithm in [19] has been modified to estimate RS product code parameters with reduced complexity and proposed a novel approach to synchronize both the row and column components of RS codes. Other major differences between the proposed algorithm for RS product codes and the algorithm in [19] are discussed in the following paragraphs.

In [19], the finite-field Gauss elimination process is applied on a data matrix of size $a \times b$, where $a = 5000 \times b$ without any iteration. In the current work, the finite-field Gauss elimination process is applied on a data matrix of size $a \times b$, where $a = 50 \times b$ with number of iterations $N = 5$. By increasing N, we perform permutations on the rows of the data matrix to obtain a new virtual realization of the received data stream without increasing the data size unlike [19].

In [19], the mean value of number of non-zero elements in each column of column echelon form F is calculated and the same is averaged over the number of columns b for obtaining the normalized non-zero-mean-ratio values. However, in the current work, the mean value of number of non-zero elements in each column of column echelon form F_j, where $j = 1$ to N, is calculated and the same is repeated for N iterations to form a matrix of size $N \times b$. The mean value of each column vector of the matrix is evaluated to form a row vector of size $1 \times b$. Finally, the obtained row vector is averaged over the number of columns b to obtain the normalized non-zero-mean-ratio values

The algorithm in [19] is restricted to estimate the parameters of RS codes and bit position adjustment parameter to achieve synchronization. In the current work, the algorithms are proposed to identify the RS product code parameters (i.e. row and column component RS code parameters). In addition, a novel approach to synchronize both the row and column component product codes has been proposed and integrated with the code parameter estimation algorithms. Precisely, it is required only to accumulate the starting bit positions of row component RS codeword in an array instead of searching over all the bit position to synchronize column component RS code (given in Algorithm 4). Thus, the proposed synchronization approach in Algorithm 4 is completely different from the approach carried out in Algorithm 3 and in our previous work [19].

The following paragraphs discuss more details of the proposed algorithms for the parameter estimation of two-dimensional RS product codes. The notations that are common to Algorithms 3 and 4 are given as follows:

Notations: Let $\hat{p}_i$ denotes the estimate of integer representation of primitive polynomial p_i. $\Omega_j(c, m_1, p_1, \phi_1)$ denotes the non-zero-ratio in c^{th} column of column echelon form considering row component code, $data\_shift(\cdot)$ refers to an array of RS product coded non-binary data symbols shifted by ϕ_i bit positions, normalized non-zero-mean-ratio of row component code is denoted as $\mu'(m_1, p_1, \phi_1)$, and ϕ_i denotes the bit position adjustment parameter to achieve frame synchronization in i^{th} dimension. Similarly, $\Omega_j(c, \phi_2)$ denotes the non-zero-ratio in c^{th} column of column echelon form considering column component code and normalized non-zero-mean-ratio of column component code is denoted as $\mu'(\phi_2)$.

In Algorithm 3, the steps for estimating the row component code parameters are given. The incoming RS product encoded data symbols are shifted by ϕ_1 bit positions, where $\phi_1 \in [0, ((2^{m_1} - 1)m_1) - 1]$, and by simultaneously varying $m_1 \in [m_1^{\text{min}}, m_1^{\text{max}}]$ and primitive polynomial p_1 corresponding to m_1. After shifting ϕ_1 bit positions, the binary data symbols are converted into a non-binary symbols such that each symbol is an integer between 0 and $2^{m_1} - 1$ and a GF array is created from the non-binary data symbols using primitive polynomial p_1. It is to be noted that $p_1 = \text{primpoly}(m_1, `all')$ given in step 2 of Algorithm 3 returns all primitive polynomials $p_1(x)$ corresponding to m_1 in an integer form, where $m_1 \geq 3$. The GF array elements are reshaped into a data matrix of size $a_1 \times b_1$, where $b_1 = 2^{m_1} - 1$. The data matrix is converted into column echelon form using finite field Gauss elimination process [25] instead of Gauss elimination process, since the data matrix contains elements from finite field. The normalized non-zero-mean-ratio, which is denoted by $\mu'(m_1, p_1, \phi_1)$, of column echelon form F_j is calculated. In order to calculate the same, the ratio of non-zero elements i.e. non-zero-ratio, which is denoted by $\Omega_j(c, m_1, p_1, \phi_1)$, in each column of F_j is evaluated and it is repeated for N iterations. Accumulate the values across all N iterations in a matrix A of size $N \times b_1$ and the mean of A is evaluated. The resultant row vector of size $1 \times b_1$ is denoted by B, where $B = [\gamma(1, m_1, p_1, \phi_1) \ : \ \gamma(b_1, m_1, p_1, \phi_1)]$ and $\gamma(c, m_1, p_1, \phi_1) = \frac{\sum_{j=1}^{N} \Omega_j(c, m_1, p_1, \phi_1)}{N}$. The row vector B is normalized with respect to the maximum value and the resultant vector is given by $B_{\text{norm}} = [\gamma'(1, m_1, p_1, \phi_1) : \gamma'(b_1, m_1, p_1, \phi_1)]$, where $\gamma'(c, m_1, p_1, \phi_1) = \frac{\gamma(c, m_1, p_1, \phi_1)}{\max(B)}$. From $\gamma'(c, m_1, p_1, \phi_1)$, the normalized non-zero-mean-ratio, which is the ratio of the sum of normalized non-zero-ratio in each column averaged across all N iterations to the total number of columns of F_j, is calculated as follows: $\mu'(m_1, p_1, \phi_1) = \frac{\sum_{c=1}^{b_1} \gamma'(c, m_1, p_1, \phi_1)}{b_1}$.

It is to be noted that the ratio of non-zero elements in c^{th} dependent column of column echelon form will be lesser compared to the independent columns. Thus, the normalized non-zero-mean-ratio $\mu'(m_1, p_1, \phi_1)$ of column echelon form of deficient rank matrix will be lesser compared to the full rank matrix for erroneous channel conditions [19]. Therefore, the row component code parameters $\hat{m_1}$ and $\hat{p_1}$ along with synchronization parameter $\hat{\phi_1}$ are identified by observing the corresponding combination of $[m_1, p_1, \phi_1]$ which minimizes $\mu'(m_1, p_1, \phi_1)$ as shown in step 15 of Algorithm 3. It is a known fact that the code polynomials share the roots of generator polynomial. Therefore, to identify the generator polynomial and code dimension of row component code, the number of roots of code polynomials is evaluated. This is performed after shifting $\hat{\phi_1}$ bit positions and converting the incoming binary data symbols into respective elements of GF using $\hat{p_1}$. It has been observed that the number of roots of code polynomials are equal to the generator polynomial in most of the cases though there exists few cases with different number of roots due to erroneous symbols. Thus, a maximum likelihood approach is adopted to evaluate the number of roots of the generator polynomial. Note that by identifying the number of roots, $2t$ can be estimated and it is known that for RS codes $n - k = 2t$. Since $\hat{n_1}$ is already identified from step 15, code dimension of row component code $\hat{k_1}$ is estimated from $n - k$. After estimating the number of roots of code polynomials, the generator polynomial of row component code $\hat{g_1}(x)$ is recognized using (2).

In Algorithm 4, the steps for estimating the parameters of column component code are given. As already mentioned, it has been assumed that both row and column component codes are generated using same primitive polynomial and hence, $\hat{m_2} = \hat{m_1}$ and $\hat{p_2} = \hat{p_1}$ as given in step 1 of Algorithm 4. To identify the code dimension and synchronization parameter, the incoming binary data symbols are shifted by ϕ_2 bit positions and converted into non-binary symbols such that each symbol is an integer between 0 and $\hat{n_2}$. A GF array is created from the non-binary data symbols using primitive polynomial $\hat{p_2}$. Similar to non-erroneous scenario, the GF array elements are reshaped into a data matrix of size $a_2 \times \hat{n_1}$ and the resultant data matrix is reshaped further into another data matrix of size $a_2 \times \hat{n_2}$. After reshaping, the resultant data matrix is converted into column echelon form using finite field Gauss elimination process. The normalized non-zero-mean-ratio, which is denoted by $\mu'(\phi_2)$, of the column echelon form is evaluated similar to Algorithm 3. The corresponding value of ϕ_2 for which $\mu'(\phi_2)$ is minimum gives the estimate of the synchronization parameter i.e. $\hat{\phi_2}$ as shown in step 15 of Algorithm 4. After shifting $\hat{\phi_2}$ bit positions, a GF array is created from synchronized binary data symbols using $\hat{p_2}$. The code dimension $\hat{k_2}$ and generator polynomial $\hat{g_2}(x)$ of column component code are identified similar to Algorithm 3 using maximum likelihood approach mentioned in steps 17 and 18 of Algorithm 4.

If the receiver starts receiving at Δ'^{th} symbol position of χ'^{th} RS code (row component code), then time synchronization can be achieved by shifting $\hat{\phi_1} = ((\beta \cdot \chi' \cdot m_1 \cdot n_1 + 1) - ((\Delta' - 1) \cdot m_1 + 1))$ bit positions, where $n_1 = 2^{m_1} - 1$. This is because, a new row component RS codeword will start at every $(\beta \cdot \chi' \cdot m_1 \cdot n_1) + 1$ bit positions. Hence, the maximum range of bit position adjustment parameter is restricted to $(n_1 \cdot m_1) - 1$ as shown in Algorithm 3. It is also essential to synchronize column component code in order to complete the time synchronization process. Since a new RS product codeword will start at every $(\beta \cdot \chi \cdot n_1 \cdot n_2 \cdot m_1) + 1$ bit positions, the time synchronization is achieved by shifting $\hat{\phi_2} = ((\beta \cdot \chi \cdot n_1 \cdot n_2 \cdot m_1 + 1) - ((\Delta' - 1) \cdot m_1 + 1))$ bit positions provided the

Algorithm 4: Estimation of RS product code parameters considering erroneous and non-synchronized scenario (column component code)

Assumptions: $a_2 > t \cdot \hat{n_i}$, where $i = 1$ and 2;

Input: RS product encoded data symbols, $\hat{n_1}$, $\hat{m_1}$, and $\hat{\phi_1}$;

Output: $\hat{k_2}$ and $\hat{\phi_2}$;

1: $\hat{m_2} = \hat{m_1}$, $\hat{n_2} = 2^{\hat{m_2}} - 1$, and $\hat{p_2} = \hat{p_1}$;

2: $\phi = \hat{\phi_1} + (\hat{n_1} \cdot \hat{m_1} \cdot kk)$, where $kk > 1$;

3: $\phi_2 = [\phi_2 \;\; \phi]$, accumulate ϕ_2 until $\phi_2 > (\hat{n_1} \cdot \hat{n_2} \cdot \hat{m_1})$;

4: Shift the erroneous incoming coded binary data symbols by ϕ_2 bit positions and convert it into the respective elements of GF using $\hat{p_2}$;

5: **for** $j = 1 : N$ **do**

6: Store the elements of GF (i.e. non-binary symbols) in a row matrix R_j;

7: Reshape the RS encoded GF array elements R_j into a data matrix of size $a_2 \times \hat{n_1}$ and transpose the data matrix and reshape again into another matrix S'_j of size $a_2 \times \hat{n_2}$;

8: Convert S'_j into F'_j using finite-field Gauss elimination process;

9: Compute $\Omega_j(c, \phi_2)$ in each column of F'_j, where $c \in \{1, 2, ..., b_2\}$;

10: Form a row matrix $A'_j = [\Omega_j(1, \phi_2) \; : \; \Omega_j(b_2, \phi_2)]$;

end

11: Accumulate all the row matrices into a single matrix A' of size $N \times b_2$, where $A' = [A'_1 \,;\, A'_2 \,;\, A'_3 \,;\, \cdots A'_N]$;

12: Compute $B' = mean(A')$, where $B' = [\gamma(1, \phi_2) \; : \; \gamma(b_2, \phi_2)]$ and $\gamma(c, \phi_2) = \frac{\sum_{j=1}^{N} \Omega_j(c, \phi_2)}{N}$;

13: Normalize B' with respect to the maximum value, where $B'_{\text{norm}} = [\gamma'(1, \phi_2) \; : \; \gamma'(b_2, \phi_2)]$ and $\gamma'(c, \phi_2) = \frac{\gamma(c, \phi_2)}{\max(B')}$;

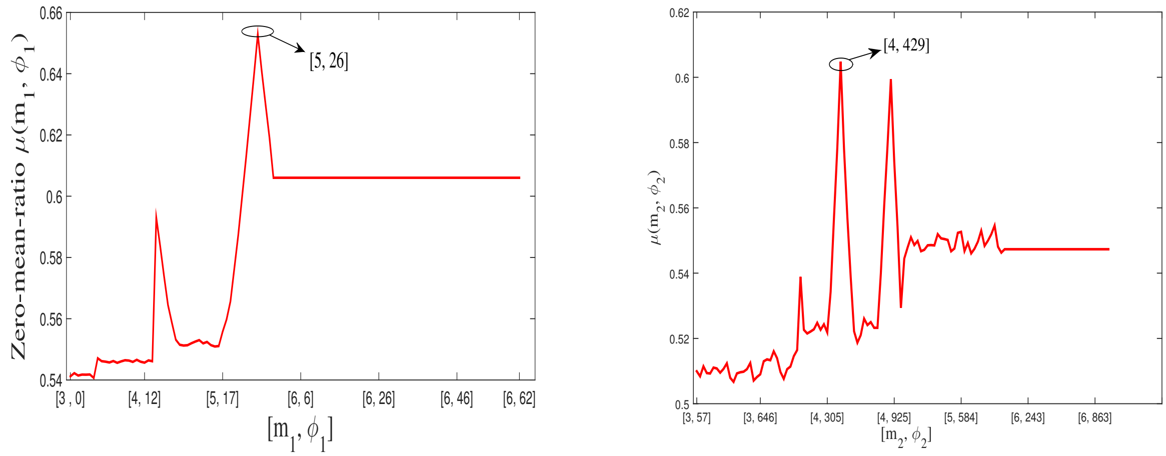
14: Calculate $\mu'(\phi_2)$, where $\mu'(\phi_2) = \frac{\sum_{c=1}^{b_2} \gamma'(c, \phi_2)}{b_2}$;

15: Obtain $[\hat{\phi_2}] = \underset{\phi_2}{\text{argmin}}(\mu'(\phi_2))$;

16: After shifting $\hat{\phi_2}$ bit positions, convert the binary data symbols into the respective elements of GF using $\hat{p_2}$;

17: Identify the number of roots of generator polynomial from code polynomials and estimate $n_2 - k_2$;

18: Obtain $\hat{k_2}$ from $n_2 - k_2$ and obtain the generator polynomial $\hat{g_2}(x)$ from (2);



(a) (b)

Fig. 3. (a) Variation of zero-mean-ratio $\mu(m_1, \phi_1)$ with respect to $[m_1, \phi_1]$ for product code BCH$(31 \times 15, 21 \times 11, 5 \times 4)$ assuming 16-QAM scheme, delay $\Delta = 37$ bit positions, and SER=10^{-2} (b) Variation of zero-mean-ratio $\mu(m_2, \phi_2)$ with respect to $[m_2, \phi_2]$ for product code BCH$(31{\times}15, 21{\times}11, 5{\times}4)$ assuming 16-QAM scheme, delay $\Delta = 37$ bit positions, and SER=10^{-2}

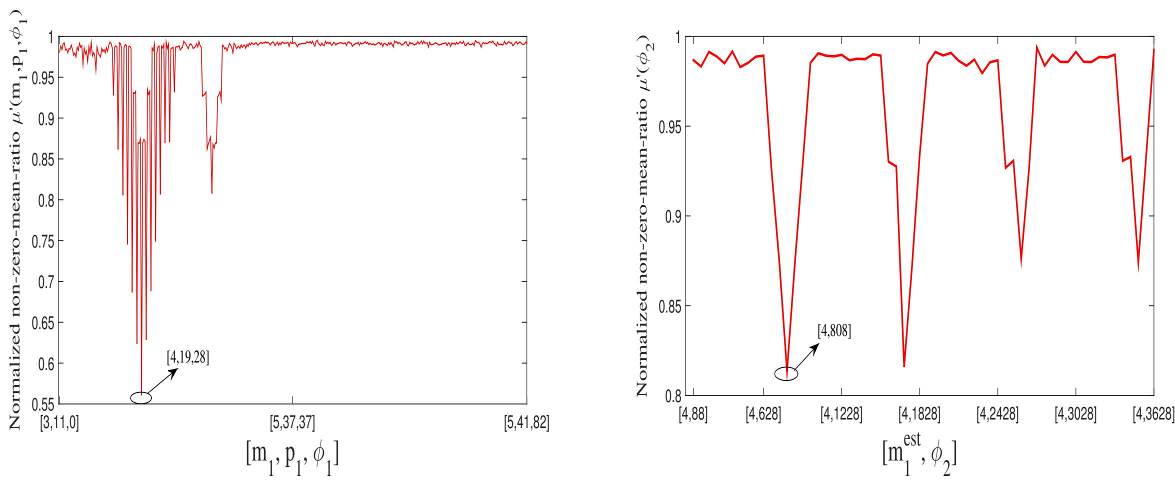
receiver starts receiving at Δ'^{th} symbol position of χ^{th} RS product code. To synchronize column component code, it is required only to accumulate the starting bit positions of row component RS codeword in an array, which is denoted by ϕ_2, until $\phi_2 > (\hat{n_1} \cdot \hat{n_2} \cdot \hat{m_1})$ instead of searching over all the bit position within the range of 0 to $(\hat{n_1} \cdot \hat{n_2} \cdot \hat{m_1}) - 1$. Finally, observe the corresponding value of $[\phi_2]$ that minimizes non-zero-mean-ratio $\mu'(\phi_2)$ as given in step 15 of Algorithm 4.

VI. Simulation Results and Discussions

Since two-dimensional product codes are constructed using component codes that have short codeword lengths, we have assumed short codeword lengths for row and column component RS and BCH codes in our simulation study. It is to be noted that except Fig. 5(a), all the simulation results are obtained by assuming $M-$ary symmetric channel model, which is defined as the generalization of binary symmetric channel (BSC) model. It comprises of non-binary input and output symbols belonging to the M-ary signal space.

The simulation results for product code BCH$(31\times15, 21\times11, 5\times4)$ over erroneous scenario assuming symbol error rate (SER) = 10^{-2}, delay $\Delta=37$ bit positions, and 16-QAM constellation are shown in Fig. 3(a) and Fig. 3(b). Algorithm 1 is executed and the variation of zero-mean-ratio $\mu(m_1, \phi_1)$ is shown in Fig. 3(a). From the plot, it can be inferred that at $m_1 = 5$ and $\phi_1=26$, $\mu(m_1, \phi_1)$ reaches maximum compared to other values of $[m_1, \phi_1]$. Hence, $\hat{m_1}=5$ and $\hat{n_1}=2^{\hat{m_1}}-1=31$ are estimated successfully. Further, synchronization of row component code can be achieved by shifting $\hat{\phi_1}=26$ bit positions. This is because, a new row component BCH codeword will start after shifting $\hat{\phi_1}=((\beta\cdot\chi'\cdot n_1)-\Delta)+1$ bit positions, where β is a positive integer, provided the receiver starts receiving the coded bits at Δ^{th} bit position of χ'^{th} BCH code. Here, in this case $\beta=1$ and $\chi'=2$. After shifting $\hat{\phi_1}$ bit positions, the roots of code polynomials are evaluated and it has been identified that $[\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^6, \alpha^8, \alpha^{12}, \alpha^{16}, \alpha^{17}, \alpha^{24}]$ are the most likelihood roots of the generator polynomial $g_1(x)$. The generator polynomial for BCH code is given by lcm$(\phi'_1(x), \phi'_2(x), ..., \phi'_{2t}(x))$, where $\phi'_i(x)$ is the minimal polynomial of α^i. It is known that $[\alpha^3, \alpha^6, \alpha^{12}, \alpha^{17}, \alpha^{24}]$ are the conjugates of α^3 and $[\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}]$ are the conjugates of α and it is also known that the finite field elements and its conjugates have the same minimal polynomials. Therefore, the estimated generator polynomial of row component code is given by $\hat{g_1}(x)=\phi'_1(x)\cdot\phi'_3(x)$, where $\phi'_1(x)=1+x^2+x^5$ and $\phi'_3(x)=1+x^2+x^3+x^4+x^5$ in GF(2^5) [40]. Note that generator polynomial for BCH code is a n_1-k_1 degree polynomial and it has been identified as $n_1-k_1=10$. Since $\hat{n_1}$ is already known, $\hat{k_1}=21$.

The proposed algorithm to extract column component BCH code parameters (i.e. Algorithm 2) is applied on product code BCH$(31\times15, 21\times11, 5\times4)$ assuming SER=10^{-2}, delay $\Delta=37$ bit positions, and 16-QAM constellation. The variation of zero-mean-ratio $\mu(m_2, \phi_2)$ with respect to $[m_2, \phi_2]$ is shown in Fig. 3(b). It is observed from the figure that at $m_2=4$ and $\phi_2=429$, $\mu(m_2, \phi_2)$ is maximum. Therefore, $\hat{m_2}=4$ and $\hat{n_2}=2^{\hat{m_1}}-1=15$ are estimated successfully. In addition, the synchronization of column component code is obtained by moving 429 bit positions. This is because, a new product codeword will start after shifting $\hat{\phi_2}=((\beta\cdot\chi\cdot n_1\cdot n_2)-\Delta)+1$ bit positions,



(a)

(b)

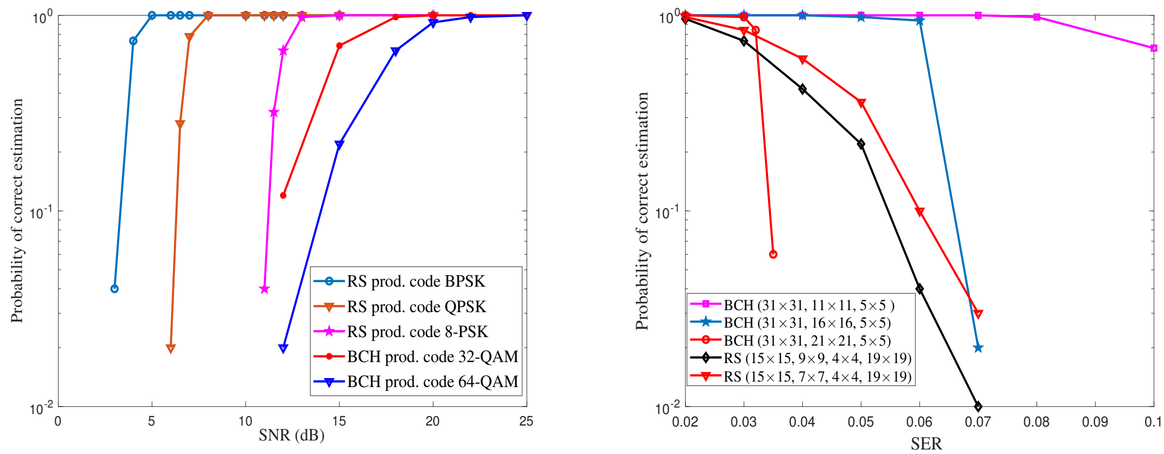
Fig. 4. (a) Variation of normalized non-zero-mean-ratio $\mu'(m_1, p_1, \phi_1)$ with respect to $[m_1, p_1, \phi_1]$ for irregular non-binary product code RS$(15{\times}15, 7{\times}11, 4{\times}4)$ assuming 16-QAM scheme, delay $\Delta' = 24$ symbol positions, and SER=10^{-2} (b) Variation of normalized non-zero-mean-ratio $\mu'(\phi_2)$ with respect to $[\phi_2]$ for irregular non-binary product code RS$(15\times15, 7\times11, 4\times4)$ assuming 16-QAM scheme, delay $\Delta' = 24$ symbol positions, and SER=10^{-2}

where $\beta = \chi = 1$, provided the receiver starts receiving the product coded bits at Δ^{th} bit position of χ^{th} product code. After shifting $\hat{\phi}_2$ bit positions, the roots of code polynomials are evaluated to identify generator polynomial $g_2(x)$ and code dimension k_2 of column component code. It has been observed that $[\alpha, \alpha^2, \alpha^4, \alpha^8]$ are the most likelihood roots of the generator polynomial $g_2(x)$. Since all the roots are the conjugates of α, the estimated generator polynomial of column component code is given by $\hat{g}_2(x) = \phi'_1(x)$, where $\phi'_1(x) = 1 + x + x^4$ in GF(2^4) [40]. It is to be noted that $g_2(x)$ is a $n_2 - k_2 = 4$ degree polynomial. As $\hat{n}_2$ is already known, $\hat{k}_2 = 11$.

In Fig. 4(a) and Fig. 4(b), the simulation results are shown for two-dimensional irregular non-binary product code RS$(15 \times 15, 7 \times 11, 4 \times 4)$ assuming 16-QAM scheme, delay $\Delta' = 24$ symbol positions, and SER=10^{-2}. To identify row component RS code parameters, Algorithm 3 is applied and the variation of normalized non-zero-mean-ratio $\mu'(m_1, p_1, \phi_1)$ with respect to $[m_1, p_1, \phi_1]$ is shown. From the plot, it is inferred that $\mu'(m_1, p_1, \phi_1)$ reaches minimum at

$[m_1, p_1, \phi_1] = [4, 19, 28]$. Therefore, the RS code parameters $\hat{m_1} = 4$, $\hat{n_1} = 15$, and $\hat{p_1} = 19$ are estimated successfully using Algorithm 3. In addition, the synchronization of row component code is obtained by shifting $\hat{\phi}_1 = 28$ bit positions. This is because, a new RS codeword (i.e. row component code) will start after shifting $\hat{\phi}_1 = ((\beta \cdot \chi' \cdot m_1 \cdot n_1 + 1) - ((\Delta' - 1) \cdot m_1 + 1))$ bit positions provided the receiver starts receiving the coded symbols at Δ'^{th} symbol position of χ'^{th} row component RS code. Here, in this case $\beta = 1$ and $\chi' = 2$. Hence, the bit adjustment parameter to achieve synchronization is also estimated correctly using Algorithm 3. After shifting $\hat{\phi}_1$ bit positions, the RS coded binary data symbols are converted into the respective elements of GF($2^{\hat{m_1}}$) using $\hat{p_1}$. Now the roots of code polynomials are evaluated and it has been identified that $[\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7, \alpha^8]$ are the most likelihood roots of the generator polynomial of RS code $g_1(x)$. It is known that the generator polynomial for RS code is given by lcm$(\phi'_1(x), \phi'_2(x), ..., \phi'_{2t}(x))$, where $\phi'_i(x)$ is the minimal polynomial of α^i, which is equal to $x - \alpha^i$. Based on the estimated roots, the estimated generator polynomial of row component code is given by $\hat{g_1}(x) = \prod_{i=1}^{8} \phi'_i(x)$. Since $g_1(x)$ is a $n_1 - k_1 = 8$ degree polynomial, the estimated code dimension of row component code is given by $\hat{k}_1 = 7$.

The proposed algorithm (i.e. Algorithm 4) is applied on irregular product code RS$(15 \times 15, 7 \times 11, 4 \times 4)$ to extract column component RS code parameters. In Fig. 4(b), the variation of normalized non-zero-mean-ratio $\mu'(\phi_2)$ with respect to ϕ_2 is shown. It is observed from the figure that at $\phi_2 = 808$, normalized non-zero-mean-ratio is minimum. Therefore, the synchronization of column component code is obtained by moving 808 bit positions. This is because, a new RS product codeword will start after shifting $\hat{\phi}_2 = ((\beta \cdot \chi \cdot n_1 \cdot n_2 \cdot m_1 + 1) - ((\Delta' - 1) \cdot m_1 + 1))$ bit positions, where $\beta = \chi = 1$, provided the receiver starts receiving the product coded symbols at Δ'^{th} symbol position of χ^{th} product code. After shifting ϕ_2^{est} bit positions, the roots of code polynomials are evaluated to recognize $g_2(x)$ and k_2. It is noticed that $[\alpha, \alpha^2, \alpha^3, \alpha^4]$ are the most likelihood roots and the estimated generator polynomial of column component code is given by $\hat{g_2}(x) = \prod_{i=1}^{4} \phi'_i(x)$, where $\phi'_i(x) = x - \alpha^i$. Since $g_2(x)$ is a $n_2 - k_2 = 4$ degree polynomial, $\hat{k}_2 = 11$.



(a) (b)

Fig. 5. (a) Probability of correct estimation of RS and BCH product codes considering $M-$PSK and $M-$QAM schemes, respectively (b) Probability of correct estimation of RS product codes considering 16-QAM scheme and BCH product codes considering 32-QAM for different values of code dimension

In Fig. 5(a), the performance plots in terms of probability of correct estimation for square RS product codes and square BCH product codes with respect to signal-to-noise ratio (SNR) are shown for RS$(15\times 15, 7\times 7, 4\times 4, 19\times 19)$ and BCH$(31\times 31, 11\times 11, 5\times 5)$, respectively, considering different $M-$ary phase-shift keying ($M-$PSK) and $M-$ary quadrature amplitude modulation ($M-$QAM) schemes. Note that AWGN channel model is assumed for obtaining the performance plots. From Fig. 5(a), it is inferred that the probability of correct estimation of BPSK scheme is better than QPSK and 8-PSK schemes and it is also observed that the probability of correct estimation of 32-QAM is better than 64-QAM scheme. Therefore, as the modulation order increases, the performance of the proposed algorithms deteriorates, as expected.

In Fig. 5(b), the probability of correct estimation of irregular RS product codes considering $n_1 = n_2 = 15$, $m_1 = m_2 = 4$, and $p_1 = p_2 = 19$, and 16-QAM scheme with respect to SER is given for different values of code dimension, which are given by $k_1 = k_2 = 7$ and $k_1 = k_2 = 9$. From the plots, it can be observed that the probability of correct estimation deteriorates with increase

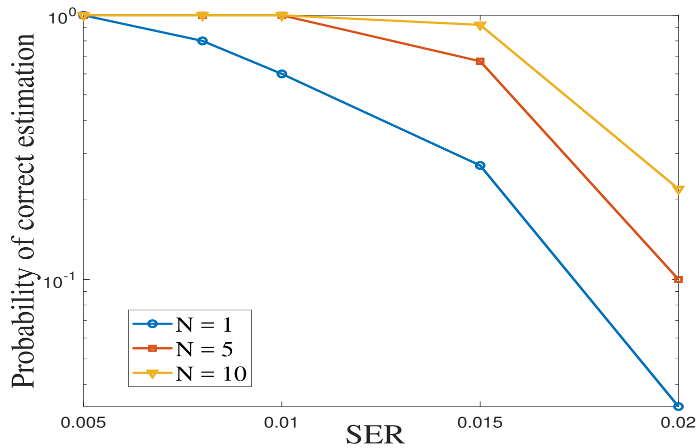


Fig. 6. Probability of correct estimation of irregular RS product codes considering $RS(31\times31, 21\times17, 5\times5, 37\times37)$ for different number of iterations N

in code dimension or code rate values. This is because, it is always difficult to distinguish rank deficient and full rank data matrices for higher values of code rate or code dimension based on non-zero-mean-ratio values due to less number of dependent columns. Similarly, in Fig. 5(b), the probability of correct estimation versus SER curves are shown for BCH product codes considering $n_1 = n_2 = 31$, $m_1 = m_2 = 5$, and 32-QAM scheme for different values of code dimension, which are given by $k_1 = k_2 =$ 11, 16, and 21. It has been inferred that the performance degrades with increase in code dimension or code rate values similar to RS product codes.

In Fig. 6, the probability of correct estimation of irregular RS product codes considering RS$(31\times31, 21\times17, 5\times5, 37\times37)$ is given for different number of iterations N. As expected, when the number of iterations N increases, improvement in the probability of correct estimation of code parameters is observed. In [19], RS code parameter estimation algorithm has been proposed for noisy scenario without iterations i.e. $N = 1$. Here, an interesting comparative study has been given in Table I, which compares the proposed RS code parameter estimation algorithm in [19] with the row component code parameter estimation algorithm (i.e. Algorithm 3) considering RS

TABLE I

PERFORMANCE COMPARISON OF ALGORITHM 2 PROPOSED IN [19] WITH ALGORITHM 3 PROPOSED IN OUR WORK CONSIDERING RS CODE $RS(15,7,4,19)$ AND 16-QAM SCHEME

SER	Probability of correct estimation ([19])	Probability of correct estimation (Algorithm 3)
0.01	1	1
0.03	1	1
0.05	1	1
0.06	0.93	1
0.08	0.9	0.9
0.09	0.6	0.6
0.1	0.2	0.2

code RS$(15,7,4,19)$ and 16-QAM scheme. In Algorithm 3, we assume $t=50$ i.e. $a=50\times b$ and $N=5$, whereas in Algorithm 2 proposed in [19], we assume $t=5000$ i.e. $a=5000\times b$ and $N=1$. From the performance comparison, it has been observed that considerably higher number of rows are required for the algorithm proposed in the prior work (without iterations) to approximately match the performance of Algorithm 3 proposed in our work. Alternatively, the minimum data size required to execute Algorithm 3 is 48205 symbols, whereas to execute Algorithm 2 proposed in the prior work, 4805000 symbols are required. In a nutshell, considerably large data size (i.e. 100 times more) is required for Algorithm 2 in [19] to approximately match the performance of Algorithm 3.

In [34], a novel packet error detection method was proposed in HARQ system to replace cyclic redundancy check (CRC) by exploiting the inherent error-detection capability of TPC code. It was shown using Monte-Carlo simulation results that the proposed CRC-free TPC-HARQ method achieves equivalent or higher throughput compared to CRC-based HARQ system with lower computational complexity. It is observed that the proposed method in [34] can be modified and used in a trial-and-error fashion to identify the product code parameters with reduced computational complexity and the same will be carried out as a part of our future work.

VII. Conclusions

In this paper, blind code parameter estimation algorithms based on non-zero-mean-ratio and zero-mean-ratio values are proposed for RS and BCH product codes, respectively, in noisy environment. The estimated parameters of two-dimensional product codes are codeword lengths, code dimensions, number of bits per symbol, primitive and generator polynomials of row and column component codes. Apart from the code parameters, the proposed algorithms also identify the bit position adjustment parameter to achieve frame synchronization. It is observed from the simulation results that the code and synchronization parameters are successfully recognized for various test cases. Further, in order to show the robustness of the proposed algorithms, the performance plots in terms of probability of correct estimation are shown for various M-QAM and M-PSK schemes, code dimension, and codeword length values. It is inferred that the probability of correct estimation improves with decrease in code dimension and modulation order values and increase in the number of iterations. Finally, the performance of the proposed algorithm for the parameter estimation of row component RS codes is compared with the RS code parameter estimation algorithm proposed in the prior work. It is noticed that the proposed algorithm requires smaller data size compared to the algorithm proposed in the prior work to attain equivalent performance.

References

[1] Perrins, E., "FEC systems for aeronautical telemetry," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2340–2352, Oct. 2013.

[2] Bluestein, L. I., "Interleaving of pseudorandom sequences for synchronization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-4, no. 4, pp. 551–556, Jul. 1968.

[3] Ziegler, J. F., “Automatic Recognition and Classification of Forward Error Correcting Codes,” M.S. thesis, M.S. thesis, Dept. Elect. Comput. Eng., George Mason Univ., Fairfax, VA, USA, 2000.

[4] Choi, C. and Yoon, D., “Enhanced blind interleaver parameters estimation algorithm for noisy environment,” *IEEE Access*, vol. 6, pp. 5910–5915, Sep. 2017.

[5] Weerackody, V., “Adaptive coding and modulation for satellite communication links in the presence of channel estimation errors,” *in Proc. IEEE MILCOM*, 2013, pp. 622–627.

[6] Zhang, J., Chen, S., Maunder, R. G., Zhang, R., and Hanzo, L., “Adaptive coding and modulation for large-scale antenna array based aeronautical communications in the presence of co-channel interference,” *arXiv:1711.09299 [cs.IT]*, Nov. 2017.

[7] Xia, T. and Wu, H.-C., “Novel blind identification of LDPC codes using average LLR of syndrome a posteriori probability,” *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 632–640, Feb. 2014.

[8] Zhang, H., Wu, H-C., and Jiang, H., “Novel blind encoder identification of Reed-Solomon codes with low computational complexity,” *in Proc. IEEE GLOBECOM*, 2013, pp. 3294–3299.

[9] Moosavi, R. and Larsson, E. G., “Fast blind recognition of channel codes,” *IEEE Trans. Commun.*, vol. 62, no. 5, pp. 1393–1405, May 2014.

[10] Marazin, M., Gautier, R., and Burel, G., “Blind recovery of k/n rate convolutional encoders in a noisy environment,” *EURASIP J. Wirel. Commun. and Netw.*, vol. 2011:168, pp. 1–9, 2011.

[11] Dingel, J. and Hagenauer, J., “Parameter estimation of a convolutional encoder from noisy observations,” *in Proc. IEEE ISIT*, 2007, pp. 1776–1780.

[12] Marazin, M., Gautier, R., and Burel, G., “Some interesting dual-code properties of convolutional encoder for standards self-recognition,” *IET Commun.*, vol. 6, no. 8, pp. 931–935, July 2012.

[13] Zrelli, Y., Marazin, M., Gautier, R., Rannou, E., and Radoi, E., “Blind identification of convolutional encoder parameters over GF(2m) in the noiseless case,” *in Proc. IEEE ICCCN*, 2011, pp. 1–5.

[14] Jing, Z., Zhiping, H., Shaojing, S., and Shaowu, Y., “Blind recognition of binary cyclic codes,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2013:218, pp. 1–17, 2013.

[15] Cluzeau, M. and Finiasz, M., “Reconstruction of punctured convolutional codes,” *in Proc. IEEE ITW*, 2009, pp. 75–79.

[16] Marazin, M., Gautier, R., and Burel, G., “Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bit stream,” *IET Signal Process.*, vol. 6, no. 2, pp. 122–131, April 2012.

[17] Swaminathan, R. and Madhukumar, A. S., “Classification of error correction codes and estimation of interleaver parameters in a robust environment,” *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 463-478, Sep. 2017.

[18] Zrelli, Y., Marazin, M., Gautier, R., Rannou, E., and Radoi, E., “Blind identification of code word length for non-binary error-correcting codes in noisy transmission,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2015:43, pp. 1–16, 2015.

[19] Swaminathan, R., Madhukumar, A. S., Wang, G., and Kee, T. S., “Blind reconstruction of Reed-Solomon encoder and interleavers over noisy environment,” *IEEE Trans. Broadcast.*, vol. 99, no. PP, pp. 1-16, Early access.

[20] Yu, P., Peng, H., and Li, J., "On blind recognition of channel codes within a candidate set," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 736–739, April 2016.

[21] Debessu, Y. G., Wu, H.-C., and Jiang, H., "Novel Blind Encoder Parameter Estimation for Turbo Codes," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 1917–1920, Dec. 2012.

[22] Yu, P., Li, J., and Peng, H., "A least square method for parameter estimation of RSC sub-codes of turbo codes," *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 644–647, Apr. 2014.

[23] Sicot, G., Houcke, S., and Barbier, J., "Blind detection of interleaver parameters," *Signal Process.*, vol. 89, pp. 450–462, April 2009.

[24] Swaminathan, R, Madhukumar, A. S., Teck, N. W., and Samson, S. C. M., "Parameter estimation of block and helical scan interleavers in the presence of bit errors," *Digital Signal Process.*, vol. 60, pp. 20–32, Jan. 2017.

[25] Lu, L., Li, K. H., and Guan, Y. L., "Blind detection of interleaver parameters for non-binary coded data streams," *in Proc. IEEE ICC*, 2009, pp. 1–4.

[26] Lu, L., Li, K. H., and Guan, Y. L., "Blind identification of convolutional interleaver parameters," *in Proc. IEEE ICICS*, 2009, pp. 1–5.

[27] Swaminathan, R, Madhukumar, A. S., Teck, N. W., and Samson, S. C. M., "Parameter estimation of convolutional and helical interleavers in a noisy environment," *IEEE Access*, vol. 5, pp. 6151–6167, 2017.

[28] Choi, C. and Yoon, D., "Novel blind interleaver parameter estimation in a non-cooperative context," *IEEE Trans. Aerosp. Electron. Syst.*, doi: 10.1109/TAES.2018.2875570.

[29] Jia, Y-Q., Li, L-P., Li, Y-Z., and Gan, L., "Blind estimation of convolutional interleaver parameters," *in Proc. IEEE WiCOM*, 2012, pp. 1–4.

[30] Rice, M., Palmer, J., Lavin, C., and Nelson, T., "Space-time coding for aeronautical telemetry: Part I-estimators," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 4, pp. 1709–1731, Aug. 2017.

[31] Rice, M., and Mcmurdie, A., "On frame synchronization in aeronautical telemetry," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 5, pp. 2263–2280, Oct. 2017.

[32] Imad, R., Houcke, S., and Jego, C., "Blind frame synchronization of product codes based on the adaptation of the parity check matrix,"*in Proc. IEEE ICC*, 2009, pp. 1–5.

[33] Imad, R., Sicot, G., and Houcke, S., "Blind frame synchronization for error correcting codes having a sparse parity check matrix," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1574–1577, June 2009.

[34] Mukhtar, H., Al-Dweik, A., and Al-Mualla, M., "CRC-free hybrid ARQ system using turbo product codes," *IEEE Trans. Commun.*, vol. 62, no. 12, pp. 4220-4229, Dec. 2014.

[35] Abdelbar, M., Tranter, W. H., and Bose, T., "Cooperative cumulants-based modulation classification in distributed networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 446–461, Sept. 2018.

[36] OShea, T. and Hoydis, J., "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[37] Ali, A. and Yangyu, F., "Automatic modulation classification using deep learning based on sparse autoencoders with nonnegativity constraints," *IEEE Signal Process. Lett.*, vol. 24, no. 11, pp. 1626–1630, Nov. 2017.

[38] Mukhtar, H., Al-Dweik, A., and Shami, A., "Turbo product codes: applications, challenges and future Directions," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 3052–3069, 2016.

[39] Mahmoud, M. S. B., Guerber, C., Pirovano, A., Larrieu, N., and Radzik, J., *Aeronautical air-ground data link communications*, Hoboken, NJ, USA: Wiley, 2014.

[40] Lin, S. and Costello, D. J., *Error control coding*, 2nd ed. Upper Saddle River, NJ, USA: Pearson Educ., 2004.

Swaminathan R has recently joined IIT Indore as an Assistant Professor after working as a Post-doctoral Research Fellow at Nanyang Technological University (NTU) Singapore from 2016 to 2019. Earlier, he has completed his PhD from IIT Kharagpur in 2016. He received the B.Tech. degree in ECE from SASTRA University, Thanjavur, in 2009 and M.E. degree in communication systems from the College of Engineering Guindy, Anna University, Chennai, in 2011. His current research focus is on designing hybrid free space optics/radio frequency communication systems for next-generation terrestrial and satellite communication, proposing algorithms for blind parameter estimation of forward error correcting codes and interleavers considering non-cooperative scenarios and validating the same using real data obtained from hardware testbed. In addition, his research interests include but not limited to power line communication, Index modulation techniques for next-generation wireless communication, Simultaneous Lightwave Information and Power Transfter (SLIPT), Non-Orthogonal Multiple Access (NOMA) techniques, Automatic channel code and interleaver identification techniques, etc. He received the gold medal from the College of Engineering Guindy, Anna University. Furthermore, he has been serving as a Reviewer for reputed IEEE journals and as a TPC member for reputed IEEE conferences.

A.S. Madhukumar A.S. Madhukumar's expertise lies in the areas of multi-tier cellular architecture, cooperative and cognitive radio systems, interference management, coding and modulation, new multiple access schemes, hybrid radio systems and other advanced signal processing algorithms for future communication systems. He is involved in a number of funded research projects, organizing international conferences, and a permanent reviewer for many internationally reputed journals and conferences. He has published over 275 referred international conference and journal papers. He is a recipient of Nanyang Award for Teaching Excellence in 2007 and obtained best paper awards in IEEE 35th Digital Avionics Conference in 2016 and in IEEE Integrated Communications, Navigations and Surveillance Conference in 2016 and in 2017.

A.S. Madhukumar received his B.Tech degree from College of Engineering, Trivandrum, India, M.Tech from Cochin University of Science and Technology, India and PhD from Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India. He is currently an Associate Professor in the School of Computer Engineering, Nanyang Technological University, Singapore. Before joining NTU, he was involved in communications and signal processing research at Centre for Development of Advanced Computing (Electronics Research and Development Centre), Govt. of India and Institute for Infocomm Research (Centre for Wireless Communications), Singapore. Dr. Madhukumar is a senior member of IEEE.

Wang Guohua received his B. Eng. degree in aircraft design engineering from Northwestern Polytechnical University, Xi'an, China, in 2003, and the M. Eng. and Ph.D. degrees, both in electronic engineering, from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2006, and Nanyang Technological University, Singapore, in 2010, respectively. He was working at Sensor Array Program, TL@NTU since 2010, as a Senior Research Scientist. His research interests are in signal processing and its applications in radar and communications, with recent focus on radar waveform diversity and design, robust beamforming, blind signal processing, and DF and wireless geolocation.