

Article

Blind Recognition of Forward Error Correction Codes Based on a Depth Distribution Algorithm

Fan Mei *, Hong Chen and Yingke Lei *

School of Electronic Countermeasures, National University of Defense Technology, Hefei 230000, China; chenye@nudt.edu.cn

* Correspondence: meifan14@nudt.edu.cn (F.M.); 22920142204021@stu.xmu.edu.cn (Y.L.); Tel.: +86-133-196-09706 (Y.L.)

Abstract: Forward error correction codes (FEC) are one of the vital sections of modern communication systems; therefore, recognition of the coding type is an important issue in non-cooperative communication. At present, the recognition of FEC codes is mainly concentrated in the field of semi-blind identification with known types of codes. However, based on information asymmetry, the receiver cannot know the types of channel coding previously used in non-cooperative systems such as cognitive radio and remote sensing of communication. Therefore, it is important to recognize the error-correcting encoding type with no prior information. Although the traditional algorithm can also recognize the type of codes, it is only applicable to the case without errors, and its practicability is poor. In the paper, we propose a new method to identify the types of FEC codes based on depth distribution in non-cooperative communication. The proposed algorithm can effectively recognize linear block codes, convolutional codes, and Turbo codes under a low error probability level, and has a higher robustness to noise transmission environment. In addition, an improved matrix estimation algorithm based on Gaussian elimination was adopted in this paper, which effectively improves the parameter identification in a noisy environment. Finally, we used a general framework to unify all the reconstruction algorithms to simplify the complexity of the algorithm. The simulation results show that, compared with the traditional algorithm based on matrix rank, the proposed algorithm has a better anti-interference performance. The method proposed is simple and convenient for engineering and practical applications.

Keywords: depth distribution algorithm; forward error correction codes; blind recognition; non-cooperative system; code type reconstruction; matrix estimation algorithm



Citation: Mei, F.; Chen, H.; Lei, Y. Blind Recognition of Forward Error Correction Codes Based on a Depth Distribution Algorithm. *Symmetry* **2021**, *13*, 1094. <https://doi.org/10.3390/sym13061094>

Academic Editor: Carlo Cattani

Received: 10 May 2021

Accepted: 11 June 2021

Published: 21 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

FEC codes are a vital encoding and encryption method in modern communication systems, whose characteristics of mathematics ensure that the receivers correct errors which have occurred during the process of transmission, which improve the quality of communication systems. The FEC coding types are widely used in non-cooperative communication fields such as adaptive modulation coding (AMC), cognitive radio and remote sensing communication [1]. In the field of information countermeasures, the type of error correction codes of the detected link must be recognized correctly, so as to obtain an effective information payload and finally to decipher the message of the sender. However, based on information asymmetry in the context of non-cooperative communication, there is no prior knowledge of the FEC coding types adopted on the detection link. Therefore, it must be identified by the corresponding data analysis methods. This is the core content of the paper—the blind identification of FEC coding types. With the development of communication technology, channel coding blind identification technology is widely used in the field of intelligent communication, multi-point broadcast communication and non-cooperative communication, and has become one of the research hotspots at home

and abroad [2–5]. Blind recognition of error-correcting code types is also of great significance to improve the accuracy and efficiency of code parameter recognition in asymmetry communication systems.

At present, the recognition of FEC coding types can be divided into two types: full blind recognition and half blind recognition. The full blind recognition is the forward error correction coding type recognition analysis under the condition that the access signal information is completely unknown. However, in some special cases, it is possible to obtain a set of forward error correction coding types that the sender may adopt through some prior knowledge. Then, using the intercepted data flow, the forward error correction coding method with the largest matching degree is searched for from the alternative set as the estimation of the coding types used in the detection link, which is the basic feature of semi-blind recognition.

At present, there are extensive researches on parameter estimation technologies such as linear block codes [6–8], convolutional codes [9–11] and Turbo codes [12]. However, these algorithms are based on a known coding type, solving its generation matrix and generation polynomial to realize parameter identification. Under the condition of non-cooperative communication, the unknown encoding type of the receiver will make it hard to recognize specific parameters. Therefore, it is crucial to study the recognition of types. At present, the recognition of FEC encoding types is mainly accomplished by mathematical methods according to the large amount of data received. In [13], the author proposed a type recognition algorithm based on rank criterion by using rank characteristics of receiving code words. However, the algorithm only analyzed the rank characteristics of each type separately, and did not comprehensively analyze and identify them. In [14], a type recognition algorithm for linear block codes and convolutional codes based on run-length features was proposed. This algorithm could recognize linear block codes and convolutional codes well, but it failed to recognize other encoding types. In [15], a coding system identification strategy for error-correcting codes with unknown coding sequences was proposed, which could provide a basic identification process for type recognition. However, it did not study and analyze the coding type identification under the condition of bit error, and failed to meet the basic requirements of engineering practice. In [16], the author proposed a general linear block code recognition and analysis method, which had a certain engineering practicability and succeeded in improving the recognition efficiency to a certain extent. Despite these considerable advantages, the algorithm lacked a strict mathematical proof. At the same time, this method was also semi-blind and was hard to be used under the condition of full blind recognition. A blind classification scheme of FEC codes is proposed in [17], which can be applied to the erroneous case. This blind classification scheme requires the knowledge of parameters of FEC codes and the classification performance heavily depends on the accuracy of rank criterion. In order to improve the accuracy of rank criterion, a random row permutation of the intercepted matrix is used, but a theoretical analysis of the accuracy of rank criterion according to the number of random row permutations was not yet performed. In [18], a new algorithm for calculating the rank of intercepted matrix is proposed. This algorithm does not need existing parameters and basically realizes blind recognition, but it does not fundamentally get rid of the problem of poor anti-jamming ability of the matrix rank algorithm. In recent years, forward error correction coding based on a convolution neural network recognition algorithm began to rise, the learning ability of powerful neural networks making it possible to automatically extract the feature identification number type [19,20]; however, this algorithm needs a large database, and a lot of learning time to ensure the correct recognition algorithm. The practical application prospect also must be observed. Hence a new identification algorithm with a low complexity and a high fault-tolerance is needed, which will be applied to many types of forward error correction codes and is convenient for engineering and practical application.

Depth distribution algorithm is a new feature of linear codes first studied by Etzion [21] from the perspective of cryptography, and a new method of equivalent linear codes—a

deep equivalence class was also proposed at the same time. In [22], the authors proposed a blind parameter estimation of linear block codes based on a depth distribution algorithm based on Etzion. The core algorithm is to obtain the distribution of the depth value through a difference operation and obtain the generation matrix of the depth value under the premise of the code length and code parameters. In this paper, convolutional codes and Turbo codes are equated to linear block codes by code type reconstruction based on traditional algorithms, and a depth distribution algorithm is used to recognize code types. The simulation results show that compared with the traditional matrix rank algorithm, the anti-noise performance of the proposed algorithm is significantly enhanced.

In this paper, an improved matrix analysis method based on Gaussian elimination is used to solve the problem of code length and a starting bit of linear block codes under the premise of a perfect frame synchronization of code words. First, use gaussian elimination to the analysis matrix [23,24], which is based on the symmetry of the matrix, and calculate the normalization of the columns. Then, under the two cases of whether the matrix column vectors are linearly independent, the decision threshold is established according to the different distribution of normalized column weight to find out the location of the maximum number of columns, and determine the length of code based on periodic distribution characteristics. In order to realize the synchronization of the starting bit, the analysis matrix was established under different starting points of the code word, and the corresponding position was found when the number of columns with a low weight was the largest according to the steps of the same length of the code. Therefore, so as to further improve the recognition accuracy, the analysis matrix is randomly exchanged and repeated several times at successive points near the starting point of real code words. Experimental results show that, compared with the traditional Gaussian elimination algorithm, its anti-noise performance is significantly enhanced.

The main contributions of this paper are as follows:

1. Aiming at FEC codes in a non-cooperative communication system, a recognition method of code type based on depth distribution algorithm is proposed in this paper. Compared with the traditional matrix rank analysis algorithm, the anti-noise performance of this method is significantly improved and the recognition effect is better.
2. According to the code type characteristics of convolutional codes and Turbo codes, the code reconstruction algorithm is adopted to convert different codes into linear block codes, making it possible to identify them with the same algorithm.
3. To recognize the code length and starting bit of linear block codes, an improved matrix analysis method based on Gaussian elimination is adopted in this paper. By establishing decision thresholds for different distributions in the normalized columns of the analysis matrix, and randomly exchanging matrix rows and repeating operations, the accuracy of algorithm identification is effectively improved.

The remaining contents are organized as follows: Section 2 presents the system model and recognition process. In Section 3, an overview of depth distribution algorithms are detailed. In Section 4, we introduce the recognition algorithms of three different coding types. In Section 5, various simulation results are provided to confirm the validity of the proposed schemes and in Section 6, conclusions are drawn.

2. The Recognition Process

The section presents the schematic diagram of the digital asymmetry communication system based on FEC coding recognition, as shown in Figure 1. It includes source coding, channel coding, a modulator, demodulator, data storage and abundant channel simulation modules which aim to put the simulated channel propagation model into use of simulated code data.

Generally, a source coding technique needs to compress data in order to transmit information more quickly, while a channel coding technique is on the contrary. So as to assure the reliability of transmission, redundant codes, called supervised codes, will

be added to the information codes to form a certain constraint relationship. We assume that bits of information from transmission is encoded to obtain the code words of code length $N(N > k)$ through channel coding at the transmitter end, and then mapped to the symbol vector through BPSK modulation. In the model, the channel adopts an additive white Gaussian noise channel (AWGN), and the Gaussian noise samples are distributed to $n \sim N(0, \sigma^2)$, $\sigma \in R$. The SNR is defined as the bit error rate (BER) and the bit energy noise ratio (E_b/N_0).

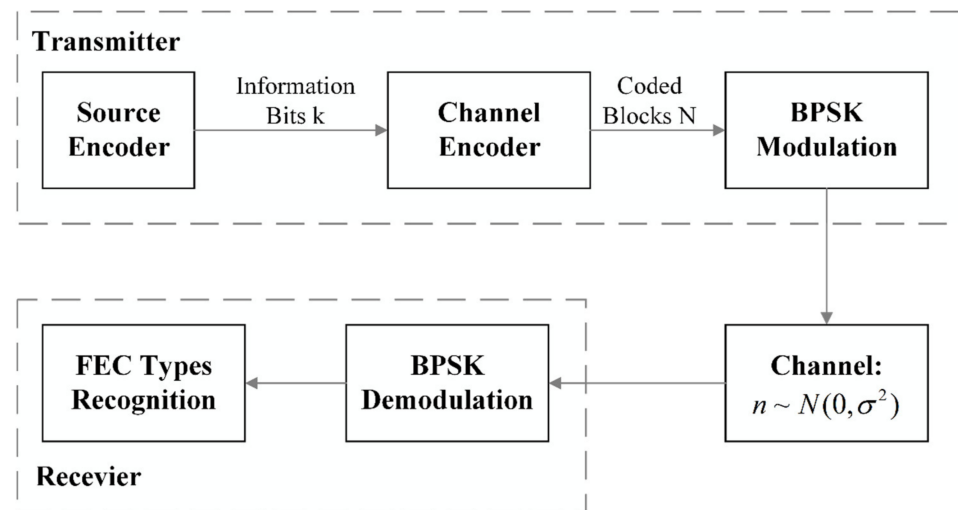


Figure 1. General block diagram of the blind recognition.

The received codeword is defined as r , which is also called the receiving codeword. Because the symbol vector is disturbed by noise through the AWGN channel, it is then defined as:

$$r = e + v \quad (1)$$

where e is the error pattern. In general, BPSK demodulation is performed before the types of forward error correction codes are identified. On the basis of the Gaussian distribution of channel noise, the logarithmic likelihood ratio (LLR) of the transmitted information is showed as:

$$LLR(r) = \ln \frac{P(v=0|r)}{P(v=1|r)} = \frac{2}{\sigma^2} \quad (2)$$

σ is the variance of the channel noise and r is the code words of reception, and the recognition model of the FEC code is used to identify the information bit vector sent by r . In this work, we focus on the recognition of types of FEC codes.

3. Overview of Depth Distribution Algorithms

Depth distribution algorithm is a new feature of linear code first studied by Etzion from the perspective of cryptography, and its definition is given as follows:

Definition 1. Let $c = b_1 b_2 \dots b_n$ be a code word of block code C of length n on $GF(q)$, defines $D(c) = (b_2, -, b_1, \dots, b_n, -, b_{n-1})$, and calls D differential operation of code c [25].

Definition 2. The depth of code word c is the minimum non-negative integer z which satisfies $D^z(c) = (0^{n-z})$. If such the z does not exist, the depth of c is the codeword length n . Where $D^z(c)$ represents the code words c to do z differential element calculation. Mark D_j is the number of code words with depth j , and set $\{D_0 D_1 \dots D_n\}$ is called the depth distribution of code c . The subscript set $\{j | 1 < j < n, D_j \neq 0\}$ is called the depth spectrum of the codeword c [26].

Theorem 1. Let C be a linear block code of (n, k) on $GF(q)$. For all non-zero code words in C , there are exactly k non-zero depth values of i_1, i_2, \dots, i_k , $i_1 < i_2 < \dots < i_k$, and $D_{i_m} = (q-1)q^{m-1}$ ($1 \leq m \leq k$) [27].

Theorem 2. On the finite field $GF(q)$, there are k non-zero depth values in the depth spectrum of any (n, k) linear block code C , and the codeword vectors with different depths are linearly independent [28,29].

The overall introduction of the depth distribution algorithm is completed, and the implementation of the specific algorithm principle will be described in the next chapter.

4. Depth Distribution Algorithm Based on Linear Block Codes

4.1. The Algorithm of Code Length Recognition

According to the definition of linear block code, the dimension of vector space composed of all code words is k . The basic theorem of matrix analysis is given in [30].

Theorem 3. For $p \times q$ order matrix X ($q < p, p > 2n$) constituted by (n, k) linear block code, when q is an integer multiple of n , the dimension of the unit matrix in the upper left corner after unitization is equal, and the rank of the matrix is not equal to the number of columns q .

Theorem 4. For $p \times q$ order matrix X (q is a multiple of $n, p > 2n$) constituted by (n, k) linear block code, when the starting point of the block code coincides with the starting point of each row of the matrix, the rank of the matrix is minimum.

Although matrix analysis algorithm is simple and easy to implement, under the condition of the Gaussian white noise channel, the correlation between codes is destroyed and the recognition effect is poor, cannot meet the requirements of the algorithm, and the practicability is not high. Therefore, an improved algorithm based on a Gaussian elimination method is adopted in this paper to achieve an accurate recognition in the condition of noise.

Let x_i^j represents the element in row i , column j of matrix X , and x_i represents row i ; x^j represents column j ; ' \oplus ' represents binary field addition. A, B are initialized to unit matrix of $M \times M, n \times n$. Make Gaussian column elimination of X , and the formula is:

$$MXN = \tilde{X} \quad (3)$$

Loop from 1 to n , and the steps of each step are as follows:

Step 1: If x_i^i equals to 0, x_i^j is the first element which equals to 1 on the right side of x_i^i , changes x_i^i and x_i^j , and changes n_i, n_j of matrix N at the same time;

Step 2: If x_i^i equals to 0, x_i^j is the first element which equals to 1 belows x_i^i , changes x_j^i and x_i^i , and changes m_i, m_j of matrix M at the same time;

Step 3: If x_i^i equals to 1, calculate as $x^i \oplus x^{j'}$, $x_i^{j'}$ includes all the elements on the right-hand side of x_i^i that are equal to 1. And calculate as $n^i \oplus n^{j'}$ of matrix N at the same time.

Which is shown in Figure 2.

After the above steps, the matrix X converts into the lower triangular matrix \tilde{X} as shown in Figure 2, and unit matrix M, N converts into \tilde{M}, \tilde{N} . The formula is:

$$\tilde{M}\tilde{X}\tilde{N} = \tilde{X} \quad (4)$$

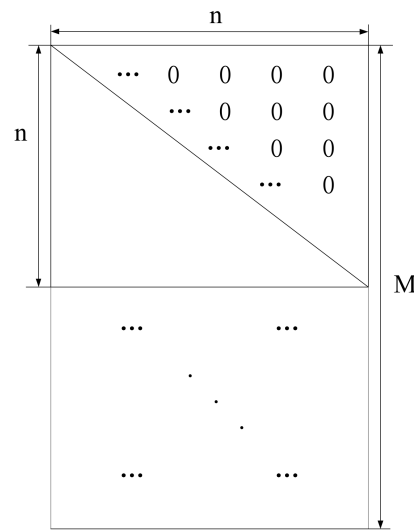


Figure 2. Low triangle matrix after Gauss elimination.

Because \tilde{M} is the elementary row operations of matrices X , based on matrix symmetry, the column vectors of $\tilde{X}\tilde{N}$ and \tilde{X} have the same weight distribution. Meanwhile, let $\tilde{M}\tilde{X}\tilde{n}^j = \tilde{x}^j$. If $n = n_0$, there are linearly dependent column directions of X , and there will be column vectors in \tilde{x}^j where all the elements are zero or where the hamming weight is small after Gaussian column elimination. At the same time, column vectors \tilde{n}^j exists in the dual space H with high probability. For \tilde{x}_i^j in \tilde{x}^j [31], the formula of (0, 1) distribution probability is:

$$P(\tilde{x}_i^j = 0) = \frac{1 + (1 - 2p_e)^{W_n}}{2} \quad (5)$$

$$P(\tilde{x}_i^j = 1) = \frac{1 - (1 - 2p_e)^{W_n}}{2} \quad (6)$$

In Formulas (5) and (6), p_e is the bit error rate of linear block codes; W_n is the code weight of \tilde{n}^j ; N_j is the code weight of \tilde{x}^j , and it satisfies a binomial distribution $B(M, (1 - (1 - 2p_e)^{W_n})/2)$. In the same way, when $n \neq n_0$ or $n = n_0$, and \tilde{x}^j is a linearly independent column vector, the formula is:

$$P(\tilde{x}_i^j = 0) = P(\tilde{x}_i^j = 1) = \frac{1}{2} \quad (7)$$

In Formula (7), N_j satisfies a binomial distribution $B(M, 1/2)$.

Let $\psi(j) = 2N_j/M, 1 \leq j \leq n$. For all possible code word lengths n , the code sequence is shifted and the matrix established by each shift is denoted as $X(n, d), 0 \leq d < n$. Defining judgment thresholds β , if there are several columns in $\tilde{X}(n, d)$ which make $\psi(j) \leq \beta$, we can prove that the corresponding columns in $X(n, d)$ are linearly dependent, which is shown as:

$$\varepsilon_{n,d} = \text{Card}(\{j \in \{1, 2, \dots, n\} / \psi(j) \leq \beta\}) \quad (8)$$

In Formula (8), $\text{Card}()$ represents the cardinality of a collection. For every possible code length n , the sum of the n shifts is $\varepsilon_{n,d}$, and the normalization is carried out. The formula is as follows:

$$\tilde{\varepsilon}_n = \frac{\sum_{d=0}^{n-1} \varepsilon_{n,d}}{n^2} \quad (9)$$

By analyzing the above formulas, it can be seen that when $n = \alpha n_0 (\alpha \in N^+)$, $\tilde{\varepsilon}_n$ is going to have a maximum. The n at the first maximum value is recognized as the code length n_0 .

Then, the optimal decision threshold mentioned above will be solved. According to the above analysis, a false alarm probability P_{fa} and a miss probability P_{nd} are shown as:

$$P_{fa} = \sum_{i=0}^{\lfloor M_\beta/2 \rfloor} \binom{i}{M} 0.5^M \quad (10)$$

$$P_{nd} = \sum_{i=\lfloor M_\beta/2 \rfloor + 1}^M \binom{i}{M} \left(\frac{1 - (1 - 2p_e)^{M_n}}{2} \right)^i \cdot \left(\frac{1 + (1 - 2p_e)^{M_n}}{2} \right)^{M-i} \quad (11)$$

According to the minimum error probability criterion, the judgment threshold β_1 is defined as:

$$\beta_1 = \arg \min_{\beta} (P_{fa} + P_{nd}) \quad (12)$$

According to the central limit theorem, when M is large enough, the binomial distribution tends to be normal. Based on the symmetry of the normal distribution, the analytic value of β_1 can be expressed as:

$$\beta_1 = \frac{-b - \sqrt{b^2 - ac}}{a} \quad (13)$$

In Formula (13):

$$a = \frac{-M(1 - 2p_e)^{2M_n}}{2} \quad (14)$$

$$b = -\frac{M[1 - (1 - 2p_e)^{M_n}](1 - 2p_e)^{M_n}}{2} \quad (15)$$

$$c = M[1 - (1 - 2p_e)^{\omega_b}](1 - 2p_e)^{\omega_b} - \frac{[1 - (1 - 2p_e)^{2M_n}] \ln[1 - (1 - 2p_e)^{2M_n}]}{2} \quad (16)$$

In general, the code weight of the dual code words of the system code is relatively small. In order to facilitate the calculation of β_1 , $M_n = \lceil \frac{n}{2} \rceil$. When M is large enough, the false alarm probability is:

$$P_{fa} = \Phi\left(\sqrt{\frac{M}{2}}(\beta - 1)\right) \quad (17)$$

In Formula (17), Φ is the standard normal distribution. As $t \leq -3.9$, $\Phi(t) \approx 10^{-4}$. Probability usually refers to $|t| \geq 3.9$ as an impossible event. Therefore, we can derive from $\sqrt{\frac{M}{2}}(\beta - 1) \leq -3.9$:

$$\beta_2 \leq 1 - 3.9\sqrt{\frac{2}{M}} \quad (18)$$

β_1 will grow as M_n grows up. At this time, if M diminishes, $\beta_1 > \beta_2$, which will cause a larger false alarm probability. In practical application, it is generally adopted $M = 20n$ and the optimal decision threshold is defined as:

$$\beta^* = \min(\beta_1, \beta_2) \quad (19)$$

The code length recognition of linear block codes has been solved, and the starting point of code words will be identified in the following chapters.

4.2. The Algorithm of Code Starting Bit Recognition

After the recognition of code length n_0 , we take out n_0 the value of $\varepsilon_{n_0,d}$, then the maximum of $\varepsilon_{n_0,d}$ is the starting bit of the code word. In practical application, the requirement of the code word starting bit recognition is higher than the code length recognition. There may be a situation in which the code length is correct but the code word starting point cannot be recognized. Therefore, a method of solving $\varepsilon_{n_0,d}$ multiple times and obtaining the average value is adopted. The specific steps are as follows:

Step 1: Select consecutive points P near the maximum value in sequence $\varepsilon_{n_0,d}$;

Step 2: For each point selected in Step 1, the row vectors of corresponding analysis matrix $X(n_0, d)$ are randomly exchanged and $\varepsilon_{n_0,d}$ is recalculated, which is carried out a total of Q times to obtain its mean value $\varepsilon'_{n_0,d}$;

Step 3: The shift value corresponding to the maximum value in the new $\varepsilon'_{n_0,d}$ sequence is selected, which is the starting bit of the code words.

The specific algorithm flow chart is shown in Figure 3.

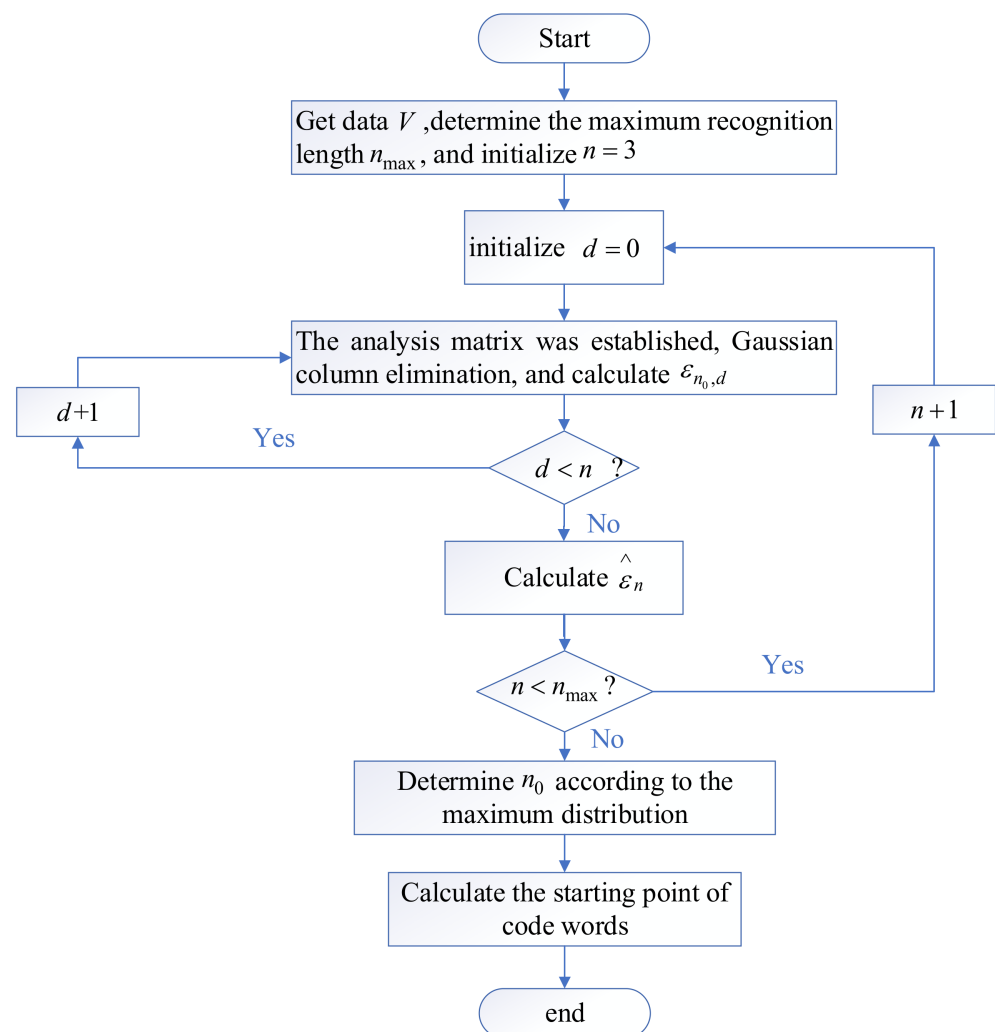


Figure 3. Flow diagram of algorithm.

4.3. Depth Distribution Algorithm Identifies Bit Rate and Generation Matrix

After the code word starting bit and code length are identified by the improved matrix analysis based on Gaussian elimination, the code rate t and matrix G are further identified by the depth distribution feature. The method of solving the code word depth value is described first. In this paper, a codeword $e = (1001011)$ of $(7, 3)$ cyclic codes on $GF(2)$ field is taken as an example in Figure 4:

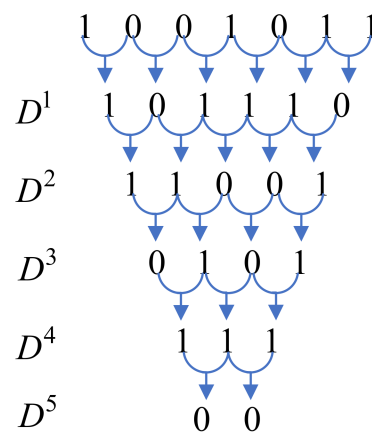


Figure 4. Diagram of depth distribution algorithm.

A differential algorithm is consequent minus referred to in the preceding paragraph in the mathematical theory, and in the $GF(2)$ for a binary linear block code is the binary addition of adjacent operations, which means every two adjacent elements add to each other in c to get a new code word, and a loop operation until the code word element is all of 0 and the emergence of a new code word number is the depth value. The depth value of code word c is calculated to be 5.

According to the example, we conclude that the depth value of the random code word n with the generated length of random sequence is $0\ 1\ 2\ \dots\ n$, and its depth distribution is $\{2^0, 2^0, 2^1, 2^2, \dots, 2^{n-1}\}$. According to Theorem 1, the depth distribution of linear block codes can be known. In view of the noise in the actual situation, the algorithm in this paper can still achieve the bit rate and generation matrix under the condition of code error.

4.3.1. The Recognition of Rate t

For the case of no error codes, on the premise that the code length and the starting point of the code word are known, if there is only a k value in the depth spectrum of (n, k) linear codes, the code rate is:

$$t = \frac{k}{n} \quad (20)$$

Under the same conditions, according to Theorem 1, the distribution of code words corresponding to the non-zero depth value of linear block codes is characterized by a two-times increase in the number of code words corresponding to the depth value with the increase in the depth value. This characteristic is used to verify the situation of error codes. Due to the existence of the error, if there appears in the depth of a non-zero value and its previous adjacent depth of non-zero values a code word number that is not about two times bigger, which shows that the depth value of the corresponding code words exist in error the ones when the ratio is far greater than or less than two times, it shows that the depth value of the corresponding code word code element is mostly an error and we then reduce the error. The remaining types of non-zero depth values are k and the bit rate is $t = \frac{k}{n}$.

4.3.2. The Recognition of Generator Matrix G

The expression form of the generation matrix G of linear block codes is not unique, but the row and column values of the generation matrix are the same, which are all matrices of k rows and n columns. According to the characteristics of the linear block code matrix generation, only k independent vectors need to be found, and each vector as a row, arranged in sequence, is to generate matrix G . According to Theorem 2, linear block codes have exactly k non-zero depth values. The non-zero codeword vectors with different depth values are linearly independent, and the matrix generated as row vectors is the generation matrix G . The matrix is identified according to this conclusion. A typical generation matrix

can be obtained by an elementary transformation of the generated matrix. For special codes—cyclic codes, the recognition of generated polynomials can be completed.

The key point of the algorithm is the recognition of information bit k , which has been recognized when the code rate t is recognized. For (n, k) linear block codes, there are and only k non-zero values in the depth spectrum. When the prior knowledge of e and n is known, the depth value of all M code words can be obtained, which is marked as j , $j \in \{0 \ 1 \ \dots \ n\}$. The sum of the code words with a statistical depth of j is D_j , and the depth distribution is $\{D_0 \ D_1 \ \dots \ D_n\}$. The depth of the spectrum is $j \in \{1 < j < n, D \neq 0\}$. According to Theorem 2, any code word corresponding to all non-zero depth values in the depth spectrum is selected and placed into a matrix, which is the generated matrix G .

The flow diagram is shown as Figure 5.

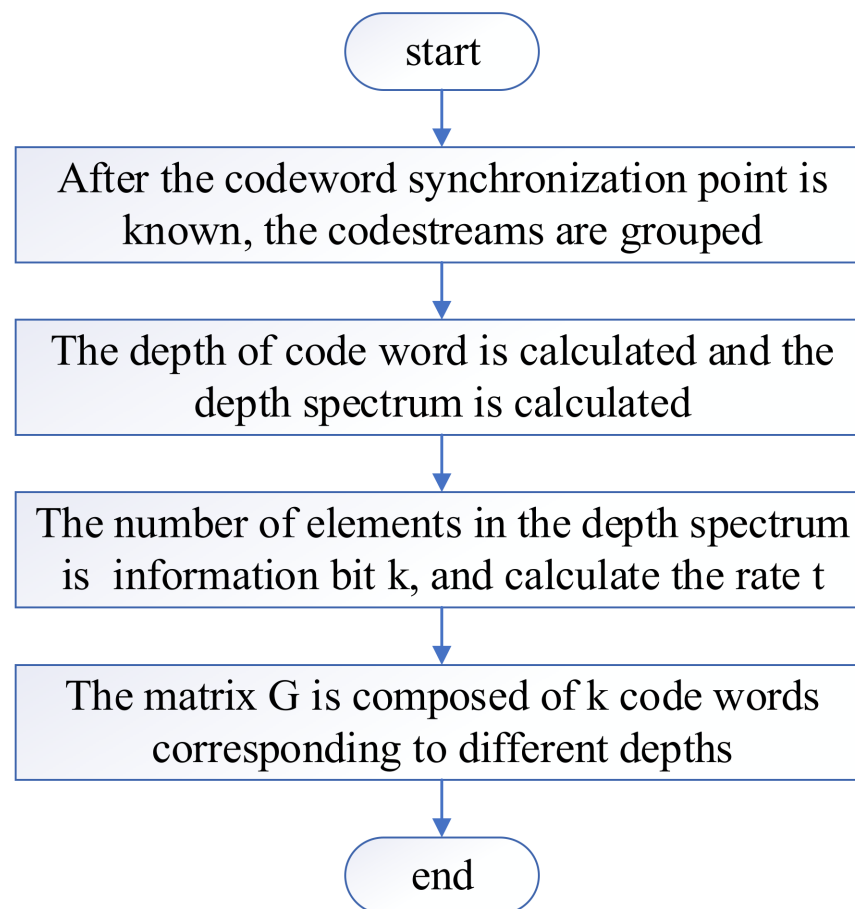


Figure 5. Flow diagram of recognition of rate and generating matrix.

Therefore, according to different parameters, the distribution characteristics of code word depth can be expressed as:

$$S(C(n, d))/n \leq S(C(n, d_e))/n \leq S(C(n_e, d_e))/n \quad (21)$$

Therefore, when the depth distribution value of the receiving codeword matrix reaches a minimum value, the parameter estimation is closest to the true value (there is a code length multiple relation), that is:

$$(n, d) = \text{ArgMin}(S(C(n_e, d_e))/n) \quad (22)$$

To sum up, when the accepted code is a linear block code, the distribution of the depth distribution value of the code word constituted by it can be expressed as:

$$\rho(S(C(n_e, d_e))) = \begin{cases} 1 & (n_e \neq an) \\ k/n \leq \rho \leq 1 & (n_e = an \ \& \ d_e \neq d) \\ k/n & (n_e = an \ \& \ d_e = d) \end{cases} \quad (23)$$

By analyzing the depth distribution characteristics of the receiving code words, the recognition of linear block codes can be realized in the case of error codes. At the same time, the recognition principle of linear block codes is the basis of convolutional codes and Turbo codes. In order to realize the code type recognition in the context of non-cooperative communication, a general unified recognition algorithm must be used to realize code type recognition according to the different characteristic quantities of different coding types obtained by the same algorithm. Therefore, in this paper, the recognition algorithm of various code types is domesticated to the equivalent linear block code recognition algorithm, so as to use the unified discriminant standard to judge all kinds of code types, and finally achieve the classification of coding types.

5. Depth Distribution Algorithm Based on Convolutional Codes

5.1. Principles of Convolutional Coding

Convolutional codes are the same as block codes, which carry out corresponding operations on information sequences according to certain mapping methods. The error correction is realized by adding redundant supervised sequences. The monitor bits within each code group are only related to the information bits within the group for block codes, and there is no constraint between each code group and other code groups. However, the convolutional code encoding system is a memory system, and its supervised code is not only related to the code input at the current moment, but is also related to the code information input at the previous moments.

Similar to block codes, convolutional codes are also determined by generation matrices or check matrices. If different generation matrices are used to encode information sequences, codeword matrices with different mathematical constraints will be obtained.

Different from block codes, the generating matrix of convolutional codes is a semi-infinite matrix. The convolutional codes can be expressed as:

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & g_m & 0 & \cdots \\ & g_0 & g_1 & \cdots & g_m & 0 & \cdots \\ & & g_0 & g_1 & \cdots & g_m & 0 & \cdots \\ & & & & \cdots & & & \end{bmatrix} \quad (24)$$

m is constraint length of convolutional codes. Equation (24) shows that the generation matrix of the convolutional code is determined by the value of the $m + 1$ preceding segments, and from $m + 2$ segments, the generation matrix equals 0. $g_i (0 \leq i \leq m)$ in Equation (24) can be shown as follows:

$$g_i = \begin{bmatrix} g_i(1,0) & g_i(1,1) & \cdots & g_i(1,n-1) \\ g_i(2,0) & g_i(2,1) & \cdots & g_i(2,n-1) \\ \vdots & \vdots & \ddots & \vdots \\ g_i(k,0) & g_i(k,1) & \cdots & g_i(k,n-1) \end{bmatrix} \quad (25)$$

According to Equation (25), there is a basic generation matrix $g = g_0, g_1, \dots, g_m$ in the generation matrix of the convolutional codes. Each row k in the generation matrix is obtained by shifting the fundamental generation matrix by n bits to the right, so G_∞ is completely determined by g .

5.2. Convolutional Code Recognition Algorithm

The recognition of the convolutional code type is based on the recognition of linear block code type, so it is necessary to deduce the internal correlation between convolutional code and linear block code. In this section, a special generation matrix based on symmetry of matrix is used to reconstruct the convolutional codes, and then the depth distribution characteristics of the codeword matrix of the convolutional codes are given.

For the subcode set of $(n - k)(k + 1, k)$, the generation matrix $G(x)$ of each subcode can be composed of the preceding line of k , and the first line of $(k + t)$, which is shown as:

$$G_t = \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,k} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k,1} & g_{k,2} & \cdots & g_{k,k} \\ g_{k+t,1} & g_{k+t,2} & \cdots & g_{k+t,k} \end{bmatrix} \quad (26)$$

$1 \leq t \leq n - k$, and t is an integer. The codeword vector $(c_1, c_2, \dots, c_k, c_{k+t})$ obtained by encoding the information bits through the generation matrix G_t will construct a new matrix with the generated matrix R_t :

$$R_t = \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,k} & c_1 \\ g_{2,1} & g_{2,2} & \cdots & g_{2,k} & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ g_{k,1} & g_{k,2} & \cdots & g_{k,k} & c_k \\ g_{k+t,1} & g_{k+t,2} & \cdots & g_{k+t,k} & c_{k+t} \end{bmatrix} \quad (27)$$

In Equation (27), the codeword vector c_t is a linear combination of the generating vector $g_{t,m}$. Therefore, the matrix R_t has linearly dependent columns whose determinant is 0.

$$\sum_{m=1 \cdots k, k+t} c_m \Delta_{m,t} = 0 \quad (28)$$

In Equation (28), $\Delta_{m,t}$ represents the minimum linearly dependent mapping of codeword vectors in the matrix R_t . There are altogether $(n - k)$ mapping relations in the subcode set, which satisfy Equations (26) and (27), and the dimension of each minimum linear correlation map is 1.

According to the characteristics of convolutional codes, the highest order D , the block frame length n and the block code rate k/n of the generated matrix should be identified to realize the coding reconstruction. Therefore, the receiving codeword matrix for convolutional code reconstruction can be expressed as:

$$C^{(m)}(n_e, v_e, D_e) = \begin{bmatrix} c_m^{D_e} & c_m^{D_e-1} & \cdots & c_m^0 \\ c_m^{D_e+1} & c_m^{D_e} & \cdots & c_m^1 \\ \vdots & \vdots & \ddots & \vdots \\ c_m^N & c_m^{N-1} & \cdots & c_m^{N-D_e} \end{bmatrix} \quad (29)$$

D_e is the estimation of D and v_e is also the estimation of k/n . Each receive matrix contains $(N - D_e)$ rows and D_e columns, the matrix consists of N bits of data, which

are successively shifted to form. We constructed the receiving codeword matrix group $C_t(n_e, v_e, D_e)$ to obtain:

$$C_t(n_e, v_e, D_e)^T = \begin{bmatrix} c_1^{D_e} & c_1^{D_e+1} & \dots & c_1^{N-1} & c_1^N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_1^0 & c_1^1 & \dots & c_1^{N-D_e-1} & c_1^{N-D_e} \\ c_2^{D_e} & c_2^{D_e+1} & \dots & c_2^{N-1} & c_2^N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_2^0 & c_2^1 & \dots & c_2^{N-D_e-1} & c_2^{N-D_e} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{t+m}^{D_e} & c_{t+m}^{D_e+1} & \dots & c_{t+m}^{N-1} & c_{t+m}^N \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{t+m}^0 & c_{t+m}^1 & \dots & c_{t+m}^{N-D_e-1} & c_{t+m}^{N-D_e} \end{bmatrix} \quad (30)$$

For the codeword matrix group formed by Equation (30), there are $(n - k)$ independent and symmetric check vectors. By solving these independent check vectors, the convolutional code reconstruction is realized. When parameters such as D_e , n_e and v_e are not correct, there is no specific constraint relationship inside the codeword group, and the depth distribution value is:

$$S = (k_e + 1)(D_e + 1) \quad (31)$$

When all parameters are recognized correctly and there is no error code, at least $k + 1$ bits in Equation (30) must satisfy the constraint relationship in Equation (28), which ensure that linear correlation vectors appear in the codeword group $C_t(n_e, v_e, D_e)$. When D_e is bigger than D with a correct code rate, there exists $(D_e - D)$ linearly independent vectors can satisfy the constraint relation of Equation (28). The depth distribution value of the accepted codeword group is solved by the following relations:

$$S(C_t(n_e, v_e, D_e)) = k(D_e + 1) + D \quad (32)$$

Using Equation (31) to normalize the depth distribution function in Equation (32), we can get the equation as follows:

$$\rho(S(C_t(n_e, v_e, D_e))) = 1 - \frac{1}{k+1} + \frac{D}{(k+1)(D_e+1)} \quad (33)$$

In order to realize the blind recognition of code types, a unified processing method must be used to process the code word matrices of various codes. For the subcode groups presented in this section, they can be equivalent to $(nD + n, nD + k)$ linear block codes. For each subcode in the subcode set, the number of check bits in the codeword matrix is $(D_e - D + 1)$. Therefore, the actual equivalent length of the code word is $n(D_e + 1)$. According to Equation (25), the depth distribution characteristics of the convolutional code codeword matrix under a different parameter estimation are calculated. So far, through the reconstruction of convolutional codes, the recognition process of the convolutional code word matrix has been equivalent to the recognition overpass of linear block code depth distribution matrix, and the equivalent normalized depth distribution of a convolutional code word matrix is shown as follows:

$$\rho(S) = \begin{cases} 1 & (n_e \neq n(D+1)) \\ 1 - \frac{n-k}{n} \left(\frac{a+1}{a+1+D} \right) \leq \rho \leq 1 & \begin{matrix} (n_e = n(D_e+1)) \\ \& d_e \neq d \end{matrix} \\ 1 - \frac{n-k}{n} \left(\frac{a+1}{a+1+D} \right) & \begin{matrix} (n_e = n(D_e+1)) \\ \& d_e = d \end{matrix} \end{cases} \quad (34)$$

With the equivalent normalized depth distribution function of convolutional code word matrix, blind recognition of linear block codes and convolutional code types can be realized.

6. Depth Distribution Algorithm Based on Parallel Convolutional Turbo Codes

6.1. Principles of Parallel Convolutional Turbo Coding

The most important feature of Turbo code is the introduction of interleave and deinterleave to achieve coding randomness, and the effective combination of several short codes by code word deletion and reuse to achieve long codes, making its performance close to the Shannon limit. Turbo codes can be divided into three categories according to the different arrangement structures of code words; namely, Parallel Cascaded Convolutional Codes (PCCC), Serial Cascaded Convolutional Codes (SCCC) and Hybrid Cascaded Convolutional Codes (HCCC) [32]. Among them, the parallel cascade convolutional code is the most widely used permutation, and it is also the main research object in this paper.

The structure of a parallel cascade convolutional code encoder mainly includes interleave, a component encoder, an eraser and a multiplexer. The function of interleave is to reset the position of each bit in the input information sequence, change the weight distribution of the code, control the distance characteristics of the code sequence, and effectively reduce the correlation between the test sequences. The eliminator and multiplexer can combine the short code to improve the transmission efficiency of the system. At the same time, the interleave is mainly used to scatter the distribution of error bits and improve the error-correcting ability of the system [33]. In parallel, a cascaded convolutional code component encodes a recursive system convolutional encoder (RSC), which is generally used to implement fractional encoding [34,35]. Recursive system convolutional encoders are based on linear feedback shift registers (LFRS) to randomize input data. At the same time, using RSC can achieve better performance compared with traditional convolutional encoding at low SNR. According to whether the RSC of an interleave frame in PCCC returns to zero, the Turbo code can be divided into non-return to zero Turbo code and a return to zero Turbo code. Generally speaking, the non-return to zero of the encoder will have a certain negative impact on the coding performance. If the length of interleaving is small, zero processing must be carried out. However, when the interleaving length is large, the return to zero processing can be ignored, so as to reduce the complexity of the coding system. The classical structure diagram of PCCC is shown in Figure 6.

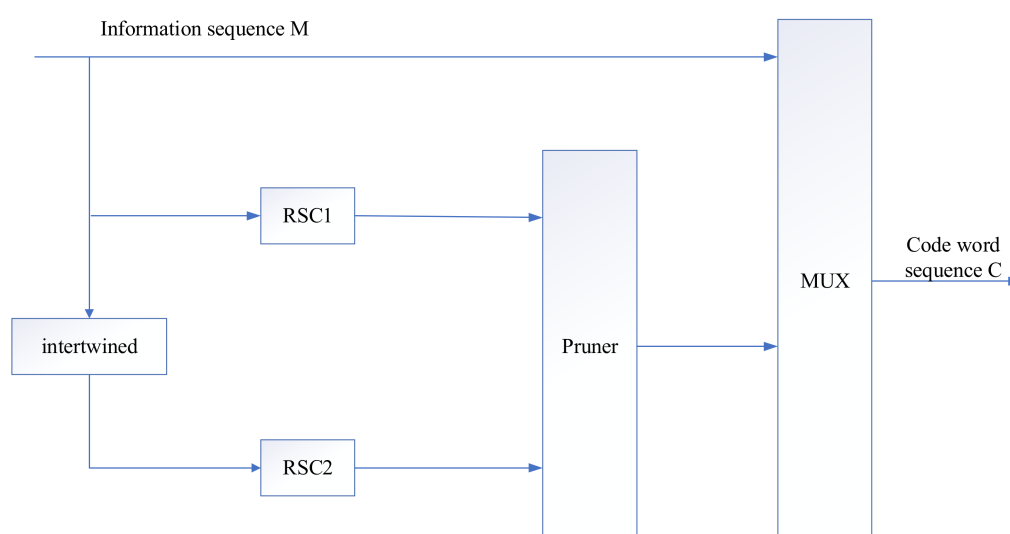


Figure 6. PCCC encoder structure.

6.2. Recognition Algorithm of Parallel Convolutional Turbo Coding

The blind recognition of parallel cascaded convolutional Turbo codes is also based on coding reconstruction, which establishes the corresponding relationship with linear block codes through reconstruction, so that the recognition method of linear block codes can be used on the identification of Turbo coding types. This section mainly analyzes the parallel cascaded convolutional Turbo codes as shown in Figure 6 and gives the depth distribution characteristics of the code words of this type of Turbo code, and then realizes the recognition of classical PCCC coding types.

First, we describe the classical PCCC. k is the number of information sequence bits; $(n_1 - k)$ is the number of supervisory positions RSC1 and $(n_2 - k)$ is the number of supervisory positions RSC2. The interleaving length is 1. Assuming that there is no deletion structure in the PCCC encoder, that RSC2 only outputs supervised check bits and that the transmission channel is error-free, the code word length of the encoder output is $(n_1 + n_2 - k)$. In this section, the convolutional code recognition method is used to identify and analyze the encoder RSC.

According to the difference of interleaves, the ones in PCCC encoders can be divided into two categories: packet interleaves and convolution interleaves. In this paper, the interleaving mode of interleave is defined as $F(x)$, and l is the weaving length. We define $\theta \in \mathbb{Z}^*$ $i \in [0, l - 1]$, which leads to Equation (35):

$$F(\theta l + i) = \theta l + F(i) \quad (35)$$

For the information bit sequence M input by the PCCC encoder is defined as:

$$M = \begin{bmatrix} m_1^0 & m_2^0 & \cdots & m_k^0 \\ m_1^1 & m_2^1 & \cdots & m_k^1 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \quad (36)$$

We construct an interleave with the length of $k(D + 1)$, and the sequence of raw information bits M passes through the interleave. A new matrix M^* is generated according to the mapping relation of $F(x)$, M^* is shown as Equation (36). We assume $i \in [0, k]$ exists, which makes the output code word equal to the original sequence of input information bits interwoven by the interleave, and then Equation (37) can be expressed as:

$$C^{(i)}(n_e, v_e, D_e, l_e) = \begin{bmatrix} m^{F^{-1}(Dk+i-1)} & m^{F^{-1}(D(k-1)+i-1)} & \cdots & m^{F^{-1}(i-1)} \\ m^{F^{-1}(i-1)+l} & m^{F^{-1}(Dk+i-1)} & \cdots & m^{F^{-1}(k+i-1)} \\ \vdots & \vdots & \ddots & \vdots \\ m^{F^{-1}(Dk+i-1)+l} & m^{F^{-1}(D(k-1)+i-1)+l} & \cdots & m^{F^{-1}(i-1)+l} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (37)$$

Turbo codes are similar to the reconstruction of linear block codes and convolutional codes in that equivalent check vectors are obtained from the code word matrix.

$$\begin{aligned} & C_i(n_e, v_e, D_e, l_e)X \\ &= (C^{(1)}(n_e, v_e, D_e, l_e) \mid C^{(2)}(n_e, v_e, D_e, l_e) \mid \cdots \mid C^{(k+i)}(n_e, v_e, D_e, l_e))X \\ &= 0 \end{aligned} \quad (38)$$

For the code-word matrix group in Equation (38), the preceding k subcode matrix of the code-word matrix group contains consecutive l bits of an information sequence, so the following constraint relation can be obtained:

$$\begin{aligned} & C_i^*(n_e, v_e, D_e, l_e) X \\ &= \begin{bmatrix} m^{l-1} & \cdots & m^0 & c_{k+i}^D & \cdots & c_{k+i}^0 \\ m^{2l-1} & \cdots & m^l & c_{k+i}^{2D+1} & \cdots & c_{k+i}^{D+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} X \\ &= 0 \end{aligned} \quad (39)$$

We selected a row vector from $C^{(1)}(n_e, v_e, D_e, l_e) \left| C^{(2)}(n_e, v_e, D_e, l_e) \right| \cdots \left| C^{(k+i)}(n_e, v_e, D_e, l_e) \right|$ whose row sequence number is $(vD + 1)$, and v is the positive integer. The corresponding equivalent matrix $M(n_e, v_e, D_e, l_e)$ can be obtained by interleaving the codes in each selected row vector. At the same time, similar to the selection of $M(n_e, v_e, D_e, l_e)$, we select the row vector from $C^{(k+i)}(n_e, v_e, D_e, l_e)$ whose row sequence number is $(vD + 1)$ to constrain matrix $C^{*(k+i)}(n_e, v_e, D_e, l_e)$. For a more general case, when the interleave length is $k(D + 1 + a)$, the coded reconstruction equation can be obtained according to Equation (39):

$$\begin{bmatrix} m^{l-1} & \cdots & m^0 & c_{k+i}^{D+a} & \cdots & c_{k+i}^0 \\ m^{2l-1} & \cdots & m^l & c_{k+i}^{2(D+a)+1} & \cdots & c_{k+i}^{D+a+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} X = 0 \quad (40)$$

In addition, for some encoders, the interleave length is not equal to $k(D + 1 + a)$ and the unity between the analysis and the identification process can be realized by solving the least common multiple l^* between them. So far, the reconstruction of Turbo code is solved skillfully by using a linear block code reconstruction algorithm, and the recognition of the Turbo code can refer to the basic algorithm of the linear block code. In order to realize the code type classification, the correlation between the depth distribution characteristics of the Turbo code word matrix and the linear block code word matrix is also analyzed.

According to previous studies on linear block codes and convolutional codes, the depth distribution characteristics of the code word matrix of Turbo codes are related to a parameter estimation of the code length, code rate, starting position and interleave length. Under different parameter estimation conditions, the depth distribution is significantly different. The parameter estimation of Turbo is also realized by reconstructing the equivalent linear block code, so the code length $(n_1 + n_2 - k)$ and rate $k/(n_1 + n_2 - k)$ can be estimated by the recognition method of linear block code length and the code rate. Therefore, the effect of interleave length estimation on the depth distribution characteristics of the Turbo code word matrix is an important research object.

We assume that the codeword matrix stores $\theta(n_1 + n_2 - k)l^*$ bits of data altogether, and μ is a positive integer. For parameter estimation of RSC1, RSC2 and interleave in encoder, when $l^* = k(D_1 + 1 + a_1) = k(D_2 + 1 + a_2)$, if $n_e = \theta(n_1 + n_2 - k)l^*$, the linear independent vectors in RSC1 and RSC2 are completely preserved. Under this condition, the number of information bits in the sequence output by RSC1 is $\theta k(D_1 + 1 + a_1)$ and the ones by RSC2 is $\theta k(D_2 + 1 + a_2)$. According to Equations (33) and (34), the total number of check bits in the code word matrix is $\theta((n_1 - k)(D_1 + 1 + a_1) + (n_2 - k)(D_2 + 1 + a_2))$. If $n_e = (n_1 + n_2 - k)(D_1 + 1 + \varphi)$, φ is a positive integer and $n_e \neq \theta(n_1 + n_2 - k)l^*$. Under this condition, based on the asymmetry of the matrix, all linear independent vectors in RSC1 are completely retained, while some linear independent vectors in RSC2 are not retained, which makes the depth distribution of the codeword matrix change.

From what is discussed above, when $n_e \neq \theta(n_1 + n_2 - k)l^* \neq (n_1 + n_2 - k)(D_1 + 1 + \varphi)$:

$$\rho(S(C(n_e, v_e, D_e, l_e))) = 1 \quad (41)$$

when $n_e = \theta(n_1 + n_2 - k)l^*$ and $d_e = d$:

$$\rho(S(C(n_e, v_e, D_e, l_e))) = 1 - \sum_{i=1,2} \left(\frac{n_i - k}{\theta k(D_i + a_i + 1)k(n_1 + n_2 - k)} \right) \quad (42)$$

when $n_e = \theta(n_1 + n_2 - k)l^*$ and $d_e \neq d$:

$$1 - \sum_{i=1,2} \left(\frac{n_i - k}{\theta k(D_i + a_i + 1)k(n_1 + n_2 - k)} \right) \leq \rho(S(C(n_e, v_e, D_e, l_e))) \leq 1 \quad (43)$$

when $n_e = (n_1 + n_2 - k)(D_1 + 1 + \varphi) \neq \theta(n_1 + n_2 - k)l^*$ and $d_e = d$:

$$\rho(S(C(n_e, v_e, D_e, l_e))) = 1 - \frac{n_1 - k}{(n_1 + n_2 - k)} \left(\frac{1 + \varphi}{D_1 + \varphi + 1} \right) \quad (44)$$

when $n_e = (n_1 + n_2 - k)(D_1 + 1 + \varphi) \neq \theta(n_1 + n_2 - k)l^*$ and $d_e \neq d$:

$$1 - \frac{n_1 - k}{(n_1 + n_2 - k)} \left(\frac{1 + \varphi}{D_1 + \varphi + 1} \right) \leq \rho(S(C(n_e, v_e, D_e, l_e))) \leq 1 \quad (45)$$

Equations (41)–(45) are the distribution function of the equivalent normalized depth distribution characteristics of the Turbo code word matrix, which can be used to realize blind recognition of the coding types of Turbo code.

7. Simulation Results

In this section, the effectiveness of the error correction coding type recognition analysis algorithm based on the depth distribution characteristics of the codeword matrix is verified through simulation experiments. In the simulation results, it is assumed that the communication system adopts a QPSK debugging mode, the signal transmission channel is the AWGN channel, and the signal-to-noise ratio is expressed by the bit error rate BER.

The simulation results listed in this section mainly include two aspects. Firstly, the code length estimation method based on the Gaussian elimination method and the code word starting point moment estimation method are simulated and verified. Secondly, the depth distribution characteristics of the codeword matrix of linear block codes, convolutional codes and Turbo codes are simulated and verified.

7.1. Simulation Results of Code Length and Starting Bits

We used MATLAB to generate 50,000 bit random 0 and 1 sequences, which were encoded by (15, 11) BCH code, a 1% error code was added, and the first 7 bits of data were deleted. The code length is first identified in the following section, and the result is shown in Figure 7. Only when the number of matrix columns is 15, 30, 45, 60 and other integer multiples of 15, can the normalized value ε reach the maximum value. When the code length is 15, the normalized value is 0.1022, which is the first maximum value, and the other length values are all 0. Then, symmetrically, a maximum occurs every 15 code words. Therefore, the corresponding code length can be estimated to be 15.

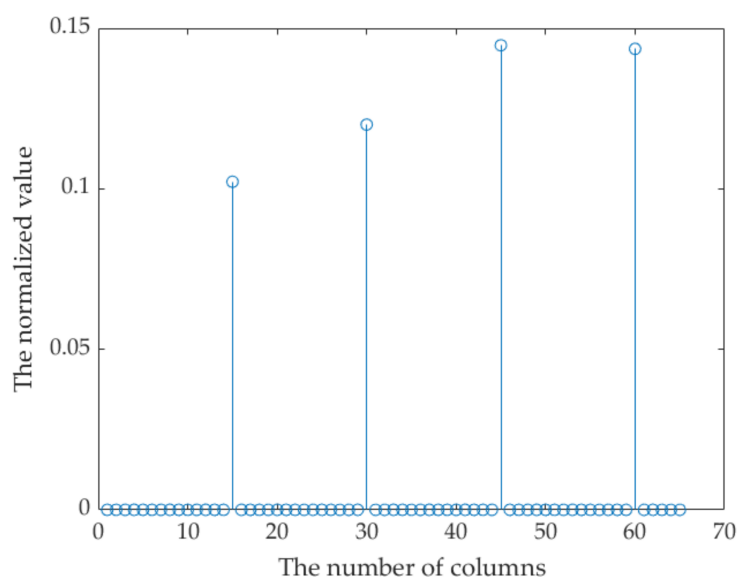


Figure 7. The code length recognition of (15, 11) BCH code.

The next step is to recognize the starting point of the code word; $n_0 = 15$ is known and we calculate the corresponding 15 kinds of shift of $\varepsilon_{n_0,d}$, as shown in Figure 8. This can be found when the code word is affected by the error condition, and originally it should only be in a state of 8 shift to the maximum, but in Figure 8, the $\varepsilon_{n_0,d}$ is identified.

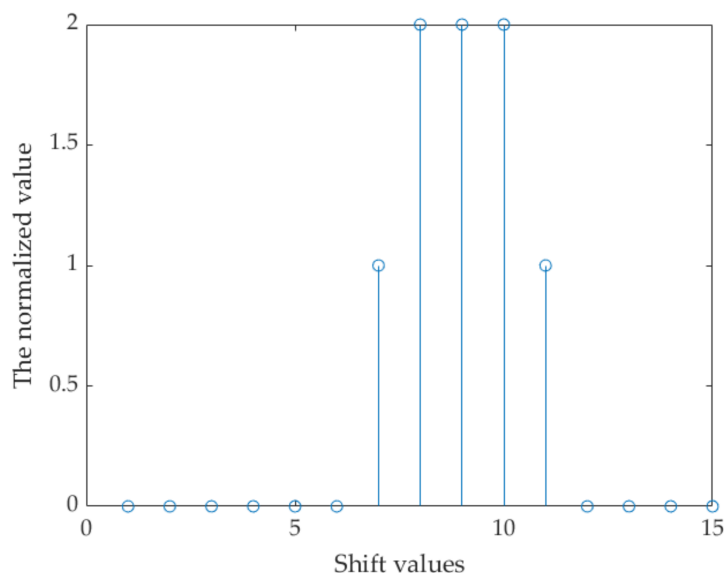


Figure 8. The recognition result of code start bit for (15, 11) BCH code.

Therefore, a total of 5 shift states from 7 to 13 were selected, and the average number of $Q = 30$ was taken to calculate $\varepsilon_{n_0,d}$ in each state again, as shown in Figure 9. It can be seen from the observation that, only when the shift state is 8 is the corresponding $\varepsilon_{n_0,d}$ value the largest, so this is the starting point of the code word, which is consistent with the actual simulation conditions, and the algorithm is valid.

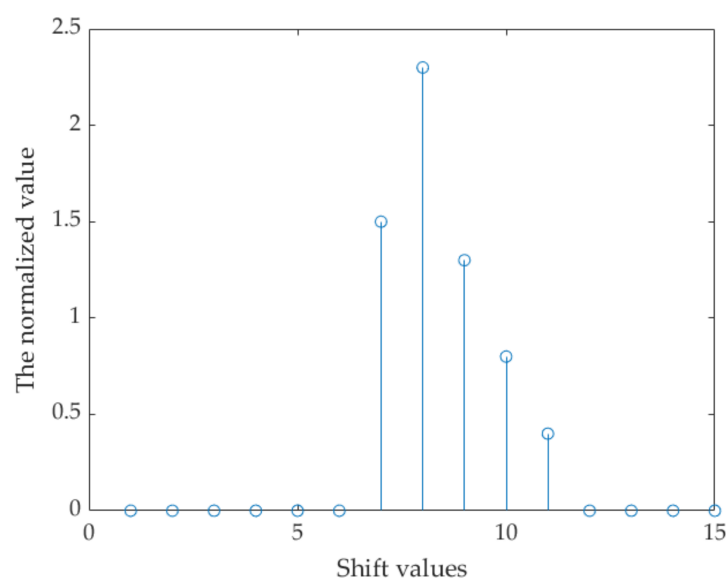


Figure 9. The recognition result of code start bit for (15, 11) BCH code.

7.2. Simulation Results of Depth Distribution Characteristics of Linear Block Codes

After the code length and the starting point of the code word are known, the depth distribution property is used to further identify the rate and generation matrix of the linear block codes. In order to facilitate a direct observation, BCHs (15, 5) are selected as the research objects in this section. A total of 10,000 code words are selected to calculate the depth value under the conditions that the bit

error rate is 1%. Code words with the same depth value are superimposed to generate the depth distribution histogram, as shown in Figure 10.

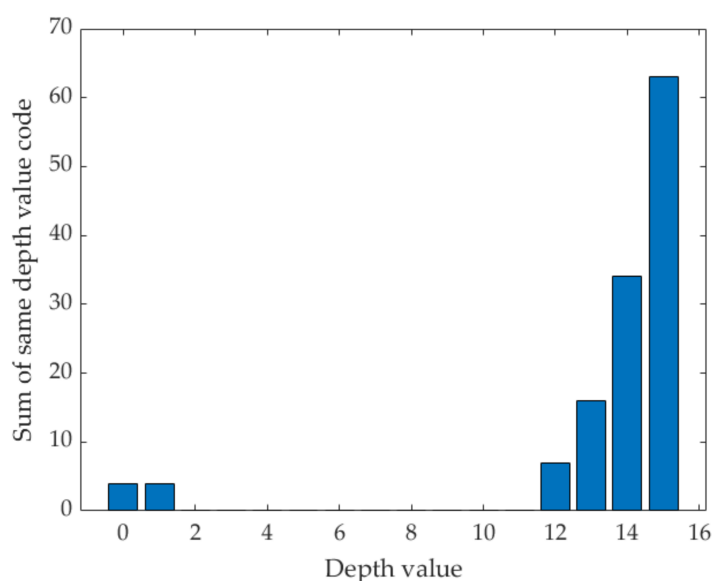


Figure 10. The depth distribution of (15, 5) BCH code.

The corresponding depth spectrum in Figure 10 is {1 12 13 14 15}, and the depth distribution is {4 400 000 000 000 7 16 34 63}. By observing the depth distribution set, the first element is the number of code words when the depth value is 0, and the second element is the number of code words when the depth value is 1 . . . , the twelfth element is the number of code words corresponding to a depth value of 11..., the last element is the number of code words corresponding to a depth value of 15. According to Theorem 2, the distribution characteristic of code words corresponding to the non-zero depth value of linear block codes is that with the increase of the non-zero depth value, the number of code words corresponding to the depth value also increases in a two-fold relationship. Due to the existence of a code error, there will be some deviation, but the relationship of two-fold depth value is still presented basically. Therefore, there are five kinds of non-zero depth values, i.e., the depth spectrum is {1 12 13 14 15}, $k = 5$, and the bit rate $t = k/n = 5/15 = 1/3$. Because the depth distribution histograms of convolutional codes and Turbo codes in this paper are relatively complex, in order to facilitate observation, the normalized depth distribution diagram will be adopted uniformly in the following text.

The normalized depth distribution is shown in Figure 11. As a comparison, we also give the normalized rank distribution diagram under the matrix rank algorithm.

As can be seen from Figure 11a, symmetrically, when the estimated value of code length is an integer multiple of the actual code length, the normalized value of code word depth distribution achieves a minimum value in every 15 codes, which is in line with the calculation result of Equation (23) above. In contrast, in Figure 11b, the matrix rank algorithm under the condition of noise cannot calculate the actual rank because of the code error, and it is always in the state of full rank. When the code length is correct, the code word distribution of linear block codes corresponding to different synchronization unknown parameters d_e is shown in Figure 12.

It can be observed from Figure 12a that, when the synchronous position is an integer multiple of the code length, the normalized value of the code word depth distribution achieves a minimum value, which conforms to the code word depth distribution characteristics of BCH (15, 5) coding. Only when the estimate of the code length is an integer multiple of the code length will a distinct waveform appear, and at the same time, the position where the minimum value of the waveform symmetrically appears corresponds to a synchronous position parameter that is an integer multiple of the code length. However, when we observe Figure 12b, we find that there is no essential difference between it and Figure 11b. This is because the code words in the matrix are still in the full rank state due to a large number of errors.

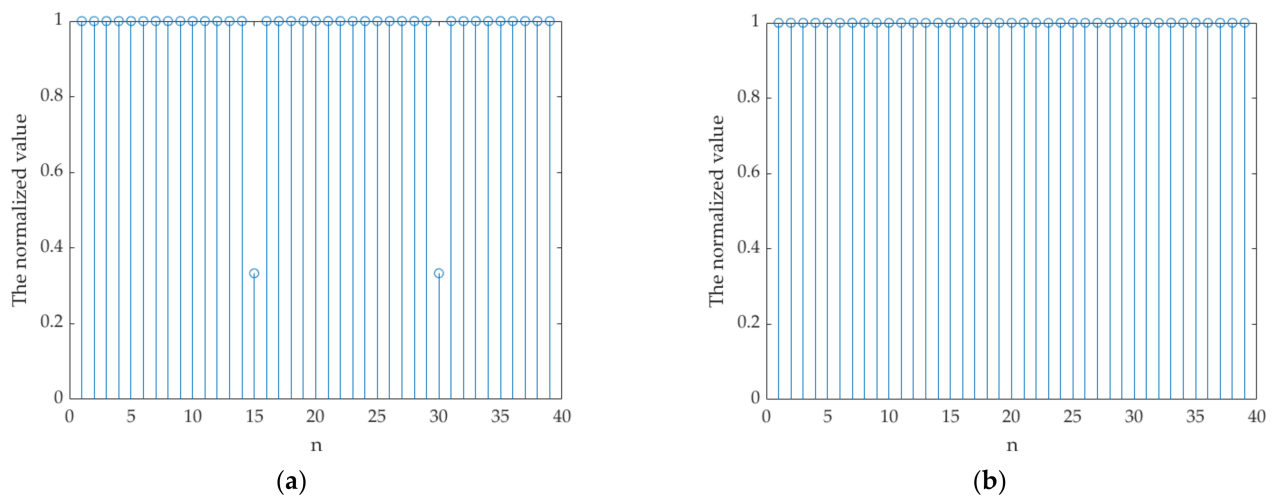


Figure 11. The distribution of (15,5) BCH code. (a) results based on depth distribution algorithm, (b) results based on matrix rank algorithm.

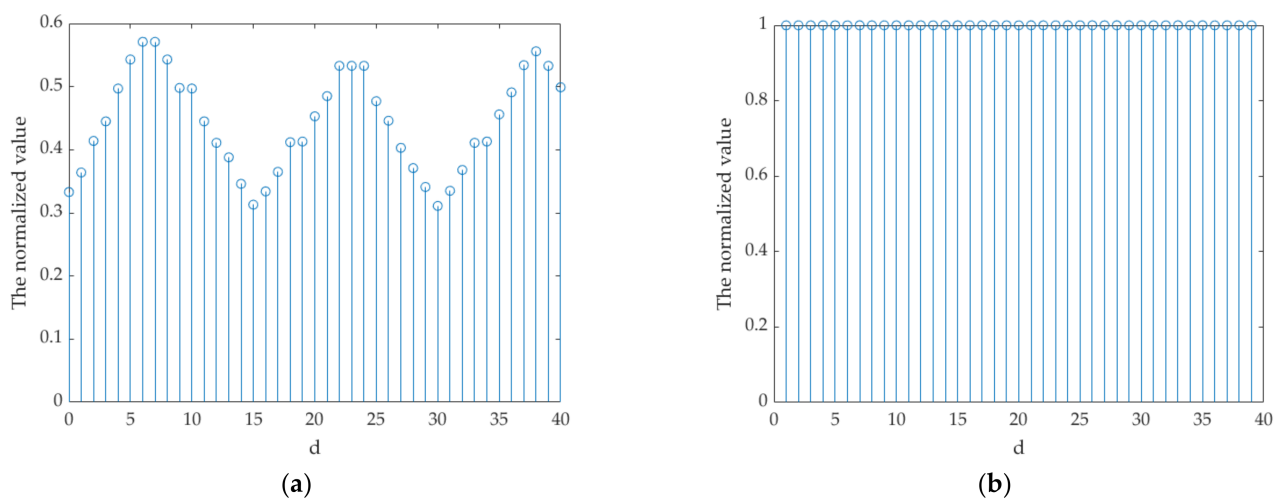


Figure 12. The distribution of different start bits. (a) results based on depth distribution algorithm, (b) results based on matrix rank algorithm.

7.3. Simulation Results of Depth Distribution Characteristics of Convolutional Codes

In this paper, the binary convolutional codes defined on $GF(2)$ with $n = 2$, $k = 1$ and $D = 6$ are taken as the research objects to analyze the depth distribution characteristics of their codewords. The sender generates random data and encodes it using the generating matrix of convolutional codes (2,1,6). The receiver obtains the codeword matrix through the noisy transmission channel. When the estimation of the synchronization position parameter d_e is correct, the distribution of the convolutional codeword matrix corresponding to different code lengths n_e is shown in Figure 13.

Combined with the encoding parameters of the convolutional code (2, 1, 6), according to Equation (34), when the parameter estimation of the synchronous position is correct, the corresponding relationship between the normalized depth distribution function and the code length of the convolutional code is completely consistent with the simulation results in Figure 13a. However, in Figure 13b of the simulation results of the matrix rank algorithm, we can observe that the overall distribution changes are disorderly. This is because, although the error-resistant performance of convolutional codes is better than that of general linear block codes, noise still causes great damage to the code-word matrix, resulting in disordered rank changes. When the estimate of code length n_e is correct, the

depth distribution of convolutional codewords corresponding to parameter d_e at different synchronous positions is shown in Figure 14.

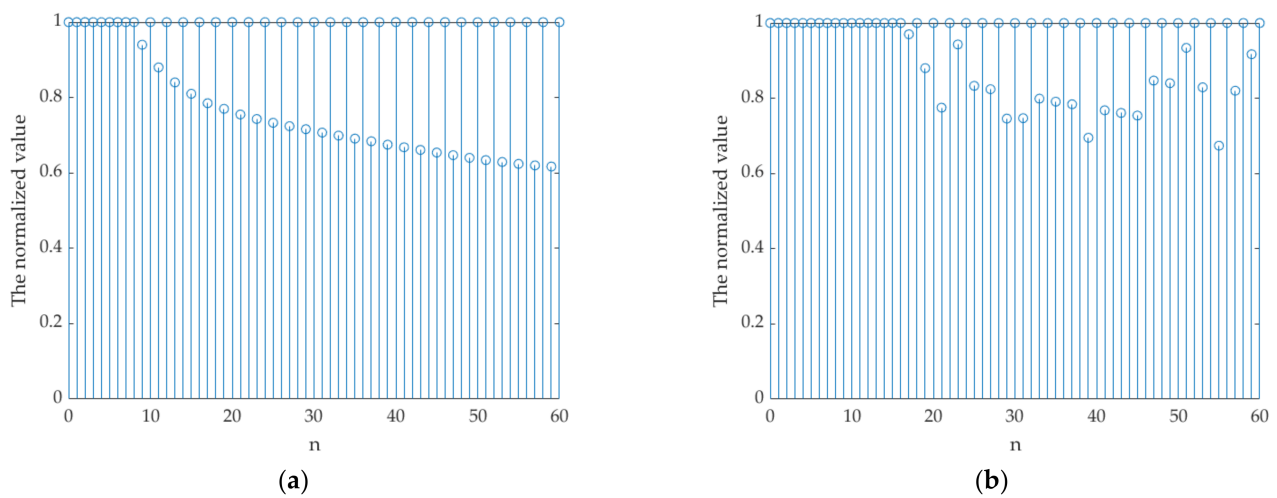


Figure 13. The distribution of different code lengths. (a) results based on depth distribution algorithm, (b) results based on matrix rank algorithm.

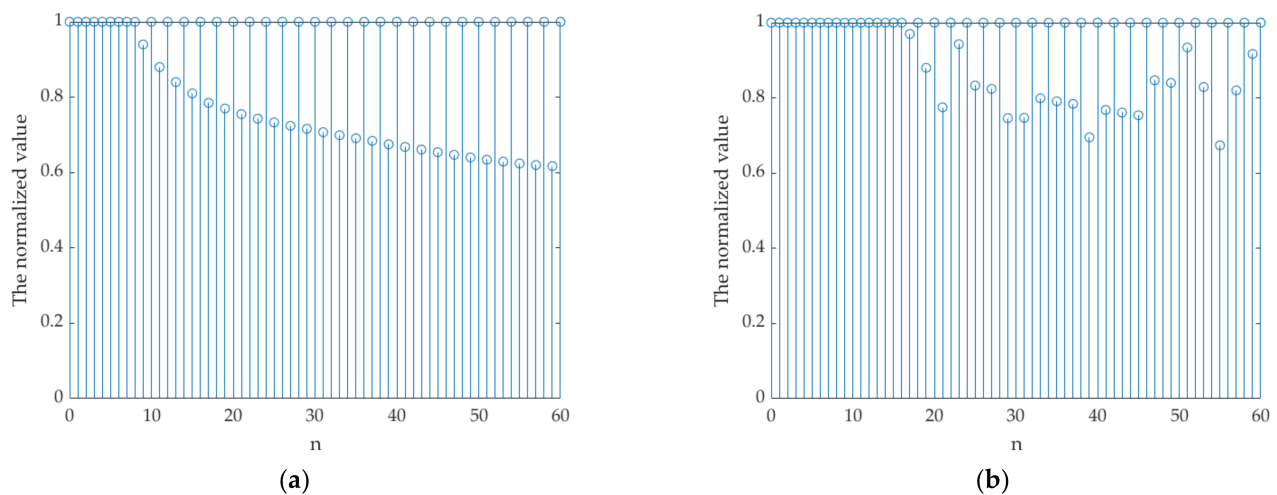


Figure 14. The distribution of different start bits. (a) results based on depth distribution algorithm, (b) results based on matrix rank algorithm. As can be seen from Figure 14a, when the synchronous position is an integer multiple of the code length, the code word normalized depth distribution value achieve a minimum value, which conforms to the code word depth distribution characteristics of convolutional code (2, 1, 6) coding. However, according to Figure 14b, the matrix is a full rank and the normalized value is 1 because of the existence of certain error codes.

7.4. Simulation Results of the Depth Distribution Characteristics of Turbo Codes

When the estimation of the synchronization position parameter d_e is correct, the code word distribution of Turbo codes corresponding to different code lengths n_e is shown in Figure 15. In combination with the encoding parameters of Equations (41)–(45) and the Turbo code, when the condition $n_e = 15\mu$ is satisfied, the condition $n_e = 9 + \varphi$ is also satisfied. Therefore, the depth distribution rule in Figure 15 is consistent with the distribution relationship expressed in Equations (41)–(45). However, in Figure 14b, the overall appearance is chaotic due to the error code factor.

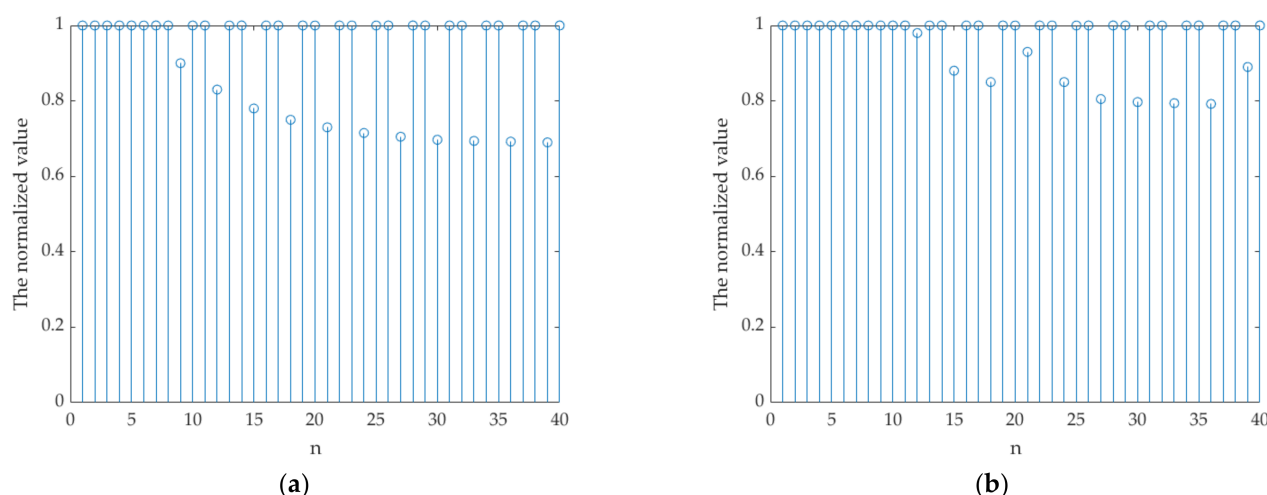


Figure 15. The distribution of different code lengths. (a) results based on depth distribution algorithm, (b) results based on matrix rank algorithm. When the code length estimation n_e is correct, the depth distribution d_e of the Turbo code word matrix corresponding to different synchronous position parameters is shown in Figure 16a. However, in Figure 14b, the normalized value of rank is always 1 due to code errors.

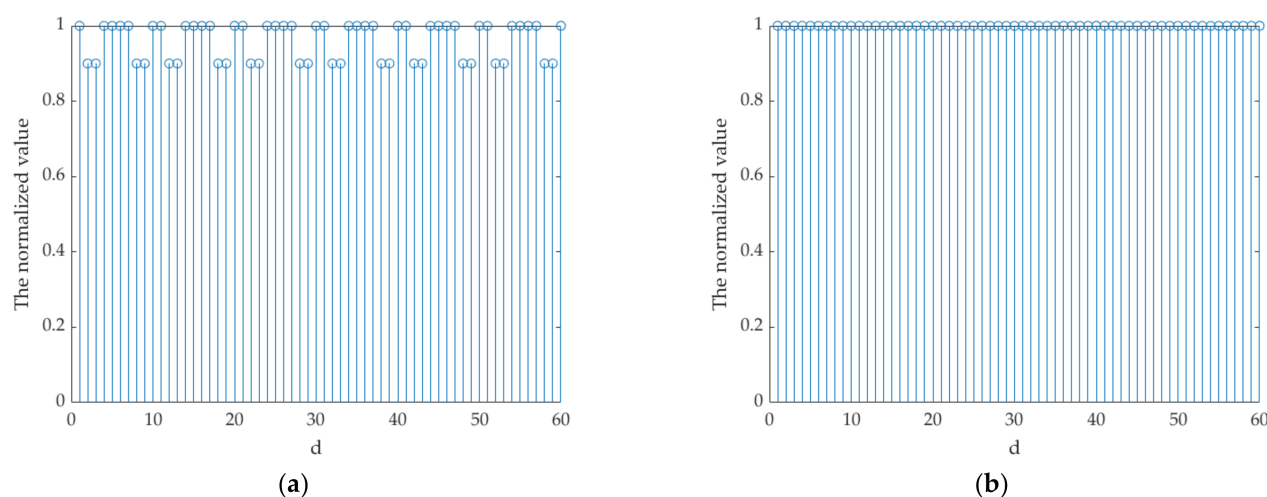


Figure 16. The distribution of different start bits. (a) results based on depth distribution algorithm, (b) results based on matrix rank algorithm. The recognition and verification of the three types of codewords based on the depth distribution algorithm is completed. We can observe from the above experiments that noise has a great impact on the matrix rank algorithm, and the error code will lead to a significant increase in the matrix rank. However, the depth distribution algorithm can effectively avoid the interference caused by noise and complete the identification of different forward error correction coding types.

8. Conclusions

The paper introduces different code type recognition based on a depth distribution algorithm. Based on linear block code recognition algorithm, this paper first introduces the recognition method based on an improved Gaussian elimination of matrix of linear block codes to estimate the code length and the starting bit. Compared with the traditional matrix estimation algorithm, the algorithm adopted in the paper has a better anti-noise performance, and compared with other parameter estimation algorithm, the algorithm is more simple and more convenient for engineering practice.

Secondly, the depth distribution algorithm, a mathematical algorithm, is used in this paper to complete the recognition of different code types. Compared with the traditional rank estimation method, which cannot complete the code type recognition under the

condition of a low bit error rate, the algorithm used in this paper has a good anti-noise performance and is more practical.

What is more, we used a general framework to unify all reconstruction algorithms to complete the reconstruction of convolutional codes and Turbo codes.

The experimental results show that using a depth distribution algorithm to identify different types of forward error correction coding is a promising classification technology, which is suitable for the recognition of different types of communication signals under the condition of noise.

Author Contributions: Conceptualization, F.M. and Y.L.; methodology, F.M.; software, F.M.; data curation, F.M. and H.C.; writing—original draft preparation, F.M.; visualization, F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their thanks to Lu Chen from the National University of Defense Technology for her valuable comments on this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goldsmith, A.J.; Chua, S. Adaptive coded modulation for fading channels. *IEEE Trans. Commun.* **1998**, *46*, 595–602. [\[CrossRef\]](#)
2. Moosavi, R.; Larsson, E.G. A fast scheme for blind identification of channel codes. In Proceedings of the IEEE Global Telecommunications Conference, Houston, TX, USA, 5–9 December 2011; pp. 1–5.
3. Debessu, Y.G.; Wu, H.-C.; Jiang, H. Novel blind encoder parameter estimation for Turbo codes. *IEEE Commun. Lett.* **2012**, *16*, 1917–1920. [\[CrossRef\]](#)
4. Xia, T.; Wu, H. Identification of non-binary LDPC codes using average LLR of syndrome a posteriori probability. *IEEE Commun. Lett.* **2013**, *17*, 1301–1304.
5. Zrelli, Y.; Gautier, R.; Rannou, E.; Marazin, M.; Radoi, E. Blind identification of the code word length for non-binary error-correcting codes in noisy transition. *EURASIP J. Wirel. Commun. Netw.* **2015**, *43*, 1–6.
6. Bonvard, A.; Houcke, S.; Gautier, R.; Marazin, M. Classification based on Euclidean distance distribution for blind identification of error correcting codes in noncooperative contexts. *IEEE Trans. Signal Process.* **2018**, *66*, 2572–2583. [\[CrossRef\]](#)
7. Valembois, A. Detection and recognition of a binary linear code. *Discret. Appl. Math.* **2001**, *111*, 199–218. [\[CrossRef\]](#)
8. Yardi, A.D.; Vijayakumaran, S.; Kumar, A. Blind reconstruction of binary cyclic codes from unsynchronized bitstream. *IEEE Trans. Commun.* **2016**, *64*, 2693–2706. [\[CrossRef\]](#)
9. Ramabadran, S.; Madhukumar, A.S.; Teck, N.W.; See, C.M.S. Parameter estimation of convolutional and helical interleavers in a noisy environment. *IEEE Access* **2017**, *5*, 6151–6167. [\[CrossRef\]](#)
10. Wengu, C.; Guoqing, W. Blind recognition of $(n-1)/n$ rate punctured convolutional encoders in a noisy environment. *J. Commun.* **2015**, *10*, 260–267.
11. Marazin, M.; Gautier, R.; Burel, G. Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bitstream. *IET Signal Process.* **2012**, *6*, 122–131. [\[CrossRef\]](#)
12. Côte, M.; Sendrier, N. Reconstruction of a turbo-code interleaver from noisy observation. In Proceedings of the 2010 IEEE International Symposium on Information Theory, Austin, TX, USA, 13–18 June 2010; pp. 2003–2007.
13. Barbier, J.; Letessier, J. Forward error correcting codes characterization based on rank properties. In Proceedings of the IEEE International Conference on Wireless Communications & Signal Processing, Nanjing, China, 13–15 November 2009; pp. 1–5.
14. Xinhao, L.; Min, Z.; Yingchuan, S.; Quan, Y. Blind Identifying of Type of Linear Block Code and Convolutional Code Based on Run Feature. *J. Data Acquis. Proc.* **2015**, *30*, 1205–1214.
15. Yisen, L. *Research on the Technology of Turbo Coding Recognition in Non-Cooperative Communication*; Xidian University: Xi'an, China, 2014.
16. Jing, Z. *Research of Recognition and Analyzation Methods of Channel Codes*; National University of Defense Technology: Changsha, China, 2014.
17. Swaminathan, R.; Madhukumar, A.S. Classification of error correcting codes and estimation of interleaver parameters in a noisy transmission environment. *IEEE Trans. Broadcast.* **2017**, *63*, 463–478. [\[CrossRef\]](#)
18. Kwon, S.; Shin, D.J. Blind Classification of Error-Correcting Codes for Enhancing Spectral Efficiency of Wireless Networks. *IEEE Trans. Broadcast.* **2021**, 1–13. [\[CrossRef\]](#)

19. Wang, J.; Li, J.; Huang, H.; Wang, H. Fine-grained recognition of error correcting codes based on 1-D convolutional neural network. *Digit. Signal Process.* **2020**, *99*, 102668. [[CrossRef](#)]
20. Li, S.; Zhou, J.; Huang, Z.; Hu, X. Recognition of error correcting codes based on CNN with block mechanism and embedding. *Digit. Signal Process.* **2021**, *111*, 102986. [[CrossRef](#)]
21. Etzion, T. The depth distribution—A new characterization for linear codes. *IEEE Trans. Inform. Theory* **1997**, *43*, 1361–1363. [[CrossRef](#)]
22. Jingli, T. *Research on Blind Recognition Technology for Linear Block Code*; Hebei University: Hebei, China, 2018.
23. Sicot, G.; Houcke, S.; Barbier, J. Blind detection of interleaver parameters. *Signal Process.* **2009**, *89*, 450–462. [[CrossRef](#)]
24. Marazin, M.; Gautier, R.; Burel, G. Blind recovery of rate convolutional encoders in a noisy environment. *EURASIP J. Wirel. Commun. Netw.* **2015**, *168*, 1–9. [[CrossRef](#)]
25. Chao, L.; Jin, G. *Depth Distribution of Linear Block Codes over Finite Fields*; National University of Defense Technology: Changsha, China, 2006.
26. Daofu, Z. *Research on Depth Distribution of Linear Codes and Generalized Derivatives of Binary Sequences*; Hefei University of Technology: Anhui, China, 2008.
27. Liye, S. *Depth Distribution of Cyclic Codes*; Central China Normal University: Hubei, China, 2007.
28. Mitchell, C.J. On integer-valued rational polynomials and depth distributions of binary codes. *IEEE Trans. Inform. Theory* **1998**, *44*, 3146–3150. [[CrossRef](#)]
29. Zhentao, Z.; Yixian, Y.; Zhengming, H. Research on Depth Distribution of Linear Codes. *J. Beijing Univ. Posts Telecommun.* **2002**, *25*, 68–72.
30. Zhang, Y.G.; Lou, C.Y.; Wang, T. A Blind Parameters Identification Method for Linear Block Code. CN201010131103, 18 June 2014.
31. Cluzeau, M. Block code reconstruction using iterative decoding techniques. In Proceedings of the IEEE International Symposium on Information Theory, Seattle, WA, USA, 9–14 July 2006; pp. 2269–2273.
32. McEliece, R.; MacKay, D.; Cheng, J. Turbo decoding as an instance of Pearl’s belief propagation algorithm. *IEEE J. Sel. Areas Commun.* **1998**, *16*, 140–152. [[CrossRef](#)]
33. Dandan, L. *Research on Blind Recognition of Turbo Codes and Compressed Sensing System Based on LDPC Codes*; Shandong University: Jinan, China, 2014.
34. Chen, J.H.; Dholakia, A.; Eleftheriou, E.; Fossorier, M.P.C.; Hu, X.-Y. Reduced-Complexity Decoding of LDPC Codes. *IEEE Trans. Commun.* **2005**, *53*, 1288–1299. [[CrossRef](#)]
35. Fossorier, M.P.C.; Mihaljevic, M.; Imai, H. Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation. *IEEE Trans. Commun.* **1999**, *47*, 673–680. [[CrossRef](#)]