

Received August 27, 2020, accepted September 16, 2020, date of publication September 18, 2020,
date of current version September 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025177

An Improved Blind Recognition Method for Synchronization Position and Coding Parameters of k/n Rate Convolutional Codes in a Noisy Environment

YONG DING¹, ZHIPING HUANG¹, AND JING ZHOU¹

College of Artificial Intelligence, National University of Defense Technology, Changsha 410073, China

Corresponding author: Zhiping Huang (kdhuangzp@163.com)

This work was supported by the National Natural Science Foundation of China (General Program) under Grant 51575517.

ABSTRACT In digital communication systems, error correction codes are used to recover data that has been disturbed in a noisy environment. Convolutional code is a commonly used error correction code. In non-cooperative communication systems, it is very meaningful to blindly identify the synchronization position and coding parameters of convolutional codes. Existing reference had proposed a blind recognition algorithm for coding parameters of k/n rate convolutional codes in a noisy environment, but it didn't take into account the case where the received code words were not synchronized. In this paper, this problem is improved by using the quasi-correlation between the receive sequence and the check matrix to estimate the synchronization position, while obtaining the check matrix. It implements blind recognition of the synchronization position and coding parameters of k/n rate convolutional codes in a noisy environment. Simulation experimental results show that the improved algorithm in this paper is effective.

INDEX TERMS Blind recognition, coding parameters, convolutional code, non-cooperative communication, synchronization position.

I. INTRODUCTION

In digital communication systems, error correction coding is the most effective method to prevent the destruction of information. Convolutional codes are another important branch of error-correcting codes. Along with linear block codes, they have become one of today's most dominant error correction codes, especially in wireless and spatial communication systems. Among them, the application of binary convolutional codes is usually dominant. At the same time, blind recognition of binary convolutional code parameters has also received attention in non-cooperative communication environments. At present, there have been many research achievements in the recognition and analysis of coding parameters of convolutional codes in the public reference. A $1/n$ code rate convolutional code recognition method under error-free conditions was proposed in [1], which was generalized to arbitrary code rates in [2], [3]. In [4], a fast merging method

based on Berlekamp-Massey algorithm was proposed. The method was fault-tolerant, but its computational complexity was relatively high, and it could only be used for blind recognition of $1/2$ rate convolutional codes. In [5], the Euclidean recognition method was used to identify $1/2$ rate convolutional codes in a noisy environment. This algorithm had a certain error tolerance performance, but it was difficult to generalize to arbitrary code rates. An iterative recognition method of $1/n$ rate convolutional codes based on the expectation-maximization algorithm was proposed in [6]. The algorithm didn't provide a method to identify the code length and the starting point of the codeword, and the algorithm had a high computational complexity. It was also difficult to extend this algorithm to arbitrary code rate. A Gauss-Jordan elimination through pivoting (GJETP) algorithm based on algebraic properties of convolutional codes [7] and the structure of dual codes [8] was proposed in [9] for parameter recognition of $(n-1)/n$ rate convolutional codes. It had good fault tolerance and low computational complexity. Furthermore, the research result of [9] was extended to the

The associate editor coordinating the review of this manuscript and approving it for publication was Rui Wang¹.

case of k/n rate in [10], and the parameter recognition method of convolutional codes at arbitrary code rates was better achieved with lower computational complexity. This was also the first proposed algorithm that can completely realize the blind recognition of k/n convolutional codes in a noisy environment. The authors of [11] studied the feasibility of using matrix rank criterion to identify convolutional code parameters, and emphasized the influence of the generator matrix on the rank deficit. In [12], the relationship between rank and convolutional code constraint length was revealed. The GJETP algorithm for blind recognition of convolutional interleaving parameters was improved in [13]. It can be seen that, for the blind recognition problem of k/n rate convolutional codes in a noisy environment, the GJETP algorithm can be applied to parameter estimation. However, the GJETP algorithm in [10] didn't consider the synchronization problem of convolutional codes, so it could only be used in the case of frame synchronization.

Before decoding the received sequence, the first bit of the codeword must be known. This step is called frame synchronization. The traditional frame synchronization method was to add the synchronization word sequence before the transmission sequence, and then the receiving side used the correlator [14] to detect the synchronization word. If the channel conditions were relatively poor, in order to maintain good performance, the traditional method was to increase the length of the synchronization word to solve the problem, but this greatly reduced the spectrum efficiency of the system. Many researchers had begun to study the method of no synchronization word to achieve frame synchronization. The authors of [15] introduced a frame synchronization method with the best maximum a posteriori probability (MAP). The method was based on the low-density parity check characteristics of the code, and didn't require the insertion of synchronization words. The authors of [16] found an important detail error in the algorithm in [15] and improved the method. The improved method was suitable for blind frame synchronization of low-density parity-check codes (LDPC) and convolutional codes with sparse check matrix codes, and the performance was relatively good. Therefore, when blindly identifying the coding parameters of the convolutional code, the frame synchronization problem without synchronization words must be considered. The authors in [17] improved on the method in [10], and proposed a method for identifying synchronization positions and parameters. However, the algorithm only achieved the recognition of synchronized positions and encoding parameters for some types of convolutional codes, and the recognition accuracy is low for other types of convolutional codes. The details will be explained in section IV below. In this paper, an improved method is proposed to blindly recognize the frame synchronization positions and encoding parameters of all types of k/n convolutional codes in a noisy environment.

The rest of the paper is organized as follows. Section II introduces the coding principle and features of convolutional codes. Section III briefly describes the algorithm for

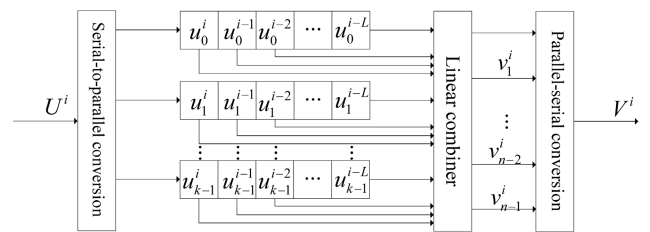


FIGURE 1. Convolutional code structure.

identifying the coding parameters of k/n convolutional codes under noisy channels in [10] and points out the shortcomings of the algorithm. Section IV introduces the improved method proposed in this paper. Section V verifies the effectiveness of the improved method for blind recognition of the synchronization position and coding parameters of convolutional codes by simulation experiments.

II. THE CODING PRINCIPLE AND CHARACTERISTICS OF CONVOLUTIONAL CODES

The constraint relationship of the convolutional code spans multiple code words, and each code word of the encoded output depends not only on the current set of input information, but also on previous sets of input information. Because the length of the constraint relationship is limited, convolutional codes can be regarded as a finite memory system. The general structure of convolutional code (n, k, L) is shown in Figure 1, where n denotes the code length of codeword and k denotes the information bit length. L denotes the constraint length, meaning that the current encoded output depends on the current set of input information and previous L sets of input information.

As can be seen in Figure 1, the convolutional code first performs serial-parallel conversion of the input information sequences and then stores them in the $k \times (L + 1)$ shifter group. The leftmost column of the shifter is the current set of input information, and the columns that follow are the sets of previously entered information. The data in the shift registers are combined linearly, and then the code words are output by parallel-serial conversion.

The mathematical description of the convolutional code is $V = UG$, where $U = (U^0, U^1, U^2, \dots, U^i, \dots)$ denotes the input information sequence with information bit length k , and $U^i = (u_0^i, u_1^i, \dots, u_{k-1}^i)$ denotes the k bit information input at the time i . $V = (V^0, V^1, V^2, \dots, V^i, \dots)$ denotes the coded output sequence with codeword length n , and $V^i = (v_0^i, v_1^i, \dots, v_{n-1}^i)$ denotes the n bit codeword output at the time i . $G = (G^0, G^1, G^2, \dots, G^L)$ denotes the influence of the input information sequence on the encoded output, where (1)

$$G^i = \begin{bmatrix} g^i(0, 0) & g^i(0, 1) & \dots & g^i(0, n-1) \\ g^i(1, 0) & g^i(1, 1) & \dots & g^i(1, n-1) \\ \vdots & \vdots & \ddots & \vdots \\ g^i(k-1, 0) & g^i(k-1, 1) & \dots & g^i(k-1, n-1) \end{bmatrix} \quad (1)$$

denotes the effect of the input sequence at time i on the output. It can be seen that the input sequence $U = (U^0, U^1, U^2, \dots, U^i, \dots)$ and the output sequence $V = (V^0, V^1, V^2, \dots, V^i, \dots)$ are both semi-infinite sequences. Their relationship can be expressed as $V = UG_\infty$, where G_∞ can be written as (2).

$$G_\infty = \begin{bmatrix} G^0 & G^1 & \dots & G^L & 0 & 0 & 0 \\ 0 & G^0 & G^1 & \dots & G^L & 0 & 0 \\ 0 & 0 & G^0 & G^1 & \dots & G^L & 0 \\ 0 & 0 & 0 & \dots & \dots & \dots & \dots \end{bmatrix} \quad (2)$$

It can be seen from (2) that each row is obtained by shifting $G = (G^0, G^1, G^2, \dots, G^L)$ to the right, so G is denoted the basic generator matrix. The entire convolutional code is obtained by summing the matrices G after linear shifting. The basic generator matrix of convolutional codes can also be written in the form of $G(D)$. Therefore, the relationship between the input information sequence and the output code-words can be expressed as $V(D) = U(D)G(D)$, where $G(D)$ is (3).

$$G(D) = \begin{bmatrix} g_{0,0}(D) & g_{0,1}(D) & \dots & g_{0,n-1}(D) \\ g_{1,0}(D) & g_{1,1}(D) & \dots & g_{1,n-1}(D) \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0}(D) & g_{k-1,1}(D) & \dots & g_{k-1,n-1}(D) \end{bmatrix} \quad (3)$$

The convolutional code is also linear, and the generating matrix and the check matrix satisfy the relationship $G_\infty H_\infty^T = 0$. The general form of the check matrix for (n, k, L) convolutional codes is (4).

$$H_\infty = \begin{bmatrix} h_0 & 0 & \dots & & & \\ h_1 & h_0 & 0 & \dots & & \\ h_2 & h_1 & h_0 & 0 & \dots & \\ \dots & \dots & \dots & \dots & \dots & \\ h_L & h_{L-1} & \dots & h_1 & h_0 & \dots \\ 0 & h_L & h_{L-1} & \dots & h_1 & h_0 \\ 0 & 0 & & \dots & \dots & \dots \end{bmatrix} \quad (4)$$

In (4), h_i is a matrix of size $(n - k) \times n$. The row $L + 1$ of H_∞ is the basic parity check matrix of the convolutional code, denoted as $h_\infty = [h_L \ h_{L-1} \ \dots \ h_0]$.

It can be seen from [17] that the form of the basic check matrix $H(D)$ can be written as (5),

$$H(D) = \begin{bmatrix} h_{0,0}(D) & \dots & h_{0,k-1}(D) & h_0(D) \\ \vdots & \ddots & \vdots & \vdots \\ h_{n-k-1,0}(D) & \dots & h_{n-k-1,k-1}(D) & h_0(D) \end{bmatrix} \quad (5)$$

where $h_0(D) = h_0(0) + h_0(1)D + \dots + h_0(L)D^L$ and $h_0(0) = 1$.

The coded output sequence and the check matrix are consistent with the relation $VH_\infty^T = 0$, thus giving

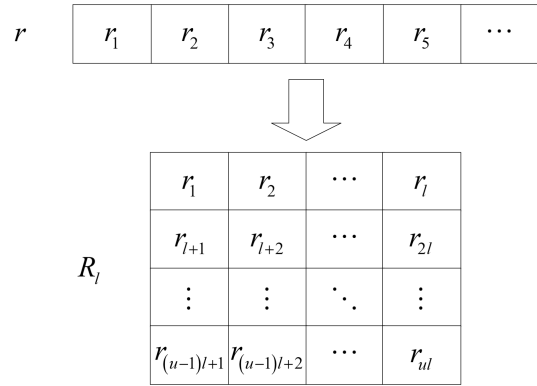


FIGURE 2. Generation of the observed matrix R_l .

$V(D)H^T(D) = 0$, which can be written as (6),

$$[v_0(D) \ \dots \ v_{k-1}(D) \ v_{k-1+s}(D)] \begin{bmatrix} h_{s,0}(D) \\ \vdots \\ h_{s,k-1}(D) \\ h_0(D) \end{bmatrix} = 0 \quad (6)$$

where $s = 1, 2, \dots, n - k$. From (6), it can be seen that the basic parity matrix can be obtained based on the linear relationship between the received convolutional code words. The generator matrix and other coding parameters are then obtained.

III. PREVIOUS BLIND RECOGNITION ALGORITHM FOR CODING PARAMETERS OF CONVOLUTIONAL CODES

The algorithm in [10] had realized the blind recognition of the coding parameters of the k/n rate convolutional code, and the computational complexity was low. The algorithm is mainly divided into two steps. The first step is to identify the code length of the convolutional code, and the second step is to identify the code rate, the constraint length and the check matrix of the convolutional code. This section briefly introduces the blind recognition algorithm in [10], as shown below.

A. CODE LENGTH RECOGNITION

Starting at any moment, the received sequence r is sequentially filled into the matrix R_l of size $(u \times l)$. The filling process is shown in Figure 2. It is assumed that the input information sequence sent to the encoder is random and u is large enough. When the storage matrix satisfies the following conditions.

1. The number of columns in R_l is equal to an integer multiple of the code length, i.e., $l = \alpha n$.
2. α is greater than the threshold value $\lfloor L/(n - k) + 1 \rfloor$.
3. The first bit filled in in R_l is exactly the first bit of a code word.
4. No errors occurred during the transfer.

Then, the rank of the storage matrix R_l is equal to $rank(R_l) = \alpha k + L$. Otherwise, the rank of R_l is l .

In the actual communication system, the ideal state without noise does not exist. The above analysis is only to provide ideas for the recognition and analysis of convolutional codes in a noisy environment. Received sequences obtained from noisy channels usually contain error codes, so it is necessary to investigate whether there are certain columns in R_l that are linearly related to other columns with probability. The GJETP method is a good solution to this problem. Given a storage matrix R_l , an elementary transformation of the matrix R_l is performed to output the lower triangular matrix X . The matrices A and B record the row and column transformations of the matrix R_l during elimination, respectively, satisfying the relation $X = AR_lB$.

When the receiving sequence is error-free, the storage matrix R_l is processed by the GJETP method to obtain the lower triangular matrix X . The lower half of the matrix X starting from row $l + 1$ is denoted as X_{low} . If some columns of X_{low} are all equal to zero, it means that these columns of the matrix R_l can be represented as linear combinations of other columns. These columns are called linear correlated columns, while the other columns are called linear independent columns, which are the largest group of linear independent column vectors of the matrix R_l .

In noisy channels where there are errors in the receive sequence, these columns may not be all-zero columns, but some columns have extremely small Hamming weights. In this paper, these columns are referred to as quasi-linear correlation columns, while other columns are referred to as quasi-linear independent columns. Thus, quasi-linear correlation columns can be distinguished in the presence of error codes. According to the analysis in [10], the GJETP transformation of the matrix R_l containing the error code is performed to obtain the lower triangular matrix X . The Hamming weight expectation of the quasi-linear independent columns of the matrix X_{low} is half of the number of rows of X_{low} , and the mathematical expectation of the Hamming weight of the quasi-linear correlated columns is almost close to zero.

The algorithm in [10] ignored the codeword synchronization problem. In this case, the algorithm in [10] proposed to make the parameter l of the storage matrix traverse a relatively wide range. After GJETP transformation of matrix R_l , the lower triangular matrix X is obtained. If there are columns in X_{low} with Hamming weights less than the threshold β_{opt} [6], then these columns are quasi-linearly correlated columns, recording the current l . Then according to condition 1, the smallest of all recorded values of l is n_a and the difference between two adjacent l is the estimated code length.

B. RECOGNITION OF CHECK MATRIX

To identify the coding parameters of the convolutional code, first identify the check matrix, and then derive other parameters such as the code rate and the generation matrix. The recognition of the check matrix of the convolutional code is the recognition of its basic check matrix h_∞ . Therefore,

according to algorithm 1 and algorithm 2 in [10], based on the code lengths n and n_a that have been analyzed, the code rate of the convolutional code and $n - k$ check vectors can be identified by the following steps.

Step 1. Set the maximum value and minimum values of the information bit length k to k_{max} and k_{min} , where $k_{max} = n - 1$ and $k_{min} = 1$. k is initialized to 1. Then execute Step 2.

Step 2. Set the maximum and minimum values of the rank deficit value Z to Z_{max} and Z_{min} , where $Z_{max} = n - k$ and $Z_{min} = 1$. Z is initialized to 1. According to [10], the convolutional code constraint length $L = n_a/n \times (n - k) - Z$. Then execute Step 3.

Step 3. Set the number of rows of the check matrix to s with a maximum value of $s_{max} = n - k$ and a minimum value of $s_{min} = 1$. s is initialized to 1. The received sequence is grouped according to the length of n bits, taking the first k bits and the $k + s$ th bit from each group. The resulting sequence of bits is rearranged in grouping order into a vector r , and the form of r is $r = (r_1, \dots, r_k, r_{k+s}, \dots, r_{n+1}, \dots, r_{n+k}, r_{n+k+s}, \dots)$. Then execute Step 4.

Step 4. Arrange r into a matrix R_l^s of size $(u \times l)$, where $l = (L + 1)(k + 1)$ and $u = 2l_{max}$. Then execute Step 5.

Step 5. Perform lower triangular transformation on R_l^s , and get $X^s = A^s R_l^s B^s$. Then execute Step 6.

Step 6. Let the variable j be incremented from 1 to l . If $W(j) < (u - l)/2 \times \beta_{opt}$ and $B^s(l, j) = 1$. Then the j th column of B^s is the s th quasi-check vector of the convolutional code. $W(j)$ denotes the Hamming weight of the j -th column vector of matrix X_{low}^s . β_{opt} is the optimal threshold, which is described in detail in [6]. $W(j) < (u - l)/2 \times \beta_{opt}$ implies that the j th column of R_l^s has a high probability of depending on other columns. Moreover, $W(j) = wt(R_l^s \times B^s(j))$, where $wt(b)$ denotes the Hamming weight of vector b . From (5), it can be seen that $h_0(0) = 1$, so the last bit of the check vector must be 1. Therefore, when the conditions $B^s(l, j) = 1$ and $W(j) < (u - l)/2 \times \beta_{opt}$ are satisfied, $B^s(j)$ is likely to be the s th quasi-check vector of the convolutional code. Then execute Step 7.

Step 7. If $s < s_{max}$, then set $s = s + 1$, and execute Step 3. Otherwise, execute Step 8.

Step 8. If $Z < Z_{max}$, then set $Z = Z + 1$, and execute Step 2. Otherwise, execute Step 9.

Step 9. If $k < k_{max}$, then set $k = k + 1$, and execute Step 1. Otherwise, the calculation ends.

The above steps of blind recognition of coding parameters can be implemented by Algorithm 1 in this paper. It can be seen from Algorithm 1 in this paper that the two algorithms in [10] are combined in this paper to make it easier to implement. The traversal of the column values l of the matrix R_l^s in Algorithm 2 in [10] is redundant, so the removal of this step in this paper makes Algorithm 1 in this paper much less computationally intensive and more efficient for the blind recognition of the encoding parameters. In this paper's Algorithm 1, once a valid quasi-check matrix is obtained for a certain combination of k and L , the corre-

Algorithm 1 GJETP algorithm**Input:** Receive codeword sequence r, n, n_a **Output:** k, L , check matrix

```

1: for  $k = 1$  to  $n - 1$  do
2:   for  $Z = 1$  to  $n - 1$  do
3:      $L = n_a/n(n - k) - Z$ ;
4:     for  $s = 1$  to  $n - k$  do
5:        $r = (r_1, \dots, r_k, r_{k+s}, \dots)$ ;
6:        $l = (L + 1)(k + 1)$ ;
7:       Build matrix  $R_l^s$  of size  $(u \times l)$  with  $r$ ;
8:        $R_l^s \rightarrow X^s = A^s R_l^s B^s$ ;
9:       for  $j = 1$  to  $l$  do
10:        if  $W(j) < (u - l)/2 \times \beta_{opt}$  and  $B^s(l, j) = 1$ 
11:          then
12:             $h_s = B^s(j)^T$ ;
13:            Output  $k, L$ ;
14:          end if
15:        end for
16:      end for
17:    end for

```

sponding k and L are used as estimates of the information bit length and constraint length of the convolutional code. The quasi-check matrix estimated in [10] is not the true check matrix of the convolutional code, and some transformations of the quasi-check matrix are required to obtain the check matrix. The specific transformations are shown in (12) and (13) below.

IV. AN IMPROVED RECOGNITION METHOD OF SYNCHRONIZATION POSITION AND CODING PARAMETERS OF CONVOLUTIONAL CODES IN A NOISY ENVIRONMENT

The GJETP method in [10] is a simple and effective method for blind recognition of k/n rate convolutional codes in a noisy environment. However, it should be noted that the algorithm in [10] reached the above conclusion without considering the code word synchronization. In a non-cooperative communication system without a synchronized word, the synchronization position may not be obtained in advance of obtaining the coding parameters. When the synchronization position is incorrect, although the estimation of the code length is not greatly affected, the recognition of the code rate and the check matrix will fail due to the increase of linear independent columns.

The algorithm was improved in [17] to achieve blind recognition of the coding parameters of partially typed convolutional codes when the receiving sequence was not synchronized. The idea behind the improved algorithm is that if the convolutional code words are not synchronized, then the smallest column value n_a when the observation matrix has a rank deficit is n larger than the value at synchronization. The receive sequence is rearranged into a matrix of size $u \times (n_a - n)$. The re-created observation matrix is processed

by GJETP to obtain a lower triangular matrix X , where no quasi-linear correlation column exists in the observation matrix. If the received sequences intercepted starting at bit t are rearranged into an observation matrix and there is at least one quasi-linear correlation column in the observation matrix, then position t is the synchronization position of the convolutional code. Once the synchronization position is obtained, Algorithms 1 and 2 in [10] can be used to correctly estimate the check matrix.

However, the synchronization algorithm in [17] works only for some types of convolutional codes, but not for others. For example, the generating polynomial for a convolutional code is (7).

$$G_1(D) = \begin{bmatrix} 1 & 0 & D+1 \\ 0 & 1 & D^2+1 \end{bmatrix} \quad (7)$$

When the receive sequence is synchronized and the bit error rate is equal to 0.0089. Arrange the sequences into a matrix of size $(u \times l)$, where $u = 2l_{max}$. Traverse the column values l of the matrix R_l and perform GJETP processing on the matrix R_l to obtain the lower triangular matrix X . If there are quasi-correlated columns in X , the current l is recorded. Figure 3 shows the number of quasi-linear correlation columns $N(l)$ corresponding to different values of l when traversing l from 1 to 30. It can be seen from Figure 3 that the code length $n = 3$ and the minimum number of columns $n_a = 9$ in which the observation matrix has a rank deficit. Algorithms 1 and 2 in [10] yield $k = 2$ and $L = 2$, and the check vector is $[0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1]$, so the check matrix is $[D+1, D^2+1, 1]$. Therefore, the check vector of the convolutional code is correctly identified upon receiving the sequence synchronization by the processing of Algorithm 1 and Algorithm 2 in [10]. However, when the received sequences are not synchronized, for example, the first bit of the received sequence is the second bit of a certain code word. Arrange the received sequences into a matrix of size $(u \times l)$. The column values l of the matrix R_l are traversed, and the matrix R_l is processed by GJETP to obtain the lower triangular matrix X . The corresponding $N(l)$ is obtained, as shown in Figure 4. Thus, the code length $n = 3$ and the minimum number of columns $n_a = 9$ in which the observation matrix has a rank deficit, neither of which changes. However, by Algorithm 1 and Algorithm 2 in [10], only a wrong check vector $[1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0]$ is obtained, so k and L cannot be estimated. Therefore, the algorithm in [10] cannot estimate the encoding parameters of the convolutional code when the convolutional code is not synchronized.

The frame synchronization algorithm in [17] is used to identify convolutional code (7). Moreover, the received sequence is not synchronized, and the first bit of the received sequence is set as the second bit of a codeword. When $n = 3$ and $n_a = 9$ are obtained, the received sequence is arranged into a matrix R_l of size $u \times (n_a - n)$, and GJETP transformation is performed to obtain a lower triangular matrix X . At this time, there is no quasi-linear correlation column in X .

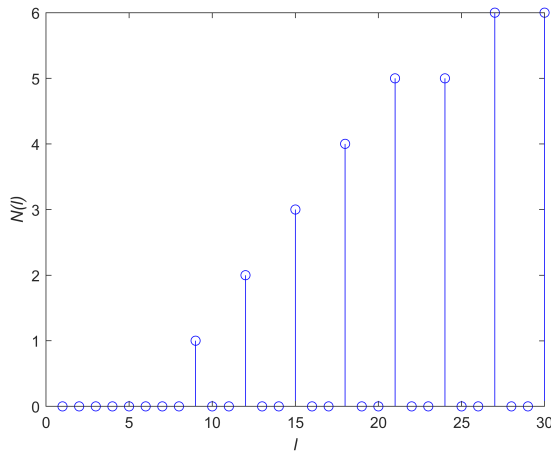


FIGURE 3. The number of quasi-linear correlation columns $N(l)$ corresponding to different values of l when traversing l .

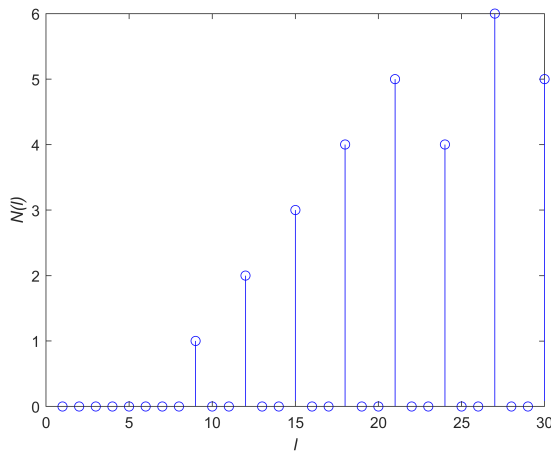


FIGURE 4. When the receiving sequence is not synchronized, the corresponding quasi-linear correlation columns $N(l)$ when traversing l .

By traversing the synchronization position, there is still no quasi-linear correlation column in X , so the synchronization position cannot be found by the algorithm in [17]. Then Algorithm 1 and Algorithm 2 in [10] cannot be used to correctly estimate the check matrix. Therefore the algorithm in [17] cannot estimate the synchronization position of such convolutional codes. It is found that such convolutional codes have a common feature that they all have at least one check vector with the first few bits being 0. For such convolutional codes, the observation matrix may still have a rank deficit when the receiving sequence is not synchronized, i.e., the frame synchronization position cannot be found directly by determining the rank deficit. Lemma 1 is proposed.

Lemma 1. When the received convolutional code sequence has a check vector whose first few bits are 0, if the received sequence is synchronized and n_a is equal to l , then n_a may still be equal to l when the received sequence is not synchronized.

Proof of Lemma 1. If the receive sequence is synchronous, i.e., the first of the receive sequence is the first of the code word, then the receive sequence vector is denoted as

$r = (r_1, r_2, r_3, \dots)$. Arrange the receiving sequence into a matrix R_l^1 of size $(u \times l)$, which can be written in the form of (8).

$$R_l^1 = \begin{bmatrix} r_1 & r_2 & \dots & r_l \\ r_{l+1} & r_{l+2} & \dots & r_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ r_{(u-1)l+1} & r_{(u-1)l+2} & \dots & r_{ul} \end{bmatrix} \quad (8)$$

The i th column vector of R_l^1 is denoted as b_i , where $i = 1, 2, \dots, l$, then $R_l^1 = (b_1 \ b_2 \ \dots \ b_l)$. There is a check vector $h = (0 \ \dots \ \lambda_j \ \dots \ \lambda_l)^T$ for matrix R_l^1 , where $1 < j \leq n$, the first $j - 1$ bits of this vector are 0. Then $R_l^1 h = 0$, which can be written as (9).

$$\sum_{i=j}^l b_i \lambda_i = 0 \quad (9)$$

If the receiving sequence is not synchronized, suppose $r = (r_s, r_{s+1}, r_{s+2}, \dots)$, where $1 < s \leq j$. Arrange the received sequence into a matrix R_l^s of size $(u \times l)$, where R_l^s can be written as (10), shown at the bottom of the next page.

Therefore $R_l^s = (b_s, \dots, b_l, b_1, \dots, b_{s-1})$. According to (9), (11) can be obtained.

$$(b_s \ \dots \ b_j \ \dots \ b_l \ b_1 \ \dots \ b_{s-1}) \begin{pmatrix} 0 \\ \vdots \\ \lambda_j \\ \vdots \\ \lambda_l \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 0 \quad (11)$$

According to (11), it is found that the rank of the matrix R_l^s is less than l , so $n_a = l$. Lemma 1 was proved.

In this paper, an improved blind recognition of synchronization positions and coding parameters for convolutional codes in a noisy environment is presented. The algorithm in [17] estimated the synchronization position after obtaining n and n_a . After estimating the synchronization position t , the information bit length k and the check matrix were estimated. According to Lemma 1, the algorithm of [17] was flawed, and the synchronization position of the convolutional code cannot be found by determining the rank deficiency only. In this paper, the idea of improvement is that after using the algorithm in [10] to obtain the n and n_a , the synchronization position t is estimated using the correlation between the code sequence r and the check matrix, and the information bit length k and the check matrix are also obtained. The improved algorithm consists of the following steps.

Step 1. The value of i is n_a or $n_a - n$. i is initialized to n_a . When the receiving sequence is synchronized, $n_a = n_a$. When the receiving sequence is not synchronized, the obtained n_a is greater than the value at the time of synchronization by n , then $n_a = n_a - n$. Then execute Step 2.

Step 2. Set the maximum value and minimum values of the information bit length k to k_{max} and k_{min} , where $k_{max} = n - 1$ and $k_{min} = 1$. k is initialized to 1. Then execute Step 3.

Step 3. Set the maximum and minimum values of the rank deficit value Z to Z_{max} and Z_{min} , where $Z_{max} = n - k$ and $Z_{min} = 1$. Z is initialized to 1. According to [10], the convolutional code constraint length $L = n_a/n(n-k) - Z$. Then execute Step 4.

Step 4. Set the maximum and minimum values of the synchronization position t to t_{max} and t_{min} , where $t_{max} = n$ and $t_{min} = 1$. t is initialized to 1. When the received sequence of the unknown synchronization position is $r = (r_1, r_2, \dots)$, the data stream starting from the t bit is $r^t = (r_t, r_{t+1}, \dots)$. Then execute Step 5.

Step 5. Arrange r^t into a matrix R_l of size $(u \times l)$, where $l = (L + 1)n$ and $u \geq 2l_{max}$. In practice, the parameters of the convolutional code are small. Set $n_{max} = 5$ and $L_{max} = 9$, then set $u = 100$. Then execute Step 6.

Step 6. Set the maximum and minimum row of the check matrix s to s_{max} and s_{min} , where $s_{max} = n - k$ and $s_{min} = 1$. s is initialized to 1. Rearrange the matrix R_l into a matrix $R_{(L+1)(k+1)}^s$, where $R_{(L+1)(k+1)}^s = R_l(1, \dots, k, k + s, n + 1, \dots, n + k, n + k + s, \dots, Ln + 1, \dots, Ln + k, Ln + k + s)$. Then execute Step 7.

Step 7. Set number of iterations of randomly swapping the rows of the matrix to w , with a maximum value w_{max} of 10 and a minimum value w_{min} of 1. The initial value of w is 1. Randomly swap the rows of the matrix $R_{(L+1)(k+1)}^s$. Then let $w = w + 1$. It is concluded in [13] that randomly swapping the rows of the matrix will not change the linear relationship between the columns of the matrix. This method in [13] has a great potential to reduce the error of the main diagonal of the matrix and thus improve the robustness of the GJETP algorithm. This method is applied in this paper to improve the accuracy of identifying synchronization positions and coding parameters. Then execute Step 8.

Step 8. Perform lower triangular transformation on $R_{(L+1)(k+1)}^s$, and get $X^s = A^s R_{(L+1)(k+1)}^s B^s$, where A^s and B^s are the corresponding row and column changes during the GJETP processing of $R_{(L+1)(k+1)}^s$. Then execute Step 9.

Step 9. Let the variable j increment from 1 to l . If $W(j) < (u - l)/2 \times \beta_{opt}$ and $B^s(l, j) = 1$. Then the j th column of B^s is the s th quasi-check vector of the convolutional code and $flag(s) = 1$. $flag(s) = 1$ indicates that the s th quasi-check vector is estimated. When $j = l$, if $flag(s) = 0$ and $w \leq 10$, then execute Step 7, otherwise execute Step 10.

Step 10. If $s < s_{max}$, then set $s = s + 1$, and execute Step 6. Otherwise, execute Step 11.

Step 11. If $flag(1)$ to $flag(n-k)$ are all equal to 1, indicating that all quasi-check vectors of the convolutional code are estimated, then output t, k, L, h_0 to h_{n-k} at this time. At this point t is the estimated synchronization position, k is the estimated information bit length, and L is the estimated convolutional code constraint length. h_0 to h_{n-k} are the estimated $n - k$ quasi-check vectors. Then execute Step 12.

Step 12. If $t < t_{max}$, then set $t = t + 1$, and execute Step 4. Otherwise, execute Step 13.

Step 13. If $Z < Z_{max}$, then set $Z = Z + 1$, and execute Step 3. Otherwise, execute Step 14.

Step 14. If $k < k_{max}$, then set $k = k + 1$, and execute Step 2. Otherwise, execute Step 15.

Step 15. If $i > n_a - n$, then set $i = i - n$, and execute Step 1. Otherwise, the calculation ends.

The above steps of blind recognition of synchronization position and coding parameters can be implemented by algorithm 2 in this paper. After algorithm 2 successfully estimates $n - k$ quasi-check vectors h_1 to h_{n-k} , the basic check matrix h_∞ of the convolutional code can be obtained. Assuming the form of h_1 to h_{n-k} are (12), as shown at the bottom of the page, then the basic check matrix h_∞ is (13), as shown at the bottom of the page.

$$R_l^s = \begin{bmatrix} r_s & r_{s+1} & \dots & r_l & r_{l+1} & \dots & r_{l+s-1} \\ r_{l+s} & r_{l+s+1} & \dots & r_{2l} & r_{2l+1} & \dots & r_{2l+s-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_{(u-1)l+s} & r_{(u-1)l+s+1} & \dots & r_{ul} & r_{ul+1} & \dots & r_{ul+s-1} \end{bmatrix} \quad (10)$$

$$\begin{cases} h_1 = [h_{1,1} \dots h_{1,k} \ h_{1,k+1} \dots h_{1,L(k+1)+1} \dots h_{1,L(k+1)+k} \ h_{1,(L+1)(k+1)}] \\ h_2 = [h_{2,1} \dots h_{2,k} \ h_{2,k+1} \dots h_{2,L(k+1)+1} \dots h_{2,L(k+1)+k} \ h_{2,(L+1)(k+1)}] \\ \vdots \\ h_{n-k} = [h_{n-k,1} \dots h_{n-k,k} \ h_{n-k,k+1} \dots h_{n-k,L(k+1)+1} \dots h_{n-k,L(k+1)+k} \ h_{n-k,(L+1)(k+1)}] \end{cases} \quad (12)$$

$$h_\infty = \begin{bmatrix} h_{1,1} & \dots & h_{1,k} & h_{1,k+1} & & \dots & h_{1,L(k+1)+1} & \dots & h_{1,L(k+1)+k} & h_{1,(L+1)(k+1)} \\ \vdots & \ddots & \vdots & & \ddots & & \vdots & \ddots & \vdots & \ddots \\ h_{n-k,1} & \dots & h_{n-k,k} & & h_{n-k,k+1} & \dots & h_{n-k,L(k+1)+1} & \dots & h_{n-k,L(k+1)+k} & h_{n-k,(L+1)(k+1)} \end{bmatrix} \quad (13)$$

Algorithm 2 An improved blind recognition method for synchronization position and coding parameters of k/n rate convolutional codes

Input: Receive codeword sequence r, n, n_a

Output: synchronization position t, k, L , check matrix

```

1: for  $i = n_a - n, n_a$  do
2:   for  $k = 1$  to  $n - 1$  do
3:     for  $Z = 1$  to  $n - k$  do
4:        $L = i/n(n - k) - Z; l = (L + 1)n;$ 
5:       for  $t = 1$  to  $n$  do
6:         Fill the matrix  $R_l$  of size  $(u \times l)$  from the  $t$  bit
           in the received data stream  $r$ ;
7:         for  $s = 1$  to  $n - k$  do
8:            $R_{(L+1)(k+1)}^s = R_l(1, \dots, k, k + s, n + 1, \dots, n + k, n + k + s, \dots, Ln + 1, \dots, Ln + k, Ln + k + s); w = 1;$ 
9:           while  $flag(s) = 0$  and  $w \leq w_{max}$  do
10:            Randomly swap rows of matrix
               $R_{(L+1)(k+1)}^s$  and  $w = w + 1;$ 
11:             $X^s = A^s R_{(L+1)(k+1)}^s B^s;$ 
12:            for  $j = 1$  to  $l$  do
13:              if  $W(j) < (u - l)/2 \times \beta_{opt}$  and
                 $B^s(l, j) = 1$  then
14:                 $h_s = B^s(j)^T; flag(s) = 1;$ 
15:              end if
16:            end for
17:          end while
18:        end for
19:        if  $flag(1)$  to  $flag(n - k)$  are all equal to 1 then
20:          Output  $t, k, L, h_1$  to  $h_{n-k};$ 
21:        end if
22:      end for
23:    end for
24:  end for
25: end for

```

The improved algorithm in this paper is applicable to blindly recognize the synchronization position and encoding parameters of all types of convolutional codes in a noisy environment. For the type of convolutional code in Lemma 1, the algorithm in [17] cannot recognize it effectively, but the improved blind recognition algorithm proposed in this paper can recognize it effectively. For example, in a noisy channel, a convolutional code with generator polynomial (7) is blindly recognized using the improved algorithm in this paper. When the SNR is 8, the BER of the received sequence is equal to 0.0127. The received sequence is unsynchronized and the first bit of the received sequence is the second bit of the codeword. After obtaining $n = 3$ and $n_a = 9$, the received sequences are processed using Algorithm 2 in this paper. When the combination of variables $i = 9, k = 2, Z = 1$, and $t = 3$, then $L = 2, l = 9$, and s can only equal 1. Traverse w from 1 to w_{max} , perform GJETP processing on the matrix R_l^1 , and get the lower triangular matrix X^1 . Calculate the Hamming weights of each column vector in the matrix X_{low}^1

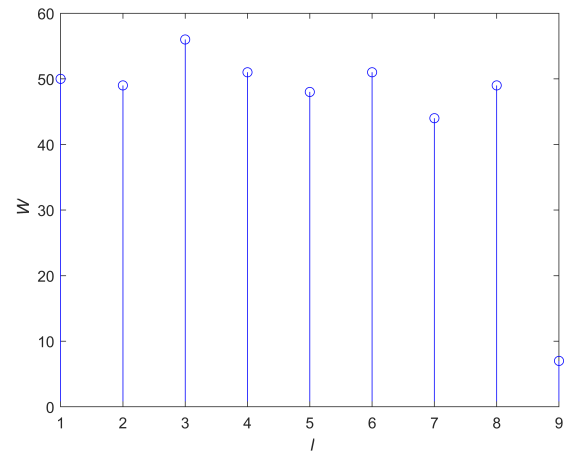


FIGURE 5. The Hamming weight of each column vector in matrix X_{low}^1 , denoted as W .

to obtain $W = [50 \ 49 \ 56 \ 51 \ 48 \ 51 \ 44 \ 49 \ 7]$, as shown in Figure 5. According to [6], an optimal threshold of $\beta_{opt} = 0.3$ is set. It is found that only $W(9) = 7 < (u - l)/2 \times \beta_{opt} = 13.65$. At this time, $(B^1(9))^T = [0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1]$, the last bit of the vector is equal to 1. Therefore, the vector is the estimated check vector. Because the estimated rank deficit value Z is 1, the convolutional code constraint length $L = i/n \times (n - k) - Z = 2$. So this estimated check vector is equal to the check vector of the convolutional code. Because the first bit of the received sequence is the second bit of the convolutional codeword, and the estimated code length is 3, the estimated synchronization position $t = 3$ is the first bit of the next convolutional codeword. So Algorithm 2 in this paper correctly identified the synchronization position and coding parameters of the convolutional code. It shows that the improved algorithm in this paper can also effectively blindly recognize the synchronization position and coding parameters of the convolutional code type in Lemma 1.

V. SIMULATION RESULTS

This section is to verify the performance of the improved method in this paper through simulation experiments. In the simulation experiment, the input information sequence is randomly generated. The modulation method adopts BPSK modulation. The transmission channel is an additive white Gaussian noise (AWGN) channel. After demodulating the received sequence, select any position as the starting point and intercept 20,000 bits to estimate the synchronization position and coding parameters of the convolutional code.

Figure 6 shows the accuracy of the blind recognition of the synchronization position and coding parameters of several types of convolutional codes under different signal-to-noise ratio (SNR) conditions. Set w_{max} to 10 during simulation. The set of convolutional codes for simulation is listed in TABLE 1. It can be concluded from Figure 6 that for the same convolutional code, the larger the SNR, the higher the recognition accuracy. The recognition accuracy of convolutional code 1 is higher than that of convolutional code 2.

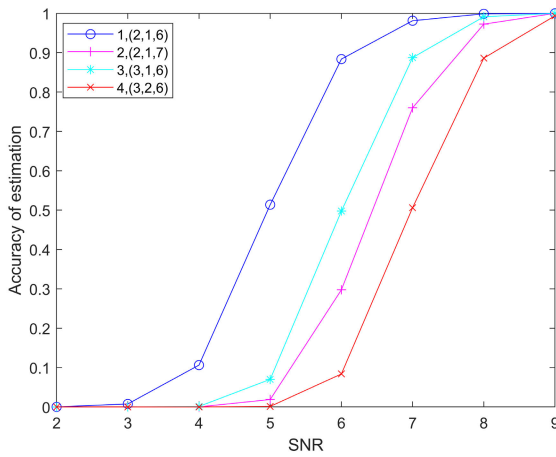


FIGURE 6. The estimation accuracy of the algorithm in this paper for different convolutional codes.

TABLE 1. The set of convolutional codes for simulations.

No.	Convolutional code type	Basic generator matrix
1	(2, 1, 6)	$\begin{bmatrix} 125 & 40 \end{bmatrix}$
2	(2, 1, 7)	$\begin{bmatrix} 137 & 301 \end{bmatrix}$
3	(3, 1, 6)	$\begin{bmatrix} 1 & 53 & 137 \\ 1 & 0 & 177 \end{bmatrix}$
4	(3, 2, 6)	$\begin{bmatrix} 1 & 0 & 177 \\ 0 & 1 & 3 \end{bmatrix}$

It is concluded that for convolutional codes with the same code length and code rate, the greater the constraint length, the greater the GJETP algorithm is affected by the error, and therefore the lower the recognition accuracy. The recognition accuracy of convolutional code 4 in Figure 6 is less than that of convolutional code 3, thus it is concluded that the smaller the code rate, the greater the recognition accuracy for convolutional codes with the same code length and constraint length. Since $n_a = n \times \lfloor L/(n - k) + 1 \rfloor$ [10], when the code rate k/n is smaller, the value of n_a is smaller. Then the less the GJETP algorithm is affected by the error code, the higher the recognition accuracy. So the simulation results are consistent with the theoretical analysis. In general, when the channel conditions are the same, for any type of convolutional code, the improved algorithm in this paper is mainly affected by the value of n_a for its recognition accuracy. The greater the n_a value, the greater the impact of error codes on the improved algorithm in this paper, and the lower its recognition accuracy. When the SNR is greater than 7, the improved algorithm in this paper can basically correctly identify the synchronization position and coding parameters of all types of convolutional codes.

Figure 7 shows the blind recognition accuracy of the improved algorithm in this paper for convolutional code 3 and convolutional code 4 when the number of iterations is 10, 20, 80 and 100, respectively. It can be seen from the figure that with the increase of the number of iterations, the improved algorithm in this paper improves the accuracy of blind recognition of the synchronization positions and parameters of the convolutional code 3 and the convolutional code 4. It shows

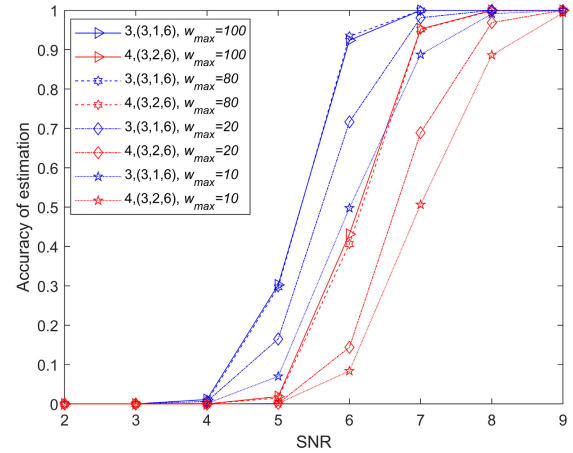


FIGURE 7. Blind recognition accuracy of the improved algorithm in this paper for convolutional code 3 and convolutional code 4 when the number of iterations is 10, 20, 80 and 100, respectively.

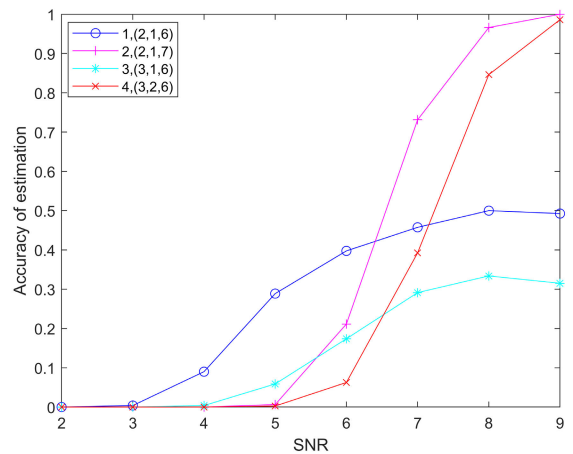


FIGURE 8. The estimation accuracy of the algorithm in [17] for different convolutional codes.

that the method of randomly swapping the rows of the observation matrix R_l can reduce the errors on the diagonal of the matrix and improve the robustness of the improved algorithm in this paper, and the method is also effective for arbitrary code rate convolutional codes. After the number of iterations reaches 80, the performance improvement is not very significant. Moreover, as the number of iterations increases, the computational complexity increases, so a trade-off between performance and computational complexity is required when applying the algorithm.

Figure 8 is the simulation result of the synchronization position and coding parameter recognition algorithm in [17] at different SNR. The set of convolutional codes for simulation is listed in TABLE 1. It can be seen from the figure that for the same code, the larger the SNR, the higher the recognition accuracy. It should be noted in the figure that the recognition accuracy of convolutional code 1 and convolutional code 3 is relatively low when the SNR is 9. By observing their check matrix, it is found that they are all types of convolutional codes in Lemma 1. The selected convolutional

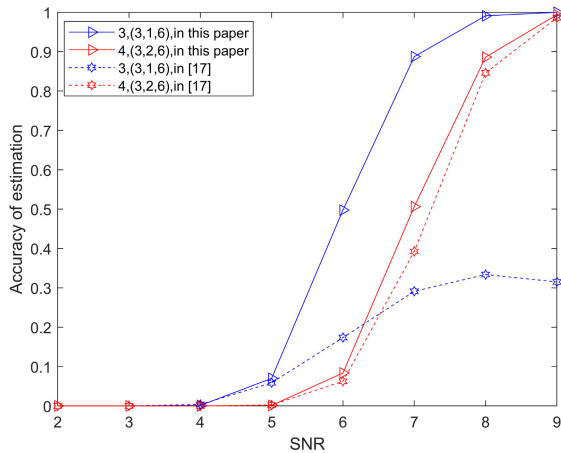


FIGURE 9. Estimation accuracy of the 2 algorithms at different SNR.

code 2 and convolution 4 are not the convolutional code types of Lemma 1, and their recognition accuracy is relatively high. The conclusion of Lemma 1 is verified, and the algorithm in [17] is not suitable for the type of convolutional code in Lemma 1.

Figure 9 is the performance comparison between the algorithm proposed in this paper and the algorithm in [17] under different convolutional code types. The set of convolutional codes for simulation is listed in TABLE 1. The conclusion can be drawn from Figure 7. For the convolutional code 1 in Lemma 1, because the algorithm in [17] is not suitable for these types, the algorithm proposed in this paper has a much higher recognition accuracy than the algorithm in [17]. For convolutional code 2 that is not in Lemma 1, the recognition accuracy of the algorithm in this paper is also higher than that in [17]. The superior performance of the improved algorithm in this paper is verified, and the synchronization position and coding parameters of all types of convolutional codes can be correctly estimated under certain noise conditions.

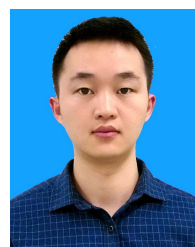
VI. CONCLUSION

This paper presents a method to identify the synchronization position and coding parameters of k/n rate convolutional codes in a noisy environment. The structure and characteristics of convolutional codes are briefly introduced. The blind recognition algorithms proposed by published references are summarized, and these algorithms all have various shortcomings. The algorithm in [10] implemented blind recognition of the coding parameters of k/n rate convolutional codes in a noisy environment, but only if the synchronization position was known. The algorithm in [17] improved on this problem, but it didn't accomplish blind recognition of the synchronization position and encoding parameters for all types of convolutional codes. This paper proposes an improved synchronization position and parameter recognition algorithm. The simulation results show that the algorithm proposed in this paper can better realize the recognition of the

synchronization position and coding parameters of k/n rate convolutional codes in a noisy environment.

REFERENCES

- [1] Y. Zrelli, M. Marazin, E. Rannou, and R. Gautier, "Blind identification of convolutional encoder parameters over GF(2m) in the noiseless case," in *Proc. 20th Int. Conf. Comput. Commun. Netw. (ICCCN)*. Lahaina, HI, USA: IEEE, Jul. 2011, pp. 1–5.
- [2] J. Barbier, "Reconstruction of turbo-code encoders," *Proc. SPIE*, vol. 5815, pp. 463–473, Jun. 2005.
- [3] E. Filiol, "Reconstruction of convolutional encoders over GF(q)," in *Cryptography and Coding*, vol. 1355. Berlin, Germany: Springer, 1997, pp. 101–109.
- [4] P. Lu and Y. Zou, "Fast computations of Gröbner bases and blind recognitions of convolutional codes," in *Arithmetic of Finite Fields*, vol. 4547. Berlin, Germany: Springer, 2007, pp. 303–317.
- [5] F. Wang, Z. Huang, and Y. Zhou, "A method for blind recognition of convolution code based on Euclidean algorithm," in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput.* Shanghai, China: IEEE, Sep. 2007, pp. 1414–1417.
- [6] J. Dingel and J. Hagenauer, "Parameter estimation of a convolutional encoder from noisy observations," in *Proc. IEEE Int. Symp. Inf. Theory*. Nice, France: IEEE, Jun. 2007, pp. 1776–1780.
- [7] G. Forney, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 6, pp. 720–738, Nov. 1970.
- [8] G. Forney, "Structural analysis of convolutional codes via dual codes," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 512–518, Jul. 1973.
- [9] M. Marazin, R. Gautier, and G. Burel, "Dual code method for blind identification of convolutional encoder for cognitive radio receiver design," in *Proc. IEEE Globecom Workshops*. Honolulu, HI, USA: IEEE, Nov. 2009, pp. 1–6.
- [10] M. Marazin, R. Gautier, and G. Burel, "Blind recovery of k/n rate convolutional encoders in a noisy environment," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, p. 168, Dec. 2011.
- [11] Y. Zrelli, R. Gautier, M. Marazin, E. Rannou, and E. Radoi, "Focus on theoretical properties of blind convolutional codes identification methods based on rank criterion," in *Proc. 9th Int. Conf. Commun. (COMM)*. Bucharest, Romania: IEEE Jun. 2012, pp. 353–356.
- [12] A. G. Sotah and H. K. Bizaki, "On the analytical solution of rank problem in the convolutional code identification context," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 442–445, Mar. 2016.
- [13] Y. Xu, Y. Zhong, and Z. Huang, "An improved blind recognition method of the convolutional interleaver parameters in a noisy channel," *IEEE Access*, vol. 7, pp. 101775–101784, 2019.
- [14] G. Lui and H. Tan, "Frame synchronization for Gaussian channels," *IEEE Trans. Commun.*, vol. 35, no. 8, pp. 818–829, Aug. 1987.
- [15] J. Sun and M. C. Valenti, "Optimum frame synchronization for preambleless packet transmission of turbo codes," in *Proc. Conf. Rec. 38th Asilomar Conf. Signals, Syst. Comput.* Pacific Grove, CA, USA: IEEE, vol. 1, Nov. 2004, pp. 1126–1130.
- [16] R. Imad, G. Sicot, and S. Houcke, "Blind frame synchronization for error correcting codes having a sparse parity check matrix," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1574–1577, Jun. 2009.
- [17] S. Su, J. Zhou, Z. Huang, C. Liu, and Y. Zhang, "Blind identification of convolutional encoder parameters," *Sci. World J.*, vol. 2014, May 2014, Art. no. 798612.



YONG DING received the B.S. and M.S. degrees in instrument science and technology from the College of Artificial Intelligence, National University of Defense Technology (NUDT), Changsha, China, in 2016 and 2019, respectively, where he is currently pursuing the Ph.D. degree. His research interests include intelligent optical communications and blind recognition of channel coding.



ZHIPING HUANG received the Ph.D. degree in electrical engineering from the Beijing Institute of Technology (BIT), Beijing, China, in 1993. He is currently a Professor with the College of Artificial Intelligence, NUDT, where he is involved in the design of smart transmitter and receiver of optical signals. His research interests include optical communication and signal processing. He is also a Principal Investigator of the projects funded by the National Key Research and Development Plan of China.



JING ZHOU received the Ph.D. degree in instrument science and technology from the National University of Defense Technology, Changsha, China, in 2004. He is currently working as a Lecturer with the Cyberspace Lab, College of Intelligent Science, NUDT. His main research interests include cyberspace intelligent monitoring and signal processing for optical transmission networks.

...