# Blind Recognition of $(n-1)/n$ Rate Punctured Convolutional Encoders in a Noisy Environment

Wengu Chen and Guoqing Wu
Institute of Applied Physics and Computational Mathematics, Beijing 100088, China
Email: {chenwg, wu_guoqing}@iapcm.ac.cn

***Abstract*** —Blind recognition of error-correcting code is an essential problem to decode intercepted data. In this paper, a method dedicated to the blind recognition of punctured convolutional encoders is presented. The blind recognition of such encoders is of great significance, because convolutional encoders are embedded in most digital transmission systems where the puncturing principle is used to increase the code rate. After a brief review of the principle of puncturing codes, a method mainly based on the Walsh-Hadamard transform is presented for blind recognition of both the mother code and the puncturing pattern when the received bits are erroneous. Compared to existing techniques, our algorithm has advantages of robustness and efficiency. Experiments are conducted to illustrate the performances of this new blind recognition method.

***Index Terms***—Blind Recognition, convolutional code, punctured pattern, walsh-hadamard transform.

## I. Introduction

Most digital transmission systems are encoded to enhance the communication quality. Redundancy bits are appended in the informative binary data stream to better withstand channel noise. In a non-cooperative context, in order to perform information analysis, it is necessary to decode intercepted data with no knowledge of the parameters of the code. In this case, the blind recognition problem needs to be addressed. Convolutional codes are a class of important codes due to their flexibility in code length, soft decodability, short decoding delay and their role as component codes in parallelly serially concatenated codes. Puncturing allows convolutional codes to flexibly change rates and is widely used in applications where high code rates are required and where rate adaptivity is desired. In this paper, we only focus on communications encoded with punctured convolutional codes.

This article is not the first to deal with blind recognition of convolutional codes in a noisy environment. A systematic algebraic approach for the reconstruction of linear and convolutional error correcting codes was introduced by J. Barbier [1]. At the same time, the methods to recover a block code are developed in [2], [3], whereas [4] deals with the blind identification of linear scramble. An iterative algorithm for blind identification of a rate $(n-1)/n$ convolutional encoder is introduced in [5]. The first approaches related with recovering the punctured code in a noiseless environment were proposed in [6], [7]. Ref. [8] discussed blind recognition of a rate $1/n$ mother code in noisy environment. Here, we describe a new method for blind recognition of a more general rate $(n-1)/n$ punctured convolutional code. In this context, the blind identification results of a punctured code consist of the true mother code and the puncturing pattern.

In this paper, we propose a novel approach for blind recognition of $(n-1)/n$ rate punctured convolutional encoders from received binary data stream in a noisy environment. The proposed method is based on Walsh-Hadamard transform and decomposition of parity check equation. Our method offers robustness to noise and better performance than prior arts. The remainder of this paper is organized as follows. In Section II the principle of punctured convolutional encoders is explained. The method of the blind recognition of this punctured code is developed in Section III. Finally, the performances of the blind identification method are discussed in Section IV. Conclusions are drawn in Section V.

## II. Punctured Convolutional Code

In this section we give the definitions and notations we use in the rest of the article. Punctured convolutional code was firstly discovered by Cain *et al.* [9]. Punctured code is obtained through a periodic elimination of certain bit of a low-rate mother codes and it depends on the low-rate mother code and the puncturing pattern of the punctured code. The punctured pattern is described in matrix form which is called the punctured matrix and denoted as $\mathbf{P}$ . For more details of punctured convolutional code, see also [7], [9]-[14]. In this section, we mainly refer to Ref. [11], [12].

### A. Convolutional Encoders

A $C(n,k,K)$ convolutional encoder is defined by a $k\times n$ polynomial generator matrix $\mathbf{G}$ in Galois Field. Parameter $n$ is the number of outputs, $k$ is the number of

Manuscript received January 9, 2015; revised April 8, 2015.
This work was supported by The Natural Science Foundation of China (11271050, 11371183, 61403036), Science and Technology Development Foundation of CAEP (2013B0403068) and Beijing Center for Mathematics and Information Interdisciplinary Sciences (BCMIIS).
Corresponding author email: wu_guoqing@iapcm.ac.cn.
doi:10.12720/jcm.10.4.260-267

inputs and thus a $C(n,k,K)$ convolutional code has an information rate of $k/n$. An important parameter of convolutional codes is constraint length $K$ which corresponds to the total size of internal memory. The polynomial generator matrix is defined by

$$\mathbf{G}(D)=\begin{bmatrix} g_{1,1}(D) & \cdots & g_{1,n}(D) \\ \vdots & \cdots & \vdots \\ g_{k,1}(D) & \cdots & g_{k,n}(D) \end{bmatrix} \quad (1)$$

where $g_{i,j}(D), \forall i=1,2,\cdots,k, \forall j=1,2,\cdots,n,$ are generator polynomials and $D$ is the delay operator. The encoding process can be described by formula

$$\mathbf{c}(D)=\mathbf{m}(D)\mathbf{G}(D) \quad (2)$$

where $\mathbf{m}(D)$ is the input sequence and $\mathbf{c}(D)$ is the output sequence.

### B. Puncturing Principle

Puncturing a convolutional code consists in transmitting only part of the output of the code, following a regular puncturing pattern. On condition that both transmit $M\times k$ bits and receives $M\times n$ bits at the output one would pass from a $C(n,k,K)$ mother code to the $M$th blocking code of $C$, denoted as $C^{[M]}$ $(M\times n,\ M\times k,K)$. The puncturing process consists of deleting a few bits from the code words through use of the puncturing matrix $\mathbf{P}$, which is a $(n\times \mathrm{M})$ binary matrix with a total of $N$ ones and $M\times n-N$ zeros where ones correspond to the transmitted bits and zeros to the deleted ones. Application of this puncturing pattern to the $C^{[M]}$ $(M\times n,\ M\times k,K)$ code leads to the $C_p(n_p,k_p,K_p)$ code, called the equivalent punctured code, where $n_p=N$ and $k_p=M\times k$.

***Example 1:*** Let's consider the encoder for the $C(2,1,K)$ convolutional encoder. The coding and puncturing processes can be represented as follows

$$\begin{pmatrix} c_1^0 & c_1^1 & c_1^2 & c_1^3 & c_1^4 & c_1^5 & \cdots \\ c_2^0 & c_2^1 & c_2^2 & c_2^3 & c_2^4 & c_2^5 & \cdots \end{pmatrix} \Rightarrow \begin{pmatrix} c_1^0 & & c_1^2 & c_1^3 & & c_1^5 & \cdots \\ c_2^0 & c_2^1 & & c_2^3 & c_2^4 & & \cdots \end{pmatrix} \quad (3)$$

where $c_j^t$ is the bit of the output, $j$, encoded at the time, $t$. Using the puncturing pattern (4)

$$\mathbf{P}=\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad (4)$$

leads to a new encoder of rate $r_p=k_p/n_p$, where, $k_p=3$, and $n_p=4$.

### C. Equivalent Punctured Code

As shown in [15], the equivalent punctured convolutional encoder can be described by a simple convolutional encoder $C_p(n_p,k_p,K_p)$. This equivalent punctured code is defined by a generator matrix $\mathbf{G}_p(D)$ of size $(k_p\times n_p)$. A high rate equivalent punctured code can be built from an original rate convolutional code through the simple process detailed hereafter. But prior to obtaining this equivalent punctured encoder, $C_p(n_p,k_p,K_p)$ code, it is worth recalling some definitions necessary to understand the constructing of this punctured code.

***Definition 1:*** If $a(D)=a_0+a_1D+a_2D^2+\cdots$ is a polynomial in the indeterminate $D$, then for any integer $M>1$, the *Mth* polyphase decomposition of $a(D)$ is the list of $(i,M)th$ polyphase components $(\forall i=0,...,M-1)$. Let us denote by $q_i(D)$ the $(i,M)th$ polyphase component of $a(D)$ such that

$$q_i(D)=D^{-i/M}a_{[i]_M}(D^{1/M}) \quad (5)$$

where $[i]_M$ (with $i$ and $M$ are integers) is the congruence class of $i$ (modulo $M$). Finally, let $a_{[i]_M}(D^r)$ be the polynomial issued from $a(D)$ by selecting only the $[i]_M$ degree terms and then substituting $D^r$ for $D$.

***Example 2:*** For illustration, let us consider the generator polynomial $a(D)$

$$a(D)=1+D^2+D^3+D^5+D^6 \quad (6)$$

or

$$a(D)=1+D+D^2+D^3+D^6 \quad (7)$$

Then, the $(i,3)(i=0,1,2)$ polyphase component of $a(D)$ is

$$q_0(D)=1+D+D^2, q_1(D)=0, q_2(D)=1+D \quad (8)$$

or

$$q_0(D)=1+D+D^2, q_1(D)=1, q_2(D)=1 \quad (9)$$

respectively.

***Definition 2:*** Let $a(D)=a_0+a_1D+a_2D^2+\cdots$ be a polynomial in the indeterminate $D$ and $[q_0(D),q_1(D),...,q_{M-1}(D)]$ be the *Mth* polyphase decomposition of $a(D)$. The *Mth* polycyclic pseudo circulant matrix (or PCPC for short) associated with $a(D)$ is the $(M\times M)$ polynomial matrix, $\mathbf{Q}^{[M]}(D)$, such that

$$\mathbf{Q}^{[M]}(D)=\begin{bmatrix} q_0(D) & q_1(D) & \cdots & q_{M-1}(D) \\ D\cdot q_{M-1}(D) & q_0(D) & \cdots & q_{M-2}(D) \\ \vdots & \vdots & \ddots & \vdots \\ D\cdot q_1(D) & D\cdot q_2(D) & \cdots & q_0(D) \end{bmatrix} \quad (10)$$

***Example 3:*** The third PCPC associated with $a(D)$ as in the Example 2 can be obtained as follows.

$$\mathbf{Q}^{[3]}(D)=\begin{bmatrix} 1+D+D^2 & 0 & 1+D \\ D+D^2 & 1+D+D^2 & 0 \\ 0 & D+D^2 & 1+D+D^2 \end{bmatrix} \quad (11)$$

or

$$\mathbf{Q}^{[3]}(D)=\begin{bmatrix}1+D+D^2 & 1 & 1\\ D & 1+D+D^2 & 1\\ D & D & 1+D+D^2\end{bmatrix} \quad (12)$$

***Theorem 1:*** If $C$ is a $(n,k)$ convolutional code, then the *Mth* blocking code of $C$, denoted by $C^{[M]}$, is an $(M\times n, M\times k)$ convolutional code. If $\mathbf{G}(D)$ is a $(n\times k)$ polynomial generator matrix for the original code $C$, then a generator matrix for $C^{[M]}$, say $\mathbf{G}^{[M]}(D)$, can be obtained from $\mathbf{G}(D)$ by substituting the corresponding *Mth* PCPC for each entry $g_{i,j}(D)$ from $\mathbf{G}(D)$ and then interleaving the columns and lines at depth $M$.

***Example 4:*** Let us consider the $C(2,1,7)$ mother code. The generator matrix of this code is $\mathbf{G}(D)=[1+D^2+D^3+D^5+D^6, 1+D+D^2+D^3+D^6]$. We denote by $\mathbf{Q}_{1,1}^{[3]}$, the third PCPC associated with $g_{1,1}(D)$ and by $\mathbf{Q}_{1,2}^{[3]}$, the third PCPC associated with $g_{1,2}(D)$. These matrices can be written as in Example 3. So, the generator matrix of the $C^{[3]}$ code is such that

$$[\mathbf{Q}^{[3]}(D)]'=\begin{bmatrix}1+D+D^2 & D+D^2 & 0\\ 1+D+D^2 & D & D\\ 0 & 1+D+D^2 & D+D^2\\ 1 & 1+D+D^2 & D\\ 1+D & 0 & 1+D+D^2\\ 1 & 1 & 1+D+D^2\end{bmatrix} \quad (13)$$

***Definition 3:*** On condition that $\mathbf{G}(D)$ is a $(n\times k)$ polynomial matrix and $\mathbf{P}$ is an $(M\times n)$ binary matrix, then the $n.M$ columns of the matrix, $\mathbf{G}^{[M]}(D)$, are in natural one-to one correspondence with the $M\times n$ entries of $\mathbf{P}$ and the matrix, $\mathbf{G}_p(D)$, is the matrix issued from $\mathbf{G}^{[M]}(D)$ after deletion of the columns corresponding to $\mathbf{P}$ entries. The code defined by the generator matrix, $\mathbf{G}_p(D)$, is called the $\mathbf{P}-$ punctured version of $C$.

Let $\phi$ be a bijection such that

$$(i,j)\mapsto \phi(i,j)=i+n(j-1) \quad (14)$$

To associate the $\mathbf{P}(i,j)$ coefficient with the $\phi(i,j)$ column of $\mathbf{G}^{[M]}(D)$ let us delete $\mathbf{G}^{[M]}(D)$ columns according to $\mathbf{P}$ coefficients; it leads to the equivalent punctured convolutional code matrix, $\mathbf{G}_p(D)$.

***Example 5:*** Further to the calculation of the matrix, $\mathbf{G}^{[3]}(D)$, in Example 4, let us assume

$$\mathbf{P}=\begin{pmatrix}1 & 1 & 1\\ 0 & 0 & 1\end{pmatrix} \quad (15)$$

The coefficients, $P(2,1)$ and $P(2,2)$, is equal to zero and correspond to the second and fourth columns of $\mathbf{G}^{[3]}(D)$. Deleting these two columns leads to the generator matrix of the equivalent punctured code

$$[\mathbf{G}_p(D)]'=\begin{bmatrix}1+D+D^2 & D+D^2 & 0\\ 0 & 1+D+D^2 & D+D^2\\ 1+D & 0 & 1+D+D^2\\ 1 & 1 & 1+D+D^2\end{bmatrix} \quad (16)$$

The equivalent punctured code rate is $r_p=3/4$ and the constraint length is $K_p=3$.

## III. Blind Recognition of a Punctured Convolutional Code

This section deals with the blind recognition of the punctured convolutional code in a noisy environment. The recognition process consists of three steps: (a) identification of the number of outputs *n*; (b) identification of the parity check matrix and (c) identifications of the mother code and the puncturing pattern.

### A. Identification of the Number of Outputs

An iterative process dedicated to the blind identification of a rate $(n-1)/n$ convolutional encoder in a noisy environment is explained in [5]. The principle of the method is to first identify the number of outputs n. Then, a basis of the dual code can be estimated. Finally, a generator matrix is obtained by solving a system. Let us recall the principle of this algorithm.

The first step is to reshape column wise the received data bit stream under matrix form of size $(L\times l)$, denoted $\mathbf{R}_l$. This matrix is computed for different values of $l(\forall l=1,2,\cdots,L/2)$ and for each matrix the Gauss Jordan Elimination through Pivoting is applied to obtain a lower triangular matrix noted $\mathbf{G}_l$

$$\mathbf{A}_l\mathbf{R}_l\mathbf{B}_l=\mathbf{G}_l \quad (17)$$

In (17), $\mathbf{A}_l$ is an $(L\times L)$ rows permutation matrix and $\mathbf{B}_l$ an $(l\times l)$ matrix describing the columns combination. To detect the value of *n*, the principle is to find matrices $\mathbf{R}_l$ which exhibit a rank deficiency. So, the gap between column lengths of two consecutive rank deficiency matrix $\mathbf{R}_l$ corresponds to *n*. Then a dual code basis can be built from the matrix $\mathbf{B}_l$. This paper describes a new method based on the Walsh-Hadamard transform to achieve the blind recognition of a rate $(n-1)/n$ punctured convolutional code.

### B. Identification of the Parity Check Matrix

By the first step of above algorithm, we assume that the parameter *n* is already known. Now we suppose the parity check polynomial of punctured convolutional code

$$\mathbf{H}(D)=[H^{(1)}(D)H^{2}(D)\cdots H^{(n)}(D)]$$

where

$$H^{(i)}(D)=h_{0,i}+h_{1,i}D+h_{2,i}D^{2}+\cdots+h_{d,i}D^{d}\ (1\le i\le n)$$

$$d=\max_{1\le i\le n}(\deg H^{(i)}(D))$$

In noiseless case, the received data sequence $\mathbf{R}(D)$ is the code word sequence, so

$$\mathbf{R}(D)=\mathbf{V}(D)=[r_1(D)r_2(D)\cdots r_n(D)]$$

$$r_i(D)=r_{0,i}+r_{1,i}D+r_{2,i}D^{2}+\cdots(1\le i\le n)$$

satisfies the relation of the code word and the parity check matrix

$$\mathbf{V}(D)\mathbf{H}(D)'=0$$

Then,

$$[r_1(D)r_2(D)\cdots r_n(D)][H^{(1)}(D)H^{(2)}(D)\cdots H^{(n)}(D)]'=0$$

$$\Leftrightarrow\sum_{j=1}^{n}r_j(D)H^{(j)}(D)=0$$

$$\Leftrightarrow\sum_{j=1}^{n}(\sum_{i=0}^{\infty}r_{i,j}D^{i}\cdot\sum_{i=0}^{d}h_{i,j}D^{i})=0$$

It reduces to the following linear equation system:

$$\sum_{j=1}^{n}\sum_{i=0}^{d}h_{i,j}r_{k-i,j}=0,\quad k=d,d+1,\cdots,d+N$$

where we assume that the length of code word sequence is $(d+N+1)n$. Write this system in the form of matrix:

$$\begin{pmatrix} r_{d,1} & \cdots & r_{0,1} & \cdots & r_{d,n} & \cdots & r_{0,n} \\ r_{d+1,1} & \cdots & r_{1,1} & \cdots & r_{d+1,n} & \cdots & r_{1,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ r_{d+N,1} & \cdots & r_{N,1} & \cdots & r_{d+N,n} & \cdots & r_{N,n} \end{pmatrix}$$

$$\times(h_{0,1}\cdots h_{d,1}\cdots h_{0,n}\cdots h_{d,n})'=0 \qquad (18)$$

Denote by (18) for short

$$\mathbf{R}\cdot\mathbf{x}'=0$$

In (18) there are $N+1$ equations and $(d+1)n$ variables $\{h_{0,1}\cdots h_{d,1}\cdots h_{0,n}\cdots h_{d,n}\}$. In order to solve (18), we assume $N\ge(d+1)\times n$. In noiseless case, Gauss elimination algorithm [16] is enough to solve (18). In noisy case, the received data sequence is $\mathbf{R}(D)=\mathbf{V}(D)+\mathbf{e}(D)$, where $\mathbf{e}(D)$ is the error polynomial. The syndrome

$$\mathbf{R}(D)\mathbf{H}(D)'=\mathbf{S}(D)\ne 0 \text{ if } \mathbf{e}(D)\ne 0$$

So (18) is a linear system with error equations. The linear system (18) can be solved by using Walsh-Hadamard transform. Walsh-Hadamard transform is defined by Walsh-Hadamard matrix. A $2^n\times 2^n$ order Walsh-Hadamard matrix $\mathbf{H}(n)$ is defined recursively by:

$$\mathbf{H}(0)=[1],\mathbf{H}(n)=\begin{bmatrix}\mathbf{H}(n-1) & \mathbf{H}(n-1)\\ \mathbf{H}(n-1) & -\mathbf{H}(n-1)\end{bmatrix}$$

By this recursive formula,

$$\mathbf{H}(1)=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix},\mathbf{H}(2)=\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

Suppose $\{f(k)\}(k=0,1,\cdots,N-1,N=2^{n-1})$ is a data vector, then the following vector

$$\{\mathbf{F}(k)\}=\frac{1}{N}\mathbf{H}(n-1)\{\mathbf{f}(k)\}$$

is called Walsh-Hadamard transform. The Walsh-Hadamard matrix satisfies some important properties such that it could be used to solve the linear system with error equations.

The element in the cross point of row $\mathbf{u}$ and column $\mathbf{v}$ of $\mathbf{H}(n)$

$$h_{uv}=(-1)^{\overrightarrow{u_n}\cdot\overrightarrow{v_n}'},\quad u,v=0,1,\cdots,2^n-1$$

where

$$(\overrightarrow{\mathbf{u}_n})_{binary}=(u_{n-1}u_{n-2}\cdots u_0)=(\mathbf{u})_{decimal}$$

$$(\overrightarrow{\mathbf{v}_n})_{binary}=(v_{n-1}v_{n-2}\cdots v_0)=(\mathbf{v})_{decimal}$$

Since

$$h_{uv}=\begin{cases}1, & if\ \overrightarrow{\mathbf{u}_n}\cdot\overrightarrow{\mathbf{v}_n}'=0\\ -1, & if\ \overrightarrow{\mathbf{u}_n}\cdot\overrightarrow{\mathbf{v}_n}'=1\end{cases}$$

the row vector $\mathbf{h}_u$ in the Walsh-Hadamard matrix presents all solutions such that the product with $\overrightarrow{\mathbf{u}_n}$ is zero. Every solution of $\overrightarrow{\mathbf{u}_n}\cdot\mathbf{x}'=0$ is the binary representation $\overrightarrow{\mathbf{v}_n}$ of the column position $v$ which corresponds to 1 in the vector $\mathbf{h}_u$. When we seek for the solutions of $\overrightarrow{\mathbf{u}_n}\cdot\mathbf{x}'=0$, we can transfer $\overrightarrow{\mathbf{u}_n}$ to its decimal number $u$. Then we look for the column positions $v$ which correspond to 1 in the row $\mathbf{u}$ in the $2^n\times 2^n$ Walsh-Hadamard matrix. The binary representation $\overrightarrow{\mathbf{v}_n}$ of $v$ is a solution of the equation, in all $2^{n-1}$ such solutions for one equation. For a system with more than one equation, we can add all row vectors in the real number field and get a new row vector. Then the binary representation $\overrightarrow{\mathbf{v}_n}$ of $v$ which is equal to the number of the equations in the system is the solution of the system. So, we can solve the system (18) by the following process, hereafter we assume $m=(d+1)n$.

The first step is to transfer the coefficient vector of each equation in the system to decimal number. Then we can obtain $N+1$ decimal numbers from $N+1$ equations.

The second step is to construct a new $2^m$ dimensional vector $\mathbf{A}'$. Set the position in $\mathbf{A}'$ to 1 corresponding to those decimal numbers, 0 for other positions.

The third step is to compute $2^m$ dimensional vector $\mathbf{B}'$ according to the following formula

$$\mathbf{A}'\times\mathbf{H}(m)=\mathbf{B}'$$

The last step is to check the vector $\mathbf{B}'$. If some element of $\mathbf{B}'$ is equal to the number of the equations, then the binary representation $\overrightarrow{\mathbf{u}_n}$ of the position $u$ of this element is the solution of this system. Maybe there is more than one solution. If every element except 0 column is less than the number of the equations, there is no solution for the system.

The third step in the above process is actually a Walsh-Hadamard transform of $\mathbf{A}'$. $\mathbf{B}'$ is the spectrum coefficients. In this paper, the spectrum coefficients $\mathbf{B}'$ possess obvious physical meaning. As pointed earlier, if $h_{uv}=1$, then it shows that $\overrightarrow{\mathbf{u}_n}\cdot\overrightarrow{\mathbf{v}_n}'=0$, and if $h_{uv}=-1$, then $\overrightarrow{\mathbf{u}_n}\cdot\overrightarrow{\mathbf{v}_n}'=1$. So, the physical meaning of $\mathbf{B}'$ is: the value of every element in $\mathbf{B}'$ denotes the difference of the number of equations satisfied by the binary representation $\overrightarrow{\mathbf{u}_n}$ of the element position $u$ and the number of equations dissatisfied by the binary representation $\overrightarrow{\mathbf{u}_n}$ of the element position $u$. Since the process of solving linear equations is Walsh-Hadamard transform, one could reduce the complexity of Walsh-Hadamard transform from $N^2$ to $N\log N$ by using the fast Walsh-Hadamard transform.

On the insight into the physical meaning of $\mathbf{B}'$, we can extend this method to solve the linear system with error equations. For the error linear equations, the most possible solution is the binary representation $\overrightarrow{\mathbf{u}_n}$ of the element position $u$ which corresponds to the maximal spectrum coefficient. Almost as before, we can solve the linear system with error equations by the following steps.

The first step and the third step are the same as before. The second and the fourth steps are of some difference.

The second step is to construct a new $2^m$ dimensional vector $\mathbf{A}'$. Set the position in $\mathbf{A}'$ to $t$ corresponding to those decimal numbers where $t$ is the time the decimal number appears 0 for other positions.

The fourth step is to check the vector $\mathbf{B}'$ and find the largest element in $\mathbf{B}'$. The most possible solution is the binary representation $\overrightarrow{\mathbf{u}_n}$ of the element position $u$ which corresponds to the largest element. Because the equations are with errors, the last step is to check the confidence level of the result. Suppose $\mathbf{c}=(c_1,c_2,\cdots,c_n)$ is a solution of the system with error equations. If $\mathbf{c}$ satisfies the *ith* equation, let $\xi_i=1$, otherwise, let $\xi_i=-1$. If the probability of the equation holds is $p$, the mean and variance of $\sum_{i=1}^{j}\xi_i$ are $E(\sum_{i=1}^{j}\xi_i)=j(p-q)$, $D(\sum_{i=1}^{j}\xi_i)=4jpg$, where $j$ is the number of equations and $q=1-p$. When $j$ is big enough, then we have

$$(\sum_{i=1}^{j}\xi_i - j(p-q))/\sqrt{4jpq}\propto N(0,1)$$

Suppose $c$ satisfies $n_1$ equations in the system and does not for another $n_2$ equations, and let $z=n_1-n_2$. Compute the statistics quantity:

$$T=(z-j(p-q))/\sqrt{4jpq} \tag{19}$$

If the significance level is $\alpha$, then

$$\alpha/2=1-F(t) \tag{20}$$

where $F(t)$ is the probability distribution function of $t$. (20) gives a standard $t$. When we solve the system, we assume $T\geq t$. Taking $p=q=0.5$, then $T=z/\sqrt{j}$. If $t\geq 3$, the error probability is 0.00135, which is called impossible event in Mathematics.

The method based on Walsh-Hadamard transform was introduced in [17] to solve the linear system with error equations and then used to blindly recognize the convolutional encoder in [18]. But, if $m=(d+1)n$ is big, the complexity of above algorithm is high and the algorithm exceeds the limitation of computer memory. In [19], the authors attempt to solve this problem and propose a modified Walsh-Hadamard transform. One purpose of this paper is to improve the method in [19] and then achieve the blind recognition of the punctured convolutional encoders. Here we use a key fact that $h_{0,n}=1$ according to [20], a convolutional encoder [21] is realizable if the polynomial $H^{(n)}(D)$ is a delay-free polynomial. A polynomial $f(D)=\sum_{i=0}^{\infty}f(i)\cdot D^i$ is called a delay-free polynomial if $f(0)=1$.

We decompose the coefficient matrix $\mathbf{R}$ in (18) into $\mathbf{R}=[\mathbf{R}_1\ \mathbf{R}_2]$ where $\mathbf{R}_1$ is a $(N+1)\times r_1$ matrix, $\mathbf{R}_2$ is a $(N+1)\times r_2$ matrix, and $r_1+r_2=(d+1)\times n$. Similarly, we decompose the variables $\{h_{0,1}\cdots h_{d,1}\cdots h_{0,n}\cdots h_{d,n}\}$ into $[\mathbf{x}_1,\mathbf{x}_2]$, where $\mathbf{x}_1$ is a $r_1$ dimensional vector and $\mathbf{x}_2$ is a $r_2$ dimensional vector. So, (18) transforms to $\mathbf{R}_1\cdot\mathbf{x}_1^{'}+\mathbf{R}_2\cdot\mathbf{x}_2^{'}=0$. For illumination consideration, we suppose $r_1$ is a multiple of $(d+1)$, for example, $r_1=(d+1)\lfloor n/2\rfloor$, where $\lfloor n/2\rfloor$ denotes the integer part of $n/2$. In this case, $\mathbf{R}_1$ and $\mathbf{R}_2$ are respectively equal to

$$\begin{pmatrix} r_{d,1} & \cdots & r_{0,1} & \cdots & r_{d,\lfloor n/2\rfloor} & \cdots & r_{0,\lfloor n/2\rfloor} \\ r_{d+1,1} & \cdots & r_{1,1} & \cdots & r_{d+1,\lfloor n/2\rfloor} & \cdots & r_{1,\lfloor n/2\rfloor} \\ \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ r_{d+N,1} & \cdots & r_{N,1} & \cdots & r_{d+N,\lfloor n/2\rfloor} & \cdots & r_{N,\lfloor n/2\rfloor} \end{pmatrix} \tag{21}$$

and

$$\begin{pmatrix} r_{d,\lfloor n/2\rfloor+1} & \cdots & r_{d,n} & \cdots & r_{0,n} \\ r_{d+1,\lfloor n/2\rfloor+1} & \cdots & r_{d+1,n} & \cdots & r_{1,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ r_{d+N,\lfloor n/2\rfloor+1} & \cdots & r_{d+N,n} & \cdots & r_{N,n} \end{pmatrix} \tag{22}$$

while

$$\mathbf{x}_1=[h_{0,1}\cdots h_{d,1}\cdots h_{0,\lfloor n/2\rfloor}\cdots h_{d,\lfloor n/2\rfloor}],$$

$$\mathbf{x}_2 = [h_{0,\lfloor n/2 \rfloor+1} \cdots h_{d,\lfloor n/2 \rfloor+1} \cdots 1 \cdots h_{d,n}]$$

For $l = 0,1,\cdots,2^{r_1-1}$, suppose the binary representation of $l$ is $\mathbf{x}'_1$. We first compute $\mathbf{R}_1 \cdot \mathbf{x}'_1$, then we obtain a $(N+1)$ dimensional vector $[y_1 \cdots y_{N+1}]$. Sum this vector to the $(r_2 - d)th$ column of $\mathbf{R}_2$. Denote by $\mathbf{R}_2$ the new matrix

$$\begin{pmatrix} r_{d,\lfloor n/2 \rfloor+1} & \cdots & r_{d,n} + y_1 & \cdots & r_{0,n} \\ r_{d+1,\lfloor n/2 \rfloor+1} & \cdots & r_{d+1,n} + y_2 & \cdots & r_{1,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ r_{d+N,\lfloor n/2 \rfloor+1} & \cdots & r_{d+N,n} + y_{N+1} & \cdots & r_{N,n} \end{pmatrix} \quad (23)$$

then

$$\begin{aligned} \mathbf{R}_2 \cdot \mathbf{x}_2^{'} = & \\ & \begin{pmatrix} r_{d,\lfloor n/2 \rfloor+1} & \cdots & r_{d,n} + y_1 & \cdots & r_{0,n} \\ r_{d+1,\lfloor n/2 \rfloor+1} & \cdots & r_{d+1,n} + y_2 & \cdots & r_{1,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ r_{d+N,\lfloor n/2 \rfloor+1} & \cdots & r_{d+N,n} + y_{N+1} & \cdots & r_{N,n} \end{pmatrix} \\ & \times (h_{0,\lfloor n/2 \rfloor+1} \cdots 1 \cdots h_{d,n})' \\ & = \mathbf{R}_1 \cdot \mathbf{x}_1^{'} + \mathbf{R}_2 \cdot \mathbf{x}_2^{'} = 0 \end{aligned} \quad (24)$$

The last linear system $\mathbf{R}_2 \cdot \mathbf{x}_2^{'} = 0$ can be handled by previous method based on the Walsh-Hadamard transform.

**Algorithm1**: Identification of generator matrix $\mathbf{G}(D)$ of mother code and puncturing pattern $\mathbf{P}$.

1: **Input** $\mathbf{H}(D) = [H^{(1)}(D) H^{(2)}(D) \cdots H^{(n)}(D)]$;
2: $M = n - 1$
3: $d = \max_{1 \le i \le n} (\deg H^{(i)}(D))$;
4: **while** ( $\mathbf{P} \in F_2^{2n-2}$ such that the Hamming weight of $\mathbf{P}$ is $n$ and $(\mathbf{P}(2i-1), \mathbf{P}(2i)) \neq 0,\ i = 1,2,\cdots,n-1$ ) **do**
5: **while** ( $(a_0, a_1, \cdots, a_d) \in F_2^{d+1}$ and $(b_0, b_1, \cdots, b_d) \in F_2^{d+1}$ )**do**
6: $g_1(D) = \sum_{i=0}^{d} a_i D^i$ ;
7: $g_2(D) = \sum_{i=0}^{d} b_i D^i$ ;
8: $\mathbf{G}(D) = (g_1(D), g_2(D))$ ;
9: build $\mathbf{G}^{[M]}(D)$ by Theorem 1;
10: build $\mathbf{G}_P(D)$ by Definition 3;
11: **if** $\mathbf{G}_P(D) \cdot \mathbf{H}(D)' = 0$ **then**
12: $\mathbf{G}(D) = \mathbf{G}(D)$ ;
13: $\mathbf{P} = \mathbf{P}$ ;
14: **break;**
15: **end;**
16: **end;**
17: **end;**
18: **Output** the generator matrix $\mathbf{G}(D)$ and $\mathbf{P}$ ;

### C. Identification of the Parity Check Matrix

Now assuming $\mathbf{H}(D)$ the parity-check matrix of punctured convolutional code is identified, the generator matrix of mother code and puncturing pattern can be obtained by the method in [7] or achieved by the following **Algorithm 1** for a special case to accelerate the identification process.

## IV. Experiment Results

Consider $C(2,1,7)$ convolutional code as in Example 4 .This encoder is used in many standards and it is described by the generator matrix and the parity check matrix such that

$$\mathbf{G} = (171 \quad 133) \text{ and } \mathbf{H} = (133 \quad 171)$$

where polynomials are represented in octal. The analysis of the performance is proposed for two punctured convolutional encoders given in Table 1. The method of blind identification of a punctured encoder is divided into two parts: one part is identification of the equivalent punctured code and the other part is identification of the mother code and puncturing pattern. In the experiments, we mainly focus on: (1) the impact of the number of iterations upon the probability of detection and (2) the global performances of probability of detection against the channel error probability $(P_e)$. The iterative process is used to first identify the length n of punctured convolutional code as described in the section III.

TABLE I: Punctured Convolutional Encoders

| $C(n,k,K)$ | $M$ | $\mathbf{P}$ | $C_p(n_p,k_p,K_p)$ |
|---|---|---|---|
| $C(2,1,7)$ | 2 | $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ | $C_p(3,2,4)$ |
| $C(2,1,7)$ | 3 | $\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ | $C_p(4,3,3)$ |

Fig. 1 and Fig. 2 depict probability of detection against channel error probability, for 1, 10, 40 and 50 iterations. From experiment results, we note that the algorithm performances are enhanced by iterations. Moreover, the marginal utility of detection decreases as the number of iterations increases. For example, the global performances of probability of detection with 40 and 50 iterations are very close. We can also see that, to obtain the best performance, the number of iteration should vary with different punctured convolutional encoders. The global performance of probability of detection decreases as channel error probability increases. In the case of $C_p(3,2,4)$, the probability to detect true generator matrix and true punctured pattern proved to be close to 1 for channel error probability less than 0.02 with more than 50 iterations.

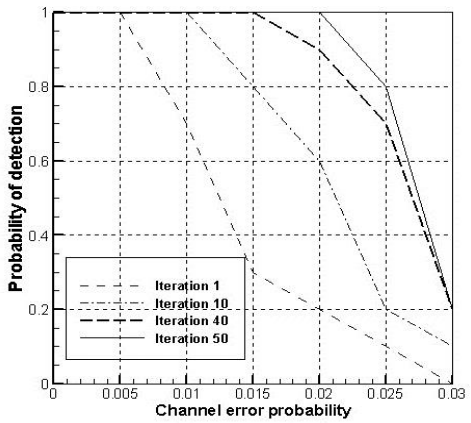


Fig. 1. $C_p(3,2,4)$ probability of detection against $P_e$.

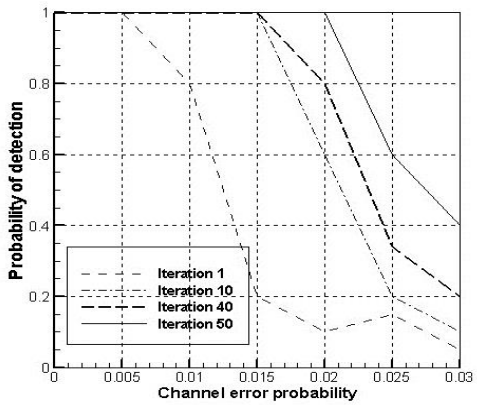


Fig. 2. $C_p(4,3,3)$ probability of detection against $P_e$.

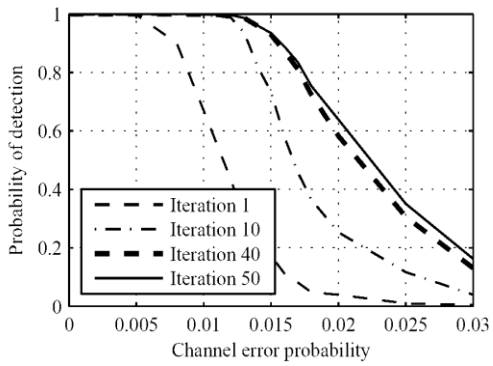


Fig. 3. $C_p(3,2,4)$ probability of detection against $P_e$ in [11].

To evaluate the result of the blind recognition, a comparison between the proposed method (Method I) and the method (Method II) in [11] was drawn. Fig.1 and Fig. 3 depict probabilities of detection against $P_e$, for 1; 10; 40 and 50 iterations for $C_p(3,2,4)$ by Method I and II, respectively. It's obvious that our method performs better significantly both on the impact of the number of iterations upon the probability of detection and the global performances of probability of detection.

## V. Conclusion

We have presented a new solution for recognition of $(n-1)/n$ rate punctured convolutional code from received binary data stream in a noisy environment. The recognition includes generator matrix of mother code and punctured pattern of punctured convolutional encoders. The algorithm offers robustness to noise and higher probability of detection against channel error probability. We achieve this by utilizing the idea of Walsh-Hadamard transform and decomposition of parity check equation. Experiments demonstrate the efficacy of our method with case studies. The probability to detect true generator matrix and true punctured pattern proved to be close to 1 with a $P_e < 0.02$ for $C_p(3,2,4)$ and $C_p(4,3,3)$ with more than 50 iterations. The proposed method outperforms the prior arts.

In the future work, we will extend the blind recognition of convolutional encoders of $(n-1)/n$ rate to the case of rate $k/n$. Then, it will be adaptive to any case rate of equivalent punctured codes and have a wide application scene. Additionally, we will discuss the new algorithm in some more complex channels, e.g., AWGN channels. Moreover, following Walsh-Hadamard transform, it should be possible to consider soft-decision input.

## References

[1] J. Barbier, G. Sicot and S. Houcke, "Algebraic approach for the reconstruction of linear and convolutional error correcting codes," *Int. J. of Applied Mathematics And Computer Science*, vol. 2, pp. 113-118, June 2006.

[2] M. Cluzeau, "Block code reconstruction using iterative decoding techniques," in *Proc. IEEE Int. Symp. on Information Theory*, Seattle, 2006, pp. 2269-2273.

[3] M. Cluzeau and J. P. Tillich, "On the code reverse engineering problem," in *Proc. IEEE Int. Symp. on Information Theory*, Toronto, 2008, pp. 634-638.

[4] M. Cluzeau, "Reconstruction of a linear scrable," *IEEE Trans. on Computers*, vol. 56, pp. 1-8, Jan. 2007.

[5] M. Marazin, R. Gautier, and G. Burel, "Dual code method for blind identification of convolutional encoder for cognitive radio receiver design," in *Proc. IEEE Broadband Wireless Acess Workshop*, Hawaii, 2009, pp. 1-6.

[6] S. Li, P. Lu, X. Luo, and Y. Zou, "Blind recognition of punctured convolutional codes," in *Proc. IEEE Int. Symp. on Information Theory*, Chicago, 2004, pp. 457-464.

[7] P. Lu, S. Li, Y. Zou, and X. Luo, "Blind recognition od punctured convolutional codes," *Sci. in China Ser. F Inf. Sci.*, vol. 48, pp. 484- 498, Aug. 2005.

[8] M. Cluzeau and M. Finiasz, "Reconstruction of punctured convolutional codes," in *Proc. IEEE Information Theory Workshop*, Seoul, 2009, pp. 546-550.

[9] J. B. Cain, G. C. Clark, and J. M. Geist, "Punctured convolutional codes of rate (n-1)/n and simplified maximum likehood decoding," *IEEE Trans. on Information Theory*, vol. 25, pp. 97-100, Jan. 1979.

[10] B. Z. Shen, A. Patapoutian, and P. A. McEwen, "Punctured recursive convolutional encoders and their applications in turbo codes," *IEEE Trans. on Information Theory*, vol.47, pp. 2300-2320, Sep. 2001.

[11] M. Marazin, R. Gautier, and G. Burel, "Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bitstream," *IET Signal Process.*, vol. 6, pp. 122-131, April 2012.

[12] M. Marazin, R. Gautier, and G. Burel, "Blind recovery of k/n rate convolutional encoders in a noisy environment," *ERUASIP*

*Journal on Wireless Communications and Networking*, vol. 168, pp. 1-9, Nov. 2011.
[13] R. Moosavi and E. G. Larsson, "Fast blind recognition of channel codes," *IEEE Trans. on Communications*, vol. 62, no. 5, pp.1393-1405, Mar. 2014.
[14] R. Ahmed, N. Raheleh, X. B. Wang, and Y. Jean, "Blind detection approach for LDPC, convolutional and turbo codes in non-noisy environment," in Proc. *IEEE Conf. on Communications and Network Security*, San Francisco, 2014, pp. 502-503.
[15] K. J. Hole, "Punctured convolutional codes for the 1-D partialresponse channel," *IEEE Trans. on Information Theory*, vol. 37, pp. 808- 817, May 1991.
[16] C. Li, T. Q. Zhang, and Y. Liu, "Blind recognition of RS codes based on galois field columns gaussian elimination," in *Proc. 7th International Congress on Image and Signal Processing*, Dalian China, 2014, pp. 836-841.
[17] L. You and Z. Zhu, "The application of walsh function in resolving of F(2) equations," *Signal Processing*, vol. 16, pp. 27-31, Dec. 2000.
[18] Liu, X. Wang and X. Zhou, "Blind recognition of convolutional coding based on Walsh-Hadamard transform," *J. of Electronics and Inf. Technology*, vol. 32, pp. 884-888, Feb. 2010.
[19] L. Qi, S. Hao, and L. Wang, "Blind decoding algorithm of punctured convolutional codes based on improved WHT," *Application Research of Computers*, vol. 28, pp. 1457- 1464, July 2011.
[20] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*, Wiley-IEEE Press, 1999, pp.145-150.
[21] M. Marazin, R. Gautier, and G. Burel, "Some interesting dual-code properties of convolutional encoder for standards self-recognition," *IET Communications*, vol. 6, no. 8, pp. 931-935, Mar. 2012.

**Wengu Chen** received the Ph.D. degree from Beijing Normal University in 1996 in mathematics. He was a visiting scholar in McMaster University during 2001-2002 and Memorial University of Newfoundland in 2010，respectively. He is now an American mathematics reviewer. He is currently a professor of Institute of Applied Physics and Computational Mathematics, Beijing. His research interests include harmonic analysis, channel coding, and compressive sensing.

**Guoqing Wu** was born in Shandong Province, China, in 1980. He received the B.S. degree in computer science in 2006 and Ph.D. degree in computational mathematics in 2009, both in Graduate Department of China Academic of Engineering and Physics. He is currently an associate researcher of Institute of Applied Physics and Computational Mathematics, Beijing. His research interests include compressive sensing, and information theory.