Blind Recovery of Convolutional Codes over a Noisy Channel

Peiman Gordany, Alireza Keshavarz-Haddad, Ali Jamshidi
School of Electrical and Computer Engineering,
Shiraz University, Shiraz, IRAN
peiman.gordany@gmail.com, keshavarz@shirazu.ac.ir, jamshidi@shirazu.ac.ir

Abstract— **In this paper we study blind recovery methods for identifying the parameters of a convolutional code from an intercepted bitstream. It is assumed that the interceptor has no prior knowledge about the encoder and the recorded bitstream is noisy. The goal is to obtain three parameters of convolutional code including: (i) the number of inputs (ii) number of outputs (iii) the constraint length of the convolutional encoder. Given these parameters one can regenerate the encoder and fully decode the intercepted bitstream. While most related works focus on blind recovery methods for convolutional codes in noiseless bitstream, we propose a method to extract coding parameters from a noisy bitstream. We use these parameters to recover the generator matrix of the encoder and fully decode the bitstream. To evaluate our proposed method, we model the effect of noise on bitstram by binary symmetric channel and burst errors. Our simulation results indicate that the proposed scheme performs accurately in various scenarios.**

Keywords— ***blind recovery; channel coding; linear block code; convolutional code, dual code;***

I. INTRODUCTION

Channel coding is a must in communication systems in order to deal with noisy channels. Some channel codes also named as forward error correcting (FEC) codes add some redundancy to the bitstream for de-noising the received data. In the primarily stages of a communication interception (COMINT) system one needs to distinct the original data bits and the redundant coding bits in a codeword. This is done usually based on blind recovery techniques for channel codes.

Blind recovery methods for channel codes have several applications in regulatory and intelligence organizations whose task is to identify and possibly intercept unknown signals that may violate regulations or pose a security threat. Channel code recognition along with automatic modulation recognition (AMR) is vital for a receiver which task is to identify an unknown signal. Other possible applications of channel codes recognition involve modem reconfiguration for various channel conditions, in which a receiver may automatically adapt the decoder to match the variable coding rate on a transmission. In addition, software defined radio (SDR) schemes may be employed where the encoding format can be changed arbitrarily for different purposes [1]. This study is intended to design a computationally efficient technique by which convolutional codes may be easily detected and recovered.

In this paper, we study blind recovery methods for the convolutional codes and try to extract all parameters needed for decoding the data from a recorded noisy bitstream. Surprisingly, only a few papers deal with the reconstruction of convolutional codes. In the primary work, Rice presented a technique to determine the parameters of convolutional encoders of rate *1/n* [2]. Filiol completed the method by generalizing it to any *k/n* rate [3]. Our paper is not the first to deal with the problem of finding parameters of a convolutional encoder. For instance Marazin has also used dual code properties to find the some parameters of convolutional codes [4]. However, our proposed scheme can identify the number of inputs, number of outputs and the constraint length of the convolutional codes in the presence of noise. Next, we use these parameters to recover the generator matrix of the encoder and fully decode the bitstream.

In simulation section, we study the performance of our method for two types of errors. In the first case, the channel is assumed to be Binary Symmetric Channel (BSC) and in the second case, we add a burst error to the bitstream [5]. Our simulation results indicate that the proposed method can accurately find the parameters of convolutional codes in various scenarios.

In the next section notations for convolutional codes are provided and dual code properties are studied. In section III we propose our technique for blind recovery of convolutional codes. The simulation results are presented in Section IV. Finally, the paper is concluded in Section V.

II. CONVOLUTIONAL CODES AND DUAL CODE PROPERTIES

Convolutional codes were first introduced by Elias in 1955 as an alternative to block codes. Then in 1967, Viterbi proposed a maximum likelihood decoding scheme that was relatively easy to implement for codes with small memory

orders. This scheme, called Viterbi decoding together with improved versions of sequential decoding, led to the application of convolutional codes to deep-space and satellite communications in early 1970s [6]. Let *C* be a convolutional code with '*n*' outputs, '*k*' inputs and 'm' memory input. It will be denoted as $C(n,k,m)$, *n* encoder outputs at any given time unit depend on the *k* inputs at that time and also *m* previous input blocks. Typically, *n* and *k* are small integers with $k<n$, but the memory order *m* must be made large to achieve low error probabilities. In a special case when $k=1$, the information sequence is not divided into blocks and can be processed continuously.

Each convolutional code has a generator matrix; convolutional codes are mainly defined with polynomial multiplications so the generator matrix of these codes is also made of polynomials. Let us denote by *G(D)* a polynomial generator matrix of rank *k* defined by:

$$G(D)=\begin{bmatrix} g_{1,1}(D) & \dots & g_{1,n}(D) \\ \vdots & & \vdots \\ g_{k,1}(D) & \dots & g_{k,n}(D) \end{bmatrix} \tag{1}$$

Each $g_{i,j}(i=1,\dots,n;j=1,..k)$ is a generator polynomial. The memory of a convolutional code is defined as follow:

$$\begin{aligned} m_i &= \max_{j=1,\dots,n} \deg(g_{i,j}(D)) \quad \forall i=1,\dots,k \\ m &= \max_{i=1,\dots,k} m_i = K-1 \end{aligned} \tag{2}$$

Here, *K* in the equation (2) is the constraint length of the code, there are also other definitions for the constraint length but this definition is the most common. This parameter is of a great importance because the cost of Viterbi decoding algorithm depends on number of inputs and the constraint length of the code by 2^{k+K-1} [7].

If the input sequence is denoted by *u(D)* the output sequence will be described by:

$$v(D)=u(D).G(D) \tag{3}$$

In practice the generator polynomials are special polynomials which give the best error correcting capabilities to the code, the encoders which have the best error correcting capabilities are named as optimum encoders. These encoders have the maximum possible free distance among all other encoders with the same parameters *(n, k, m)* [12].

A convolutional code can be described with a dual code generator matrix, this matrix is a $(n-k)\times n$ dimensional matrix and will be denoted by *H(D)*. For a parity check matrix we have:

$$G(D).H^T(D)=0 \tag{4}$$

We can conclude from the equation (4) that for every codeword *V(D)* the following holds

$$v(D).H(D)^T=0 \tag{5}$$

For any parity check matrix there is a general format as equation (6): [4]

$$H(D)=\begin{bmatrix} h_{1,1}(D) & \dots & h_{1,k}(D) & h_0(D) & & \\ \vdots & & \vdots & & \ddots & \\ h_{n-k,1}(D) & \dots & h_{n-k,k}(D) & & & h_0(D) \end{bmatrix} \tag{6}$$

Now we introduce an important property of parity check matrix called dual code memory, from [8, 9] we can compute the dual code memory directly from the generator matrix as:

$$\mu^{\perp}=\sum_{i=1}^{k} m_i \tag{7}$$

where m_i's are defined in equation (2).

So far we have used polynomials to describe convolutional encoders but in practice we have to deal with binary sequences so the binary form of matrices and equations will be very helpful for utilization of properties discussed.

Binary form of the generator matrix can be written as: [6]

$$G=\begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_m & & & \\ & G_0 & G_1 & \dots & G_{m-1} & G_m & & \\ & & G_0 & \dots & G_{m-2} & G_{m-1} & G_m & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & & \ddots & \ddots & \ddots & \ddots \\ & & & & & \ddots & \ddots & \ddots \end{bmatrix} \tag{8}$$

where

$$G_l=\begin{bmatrix} g_{1,l}^1 & g_{1,l}^2 & \cdots & g_{1,l}^n \\ g_{2,l}^1 & g_{2,l}^2 & \cdots & g_{2,l}^n \\ \vdots & \vdots & \vdots & \vdots \\ g_{k,l}^1 & g_{k,l}^1 & \cdots & g_{k,l}^1 \end{bmatrix} \tag{9}$$

Similarly we can write a binary form of parity check matrix which was declared in equation (6), this form is demonstrated in equation (10): [4]

$$H=\begin{bmatrix} H_{\mu^{\perp}} & \dots & H_1 & H_0 & & \\ & H_{\mu^{\perp}} & \dots & H_1 & H_0 & \\ & & H_{\mu^{\perp}} & \dots & H_1 & H_0 \\ & & & \vdots & \vdots & \vdots \end{bmatrix} \tag{10}$$

In equation (10), each sub matrix is:

$$H_i = \begin{bmatrix} h_{1,1}(i) & \dots & h_{1,k}(i) & h_0(i) & & \\ \vdots & \dots & \vdots & & \vdots & \\ h_{n-k,1}(i) & \dots & h_{n-k,k}(i) & & & h_0(i) \end{bmatrix} \quad (11)$$

It is straightforward to show that the rows of the matrix (9) are shifted versions of a vector. As stated in [4] we can use the binary format of parity check matrix along with equation (5) and reconstruct the whole convolutional encoder from the noisy bitstream. But as it can be seen these equations are in need of convolutional encoder parameters, hence, we need to derive these parameters first. In the next section we describe our method for deriving these parameters from the recorded noisy bitstream.

III. The Prposed Method to Recover Coding Paramters

Based on the construction of convolutional codes, the code space *C* is the span of the rows of generator matrix which is a *k* dimensional space. The dual codes are in the complementary orthogonal space of the code space. The dimension of this space will be equal to *n-k*.

We place the bits of the recorded bitstream row by row in a matrix with L columns and large number of rows. This is called the *interception matrix* and denoted by R_L. If the recorded bitstream is not noisy and the number of columns of R_L is set to be a multiple of *n* (*L=2n, 3n, ...*), then the matrix will become a rank deficit, but if *L* is not a multiple of *n,* then the matrix will be full rank with high probability.

When the recorded bitstream is noisy the problem will be more complicated, since R_L is likely to be full rank always. Note that equation (5) shows a linear relation between the columns of the interception matrix. When the bistream is noisy this relation still holds for most entries.

There are two approaches to obtain dual codes from the noisy bitstream. The first approach is to look for low noise submatrices and try to find dual codes from these submatrices [10]. The second approach is to find dual codes from a matrix and check whether this code can be a candidate for dual code basis. Valembois algorithm is a primary work based on this technique [11].

In the case of convolutional codes a similar method can be used; we will use the Valembois algorithm and reform it to find the parameters of a convolutional code.

The rank of interception matrix has two possible values [4]:

If $l \neq \alpha n$ or $l < n_a$

$$\text{Rank}(R_l) = l \quad (12)$$

If $l = \alpha n$ & $l > n_a$

$$\text{Rank}(R_l) = l\frac{k}{n} + \mu^{\perp} < l \quad (13)$$

where $\alpha = 1,2,\dots$ and parameter n_a in equations (12), (13) is the smallest value of *L* for which R_L is rank deficient.

Although the above equations have been derived for block linear code, we can derive some equations in similar form for the convolutional codes.

In our proposed scheme, we create several R_L matrices for *L = n, 2n, 3n,...* . By using equation (13) if *n* is chosen correctly the normalized rank of these matrices will be a decreasing function as follow:

$$\frac{Rank(\mathrm{R}_{\alpha n})}{\alpha n} = \frac{k}{n} + \frac{\mu^{\perp}}{\alpha n} \quad (14)$$

It should be mentioned that this property of convolutional code can be used to decide whether the received bitstream is encoded with a block code or a convolutional code. In the case of block encoder equation (14) will always have one value, and that value will be the rate of the encoder.

For special case of *1/n* rate codes which are very common, we can say $= \mu^{\perp}$, also

$$if \quad \alpha = m \rightarrow \frac{Rank(R_{\alpha n})}{\alpha n} = \frac{2}{n} \quad (15)$$

For other coding rates:

$$if \quad \alpha = \mu^{\perp} \rightarrow \frac{Rank(R_{\alpha n})}{\alpha n} = \frac{k+1}{n} \quad (16)$$

In general case we will find some pairs for $[k, \mu^{\perp}]$ and we can examine each pair and to reconstruct the parity check matrix with these parameters.

As mentioned earlier when we are dealing with noise, the rank cannot be computed directly by reduced row echelon form. Therefore, we compute dual space, and obtain the rank indirectly.

$$Rank(R_{\alpha n}) = \alpha n - \dim(\mathrm{R}_{\alpha n}{}^{\perp}) \quad (17)$$

Valembois algorithm can be used to find $dim(R_{\alpha n}^{\perp})$.

Our proposed method is summarized as follow by two algorithms:

Algorithm1: Checking whether bitstream is encoded with convolutional or block encoder, and obtaining the parameter *n*

```
for  n = 2 to n_max
    for α = 1, 2, 3, ...
        build intercepted matrix R_αn
        compute Rank(R_αn) / α.n
    end
        if values are decreasing
    then it is a convolutional code and
    n is obtained
        break
    end
```

```
      else if  values are constant
            bitstream is block encoded
            n and k are obtained
            break
      end
end
if n is not found
      bitstream    is    not    block    or
convolutional code
end
```

Algorithm2: computing number of inputs and memory

```
for k=1 to n-1
      compute α which is equal to (k+1)/n
      α = μ⊥
      if H is found for (n,k, μ⊥)
            H is parity check matrix
            break
      end
end
```

IV. SIMULATION RESULTS

In this section we present our simulation results on the proposed method. We study the performance of the proposed method for two types of errors: BSC and burst. In a BSC channel the probability of a '0' bit changing to '1' bit is equal to probability of a '1' bit changing to a '0' bit, in other words the channel has an equal crossover probability. Since the error occurring in every single bit is independent of the other bits, the probability of error happening in this channel is modeled as an AWGN channel with hard decoding. In the second case, we add a burst error to the bitstream, in telecommunication a burst error is a contiguous sequence of symbol, received over a data transmission channel, such that the first and last symbols are in error and there exists no contiguous subsequence of q correctly received symbols within the error burst [5]. The integer parameter q is referred to as the guard band of the error burst. The last symbol in a burst and the first symbol in the following burst are accordingly separated by q correct bits or more. The parameter q must be specified when describing an error burst.

Results of the technique proposed earlier in the paper is plotted for a C(3,1,4) encoder in Fig.1, and to demonstrate the performance in presence of noise, a normalized rank graph is plotted for a bitstream encoded via a C(4,1,4) encoder and transmitted through a channel with BSC noise and burst noise respectively in Fig.2 and Fig.3.

In Fig.1, it can be seen that normalized rank which is depicted for n=3, is a decaying function. The red line in the Fig.1 is a constant value equal to $\frac{k+1}{n} = \frac{2}{3}$.

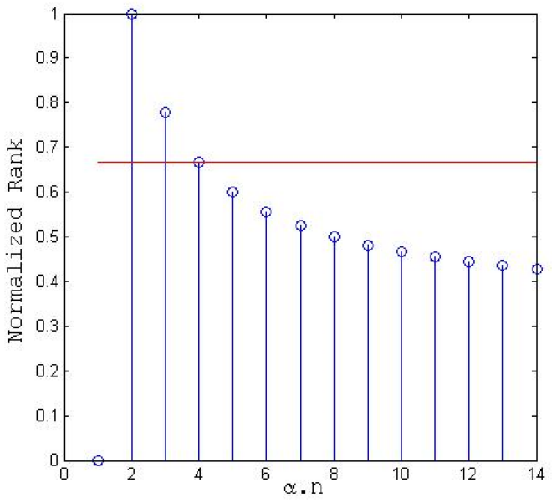

Fig.1. Normalized Rank for C(3,14)

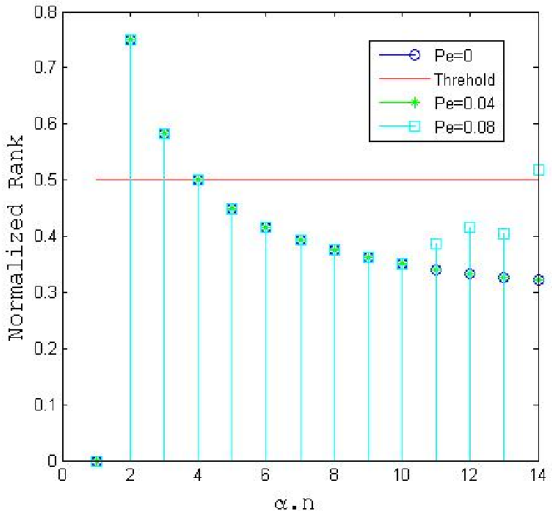

Fig. 2. Normalized Rank for *C(4,1,4)* –BSC Noise

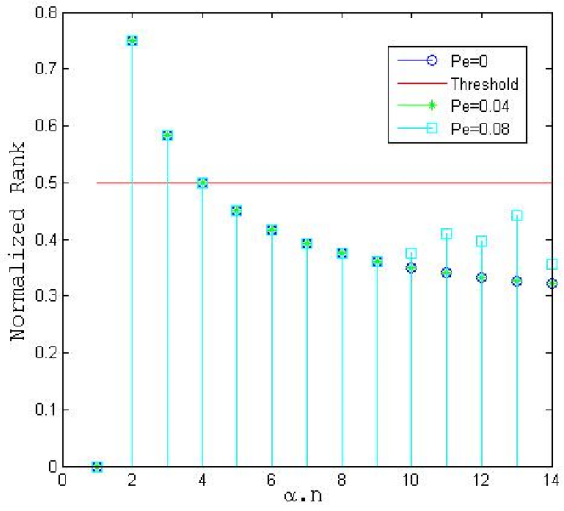

Fig.3. Normalized Rank for *C(4,1,4)* –Burst Noise

In Fig.2 the decreasing manner of the function is still apparent even with error probability of 0.04 and 0.08.

In Fig.3, we have added a burst error to the bitstream, the maximum length of the burst error is q=8, and the probability of occurrence for this error is 0.04 and 0.08.

By using this technique, parameters of the convolutional encoders with different rates are derived and additionally many vectors belonging to dual code space have been found which can be used to reconstruct the parity check matrix.

The parameters which are derived from these algorithms can be further used in Filiol [3] or Marazin [4] algorithm to reconstruct the generator matrix of the code. If the algorithm stated in [4] is used, it is possible to use the vectors of the dual code that have been found through the introduced technique.

V. CONCLUTION

In this paper we studied blind recovery for convolutional codes. We introduced a new method for extracting convolutional coding parameters such as the number of inputs, number of outputs and the constraint length of the convolutional code from a noisy recorded bitstream. Next, we used these parameters to recover the generator matrix of the encoder and fully decoded the bitstream. Our simulation results show that the proposed can correctly recover the coding parameters under various scenarios. Our future work concerns reducing the computational complexity of our convolutional decoder by finding proper decoding matrices.

References

[1] Ziegler, Joseph Frederick. "Automatic recognition and classification of forward error correcting codes". Diss. George Mason University, 2000.

[2] Rice, B. "Determining the parameters of a rate 1 n convolutional encoder over GF (q)." Third International Conference on Finite Fields and Applications. 1995.

[3] Filiol, Eric. "Reconstruction of convolutional encoders over GF (q)." Crytography and Coding. Springer Berlin Heidelberg, 1997. 101-109.

[4] Marazin, Melanie, Roland Gautier, and Gilles Burel. "Blind recovery of k/n rate convolutional encoders in a noisy environment." EURASIP Journal on Wireless Communications and Networking 2011.1, 1-9.

[5] Federal Standard 1037C

[6] Costello, Daniel J., et al. "Error control coding." Fundamentals and Applications, Printice Hall, Upper Saddle River, NJ. 2004.

[7] Cluzeau, Mathieu, and Matthieu Finiasz. "Reconstruction of punctured convolutional codes." Information Theory Workshop (ITW), 2009.

[8] Forney Jr, G. David. "Convolutional codes I: Algebraic structure." Information Theory, IEEE Transactions on 16.6 (1970): 720-738.

[9] Forney Jr, G. DAVID. "Structural analysis of convolutional codes via dual codes." Information Theory, IEEE Transactions on 19.4 (1973): 512-518.

[10] Barbier, Johann, Guillaume Sicot, and Sébastien Houcke. "Algebraic Approach for the Reconstruction of Linear and Convolutional Error Correcting Codes." Enformatika 16 (2006).

[11] Valembois, Antoine. "Detection and recognition of a binary linear code." Discrete Applied Mathematics 111.1 (2001): 199-218.

[12] Frenger, Pal, Pal Orten, and Tony Ottosson. "Convolutional codes with optimum distance spectrum." *IEEE Communications Letters,* 3.11 (1999): 317-319.