Research on Decoding Algorithm for \((n, 1, m)\) Convolutional Code

Zou Yamei \(^1,\, a\), Pu Hongqiao \(^1,\, b\), Xu Zhiqiang \(^2,\, c\)

\(^1\)Chengdu Aeronautic Polythchnic, Sichuan, 610100 China
\(^2\)School of Sichuan College of Media and Communications, Sichuan, 611745, China
\(^a\)zymyaya@163.com, \(^b\)309120411@qq.com, \(^c\)1920281000@qq.com

Keywords: Convolutional code; Decoding; Syndrome; Error correction.

Abstract. This paper takes \((2, 1, 5)\) convolutional codes as example, discusses the decoding algorithm for \((n, 1, m)\) convolutional codes according to the principle of the reversibility of the modulo 2 operation. We designed the decoder and the syndrome of the \((2, 1, 5)\) convolutional codes based on its encoder firstly; then studied on the error correction method used for one or two bits error within a constraint length according to the corresponding relationships between syndrome and the type of error codes; finally validity of this algorithm is verified through the computer simulation.

1. Introduction

The noise and attenuation always exist in the channel of digital communication, it will produce inaccurate code consequently. Improve the reliability and availability of transmission is the goal of the researchers worked in the field of communication all the time, and error control coding is a method to improve the reliability of communication. The research on this coding method is aimed to realize the reliable data transmission by reducing error rate by coding and decoding under the condition of certain transmission efficiency and make sure the decoder is as simple as possible.

Error control coding can be classified as block code and convolutional code according to information processing method. It proved that the performance of the later is better than the former at the same error rate and the same complexity of the devices condition both in theory and in practical, and the convolutional code gets high coding gain in the Additive White Gaussian Noise channel, and error code rate of 10\(^{-5}\) can be realized). Owing to its superior performance, the convolutional code is widely used in satellite communication, space communication and mobile communication today, it takes an important role in improving reliability of digital communication.

2. Convolutional codes

Convolutional code was presented by Elas of MIT in 1955, it is a type of method to deal with burst error codes, and always be marked as \((n, k, m)\), in which \(k\) information bits constitute \(n\) bits, that is the coding efficiency can be denoted by the equation \(R_c = k/n\), and constraint length is \(N\), that means there are \(Nk\) shifting registers, \(n\) modular 2 adders, \(n\) shifting registers are output, and the structure of the encoder can be illustrated in fig 1.

Fig 1. Structure schematic of convolutional codes

There are three main decoding methods of convolutional codes: one is the threshold coding raised by Massey in 1963, another is the sequential decoding introduced by Wozencraft in 1961 then improved by Fano in 1963, the last is the Viterbi algorithm proposed by Viterbi in 1967. Viterbi algorithm has gained great popular in many standards such as IS-95, GSM, wireless local area and...
3G mobile communication system since it introduced.
It has been studied for so many years, and there are some matured algorithms of convolutional
codes, but seek a new simple and good performance coding algorithm is of great importance and
practicality.

3. \((n, 1, m)\) convolutional codes

\((2,1,5)\) convolutional codes presented in bibliographic reference[2] is a good method without no
error propagation in which the constraint length is 5 with the maximum free distance \(d_f\), and the
code rate \(R\) is 1/2. Let’s go on studying the decoding methods of \((n,1,m)\) convolutional codes by
taking \((2,1,5)\) for example.

3.1 \((2, 1, 5)\) convolutional encoding algorithm

The subgenerator of \((2,1,5)\) convolutional encoder is
\[g^{(1,1)} = (73)_k = (111,011), g^{(1,2)} = (61)_k = (110,001)\], as indicated in fig 2. There are 5 storage units
in encoder, the registers from left to right are \(m_{k-1}, m_{k-2}, m_{k-3}, m_{k-4}, m_{k-5}\). Every input
code \(m_k\) will produce two output codes \(c_{k1}\) and \(c_{k2}\).

Fig 2. \((2, 1, 5)\) convolutional encoder

in which, \(c_{k1} = m_k \oplus m_{k-1} \oplus m_{k-2} \oplus m_{k-4} \oplus m_{k-5}\)
\(c_{k2} = m_k \oplus m_{k-1} \oplus m_{k-5}\)

The output encoding sequence is \(c = (c_{11}c_{12}, c_{21}c_{22}, \cdots, c_{k1}c_{k2}, \cdots)\), each code segment
includes two code elements.

3.2 Algorithm of \((2, 1, 5)\) convolutional codes

The reversibility of modular 2 operation is the theoretical foundation of decoding algorithm of
\((n,1,m)\) convolutional codes discussed in this article, that is if \(a \oplus b = c\), then \(a \oplus c = b\) and
\(b \oplus c = a\), where \(a,b,c \in \{0,1\}\). so the decoder of \((2,1,5)\) convolutional codes can be expressed in fig 3.

Fig 3. \((2, 1, 5)\) convolutional decoder

As indicated in fig 3, the input of decoder are \(R_{k1}\) and \(R_{k2}\), and the output of decoder are \(a_k\)
and \(b_k\), \(a_k\) can be feedback to the storage unit of decoder(result will be the same if \(b_k\) is
feedback). We can get the following formulas from the figure:
\[a_k = R_{k1} \oplus a_{k-1} \oplus a_{k-2} \oplus a_{k-4} \oplus a_{k-5}\]
\[b_k = R_{k2} \oplus a_{k-1} \oplus a_{k-5}\]
There is no inaccurate codes if \( R_{k1} = c_{k1}, R_{k2} = c_{k2} \) after transmission in the channel, and \( a_k = b_k = m_k \), we can get the conclusion that decoding is correct, if there are error bits in \( R_{k1} \) or \( R_{k2} \), then the output of decoder will be incorrect. However, errors can not be avoided in practical communication, so first we should analyze the type of error codes, then design the rule for error correction, and complete decoding rightly.

For ease of analysis, the input of decoder in 6 continuous time \( (k, k+1, k+2, k+3, k+4, k+5) \) of (2,1,5) convolutional codes are expressed in \( (R_{k1}, R_{k2}, \ldots, R_{k+5}) \), respectively.

If there are errors in 1 or 2 time, it will produce 48 error types according to analysis. While error bit rate in practical communication systems are \( 10^{-3} \sim 10^{-6} \), and correct 1 or 2 random errors can meet the needs of common communication systems, so only need to discuss 23 types of error codes as indicated in table 1.

<table>
<thead>
<tr>
<th>error code in 1 time</th>
<th>( R_{k1} )</th>
<th>( R_{k2} )</th>
<th>( R_{k1} R_{k2} )</th>
<th>( R_{k1} R_{k1} )</th>
<th>( R_{k1} R_{k2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{k1} )</td>
<td>( R_{k1} R_{k1+1} )</td>
<td>( R_{k1} R_{k1+2} )</td>
<td>( R_{k1} R_{k1+3} )</td>
<td>( R_{k1} R_{k1+4} )</td>
<td>( R_{k1} R_{k1+5} )</td>
</tr>
<tr>
<td>( R_{k1} )</td>
<td>( R_{k1} R_{k1+1} )</td>
<td>( R_{k1} R_{k1+2} )</td>
<td>( R_{k1} R_{k1+3} )</td>
<td>( R_{k1} R_{k1+4} )</td>
<td>( R_{k1} R_{k1+5} )</td>
</tr>
<tr>
<td>( R_{k1} )</td>
<td>( R_{k1} R_{k1+1} )</td>
<td>( R_{k1} R_{k1+2} )</td>
<td>( R_{k1} R_{k1+3} )</td>
<td>( R_{k1} R_{k1+4} )</td>
<td>( R_{k1} R_{k1+5} )</td>
</tr>
<tr>
<td>( R_{k1} )</td>
<td>( R_{k1} R_{k1+1} )</td>
<td>( R_{k1} R_{k1+2} )</td>
<td>( R_{k1} R_{k1+3} )</td>
<td>( R_{k1} R_{k1+4} )</td>
<td>( R_{k1} R_{k1+5} )</td>
</tr>
<tr>
<td>( R_{k1} )</td>
<td>( R_{k1} R_{k1+1} )</td>
<td>( R_{k1} R_{k1+2} )</td>
<td>( R_{k1} R_{k1+3} )</td>
<td>( R_{k1} R_{k1+4} )</td>
<td>( R_{k1} R_{k1+5} )</td>
</tr>
</tbody>
</table>

From table 1, we can find that if there is no error in \( R_{k1} \) and \( R_{k2} \), then \( a_k \oplus b_k = 0 \); if one error code occurs in \( R_{k1} \) and \( R_{k2} \), then \( a_k \oplus b_k = 1 \); if both \( R_{k1} \) and \( R_{k2} \) are not correct, then \( a_k \oplus b_k = 0 \). So if \( a_k \oplus b_k = 1 \), it indicates that there must be error codes exist in time k; while \( a_k \oplus b_k = 0 \) can not prove the codes are correct in time k. So, suppose: if \( a_k \oplus b_k = 0 \), the codes are correct; if \( a_k \oplus b_k = 1 \), there must be error codes. Of course the assumption can not judge that both \( R_{k1} \) and \( R_{k2} \) are incorrect simultaneously, it can judge that there are incorrect codes in time k, but it can not give the type of error codes.

When there is no error codes, \( R_{k1} = c_{k1}, R_{k2} = c_{k2} \), \( m_k \) can be got from \( c_{k1} \) and \( c_{k2} \); while if there are error codes \( e_{k1} \) and \( e_{k2} \), then \( R_{k1} = c_{k1} \oplus e_{k1}, R_{k2} = c_{k2} \oplus e_{k2} \).

In order not to confuse, assume the output of decoder which have no error codes during transmission are \( A_k \) and \( B_k \), and \( A_k = B_k = m_k \), then

\[
\begin{align*}
a_k &= R_{k1} \oplus a_{k-1} \oplus a_{k-2} \oplus a_{k-4} \oplus a_{k-5} \\
&= c_{k1} \oplus e_{k1} \oplus A_{k-1} \oplus A_{k-2} \oplus A_{k-4} \oplus A_{k-5} = A_k \oplus e_{k1} \\
b_k &= R_{k2} \oplus a_{k-1} \oplus a_{k-5} = C_{k2} \oplus e_{k2} \oplus A_{k-1} \oplus A_{k-5} = B_k \oplus e_{k2}
\end{align*}
\]

And the syndrome \( s_k = a_k \oplus b_k = e_{k1} \oplus e_{k2} \) is only related to the error pattern.

Thus the syndrome S at 6 continuous time of 23 error types can be gained as indicated in table 2, in which \( S = (s_k, s_{k+1}, s_{k+2}, s_{k+3}, s_{k+4}, s_{k+5}) \). From table 3, we can get that:

1. The form of S is “xxxxxx”, \( x \in \{0,1\} \), while there are 32 datas of this type, 17 are in table 2, which is lack of other 15 which are “100011”, “100111”, “101001”, “101101”, “110100”, “110110”, “110001”, “110101”, “111010”, “111011”, “111100”, “111101”and“111111”.

2. In table 2, there is one-to-one correspondence between error codes types in the left column
and S, while the error codes types in the right column has a vague relationship with S. So we need to build the one-to-one correspondence between error codes types and syndrome to eliminate the “vague relationship”.

Table 2  S after classification

<table>
<thead>
<tr>
<th>No</th>
<th>error codes type</th>
<th>S</th>
<th>No</th>
<th>error codes type</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$R_{k_1}$</td>
<td>100000</td>
<td>12</td>
<td>$R_{k_1} R_{(k+2)l}$</td>
<td>100101</td>
</tr>
<tr>
<td>2</td>
<td>$R_{k_2}$</td>
<td>101000</td>
<td>13</td>
<td>$R_{k_2} R_{(k+3)l}$</td>
<td>100101</td>
</tr>
<tr>
<td>3</td>
<td>$R_{k_2} R_{(k+3)l}$</td>
<td>100100</td>
<td>14</td>
<td>$R_{k_1} R_{(k+3)l}$</td>
<td>101011</td>
</tr>
<tr>
<td>4</td>
<td>$R_{k_1} R_{(k+2)l}$</td>
<td>100110</td>
<td>15</td>
<td>$R_{k_2} R_{(k+2)l}$</td>
<td>101011</td>
</tr>
<tr>
<td>5</td>
<td>$R_{k_1} R_{(k+3)l}$</td>
<td>101010</td>
<td>16</td>
<td>$R_{k_1} R_{(k+4)l}$</td>
<td>101100</td>
</tr>
<tr>
<td>6</td>
<td>$R_{k_1}$</td>
<td>101110</td>
<td>17</td>
<td>$R_{k_1} R_{(k+4)l}$</td>
<td>101100</td>
</tr>
<tr>
<td>7</td>
<td>$R_{k_2} R_{(k+1)l}$</td>
<td>110000</td>
<td>18</td>
<td>$R_{k_1} R_{(k+5)l}$</td>
<td>101111</td>
</tr>
<tr>
<td>8</td>
<td>$R_{k_2} R_{(k+1)l}$</td>
<td>110111</td>
<td>19</td>
<td>$R_{k_1} R_{(k+5)l}$</td>
<td>101111</td>
</tr>
<tr>
<td>9</td>
<td>$R_{(k-2)l} R_{(k-2)l}$</td>
<td>111000</td>
<td>20</td>
<td>$R_{k_2} R_{(k+3)l}$</td>
<td>100010</td>
</tr>
<tr>
<td>10</td>
<td>$R_{k_1} R_{(k+2)l}$</td>
<td>111001</td>
<td>21</td>
<td>$R_{k_2} R_{(k+4)l}$</td>
<td>100010</td>
</tr>
<tr>
<td>11</td>
<td>$R_{k_1} R_{(k+2)l}$</td>
<td>111110</td>
<td>22</td>
<td>$R_{k_2} R_{(k+5)l}$</td>
<td>100001</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>23</td>
<td>$R_{k_2} R_{(k+5)l}$</td>
<td>100001</td>
</tr>
</tbody>
</table>

Since the specific error codes types can not be known at 6 time, more observation time should be added to judge. Only 1 added time can judge the error codes types when $S = (101100)$ or $S = (100010)$. While 2 observation time should be added when $S = (100101)$, $S = (101011)$, $S = (101111)$ or $S = (100001)$ as indicated in table 3, in which syndrome at added observation time is $t_k$. The premise of this conclusion is: there is no error codes at time $k + 6$ and $k + 7$. If the condition is not set up, it beyond it’s capability.

Table 3  Judgement of error codes type with “vague relationship”

<table>
<thead>
<tr>
<th>No</th>
<th>error codes type</th>
<th>S</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$R_{k_1} R_{(k+2)l}$</td>
<td>100101</td>
<td>100101, 10</td>
</tr>
<tr>
<td>2</td>
<td>$R_{k_2} R_{(k+3)l}$</td>
<td>100101</td>
<td>100101, 11</td>
</tr>
<tr>
<td>3</td>
<td>$R_{k_1} R_{(k+3)l}$</td>
<td>101011</td>
<td>101011, 11</td>
</tr>
<tr>
<td>4</td>
<td>$R_{k_2} R_{(k+2)l}$</td>
<td>101011</td>
<td>101011, 10</td>
</tr>
<tr>
<td>5</td>
<td>$R_{k_1} R_{(k+4)l}$</td>
<td>101100</td>
<td>101100, 1</td>
</tr>
<tr>
<td>6</td>
<td>$R_{k_1} R_{(k+4)l}$</td>
<td>101100</td>
<td>101100, 0</td>
</tr>
<tr>
<td>7</td>
<td>$R_{k_1} R_{(k+5)l}$</td>
<td>101111</td>
<td>101111, 01</td>
</tr>
<tr>
<td>8</td>
<td>$R_{k_1} R_{(k+5)l}$</td>
<td>101111</td>
<td>101111, 00</td>
</tr>
<tr>
<td>9</td>
<td>$R_{k_2} R_{(k+4)l}$</td>
<td>100010</td>
<td>100010, 1</td>
</tr>
<tr>
<td>10</td>
<td>$R_{k_2} R_{(k+4)l}$</td>
<td>100010</td>
<td>100010, 0</td>
</tr>
<tr>
<td>11</td>
<td>$R_{k_2} R_{(k+5)l}$</td>
<td>100001</td>
<td>100001, 01</td>
</tr>
<tr>
<td>12</td>
<td>$R_{k_2} R_{(k+5)l}$</td>
<td>100001</td>
<td>100001, 00</td>
</tr>
</tbody>
</table>
3.3 Decoding rule

We can get decoding rules by analyzing as follows:

1. If \( a_k \oplus b_k = 0 \), codes are correct, and decoding process can continue.

2. If \( a_k \oplus b_k = 1 \), codes are incorrect, specify the time as \( k \), and \( s_k = a_k \oplus b_k \), work out the value of syndrome \( S \) at 6 continuous time.

*If \( S = (100011) \) or \( S = (100111) \), \( S = (101001) \), \( S = (101101) \), \( S = (110010) \), \( S = (110100) \), \( S = (110110) \), \( S = (111011) \), then there must be error codes, and beyond correction capability.

*The error codes type can be sure if \( S \) is any one of 11 types in left side in table 2. Do modulo 2 addition with number 1 and error codes respectively, change the datas in register units with \( a_{k-1} \), \( a_{k-2} \), \( a_{k-3} \), \( a_{k-4} \), \( a_{k-5} \) from left to right, then redecoding the datas from time \( k \). For example: when \( S = (100000) \), the \( R_{k2} \) is not correct, do \( R_{k2} \oplus 1 \), change the datas in decoder, then do the decoding.

*If \( S \) is one of the datas in table 3, error codes type can be known according to the \( T \) by adding 1 or 2 observation time, do modulo 2 addition with 1 and error codes respectively. Modify the data in register units in decoder with \( a_{k-1} \), \( a_{k-2} \), \( a_{k-3} \), \( a_{k-4} \), \( a_{k-5} \) from left to right, then do the decoding from time \( k \). For example: when \( S = (100101) \), 2 observation time \( k + 6, k + 7 \) should be added to calculate the value of \( T \). Provided \( T = (100101,10) \), \( R_{k1}, R_{(k+2)l} \) are not correct, do \( R_{k1} \oplus 1 \) and \( R_{(k+2)l} \oplus 1 \), update the datas in decoder, then decode; If \( T = (100101,11) \), then \( R_{k2}, R_{(k+3)l} \) are incorrect, do \( R_{k2} \oplus 1 \) and \( R_{(k+3)l} \oplus 1 \), modify the datas in decoder, then decode again.

(3) Calculate \( S \) again after redecoding, if \( S \neq (000000) \) or \( T \neq (000000,00) \), then we can judge it beyond correction capability.

Either transmit the datas again or give up the data packdges can be done when beyond its correction capability.

4. Simulated analysis of decoding algorithm

The flow diagram designed according to decoding rule is as fig 4.

![Flow diagram](image)

Fig 4 the flow diagram of (2, 1, 5) decoding algorithm

Combine the module built in simulink with the fixed program, transmit the information sequence, the relation curve between error bit rate and \( E_b/n_0 \) decoded in the way discussed in this article and in Viterbi decoding way are shown in fig 5.
The BER performance designed in this article can not compare with Viterbi decoding method as shown in fig 5, which is in accordance with theoretical analysis. The reason is that the program is immature compared with the simulation program supplied by Matlab, it’s impossible to make comparation of their speed accurately.

5. Summary

This paper analyze the decoding method in detail and summarize the decoding rules by taking (2, 1, 5) convolutional code for example, though this decoding rule can not applied in other convolutional code, but the whole analyze method can be used in (n,1,m) convolutional code especially when the number m is small.

The decoding analyze method and decoding principle of (n,1,m) convolutional code mentioned in this paper is easy to understand, it is characterized by fast speed, small calculating amount and easy implementation. The advantage of this convolutional code is obvious especially applied in the situation of demand of low bit error rate even though it may be immature and of less performance compared with Viterbi , for example, it’s of great practical value in voice and image communication.

Reference


