Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/14/20230792

Performance analysis of several common error correction codes

Xiwen Du

Xidian University, Xi'an, China

1321211105@qq.com

Abstract. With the rapid development of information technology and advances in science and technology, people are constantly interacting with data and coding. Error correction codes are integral to ensuring the security, robustness, and accuracy of information and systems, used not only in various engineering fields but in our daily lives as well. Systems that incorporate error correction codes have been proven to perform significantly better than those that do not, which makes it essential to understand the principles behind coding, encoding techniques, and their performance in specific domains.This paper focuses on three widely used error correction codes: Hamming code, RS code, and convolutional code. By introducing each code separately and evaluating their performance in different practical areas using proper simulations and relevant research results, this paper provides a comprehensive understanding of the vital role of error correction codes in channel encoding. Each code has its strengths, weaknesses, and specific applications. Hamming code, for example, is suitable for correcting bit errors in computer memories and communication systems. RS code is widely used in digital storage and broadcast systems due to its high level of error correction capability. Convolutional code is ideal for correcting bit-errors in continuous channels, making it an excellent choice for wireless communication systems. Overall, the results demonstrate that these error correction codes are crucial to guaranteeing data transmission integrity and system robustness. Understanding their principles and performance can guide future research and development of coding techniques in advancing technological progress and improving our daily lives.

Keywords: Code performances, Hamming code, RS code, Convolutional code

1. Introduction

In the transmission of digital signals, errors often occur in the transmitted data stream due to various reasons, resulting in image hopping, discontinuity, mosaic, and other phenomena at the receiving end [1]. Therefore, through the channel coding process, the digital stream is processed accordingly, so that the system has certain error correction and anti-interference capabilities, which can greatly avoid the occurrence of code errors in the transmission of code streams [2].

Error code processing techniques include error correction, interleaving, linear interpolation, and so on [3]. Improving data transmission efficiency and reducing error rate is the task of channel coding [4]. The essence of channel coding is to increase the reliability of communication [5]. However, channel coding can reduce the transmission of useful information and data. The process of channel coding is to insert some symbols into the source data stream to achieve the purpose of error judgment and correction

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/14/20230792

at the receiver, which is often referred to as overhead [6]. Hamming Code is a linear debugging code in the field of telecommunications, named after the inventor Richard Wesley Hamming. Hamming code inserts a verification code into the transmitted message stream. When a computer stores or moves data, it may generate data bit errors to detect and correct single bit errors. Due to their simplicity, hamming codes are widely used in memory (RAM). RS code is a linear error correction code with strong error correction performance, which can correct random and burst errors. RS code is a multi-band BCH code that can correct multiple symbol errors simultaneously. Because the data is related to binary polynomial sliding, it is called convolutional code. Convolutional codes are widely used in communication systems.

2. Hamming code

2.1. Coding principle

(m, k) Hamming codes are linear packet codes with the ability to correct errors. The code is transmitted by M-bit oversight bits that are introduced into the original information sequence to guarantee the code's ability to correct errors. T = 1 is also a full code for rectifying errors. The formula (1) determines the number of information bits k and the length m of the Hamming code.

$$(\mathrm{m},\mathrm{k}) = 2\mathrm{m}-1,2\mathrm{m}-\mathrm{m}-1 \tag{1}$$

In the above formula, m is the oversight bit, m=n-k. Take (7,4) Hamming code as an example to illustrate the coding principle of Hamming code. Suppose the resulting matrix G is shown in Equation (2).

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{2}$$

Given the sequence of information M shown in equation (3), according to equation (4) The corresponding Hamming codeword C for this information sequence M can be calculated [7].

$$M = m_3 m_2 m_1 m_0 \tag{3}$$

$$C = M \cdot G = (c_6 c_5 c_4 c_3 c_2 c_1 c_0) \tag{4}$$

Since the information sequence's length k=4, it can be calculated using equation (4). Between each code element and the information bits in the codeword, there are a total of 16 codewords. Equation (5) depicts the connection; 16 codewords match the informational flow.

$$\begin{cases} c_6 = m_3 \\ c_5 = m_2 \\ c_4 = m_1 \\ c_3 = m_0 \\ c_2 = m_3 \oplus m_2 \oplus m_1 \\ c_1 = m_3 \oplus m_2 \oplus m_0 \\ c_0 = m_3 \oplus m_1 \oplus m_0 \end{cases} \tag{5}$$

2.2. Decoding principle

According to the connection between the generation matrix G and the check matrix H shown in equation (6), the check matrix H is obtained as shown in equation (2) from the generation matrix G shown in equation (7). O in equation (6) stands for a 4 * 3 matrix of zeros [8].

$$G \cdot H^T = O \tag{6}$$

DOI: 10.54254/2755-2721/14/20230792

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Let the receive sequence R be shown in Equation (8) and the corresponding transmit code word be C, then E = R - C or E = R + C is called the error pattern, and define E as shown in Equation (9).

$$R = (r_6 r_5 r_4 r_3 r_2 r_1 r_0) \quad (8)$$

$$E = (e_6 e_5 e_4 e_3 e_2 e_1 e_0) \quad (9)$$

Define the syndrome S as shown in Equation (10).

$$S = RH^T = EH^T \quad (10)$$

The result of the operation of RH^T in Equation (11) is defined as the syndrome S = ($s_2 s_1 s_0$), which can be used to verify whether the transmission is wrong and correct the corresponding error [9]. If the receive sequence R is the sent codeword, it means that no error occurred during channel transmission, and the relation $S=RH^T = CH^T$ = (0 0 0). Conversely, if an error occurs, it can be decoded by the decoding table corresponding to the accompanying formula and the error pattern shown in Table 1 Suppose R = (1000110) and get the syndrome S = (001), Table 1 shows that E = (0000001) and thus the estimated value of the code word issued C' = (1000111).

Table 1. Decoding table 1.

Syndrome S	Error pattern E
000	0000000
001	0000001
010	0000010
100	0000100
011	0001000
101	0010000
110	0100000
111	1000000

2.3. Simulation analysis

The bit error rate specifications for a digital communication system must be met by the system itself and any connected components. Since bit error rate is a crucial metric for gauging system performance, bit error rate testing issues are common in the field of digital communication. The bit error rate is the probability of an error occurring after the binary bitstream is transmitted by the system. Its measurement method involves entering a predetermined form of bitstream from the system's input end, detecting its output, and comparing it with the input code stream to identify the bit e where the error occurred. The bit error rate is calculated as the ratio of the number of bits e to the total number of bits n that have been transmitted [10]. Which is,

$$BER = e/n \quad (11)$$

For the simulation parameters, we use Bernoulli binary pseudo-random number sequence as the signal generator, use $3*10^3$ sample rate, do the real-time simulation and take $1e*10^{10}$ as the number of the sampling points. The actual measured bit error rate is a theoretical estimate. The measurement accuracy depends on the test duration or the amount of bits broadcast, and in order for the measurement to be accurate enough, it must last for a long enough test period or transmit enough bits.

The detailed simulation process is as follows,

DOI: 10.54254/2755-2721/14/20230792

- Imitate the actual experiment to build the simulation model, the simulation model mainly contains the signal generation part, the channel encoding part, binary sending part, the decoding part and the error code analysis part.
- Set the model parameters, the main parameter variables that the simulation use is: sampling rate, samples per frame, encoding format, sample number, decoding format and the receiving delay.
- Under the aforementioned parameters, we use the Hamming encoding to do the simulation under the certain possibility of the channel error rat, to verify the encoding algorithm's improvement on the channel performance, which is the error rate.
- When the channel bit error rate gradually increases, the channel performance of the simulation coding algorithm is how improved.

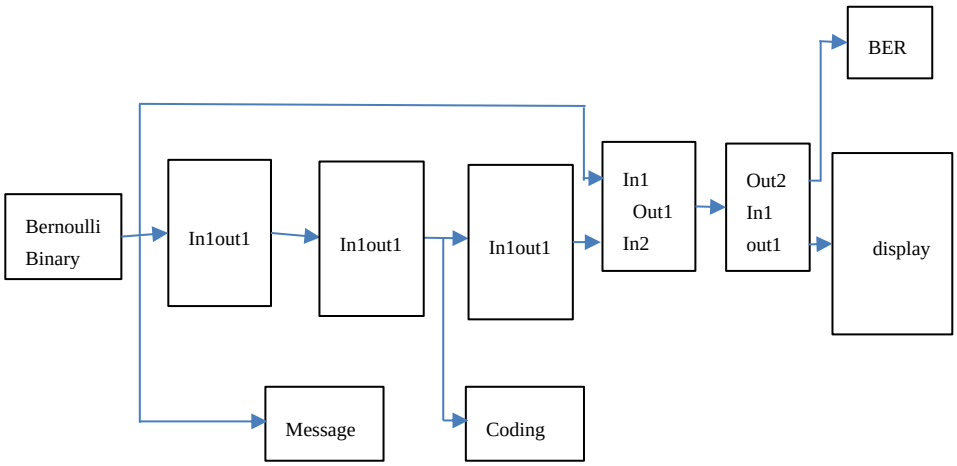

Figure 1. Simulation model.

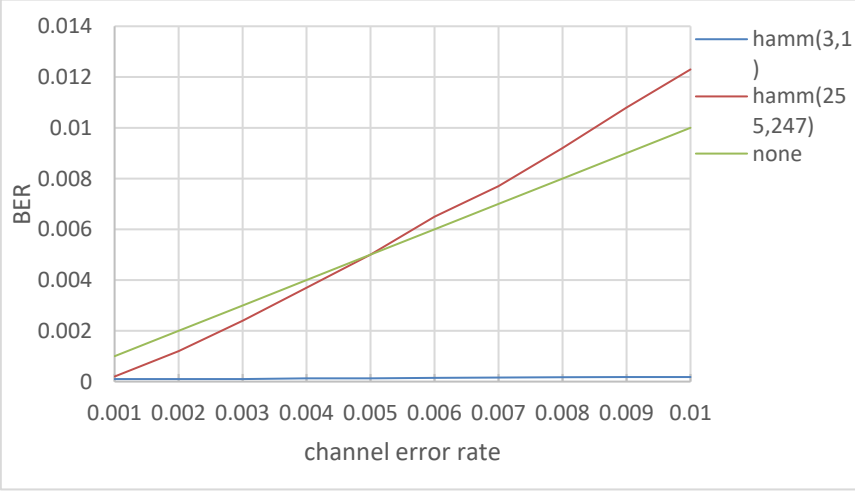

Figure 2. Hamming coding error probability figure.

From the figure 1 and 2 that the BER is significantly decreased under the hamming (3,1), it performs perfectly at any channel error rate, but the hamming (255,247) performs not that great, though it has 96.86% encoding efficiency, but only when the channel error rate is between 0.001 and 0.005 did the hamming (255,247) performs better than the uncoded one.

DOI: 10.54254/2755-2721/14/20230792

2.4. Application scenarios

Hamming Code has a wide range of applications. Digital telemetry remote control systems, electrical and digital communication, image transmission over satellites, deep space communication. The applications of Hamming code in c watermarking systems is listed below.

Application of Hamming Code in FEC watermarking system The bit error rate of the watermark in the FEC watermarking system depends on the watermark's capacity, spread spectrum gain, embedding strength, and bitrate of the error-correcting code. The in-information redundancy provided by the error correction algorithm lowers the embedding strength of the watermark in the watermark channel, which is constrained by the watermark's imperceptibility. When the scaling factor is small, the extracted watermark's bit error rate rises as a result of the significant change in image quality, and the watermark essentially stays the same after the brightness scaling shift. As of now, if the use of Hamming code is added, the error rate of the watermark is also reduced dramatically when the scaling factor is low. It is known from existing literature that when the scaling factor is between 0.1 and 0.2, the bit error rate of the extracted watermark utilizing Hamming code is decreased from the original 0.09 to less than 0.06, a reduction of about 33%. so that the quality of the extracted watermark is significantly improved.

3. RS code

3.1. Introduction of RS code

In 1960, Irving Reed and Gus Solomon first proposed RS code in the form of multi-term codes whose compilation process was based on a finite field. Because finite fields were discovered by the French mathematician Evariste Galois, they are also called Galo fields. For any q that satisfies the upper form, the finite field GF(q)'s different structures are isomorphism. Thus, a finite field can be fully described according to its size.

3.2. Code construction

The basic construction of RS code is the polynomial. Assume we have k information symbols, denote as$\{m_0,m_1,\ldots m_{k-2},m_{k-1}\}$, each of them is the element in the GF(q). These symbols can be used to form a polynomial like equation (11)

$$P(x) = m_0 + m_1 x + \cdots + m_{k-2}x^{k-2} + m_{k-1}x^{k-1} \tag{12}$$

The code word **c** can be formed by taking value of the polynomial P(x) on different q places in the GF(q).

$$c = \left(c_0, c_1, \cdots c_{q-1}\right) = (P(0), P(\alpha), \ldots, P(\alpha^{q-1})) \tag{13}$$

The set of all codewords for an RS code is constructed by all possible values on the finite field consisting of k information symbols. This RS code comprises q^k different codewords since each info symbol might have q different values. If the linear combination of any two codewords remains the codeword in the code, the code is said to be linear. Equations 11 and 12 tell us that the product of any two polynomials with powers less than k-1 is another polynomial with powers less than k-1, proving that the RS code is a linear code. The dimension of the RS code is the value of the information symbol k. This is so because the RS code's codeword creates a finite field's GF(q) in a k-dimension space. We defined the length n of the code word as n=q since each codeword can only contain a maximum of q symbols. When the RS code was first established, we frequently referred to it as the (n, k) code and used the parity number t=n-k. The other Linear packet codes are the same.

3.3. Decoding

The codeword of the RS code is as equation (13)

DOI: 10.54254/2755-2721/14/20230792

$$\begin{aligned} P(0) &= m_0 \\ P(\alpha) &= m_0 + m_1\alpha + m_2\alpha^2 + \cdots + m_{k-1}\alpha^{k-1} \\ P(\alpha^2) &= m_0 + m_1\alpha^2 + m_2\alpha^4 + \cdots + m_{k-1}\alpha^{2(k-1)} \\ &\vdots \\ P(\alpha^{q-1}) &= m_0 + m_1\alpha^{q-1} + m_2\alpha^{2(q-1)} + \cdots + m_{k-1}\alpha^{(k-1)(q-1)} \end{aligned} \tag{14}$$

The k-variable system of linear equations can be solved using any k equation. The MDS code's property, which is the minimum distance equal to its parity bits plus one, also applies to the RS code. After rewriting the previous equation in matrix form, we obtain equation (14)

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{(k-1)} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(k-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{q-1} & \alpha^{2(q-1)} & \cdots & \alpha^{(k-1)(q-1)} \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_{k-1} \end{bmatrix} = \begin{bmatrix} P(0) \\ P(\alpha) \\ P(\alpha^2) \\ \vdots \\ P(\alpha^{q-1}) \end{bmatrix} \tag{15}$$

The matrix on the left is the Generator Matrix of the RS code. Because the arbitrary sub-matrix of the Generator Matrix is the Van der Ment matrix, and the elements in the $\{0,\alpha,\ldots,\alpha^{q-1}\}$ are differ from each other, so this matrix in invertible. During the process of decoding, we can arbitrarily take k symbols of the (n, k) RS code to solve the k variables $\{m_0,m_1,\ldots,m_{k-1}\}$.

3.4. Simulation analysis

Like the way we used in the hamming code performance analysis, we get the result of the RS code simulation.

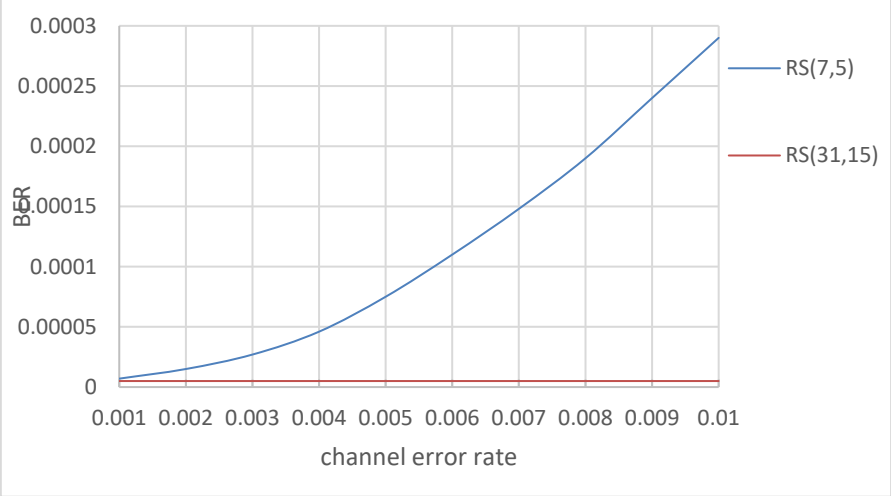

Figure 3. RS coding error probability figure.

We can conclude that the RS(7,5) have the best error correction performance, RS(7,5) and RS(31,15) can correct 1 and 8 bit error separately, RS(7,5) is a short code and has a high coding efficiency. Compared with RS (31,15), RS (31,15) is currently the preferred code for military communications, and its error correction performance is far superior to RS (7,5). As shown in Figure 3.

3.5. Application scenarios

Application of RS Code in ATM Switch

DOI: 10.54254/2755-2721/14/20230792

In wireless communication, ATM switches are utilized. As signals are subjected to the effects of topography and terrain, gradual fading occurs due to the relatively weak channel conditions in wireless communication. More critically, multipath effects lead to frequency selective fading since different signals produce different delays. Practical measurement reveals that short string mistakes and random error codes frequently occur in the received signal, substantially impairing communication. When ATM switches use RS error correction and decoding modules, the error rate can be efficiently decreased.

4. Convolutional code

4.1. Introduction of convolutional code

Elias proposed the convolutional code in 1955. It is also necessary to extract pertinent information from the code sets received at earlier and later times when decoding. Each group of convolutional codes also typically has modest information bits and code lengths. This is so because n and k are quite modest and the correlation between groups is completely leveraged during convolutional code encoding. And hence, it has been demonstrated both theoretically and practically that convolutional codes perform at least as well as other codes under the same code rate R and equipment complexity circumstances, and that it is also simpler to reach optimal and quasi optimal results. However, consistent performance analysis is very challenging in engineering applications due to the correlation between groups of convolutional codes, and the number of findings from analysis is limited. To find good codes, it is frequently required to use computer search.

4.2. Encoding

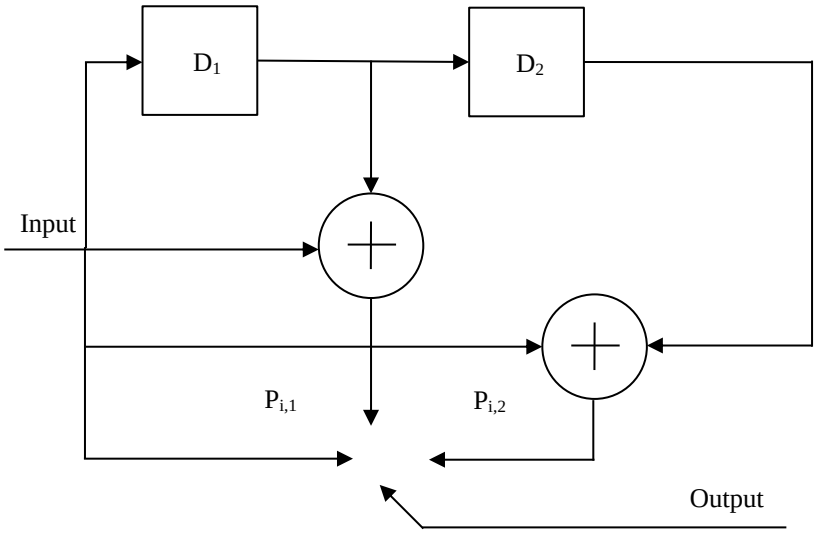

Figure 4. Binary convolutional coding system.

As the figure 4 shows, If each time unit inputs a new information element m_i into the encoder and the data in the memory is shifted to the right by one bit, on the one hand, the m_i is directly output to the channel, and on the other hand, it is operated with the m_{i-1} and m_{i-2} input from the previous two time units according to the rules determined by the lines in the figure to obtain the two inspection elements $p_{i,1}$, $p_{i,2}$ at this time, followed by the m_i to form a subcode $c_i=(m, p_{i,1}, p_{i,2})$ to be sent to the channel.

$$\begin{cases} p_{i,1} = m_i + m_{i-1} \\ p_{i,2} = m_i + m_{i-2} \end{cases} \tag{16}$$

The information element input for the next time unit is m_{i+1}, which corresponds to the two corresponding test elements:

DOI: 10.54254/2755-2721/14/20230792

$$\begin{cases} p_{i+1,1} = m_{i+1} + m_i \\ p_{i+1,2} = m_{i+1} + m_{i-1} \end{cases} \quad (17)$$

The second subcode composed of $c_{i+1} = (m_{i+1}, p_{i+1},1, p_{i+1,2})$ is sent to the channel, and so on. At each time unit, it is sent to the encoder k_0 (1 in this example) information elements, and the encoder sends corresponding n_0 (3 in this example) symbol groups to form a subcode c_i into the channel.

4.3. Decoding

When the code constraint is small, the Viterbi algorithm, which Viterbi introduced in 1967, is more effective, quicker, and simpler than sequential decoding algorithms. As a result, the Viterbi algorithm has advanced extremely quickly both in theory and in practice since it was first proposed, and it is now extensively used in many different types of data transmission systems, particularly satellite communication systems. As a result, we will only outline the stages of the Viterbi decoding algorithm below:

(1) The partial path metric is computed for all partial paths of j segment branches entering each state starting from a specific time unit of j=m. Choose and save the partial route that has the highest metric and its metric value for each state. This partial path is then referred to as the reserved (surviving) path.

(2) The reserved pathways that enter each state at this time are obtained, stored, and all other paths are deleted by adding 1 to the metrics of all the branches that enter each state at this time and the metrics of the reserved paths connected to these branches at the previous time. Thus, one branch is added to the reserved routes.

(3) If j< L + m, repeat the above steps, otherwise stop, and the decoder obtains the path with the maximum path metric.

4.4. Simulation analysis

Like the way we used in the hamming code and RS code performance analysis, we get the result of the convolutional code simulation.

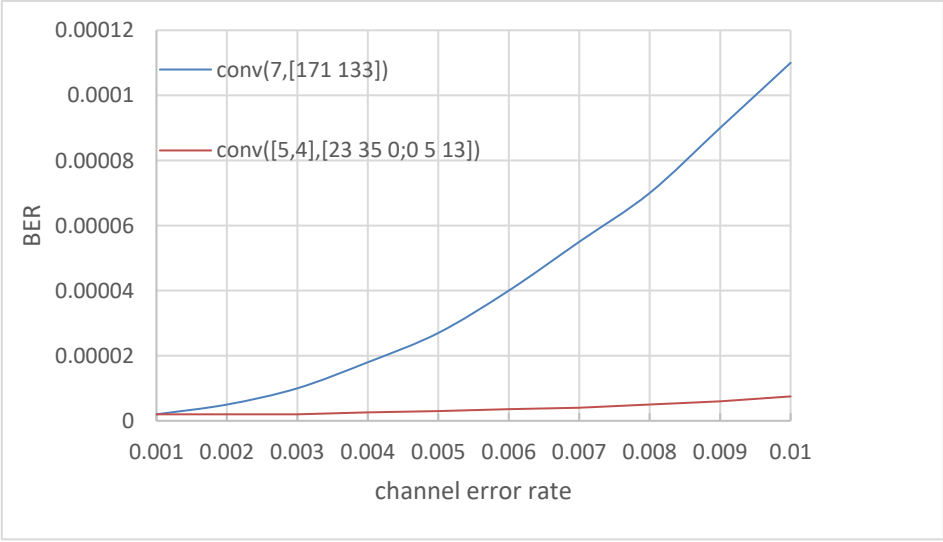

Figure 5. Convolutional coding error probability figure.

From the figure 5, the Conv ([5,4], [23,35,0; 0,5,13]) has better error correction performance than conv (7, [171133]), and can achieve higher coding efficiency. By sacrificing coding efficiency, the error rate can be reduced by 2-3 orders of magnitude, thereby meeting the basic requirements of communication.

DOI: 10.54254/2755-2721/14/20230792

4.5. Application scenarios

The most cutting-edge mobile communication system that was extensively used in the 1990s is known by its initials, GSM, which stand for digital cellular mobile communication system. Based on GSM, GPRS seeks to increase the speed of GSM data transmission and satisfy the demands of the global mobile data market. GPRS still uses convolutional code technology in channel coding, just like GSM, despite adopting an efficient data transmission method based on packet switching, which transfers packet data services between the air interface and external networks and seamlessly connects with existing data services.

5. Conclusion

This article introduces three different channel coding methods, namely, Hamming code, RS code, and convolutional code, and conducts relevant simulations and uses in corresponding fields. Through this article, people can have a good understanding of the relevant knowledge of channel and error correction coding, understand the similarities and differences between different codes, and application scenarios. Based on the inspiration of this article, they can develop new coding methods that are more efficient and combine the advantages of various codes, In the future, the author will also focus on cutting-edge knowledge in relevant fields and expect to make relevant contributions in the field of coding.

References

[1] Rakovic V, Adamovski R, Risteski A, et al. Improving energy efficiency and reliability in WuR-based IoT systems: An error correction approach[J]. Wireless Personal Communications, 2020: 1-12.

[2] Mei F, Chen H, Lei Y. Blind Recognition of Forward Error Correction Codes Based on Recurrent Neural Network[J]. Sensors, 2021, 21(11): 3884.

[3] Wang J, Li J, Huang H, et al. Fine-grained recognition of error correcting codes based on 1-D convolutional neural network[J]. Digital Signal Processing, 2020, 99: 102668.

[4] Moon T K. Error correction coding: mathematical methods and algorithms[M]. John Wiley & Sons, 2020.

[5] Marey M, Mostafa H. Power of Error Correcting Codes for SFBC-OFDM Classification Over Unknown Channels[J]. IEEE Access, 2022, 10: 35643-35652.

[6] Farbeh H, Delshadtehrani L, Kim H, et al. ECC-United cache: Maximizing efficiency of error detection/correction codes in associative cache memories[J]. IEEE Transactions on Computers, 2020, 70(4): 640-654.

[7] Ar-Reyouchi E M, Rattal S, Ghoumid K. A Survey on Error–Correcting Codes for Digital Video Broadcasting[J]. SN Computer Science, 2022, 3(2): 105.

[8] Patel M, de Oliveira G F, Mutlu O. HARP: Practically and effectively identifying uncorrectable errors in memory chips that use on-die error-correcting codes[C]//MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture. 2021: 623-640.

[9] Evron I, Onn O, Orzech T W, et al. The Role of Codeword-to-Class Assignments in Error-Correcting Codes: An Empirical Study[J]. arXiv preprint arXiv:2302.05334, 2023.

[10] Vaz A C, Nayak C G, Nayak D, et al. Performance analysis of Forward Error Correcting Codes in a Visible Light Communication System[C]//2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). IEEE, 2021: 1-5.