Bluetooth Low Energy Error Correction Based on Convolutional Coding

Yue Cao¹,†, Zaixin Liu²,*,†, Longyu Wu³,†

¹School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing, China
²Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, 77840, United States of America
³School of Information Science and Technology, Nantong University Nantong, China

*Corresponding author’s e-mail: Liuzx98@tamu.edu
†These authors contributed equally.

Abstract—Contemporarily, the Internet of Things (IoT) is recently a newly emerging technology for connecting small devices into a platform; the IoT has been an increasingly demanded front-edge technology in terms of connecting different devices using information transmission and storage technology. To adapt to the small capacities of device batteries, Bluetooth Low Energy is adopted as the protocol of communication. However, the existing standards do not have a suitable and specific error correction method. As there is no ideal information transmission channel, there must be errors that occurred during message transmission. The performance and capacity of error correction become decisive factors in evaluating how efficient the IoT communication system performs. This article uses convolutional coding—a better-performing coding scheme than block coding—to correct errors in information transmission and reception on Internet of Things devices. It is better competent to control and correct bit errors in information transmission. To achieve this goal, convolutional coding algorithms devised by Dr Justin Coon at the University of Oxford have been referred to. By simulation using MATLAB, it has been found that the error rate is enhanced significantly for high Signal-to-Noise Ratio (SNR) in convolutional codes compared to uncoded messages.

1. Introduction

The Bluetooth Low Energy (BLE) standard is one of the most promising Internet of Things (IoT) solutions. Compared with classic Bluetooth, BLE aims to maintain the same communication range while reducing power consumption and cost. Digital binary coding schemes can usually be classified into two types: block code and convolutional code. In [1], convolutional code, compared to block code, can control and correct errors better. Most of the existing designs use Cyclic Redundancy Check (CRC) to correct errors. The CRC code has a great correction capability. However, the CRC code will consume too much energy. In this case, the convolutional codes were chosen to be our error-correcting codes.

This research mainly compares and analyzes three Bluetooth Low Energy 5.0 (BLE5.0) physical layer patterns to explore their differences and characteristics. At the same time, this research summarizes the advantages of GFSK modulation which is always used in Bluetooth Low Energy 4.0...
(BLE4.0) and later versions. A suitable convolutional coding is designed based on the characteristics of physical layer transmission. By utilizing MATLAB simulation, the performance of the design above can be demonstrated. The design of convolutional coding is compliant with the characteristics of the physical layer transmission. On this basis, the performance comparison of the convolutional codes and block codes proves that the convolutional codes perform better under the same condition.

The rest of this paper is organized as follows. A standard Bluetooth error correction mechanism is introduced in detail. The error correction mechanism of the Bluetooth system can be divided into forward error correction (FEC) and ARQ scheme for data. FEC supports both 1/3 rate and 2/3 rate error correction codes. We adopted different schemes according to the different bit error rates required for voice data and control information. Through discussing the basic principles and applications of several schemes, it can be found that the 1/3 scale FEC scheme is a relatively simple error-correcting code manner. 1/3 scale FEC scheme can be automatically corrected. ARQ scheme only guarantees the correct transmission of payload, while it cannot protect data and voice information. Objectively speaking, the error correction mechanism adopted by Bluetooth is not the most advanced technology in the current field. Bluetooth adopts more straightforward 1/3 rate repetition codes and 2/3 rate Hamming codes instead of more advanced convolutional codes and Turbo codes with the same coding rate to improve decoding speed and reduce commercial costs.

2. Methodology
In this work, we first collect relevant theoretical knowledge. Through the integration of the corresponding knowledge, we determine the direction of research. With this direction, we design and simulate a convolutional coding in MATLAB. By analyzing the experimental results, the convolutional coding achieves good performance. Compared with other error-correcting codes, convolutional code is more suitable for BLE.

2.1. Theory Collection and integration

2.1.1. Bluetooth Low Energy Physical Layer
The research uses the BLE protocol for transmission. BLE 5.0 PHY Layer has three patterns: two of them are uncoded physical layer standards. The encoded physical layer standard has two different coding methods.

2.1.2. GFSK Modulation
GFSK modulation is always used in Bluetooth BLE4.0 and later versions. The reasons for using GFSK modulation are summarized in this section.

2.1.3. Design Direction
According to the analysis of the theory above, we compare different error-correcting codes. Considering the cost and performance, we finally choose convolutional code to be our error-correcting code.

2.2. Two Types of Channel Codes

2.2.1. Convolutional Code
As the type of channel codes formed by simple mathematical operations, convolutional codes are adopted the most in current communication applications. This type of code provides a reliable capability of error detection and correction.

To generate concatenated convolutional codes, there needs to be a string of raw binary source codes. The string is then pushed into the buffer bit by bit. There are two raw output strings, and these two strings are different based on distinct bit combinations selected for modulo arithmetic. Selected bits on each side operate modulo arithmetic by two and generate two raw output bit strings which are
usually different because the two modulo units select different bits. After all source bits and bits in the buffer are exhaustively pushed, the two raw output strings are concatenated bitwise alternatively in the opposite direction, i.e., from right to left. The algorithm then once takes one bit from the upper string and then takes another one from the lower string. The concatenated string is the convolutional code.

2.2.2. **Block Coding**
Compared to convolutional codes, block codes are much simpler. If the message has the length of $k$ bits, there will be $(n-k)$ bits added after that forms codewords with the length of $n$ bits. Usually, people utilize Hamming code, Bose, Chaudhari and Hocquenghem (BCH) code and Reed Solomon code the most among block codes. Their linearity and cyclicity often evaluate block codes. Linear block codewords are derived from combining generator words. In the generator matrix, $k$ and $n$ are the number of rows and columns, respectively. $k$ information bits are concatenated with $(n-k)$ parity bits through the block coder that forms $n$-bit codewords.

Block codes can be generated by multiplying all possible binary numbers from 0 to $2^{k-1}$ sequentially with the generator matrix or adding combinations of two generators while adding the $n$-length all-0 codeword into the codeword collection.

2.2.3. **Hamming Code**
The minimum number of different bits between any two codewords within a codeword set in Hamming code is the minimum Hamming distance. This distance can be simply found by counting the minimum number of 1s in a codeword among all codewords except for the all-0 codeword. With the
Hamming distance, the error-correcting capacity can be calculated by
\[
\frac{(d_{\text{min}} - 1)}{2}.
\]
Error correction and data storage are two of the most common applications of Hamming code.

2.2.3.1. Reed Solomon Code
Reed Solomon code is a derivative of the BCH code. A Reed Solomon codeword consists of \( n \) symbols, including \( k \) data symbols and \( 2t \) parity symbols. \((n-k+1)\) is the minimum distance of Reed Solomon codewords. \( 2^{n-k} \) equals to the ability Reed Solomon codes can correct errors.

For more background knowledge about convolutional and block codes, including their performance analyses, it is recommended to refer to articles [2-5].

2.3. The error correction mechanism of the Bluetooth system

2.3.1. 1/3 scale FEC scheme
1/3 scale FEC scheme is a relatively simple error-correcting code way, which uses a simple 3-fold repetition format to repeat each message 3 times. It then decodes it by using the large number decision method at the receiving end [6].

2.3.2. 2/3 scale FEC scheme
1/3 scale FEC scheme is a relatively simple error-correcting code way, which uses a simple 3-fold repetition format to repeat each message 3 times. It then decodes it by using the large number decision method at the receiving end. In the 2/3 scale FEC scheme, \((15,10)\) shortened Hamming codes are used to generate polynomial
\[
g(D) = (D+1)(D^4 + D + 1)
\]
The linear feedback shift registers for generating this cyclic code are shown in Figure. 3 [6].

2.3.3. ARQ scheme
Bluetooth adopts a fast unnumbered ARQ scheme. ARQ scheme’s basic principle is, after sending a packet, if the originator does not receive the acknowledgment signal (ACK) correctly received by the other party, the originator will resend the packet. The acknowledgement information is placed in the header of the response packet sent to the other party, and whether the received information is correct is completed by the cyclic redundancy check code (CRC) in the payload. The 16-bit CRC check code is generated by using CCITT polynomial 210041 (octal) [7, 8].

2.3.4. Error Checking
To ensure the reliable transmission of data, it is necessary to check whether the identification access code is correct when receiving packets, to prevent erroneous reception of information from other piconets. Since the source of the 64-bit synchronization word in the access code comes from the low address part (LAP) of the 24-bit master unit, it is possible to check whether the LAP is correct and receive information from other piconets. HEC and CRC detection usually include High-end addresses (UAP). Even if a message can be detected by access code, it will still be discarded if UAP does not match. Before calculating HEC and CRC [9], the shift register of the HEC/CRC generator is initialized.
with 8-bit UAP. Header and payload into HEC and CRC generators, respectively. The generation and verification processes of HEC and CRC codes are shown in Figure. 4 and Figure. 5.

![Figure 4. HEC generator and detection](image1.png)

![Figure 5. CRC generator and detection](image2.png)

**HEC** the generation polynomial is

\[
g(D) = (D^1 + D^2 + D^3 + D^4 + D^5 + 1) = D^6 + D^3 + D^2 + D + 1
\]  

(1)

When coding, the circuit is initialized with 8-bit UAP, and the data enters the encoder bit by bit. When the last bit enters, all parity bits have been generated.

**CRC** the generation polynomial can be represented as:

\[
g(D) = D^6 + D^3 + D^2 + 1
\]  

(2)

At the beginning of coding, the leftmost 8 bits are first loaded with 8-bit UAP, and the rightmost 8 bits are set to 0.

3. **Result**

3.1. **BLE PHY Layer**

3.1.1. The BLE5.0 PHY Layer has three patterns:

| Table 1. BLE5.0 PHY Layer |
|---------------------------|
| Symbol Rate | Bit Rate |
| LE 1M PHY    | 1 Msym/s | 1 Mbps  |
| LE 2M PHY    | 2 Msym/s | 1 Mbps  |
| LE encode PHY| 1 Msym/s | 125kbps (s=8) |
|              |          | 500kbps (s=2) |
- LE 1M PHY and LE 2M PHY are low energy uncoded physical layer standard. Neither of them has an error correction stage [10].
- b) LE encode PHY have two different encoding methods, s=8 and s=2. s is the number of symbols per bit [10].

3.1.2. The comparison of low energy uncoded physical layer standard and low energy encode physical layer standard is as follows:

![Figure 6. Uncoded physical layer standard](image1)

![Figure 7. Coded physical layer standard](image2)

3.1.3. GFSK modulation: BLE4.0 and later versions all used GFSK modulation. The Gaussian filter optimizes the conversion from one symbol to another. If there is no Gaussian filter, the frequency conversion will be too fast to cause much noise. More than that, GFSK can also control the modulated signal spectrum by changing the 3dB bandwidth of the Gaussian low-pass filter. GFSK modulation has many advantages like a constant envelope, concentrated power spectrum and narrow spectrum [11].

3.2. Design direction
In the beginning, we have many choices like CRC, block codes and convolutional codes. But to save energy, we won’t transfer too much data, so we exclude CRC. And in the same coding condition, convolutional codes perform better than block codes [12]. Finally, we choose convolutional codes to be our Error-correcting codes.

3.3. Simulation Results

3.3.1. By mapping bits— 0 and 1— to the two BPSK symbols— “-1” and “1”— and simulating 7-bit convolutional codes, we have got a result that for convolutional codes with Additive White Gaussian Noise (AWGN), the bit error rate (BER) is the highest while the Signal to Noise Ratio (SNR)’s value falls in the range between 1 and 2. When the SNR exceeds 3, especially when it is 10 or higher, the SNR is very low.
3.3.2. When the size of the sample increases to 10,000, the BER vs SNR curve plot becomes less abrupt. In the low SNR region, the error rate is relatively high. However, the BER drastically decreases after the SNR exceeds 2 on the linear scale.

3.3.3. Compared to uncoded binary messages with AWGN, the convolutional coding scheme significantly reduces the error rate once the SNR exceeds 2. In this way, we can conclude that convolutional coding is an efficient coding scheme in ensuring the codes' precision and accuracy.
4. Conclusion
We use convolutional codes to design an error-correcting code for Bluetooth Low Energy (BLE). As one of the most promising Internet of Things (IoT) solutions, BLE does not have a standard error correction code. In order to be applied in IoT, the BLE needs an appropriate error correction code to improve the transmission efficiency. Based on BLE physical layer transmission characteristics, it can be found that convolutional code is better for BLE. With theoretical assist, we design a convolutional error-correcting code in MATLAB. From the simulation results, the performance of the convolutional error-correcting code can be demonstrated.

In the future, the IoT will become mainstream. At the same time, the BLE will have more applications. For example, smart wearable devices are perfect for the BLE. With the BLE and a suitable error-correcting coding, the smart wearable devices battery will last longer, and the reliability of the smart wearable devices will also be greatly improved. Based on these advantages, people can create a variety of smart wearable devices. Furthermore, we can connect all the tools that we use in our daily life. The low power consumption and reliability reduce the cost without the performance taking a hit. A garden, a water tank, or even a sunshade could be regarded as part of our connection network. Users can receive data and manage equipment through the terminals. The production and operation of the whole society will be changed. We will enter a more efficient and convenient era.

References
[1] S. R. Chopra, J. Kaur, and H. Monga, “Comparative performance analysis of block and convolutional codes,” Indian Journal of Science and Technology, vol. 9(47), December 2016.
[2] L. V. Truong, “Performance of Viterbi decoding with and without ARQ on Rayleigh fading channels,” IEEE Transactions on Communications, October 2018.
[3] M. R. D. Rodrigues, I. Chatzigeorgiou, I. J. Wassell, and R. Carasco, “On the performance of turbo codes in quasi-static fading channels,” ISIT, pp. 622-626, September 2005.
[4] Y. Chen and C. Wei, “On the performance of rate $\frac{1}{2}$ convolutional codes with QPSK on Rician fading channels,” IEEE Transactions on Vehicular Technology, vol. 39, pp. 161-170, May 1990.
[5] K. Ramasamy, M. U. Siddiqi, M. Y. Alias, and A. Arunagiri, “Performance comparison of convolutional and block turbo codes,” IEICE Electronics Express, vol. 3, pp. 322-327, July 2006.
[6] Yufeng Shan, Su Yi, Shivkumar Kalyanaraman, and W. Woods. John, “Two-Stage FEC scheme for scalable video transmission over wireless networks,” Signal Processing: Image Communication, vol. 24, no. 9, pp. 718-729, 2009.

[7] Huang Ying-zong, Sanjeev Mehrotra, and Jin Li, “A hybrid FEC-ARQ protocol for low-delay lossless sequential data multimedia and Expo 2009,” ICME 2009. IEEE International Conference, pp. 718-725, 2009.

[8] A. S. Panda, and G. L. Kumar, “Comparison of serial data-input CRC and parallel data-input CRC design for CRC-8 ATM HEC employing MLFSR,” 2014 International Conference on Electronics and Communication Systems (ICECS), pp. 1-4, 2014

[9] P. Wang, L. Song, S. Yu, and L. Yang, “Analysis and comparison of FEC and FEC-ARQ protection schemes based on RS and Raptor code,” 2010 International Conference on Wireless Communications & Signal Processing (WCSP), pp. 1-6, 2010.

[10] Bluetooth Core Specification Version 5.0, December 2016.

[11] G. Kennedy, Electronic Communication Systems, New York: McGraw-Hill, 1984.

[12] S. Y. Mui and J. W. Modestino, "Comparison of Convolutional and Block Code Performance on the Rician Channel", National Telecommunications Conference, vol. 2, pp. 20:4-1-20:4-7, 1977.