[UCI]I804:48009-200000109560

RESEARCH ON EFFICIENT TRANSMISSION FOR DIMMING CONTROL IN VISIBLE LIGHT COMMUNICATIONS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

BY

He Wang

School of Electrical Engineering

Graduate School of the University of Ulsan

May, 2018

RESEARCH ON EFFICIENT TRANSMISSION FOR DIMMING CONTROL IN VISIBLE LIGHT COMMUNICATIONS

A THESIS

SUBMITTED

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

BY

He Wang

UNDER THE SUPERVISION OF

Prof. Sunghwan Kim

School of Electrical Engineering

Graduate School of the University of Ulsan

May, 2018

RESEARCH ON EFFICIENT TRANSMISSION FOR DIMMING CONTROL IN VISIBLE LIGHT COMMUNICATIONS

Approved by:

Prof. Sungoh Kwon
School of Electrical Engineering (Signature)
University of Ulsan

Prof. Sunghwan Kim
School of Electrical Engineering (Signature)
University of Ulsan

Prof. Sangjo Choi
School of Electrical Engineering (Signature)
University of Ulsan

Prof. Kyoung-Young Song
Ulsan College (Signature)

Prof. Min-Ho Jang
Ulsan College (Signature)

Date Approved: May, 2018

ACKNOWLEDGMENTS

During the doctor course，I experienced and learned a lot of things. I would like to thank many people who have contributed to this thesis.

First of all, I would like to take this opportunity to express my deepest gratitude to my advisor Professor Sunghwan Kim. His brilliant insight, patient guidance and countless encouragement throughout the last three year helped me to complete the writing papers, as well as Ph.D degree successfully, and find a confidence. It is very honored for me to study with him. Without his consistent and illuminating instruction, the completion of this thesis would have never become possible.

My heartfelt thanks go to the professor Hyungyun Kong and Sungoh Kwon. They have been wise and trusted professors. In addition, they gave me the opportunity to develop the knowledge of detection & estimation, probability & random process, respectively; which is important and helpful for me to study in communication coding filed.

I would also like to thank the professor Yun Mi Hwang, who had an English writing lecture for me. She gave me instruction so that I had a chance to develop the skill of academic writing such as constructing a research paper, data commentary, writing summaries and writing critiques.

Moreover, I would like to thank the professor Changrim Jang, who taught me the elementary algebra coding theory. Through his lectures, I had deeper understanding about my research content. I would also like to thank the professor Young Soo Suh, who taught me estimation theory. I learned how to use Kalman filter in his lecture.

I would like to thank all my colleagues of coding and information theory laboratory, they are also my friends who always gave me a cheering and understanding such as Taesoo Kim, Fatema Khanam, Muhammad Awais Jadoon, Nguyyen Manh Duc, Dao Trong Hieu, Le Tran Manh, Ranman Manzoor.

I am very much grateful to University of Ulsan for giving me such a wonderful research environment and financial support. This university gave me the chance to conduct a number of

useful courses with excellent instructors that will give me a way to think and work properly in the rest of my life.

Finally, this acknowledgement would be incomplete without mentioning my family. I sincerely thank my parents for devotion, encouragement, and education. Without heir unconditional love and support, my dream of this thesis would never have become true. I would like to specially thank my beloved girlfriend (Jia-Hong Li) for her endless love, understand, and support.

University of Ulsan

May, 2018

He Wang

SUMMARY

RESEARCH ON EFFICIENT TRANSMISSION FOR DIMMING CONTROL IN VISIBLE LIGHT COMMUNICATION

He Wang

School of Electrical Engineering

The Graduate School

University of Ulsan

In this thesis, first we consider 50% dimming case with run-length limited (RLL) codes, and propose a new decoding algorithm based on maximum likelihood (ML) decoding of RLL codes with on-off keying (OOK) modulation and Reed-Solomon (RS) codes in visible light communication (VLC) systems. Conventional RLL codes in VLC systems are used for 50% dimming and hard decision of RLL decoder is considered. However, in the receiver of our model, the RLL decoder based on ML decoding makes better estimation for channel decoder. Simulation results show that our proposed method get signal-to-noise ratio (SNR) gain in bit error rate (BER) compared with one of RLL hard decoding.

In order to improve performance, then a modified decoding algorithm based on multiple output candidates for a RLL decoder in VLC with OOK modulation and RS codes is studied. In this receiver model, the proposed RLL decoder produces multiple candidates with greater probabilities to enhance the throughput in the VLC system. We also propose a selection method based on a cyclic redundancy check (CRC). A significant advantage of our proposed method is that it leads to performance enhancement without any change in the transmitter in VLC systems. Simulation results show that the BER of our proposed method is better than that of RLL hard decoding; the performance also improves further with a larger number of candidates.

Then, we proposed a symbol level soft (SLS) output RLL decoding algorithm since previous ML decoding belong to hard output RLL decoding (even with multiple candidates), the soft output of RLL decoding can provide more information for concatenated error correction codes (ECC), here consider RS codes. Moreover, conventional RS codes adopt Berlekamp-Massey (BM) algorithm which use hard input. Therefore, SLS RLL are concatenated with RS codes which use geometric decoding algorithm. Simulation results show that that the BER performance of SLS RLL is better than ML RLL decoding.

Next, we proposed a bit level soft (BLS) output RLL decoding algorithm, which has more widely applications, it can be concatenated with powerful ECC, such as low-density parity-check (LDPC) codes, turbo codes, and polar codes. However, we first consider RS codes with iterative decoding, in order to compare with ML RLL, and SLS RLL which are also concatenated with RS codes. Simulation result show that proposed BLS RLL has better frame error rate (FER) performance, which is advantage for application to hybrid automatic repeat request (ARQ).

Next, modified Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm is considered to connect with BLS RLL decoding. Viterbi algorithm is generally used for convolutional codes decoding which is hard output decoding, however, the modified BCJR can make soft output. The simulation result show that modified BCJR has better performance once convolutional codes are considered.

Finally, dimming control (for various dimming values) algorithm is proposed. RLL codes are used to provide 50% dimming, other dimming value can be obtained by using compensation symbol (or puncturing). The state of art polar codes, which is the first codes proven to achieve the Shannon capacity, are considered. The reference considers previous result about LDPC codes and RS codes. Simulation results show that our proposed algorithm has best performance to compare with referenced ones

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background and Objectives

With the explosive demanding of bandwidth, the scarcity of spectrum leads visible light communication (VLC) to be attractive in people's view during recent years. VLC using visible light spectrum from 380 to 780 nm can provide both illumination lighting and communication functions together base on light emitting diodes (LEDs) technique. LEDs enhance energy efficiency under the fast nanosecond switching time. To invigorate the VLC industry, VLC standard is recently published by IEEE standard association [1].

An indoor VLC system based on white LED lights was first proposed in [2]. The influence of interference and reflection was discussed, and numerical analyses were presented. In [3], the indoor visible VLC which utilizes multiple white LED lighting equipment is adopted as system background. The proposed adaptive equalization system overcomes an intersymbol interference (ISI) problem by assigning the effectual interval to alleviate the influence of shadowing.

The VLC system which is designed to reduce illumination interference was proposed in [4]. To reduce interferences effects, optical filters is used in the reception system. Furthermore, signal modulation techniques are also useful to reduce the disruption that has not been eliminated by the filtering. A recursive equalizer algorithm in [5] was proposed to reduce the computation complexity of blind decision feedback equalizer algorithm, to copy with DC bias and impulsive noise problems in integrated systems of indoor VLC and power line communication (PLC). The integration approaches of VLC systems combined with PLC systems are emerging for home sensor networking.

It is known that Reed-Solomon (RS) codes are used in many applications due to good error-correction capability of RS codes. Application of RS codes to wireless ATM Network was proposed [6] to compensate the performance degradation. Bit error rate performance of the proposed system was discussed by maximizing link availability and minimizing the impact of atmospheric

conditions. In [7], a reconfigurable forward error correction (FEC) system based on RS codes for DVB is proposed. Efficient RS decoding based on remainder polynomial was proposed in [8]. The proposed method gives a characterization of the set of minimal solution to the WB key equation and has a low complexity. In [9], a novel shortening method of RS codes was proposed, and the low complexity of corresponding shorted BCH codes was also explained. Simulation results show 1.5dB coding grain at word error rate 10^{-5}.

In VLC standards, Reed-Solomon (RS) codes are also widely used as FEC codes because of robust ability to correct burst errors [10]. Run-length limited (RLL) codes are also important since they guarantee the DC balance with equal 1s and 0s, such as Manchester, 4B6B and 8B10B codes in VLC standard. For the hardware constraint for practical usage, on-off keying (OOK) as a simple modulation is considered because of just sending binary data by on/off pulses in VLC [11].

Recent years, many researchers have been developing the efficient FEC codes for VLC [12]-[14]. A novel FEC coding scheme was proposed for VLC base on the Reed-Muller (RM) codes and OOK, and it provided more coding gain than VLC scheme with the conventional FEC codes [12]. A bent function is considered in VLC based on RM codes and an OOK modulation, which increases performance with accurate dimming target value [13]. Furthermore, RS codes with RLL codes and FSK modulation is proposed in [14], and it presented the spectral analysis of RLL codes. Even though these research [12]-[14] have been done, the bit error rate (BER) performance of RLL codes and RS codes in VLC is rarely considered. Therefore, RLL codes which are used for guaranteeing DC balance and dimming 50% are considered, and BER performances are also shown.

In this thesis, we first propose an enhanced decoding based on maximum likelihood (ML) method of RLL codes with OOK modulation and RS codes in VLC system. In our model, the receiver have a new architecture, the RLL decoder makes better outputs based on ML rule which takes advantage of soft value of VLC channel. Simulation results show that our proposed method get signal-to-noise ratio (SNR) gain in bit error rate (BER) compared with one of RLL hard decoding. In order to improve performance, furthermore, we modified in the receiver part, the RLL decoder produces multiple superior candidates composed of symbols with higher probabilities. A selector then analyzes the

candidates based on a cyclic redundancy check (CRC) and chooses one of the candidates as an estimated message. A significant advantage of our proposed method is that it leads to performance enhancement without any change in the transmitter in VLC systems. Then, we proposed a symbol level soft (SLS) output RLL decoding algorithm since previous ML decoding belong to hard output RLL decoding (even with multiple candidates), the soft output of RLL decoding can provide more information for concatenated error correction codes (ECC), here consider RS codes. Moreover, conventional RS codes adopt Berlekamp-Massey (BM) algorithm which use hard input. Therefore, SLS RLL are concatenated with RS codes which use geometric decoding algorithm. Next, we proposed a bit level soft (BLS) output RLL decoding algorithm, which has more widely applications, it can be concatenated with powerful ECC, such as low-density parity-check (LDPC) codes, turbo codes, and polar codes. However, we first consider RS codes with iterative decoding, in order to compare with ML RLL, and SLS RLL which are also concatenated with RS codes. Next, modified Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm is considered to connect with BLS RLL decoding. Viterbi algorithm is generally used for convolutional codes decoding which is hard output decoding, however, the modified BCJR can make soft output. The simulation result show that modified BCJR has better performance once convolutional codes are considered. Finally, dimming control algorithm is proposed. RLL codes are used to provide 50% dimming, other dimming value can be obtained by using compensation symbol (or puncturing). The state of art polar codes, which is the first codes proven to achieve the Shannon capacity, are considered.

The rest of this thesis is organized as follows: we first review some background about RLL codes and RS codes in Chapter 1, then RLL decoding is studied in Chapter 2 - 5. In Chapter 2, an RLL decoding based on ML rule is presented. In Chapter 3, a modified RLL decoding which output multiple candidates is presented. SLS RLL and BLS RLL are presented in Chapter 4 and Chapter 5, respectively. RLL concatenated with convolutional codes and polar codes (consider other dimming values) are shown in Chapter 6 and Chapter 7, respectively. Finally, we summarize the main contribution of this thesis in Chapter 8. Notably, each chapter use its own variable notation.

1.2 Introduction of RLL codes and necessity

For VLC, the flickering is one of main issue need to consider once light is used to transmit signal. The RLL codes are general used to avoid flickering problem since same number of zeros and ones in each RLL symbol. However, even RLL codes are defined in VLC standard, they are non-linear codes and decoding algorithm dose not given by VLC standard. Therefore, efficient decoding algorithm of RLL codes is our research focus. Furthermore, RLL codes can provide 50% dimming. Dimming control is another important issue need to consider, since different environment need different level of illumination for aesthetic, comfort, and energy saving. Other dimming value can be construct by insert compensation symbols based on 50% dimming from RLL codes.

In the VLC standards [1], three RLL codes are considered, which are Manchester codes, 4B6B codes, and 8B10B codes. To explain the decoding of 4B6B codes and 8B10 codes, mapping relations between input and output of RLL codes need to be discussed.

The mapping rules of 4B6B codes are listed in Table 1 [4]. The left column '4-bit' and the right column '6-bit' denote the 4-bit input and 6bit output of RLL encoder, respectively. The weight of 4B6B codeword is 3 and maximum run of 0s or 1s is 4.

Table 1. Mapping's relation between a 4-bit symbol and a 6-bit symbol

4-bit	6-bit		4-bit	6-bit
0000	001110		1000	011001
0001	001101		1001	011010
0010	010011		1010	011100
0011	010110		1011	110001
0100	010101		1100	110010
0101	100011		1101	101001
0110	100110		1110	101010
0111	100101		1111	101100

8B10B codes are compose of 5B6B codes and 3B4B codes, where corresponding tables are Table 2 and Table 3, respectively. 'RD' in Table 2 and Table 3 denotes running disparity. Since the number of 0's and the number of 1's in 5B6B codewords and 3B4B codewords are different. The difference between the number of ones transmitted and the number of zeros transmitted is always limited to ±2, and at the end of each symbol, it is signed either +1 or −1, this difference is known as the running disparity. RD- codewords and RD+ codewords are used to keep number of 0s and 1s, equally to maintain DC balance.

Table 2. Mapping's relation between a 5-bit symbol and a 6-bit symbol

5-bit	6-bit RD -	6-bit RD+	5-bit	6-bit RD -	6-bit RD +
00000	100111	011000	10000	011011	100100
00001	011101	100010	10001	100011	
00010	101101	010010	10010	010011	
00011	110001		10011	110010	
00100	110101	001010	10100	001011	
00101	101001		10101	101010	
00110	011001		10110	011010	
00111	111000	000111	10111	111010	000101
01000	111001	000110	11000	110011	001100
01001	100101		11001	100110	
01010	010101		11010	010110	
01011	110100		11011	110110	001001
01100	001101		11100	001110	
01101	101100		11101	101110	010001
01110	011100		11110	011110	100001
01111	010111	101000	11111	101011	010100

Table 3. Mapping's relation between a 3-bit symbol and a 4-bit symbol

3-bit	4-bit RD -	4-bit RD +	3-bit	4-bit RD -	4-bit RD +
000	1011	0100	100	1101	0010
001	1001		101	1010	
010	0101		110	0110	
011	1100	0011	111	1110	0001

1.3 Brief review of Reed-Solomon codes

In general case, RS codes [10] are defined in Galois Field, GF(2^q), where q are a positive integer, respectively. Then, the parameters of (n, k) RS codes with length n and dimension k, are defined as $n=2^q - 1$, $\tau=(n-k)/2$, where τ denotes the maximum number of errors which the decoder can correct.

Since the RS codes are a class of cyclic codes, the message and codeword can be expressed as a message polynomial $u(X)$ and a codeword polynomial $c(X)$, respectively [3].

$$u(X) = u_0 + u_1 X + \ldots + u_{k-1} X^{k-1},$$

$$c(X) = c_0 + c_1 X + \ldots + c_{n-1} X^{n-1},$$

where $u_i \in (2^q)$ for i= 0, 1, …, k-1 and $c_i \in (2^q)$ for j= 0,1, …, n-1. For up to t-error correction, the generator polynomial is defined as

$$\begin{aligned} g(X) &= (X-\alpha)(X-\alpha^2)\ldots(X-\alpha^{2\tau}) \\ &= g_0 + g_1 X + \ldots + g_{2\tau} X^{2\tau} \end{aligned},$$

where α is a primitive element of GF(2^q).

To make systematic codewords, encoding can be defined as

$$c(X) = p(X) + X^{2\tau} u(X),$$

where $p(X)$ is the parity polynomial with degree $< 2\tau$. The $p(X)$ can be calculate as a remainder when $g(X)$ is divided into $X^{2\tau} u(X)$. The division algorithm of systematic encoding process of RS codes is shown in Fig. 1.

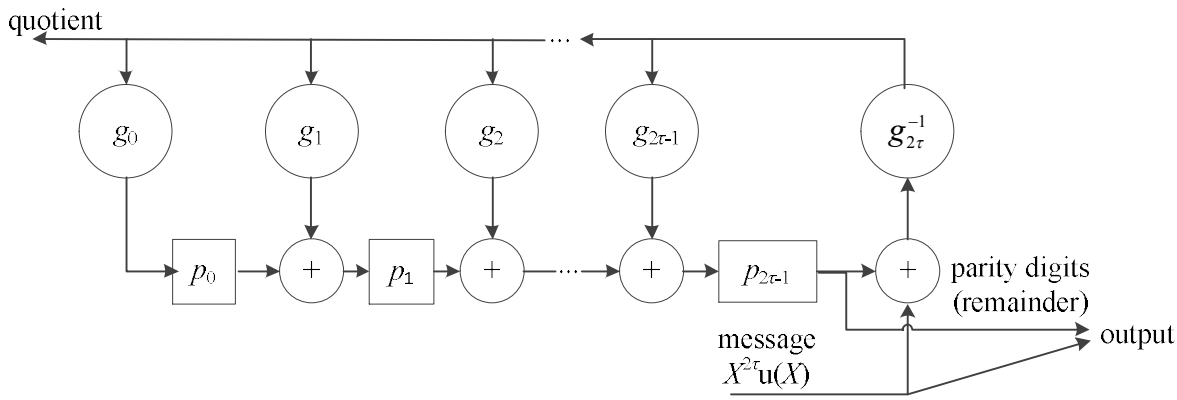

Fig 1. Division for systematic encoding of RS codes

Hard decision RS decoding wildly adopt Berlekamp-Massey (BM) algorithm [10], which is efficient to find error location polynomial based linear feedback shift register (LFSR), then error location and error value can be easily calculated.

CHAPTER 2
ML RLL DECODING SYSTEM DESIGN

2.1 VLC systems based on ML RLL decoding

The block diagram of the proposed scheme is shown in Fig. 2. In Fig. 2 (a), the transmitter of the proposed system is shown, and each components of the transmitter are also defined in the VLC standard [1]. The 'message' in the transmitter denotes binary m-bit message corresponding to input of the VLC system. Then (n, k) RS codes over GF(2^q) in [10] are considered, where kq is equal to m. The output of RS encoder is nq-bit codeword. Next, in the RLL encoder, each of q-bit data, $\boldsymbol{s}$, is mapped to an o-bit codeword, $\boldsymbol{c}$, in $\mathbf{C}_R$ according to the mapping rule [1]. The mapping rule is denoted as $\boldsymbol{c}= M(\boldsymbol{s})$ and $\boldsymbol{s}=M^{-1}(\boldsymbol{c})$, where M and M^{-1} denote encoding and decoding of RLL codes, respectively. The number of $\boldsymbol{s}$ and $\boldsymbol{c}$ are equal value, 2^q, in $\mathbf{C}_R$. Next, the LED emits OOK-modulated signal $x(t)$, which has the average optical power expressed as

$$P_t = \frac{1}{T}\int_0^T x(t)\,dt\ ,$$

where T denotes light signal duration.

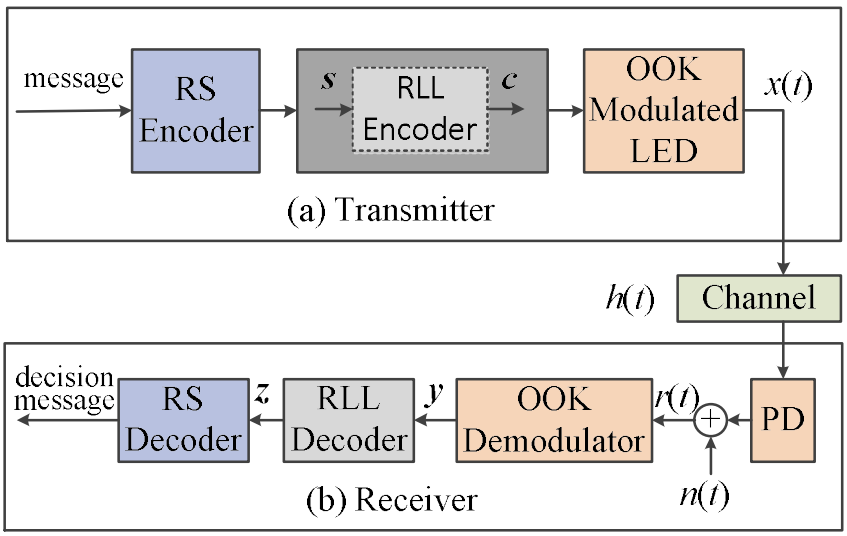

Fig. 2. Block diagrams of the transmitter and the receiver of the proposed VLC system

After passing through the VLC channel $h(t)$, the photodiode (PD) receives light in Fig. 1(b). Then, the received signal $r(t)$ is given as [12], [13]:

$$r(t) = R \cdot x(t) \otimes h(t) + n(t),$$

where "$\otimes$" denotes convolution; R is the PD conversion efficiency (A/W); and $n(t)$ is additive white Gaussian noise (AWGN) which contains the shot and thermal noise. Since an OOK modulation is used for data transmission, according to [12], [13] the received signal-to-noise ratio ($\mathrm{SNR_{rx}}$) is expressed as

$$\mathrm{SNR_{rx}} = \frac{\left(R \cdot P_{rSignal}\right)^2}{\sigma_{shot}^2 + \sigma_{thermal}^2 + \left(R \cdot P_{rISI}\right)^2},$$

where the desired signal power is

$$P_{rSignal} = \int_0^T x(t) \otimes h(t)\, dt,$$

and the received power affected by inter-symbol interference is

$$P_{rISI} = \int_T^\infty x(t) \otimes h(t)\, dt.$$

A shot noise variance is expressed

$$\sigma_{shot}^2 = 2\rho\gamma\left(P_{rSignal} + P_{rISI}\right)B + 2\rho I_{bg} I_2 B,$$

where ρ is the electronic charge, B is equivalent noise bandwidth, and I_{bg} and I_2 are background current and the noise bandwidth factor, respectively. In addition, the thermal noise variance is also expressed as

$$\sigma_{thermal}^2 = \frac{8\pi k T_k}{G}\eta A I_2 B^2 + \frac{16\pi^2 k T_k \Gamma}{g_m}\eta^2 A^2 I_3 B^3,$$

where the first two terms represent feedback-resistor noise and field-effect-transistor (FET) channel noise, respectively. K is Boltzmann's constant, T_k is absolute temperature, G is the open-loop voltage gain, η is the fixed capacitance of PD/unit area, Γ is the FET channel noise factor, g_m is the FET transconductance, and I_3 is the noise bandwidth factor.

In Fig. 2 (b), $r(t)$ is the input of OOK-demodulator, then OOK-demodulated signal $\boldsymbol{y}$ is the input of RLL decoder, our proposed method based on ML decoding is designed to make output $\boldsymbol{z}$ from the

signal $\boldsymbol{y}$ and the decoding rules are explained in next subsection. Conventional Berlekamp-Massey (BM) decoding algorithm in [10] is considered in this paper. After decoding of RS codes, estimated message is determined.

2.2 ML decoding of RLL codes

In the VLC standards [1], three RLL codes are considered, which are Manchester codes, 4B6B codes, and 8B10B codes. To denote mapping relation of 4B6B codes or 8B10B codes simultaneously, (q, o) is used instead of (4, 6) or (8, 10). The block diagram about RLL codes mapping between encoding and decoding is shown in Fig. 3.

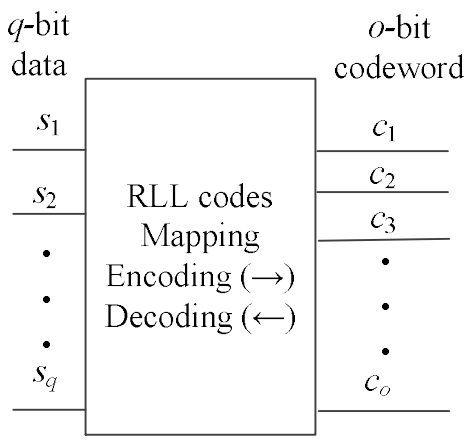

Fig. 3. The RLL code's mapping rule between q-bit data, $\boldsymbol{s}$, and o-bit codeword, $\boldsymbol{c}$.

The input of OOK demodulator can be presented as vector:

$$\boldsymbol{r}=(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3,\ldots, \boldsymbol{r}_i,\ldots, \boldsymbol{r}_n),$$

where $\boldsymbol{r}_i=(r_{i,1}, r_{i,2}, \ldots, r_{i,o})$ is the i-th symbol and n is the code length of RS codes.

Similarly, the input of RLL decoder is defined as:

$$\boldsymbol{y}=(\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3,\ldots, \boldsymbol{y}_i,\ldots, \boldsymbol{y}_n).$$

The output of RLL decoder is defined as:

$$\boldsymbol{z}=(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3,\ldots, \boldsymbol{z}_i,\ldots, \boldsymbol{z}_n).$$

The RLL decoder decodes $\boldsymbol{y}$ by processing (o-bit codeword) $\boldsymbol{y}_i$ and corresponding (q-bit data) $\boldsymbol{z}_i$ from i = 1 to n, respectively.

The proposed decoding rule of RLL codes is composed of 6 steps and is shown in Fig. 4. For n symbols, channel values are used for soft input in the RLL decoder. The detail explanation about ML decoding steps is described as follows.

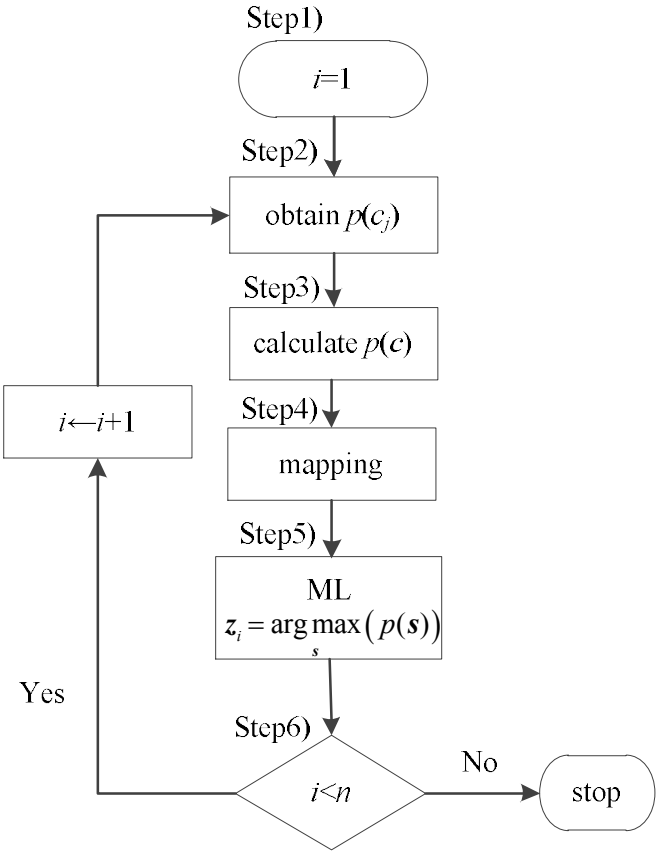

Fig. 4. ML decoding process of the proposed RLL decoding.

Proposed decoding algorithm of RLL decoder:

Step 1: Initialization: set i=1

Step 2: The probability of the j-th bit of the i -th symbol ($\boldsymbol{c} = \boldsymbol{y}_i$) for OOK demodulation is presented as

$$p\left(c_j = 1\right) = \frac{1}{1 + e^{-2r_{i,j}+1 / 2\sigma^2}},$$

$$p\left(c_j = 0\right) = \frac{e^{-2r_{i,j}+1 / 2\sigma^2}}{1 + e^{-2r_{i,j}+1 / 2\sigma^2}}.$$

Since log-likelyhood ratio (LLR) is calculated as:

$$\Lambda(c_j) = \log\frac{p(c_j=0)}{p(c_j=1)} = \frac{\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(r_{i,j}-0)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(r_{i,j}-1)^2}{2\sigma^2}}} = \frac{-2r_{i,j}+1}{2\sigma^2}.$$

Step 3: The o-bit codeword probability is defined as

$$p(\boldsymbol{c}) = \prod_{j=1}^{o} p(c_j).$$

Since there are 2^q values per one symbol, 2^q values about probability corresponding to o-bit codewords is calculated in this step.

Step 4: Mapping: According to Fig. 3, corresponded q-bit data have same probabilities to o-bit codewords, then

$$p(\boldsymbol{s}) = p(\boldsymbol{c}).$$

Step 5: Maximum likelihood (ML): According to ML algorithm, estimated output z_i is calculated as

$$z_i = \arg\max_{\boldsymbol{s}\in \mathbf{C}_R}\left(p(\boldsymbol{s})\right).$$

Step 6: If $i<n$, set $i \leftarrow i+1$, then go to Step 2 for another symbol. Else ($i=n$), stop.

Finally, $z=(z_1, z_2, \ldots, z_n)$ is output of RLL decoder, and it can be decoded by RS decoder by BM algorithm.

In this RLL decoding base on ML rule model, we can expect better performance compare with conventional hard RLL decoding. However, in Step 5, we just output symbol with highest probability, we may further enhance the performance by outputting more candidates. Therefore, the modified RLL decoding with multiple candidates is proposed.

2.3 Simulation results

Six cases are considered to check the performance. The first three cases are performances of 4B6B RLL codes with RS (15, 3) codes, RS (15, 7) codes, and RS (15, 11) codes, and the last three cases are about 8B10B RLL codes with RS (64, 32) codes, RS (64, 40) codes, and RS (64, 48) codes.

The reason why RS (15, 3) codes, RS(15, 7) codes, and RS(15,11) codes are chosen for 4B6B RLL codes is that in VLC standard [1], the lowest code rate and the highest code rate of RS codes corresponding to 4B6B RLL codes are RS(15, 3) codes and RS(15, 11) codes, respectively. Moreover, since the code rate of RS (15, 7) codes is in the middle of code rates of those RS codes, RS (15, 7) codes are considered. Similarly, RS (64, 32) codes, RS (64, 40) codes, and RS (64, 48) codes for 8B10B codes are chosen.

BER performances of the proposed RLL decoding and conventional RLL decoding for 4B6B codes and RS codes are shown in Fig. 5, Fig. 6, and Fig. 7, respectively. It is assumed that the output of the conventional RLL decoder is determined, where codeword of the output has the smallest distance from the corresponding output of the demodulator. E_b/N_0 on x-axis in each figure denotes SNR, where E_b is bit energy and N_0 is noise energy. The definition of this SNR is same with one in [12] and [13]. 'RLL referred' and 'RLL proposed' in each figure is corresponding to conventional decoding and proposed decoding, respectively. In Fig.5, Fig. 6, and Fig. 7, SNR is started from 4dB since the different between proposed decoding and conventional decoding is not obvious when the SNR is smaller than 4dB.

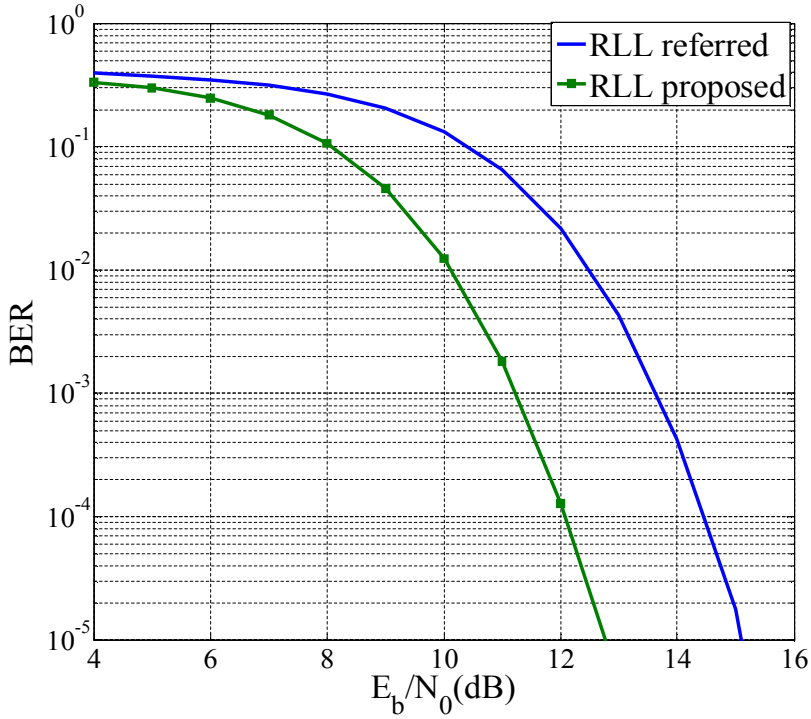

Fig. 5. BER performances of the RS (15, 3) codes over GF(2^4) with 4B6B RLL codes.

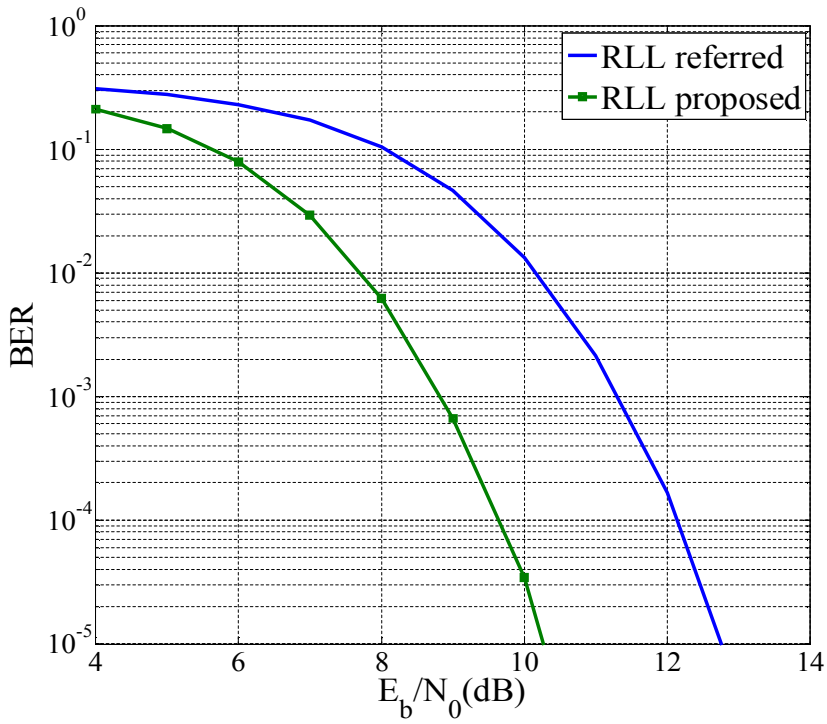

Fig. 6. BER performances of the RS (15, 7) codes over GF(2^4) with 4B6B RLL codes.

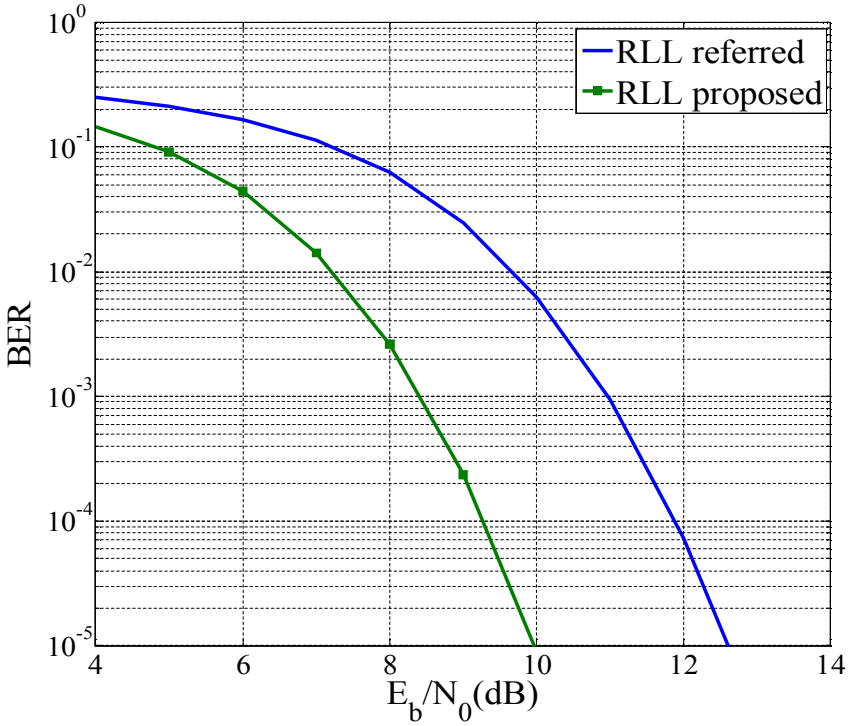

Fig. 7. BER performances of the RS (15, 11) codes over GF(2^4) with 4B6B RLL codes.

It is shown that BER performances of the proposed scheme are better than ones of the RLL with hard decoding in Fig. 5, Fig.6 and Fig. 7. Furthermore, BER performance in Fig. 5 is better than ones in Fig.6 and Fig. 7 since low-rate RS codes have better error-correcting capability.

Following Fig. 8, Fig. 9, and Fig. 10 show BER performances of 8B10B RLL codes with low-rate RS (64, 32) codes, middle rate RS (64, 40) codes, and high rate RS (64, 48) codes, respectively. In those figures for 8B10B codes, the performance of the proposed scheme is better than one of conventional RLL with hard decoding, but performance gain of 8B10B codes is not better than one of 4B6B codes.

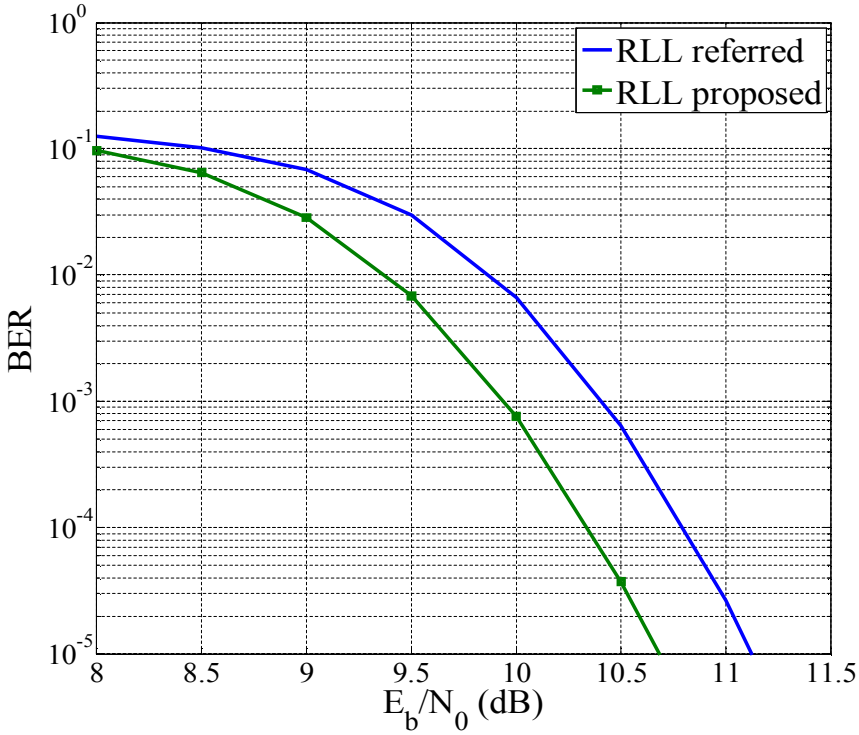

Fig. 8. BER performances of the RS (64, 32) codes over GF(2^8) with 8B10B RLL codes.

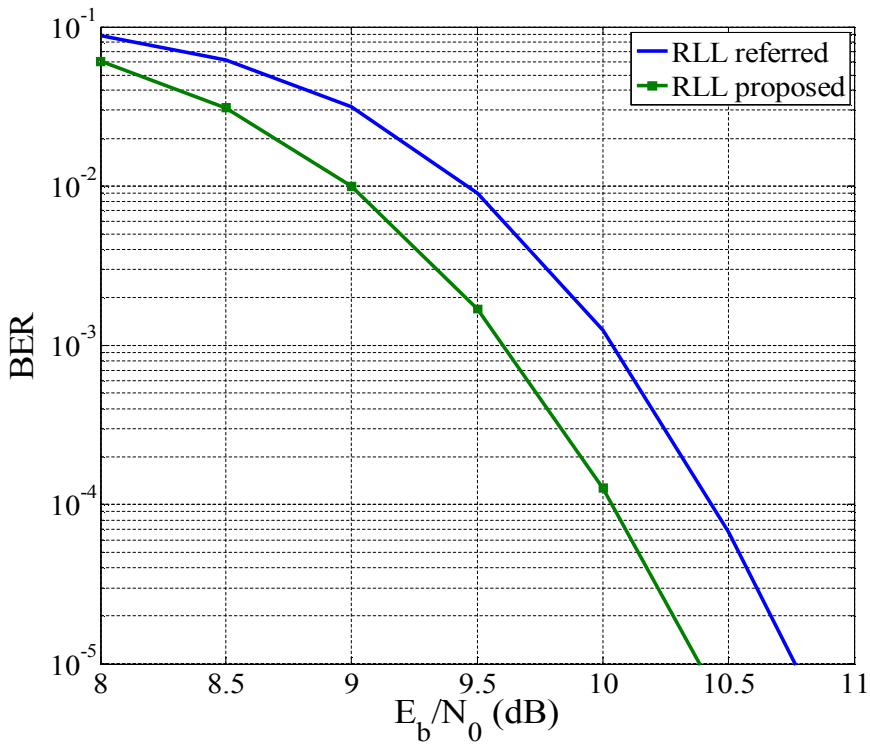

Fig. 9. BER performances of the RS (64, 40) codes over GF(2^8) with 8B10B RLL codes.

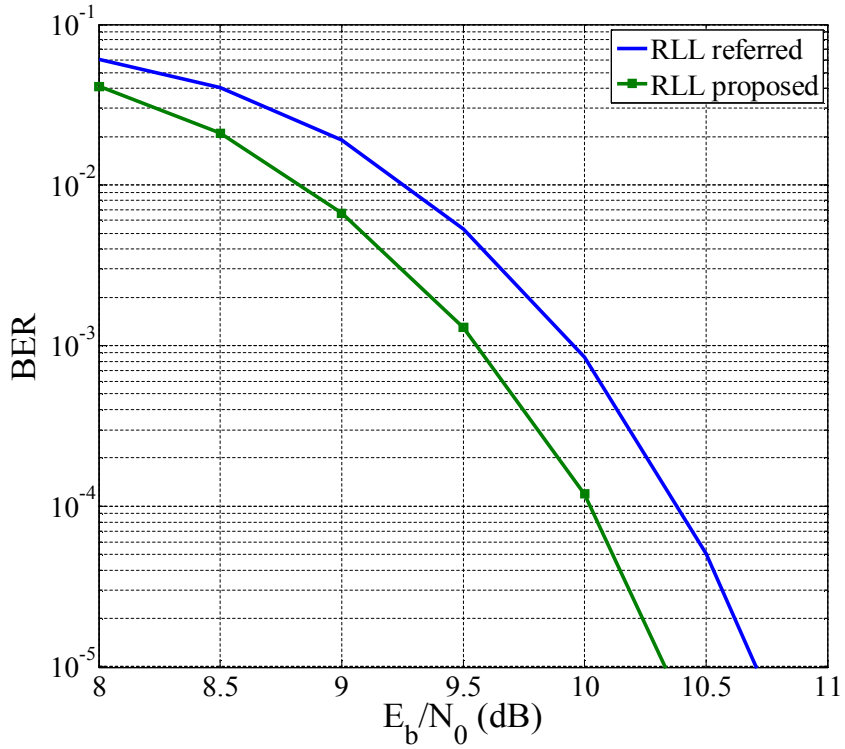

Fig. 10. BER performances of the RS (64, 48) codes over GF(2^8) with 8B10B RLL codes.

To investigate exact performance enhancement, the required SNR and gain at BER=10^{-5} are listed in Table 4.

Table 4. The E_b/N_0 gians between the proposed RLL based on ML rule and referred RLL codes and codes at BER=10^{-5}

		RLL referred	RLL proposed	Gain
4B6B RLL	RS (15, 3)	15.10	12.77	2.33
	RS (15, 7)	12.77	10.28	2.49
	RS (15, 11)	12.61	9.97	2.64
8B10B RLL	RS (64, 32)	11.12	10.68	0.44
	RS (64, 40)	10.76	10.38	0.38
	RS (64, 48)	10.71	10.33	0.38

For 4B6B RLL codes, the proposed algorithm has more than 2dB SNR gain than referred conventional decoding but there is about 0.4dB SNR gain for 8B10B RLL codes. The reason why SNR gain of 4B6B RLL codes is better is that since the code rate of 4B6B is lower than one of 8B10B RLL codes, decoding capability made from 4B6B codes is better than one of 8B10B RLL codes.

CHAPTER 3
MODIFIED MULTIPLE OUTPUT RLL DECODING

3.1 VLC systems based on multiple output RLL decoding

The proposed system considers the RS codes and the RLL codes, $\mathbf{C}_R$; a block diagram of the proposed system is shown in Fig. 11. The transmitter in Fig. 11 (a) is composed of an RS encoder, an RLL encoder, and OOK modulated LEDs; it is same with previous model except considering cyclic redundancy check (CRC). The binary m-bit message is composed of (m-16)-bit data and the 16-bit CRC [1]. The standard generator polynomial of CRC-16 is:

$$G_{16}(X) = 1 + X^5 + X^{12} + X^{16}.$$

(n, k) RS codes over GF(2^q) are considered, where kq is equal to m. The output of the RS encoder, $\boldsymbol{s}$, is composed of n symbols. Next, in the RLL encoder, each q-bit symbol $\boldsymbol{s}$ is mapped to an o-bit codeword, $\boldsymbol{c}$, in $\mathbf{C}_R$ according to the mapping rule [1]. The total number of codewords in $\mathbf{C}_R$ is 2^q. Finally, the LED emits OOK-modulated signal.

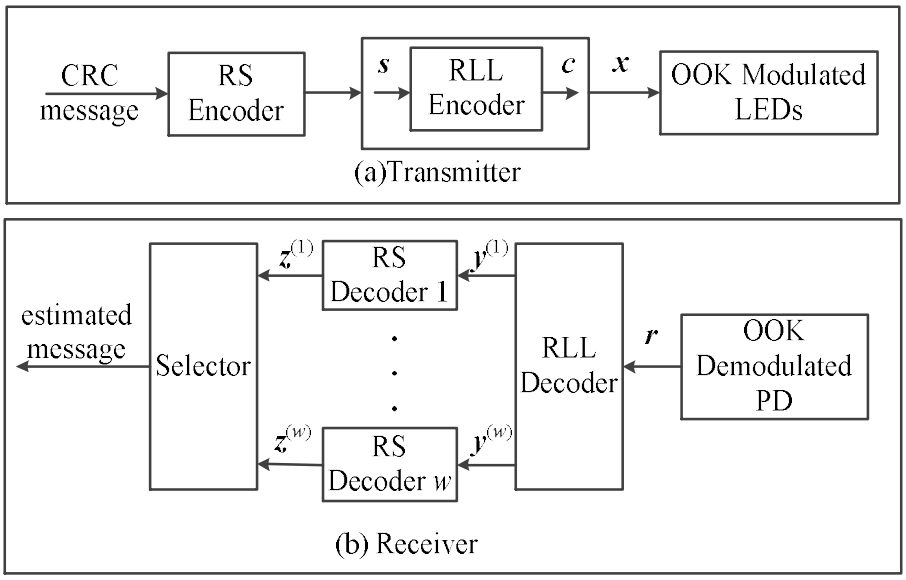

Fig. 11. Block diagrams of the transmitter and the receiver of the proposed VLC system

The receiver in Fig. 11 (b) is composed of an OOK demodulated photodiode (PD), an RLL decoder, w RS decoders and a selector. The received OOK-demodulated signal, $\boldsymbol{r}$ (= $\boldsymbol{x}$+$\boldsymbol{n}$), is the input of the RLL decoder, where $\boldsymbol{n}$ is AWGN with variance σ^2 [12], [13]. The RLL decoder then outputs w candidates that are defined as $\boldsymbol{y}^{(1)}$, $\boldsymbol{y}^{(2)}$, $\cdots$, $\boldsymbol{y}^{(w)}$. The Berlekamp-Massey (BM) algorithm [10] is used as the decoding algorithm of the RS decoders. Finally, according to the selection criterion based on the CRC, the estimated message is chosen from the outputs of the RS decoders.

Our paper focuses on RLL decoding based on multiple output candidates and selection based on a CRC, and a detailed description of our algorithm is given.

3.2 Criterion for outputting multiple candidates of RLL decoder

To denote the RLL codes such as 4B6B and 8B10B [1], (q, o) is used instead of (4, 6) or (8, 10) for general usage. Then, the output of the RLL encoder is defined as:

$$\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,o}),\ i = 1, 2, \cdots, n.$$

The input of the RLL decoder is defined as:

$$\boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_i, \cdots, \boldsymbol{r}_n),$$

where $\boldsymbol{r}_i = (r_{i,1}, r_{i,2}, \cdots, r_{i,o})$. Similarly, the output of the RLL decoder is defined as:

$$\boldsymbol{y}^{(u)} = (\boldsymbol{y}_1^{(u)}, \boldsymbol{y}_2^{(u)}, \cdots, \boldsymbol{y}_i^{(u)}, \cdots, \boldsymbol{y}_n^{(u)}),$$

where $\boldsymbol{y}_i^{(u)} = (\boldsymbol{y}_{i,1}^{(u)}, \boldsymbol{y}_{i,2}^{(u)}, \cdots, \boldsymbol{y}_{i,l}^{(u)})$, $u = 1, 2, \cdots, w$. The number of candidates, w, is considered a power of 2, 2^h. Let $\boldsymbol{d} = (d_1, d_2, \cdots, d_n)$ be the vector for the difference between the highest probability and the second highest probability of the i-th symbol for i=1, 2, ⋯, n. In order to create the larger probability candidates, $\boldsymbol{\lambda}_{i,1}$ and $\boldsymbol{\lambda}_{i,2}$ are defined to store the q-bit symbol, $\boldsymbol{s}$, they have the highest probability and the second highest probability, respectively. The following steps (from Step 1 to Step 6) present the detailed decoding process. Then different with ML RLL decoding is major in Step 5 and Step 7.

Step 1: Initialization: set i=1.

Step 2: The LLR of the j-th bit of the i -th symbol when the codeword in $\mathbf{C}_R$ is assumed to be $\boldsymbol{c}$ (=$\boldsymbol{x}_i$) can be defined as

$$\Lambda(c_j) = \log\frac{p(c_j = 0)}{p(c_j = 1)} = \frac{-2r_{i,j} + 1}{2\sigma^2},$$

where j = 1, 2, ⋯, o. The j-th bit probability of the i-th symbol, $\boldsymbol{x}_i$, can then be calculated as

$$p(c_j) = \frac{c_j + (1 - c_j)e^{\Lambda(c_j)}}{1 + e^{\Lambda(c_j)}}.$$

Step 3: The o-bit codeword probability is defined as:

$$p(\boldsymbol{c}) = \prod_{j=1}^{o} p(c_j).$$

Step 4: Mapping, when $\boldsymbol{c}$ is a codeword in $\mathbf{C}_R$ and $\boldsymbol{s}=M^{-1}(\boldsymbol{c})$, therefore

$$p(\boldsymbol{s}) = p(\boldsymbol{c}) .$$

Step 5: The highest probability $p1$ and corresponded symbol $\boldsymbol{\lambda}_{i,1}$ are calculated as $p1 = \max p(\boldsymbol{s})$ and $\boldsymbol{\lambda}_{i,1} = \arg\max_{\boldsymbol{s}}(p(\boldsymbol{s}))$, respectively. Similarly, the second highest probability, $p2$, and $\boldsymbol{\lambda}_{i,2}$ are calculated as $p2 = \max\ (p(\boldsymbol{s})\backslash\ p1)$ and $\boldsymbol{\lambda}_{i,2} = \arg\max_{\boldsymbol{s}\backslash\lambda_{i,1}}(p(\boldsymbol{s}))$, respectively. The difference d_i between the two probabilities is obtained using $d_i=p1-p2$.

Step 6: If $i<n$, set $i\leftarrow i+1$, then go to Step 2 for another symbol, else go to step 7.

Step 7: In this step, w candidates, $\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \cdots, \boldsymbol{y}^{(w)}$, are created from $\boldsymbol{\lambda}_{i,1}$ and $\boldsymbol{\lambda}_{i,2}$ according to the index of $\boldsymbol{d}$. Since w is 2^h, the w candidates are determined from the h symbols with the smallest difference. There are $(h + 1)$ stages from the index of $\boldsymbol{d}$.

Stage 0: At the initial stage (Stage 0), the first candidate, $\boldsymbol{y}^{(1)}$, is composed of the symbol with the highest probability.

$$\boldsymbol{y}_i^{(1)} = \boldsymbol{\lambda}_{i,1} \text{ for } i = 1, 2, \cdots, n.$$

Stage 1: At Stage 1, the index i_1 is determined using $i_1 = \arg\min_i(\boldsymbol{d}_i)$. Next, the second candidate $\boldsymbol{y}^{(2)}$ is copied with the $\boldsymbol{y}^{(1)}$. Then, $\boldsymbol{y}_{i_1}^{(2)}$ is changed using $\boldsymbol{y}_{i_1}^{(2)} = \boldsymbol{\lambda}_{i_1,2}$. The second candidate is determined by changing the i_1-th symbol in the previous candidate.

Stage 2: At Stage 2, the index i_2 is determined using $i_2 = \arg\min_{i\neq i_1}(\boldsymbol{d}_i)$. Next, the third and fourth candidates are copied with the previous candidates using $(\boldsymbol{y}^{(3)}, \boldsymbol{y}^{(4)}) = (\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$; the i_2-th symbols of the candidates are then changed using $\boldsymbol{y}_{i_2}^{(3)} = \boldsymbol{\lambda}_{i_2,2}$ and $\boldsymbol{y}_{i_2}^{(4)} = \boldsymbol{\lambda}_{i_2,2}$.

In each stage one new index is determined from the vector, $\boldsymbol{d}$, and new candidates are formed from the previous candidates by changing the symbol according to the new index. This process continues from Stage 1 to Stage h.

Stage h : At Stage h, the index i_h is determined using $i_h = \arg\min_{i\neq i_1, i_2, \cdots i_{h-1}}(\boldsymbol{d}_i)$. Next, the new 2^{h-1}

candidates are copied with the previous 2^{h-1} candidates using $(\boldsymbol{y}^{(2^{h-1}+1)}, \boldsymbol{y}^{(2^{h-1}+2)}, \cdots, \boldsymbol{y}^{(2^h)}) = (\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \cdots, \boldsymbol{y}^{(2^{h-1})})$, the i_h-th symbols of the new candidates are then changed using $\boldsymbol{y}_{i_h}^{(2^{h-1}+1)} = \boldsymbol{\lambda}_{i_h,2}$, $\boldsymbol{y}_{i_h}^{(2^{h-1}+2)} = \boldsymbol{\lambda}_{i_h,2}$, $\cdots$, $\boldsymbol{y}_{i_h}^{(2^h)} = \boldsymbol{\lambda}_{i_h,2}$.

Finally, the RLL decoder creates $\boldsymbol{y}^{(u)}$, where u=1, 2, $\cdots$, w.

These w candidates are the inputs of the w RS decoders. From the BM algorithm, RS decoders produce the output, $\boldsymbol{z}^{(u)}$. The selection method among the $\boldsymbol{z}^{(u)}$ of the RS decoder, u, for u = 1, 2, $\cdots$, w is explained in the next subsection.

3.3 Criterion of selection based on CRC

Among the c candidates, the first CRC of $\boldsymbol{z}^{(1)}$ is calculated and analyzed to determine whether the CRC succeeds or not. If the CRC succeeds, then selector produces the estimated output: $\boldsymbol{z}^{(1)}$. If the CRC fails, the selector moves to the second candidate. This process continues from u=1 to w. The following steps (from Step 1 to Step 4) provide a detailed explanation of the selector.

Step 1: Set u = 1.

Step 2: The remainder, $\boldsymbol{\gamma}_u$, is calculated by dividing the (m-16)-bit data of $\boldsymbol{z}^{(u)}$ by the CRC-16 generator.

Step 3: If $\boldsymbol{\gamma}_u$ is equal to the rest of the 16-bits of $\boldsymbol{z}^{(u)}$, $\boldsymbol{z}^{(u)}$ satisfies the CRC; $\boldsymbol{z}^{(u)}$ is selected as the estimated message, and this process stops. If not, go to Step4).

Step 4: If $u<w$, set $u \leftarrow u+1$, then go to Step2) to analyze another candidate. If not ($u = w$), the estimated message is $\boldsymbol{z}^{(1)}$.

When the CRCs of multiple candidates succeed, the candidate that has the first successful CRC becomes the estimated message without checking the CRCs of the other candidates. If there is no successful CRC among the candidates, the selector forms the estimated message as $\boldsymbol{z}^{(1)}$, since the $\boldsymbol{y}^{(1)}$ has the highest probability among all candidates.

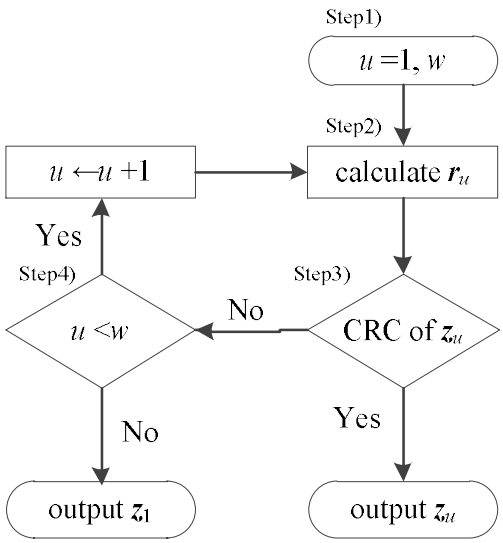

Fig. 12. Select optimal one from w candidates based on CRC.

3.4 Simulation results

To verify the proposed system, two cases are considered: 4B6B RLL codes with RS (15, 7) codes over GF(2^4), and 8B10B RLL codes with RS (64, 32) codes over GF(2^8) in the VLC standard [4].

The performances of the proposed RLL decoding which outputs w (= 1, 2, 4, 8, 16) candidates are compared with that of referred RLL hard decoding in Fig. 13 and Fig. 14. E_b/N_0 is used for the signal-to-noise ratio (SNR), where E_b is the bit energy and N_0 is the noise energy. This SNR is the same as that in [12] and [13]. The BER results are shown in Fig. 13 and Fig. 14, and the SNR gain is listed in Table 5, percentage gain is defined as:

$$\frac{\text{gain of } w \text{ candidates (dB)}}{\text{gain of one candidates (dB)}} \times 100\%$$

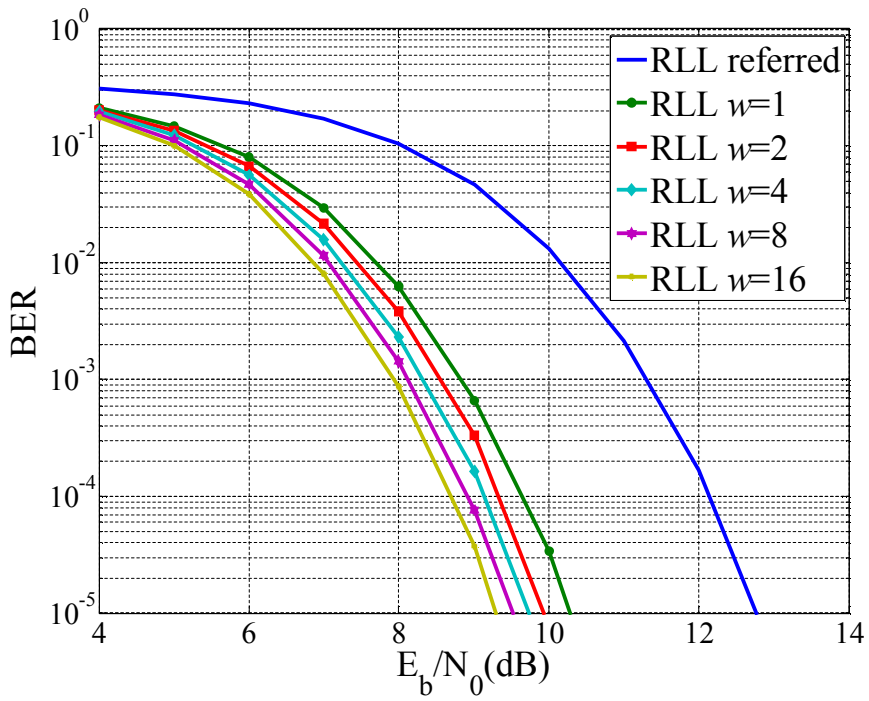

Fig. 13. BER performances of the RS (15, 7) codes over GF(2^4) with 4B6B RLL codes

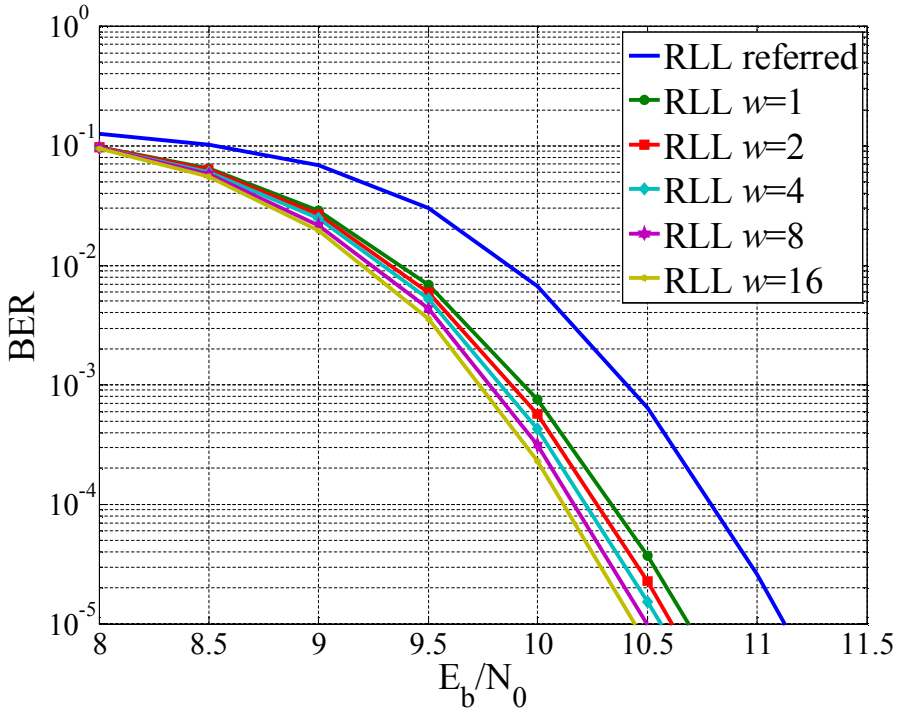

Fig. 14. BER performances of the RS (64, 32) codes over GF(2^8) with 8B10B RLL codes

As shown in Fig. 13 and Fig. 14, the BER performances of the proposed scheme are much better than that of the RLL hard decoding. Furthermore, the performace gain increases when the number of candidates increases. The performance enhancement is greater between referred RLL decoding and proposed RLL decoding with w = 1; however, the performance enhancement is smaller between different value of c for proposed RLL decoding, this means that a better link performance with less complexity can be expected without an increase in the number of RS decoders by using only our proposed LLR decoder.

Table 5. The E_b/N_0 gain between the proposed scheme with w candidates and the referred RLL in Fig. 13 and Fig. 14 with BER=10^{-5}

		w=1	w=2	w=4	w=8	w=16
RS (15, 7)	dB	2.49	2.82	3.03	3.24	3.48
	%	100	113.2	121.6	130.1	139.7
RS (64, 32)	dB	0.44	0.52	0.57	0.63	0.68
	%	100	118.1	129.5	143.1	154.5

In Table 5, the performance gains between the RLL decoding with different value of w and the referred RLL decoding with BER=10^{-5} are shown. The performance gain for 4B6B is at least 2dB better than that of 8B10B since the code rate of the 4B6B codes is lower that of 8B10B. If other RLL codes are designed for not only DC-balance, but also for better link performance, then the performance gain will increase further.

CHAPTER 4
SYMBOL LEVEL SOFT RLL DECODING

4.1 Introduction

With widely used optical sources, such as LEDs and laser diodes, VLC techniques recently have garnered increasing attention. In VLC, modified lighting sources can be used to transmit information using LEDs. If we consider the hardware constraints for practical usage, OOK, the transmission of binary data by on/off pulses is widely adopted in VLC due to its simplicity. Run-length-limited (RLL) codes take in random data symbols at the input and guarantee a DC balance with equal 1s and 0s at the output in order to allow every symbol to adjust dimming in VLC. Three RLL codes, Manchester, 4B6B, and 8B10B, are adopted in the VLC standard [1].

Since the emergence of VLC, a lot of work has been done to produce efficient transmission in VLC. In [12], a novel forward error correction (FEC) coding method based on modified Reed-Muller (RM) codes was proposed for providing accurate dimming control in OOK-modulated VLC. Moreover, a modified RM coding scheme made from the bent function for dimmable VLC was proposed in [13], suggesting a new method to simultaneously contain minimal compensation symbols in supporting multiple dimming target values and to improve coding gain. In power line communication with frequency shift keying, the efficient combination of RLL sequences and Reed-Solomon (RS) codes was proposed [14]. Recently, a novel soft-input hard-output RLL decoder was proposed, and it showed enhancement of the bit error rate (BER) performance in VLC [16]. However, the drawbacks of the RLL decoder [16] was that the decoder was designed to produce a hard-decision output for the RS decoder based on a Berlekamp-Massey (BM) algorithm, which sacrificed some more valuable soft information; and complexity improved much more with the number of candidates increased since more RS decoders were needed. Therefore, research relating to soft-output RLL decoder to enhance the VLC performance is needed.

In this chapter, we propose a new decoding algorithm based on the soft-input soft-output (SISO)

RLL decoder in VLC with an OOK modulation and RS codes. The proposed SISO RLL decoder produces reliable soft output, which can be used as a soft input of the RS decoder to enhance the throughput in the VLC system. Simulation results show that our proposed algorithm obtains better BER and frame error rate (FER) performances than those of conventional RLL decoding and referred RLL decoding [16].

4.2 Proposed VLC systems

A block diagram of the proposed system is shown in Fig. 15. The transmitter in Fig. 15 (a) is composed of an RS encoder, an RLL encoder, and OOK-modulated LEDs; it is similar to the transmitter in the VLC standard [1].

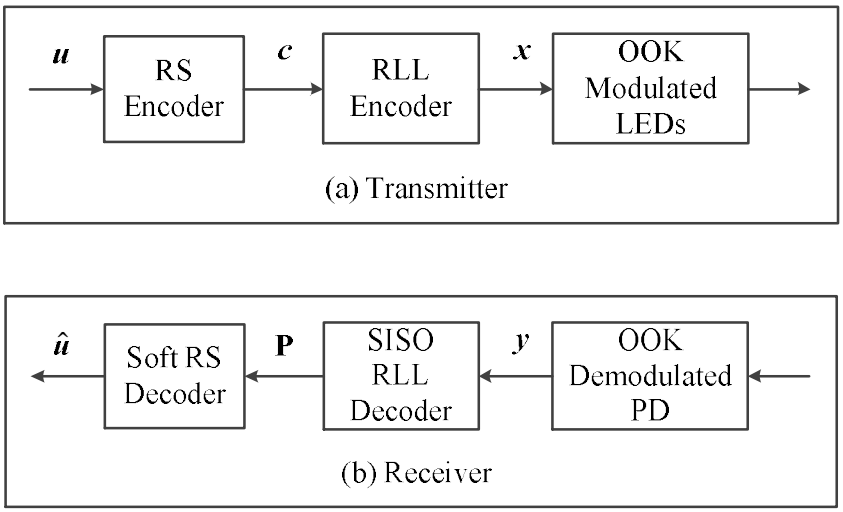

Fig. 15. Block diagrams of the transmitter and receiver in the proposed VLC system.

Original RS (n, k) codes [9] over GF(2^q) are considered, where $n \leq 2^q$ and GF(2^q) is $\{\beta_1, \beta_2, \ldots, \beta_{2^q}\}$. The k-symbol message $\boldsymbol{u}=(u_1, u_2,\ldots, u_k)$, where u_i is in GF(2^q), can be mapped to a corresponding polynomial $u(x) = u_1+u_2x+\cdots+u_kx^k$. The codeword of the RS encoder, $\boldsymbol{c} = (c_1, c_2,\ldots, c_n)$, is defined as $c_i = u(\alpha_i)$ for $i = 1, 2,\ldots, n$, where $\alpha_1, \alpha_2,\ldots, \alpha_n$, are n distinct elements in GF(2^q). A frame in this paper is considered as a codeword in the physical layer. To represent one q-bit vector, $\boldsymbol{t} = (t_1, t_2,\ldots, t_q)$, from a Galois field element for RLL encoding, the transfer function V is considered to be $V(\alpha) = \boldsymbol{t}$. Since the transfer function is a bijection, if $\alpha \neq \beta$ for α and β in GF(2^q), then $V(\alpha) \neq V(\beta)$. Therefore, any element β in GF(2^q) can be mapped to one of 2^q distinct binary vectors $\boldsymbol{t}$. Next, in an RLL encoder, the q-bit vector $V(c_i)$ corresponding to the symbol c_i is encoded to an s-bit RLL codeword $\boldsymbol{x}_i = (x_{i,1}, x_{i,2},\ldots, x_{i,s})$ according to the mapping rule E [1] as $E(V(c_i)) = \boldsymbol{x}_i$ for $i = 1, 2,\ldots, n$. Then, the concatenated output of RLL codes $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_n)$ is OOK-modulated and emitted to the channel.

The receiver in Fig. 15 (b) is composed of an OOK demodulated photodiode (PD), a SISO RLL decoder, and a soft RS decoder. The received OOK-demodulated signal, $\boldsymbol{y}$ (= $\boldsymbol{x}$+$\boldsymbol{n}$), is the input of the RLL decoder, where $\boldsymbol{n}$ is AWGN with variance σ^2. The RS decoder based on the Koetter-Vardy (KV) algorithm [18], which is an extension of the Guruswami-Sudan (GS) algorithm [19], is considered for soft decoding. GS is a list decoding algorithm based on polynomial interpolation for decoding beyond the guaranteed error-correction distance of RS. Although in principle there are more than one codeword within such an expanded distance, in fact, with high probability, only one will occur. KV extended it to handle soft decisions [23].

This chapter focuses on SISO RLL decoding combined with soft RS decoding, and a detailed description of our algorithm is given.

4.3 Soft-input soft-output RLL decoder

Since 4B6B and 8B10B RLL codes in the VLC standards [1] are provided for combination with RS codes, they are considered in this paper. Here (q, s) represent (4, 6) or (8, 10). The (q, s) RLL codes have an one-to-one mapping rule E between q-bit vectors and s-bit codewords, and the average weight of each s-bit codeword is $s/2$ in order to maintain a DC balance.

The input of the RLL decoder is defined as $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n)$, where $\boldsymbol{y}_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,s})$ for $i = 1, 2, \ldots, n$. The RLL decoder decodes $\boldsymbol{y}$ by processing (s-bit codeword) $\boldsymbol{y}_i$ and outputs a reliable matrix, which can be used in soft RS decoding [10]. The reliable matrix, **P**, is defined as

$$\mathbf{P} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{2^q,1} & p_{2^q,2} & \cdots & p_{2^q,n} \end{bmatrix},$$

where $p_{z,i}$ is the posterior probability that the transmitted symbol c_i is β_z for $z = 1, 2, \ldots, 2^q$ when the $\boldsymbol{y}_i$ is received. It is assumed that, if the transmitted variable c_i is uniformly distributed with equal probability, then $p_{z,i}$ can be calculated using Bayes' theorem,

$$p_{z,i} = p\left(c_i = \beta_z \,|\, \boldsymbol{y}_i\right) = \frac{p\left(\boldsymbol{y}_i \,|\, c_i = \beta_z\right) p\left(c_i = \beta_z\right)}{p\left(\boldsymbol{y}_i\right)}.$$

Therefore, $p_{z,i} = \Delta p(\boldsymbol{y}_i \mid c_i = \beta_z)$, where Δ is a constant value since $p(c_i = \beta_z)$ is $1/2^q$.

The proposed decoding rule of RLL codes is composed of five steps. The received signal $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n)$ is used for soft input in the RLL decoder, the i-th column of matrix **P** is produced for $i = 1, 2, \ldots, n$. A detailed explanation about SISO RLL decoding steps is provided below.

Step 1: Initialization: set i=1

Step 2: The s-bit codeword probability is defined as

$$p\left(\boldsymbol{y}_i \mid \boldsymbol{x}_i\right) = \prod_{j=1}^{s} p\left(y_{i,j} \mid x_{i,j}\right),$$

where $p\left(y_{i,j} \mid x_{i,j} = \theta\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_{i,j}-\theta\right)^2}{2\sigma^2}}$, for $\theta = 0, 1$.

Step 3: According to the mapping rule [1] that $E(V(c_i)) = \boldsymbol{x}_i$, the corresponding q-bit vectors have the same probabilities with o-bit codewords, then

$$p(V(c_i) = V(\beta_z)) = p(\boldsymbol{x}_i).$$

Step 4: Calculate the i–th column in the reliable matrix

$$\begin{aligned} p_{z,i} &= \Delta p\left(\boldsymbol{y}_i \middle| c_i = \beta_z\right) = \Delta p\left(\boldsymbol{y}_i \middle| V\left(c_i\right) = V\left(\beta_z\right)\right) \\ &= \Delta p\left(\boldsymbol{y}_i \mid \boldsymbol{x}_i\right) \end{aligned},$$

where $V(\beta_z)$, $z = 1, 2, \ldots, 2^q$ correspond to all indicated vectors from elements of GF(2^q).

Step 5: Calculate all the probabilities of the i-th column from $i = 1$ to n, if $i < n$, $i \leftarrow i+1$ and go to **Step 2**, else stop

---

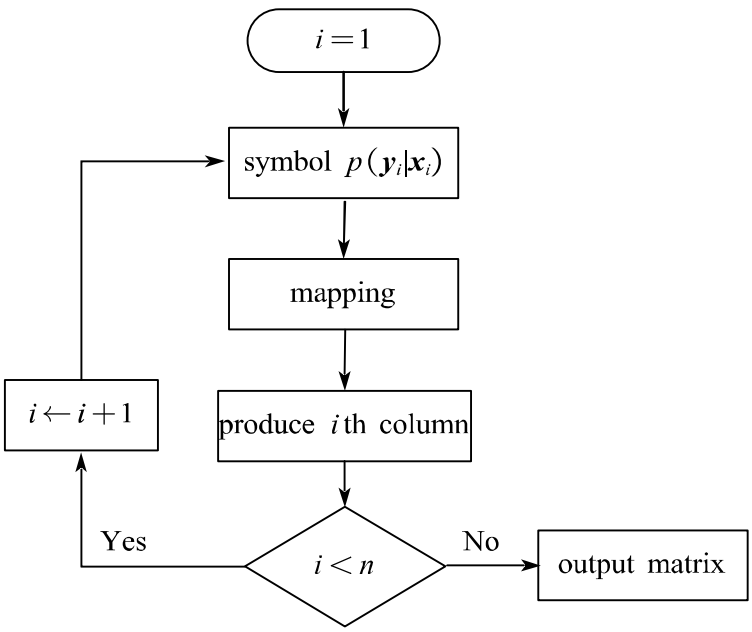

Fig. 16. Flow chart of the proposed SISO RLL decoding algorithm

4.4 Simulation result

To verify the proposed system, two cases of RLL codes are considered: 4B6B RLL codes with RS (15, 3) codes, RS (15, 7) codes, and RS (15, 11) codes over GF(2^4) and 8B10B RLL codes with RS (64, 32) codes over GF(2^8) in the VLC standard [1]. The reason why three RS codes are considered in the 4B6B RLL codes is that the effect of the proposed scheme needs to be investigated in various code rates, and the RS codes over GF(2^4) where the lowest, middle, and the highest code rates are RS (15, 3) codes, RS (15, 7) codes, and RS (15, 11) codes, respectively [1].

The performances of the proposed RLL decoding based on soft RS decoding are compared with those of hard-input RLL decoding with BM and soft-input RLL decoding with BM [8] which corresponding to 'proposed', 'h-BM', and 's-BM', respectively (in from Fig. 17 to Fig. 24). In these figures, E_b/N_0 is used for the signal-to-noise ratio (SNR), where E_b is the bit energy, and N_0 is the noise energy. To demonstrate performance gain, the required SNRs of the proposed scheme and the referenced scheme for target performance are listed in Table 6 and Table 7. The dB gain in these tables is the difference between the required SNR of the 'referenced' scheme and the 'proposed' scheme, and the percent gain is defined as

$$\frac{\text{required SNR of 'referenced' one}}{\text{required SNR of 'proposed' one}} \times 100(\%).$$

The 'Gain_A' and 'Gain_B' in Table I and Table II are the gains when 'h-BM' and 's-BM' are used as the 'referenced' scheme, respectively.

The BER results of 4B6B RLL codes are shown in Fig. 17- Fig. 20, and the required SNRs and gains at BER=10^{-5} are listed in Table 6.

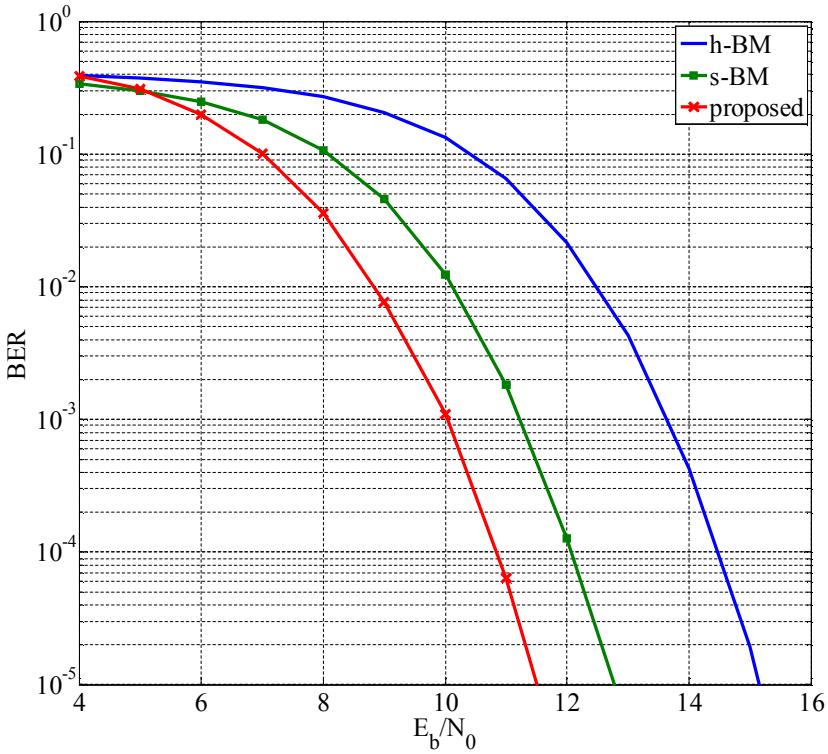

Fig. 17. BER performances of 4B6B RLL codes with RS (15, 3) codes.

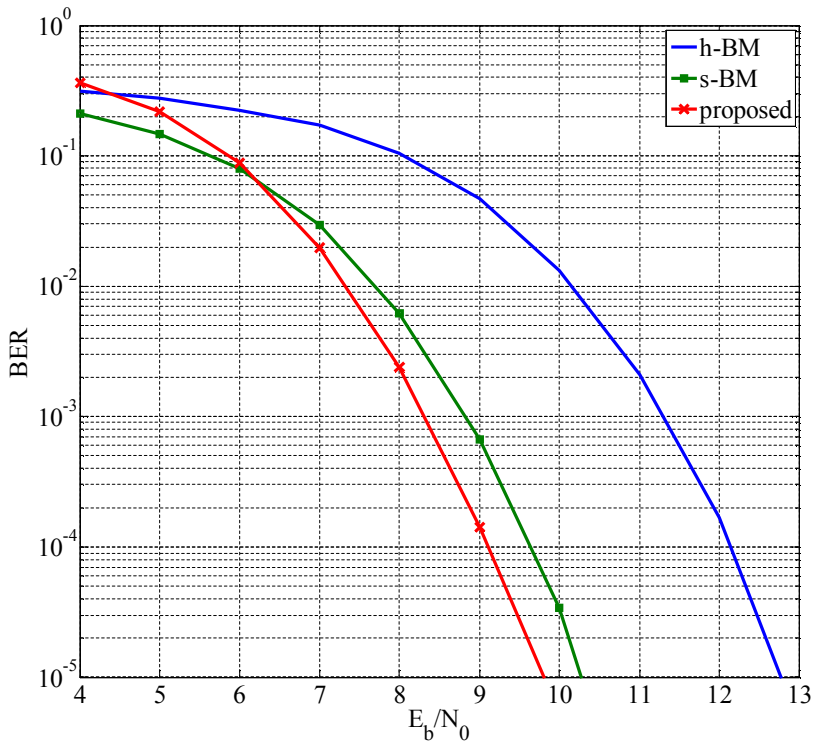

Fig. 18. BER performances of 4B6B RLL codes with RS (15, 7) codes.

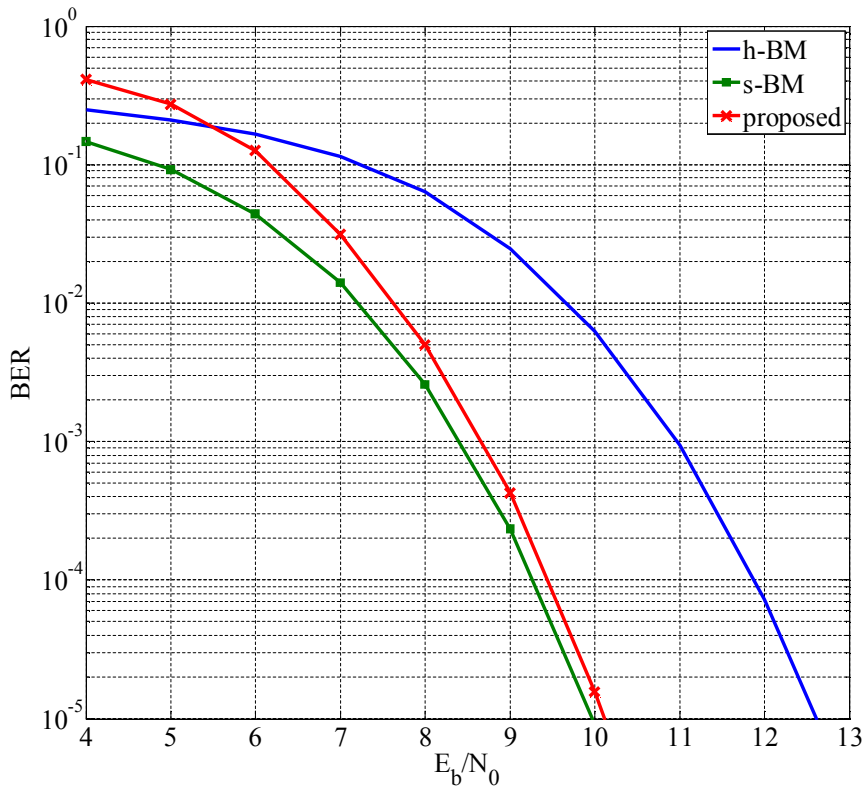

Fig. 19. BER performances of 4B6B RLL codes with RS (15, 11) codes.

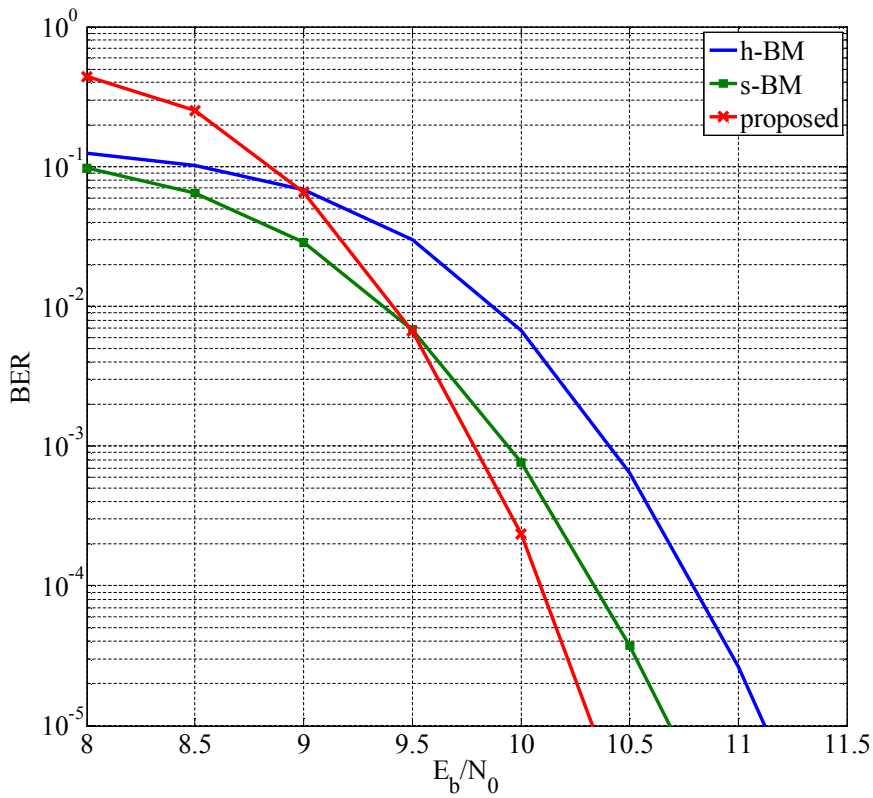

Fig. 20. BER performances of 8B10B RLL codes with RS (64, 32) codes.

Table 6. The required SNR and gains at BER=10^{-5}

BER		Proposed	s-BM [8]	h-BM	$Gain_A$		$Gain_B$	
RLL	RS				dB	%	dB	%
4B6B	(15, 3)	11.51	12.77	15.16	1.26	110.9	3.65	131.7
	(15, 7)	9.81	10.28	12.77	0.47	104.7	2.96	130.1
	(15, 11)	10.11	9.97	12.61	-0.14	98.6	2.50	124.7
8B10B	(64, 32)	10.33	10.69	11.12	0.36	103.4	0.79	107.6

As shown, the BER performances of the proposed scheme are better than those of conventional RLL decoding ('h-BM') and those of 's-BM' [16] except in Fig. 19. In Fig. 17, Fig. 18, and Fig. 19, the performance gain of 4B6B RLL codes is larger when low-rate RS codes are used since the symbol probabilities updated from the proposed scheme are more effective for RS codes with a larger minimum distance. In Fig. 18 and Fig. 20, the rates of RS codes are nearly the same, but the performance gain for 4B6B is better than that of 8B10B since the code rate of the 4B6B codes is lower than that of 8B10B code.

Under the same conditions, FER results are shown in Fig. 21- Fig. 24, and the required SNRs and gains at FER=10^{-4} are listed in Table 7.

Table 7. The required SNR and gains at FER=10^{-4}

FER		proposed	s-BM [8]	h-BM	$Gain_A$		$Gain_B$	
RLL	RS				dB	%	dB	%
4B6B	(15, 3)	11.05	12.42	14.90	1.37	112.3	3.85	134.8
	(15, 7)	9.31	10.12	12.61	0.81	108.7	3.30	135.4
	(15, 11)	9.64	9.89	12.52	0.25	102.5	2.88	129.8
8B10B	(64, 32)	10.16	10.72	11.14	0.56	105.5	0.98	109.6

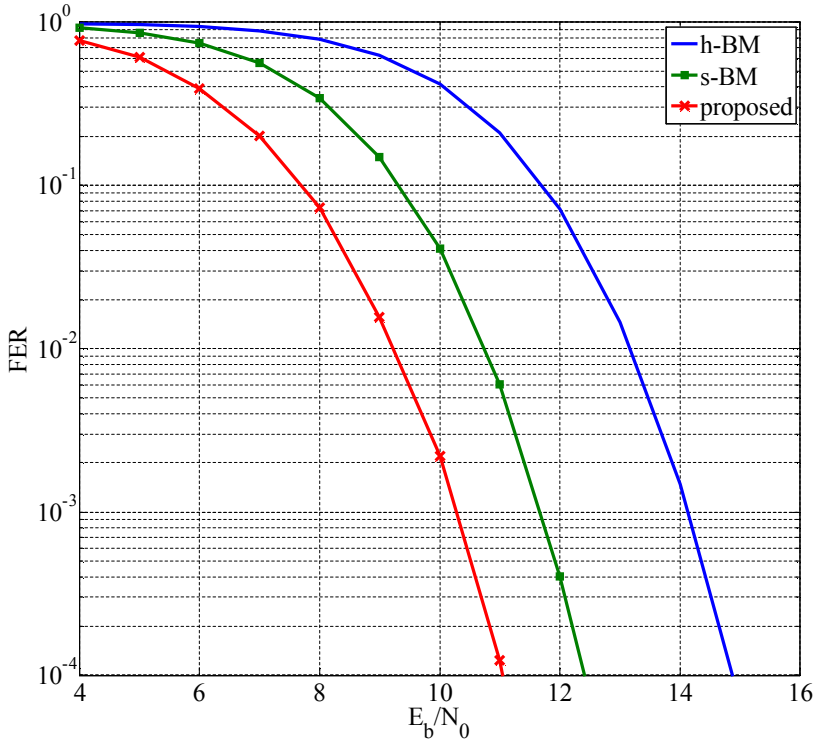

Fig. 21. FER performances of 4B6B RLL codes with RS (15, 3) codes.

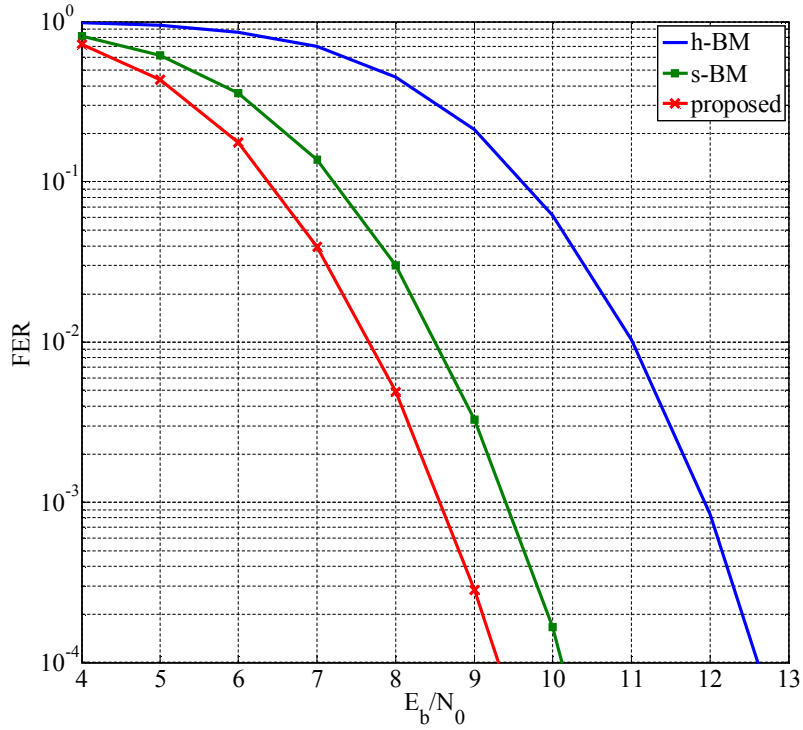

Fig. 22. FER performances of 4B6B RLL codes with RS (15, 7) codes.

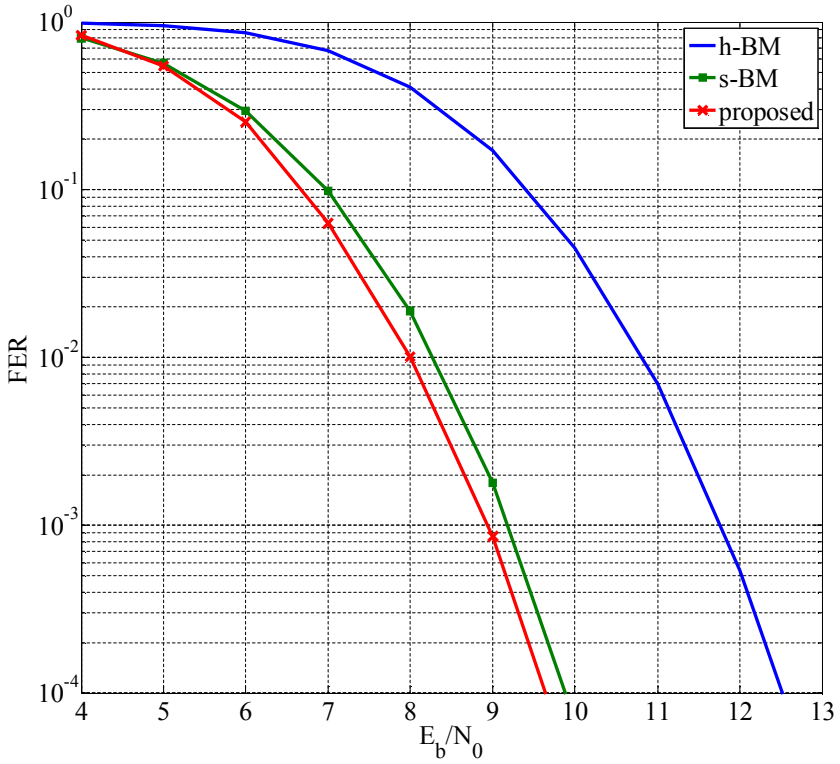

Fig. 23. FER performances of 4B6B RLL codes with RS (15, 11) codes.

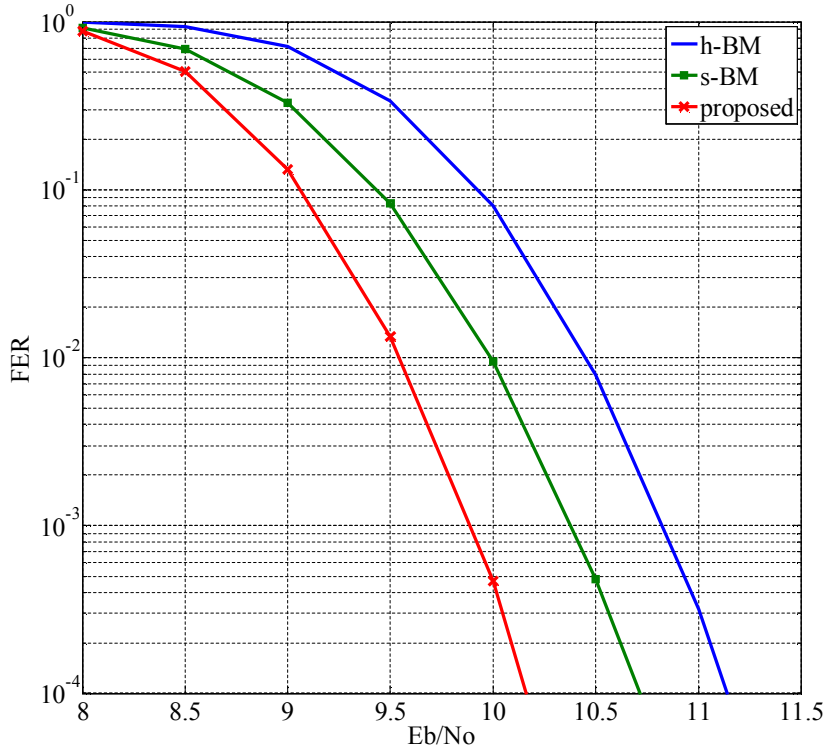

Fig. 24. FER performances of 8B10B RLL codes with RS (64, 32) codes.

As shown, the FER performances of the proposed scheme are always better than those of the referred RLL decoding [16]. In comparison between the BER and the FER, the FER gain of our proposed codes is better than the BER gain. If an automatic repeat request (ARQ) scheme is operated in VLC systems, the proposed scheme is more effective since the total throughput of ARQ is affected by FER.

4.5 Conclusion

A new soft-input soft-output RLL decoding in VLC has been proposed in this paper. We modified receiver part in VLC, as the new receiver architecture, the soft information from channel is decoded into a reliable matrix by the proposed RLL decoder, then the reliable matrix can be used for the conjugated soft RS decoder to enhance the throughput in the VLC system. The proposed scheme shows better BER and FER performances than those of the referenced schemes ('h-BM' and 's-BM'). Our proposed schemes can especially increase throughput of a VLC scheme with ARQ.

CHAPTER 5
BIT LEVEL SOFT RLL DECODING

5.1 Introduction

VLC [1] is a type of wireless communication that utilizes visible light waves, wavelength between 375 nm and 780 nm, to transmit information. With the development of lighting techniques, an increasing number of light sources have adopted LEDs [2], which have advantages over legacy tungsten and florescent-based lamps such as high brightness, long life, low power consumption, and fast nanosecond switching times. Therefore, VLC will become a competitive candidate (with the assistance of LEDs in the wireless communication field in the near future. Among the significant techniques in VLC, OOK is one of simplest modulations to transmit a binary 1/0 through an on/off pulse. Moreover, RLL codes are widely used in VLC for dimming control; RLL codewords (with RLL decoding) contain the same 0s and 1s to guarantee a DC balance. In the VLC standard [1], three kinds of RLL codes are considered, including Manchester, 4B6B codes, and 8B10B codes.

RS codes have perfect error correction ability since they are a class of maximum distance separable (MDS) codes. They are widely used in data transmission technologies, broadcast systems, and optical communication. The BM algorithm [17] that utilizes a linear feedback shift register (LFSR) is one of the efficient hard RS decoding algorithms. In general, the parity check matrix, $\boldsymbol{H}_s$, of the RS codes is a Vandermonde matrix [17] on a symbol level; it can be easily transferred to an matrix $\boldsymbol{H}_b$ on a bit level [21]. The main novelty of BLS RS decoding is to reduce the density of $\boldsymbol{H}_b$ in which the least reliable bits correspond to a spare submatrix. The soft RS decoding algorithm [21] based on bit-level soft (BLS) information was proposed by iteratively updating the parity check matrix and the log-likelihood ratio (LLR), and it was shown that the performances of the proposed decoding were better than those of conventional RS decoding.

Since the emergence of VLC, a lot of research has been studied to produce an efficient transmission of VLC. A novel forward error correction (FEC) coding method in VLC systems was proposed in [12],

which used a modified Reed-Muller (RM) code for providing accurate dimming control. Moreover, a modified RM coding scheme based on the bent function for dimmable VLC was proposed in [13] and suggests a new method to contain minimal compensation symbols to simultaneously support multiple dimming target values and improve coding gain. In [14], an efficient transmission method combining RS coding with RLL codes in power line communication was proposed and presented a spectral analysis of RLL codes. Adaptive methods based on low-density parity check codes for VLC [20] were also proposed to enhance the transmission efficiency. Despite these research, the bit error rate (BER) performance of RLL codes with RS codes in VLC is rarely considered. In [16], multiple output RLL decoding was proposed for RLL decoder output candidates with higher probabilities; however, the method was a type of hard RLL decoding that did not utilize soft information from the channel. However, only symbol level decoding is previously considered such as in [16] or conventional hard RLL decoding, therefore the study about bit-level RLL decoding is necessary for efficient transmission technique in VLC.

In this chapter, we propose a BLS RLL decoding in VLC with OOK modulation and BLS RS codes. In the receiver part of our system, the BLS RLL decoder produces BLS output, and all of this soft information can be matched as input of the BLS RS decoder. The simulation results show that our proposed BLS RLL decoding has a significantly better BER performance than that of conventional hard RLL decoding and compares favorably with that of the referenced RLL decoding methods.

5.2 Proposed VLC systems

A block diagram of the proposed system is shown in Fig. 25. The transmitter in Fig. 25 (a) is composed of an RS encoder, an RLL encoder, and OOK-modulated LEDs; it is similar to the transmitter in the VLC standard [1].

RS (n, k) codes [4] over GF(2^q) are considered where $n \leq 2^q$ and GF(2^q). The kq-bit message is $\boldsymbol{u}$ = ($\boldsymbol{u}_1$, $\boldsymbol{u}_2$,…, $\boldsymbol{u}_i$,…, $\boldsymbol{u}_k$), where $\boldsymbol{u}_i$ = ($u_{i,1}$, $u_{i,2}$,…, $u_{i,q}$) is encoded to nq-bit codeword $\boldsymbol{c}$ = ($\boldsymbol{c}_1$, $\boldsymbol{c}_2$,…, $\boldsymbol{c}_i$,…, $\boldsymbol{c}_n$), where $\boldsymbol{c}_i$ = ($c_{i,1}$, $c_{i,2}$,…, $c_{i,q}$). Next, in the RLL encoder, the q-bit vector, $\boldsymbol{c}_i$, is encoded to the s-bit RLL codeword $\boldsymbol{x}_i$= ($x_{i,1}$, $x_{i,2}$, …, $x_{i,s}$) according to the mapping rule, E [3], as $E(\boldsymbol{c}_i) = \boldsymbol{x}_i$ for i = 1, 2,…, n. Then, the concatenated output of RLL codes $\boldsymbol{x}$ = ($\boldsymbol{x}_1$, $\boldsymbol{x}_2$,…, $\boldsymbol{x}_n$) is OOK-modulated and emitted to the channel.

The receiver in Fig. 25 (b) is composed of an OOK demodulated photodiode (PD), a BLS RLL decoder, and a BLS RS decoder. The BLS RLL decoder is composed of an inverse mapping, E^{-1} (as $E^{-1}(\boldsymbol{x}_i) = \boldsymbol{c}_i$), and an adder. The received OOK-demodulated signal, $\boldsymbol{y}$ (= $\boldsymbol{x}$+$\boldsymbol{n}$), is the input of the RLL decoder, where $\boldsymbol{n}$ is AWGN with variance σ^2 [6], [7]. The considered RS decoder adopts a soft decoding algorithm [5] based on iteratively updating the LLR of messages in order to enhance the performance.

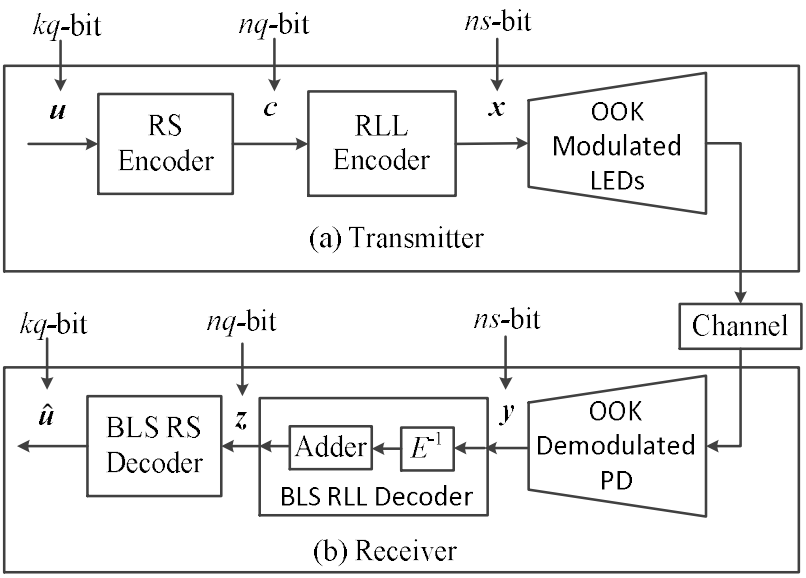

Fig. 25. Block diagrams of the transmitter and the receiver of the proposed VLC system.

This chapter focus on BLS RLL decoding combined with BLS RS decoding, and a detailed description of our algorithm is given.

5.3 Bit level RLL decoder

To denote the RLL codes such as 4B6B in the VLC standard [4], (q, s) is used instead of (4, 6) since (q, s) is conventional case in algorithm, it also instead of (1, 2) or (8, 10) which corresponds to Manchester codes or 8B10B codes. The (q, s) RLL codes contain q-bit vectors and s-bit codewords that correspond to each other, and the average weight of each s-bit codeword is $s/2$ in order to maintain a DC balance.

The input of the RLL decoder is defined as $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n)$, where $\boldsymbol{y}_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,s})$ for i = 1, 2,…, n; the output of the RLL decoder is defined as $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n)$, where $\boldsymbol{z}_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,s})$ for i = 1, 2,…, n. The RLL decoder decodes $\boldsymbol{y}$ by processing (s-bit codeword) $\boldsymbol{y}_i$ and the output BLS information that can be used in RS decoding [21]. A detailed explanation of BLS RLL decoding follows.

The s-bit, $\boldsymbol{y}_i$, is the input of E^{-1}, and the corresponding posteriori probabilities (of 2^q possible RLL codewords, $\boldsymbol{x}_i$) are defined as

$$p(\boldsymbol{y}_i \mid \boldsymbol{x}_i) = \prod_{j=1}^{s} p(y_{i,j} \mid x_{i,j}).$$

Each bit probability based on OOK can be calculated as

$$p(y_{i,j} \mid x_{i,j} = \Delta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_{i,j}-\Delta)^2}{2\sigma^2}},$$

where Δ equals 0 or 1. According to the reverse mapping rule that $E^{-1}(\boldsymbol{x}_i) = \boldsymbol{c}_i$, the corresponding q-bit vectors have the same probabilities as the s-bit codewords. Then,

$$p(\boldsymbol{y}_i \mid \boldsymbol{c}_i) = p(\boldsymbol{y}_i \mid \boldsymbol{x}_i).$$

The q-bit vector probabilities, $p(\boldsymbol{y}_i|\boldsymbol{c}_i)$, are the input of the adder. Similar to calculate marginal probability from the joint probability [22], bit probability is calculated by summation for each corresponded bit. Then, the j-th bit probability in the i-th symbol in the adder of the RLL decoding is calculated as

$$p(y_{i,j} \mid c_{i,j} = \Delta) = \sum_{c_{i,j}=\Delta} p(\boldsymbol{y}_i \mid \boldsymbol{c}_i), \text{ for } j = 1, 2, \ldots, q,$$

where Δ equals 0 or 1. Finally, in the BLS RLL decoder, LLR $z_{i,j}$ is produced as

$$z_{i,j} = \log \frac{p\left(y_{i,j} \mid c_{i,j} = 0\right)}{p\left(y_{i,j} \mid c_{i,j} = 1\right)}, \tag{5.1}$$

where the LLR is used as the input of the BLS RS decoder.

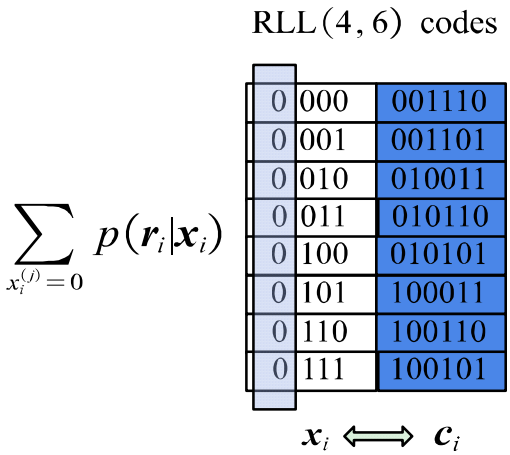

Fig. 26. Example of soft output decoding $x_i^{(j)} = 0$ for RLL (4, 6) codes.

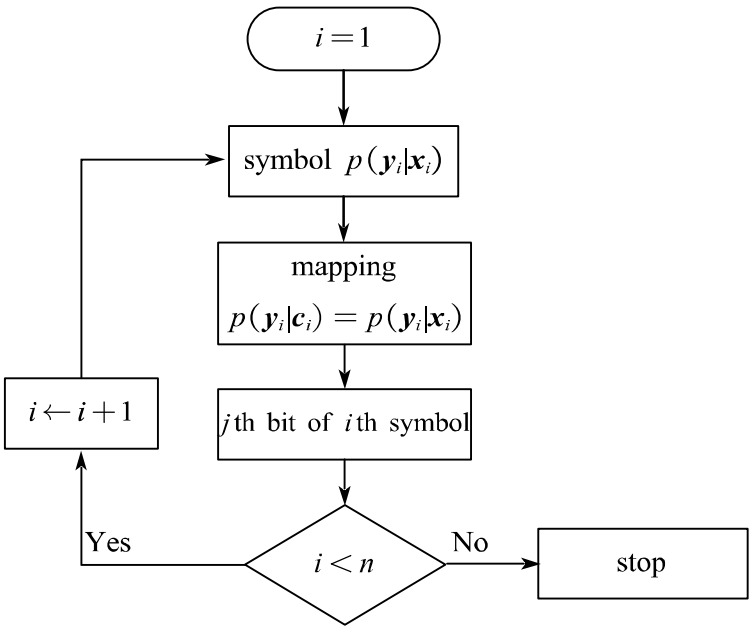

Fig. 27. Flow chart of the proposed SISO RLL decoding algorithm

5.4 Bit-level soft RS decoder

The parity-check matrix of RS codes over GF(2^q) can be represented as a (n-k) × n Vandermonde matrix in symbol level, where α is a primitive element in GF(2^h).

$$H_s = \begin{pmatrix} 1 & \alpha & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \dots & \alpha^{2(n-1)} \\ \dots & \dots & \dots & \dots \\ 1 & \alpha^{n-k} & \dots & \alpha^{(n-k)(n-1)} \end{pmatrix}$$

Due to each element of $\boldsymbol{H}_s$ in GF(2^q) can be expended to a $q \times q$ sub-matrix. For example, α = (0 0 1 0) in binary case from GF(2^4), it can be rewrite in a sub-matrix as

$$\alpha = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Then (n-k) × n matrix $\boldsymbol{H}_s$ transform to a (n-k)q×nq matrix $\boldsymbol{H}_b$. For example, parity check matrix of RS (7, 5) codes is

$$H_s = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \end{bmatrix}$$

And its binary form is

$$H_b = \left[\begin{array}{ccc|ccc|ccccccccccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right]$$

The main work of soft decoding of RS codes is to reduce the density of H_b. According input of soft RS decoder, the corresponded magnitude | $z_{i,j}$ | in (5.1), needs to be sorted in ascending order where corresponding positions are also stored. Then, Gaussian elimination is implemented to systematize (n-k)q positions with low reliability as [21]

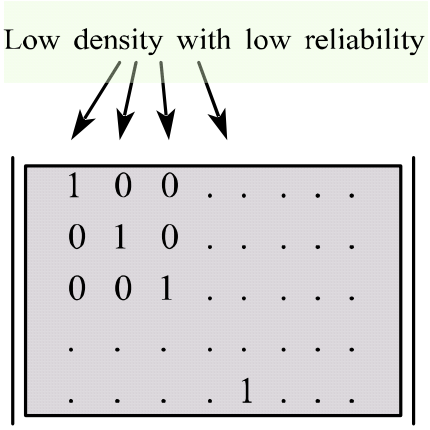

Fig. 28. The parity check matrix in bit level is transformed to low density parity check matrix.

Next, the sum product algorithm (SPA) is used to update the LLR, $z_{i,j}$, for improving performance. At each iteration, the matrix and LLR are updated. This process is continued until the hard decision of the LLR is an RS codes, or the given number of iterations is reached.

5.5 Simulation result

To verify the proposed system, 4B6B RLL codes with RS (15, 3) codes, RS (15, 7) codes, and RS (15, 11) codes over GF(2^4) in the VLC standard [1] are considered in the simulations. The reason why three RS codes are considered is that the effect of the proposed scheme needs to be invested in different code rates, and the RS codes over GF(2^4) with the lowest, middle, and the highest code rates are RS (15, 3) codes, RS (15, 7) codes, and RS (15, 11) codes, respectively [1].

The performances of the proposed RLL decoding ('BLS RLL' in Fig. 29 to Fig. 34) are compared with those of the conventional hard RLL decoding ('Hard RLL' in Fig. 29 to Fig. 34), MPS RLL decoding [10] ('MPS RLL' in Fig. 29 to Fig. 34), and SLS RLL decoding ('SLS RLL' in Fig. 29 to Fig. 34). In these figures, E_b/N_0 is used for the signal-to-noise ratio (SNR), where E_b is the bit energy, and N_0 is the noise energy. To show the BER and FER performance gains, the required SNRs of the proposed scheme and the referenced scheme for the target performance are listed in Table I and Table II, respectively. 'Gain_A', 'Gain_B', or 'Gain_C' in Table I and Table II denote the difference between 'Hard RLL', 'MPS RLL', or 'SLS RLL' and 'BLS RLL', respectively. The gain '%' is defined as

$$\frac{\text{SNR of the referenced scheme}(\text{dB})}{\text{SNR of the proposed scheme}(\text{dB})}\times 100\%,$$

where the referenced scheme indicates 'Hard RLL', 'MPS RLL', or 'SLS RLL', and the proposed scheme indicates 'BLS RLL'. The BER performances of the 4B6B RLL codes are shown in Fig. 29, Fig. 30, and Fig. 31, respectively.

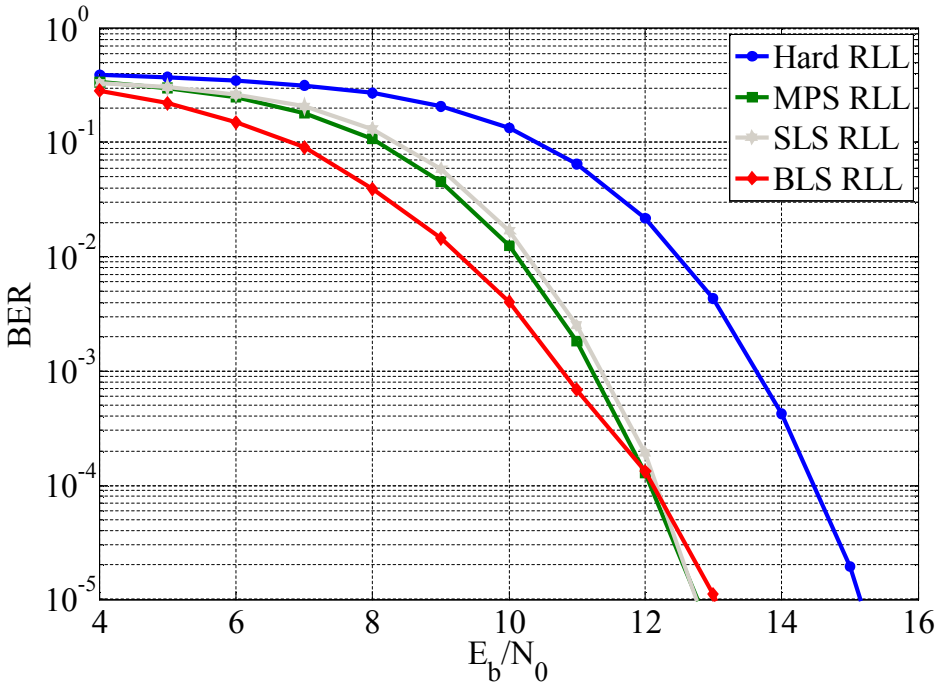

Fig. 29. BER performances of 4B6B RLL codes with RS (15, 3) codes.

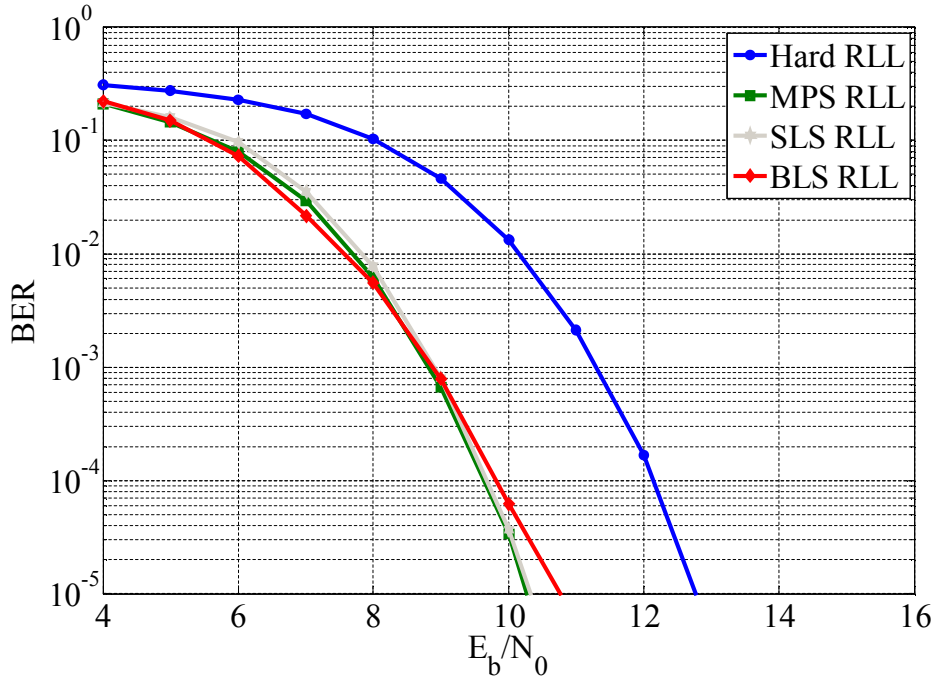

Fig. 30. BER performances of 4B6B RLL codes with RS (15, 7) codes.

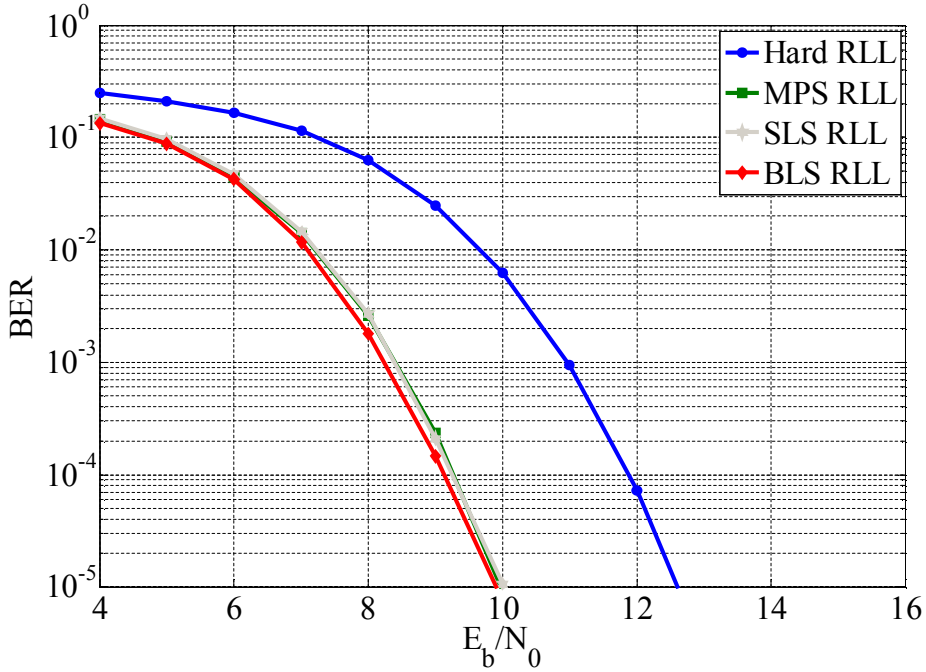

Fig. 31. BER performances of 4B6B RLL codes with RS (15, 11) codes.

In the BER results in Fig. 29 to Fig. 31, the performances of the proposed decoding are significantly better than the conventional hard RLL decoding. In comparison with a recent decoding based on 'MPS RLL' [16] and 'SLS RLL', the proposed decoding exhibits better results when the SNR is relatively small; however, at a high SNR, it exhibits a similar performance.

In Table 8, the exact BER performances and corresponding SNR gains are listed for performance comparison purposes. At BER = 10^{-5}, the 4B6B codes combined with the RS codes with a higher code rate have a better performance since the RS decoding converges quickly. The dB gains of '$Gain_A$' are higher than 2.0 dB for the 4B6B code; the dB gains of '$Gain_B$' (similar to the dB gains of '$Gain_C$') are about 0 dB, but the decoding processes are different. At the same time, percentage gains of '$Gain_A$' are higher than 115% for the 4B6B code; the percentage gains of '$Gain_B$' (similar to the percentage gains of '$Gain_C$') are about 100 % which means they have similar performance.

Next, the FER performances of 4B6B are shown in Fig. 32, Fig. 33, and Fig. 34, respectively. In these simulations, one codeword of the RS codes corresponds to one frame. For application to hybrid automatic repeat request (ARQ), the range of FER is from 1 to 10^{-2} since the FER of the initial transmission is considered to be around 10^{-1}. The exact FER performances and corresponding SNR gains are shown in Table 9.

Table 8. The required SNR and gains at BER = 10^{-5}

		RS	(15, 3)	(15, 7)	(15, 11)
BER	RLL	Hard	15.16	12.77	15.16
		MPS	12.77	10.28	9.97
		SLS	12.76	10.33	10.01
		BLS	13.04	10.77	9.91
	$Gain_A$	dB	2.12	2.00	2.67
		%	116.3	118.6	127.3
	$Gain_B$	dB	-0.27	-0.49	0.06
		%	97.9	95.5	100.1
	$Gain_C$	dB	-0.28	-0.44	-0.10
		%	97.9	95.9	101.0

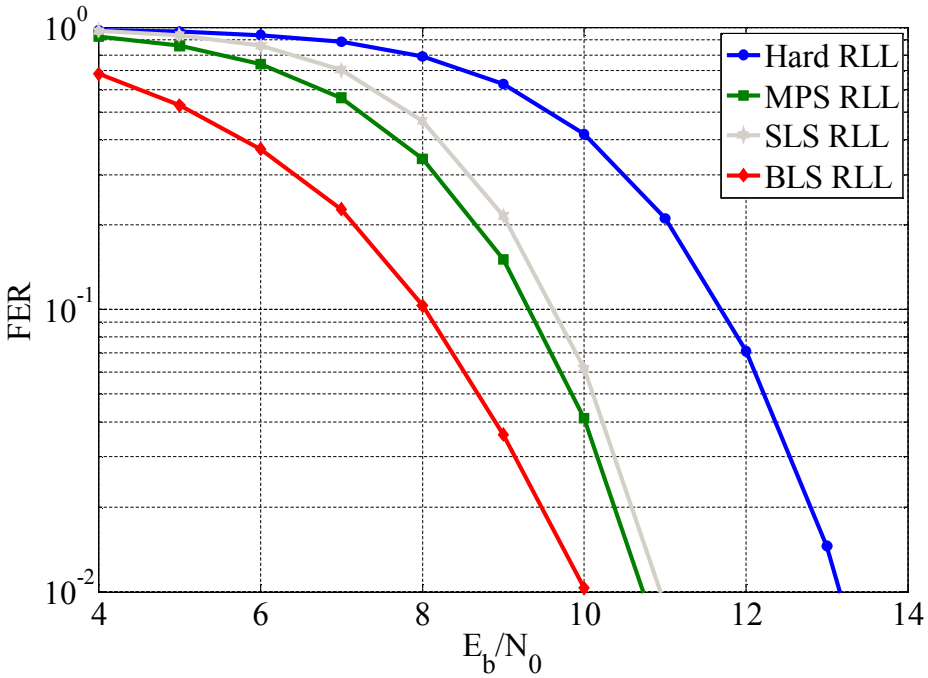

Fig. 32. FER performances of 4B6B RLL codes with RS (15, 3) codes.

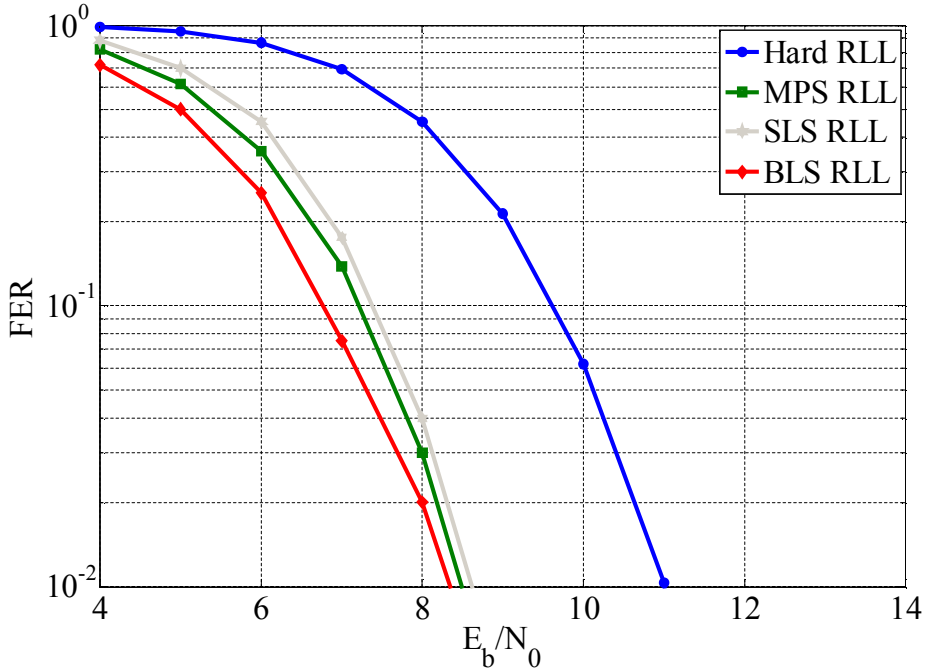

Fig. 33. FER performances of 4B6B RLL codes with RS (15, 7) codes.

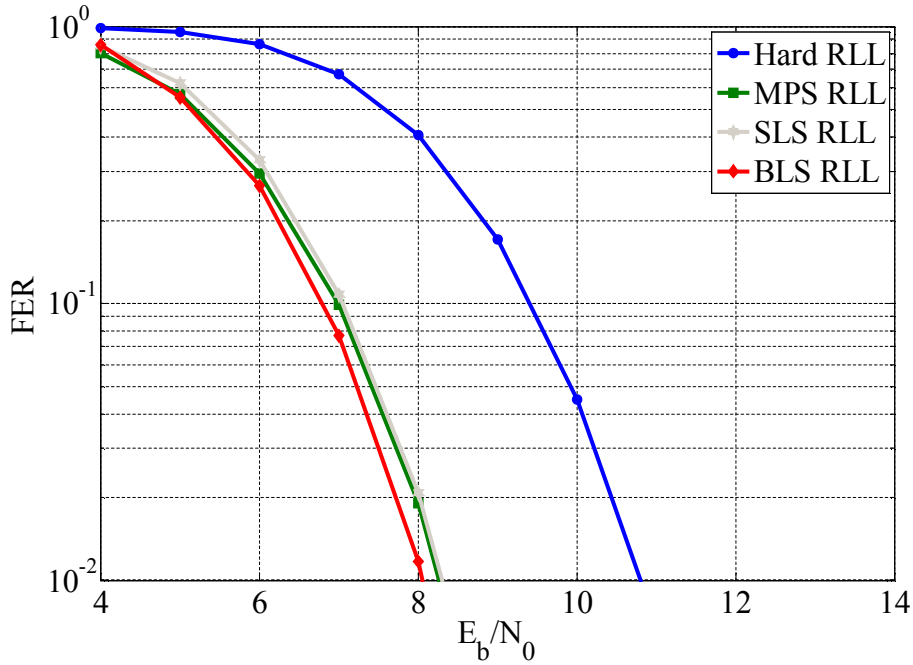

Fig. 34. FER performances of 4B6B RLL codes with RS (15, 11) codes.

From the FER results in Fig. 32 to Fig. 34, the performances of the proposed 'BLS' are always better than the 'Hard RLL', 'MPS RLL' and 'SLS RLL', although BER performances of the proposed

‘BLS RLL’ are not better than ‘MPS RLL’ in Fig. 2 and Fig. 3. Therefore the proposed algorithm is meaningful for application to hybrid ARQ. Compare FER performances in Fig. 5 to Fig. 7, 4B6B codes combined with the RS codes with a higher code rate have a better performance since the RS decoding [21] quickly converges.

Table 9. The required SNR and gains at FER = 10^{-2}

		RS	(15, 3)	(15, 7)	(15, 11)
FER	RLL	Hard	13.16	11.01	10.81
		MPS	10.74	8.50	8.27
		SLS	10.95	8.63	8.30
		BLS	10.02	8.36	8.06
	$Gain_A$	dB	3.14	2.65	2.75
		%	131.3	131.7	134.1
	$Gain_B$	dB	0.72	0.14	0.21
		%	107.2	101.7	102.6
	$Gain_C$	dB	0.93	0.27	0.24
		%	109.3	103.2	103.0

In Table 9, the performance gains of the proposed scheme are greater than 2.5 dB, 0.1 dB, and 0.2 dB in comparison with the conventional ‘Hard RLL’, recent ‘MPS RLL’, and ‘SLS RLL’, respectively. Similar gains are obtained if the required FER is less than 10^{-2}.

5.6 Conclusion

A new RLL decoding algorithm from the BLS decoding in VLC in combination with RS codes is proposed. In the receiver part of our system, the proposed BLS RLL decoder produces BLS output, and all of this soft information can be matched to input of the BLS RS decoder. The simulation results show that our proposed BLS RLL decoding has significantly better BER performances than those of the conventional hard RLL decoding and compares favorably with that of the referenced RLL decoding. When bit-level decoding is considered to enhance the throughput in VLC, the proposed decoding can be efficiently applied.

CHAPTER 6
RLL WITH MODIFIED BCJR

6.1 Introduction

VLC [1] is a developing technique that transmits information using visible light waves. This field is gaining wide attention from other researchers, since visible light is not harmful to people, and the LED is spreading quickly all over the world. The LED is a high-efficient light source compared with previous filament lamps, and LEDs have good properties, such as long life, avoidance of moisture, low cost, etc. There are some appropriate modulations [24] in VLC, such as OOK [25], pulse position modulation (PPM) [26], and colour shift keying modulation (CSK) [27], and the OOK is widely adopted since its simplicity of sending 0/1 by off/on pulse.

Convolutional codes are linear codes that have an additional structure in the generator matrix, so that the encoding operation can be viewed as a filter or convolutional operation. Therefore, convolutional codes are broadly used in digital video, radio, mobile communications, and satellite communications. These codes are also implemented in concatenation with other codes, particularly RS codes. RS codes are linear block and are also maximum distance separate codes that have good properties to correct bursty errors. RLL codes are used to keep DC balance, since each codeword has the same number of 0s and 1s. Although there are many kinds of RLL codes, the only three RLL codes based on the VLC standard [1], are Manchester codes, 4B6B codes, and 8B10B codes. Convolution codes can be conjugated with RS codes and RLL codes under the VLC standard [1].

Recently, there has been a great deal of effort to find more efficient transmission techniques in VLC systems. In [12], a forward error correction (FEC) method using a modified Reed-Muller (RM) codes was proposed in VLC systems, which can contain minimal compensation symbols in supporting multiple dimming target values. Furthermore, a bent function was considered in VLC based on RM and OOK [13], which increased performance with accurate dimming target values. Moreover, adaptive methods based on low-density parity check codes for VLC [20] were also proposed to

enhance transmission efficiency. An adaptive normalised least mean square algorithm was proposed for the estimation of LED distortion [27] which had better performance than the conventional predistorter in VLC systems. In [28], an incremental scheduling scheme was proposed to improve system capacity with reduced complexity. An adaptive channel estimation algorithm was proposed to enhance the bit error rate of VLC systems [29]. The concatenated codes are popular used in VLC.

In general, the Viterbi algorithm [30] was used in convolutional decoding, since the algorithm is based on a dynamic programming algorithm to output a maximum likelihood code sequence based on a trellis, it is highly parallelizable, and Viterbi decoders are easy to implement in hardware and software. The trellis is an efficient structure to reduce the complexity of maximum likelihood decoding, since only one path remains for one terminal state point, by comparison. The Bahl, Cocke, Jelinek, Raviv (BCJR) algorithm [31] is an algorithm for maximum a posteriori decoding of error correcting codes defined on trellises, this algorithm is critical to modern iteratively-decoded error-correction codes including turbo codes and LDPC codes, but it is less used for only convolutional codes. Therefore, we consider used BCJR algorithm to improve performance in concatenated codes. In conventional RS decoding, the BM [31] algorithm was used to find error location since it can be easily applied with a LFSR, but it is one type of hard decoding, which performance is not good enough. Bit-level soft RS decoding [32] can be concatenated with BCJR, since this RS decoding [32] provide better performance by iteratively updating the RS parity check matrix and log-likelihood ratio (LLR) which come from the output of BCJR decoding. Soft-input hard-output (SIHO) RLL decoding was proposed [16], combining RLL codes and RS codes and RLL decoder output candidates with higher probabilities. However, the hard output of the RLL decoder loses some valuable information. Studies into RS codes and convolutional codes are mature; however, the lack of an efficient RLL decoding algorithm is a long-standing problem. One soft-input soft-output (SISO) RLL decoding is proposed in [40], they are used for 4B6B or 8B10B, when Manchester is considered, the decoding procedure is simple since Manchester codes are a special case of RLL codes which decoding output one bit. Hard convolutional decoding can easily be applied based on Viterbi algorithm. BCJR is equivalent with Viterbi for only convolutional codes. However,

when soft convolutional decoding is considered, efficient combination between RLL codes and convolutional is considerable, and our motive is find best decoding algorithm of concatenated codes in VLC. Therefore, we adopt BCJR for convolutional decoder as a bridge, between RS decoder and RLL decoder, to improve the total concatenated codes performance.

In this chapter, we propose soft convolutional codes decoding algorithm for concatenated codes in VLC, which can be considered a modified BCJR algorithm. RLL codes are conjugated with RS codes and convolutional codes; RS codes and convolutional codes are outer code and inner code, respectively. The RLL decoder produces bit-level soft information that can be used as input for concatenated codes. To produce soft input for RS decoder, a modified BCJR algorithm is proposed to enhance convolutional decoding with the soft RLL decoder. Simulation results show that our proposed convolutional decoding algorithm has a significantly better BER performance than referenced algorithms.

6.2 Proposed VLC systems

A block diagram of the proposed VLC system, which is composed of transmitter and receiver, is shown in Fig. 35. In Fig. 35 (a), the transmitter is composed of an RS encoder, an interleaver, a convolutional encoder, an RLL encoder, and an LED. The RS (n, k) codes are considered in GF(2^q). The k-symbol message, $\boldsymbol{m} = (\boldsymbol{m}_1, \boldsymbol{m}_2,\ldots, \boldsymbol{m}_k)$, is encoded into an n-symbol codeword, $\boldsymbol{s}= (\boldsymbol{s}_1, \boldsymbol{s}_2,\ldots, \boldsymbol{s}_i,\ldots, \boldsymbol{s}_n)$, in the RS encoder, where $\boldsymbol{s}_i = (s_{i,1}, s_{i,2},\ldots, s_{i,q})$. Then, codeword $\boldsymbol{s}$ is interleaved to $\boldsymbol{s}'$, where $\boldsymbol{s}' = (\boldsymbol{s}'_1, \boldsymbol{s}'_2,\ldots, \boldsymbol{s}'_j,\ldots, \boldsymbol{s}'_q)$ with $\boldsymbol{s}'_j = (s_{1,j}, s_{2,j},\ldots, s_{n,j})$. Next, the convolutional encoder produces convolutional codeword $\boldsymbol{c}$ which is RLL-encoded to $\boldsymbol{x}$, the convolutional codes with constraint length ζ; therefore, with code rate ε, the convolutional codeword $\boldsymbol{c} = (c_1, c_2,\ldots, c_\eta)$, where $\eta= (nq+\zeta-1)/ \varepsilon$, after tail padding ζ-1 zeros in $\boldsymbol{s}'$. The RLL codes are denoted as vBuB codes. Therefore, the input of RLL encoder $\boldsymbol{c}$ is divided into $(\boldsymbol{c}_1, \boldsymbol{c}_2,\ldots, \boldsymbol{c}_i,\ldots, \boldsymbol{c}_l)$ with padding $\lambda = (lv-\eta)$ 0s for RLL encoding, where $\boldsymbol{c}_i = (c_1, c_2,\ldots, c_v)$ and $l = \lceil \eta/v \rceil$, the output of RLL encoder $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2,\ldots, \boldsymbol{x}_i,\ldots, \boldsymbol{x}_l)$ and $\boldsymbol{x}_i= (x_1, x_2,\ldots, x_u)$. The v-bit $\boldsymbol{c}_i$ is encoded to $\boldsymbol{x}_i$, which is denoted as $E(\boldsymbol{c}_i)= \boldsymbol{x}_i$, based on the VLC standard [1]. Finally, the OOK modulated signal is sent via LEDs.

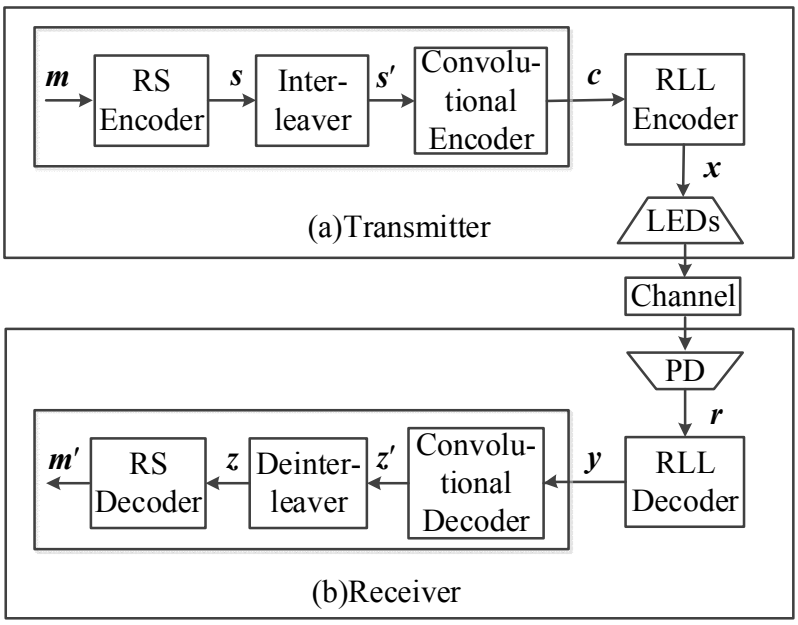

Fig. 35. Block diagrams of the proposed VLC system.

In Fig. 35 (b), the receiver is composed of a photodiode (PD), an RLL decoder, a convolutional decoder, a deinterleaver, and an RS decoder. The PD receives the signal affected by the noise, and produces $\boldsymbol{r}$ (=$\boldsymbol{x}$+$\boldsymbol{w}$) which is the input of the RLL decoder, where $\boldsymbol{w}$ is additive white Gaussian noise (AWGN) with variance σ^2. Then the input of RLL codes is denoted as $\boldsymbol{r} = (\boldsymbol{r}_1, \boldsymbol{r}_2,\ldots, \boldsymbol{r}_i,\ldots, \boldsymbol{r}_l)$, where $\boldsymbol{r}_i = (r_1, r_2,\ldots, r_u)$. Then, RLL-decoded $\boldsymbol{y}$ is convolutionally decoded to $\boldsymbol{z}'$, where $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2,\ldots, \boldsymbol{y}_i,\ldots, \boldsymbol{y}_l)$ with $\boldsymbol{y}_i = (r_1, r_2,\ldots, r_u)$ and $\boldsymbol{z}' = (\boldsymbol{z}'_1, \boldsymbol{z}'_2,\ldots, \boldsymbol{z}'_j,\ldots, \boldsymbol{z}'_q)$ with $\boldsymbol{z}'_j = (z_{1,j}, z_{2,j},\ldots, z_{n,j})$. Then, $\boldsymbol{z}'$ is deinterleaved to $\boldsymbol{z}$, where $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2,\ldots, \boldsymbol{z}_i,\ldots, \boldsymbol{z}_n)$ with $\boldsymbol{z}_i = (z_{i,1}, z_{i,2},\ldots, z_{i,q})$. Finally, after RS decoding, estimated message $\boldsymbol{m}'$ is obtained. The likelihood probability for $x_{i,j}$ is defined as

$$p\left(r_{i,j} \mid x_{i,j} = \theta\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(r_{i,j} - \theta\right)^2}{2\sigma^2}\right),$$

where θ is 0 or 1 for $i = 1, 2,\ldots, l$ and $j = 1, 2, \ldots, u$.

In the conventional decoding algorithm, hard decoding algorithm, Viterbi decoding algorithm [30], and BM algorithm [31] can be considered for RLL codes, convolution codes, and RS codes, respectively. However, our paper focuses on SISO RLL decoding combined with convolutional decoding and RS decoding.

6.3 SISO RLL decoder

The SISO RLL was described [40] for 4B6B codes and 8B10B codes, however it is special case for Manchester, since the encoder of Manchester codes encodes 1 to (1, 0) and 0 to (0, 1), which can be briefly show as follow.

The probability of codewords in vBuB RLLcodes can be calculated as

$$p(\boldsymbol{r}_i|\boldsymbol{x}_i) = \prod_{j=1}^{u} p(r_{i,j}|x_{i,j}) \tag{6.1}$$

where i = 1, 2,…, l and j = 1, 2, …, u.

For Manchester codes, which are the simplest RLL codes with v = 1, u = 2, $l = \lceil \eta/v \rceil = \eta$. According to Manchester encoding rules, the bit y_i has same probability as $\boldsymbol{r}_i$. Therefore,

$$p(y_i|c_i = 1) = p(\boldsymbol{r}_i|\boldsymbol{x}_i = (1, 0)), \tag{6.2}$$

$$p(y_i|c_i = 0) = p(\boldsymbol{r}_i|\boldsymbol{x}_i = (0, 1)), \tag{6.3}$$

where i = 1, 2, …, l.

For 4B6B codes or 8B10B codes, it is same with [20]. Which can be briefly show as follow, $v > 1$, according to mapping rule E($\boldsymbol{c}_i$) = $\boldsymbol{x}_i$, such that

$$p(\boldsymbol{y}_i|\boldsymbol{c}_i) = p(\boldsymbol{y}_i|\boldsymbol{x}_i) = p(\boldsymbol{r}_i|\boldsymbol{x}_i)$$

where i = 1, 2,…, l. The soft bit information from the RLL decoder can be calculate similar to a marginal calculation, such as

$$p\left(y_{i,j} \mid c_{i,j} = \theta\right) = \sum_{\substack{c_{i,j} \\ c_{i,j}=\theta}} p\left(\boldsymbol{y}_i \mid \boldsymbol{c}_i\right), \tag{6.4}$$

where i = 1, 2,…, l, and j = 1, 2,…, v.

6.4 Convolutional decoder

In conventional decoding for convolutional codes, the Viterbi algorithm [30] is used since the maximum possible output paths in trellis, therefore, the Viterbi algorithm is a maximum likelihood sequence estimator. In each stage of trellis, the path metric is updated by adding previous metric and branch metric, where the paths that merge to the same state point are compared with metrics. For each state, the path with minimum metric or maximum probability is remained. The errors from the Viterbi algorithm occur when the wrong path is retained, since a wrong path has a larger probability than a correct path. However, to achieve the soft output of convolutional decoding, the probability for each bit of the data in the paths is more useful. Moreover, the soft output for every codeword bit in convolutional codes are required to make soft input for RS decoder. Therefore, BCJR algorithm is considered for convolutional codes.

BCJR is a more efficient algorithm to calculate probability of a data bit when convolutional codes are concatenated with other codes. In the Viterbi algorithm, the path is updated by adding one more bit for each stage; the path with the lower probability is ignored, but all the paths contain the padded one-bit information, so they can be used to calculate bit output. The basic idea of BCJR [17] is summation of all the cases for bit-conditional probability as

$$\begin{aligned} p\left(s_{i,j}=\Delta \mid \boldsymbol{y}\right) &= \sum_{s_{i,j}=\Delta} p\left(s_{i,j} \mid \boldsymbol{y}\right) \\ &= \sum_{\psi,\varphi,\Delta} \alpha_t\left(\psi\right)\gamma_t\left(\psi,\varphi\right)\beta_{t+1}\left(\varphi\right) \end{aligned}, \qquad (6.5)$$

where i=1,2,…, n; j=1,2,…, q; t = (i-1) ×n+j; the state at the t-th stage and (t+1)-th stage of trellis are ψ and φ, respectively; and the bit bridges ψ to φ is Δ equals 0 or 1. The BCJR algorithm is composed of forward probability α_t, backward probability β_t, and smoothed probability γ_t. The main part is smoothed probability, since forward probability and backward probability depend on smoothed probability as follows:

$$\alpha_{t+1}\left(\varphi\right)=\sum_{\psi}\alpha_t\left(\psi\right)\gamma_t\left(\psi,\varphi\right), \quad \beta_t\left(\psi\right)=\sum_{\varphi}\gamma_t\left(\psi,\varphi\right)\beta_{t+1}\left(\varphi\right)$$

for trellis stage t = 1, 2,…, nq, respectively. The initial state for α_1 and β_{nq} are known to be all 0 states. The smoothed probability can be calculated as

$$\gamma_t = p\left(\boldsymbol{y}_t \mid \boldsymbol{c}_t\right) \tag{6.6}$$

In original BCJR, the smoothed probability need be calculated based on channel property, however it can be directly calculated in our algorithm from output of RLL decoder. When Manchester codes are considered, from (6.2), (6.3), and (6.6), the smoothed probability is

$$\gamma_t = p\left(\boldsymbol{y}_t \mid \boldsymbol{c}_t\right) = \prod_i p\left(y_i \mid c_i\right), \tag{6.7}$$

for stage t = 1, 2,…, nq. The remaining η - nq stages in trellis are ignored since they corresponding to padding 0's; i =1/ε*t+(1, 2,…, 1/ε) since convolutional codes rate ε = 1/3, 1/4 and 2/3 are defined in VLC standard [9], and convolutional codes with ε = 1/4 or 2/3 can be obtained by puncturing convolutional codes with ε = 1/3.

For 4B6B codes or 8B10B codes, from (4), (6), the smoothed probability is

$$p\left(\boldsymbol{y}_t \mid \boldsymbol{c}_t\right) = \prod_{i,j} p\left(y_{i,j} \mid c_{i,j}\right), \tag{6.8}$$

where stage t = 1, 2,…, nq; (v-1) i +j =1/ε*t+(1, 2,…, 1/ε). Therefore, it shows that 4B6B codes or 8B10B codes with v > 1 is more complex than Manchester codes with v = 1. From (6.7) or (6.8) based on RLL codes, the proposed concatenated codes can use BCJR directly in convolutional decoding.

Then, the LLR output can be calculated from (6.5) as

$$z_{i,j} = \frac{p\left(s_{i,j}=0 \mid \boldsymbol{y}\right)}{p\left(s_{i,j}=1 \mid \boldsymbol{y}\right)},$$

which can be used for soft RS decoding.

6.5 Deinterleaver and soft RS decoding

The interleaver is used according to the VLC standard [1], which is general adopted for concatenated codes. Interleaving ameliorates the burst error problem. In our system, for a corresponding deinterleaver, the input is $\boldsymbol{z}' = (\boldsymbol{z}'_1, \boldsymbol{z}'_2,\ldots, \boldsymbol{z}'_j,\ldots, \boldsymbol{z}'_q)$ with $\boldsymbol{z}'_j = (z_{1,j}, z_{2,j},\ldots, z_{n,j})$, and the output is $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2,\ldots, \boldsymbol{z}_i,\ldots, \boldsymbol{z}_n)$ with $\boldsymbol{z}_i = (z_{i,1}, z_{i,2},\ldots, z_{i,q})$. If the convolutional decoder adopts the Viterbi algorithm, the hard output is easily deinterleaved to $\boldsymbol{z}$, which can be used as input for the RS decoder with the BM algorithm [17]. The BM algorithm [17] will find error locator polynomial with minimal degree based on syndromes which can be calculated from $\boldsymbol{z}$ and primitive elements in GF(2^q), to find error locations, and further error values can be calculated. However, if the convolutional decoder adopts the BCJR algorithm, $\boldsymbol{z}$ will be the bit-level sequence of LLR, total system performance can be enhanced by changing the BM algorithm to soft RS decoding, and the remained decoding procedure is same to [32], the general symbol-level $(n\text{-}k)\times n$ parity check matrix is transferred to bit-level $(n\text{-}k)q\times nq$ parity check matrix $\boldsymbol{H}_b$. In each iteration, $|z_{i,j}|$ is sorted in ascending order, and Gaussian elimination is implemented to systematize $(n\text{-}k)q$ columns of $\boldsymbol{H}_b$, which corresponds to low reliable positions; LLR, $z_{i,j}$, is updated based on $\mathbf{H}_b$ with a sum-product algorithm, which is similar to a LDPC algorithm, since $\mathbf{H}_b$ is a suboptimal low-density matrix. At each iteration, hard decision of $\boldsymbol{z}$ is multiplied with $\mathbf{H}_b'$ to check that solution and the iteration is stopped when the solution is all 0s. Since systematic RS encoding is considered: $\boldsymbol{m}_i = \boldsymbol{s}_{i+n-k}$, for i=1, 2,…, k, the estimated message is $\boldsymbol{m}_i' = \boldsymbol{z}_{i+n-k}$. for i=1, 2, …, k.

6.6 Simulation results

In simulations, AWGN channel with variance σ^2 in VLC is considered, and convolutional codes with rates of 1/3 and 1/4 [1] and RS (15, 7) codes, RS (15, 11) codes, and RS (31, 15) codes are considered to check the effect of different code rates and lengths of codes. Similarly, Manchester and 4B6B codes are considered for RLL codes in order to check the effect of different types of RLL codes. In our simulation, the Manchester codes combine with RS (15, 7) codes and convolutional codes of rate 1/4; RS (15, 11) codes and convolutional codes of rate 1/3; the 4B6B RLL codes combine with RS (15, 7) codes and convolutional codes of rate 1/4, the Manchester codes combine with RS (31, 15) codes and convolutional codes of rate 1/4, respectively, where they are defined in the VLC standard [1]. For our proposed algorithm, the number of iteration in RS decoding is 30 based on soft RS decoding algorithm [21]. The number of iteration is 20 in [21], the performances become better with more iteration, but the iteration 40 is too complexity in our experiment, so the 30 is set as iteration number in our simulation.

The BER performances are shown in Fig. 36, Fig. 37, Fig. 38, and Fig. 39. The FER performances are shown in Fig. 40. In these figures, 'Ref.1', 'Ref.2', and 'Prop.' are used to present different algorithms. 'Ref.1' means HIHO RLL and hard convolutional codes; 'Ref.2' means SIHO RLL [16] and hard convolutional codes; 'Prop.' means SISO RLL and soft convolutional codes; the 'Prop.' is the proposed algorithm.

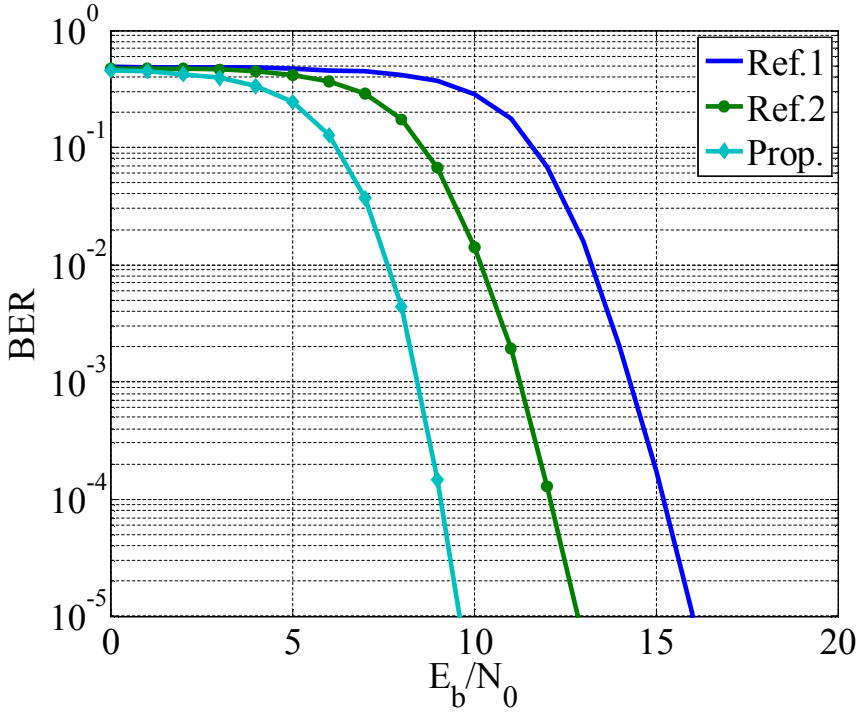

Fig. 36. BER performances of the system with Manchester codes, RS (15, 7) codes, and convolutional codes with rate 1/4.

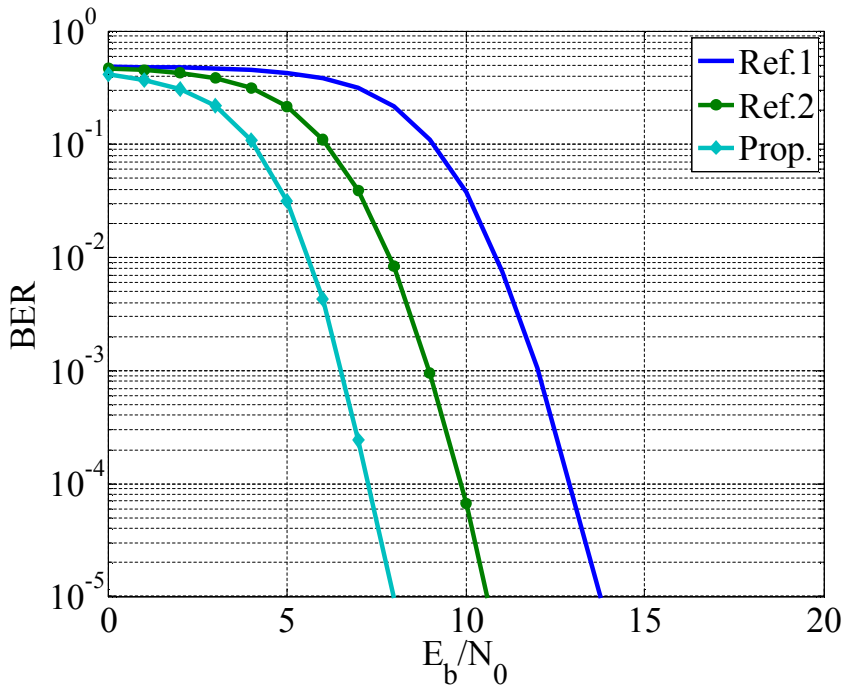

Fig. 37. BER performances of the system with Manchester codes, RS (15, 11) codes, and convolutional codes with rate 1/3.

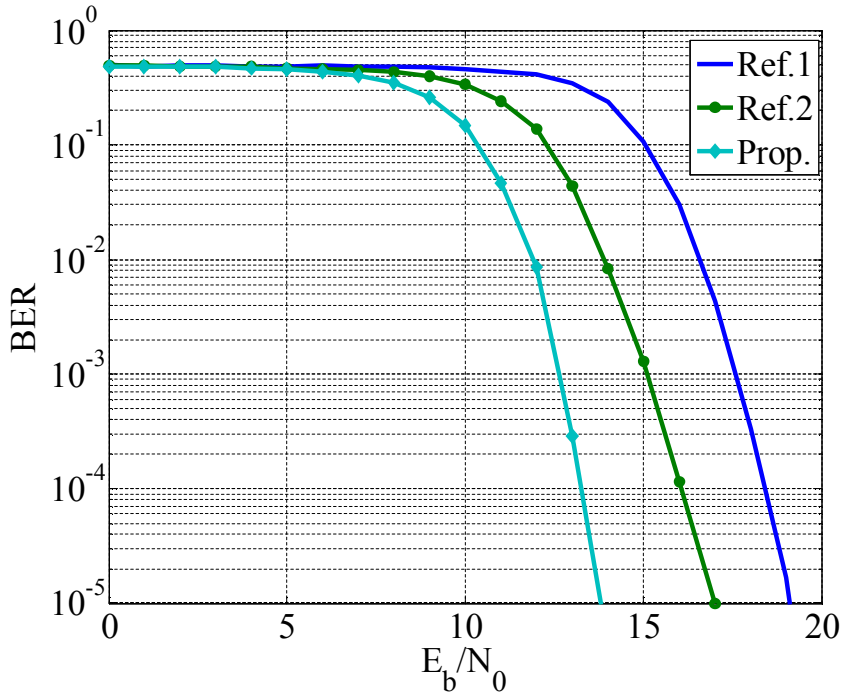

Fig. 38. BER performances of the system with 4B6B codes with RS (15, 7) codes, and convolutional codes with rate 1/4.

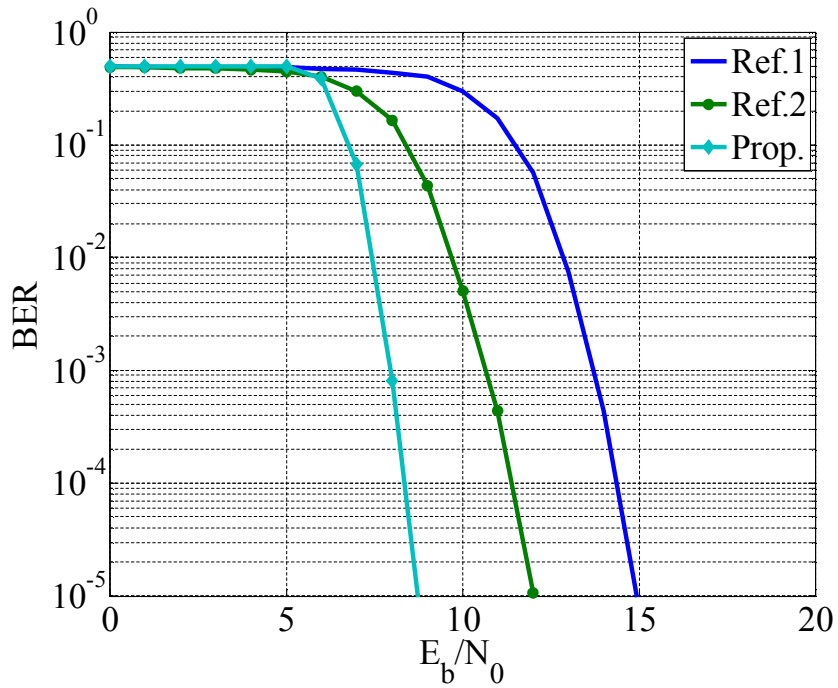

Fig. 39. BER performances of the system with Manchester codes, RS (31, 15) codes, and convolutional codes with rate 1/4.

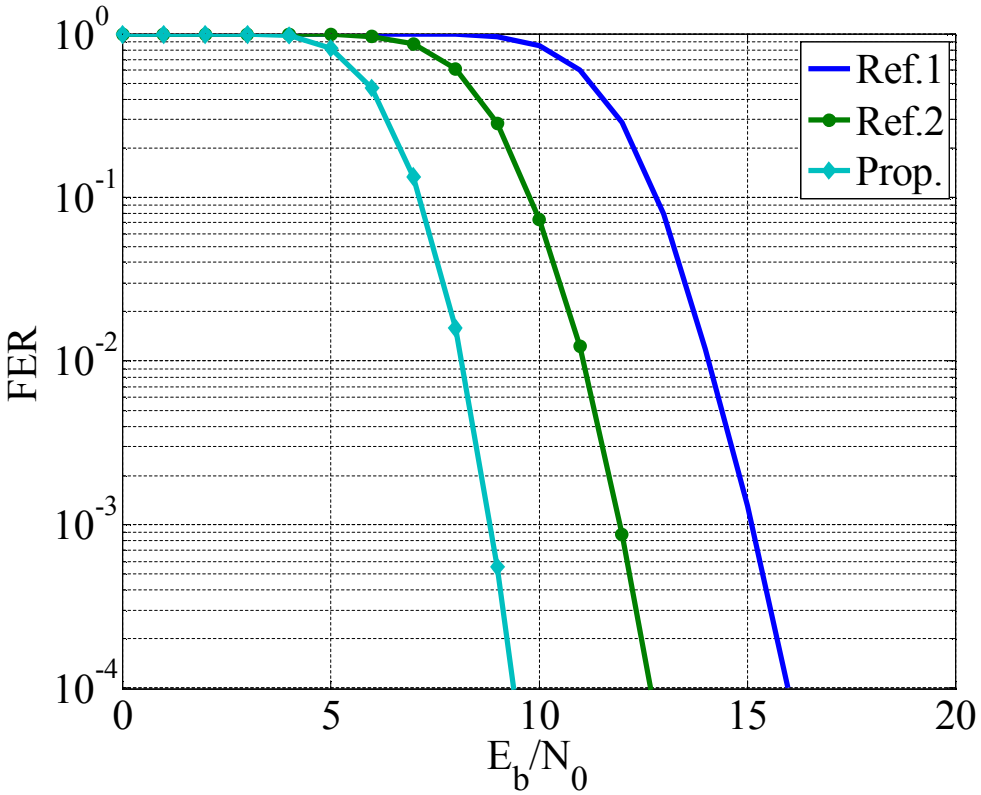

Fig. 40. FER performances of the system with Manchester codes, RS (15, 7) codes, and convolutional codes with 1/4.

In the BER results in Fig. 36 to Fig. 39, the performances of the proposed algorithm are better than all referenced algorithm. In comparison with Fig. 36 and Fig. 39, the BER performance in Fig. 37 is better than in Fig.36, since convolutional codes with rate 1/4 are punctured from convolutional codes with code rate 1/3. Comparing Fig. 36 and Fig. 38, which have the same RS codes and convolutional codes, Manchester codes have better performance than 4B6B codes, since codes with low rate have longer redundancy to correct errors. Comparing Fig. 36 and Fig. 39, the RS codes are nearly the same at 1/2, and they have similar performance. In these results, 'Prop.' shows the best performance.

In the FER results in Fig. 40, we observe that FER performances have tendency similar to BER performance under the same RS codes and convolutional codes in Fig. 36. Therefore, there is only one FER performances shown.

Table 10. The required SNR and gains at BER = 10^{-5}

		BER = 10^{-5}				FER=10^{-4}
	RS	(15,7)	(15,11)	(15,7)	(31, 15)	(15,7)
	CC	1/4	1/3	1/4	1/4	1/4
	RLL	Man.	Man.	4B6B	Man.	Man.
	Ref.1	16.01	13.78	19.12	14.75	16.00
	Ref.2	12.87	10.61	17.00	12.02	12.70
	Prop.	9.59	7.99	13.80	8.73	9.39
$Gain_A$	dB	6.42	5.79	5.32	6.02	6.50
	%	166.9	172.5	138.6	169.0	170.4
$Gain_B$	dB	3.28	2.62	3.20	3.29	3.72
	%	134.2	132.8	123.9	137.7	135.3

In Table 10, the exact BER and FER performances and corresponding SNR gains are listed for performance comparison. 'CC' and 'Man.' in Table 10 denote convolutional codes and the Manchester code used for RLL codes, respectively. '$Gain_A$' and '$Gain_B$' in Table 10 show gain when the proposed algorithm is 'Prop." and the referenced algorithms are 'Ref.1', 'Ref.2', respectively. The percentage gain is defined as

$$\frac{\text{dB of referenced algorithm}}{\text{dB of proposed algorithm}} \times 100\% .$$

From Table 10, '$Gain_A$' (the proposed algorithm) offers performance 5dB better than 'Ref.1', and the corresponding percentage gains are more than 130%. From Table 10, '$Gain_B$' (the proposed algorithm) offers performance about 2dB better than 'Ref.2', and the corresponding percentage gains are more than 120%. Therefore, our proposed algorithm is meaningful especially from a BER performance aspect.

6.7 Conclusion

A new soft convolutional codes decoding is proposed for VLC systems with concatenated RS codes and RLL codes. Comparing with general Viterbi algorithm, efficient soft convolutional decoding with modified BCJR algorithm is proposed, it decodes soft information from soft RLL decoding output, and the soft convolutional output can be deinterleaved and be used for soft RS decoding to improve performance. Soft RLL decoding is briefly reviewed and especially for Manchester codes, the soft RLL decoding process is simplified expressed. Simulation results based on the VLC standard show that the FER performances tend to be similar to BER performances. Moreover, BER performances of the proposed algorithm are better than those of referenced algorithms. Therefore, our proposed scheme can efficiently increase throughput of a VLC system.

CHAPTER 7
DIMMING CONTROL WITH POLAR AND RLL

7.1 Introduction

Polar codes, proposed by Arikan, were the first codes proven to achieve the Shannon capacity [33]-[34] for a binary-input discrete memoryless channel (B-DMC), in comparison with known turbo codes and LDPC codes. For B-DMCs, polar codes can split the channels to complete noisy channels and error-free channels through combination and polarization, and then, message bits can be transmitted through error-free channels, and frozen bits can be transmitted through complete noisy channels. Successive cancelation (SC) algorithm has generally been used for decoding of polar codes due to its low complexity. One construction method for polar codes in an AWGN channel was proposed [35]; the Bhattacharyya parameter was calculated to measure the channel's reliability based on binary phase-shift keying (BPSK) modulation. The tighter upper bound of scaling exponent was provided for any B-DMC, and polar codes were proved not affected by error floors [36]. Systematic polar codes [37] had better BER performance than the original polar codes [33]; even the FER performance is the same, but the reason for this phenomenon needs to be studied further [37]. Vangala et al. compared various constructions of polar codes in AWGN channel [38]; the constructions were based on Bhattacharyya bounds, Monte-Carlo estimation [33], transition probability matrices estimation [39], and Gaussian approximation [35] of the channel. Moreover, different constructions have the same performance if the designed SNR is optimized [38]. Therefore, these constructions are equivalent, in essence, and any one of them can be used in our proposed system for polar codes. It is also known that SC decoding have lower complexity than LDPC codes decoding [41].

VLC provides lighting and communication, simultaneously. Variable pulse-position modulation (VPPM) and OOK modulation are defined to combine with RLL codes in VLC standard [1], OOK modulation [1] is used in VLC due to its simplicity to send binary data easily by on/off pulses for single carrier modulation. Orthogonal frequency division multiplexing (OFDM) are potential

candidates for multiple carrier modulation [44].

Dimming can be controlled by adjusting the ratio of ONs to OFFs. Dimming range can be from 0% to 100% [42]. Conference rooms and restaurants can require light level as low as 1% of maximum illumination for aesthetic and comfort purposes [43]. It is desirable to maintain communication while a user arbitrarily dims the light source [1]. However, at extreme case the quality of communication link is too bad since signal energy can be used for only dimming. In the VLC standard [1], a dimming function can be realized by RLL codes and compensation symbols (CS). However, there is not efficient dimming system discussed in the VLC standard.

In recent research, efficient dimming systems with error control codes were discussed to enhance VLC performance. Kim and Jung [12] proposed a FEC coding method using modified RM codes to provide diming control; it could contain minimal CS to support multiple dimming target values. Moreover, modified RM codes composed of cosets made from bent functions were proposed to provide dimming control [13]. Kim [20] proposed adaptive FEC codes based on LDPC codes to adjust dimming values. Punctured methods in term of threshold of LDPC codes were proposed in [46]. Other research [16], [32], [40] showed that RLL codes are also used to enhance communication performance by using the soft information from the channel, especially when RLL codes are concatenated with other codes in VLC systems. Wang and Kim [16] proposed multiple output combined with a cyclic redundancy check (CRC) for RLL codes concatenated with Reed-Solomon (RS) codes. Wang and Kim proposed bit-level [32] and symbol-level [40] soft output RLL decoding.

In this chapter, we propose a dimming control algorithm composed of polar codes, RLL codes, puncturing, and CS in our VLC system. Since the proposed algorithm is coding-based dimming scheme [45], OOK modulation is considered for its simplicity, but efficient modulation can be used instead of OOK. Efficient construction is from soft RLL codes providing dimming 50% and likelihood ratio information for polar decoding; CS can be used to provide dimming values that are not 50%. The simulation results show that our proposed system has better performance than referenced systems, and the proposed polar coding systems can be used to improve development of the VLC industry.

7.2 Proposed VLC system

A block diagram of the proposed VLC system is shown in Fig. 41. The transmitter shown in Fig. 41 (a) comprises the polar encoder, RLL encoder, dimming control, and OOK modulated light. The dimming control comprises puncturing and CS insertion. The receiver shown in Fig. 41 (b) comprises the polar decoder, RLL decoder, and OOK demodulated photodiode (PD). It is assumed that dimming value d is determined by the upper layer and is provided to both transmitter and receiver.

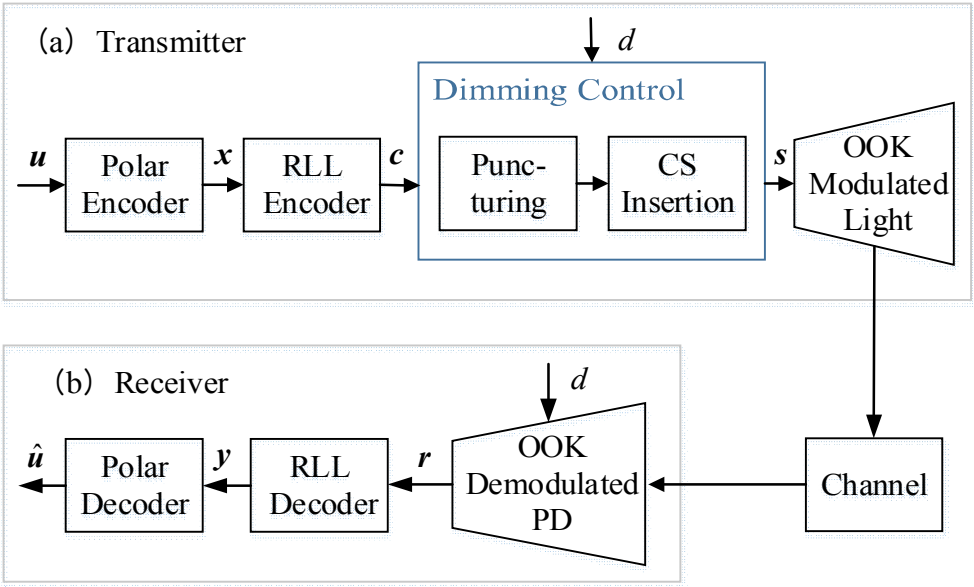

Fig. 41. Block diagrams of the proposed transmitter and receiver in a VLC system.

In Fig. 41 (a), $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ is the input of the polar encoder, where k of n bits are message bits, and the other $(n\text{-}k)$ bits are frozen bits. The output of the polar encoder is $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, where $n = 2^h$ and h is an integer. The code rate of the polar codes is $\gamma_1 = k/n$. Then, $\boldsymbol{x}$ is RLL encoded to $\boldsymbol{c} = (c_1, c_2, \ldots, c_q)$, and the RLL code rate is $\gamma_2 = n/q$. According to dimming d, $\boldsymbol{s} = (s_1, s_2, \ldots, s_l)$ is OOK-modulated and sent to the receiver.

In Fig. 41 (b), $\boldsymbol{r} = (r_1, r_2, \ldots, r_l)$ is a received OOK-modulated signal, and can be written as $\boldsymbol{r} = \boldsymbol{s} + \boldsymbol{n}$, where $\boldsymbol{n}$ is AWGN with variance σ^2 [11]-[13]. After RLL decoding, $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ is the input of the polar decoder, and finally $\hat{\boldsymbol{u}} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_n)$ is the estimated value.

This chapter focus on dimming control for concatenated polar codes and RLL codes with assist of puncture and CS, and a detailed description of our algorithm is given.

7.3 Review of polar codes

The construction of polar codes is different from traditional linear block codes, since the generator design is not base on maximizing the minimum hamming distance, but on minimizing the error probability of the information-bearing polarized channels [47].

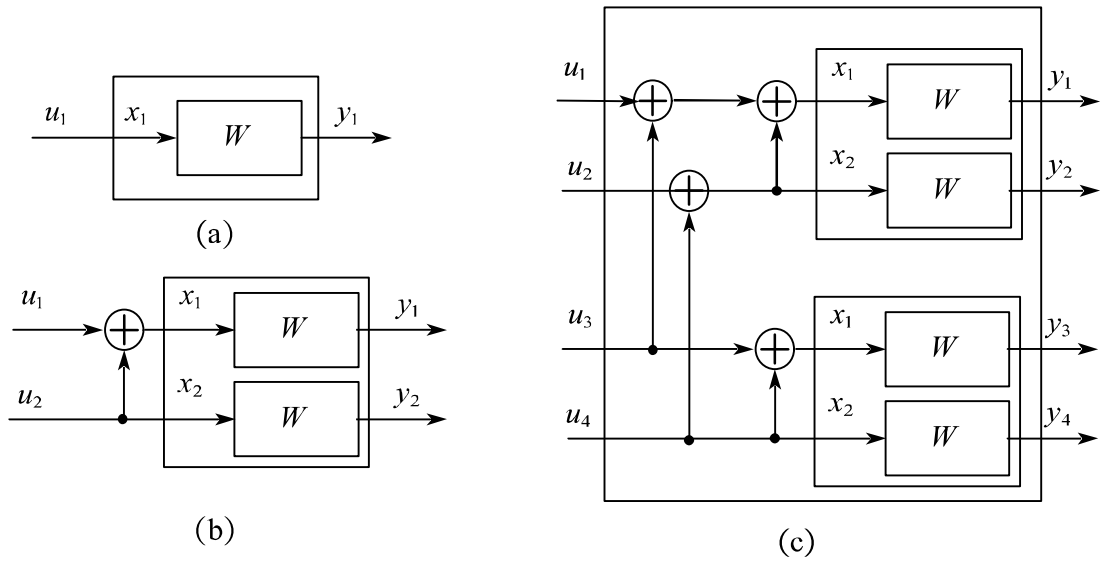

Fig. 42. Channel polarization example for (a) n=1, (b) n=2, (c) n=4.

To combine and split (polarize) channels, Bhattacharyya parameter $Z(W)$ should first be calculated for channel W. Bhattacharyya parameter $Z(W)$ is defined as [1]-[4]

$$Z(W) \triangleq \sum_{y \in Y} \sqrt{p(y|0)\,p(y|1)} \tag{7.1}$$

where $W(y|0)$ and $W(y|1)$ represent the conditional probability for a binary symmetric channel with output y. Arikan [33] used a Bhattacharyya parameter to determine error-free channels and complete noisy channels for message bits and frozen bits. It has been shown that channel capacity $C(W) \approx 1$ iff $Z(W) \approx 0$, and $C(W) \approx 0$ iff $Z(W) \approx 1$ [33]. We can recursively to estimate channel reliability base on

$$Z(W^{+}) = Z(W)^{2} \tag{7.2}$$

$$Z(W^{-}) \leqslant 2Z(W) - Z(W)^{2} \tag{7.3}$$

Equality holds in (7.3) iff W is binary erasure channel (BEC). For example, for BEC with erasure probability 0.5, if n=2 as Fig. 42 (b), then $Z(W^{+}) = 1/4$, $Z(W^{1}) = 3/4$, then capacity $C(W^{+}) = 1/4$, $C(W^{1}) = 3/4$. Similarly, the channel capacity for larger value n can be calculated as Fig. 43.

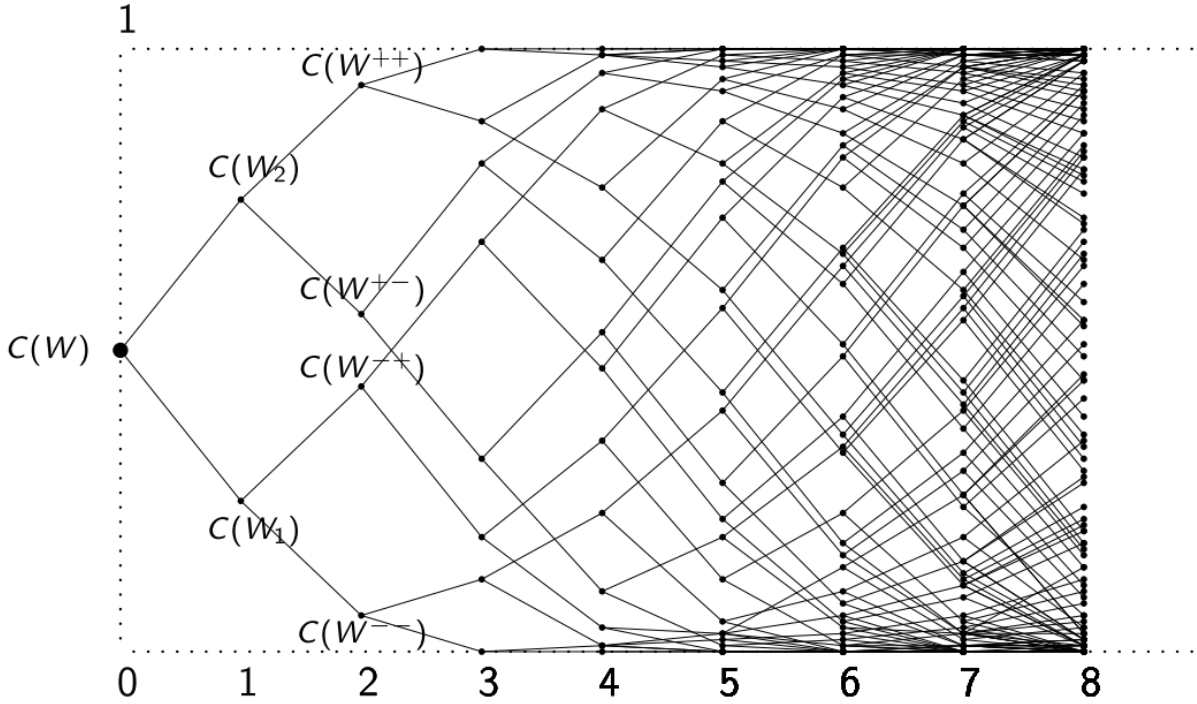

Fig. 43. Channel capacity for $h = \log(n) = 0, 1, 2, \ldots 8$.

7.3 Encoding Processor

Polar code construction includes combing and splitting. The channels are divided into error-free channels and complete noisy channels. To measure and divide channels, Bhattacharyya parameter $Z(W)$ should first be calculated for channel W. Bhattacharyya parameter $Z(W)$ is defined (7.1). Over AWGN channel, $Z(W)$ can be computed based on OOK modulation as [35]

$$\begin{aligned} Z(W) &= \int_{-\infty}^{\infty} \sqrt{p(y|0)\,p(y|1)}\,dy \\ &= \int_{-\infty}^{\infty} \sqrt{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-0)^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-1)^2}{2\sigma^2}}}\,dy \\ &= e^{-\frac{1}{8\sigma^2}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-1/2)^2}{2\sigma^2}}\,dy \\ &= e^{-\frac{1}{8\sigma^2}} \end{aligned} \tag{7.4}$$

The generator matrix is $G = CF^{\otimes h}$, where $F = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, $\otimes$ denotes the Kronecker product, and C is an $n \times n$ bit-reversal permutation matrix. The indices of information bits and frozen bits are defined as A and A^c, respectively. Then [5]

$$\boldsymbol{x} = \boldsymbol{u}_A G_A + \boldsymbol{u}_{A^c} G_{A^c} . \tag{7.5}$$

Systematic polar codes were introduced by Arikan [37] to make enhanced polar codes. From an index B, a systematic codeword where $\boldsymbol{x} = (\boldsymbol{x}_B, \boldsymbol{x}_{B^c})$ can be expressed as [37]

$$\boldsymbol{x}_B = \boldsymbol{u}_A G_{AB} + \boldsymbol{u}_{A^c} G_{A^c B}, \quad \boldsymbol{x}_{B^c} = \boldsymbol{u}_A G_{AB^c} + \boldsymbol{u}_{A^c} G_{A^c B^c} \tag{7.6}$$

where $\boldsymbol{u}_A$ and $\boldsymbol{u}_{A^c}$ are equivalent message and parity for nonsystematic code, respectively. Then, equivalent message [5] is given as

$$\boldsymbol{u}_A = \left(\boldsymbol{x}_B - \boldsymbol{u}_{A^c} G_{A^c B}\right)\left(G_{AB}\right)^{-1} . \tag{7.7}$$

RLL codes such as Manchester codes, 4B6B and 8B10B in VLC standard [9] are considered for RLL encoder and decoder.

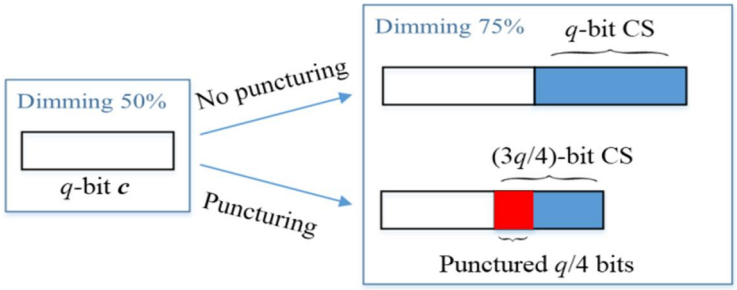

Fig. 44. Example of CS length for dimming 75%, based on puncturing.

Let us first consider a simple case that RLL codes and CS are used. If dimming value d = 50%, the length of CS is 0 and $\boldsymbol{s} = \boldsymbol{c}$, since RLL codes provide 50% dimming. The length of $\boldsymbol{c}$ can be calculated as $q = n/\gamma_2$ for a given RLL code rate and there are $q/2$ ones and $q/2$ zeros in $\boldsymbol{c}$. For various dimming d, $\boldsymbol{s}$ with length l is constructed to satisfy the conditions where the number of 1's divided into l equals d. If d is higher than 50%, l is determined from $(q/2 + l - q)/l = d$ and new CS with $(l - q)$ ones are inserted in $\boldsymbol{s}$. For example, if d = 75%, $(q/2 + l - q)/l$ = 75%, and we can get $l = 2q$. In this case, the length of the CS is $l - q = q$. If d is lower than 50%, CS are composed of all 0s and CS length is $(l - q)$ which is same with the CS length for dimming value 1-d.

Next, consider puncturing for dimming control. If it is assumed that the punctured length is v, then $((q-v)/2 + v + l - q)/l = d$. For example, if d = 75% and $v=q/4$, then $l=3/2q$, and CS length is $l-(q-v)$ $=3/4q$. Main advantage of the puncturing method is that the CS length is shorter than one with no puncturing scheme. For polar code encoding, the channels are divided into two parts, where in one part the channel capacity is near 1, and in the other, channel capacity is near 0. To enhance coding efficiency, the punctured bits are assigned to a channel with capacity 0.

7.4 Decoding processor

RLL decoding adopts a soft RLL decoding algorithm [32], where output likelihood ratio (LR) information is input for the polar decoder. For Manchester codes, the likelihood is [32]

$$p(r_i|s_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(r_i - s_i)^2}{2\sigma^2}}, \tag{7.8}$$

where s_i is 0 or 1. Then, the conditional probability in (1) is given as

$$\begin{aligned} W(y_i|x_i = 0) &= p(r_j|s_j = 0)\, p(r_{j+1}|s_{j+1} = 1) \\ &= \frac{1}{2\pi\sigma^2} e^{-\frac{r_j^2 + (r_{j+1} - 1)^2}{2\sigma^2}} \end{aligned}, \tag{7.9}$$

since $x_i = 0$ is encoded to $s_j = 0$, and $s_{j+1} = 1$, where $j = 2i - 1$. Similarly, probability $W(y_i \mid x_i = 1)$ is given as

$$\begin{aligned} W(y_i|x_i = 1) &= p(r_j|s_j = 1)\, p(r_{j+1}|s_{j+1} = 0) \\ &= \frac{1}{2\pi\sigma^2} e^{-\frac{(r_j - 1)^2 + r_{j+1}^2}{2\sigma^2}} \end{aligned}. \tag{7.10}$$

Polar decoding uses an SC decoding algorithm [33], which has low complexity. Estimation bits $\hat{u}_1^n = \hat{\boldsymbol{u}} = (\hat{u}_1, \hat{u}_{2,\ldots,} \hat{u}_n)$ can be calculated based on recursive formulas [33]

$$\begin{aligned} & L_n^{(2i-1)}\left(y_1^n, \hat{u}_1^{2i-2}\right) \\ &= \frac{L_{n/2}^{(i)}\left(y_1^{n/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) L_{n/2}^{(i)}\left(y_{n/2+1}^n, \hat{u}_{1,e}^{2i-2}\right) + 1}{L_{n/2}^{(i)}\left(y_1^{n/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right) + L_{n/2}^{(i)}\left(y_{n/2+1}^n, \hat{u}_{1,e}^{2i-2}\right)} \end{aligned} \tag{7.11}$$

$$\begin{aligned} & L_n^{(2i)}\left(y_1^n, \hat{u}_1^{2i-1}\right) \\ &= \left[L_{n/2}^{(i)}\left(y_1^{n/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}\right)\right]^{1-2\hat{u}_{2i-1}} \cdot L_{n/2}^{(i)}\left(y_{n/2+1}^n, \hat{u}_{1,e}^{2i-2}\right) \end{aligned}, \tag{7.12}$$

where subscript o and e indicate odd index and even index, respectively. The calculation of an LR at length n can be reduced to length $n/2$, and the recursion will stop at length 1, where $L_1^{(1)}(y_i) \triangleq W(y_i|0)/W(y_i|1)$ can be calculated from soft RLL output in (7.11) and (7.12). The message bits can be

determined by hard decision (output 0 when LR>0, otherwise output 0) that depends on the LR, and frozen bits are zeros since we set zeros as frozen bits during encoding.

7.5 Simulation result

Four representative dimming values (50%, 62.5%, 75%, and 87.5%) were considered to examine the performance of the proposed system. These dimming values were considered, and different dimming values can be obtained by adaptive CS and puncturing. We assumed an AWGN channel with variance σ^2 in VLC, and Bhattacharyya bounds to construct the polarized channels. Polar codes with h=10 and code rate 1/2 were considered, where n = 1024, k = 512. We assumed that Manchester codes are used for the RLL encoder.

Simulation results corresponding to 50% dimming, 62.5% dimming, 75% dimming, and 87.5% dimming are shown in Fig. 45-48, respectively. The lines Non-sys, Sys, Non-sys-p, and Sys-p represent non-systematic code, systematic code, non-systematic code with puncture, and systematic code with puncture, respectively. Ref. RS indicates that RS (64, 32) codes in VLC dimming [10]; Ref. LDPC means LDPC codes in VLC dimming [13]. RS (64, 32) codes are chosen among codes in VLC standard [10] since they have same code rate with our proposed polar codes.

For a fair comparison, the energy of the message bits E_b is fixed as E_b =1, and the energy of codeword bit E_c should contain the CS energy according to the code rates. For example, consider 75% dimming and polar codes with rate γ_1 =1/2, and RLL codes with length q rate γ_2 = 1/2. Then, the CS length is designated q. E_c is calculated as follows:

$$E_c = E_b \times \gamma_1 \times \gamma_2 \times \frac{50\%}{d} = E_b \times 1/2 \times 1/2 \times \frac{50\%}{75\%} = E_b \times 1/6 .$$

For dimming d less than 50%, since CS is composed of 0s, E_c for dimming d is larger than E_c for dimming 1-d. To investigate puncturing effect, half the frozen bits are punctured. Table 11 shows CS length and energy relations for no punctured and half punctured frozen bits under different dimming values.

Table 11. CS length and energy relation for the proposed system

d	CS length		Energy relation	
	No puncture	Puncture	No puncture	Puncture
50%	0	0	$E_c = E_b \times 1/4$	$E_c = E_b \times 1/4$
62.5%	683	512	$E_c = E_b \times 1/5$	$E_c = E_b \times 4/15$
75%	2048	1536	$E_c = E_b \times 1/6$	$E_c = E_b \times 2/9$
87.5%	6144	4608	$E_c = E_b \times 1/7$	$E_c = E_b \times 4/21$

For 50% dimming, puncturing is not necessary since usage of RLL codes provide 50% dimming. As dimming increase, the energy of codeword bit E_c becomes smaller since more CS insertion is required. However, since punctured scheme requires smaller CS length than no punctured scheme, bit energy of the codes of punctured scheme is relatively larger.

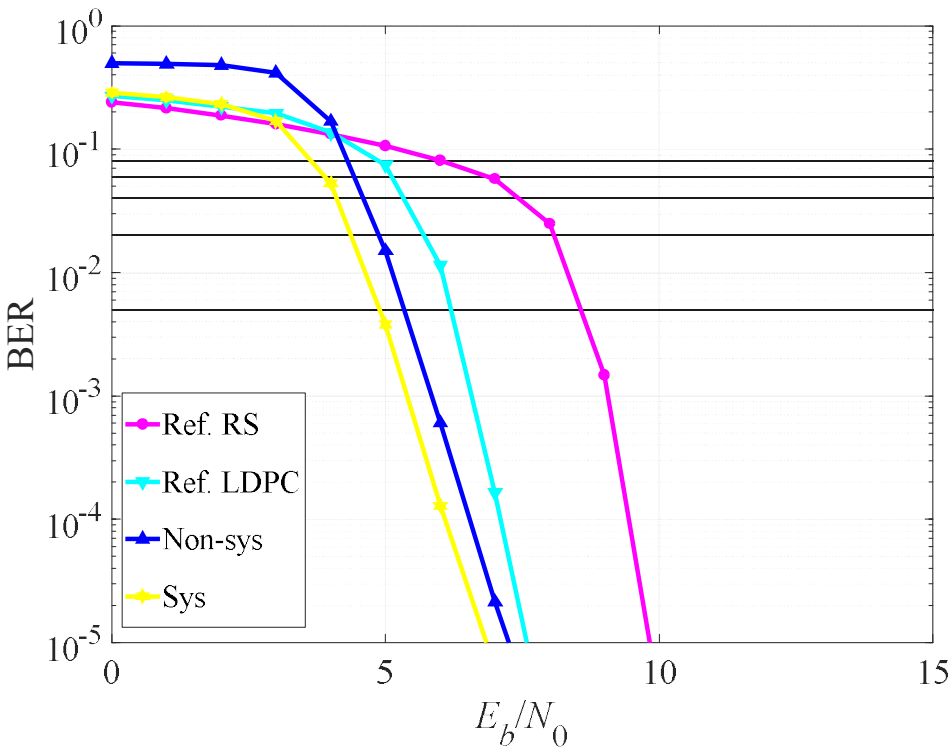

Fig. 45. BER performance of the proposed system for 50% dimming.

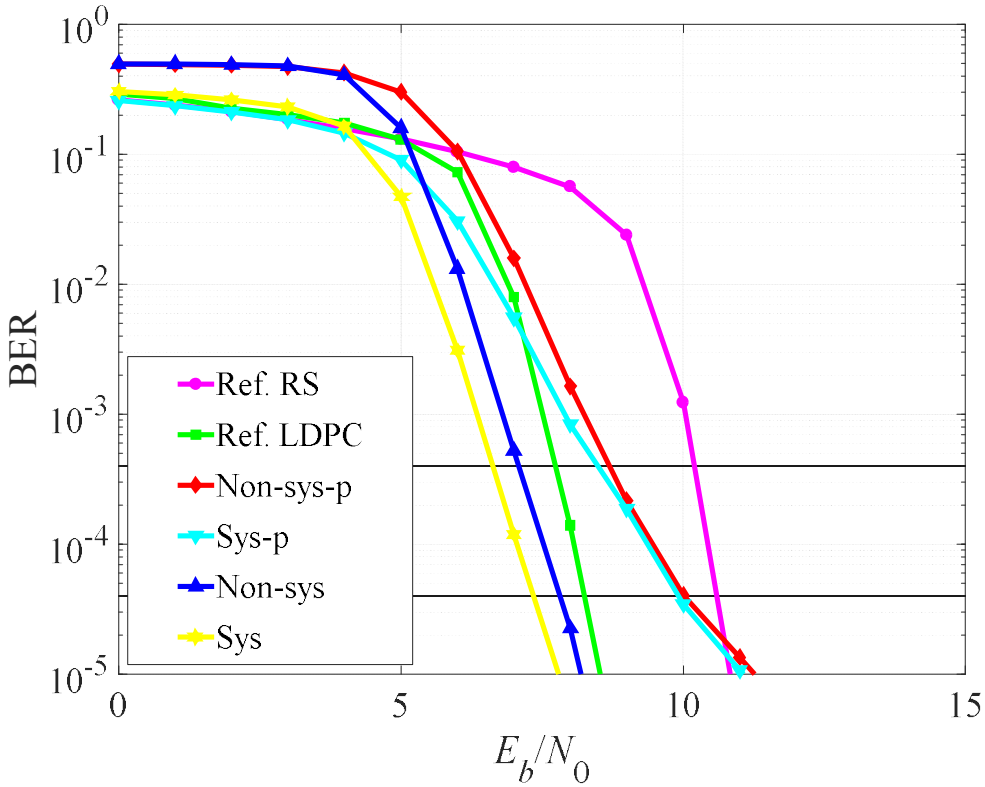

Fig. 46. BER performance of the proposed system for 62.5% dimming.

In Fig. 45, puncturing is not necessary for 50% dimming and CS is used in Fig. 46-48. Fig. 45 shows that systematic codes are better than non-systematic codes for 50% dimming, and this trend is also shown for all other dimming values. In Fig. 46-48, the codes with puncture show an error floor trend for a larger SNR = E_b/N_0, since LLRs of punctured bits are 0 and minimum distances of punctured codes are not larger than ones of original codes. The error floor effects by puncturing codeword bits are similar with the results of the punctured LDPC codes [26].

From Fig. 45-48, the exact BER values at 10^{-5} are shown in Table 12. Gain_1 shows the difference between Sys and Ref. 1, and Gain_2 shows the difference between Sys and Ref. 2.

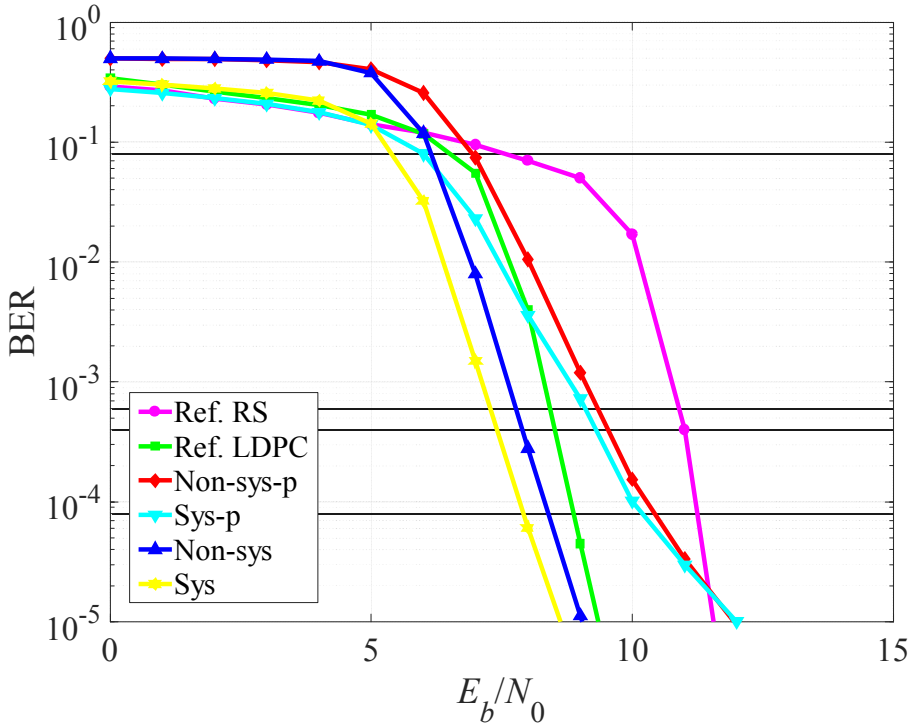

Fig. 47. BER performance of the proposed system for 75% dimming.

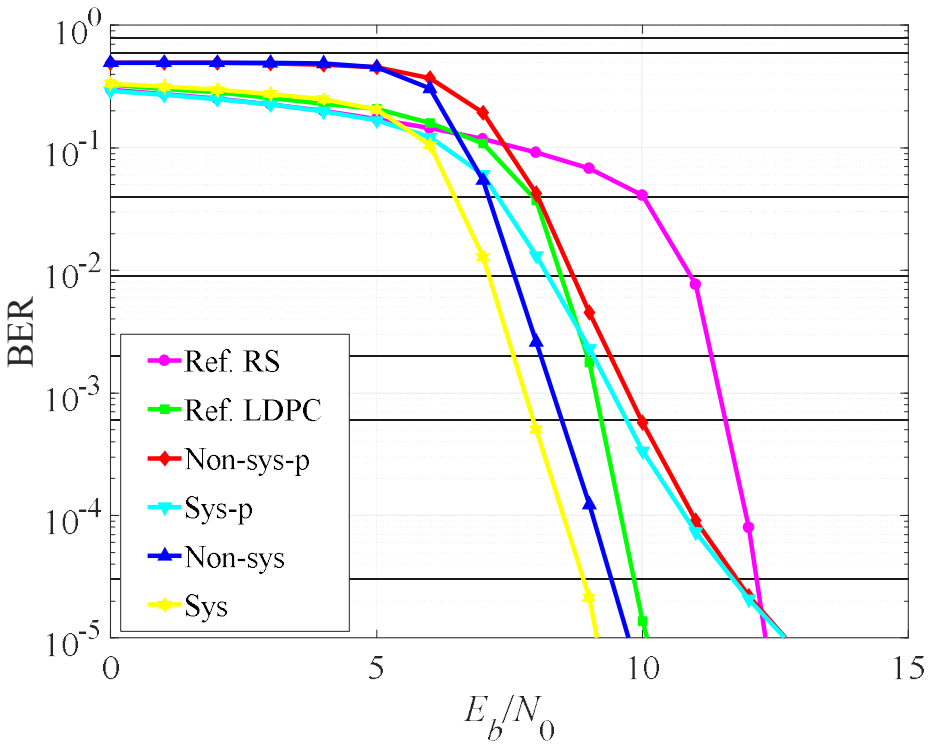

Fig. 48. BER performance of the proposed system for 87.5% dimming.

Table 12. The required SNR and gains at BER = 10^{-5}

Dimming	50%	62.5%	75%	87.5%
Ref. 1 [10]	9.83	10.83	11.55	12.31
Ref. 2 [13]	7.58	8.55	9.35	10.08
Non-sys-p	—	11.16	11.76	12.69
Sys-p	—	11.05	12.19	12.67
Non-sys	7.25	8.22	9.03	9.73
Sys	6.84	7.99	8.62	9.14
$Gain_1$	2.99	2.84	2.93	3.17
$Gain_2$	0.74	0.56	0.73	0.94

In Table 12, the notation — means there is no value for corresponding parameter in the figures. Our proposed algorithm has gains which are over 2.5dB and 0.5dB than Ref. 1 [1] and Ref. 2 [20], respectively.

7.6 Conclusion

In this chapter, polar codes combined with RLL codes in VLC was proposed to control dimming based on puncture and CS. In general, RLL codes are used to keep the number of ones and the number of zeros the same, which provides 50% dimming. However, soft decoding RLL can be efficiently combined with polar codes, which easily use an LR obtained from RLL output. Furthermore, when the dimming value is not 50%, CS can be used to control dimming. The simulation results show that our proposed algorithm has better performance than referenced systems that use LDPC codes or RS codes for dimming control in VLC.

CHAPTER 8
CONCLUSION AND FUTURE WORK

8.1 Conclusion of dimming control

RLL codes are considered for 50% dimming. First, ML decoding of RLL codes with an OOK modulation and RS codes in VLC has been proposed in Chapter 2. As the new receiver architecture, the proposed RLL decoder makes better output based on ML rule which making use of soft value from a VLC channel. From simulation results, the proposed ML decoding of RLL codes shows better performance than conventional RLL codes.

Then a modified RLL decoding which output multiple candidates in VLC has been proposed in Chapter 3. We just modify the receiver architecture, the proposed RLL decoder produces multiple candidates composed of symbols with higher probabilities calculated from the received signal, and the proposed selector selects the decision message according to the CRCs of multiple candidates. The proposed scheme performs better than RLL hard decoding; the performance also improves further with a larger number of candidates. The main advantage of our proposed scheme is that it can lead to performance enhancement without a change in the transmitter in VLC.

Then, SLS RLL decoding algorithm was proposed in Chapter 4, since previous ML decoding belong to hard output RLL decoding (even with multiple candidates), the soft output of RLL decoding can provide more information for ECC, here consider RS codes. Moreover, conventional RS codes adopt BM algorithm which use hard input. Therefore, SLS RLL are concatenated with RS codes which use geometric decoding algorithm. Simulation results show that that the BER performance of SLS RLL is better than ML RLL decoding.

Next, BLS RLL decoding algorithm was proposed in Chapter 5, which has more widely applications, it can be concatenated with powerful ECC, such as low-density parity-check (LDPC) codes, turbo codes, and polar codes. However, we first consider RS codes with iterative decoding, in order to compare with ML RLL, and SLS RLL which are also concatenated with RS codes.

Simulation result show that proposed BLS RLL has FER performance, which is advantage for application to hybrid ARQ.

Next, modified BCJR algorithm is considered to connect with BLS RLL decoding in Chapter 6. Viterbi algorithm is generally used for convolutional codes decoding which is hard output decoding, however, the modified BCJR can make soft output. The simulation result show that modified BCJR has better performance once convolutional codes are considered.

Finally, dimming control (with various dimming value) algorithm was proposed in Chapter 7. RLL codes are used to provide 50% dimming, other dimming value can be obtained by using compensation symbol (or puncturing). The state of art polar codes, which is the first codes proven to achieve the Shannon capacity, are considered. The reference considers previous result about LDPC codes and RS codes. Simulation results show that our proposed algorithm has best performance to compare with referenced ones.

8.2 Future work

The state-of-art polar codes were considered in Chapter 7 to concatenated with BLS RLL, and using CS to provide various dimming values. However, polar codes have their own properties and construction, such as polar codes selves can achieve nearly 50% dimming [48], joint design should be considered once concatenated with RLL codes. Moreover, simple puncturing was discussed in Chapter 7 which based on an original frozen index of polarization, is not optimal, more efficient puncturing can be considered with joint RLL consideration. A novel puncturing method for polar codes was proposed in [49], which used frozen bits to construct punctured bits that would be known on the receiver side, and this puncturing of polar codes was used to improve code efficiency with adaptive codeword lengths. Therefore, to joint design punctured polar codes and RLL codes can be one possible research direction.

In VLC development, dimming control and interference is important issue which need to solve, we have considered dimming in this thesis, such as BLS RLL and SLS RLL for 50% dimming, and concatenated polar codes for various dimming value. However, referring to the interference, the related work is not enough. Rahaim and Little [50] discussed Gaussian interference and non-Gaussian distribute interference, and different modulations were simulated in the interference channel, especially different types of OFDM. The effect of field of view (FOV) for interference was analyzed [51]. Due to position of LEDs, reflection of wall, bit interval and so on, Intersymbol interference (ISI) can be main problem in VLC. ISI is a phenomenon where the same signal is transmitted through multipath in a VLC system. In general, methods to alleviate ISI include equalization and error correction code, but our research focuses on the error correction code aspect. Polar codes were originally intended for a memoryless channel. The polarization was main step in polar codes construction, which will be affected by memory channel when consider ISI, frozen bit index and message bit index could be exchanged with the influence of memory, which will decrease performance. However, polarization procession with memory was proposed in [52]. It considered polar code with memory underlying finite state Markov chain, where ISI also belong to finite state

channel, therefore, polar codes used in memory channel are feasible. Therefore, the inference can one possible research direction need to study with polar codes application.

[REFERENCE]

[1] IEEE Std 802.15.7, "IEEE standard for local and metropolitan area networks-part 15.7: short-range wireless optical communication using visible light," *IEEE Computer Society*, pp. 248-271, 2011.

[2] T. Komine, M. Nakagawa, "Fundamental analysis for visible-light communication system using LED lights," *IEEE Transaction on Consumer Electronics,* vol. 50, no. 1, pp. 100-107, pp. 100-107, Feb. 2004.

[3] T. Komine, J. H. Lee, S. Haruyama, and M. Nakagawa, "Adaptive equalization system for visible light wireless communication utilizing multiple white LED lighting equipment," *IEEE Transaction on Wireless Communications*, vol. 8, no. 6, pp. 2892-2900, Jun. 2009.

[4] V.C. Yanez, J.R. Torres and J.B. Alonso et al, "Illumination interference reduction system for VLC Communication," in *Proc. the 11th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems*, pp. 252-257, 2009.

[5] N. Kim, "Decision feedback equalization for indoor visual light communication," *International Journal of Circuits, System and Signal Processing*, vol. 8, pp. 47-53, 2014,.

[6] B. Fong, P.B. Rapajic, G.Y. Hong and A.C.M. Fong, "Forward error correction with Reed-Solomon codes for wireless ATM networks," *WSEAS International Conference on NNA-FSFS-EC*, Feb. 2002.

[7] L. Chaari, M. Fourati, N. Masmoudi and L. Kamoun, "A reconfigurable FEC system based on Reed-Solomon codes for DVB and 802.16 network," *WSEAS Transactions on Circuits and Systems*, vol. 8, no. 8, pp.729-744, 2009.

[8] M.A. Armand, "Efficient decoding of Reed Solomon codes over Z_q based on remainder polynomial," *WSEAS International Conference on NNA-FSFS-EC*, Feb. 2002.

[9] T.H. Hu, M.H. Chang, "Decoding shortened Reed Solomon codes at bit level," *WSEAS Transaction on Communication*, vol. 9, no. 11, pp. 695-707, 2010,.

[10] J.G. Proakis and M. Salehi, "Digital Communications", *McGraw-Hil*, pp.471-475, 2008.

[11] K. Lee and H. Park, "Modulations for visible light communications with dimming control," *IEEE Photonics Technology Letters*, vol. 23, no. 16, pp. 1136-1138, Aug. 2011.

[12] S. Kim and S.-Y. jung, "Novel FEC coding scheme for dimmable visible light communication based on the modified Reed-Muller codes", *IEEE Photonics Technology Letters,* vol.23, no.20, pp. 1514-1516, 2011.

[13] S. Kim and S.-Y. jung, "Modified Reed-Muller coding scheme made from the bent function for dimmable visible light communication", *IEEE Photonics Technology Letters,* vol. 25, no.1, pp. 11-13, 2013.

[14] A. Mengi. and A.J.H. Vinck, "Code frequency modulation with RLL sequences for CENELEC band applications," *IEEE International Symposium on Power Line Communication and Its Applications*, pp. 133-137, 2007.

[15] S. Rajagopal, R.D. Roberts and S.-K. Lim, "IEEE 802.15.7 visible light communication: modulation schemes and dimming support," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 72-82, Mar. 2012.

[16] H. Wang and S. Kim, "New RLL decoding algorithm for multiple candidates in visible light communication," *IEEE Photon. Technol. Lett.*, vol. 27, no. 1, pp. 15–17, Jan. 2015

[17] T. K. Moon, *Error correction coding*, Wiley-Interscience, 2004, pp. 242-243.

[18] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[19] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, Sept. 1999.

[20] S. Kim, "Adaptive FEC codes suitable for variable dimming values in visible light communication," *IEEE Photon. Technol. Lett.*, vol. 27, no. 9, pp. 967–969, May 2015.

[21] J. Jiang, K.R. Narayanan, "Iterative soft-input soft-output decoding of Reed-Solomon codes by adapting the parity check matrix," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3746–3756, Aug. 2006.

[22] F.R. Kschischang, B.J. Frey and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 498–519, Feb. 2001.

[23] G. D. Forney and D. J. Costello, "Channel coding: The road to channel capacity," *Proc. IEEE*, vol. 95, no. 6, pp. 1150–1177, 2007.

[24] Ma, X., Lee, K., Lee, K.: 'Appropriate modulation scheme for visible light communication systems considering illumination', *Electro. Lett.*, vol. 48, no. 18, pp. 1137-1139, 2012

[25] Jang, H.-J., Choi, J.-H., Cho, E.B., Lee, C.G.: 'Simulation of a VLC system with 1Mb/s NRZ-OOK data with dimming signal', *Int. Conf. on Advan. Infoco. Techno. (ICAIT)*, 2011, pp. 28-28

[26] Yoo, J.-H., Kim, B.W., Jung, S.-Y.: 'Modelling and analysis of M-ary variable pulse position modulation for visible light communications', *IET Optoelectro.*, vol. 9, no. 5, pp.184-190, 2015

[27] Kim, J.K., Hyun, K., Park, S.K., 'Adaptive predistorter using NLMS algorithm for nonlinear compensation in visible-light communication system', *Electro. Lett.*, vol. 50, no. 20, pp. 1457-1459, 2014

[28] Huang, X., Fu, X., Xu, W.: 'Incremental scheduling scheme for indoor visible light communication', *Electro. Lett.*, vol. 51, no. 3, pp. 317-324, 2015

[29] Zhang, T., Guo, S., Chen, H., Zhong, F., Ma, C.: 'Enhancing the bit error rate of indoor visible light communication systems using adaptive channel estimation algorithm', *IET Commun.*, vol. 9, no. 4, pp.501-507, 2015

[30] Moon, T. K.: 'Error correction coding' (Wiley-Interscience, 2004) pp. 475–481

[31] Proakis, J. G., Salehi, M.: 'Digital Communications' (McGraw-Hill, New York, 2008, 5th edn.), pp. 471–548.

[32] H. Wang and S. Kim, "Bit-level soft run-length limited decoding algorithm for visible light communication," *IEEE Photon. Technol. Lett.*, vol. 28, no. 3, pp. 237–240, Feb. 2016.

[33] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[34] E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes,” in *Proc. IEEE Int. Symp. On Inf. Theory*, Toronto, ON, Canada, 2008, pp. 1173–1177.

[35] H. Li and J.Yuan, “A practical construction method for polar codes in AWGN channels,” in *Proc. IEEE TENCON Spring Conf.*, Sydney, NSW, Australia, 2013, pp. 223–227.

[36] M. Mondelli, S. H. Hassani, and R. L. Urbanke, “Unified scaling of polar codes: Error exponent, scaling exponent, moderate deviations, and error floors,” *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6698–6712, Dec. 2016.

[37] E. Arikan, “Systematic polar coding,” *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 860–862, Aug. 2011.

[38] H. Vangala, E. Viterbo, and Y. Hong, “A Comparative Study of Polar Code Constructions for the AWGN Channel,” *arXiv1501.02473v1 [cs, IT]*, pp. 1–9, Jan. 2015.

[39] I. Tal and A. Vardy, “How to construct polar codes,” *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.

[40] H. Wang and S. Kim, “Soft-input soft-output run-length limited decoding for visible light communication,” *IEEE Photon. Technol. Lett.*, vol. 28, no. 3, pp. 225–228, Feb. 2016.

[41] J. Li, Y. Guo, and M. H. Lee, “Iterative decoding algorithm in the Turbo, LDPC, Polar codes,” in *12th Inter. Conf. on Comm. Tech. Proce., ICCT*, pp. 1453–1456, Nov. 2010.

[42] B. Bai, Z. Xu, and Y. Fan, “Joint LED dimming and high capacity visible light communication by overlapping PPM,” *Wire. And Optic. Comm. Conf.*, *WOCC*, May 2010.

[43] J. Gancarz, H. Elgala, and T. D. C. Little, “Impact of lighting requirements on VLC systems,” *IEEE Commun. Mag.,* vol. 51, no. 12, pp. 34–41, Dec. 2013.

[44] M. S. Islim, H. Hass, “Modulation techniques for LiFi,” *ZET Comm.*, vol. 14, no. 2, pp. 29–40, Apr. 2016.

[45] F. Zafar, D. Karunatilake, and P. Rajendran, “Dimming schemes for visible light communication: the state of research,” *IEEE Wirel. Commun.*, vol. 22, no. 2, pp. 29–35, Apr. 2015.

[46] J. Ha, J. Kim, and S. W. Mclaughlin, “Rate-compatible puncturing of low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.

[47] K. Niu, K. Chen, J. Lin, and Q. T. Zhang, “Polar codes: Primary concepts and practical decoding algorithms,” *IEEE Commun. Mag.*, vol. 52, no. 7, Jul. 2014.

[48] J. Fang *et al.*, “An efficient flicker-Free FEC coding scheme for dimmable visible light communication based on polar codes,” *IEEE Photonics J.*, vol. 9, no. 3, 2017.

[49] R. Wang and R. Liu, “A novel puncturing scheme for polar codes,” *IEEE Commun. Lett.*, vol. 18, no. 12, pp. 2081–2084, 2014.

[50] M. Rahaim and T. D. C. Little, “Optical interference analysis in visible light communication networks,” in *Proc. IEEE Int. Conf. Commun. Work. (ICCW)*, London, UK, Jun. 2015, pp. 1410–1415.

[51] J. H. Lee, D. W. Song, and S. Y. Jung, “Interference analysis of visible light communication based on PD Array,” in *Proc. Int. Conf. Electron. Inf. Commun. (ICEIC)*, Kota Kinabalu, Malaysia, Jan. 2014, pp. 1–2.

[52] B. Shuval and I. Tal, (Oct. 2017). “Fast Polarization for Processes with Memory.” [Online]. Available: https://arxiv.org/abs/1710.02849.

[CURRICULUM VITAE]

Education:

- **University of Ulsan**
  - Ph.D. in Coding and Information Theory Lab
    - September 2015 – August 2018
  - M.S in Coding and Information Theory Lab
    - September 2013 – August 2015
- **Harbin University of Science and Technology**
  - Bachelor in Electronic Information Science and Technology
    - September 2009 – August 2013

[PUBLICATION]

Journal Publications:

[J1] H. Wang and S. Kim, "New RLL decoding algorithm for multiple candidates in visible light communication," *IEEE Photon. Technol. Lett.*, vol. 27, no. 1, pp. 15–17, Jan. 2015. (SCI)

[J2] H. Wang and S. Kim, "Soft-input soft-output run-length limited decoding for visible light communication," *IEEE Photon. Technol. Lett.*, vol. 28, no. 3, pp. 225–228, Feb. 2016. (SCI)

[J3] H. Wang and S. Kim, "Bit-level soft run-length limited decoding algorithm for visible light communication," *IEEE Photon. Technol. Lett.*, vol. 28, no. 3, pp. 237–240, Feb. 2016. (SCI)

[J4] H. Wang and S. Kim, "Soft convolutional codes decoding algorithm for concatenated codes in visible light communication," *Wirel. Pers. Commun.*, Aug. 2017. (SCIE)

[J5] H. Wang and S. Kim, "Dimming control systems with polar codes in visible light communication," *IEEE Photon. Technol. Lett.*, vol. 29, no. 19, pp. 1651–1654, Oct. 2017. (SCI)

[J6] H. Wang and S. Kim, "Decoding of polar codes for intersymbol interference in visible light communication," *IEEE Photon. Technol. Lett.*, accepted at Apr. 2018. (SCI)

Conference Publications:

[C1] W. He and S. Kim, "Implementation and Analysis of Encoding and Decoding of Reed Solomon Codes by Using Matlab," in *KISPS*, Korea, pp. 322–323, 2013

[C2] H. Wang and S. Kim, “Novel Decoding of Run-Length Limited Codes for Visible Light Communication,” in *CSCC*, Greece, pp. 157–160, 2014

[C3] H. Wang and S. Kim, “Novel Soft Decoding Algorithm of RLL Codes for Secure Physical-Layer Visible Light Communications,” in *FiCloud*, Italy, pp. 760–764, 2015

[C4] H. Wang and S. Kim, “A Study on Decoding of Serially Concatenated Codes with RLL Codes and RS Codes in OOK Modulated Systems,” *KICS*, Korea. pp. 250–251, 2016.

[C5] H. Wang and S. Kim, “A Study on Soft Run-Length Limited Codes in Visible Light Communication,” *ICCNT*, Switzerland, pp. 1–4, 2017.