Codes for DC Balance and Dimming Control in Visible Light Communications

Thummaluri Uday Reddy

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology

Department of Electrical Engineering

June 2018

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Uday.

(Signature)

(Thummaluri Uday Reddy)

EE16MTECH11043.

(Roll No.)

Approval Sheet

This Thesis titled Codes for DC Balance and Dimming Control in Visible Light Communications by Thummaluri Uday Reddy is approved for the degree of Master of Technology from IIT Hyderabad

21/6/18

(Dr. Abhinav Kumar) Adviser
Dept. of Electrical Eng
IITH

21/06/18

(Dr. Antony Franklin)
External Examiner
Dept. of CSE
IITH

21/6/18

(Dr. Sumohana C.)
Internal Examiner
Dept. of EE
IITH

P. Raja

(Dr. P. Rajalakshmi)
Chairman
Dept. of EE
IITH

Acknowledgements

I am fortunate to have Dr. Abhinav Kumar as my thesis adviser. I would like to heart fully thank him for motivating me towards research. His time given to listen my thoughts, validating, reviewing and improving them are unforgettable. Without his encouragement, ideas and suggestions I could not have done this work.

I would like to thank Dr. Lakshmi Prasad Natarajan for his constructive comments, suggestions and reviews during the course of this thesis.

I would like to thank my batch friends Kishore, Anil, Dheeraj, Rakesh and Avinash for their company and the care they have taken in odd days. I would also like to thank my WiCon lab friends Yoghitha, Pranay, Bala, Dhanuja and Sreenivas for bearing me during this period. The time spent with either of the batches, parties and the fun that we had are memorable.

I thank my parents who were always supporting me and for the flexibility and guidance they gave in making decisions.

This work was supported in part by the Department of Science and Technology (DST), Govt. of India (Ref. No. TMD/CERI/BEE/2016/059(G)). Finally, thanks to every one who directly or indirectly helped me in completing this work.

Abstract

In visible light communications (VLC), direct current (DC) level balancing is important to maintain constant illumination while the light emitting diodes are being used for communication. In this thesis, perfectly DC balanced codes which have equal number of zeros and ones in all their codewords are proposed. Given codeword length of n bits, we provide a relationship between maximum possible input bits, k, and n for a perfectly DC balanced kb/nb code. We propose an algorithm to generate the codebook of these perfectly DC balanced codes that avoid flickering and maintain consistency in the brightness. The performance of the proposed codes is compared with several existing codes in terms of code rate, Hamming distance, frame error rate (FER), and bit error rate (BER). The numerical results show that the proposed codes provide perfect DC balance and perform better than existing codes in terms of minimum Hamming distance, FER, and BER without significant loss in code rate. We also derive a lower bound on average Hamming distance for the proposed perfectly DC balanced codes for VLC. In VLC, dimming control is required to maintain desired ambiance. For dimming control, addition of compensation symbols (CS) at the end of the runlength limited (RLL) codes has been proposed in the literature. In this thesis, we propose novel RLL codes for VLC that utilize these CS to improve the BER performance in addition to providing dimming control. For various dimming factors, we compare the BER performance and mutual information of the proposed and existing codes to show that the proposed codes result in improved performance. A brief discussion on high rate codes for VLC is also presented.

Contents

Chapter 1

Introduction

1.1 Visible light communications

Visible light communication (VLC) system is an optical wireless communication system that has come up as an alternative to the existing radio frequency (RF) based communication systems due to its several advantages over RF systems as discussed in [1]. The wavelength range of visible light is 375 to 780 nano meters which corresponds to the frequency range of 400 to 700 THz. Given this wavelength/frequency range VLC provides 1000 times more bandwidth than RF [2]. In VLC, the data transmitted through the light emitting diodes (LEDs) lies in this wavelength range. Two important areas where VLC systems are used is indoor communication systems and intelligent transport systems [3]. A brief note on the areas of application of VLC is as follows:

- Indoor communications: VLC systems can be used for indoor data transfer. When used for indoor applications, the effect of solar radiation is negligible.

- Vehicle-to-vehicle (V2V) communications: VLC can be used for V2V communications to prevent road accidents. VLC can also be used for vehicle-to-infrastructure and infrastructure-to-vehicle communications.

- Under water communications: Generally under water, RF waves do not perform well because of their conductivity [4] . Visible light performs better than RF when used for under water communication systems.

- Hospitals: Generally hospitals have electromagnetic sensitive devices like MRI scanners which may interfere with other RF frequencies. Hence, VLC based systems can be used for such applications.

VLC can potentially achieve very high data rates. The existing IEEE 802.15.7 standard for VLC already considers data rates close to 100 Mbps [5]. The usage of visible light for communications was initially demonstrated by Alexander Grahambell where photophone was used to modulate sunlight. Some of the key features of LEDs[1] which lead to their usage in VLC over the conventional sources of light are as follows:

- Fast switching speeds [1].

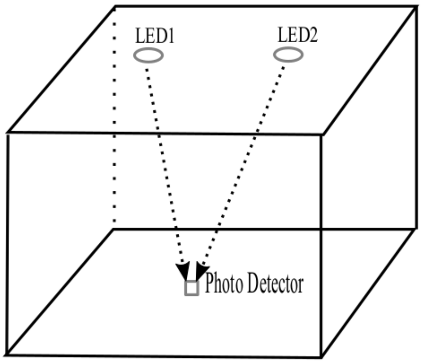

Figure 1.1: An example of a room with two LEDs and one photo detector (PD).

- Low cost of LEDs [1].
- The life time of LEDs is high when compared to the conventional sources [1].
- High luminous efficacy [1].
- Safety in terms of health as they are Mercury free [6], [7].
- Less energy consumption [8].
- Low power dissipation [1].

Luminous efficacy is defined as the efficiency with which input electrical energy is converted to light, measured in lumens/watt. Some important reasons which have generated research interest in VLC in recent times are as follows:

- Congestion in the RF spectrum.
- VLC is secure since it works only in the line of sight range.
- Integration into the existing infrastructure is easy.
- Does not interfere with existing RF technologies.
- Low cost photo receiver circuits.

In VLC, the LED based lighting system serves dual purpose of illumination and communication as shown in Fig. 1.1. The binary data in these VLC systems is transmitted at a high rate such that human eye cannot perceive the flickering [9]. However, the binary stream of data may contain large stream of 0s. Thus, a user may perceive flickering with continuous 0s in the data stream even with high data rate transmissions of the VLC system. Flickering occurs when the duration of zeros exceeds the maximum flickering time period (MFTP). The number of consecutive zeros in the binary data is termed as run length. The MFTP is defined as the maximum time period over which the change in light intensity is not perceived by the human eye. The MFTP should be 5ms or less (i.e., frequency greater than or equal to 200Hz) for human eye not to perceive flickering. Thus, the relationship between MFTP, run length, and the bit period denoted by T_b is given as

$$\text{run length} \times T_b \leq \text{MFTP}.$$

To mitigate the problem of flicker and achieve consistency in illumination, we introduce the concept of perfectly direct current (DC) balanced codes.

Perfectly DC balanced codes have equal number of 0s and 1s in all their codewords. Thus, perfect DC balance implies zero running disparity (RD) and ensures no flickering along with consistency in the illumination. DC level balanced codes have been proposed in [10, 11]. Similarly, a novel $4b/5b$ run length limited (RLL) code has been proposed for VLC in [12] that ensures no flickering. However, the code in [12] does not ensure perfect DC balance. Thus, the consistency in illumination does not exist for the code in [12]. Hence, in this work, we present a generalized algorithm to obtain perfectly DC balanced kb/nb codes for any given codeword length of n bits. These codes ensure no flickering and consistent illumination.

Given different times of the day, maintaining suitable illumination level is required to ensure desired ambiance. Therefore, varying brightness levels are required from the LEDs. Hence, dimming level control of the LEDs is required to achieve these variations in brightness levels. For On-Off keying modulated VLC systems, the dimming factor is defined as the ratio of number of 1s to the length of the codeword. Thus, higher the dimming factor, more will be the brightness. To achieve any arbitrary dimming level given a perfectly DC balanced code, compensation symbols (CS) are introduced into the codewords of the perfectly DC balanced code. These CS are the additional symbols added to the actual codeword. They have only either 0s or 1s depending on whether to increase or decrease the brightness levels, respectively. A novel dimming control system has been proposed in [13] that utilizes addition of CS at the end of the codewords corresponding to the desired dimming factor. However, CS can be simultaneously used to improve the performance of the RLL codes. The improvement in the performance of the codes is possible by properly using the CS among the bits of the codewords of the given perfectly DC balanced code such the nearest neighbors for a given codeword are reduced as compared with the actual code. Motivated by this, we use CS for dimming control and performance improvement of RLL codes.

Existing codes for VLC systems are designed to mitigate flicker and provide consistent illumination at the cost of code rate. However, certain line of sight use cases exist where only flicker mitigation is required with high code rate in VLC systems. Motivated by this, we propose complemented message last bit inversion/uncomplemented message last bit inversion (UMLBI/CMLBI) algorithm to generate high rate RLL codes that avoid flicker during data transmission. We also provide a simple encoder and decoder structures for the proposed UMLBI/CMLBI algorithm. Though the proposed codes does not have equal number of 1s and 0s in their codewords, the changes in the intensity in the long run is reasonable.

There are several modulation schemes for VLC. Few important schemes are as follows:

- On-Off keying (OOK): In VLC, unipolar OOK is used because each level in the symbol corresponds to the intensity of LED. Manchester, $4b/6b$, and $8b/10b$ codes [10] are some of the existing coding schemes that use OOK.

- Pulse width modulation (PWM): In PWM, the information lies in the width of the pulse in the given symbol duration. The amount of power transmitted from LEDs can be controlled by varying the width of the pulses which is used to control the brightness levels which is an important consideration for VLC to maintain the desired ambiance when used for indoor applications.

- Pulse position modulation (PPM): In PPM, the information lies in the position of the pulse in the given symbol duration [14], [15], [16].

- Pulse amplitude modulation (PAM): In VLC, data can also be transmitted by varying intensities of the LEDs. PAM for VLC is different from the conventional PAM that is used for RF in that amplitude levels in PAM for VLC are strictly non-negative where as conventional PAM can have negative levels.

- Pulse dual slope modulation (PDSM): In PDSM, the slope of leading and falling edges is used to transfer 0 and 1, respectively [17]. In this scheme the duration of 0 and 1 is same and the power transmitted is also same which keeps the illumination constant. However, the detection of leading and falling edges is relatively difficult when compared to OOK, PWM.

1.2 Contributions

The contributions of this thesis are as follows:

- For various values of codeword length n, we present a mechanism to obtain the maximum permitted value of the input k bits for which a perfectly DC balanced code can be generated. We propose an algorithm to form the codebook for these perfectly DC balanced codes, which provides the maximum possible code rate for a given n. We derive a lower bound on the average Hamming distance for the proposed perfectly DC balanced codes. Through extensive numerical results, we compare the performance of the proposed codes with some well known existing codes.
- For dimming control, given the codebook of the standard $4b/6b$ code for VLC [10], we propose an algorithm that gives the codebook corresponding to the desired dimming factor. Through numerical results, we show that the proposed algorithm results in improved performance of BER and mutual information than the existing method.
- To generate high rate codes for VLC, an uncomplemented message last bit inversion/complemented message last bit inversion (UMLBI/CMLBI) algorithm is proposed. Simple encoder and decoder structures are provided for UMLBI/CMLBI algorithm. The proposed high rate codes are evaluated for various performance metrics and suitable numerical results are presented.

1.3 Organization of the thesis

The thesis is organized in the following five chapters.

- In Chapter II, we present the perfectly DC balanced codes with an algorithm for code design. The performance metrics for perfectly DC balance codes and numerical results to compare the proposed codes with the existing codes are also presented.
- In Chapter III, we propose codes for dimming control and present numerical results to compare the performance of the proposed codes with the existing method for dimming control.
- In Chapter IV, we present a generalized algorithm for high rate codes generation for VLC with suitable numerical results.
- In Chapter V, we share some concluding remarks.

Chapter 2

Perfectly DC Balanced Codes

In this chapter, we first discuss the code for achieving maximum possible coderate with perfect DC balance. Then, algorithm to form codebook of such codes is presented. Note that, though the discussion is on maximum possible rate codes for given n, any lower rate achieving code can also be formed. Later, performance metrics and numerical results are presented for the proposed codes.

2.1 Code formulation

Let n be the number of bits in the encoded output sequence. Then, we consider the code to be of the format kb/nb , where, b represents bits and k denotes the number of information bits to be encoded such that the total number of possible message vectors are 2^k. Out of the 2^n possible vector sequences of length n, we want to select only those sequences that have 0 RD, i.e., equal number of 0s and 1s. Hence, n has to be even to achieve perfect DC balancing. The number of sequences with length n and 0 RD is given by

$$\binom{n}{n/2} = \frac{n!}{(\frac{n}{2}!)^2} \,. \tag{2.1}$$

To map all possible input message vectors to the DC balanced sequences, the number of message vectors, 2^k, cannot be greater than the number of possible DC balanced sequences as given in (2.1). Thus,

$$2^k \leq \binom{n}{n/2} \,. \tag{2.2}$$

For a given n sized sequence length, the expression in (2.2) can be used to compute the maximum permitted value of k that results in the perfectly DC balanced codes of length n with maximum possible code rate of k/n. A lower value of k can also be selected for perfect DC balancing. However, this comes at the cost of decreased code rate. In Table 2.1, various values of permitted k for given n computed using (2.2) are presented. Next, we present an algorithm to generate perfectly DC balanced codes.

Table 2.1: Perfectly DC balanced kb/nb codes.

Code Format	Range of n	Maximum permitted value of k
$(n-2)b/nb$	≤ 8	n-2
$(n-3)b/nb$	≥ 10 and ≤ 40	n-3
$(n-4)b/nb$	≥ 42 and ≤ 162	n-4
$(n-5)b/nb$	≥ 164	n-5

Algorithm 1 Proposed algorithm for perfectly DC balanced codebook formation

```
INPUT : $n$
OUTPUT : $\{\underline{C}_i\} \forall i \in \{0, \ldots, 2^k - 1\}$
Based on the range of $n$, select the value of $k$ from Table 2.1.
Generate all possible $2^n$ vectors of length $n$.
Separate the $\binom{n}{n/2}$ vectors with 0 RD.
From the $\binom{n}{n/2}$ vectors with 0 RD, select $2^{k-1}$ random vectors excluding two vectors with maximum run length, i.e., starting and ending with 0s. Further, the complement of any vector should not exist in the selected $2^{k-1}$ vectors.
Complement the vectors selected in Step 6. Together with the original $2^{k-1}$ vectors, these $2^{k-1}$ complemented vectors will constitute the codebook $\{\underline{C}_i\}$ with $2^k$ codewords that are DC balanced by design and have large Hamming distances due to the complements.
```

2.2 Codebook formation

Let the minimum Hamming distance of a code be the minimum distance between any arbitrary pair of codewords. Then, the probability of bit error for a codebook decreases with increasing minimum Hamming distance [18]. Using this as a motivation, we form the codebook as follows. Given the required codeword length n, we select the value of k from Table 2.1 that results in maximum possible code rate with perfect DC balance. We first generate all possible 2^n vectors of length n. Then, we separate out the $\binom{n}{n/2}$ vectors with zero RD. From these $\binom{n}{n/2}$ vectors, we select any 2^{k-1} vectors excluding two vectors with maximum run length, i.e., starting and ending with 0s. For e.g., for $4b/6b$ code, we exclude 111000 and 000111 which have maximum run length. Further, the complement of any vector should not exist in the selected 2^{k-1} vectors. Next, we complement the 2^{k-1} selected vectors with zero RD. Together with the original 2^{k-1} vectors, these 2^{k-1} complemented vectors will constitute the proposed codebook $\{\underline{C}_i\}$ with 2^k codewords that are perfectly DC balanced by design and have large Hamming distances due to the complements. We present this codebook formation as an algorithm in Algo. 1 [20]. Next, we present suitable metrics used to evaluate the performance of the proposed codes.

2.3 Performance metrics

The following are the performance metrics considered to compare the proposed codes with the existing codes

- Minimum Hamming distance
- Average Hamming distance
- Coderate

- Run length
- Bit error rate
- Frame error rate

Let $\underline{C}_i$ denote the i^{th} codeword in the codebook. We denote the minimum Hamming distance [18] of a codebook by d_{min} such that

$$d_{min} = \min_{\forall i,j,i\neq j} d(\underline{C}_i, \underline{C}_j)\,, \tag{2.3}$$

where, i, $j \in \{0, \ldots, 2^k - 1\}$. The various possible Hamming distances, $d(\underline{C}_i, \underline{C}_j)$, for the proposed kb/nb code will be $(n - 2i)\ \forall i \in \{0, 1, \ldots, (n/2 - 1)\}$. Thus, d_{min} of the proposed codes is 2. Further, by construction in Step 6 of Algo. 1, the codewords with $n/2$ continuous 0s and $n/2$ continuous 1s are avoided in the codebook in order to reduce the run length.

Given any codeword $\underline{C}_i$ from the proposed codebook, another possible codeword $\underline{C}_j$ with Hamming distance u can be constructed by complementing any $u/2$ 0s and $u/2$ 1s. Thus, the total number of possible codewords with Hamming distance u from C_i are $\binom{n/2}{u/2}^2$ such that

$$\binom{n/2}{u/2}^2 = \binom{n/2}{n/2 - u/2}^2. \tag{2.4}$$

This is using $\binom{n}{r} = \binom{n}{n-r}$. From (2.4), the number of possible codewords with Hamming distance l from any $\underline{C}_i$ will be equal to the number of possible codewords with Hamming distance m such that

$$l + m = n\,. \tag{2.5}$$

The average Hamming distance for a given codebook is defined as

$$d_{avg} = \sum_{i=0}^{2^k-1} \sum_{j=0,j\neq i}^{2^k-1} \frac{d(\underline{C}_i, \underline{C}_j)}{2^k(2^k - 1)}\,. \tag{2.6}$$

Given any codeword $\underline{C}_i$ with 0 RD, there exists exactly one codeword with 0 RD and Hamming distance n from $\underline{C}_i$. Further, there can exist at most $\binom{n/2}{1}^2$ codewords at Hamming distance 2 obtained by arbitrarily exchanging position of exactly one 1 and one 0 in $\underline{C}_i$. Thus, using (2.5), (2.6), and considering all possible perfectly DC balanced sequences, we have the following lower bound $d_{avg}^{(lb)}$ on d_{avg} when n is a multiple of 4

$$\begin{aligned} d_{avg}^{(lb)} &= \left\{ n + 2\binom{n/2}{1}^2 + 4\binom{n/2}{2}^2 + (n-2)\binom{n/2}{1}^2 + (n-4)\binom{n/2}{2}^2 + \ldots \right. \\ &\quad \left. +(n/2)\binom{n/2}{n/4}^2 \right\} \Big/ \left(\binom{n}{n/2} - 1 \right) \\ &= \frac{n\left\{1 + \sum_{i=1}^{n/4-1} \binom{n/2}{i}^2 + 0.5\binom{n/2}{n/4}^2\right\}}{\left(\binom{n}{n/2} - 1\right)}. \end{aligned}$$

Table 2.2: Proposed kb/nb codes.

kb/nb	d_{min}	d_{avg}	$d_{avg}^{(lb)}$	Code rate	% of codewords with d_{min}
$2b/4b$	2	2.667	2.4000	0.500	66.67
$4b/6b$	2	3.200	3.1578	0.667	44.67
$6b/8b$	2	4.064	4.0570	0.750	23.31
$7b/10b$	2	5.039	5.0199	0.700	14.84
$9b/12b$	2	6.012	6.0060	0.750	05.87
$11b/14b$	2	7.003	7.0020	0.785	02.05
$13b/16b$	2	8.001	8.0006	0.812	00.68

Similarily, when $n/2$ is odd and $n \geq 6$

$$d_{avg}^{(lb)} = \frac{n\left\{1 + \sum_{i=1}^{(n-2)/4} \binom{n/2}{i}^2\right\}}{\left(\binom{n}{n/2} - 1\right)}. \tag{2.7}$$

Let $\underline{C}_i$ and $\underline{C}_j$ be two codewords from the proposed kb/nb codebook. Then, by construction, including Manchester code ($k = 1$ and $n = 2$), the maximum run length for either $\underline{C}_i$ and $\underline{C}_j$ is $n/2$. Since we excluded the two vectors with $n/2$ continuous 0s and $n/2$ continuous 1s, the worst case run length of bit stream (except for Manchester code) for the overall code, i.e., maximum number of consecutive 0s is given by

$$\left(\frac{n}{2} - 1\right) + \left(\frac{n}{2} - 1\right) = n - 2\,. \tag{2.8}$$

Specifically, for $n \leq 40$, from Table I and (2.8),

$$\text{Worst case run length of bit stream} = \begin{cases} k, & \text{if } k \text{ is even} \\ k+1, & \text{if } k \text{ is odd} \end{cases}.$$

In Table 2.2, for some of the proposed codes the various performance metrics like average Hamming distance, lower bound on the Hamming distance, code rate (k/n), and percentage of codewords with d_{min} are presented. Similarly, in Table 2.3, we present the performance metrics of some of the existing codes like the $1b/2b$ (Manchester code), $4b/5b$ as presented in [12], and $5b/6b$ as proposed in [19] with ($RD = -1$) or ($RD = +1$). Please note that except for the existing $4b/6b$ [10], $6b/8b$ [28], and the Manchester code which is a subset of the proposed class of codes, all the other codes in Table 2.3 do not have exactly 50% of 1s in the codebook. Hence, they are not perfectly DC balanced unlike the proposed codes in Table 2.2. Further, except Manchester code no other perfectly DC balanced $(n-1)b/nb$ code is possible since (2.2) cannot be satisfied. Next, we present the numerical results for the proposed codes.

2.4 Numerical results

For the simulation results, we consider only $2b/4b$, $4b/6b$, $6b/8b$, and $7b/10b$ proposed codes. Further, we consider an additive white Gaussian noise (AWGN) channel for VLC as considered in [23, 24, 25].

Table 2.3: Existing VLC/RLL codes.

kb/nb	d_{min}	d_{avg}	Code rate	% of 1s in the codebook
$1b/2b$ (Manchester code)	2	2.000	0.500	50.00
$4b/5b$ [12]	1	2.342	0.800	66.26
$5b/6b$ ($RD = -1$) [19]	1	3.034	0.833	56.75
$5b/6b$ ($RD = +1$) [19]	1	3.034	0.833	43.25
$4b/6b$ [10]	2	3.200	0.6667	50.00
$6b/8b$ [28]	2	4.059	0.7500	50.00

Thus, the received signal can be modeled as

$$\underline{Y} = \underline{X} + \underline{N}\,,$$

where, $\underline{Y}$ is the received signal, $\underline{X}$ is the transmitted signal, and $\underline{N}$ is the Gaussian noise with zero mean and Identity covariance matrix. Then, the maximum likelihood detection rule for the transmitted signals $\underline{X}_i$ and $\underline{X}_j$ in terms of the likelihood ratio $\Lambda(\underline{Y})$ is given as

$$\Lambda(\underline{Y}) = \frac{f(\underline{Y}/\underline{X}_i)}{f(\underline{Y}/\underline{X}_j)} > 1\,,$$

where, $f(.)$ is the probability density function of $\underline{Y}$ given $\underline{X}_i$. Hence, the detection rule decides in favor of C_q if

$$\sum_{i=1}^{n}(Y_i - \mu_i^q)^2 < \sum_{i=1}^{n}(Y_i - \mu_i^p)^2 \;\; \forall p \in \{0, 1, \ldots, 2^k - 1\}, p \neq q,$$

where, μ_i^q and μ_i^p represent the mean of i^{th} component of the received signal given that C_q and C_p were transmitted, respectively. Given OOK as the modulation scheme considered for VLC in this work, μ_i can be either 0 or 1 as in [24, 25]. This is calculated using the Gaussian pdf which is given as

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right),$$

where, μ is the mean and σ^2 is the variance.

In Fig. 2.1, the variation of frame error rate (FER) with varying SNR is presented for the various codes considered in this work. The FER is the ratio of received frames in error with the total number of transmitted frames. The frame size for kb/nb code is k bits. We performed Monte Carlo simulations for the AWGN channel. The presented results are averaged over 10^6 bits. We see a DC imbalance for $5b/6b$ code with $RD = +1$ and $RD = -1$. Further, on an average codewords of $RD = -1$ have relatively higher number of 1s than $RD = +1$. Hence, there is a performance variation between $RD = +1$ code and $RD = -1$ code.

In Fig. 2.2, the variation of bit error rate (BER) with varying SNR is presented for the various codes considered in this work. The performance of existing Manchester code in terms of FER and BER is the best as observed from Fig. 2.1 and Fig. 2.2, respectively. However, the proposed codes exhibit better error performance than the existing higher rate VLC codes from [12, 19]. This

Table 2.4: Codebook of the Proposed $4b/6b$ code.

Message vector	Codeword
0 0 0 0	1 1 0 1 0 0
0 0 0 1	0 0 1 0 1 1
0 0 1 0	1 0 1 1 0 0
0 0 1 1	0 1 0 0 1 1
0 1 0 0	0 1 1 1 0 0
0 1 0 1	1 0 0 0 1 1
0 1 1 0	1 1 0 0 1 0
0 1 1 1	0 0 1 1 0 1
1 0 0 0	1 0 1 0 1 0
1 0 0 1	0 1 0 1 0 1
1 0 1 0	0 1 1 0 1 0
1 0 1 1	1 0 0 1 0 1
1 1 0 0	1 0 0 1 1 0
1 1 0 1	0 1 1 0 0 1
1 1 1 0	0 1 0 1 1 0
1 1 1 1	1 0 1 0 0 1

performance improvement is achieved at the cost of only a marginal loss in code rate compared to [12, 19] as observed from Table 2.2 and Table 2.3. It is observed from Table 2.2 and Table 2.3 that the performance metrics like d_{min}, d_{avg}, % of ones in the codebook of the proposed perfectly DC balanced $4b/6b$ and $6b/8b$ codes are similar to the existing $4b/6b$ and $6b/8b$ codes from [10] and [28], respectively. Hence, in Fig. 2.1 and Fig. 2.2, we present the simulation results only for the proposed $4b/6b$ and $6b/8b$ codes. For illustration, we have added the codebooks for the proposed $4b/6b$, $6b/8b$, and $7b/10b$ as Table 2.4, Table 2.5, Table 2.6, respectively. Further, the proposed Algo. 1 allows us to generate higher rate codes at the cost of BER.

In Fig. 2.3, we show the DC balancing capacity of various codes like $4b/5b$, $6b/8b$, $8b/10b$, and $7b/10b$ by showing a sample codeword from each of the codes. For $8b/10b$ code only 4 out of all the codewords are not perfectly DC balanced. In Fig. 2.3, one of the unbalanced codewords is presented. We can also see that except $4b/5b$ and $8b/10b$ all are perfectly DC balanced.

In this work, we compared all the codes for 50 % dimming level. However, to maintain different dimming levels, the duration of the transmitted symbol corresponding to bit 1 and 0 in the proposed codewords can be varied. In future, the performance of the proposed perfectly DC balanced codes can be tested on hardware for various dimming levels.

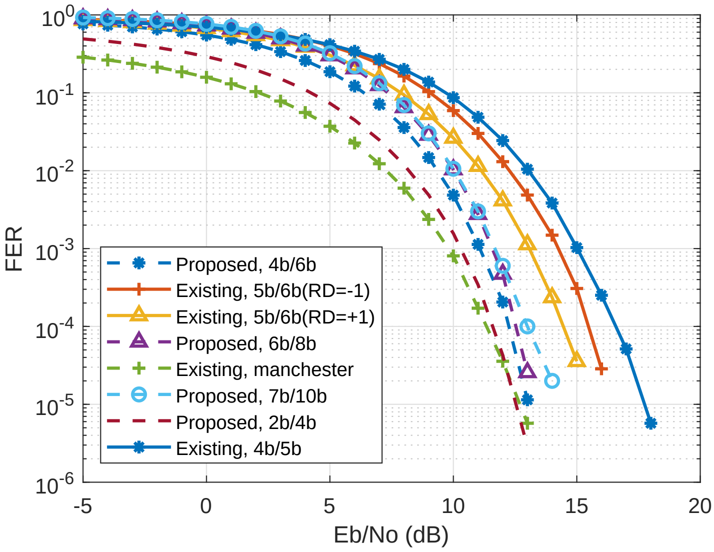

Figure 2.1: FER performance of various VLC codes with varying SNR.

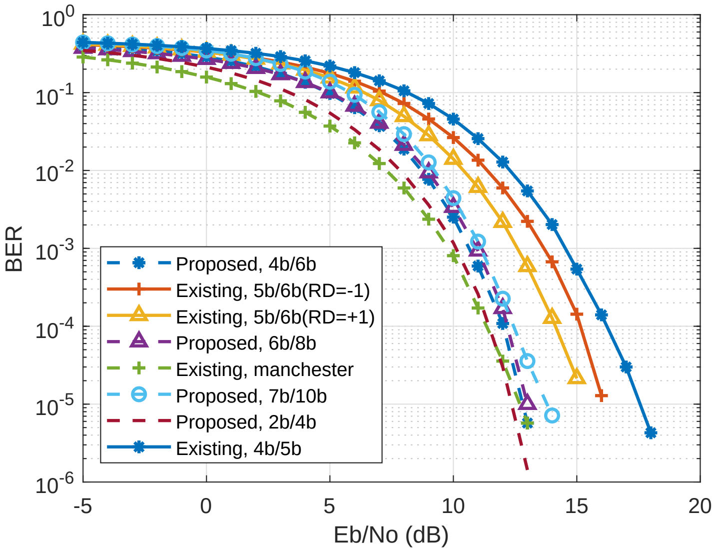

Figure 2.2: BER performance of various VLC codes with varying SNR.

Table 2.5: Codebook of the Proposed 6*b*/8*b* code.

Message vector	Codeword
0 0 0 0 0 0	1 1 1 0 1 0 0 0
0 0 0 0 0 1	0 0 0 1 0 1 1 1
0 0 0 0 1 0	1 1 0 1 1 0 0 0
0 0 0 0 1 1	0 0 1 0 0 1 1 1
0 0 0 1 0 0	1 0 1 1 1 0 0 0
0 0 0 1 0 1	0 1 0 0 0 1 1 1
0 0 0 1 1 0	0 1 1 1 1 0 0 0
0 0 0 1 1 1	1 0 0 0 0 1 1 1
0 0 1 0 0 0	1 1 1 0 0 1 0 0
0 0 1 0 0 1	0 0 0 1 1 0 1 1
0 0 1 0 1 0	1 1 0 1 0 1 0 0
0 0 1 0 1 1	0 0 1 0 1 0 1 1
0 0 1 1 0 0	1 0 1 1 0 1 0 0
0 0 1 1 0 1	0 1 0 0 1 0 1 1
0 0 1 1 1 0	0 1 1 1 0 1 0 0
0 0 1 1 1 1	1 0 0 0 1 0 1 1
0 1 0 0 0 0	1 1 0 0 1 1 0 0
0 1 0 0 0 1	0 0 1 1 0 0 1 1
0 1 0 0 1 0	1 0 1 0 1 1 0 0
0 1 0 0 1 1	0 1 0 1 0 0 1 1
0 1 0 1 0 0	0 1 1 0 1 1 0 0
0 1 0 1 0 1	1 0 0 1 0 0 1 1
0 1 0 1 1 0	1 0 0 1 1 1 0 0
0 1 0 1 1 1	0 1 1 0 0 0 1 1
0 1 1 0 0 0	0 1 0 1 1 1 0 0
0 1 1 0 0 1	1 0 1 0 0 0 1 1
0 1 1 0 1 0	0 0 1 1 1 1 0 0
0 1 1 0 1 1	1 1 0 0 0 0 1 1
0 1 1 1 0 0	1 1 1 0 0 0 1 0
0 1 1 1 0 1	0 0 0 1 1 1 0 1
0 1 1 1 1 0	1 1 0 1 0 0 1 0
0 1 1 1 1 1	0 0 1 0 1 1 0 1
1 0 0 0 0 0	1 0 1 1 0 0 1 0
1 0 0 0 0 1	0 1 0 0 1 1 0 1
1 0 0 0 1 0	0 1 1 1 0 0 1 0
1 0 0 0 1 1	1 0 0 0 1 1 0 1
1 0 0 1 0 0	1 1 0 0 1 0 1 0
1 0 0 1 0 1	0 0 1 1 0 1 0 1
1 0 0 1 1 0	1 0 1 0 1 0 1 0
1 0 0 1 1 1	0 1 0 1 0 1 0 1
1 0 1 0 0 0	0 1 1 0 1 0 1 0
1 0 1 0 0 1	1 0 0 1 0 1 0 1
1 0 1 0 1 0	1 0 0 1 1 0 1 0
1 0 1 0 1 1	0 1 1 0 0 1 0 1
1 0 1 1 0 0	0 1 0 1 1 0 1 0
1 0 1 1 0 1	1 0 1 0 0 1 0 1
1 0 1 1 1 0	0 0 1 1 1 0 1 0
1 0 1 1 1 1	1 1 0 0 0 1 0 1
1 1 0 0 0 0	1 1 0 0 0 1 1 0

1 1 0 0 0 1	0 0 1 1 1 0 0 1
1 1 0 0 1 0	1 0 1 0 0 1 1 0
1 1 0 0 1 1	0 1 0 1 1 0 0 1
1 1 0 1 0 0	0 1 1 0 0 1 1 0
1 1 0 1 0 1	1 0 0 1 1 0 0 1
1 1 0 1 1 0	1 0 0 1 0 1 1 0
1 1 0 1 1 1	0 1 1 0 1 0 0 1
1 1 1 0 0 0	0 1 0 1 0 1 1 0
1 1 1 0 0 1	1 0 1 0 1 0 0 1
1 1 1 0 1 0	0 0 1 1 0 1 1 0
1 1 1 0 1 1	1 1 0 0 1 0 0 1
1 1 1 1 0 0	1 0 0 0 1 1 1 0
1 1 1 1 0 1	0 1 1 1 0 0 0 1
1 1 1 1 1 0	0 1 0 0 1 1 1 0
1 1 1 1 1 1	1 0 1 1 0 0 0 1

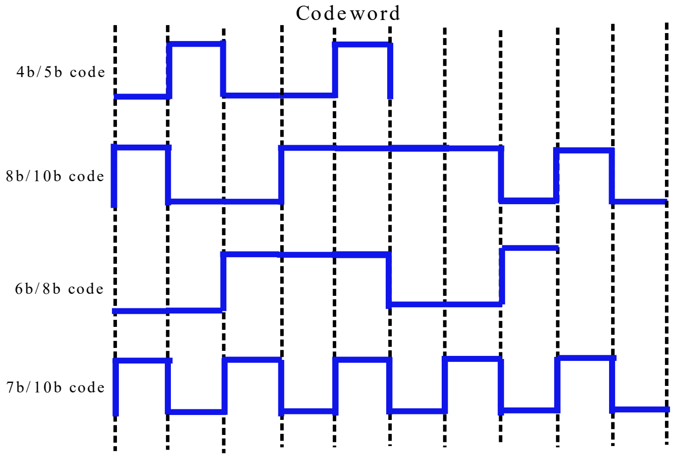

Figure 2.3: DC balance of various codes.

Table 2.6: Codebook of the Proposed $7b/10b$ code.

Message vector	Codeword
0 0 0 0 0 0 0	1 1 1 1 0 1 0 0 0 0
0 0 0 0 0 0 1	1 1 1 0 1 1 0 0 0 0
0 0 0 0 0 1 0	1 1 0 1 1 1 0 0 0 0
0 0 0 0 0 1 1	1 0 1 1 1 1 0 0 0 0
0 0 0 0 1 0 0	0 1 1 1 1 1 0 0 0 0
0 0 0 0 1 0 1	1 1 1 1 0 0 1 0 0 0
0 0 0 0 1 1 0	1 1 1 0 1 0 1 0 0 0
0 0 0 0 1 1 1	1 1 0 1 1 0 1 0 0 0
0 0 0 1 0 0 0	1 0 1 1 1 0 1 0 0 0
0 0 0 1 0 0 1	0 1 1 1 1 0 1 0 0 0
0 0 0 1 0 1 0	1 1 1 0 0 1 1 0 0 0
0 0 0 1 0 1 1	1 1 0 1 0 1 1 0 0 0
0 0 0 1 1 0 0	1 0 1 1 0 1 1 0 0 0
0 0 0 1 1 0 1	0 1 1 1 0 1 1 0 0 0
0 0 0 1 1 1 0	1 1 0 1 1 1 0 0 0 0
0 0 0 1 1 1 1	1 0 1 0 1 1 1 0 0 0
0 0 1 0 0 0 0	0 1 1 0 1 1 1 0 0 0
0 0 1 0 0 0 1	1 0 0 1 1 1 1 0 0 0
0 0 1 0 0 1 0	0 1 0 1 1 1 1 0 0 0
0 0 1 0 0 1 1	0 0 1 1 1 1 1 0 0 0
0 0 1 0 1 0 0	1 1 1 1 0 0 0 1 0 0
0 0 1 0 1 0 1	1 1 1 0 1 0 0 1 0 0
0 0 1 0 1 1 0	1 1 0 1 1 0 0 1 0 0
0 0 1 0 1 1 1	1 0 1 1 1 0 0 1 0 0
0 0 1 1 0 0 0	0 1 1 1 1 0 0 1 0 0
0 0 1 1 0 0 1	1 1 1 0 0 1 0 1 0 0
0 0 1 1 0 1 0	1 1 0 1 0 1 0 1 0 0
0 0 1 1 0 1 1	1 0 1 1 0 1 0 1 0 0
0 0 1 1 1 0 0	0 1 1 1 0 1 0 1 0 0
0 0 1 1 1 0 1	1 1 0 0 1 1 0 1 0 0
0 0 1 1 1 1 0	1 0 1 0 1 1 0 1 0 0
0 0 1 1 1 1 1	0 1 1 0 1 1 0 1 0 0
0 1 0 0 0 0 0	1 0 0 1 1 1 0 1 0 0
0 1 0 0 0 0 1	0 1 0 1 1 1 0 1 0 0
0 1 0 0 0 1 0	0 0 1 1 1 1 0 1 0 0
0 1 0 0 0 1 1	1 1 1 0 0 0 1 1 0 0
0 1 0 0 1 0 0	1 1 0 1 0 0 1 1 0 0
0 1 0 0 1 0 1	1 0 1 1 0 0 1 1 0 0
0 1 0 0 1 1 0	0 1 1 1 0 0 1 1 0 0
0 1 0 0 1 1 1	1 1 0 0 1 0 1 1 0 0
0 1 0 1 0 0 0	1 0 1 0 1 0 1 1 0 0
0 1 0 1 0 0 1	0 1 1 0 1 0 1 1 0 0
0 1 0 1 0 1 0	1 0 0 1 1 0 1 1 0 0
0 1 0 1 0 1 1	0 1 0 1 1 0 1 1 0 0
0 1 0 1 1 0 0	0 0 1 1 1 0 1 1 0 0
0 1 0 1 1 0 1	1 1 0 0 0 1 1 1 0 0
0 1 0 1 1 1 0	1 0 1 0 0 1 1 1 0 0
0 1 0 1 1 1 1	0 1 1 0 0 1 1 1 0 0
0 1 1 0 0 0 0	1 0 0 1 0 1 1 1 0 0

0 1 1 0 0 0 1	0 1 0 1 0 1 1 1 0 0
0 1 1 0 0 1 0	0 0 1 1 0 1 1 1 0 0
0 1 1 0 0 1 1	1 0 0 0 1 1 1 1 0 0
0 1 1 0 1 0 0	0 1 0 0 1 1 1 1 0 0
0 1 1 0 1 0 1	0 0 1 0 1 1 1 1 0 0
0 1 1 0 1 1 0	0 0 0 1 1 1 1 1 0 0
0 1 1 0 1 1 1	1 1 1 1 0 0 0 0 1 0
0 1 1 1 0 0 0	1 1 1 0 1 0 0 0 1 0
0 1 1 1 0 0 1	1 1 0 1 1 0 0 0 1 0
0 1 1 1 0 1 0	1 0 1 1 1 0 0 0 1 0
0 1 1 1 0 1 1	0 1 1 1 1 0 0 0 1 0
0 1 1 1 1 0 0	1 1 1 0 0 1 0 0 1 0
0 1 1 1 1 0 1	1 1 0 1 0 1 0 0 1 0
0 1 1 1 1 1 0	1 0 1 1 0 1 0 0 1 0
0 1 1 1 1 1 1	0 1 1 1 0 1 0 0 1 0
1 0 0 0 0 0 0	0 0 0 0 1 0 1 1 1 1
1 0 0 0 0 0 1	0 0 0 1 0 0 1 1 1 1
1 0 0 0 0 1 0	0 0 1 0 0 0 1 1 1 1
1 0 0 0 0 1 1	0 1 0 0 0 0 1 1 1 1
1 0 0 0 1 0 0	1 0 0 0 0 0 1 1 1 1
1 0 0 0 1 0 1	0 0 0 0 1 1 0 1 1 1
1 0 0 0 1 1 0	0 0 0 1 0 1 0 1 1 1
1 0 0 0 1 1 1	0 0 1 0 0 1 0 1 1 1
1 0 0 1 0 0 0	0 1 0 0 0 1 0 1 1 1
1 0 0 1 0 0 1	1 0 0 0 0 1 0 1 1 1
1 0 0 1 0 1 0	0 0 0 1 1 0 0 1 1 1
1 0 0 1 0 1 1	0 0 1 0 1 0 0 1 1 1
1 0 0 1 1 0 0	0 1 0 0 1 0 0 1 1 1
1 0 0 1 1 0 1	1 0 0 0 1 0 0 1 1 1
1 0 0 1 1 1 0	0 0 1 1 0 0 0 1 1 1
1 0 0 1 1 1 1	0 1 0 1 0 0 0 1 1 1
1 0 1 0 0 0 0	1 0 0 1 0 0 0 1 1 1
1 0 1 0 0 0 1	0 1 1 0 0 0 0 1 1 1
1 0 1 0 0 1 0	1 0 1 0 0 0 0 1 1 1
1 0 1 0 0 1 1	1 1 0 0 0 0 0 1 1 1
1 0 1 0 1 0 0	0 0 0 0 1 1 1 0 1 1
1 0 1 0 1 0 1	0 0 0 1 0 1 1 0 1 1
1 0 1 0 1 1 0	0 0 1 0 0 1 1 0 1 1
1 0 1 0 1 1 1	0 1 0 0 0 1 1 0 1 1
1 0 1 1 0 0 0	1 0 0 0 0 1 1 0 1 1
1 0 1 1 0 0 1	0 0 0 1 1 0 1 0 1 1
1 0 1 1 0 1 0	0 0 1 0 1 0 1 0 1 1
1 0 1 1 0 1 1	0 1 0 0 1 0 1 0 1 1
1 0 1 1 1 0 0	1 0 0 0 1 0 1 0 1 1
1 0 1 1 1 0 1	0 0 1 1 0 0 1 0 1 1
1 0 1 1 1 1 0	0 1 0 1 0 0 1 0 1 1
1 0 1 1 1 1 1	1 0 0 1 0 0 1 0 1 1
1 1 0 0 0 0 0	0 1 1 0 0 0 1 0 1 1
1 1 0 0 0 0 1	1 0 1 0 0 0 1 0 1 1
1 1 0 0 0 1 0	1 1 0 0 0 0 1 0 1 1
1 1 0 0 0 1 1	0 0 0 1 1 1 0 0 1 1
1 1 0 0 1 0 0	0 0 1 0 1 1 0 0 1 1

1 1 0 0 1 0 1	0 1 0 0 1 1 0 0 1 1
1 1 0 0 1 1 0	1 0 0 0 1 1 0 0 1 1
1 1 0 0 1 1 1	0 0 1 1 0 1 0 0 1 1
1 1 0 1 0 0 0	0 1 0 1 0 1 0 0 1 1
1 1 0 1 0 0 1	1 0 0 1 0 1 0 0 1 1
1 1 0 1 0 1 0	0 1 1 0 0 1 0 0 1 1
1 1 0 1 0 1 1	1 0 1 0 0 1 0 0 1 1
1 1 0 1 1 0 0	1 1 0 0 0 1 0 0 1 1
1 1 0 1 1 0 1	0 0 1 1 1 0 0 0 1 1
1 1 0 1 1 1 0	0 1 0 1 1 0 0 0 1 1
1 1 0 1 1 1 1	1 0 0 1 1 0 0 0 1 1
1 1 1 0 0 0 0	0 1 1 0 1 0 0 0 1 1
1 1 1 0 0 0 1	1 0 1 0 1 0 0 0 1 1
1 1 1 0 0 1 0	1 1 0 0 1 0 0 0 1 1
1 1 1 0 0 1 1	0 1 1 1 0 0 0 0 1 1
1 1 1 0 1 0 0	1 0 1 1 0 0 0 0 1 1
1 1 1 0 1 0 1	1 1 0 1 0 0 0 0 1 1
1 1 1 0 1 1 0	1 1 1 0 0 0 0 0 1 1
1 1 1 0 1 1 1	0 0 0 0 1 1 1 1 0 1
1 1 1 1 0 0 0	0 0 0 1 0 1 1 1 0 1
1 1 1 1 0 0 1	0 0 1 0 0 1 1 1 0 1
1 1 1 1 0 1 0	0 1 0 0 0 1 1 1 0 1
1 1 1 1 0 1 1	1 0 0 0 0 1 1 1 0 1
1 1 1 1 1 0 0	0 0 0 1 1 0 1 1 0 1
1 1 1 1 1 0 1	0 0 1 0 1 0 1 1 0 1
1 1 1 1 1 1 0	0 1 0 0 1 0 1 1 0 1
1 1 1 1 1 1 1	1 0 0 0 1 0 1 1 0 1

Chapter 3

Improved RLL Codes

Traditionally to control dimming, addition of CS at the end of perfectly DC balanced codes was proposed. However, we can also use these CS to improve the performance of the codes while achieving the same dimming that is achieved by traditional method of dimming control. In this chapter, we first present the system model considered to implement the proposed codes. Then, we propose an algorithm to generate the codebooks of the proposed codes. Next, performance metrics to compare the proposed codes with the existing codes and numerical results related to BER and mutual information are presented.

3.1 System model

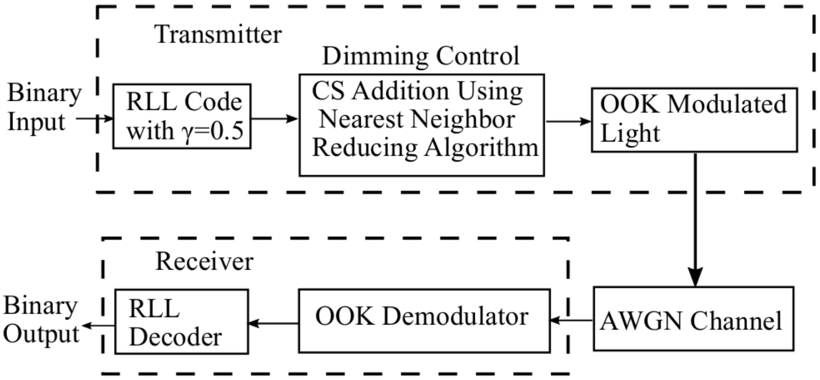

Figure 3.1: Schematic of the VLC system.

A schematic of the VLC system considered in this work is shown in Fig. 3.1. The input bit stream at the transmitter is passed through an RLL code to avoid flickering and ensure uniform illumination. This is followed by dimming control to obtain the desired illumination and OOK modulation as per the VLC standard [21]. Let the RLL code be denoted as kb/nb, where, k denotes the size of message vector and n denotes the size of the codeword. Then, for m number of 1s in a codeword of length n, the dimming factor, γ, is defined as

$$\gamma = \frac{m}{n}\,, \tag{3.1}$$

where, $m = n/2$ for a perfectly DC balanced code. Post the dimming control showed in Fig. 3.1, the number of 0s and 1s need not be equal.

For a given perfectly DC balanced RLL code, if required $\gamma > 0.5$ then extra 1s have to be added as CS to increase the resultant γ post dimming control. This number of CS, denoted by l, for $\gamma > 0.5$ is given as

$$\gamma = \frac{\frac{n}{2} + l}{n + l}\,.$$

Then, to obtain l from the above equation

$$\begin{aligned}(n + l)\gamma &= \frac{n}{2} + l\,,\\ l - l\gamma &= n\gamma - \frac{n}{2}\,,\\ l &= \frac{n(\gamma - 0.5)}{1 - \gamma}\,. \end{aligned} \tag{3.2}$$

Similarly, for $\gamma < 0.5$., the number of 0s added as CS are

$$\gamma = \frac{\frac{n}{2}}{n + l}\,.$$

This results in,

$$l = \frac{n(0.5 - \gamma)}{\gamma}\,. \tag{3.3}$$

We know that $0 \leq \gamma \leq 1$ and if a is the number of 0s, then from (3.1) ratio of 0s and 1s at the boundary conditions, i.e., as γ tends to 0 is given as

$$\begin{aligned}\frac{a}{m} &= lim_{\gamma \to 0} \frac{1 - \gamma}{\gamma},\\ &= \infty.\end{aligned}$$

Similarly, as γ tends to 1,

$$\begin{aligned}\frac{a}{m} &= lim_{\gamma \to 1} \frac{1 - \gamma}{\gamma},\\ &= 0.\end{aligned}$$

We consider an AWGN channel for VLC as in [23]–[25]. Thus, the received vector, $\underline{\mathrm{Y}}$, can be modeled as

$$\underline{\mathrm{Y}}_{1\times(n+l)} = \underline{\mathrm{X}}_{1\times(n+l)} + \underline{\mathrm{N}}_{1\times(n+l)}, \tag{3.4}$$

where, $\underline{\mathrm{X}}$ is the transmitted OOK modulated vector corresponding to the RLL code's output post dimming control of length $(n + l)$, and $\underline{\mathrm{N}}$ is the noise vector. Next, we propose the algorithm to construct the RLL codes using CS for a given γ. In the existing method, the l number of 0s or 1s corresponding to the desired γ are appended at the end of the codewords of the perfectly DC balanced code as shown in [13]. In this work, we aim to reduce the number of nearest neighbors for any given codeword by properly inserting these CS.

3.2 Code design

For the given standard perfectly DC balanced $4b/6b$ code (as in [21]) to achieve the desired γ, we can obtain l using (3.2) for $\gamma > 0.5$ and (3.3) for $\gamma < 0.5$. In case l is even, we construct $l/2$ sequences of bits 11 as CS for $\gamma > 0.5$ and 00 for $\gamma < 0.5$. Alternatively, if l is odd, we construct $\lfloor l/2 \rfloor$ sequences of CS with bits 11 for $\gamma > 0.5$ and exactly one CS with bit 1. Similarly, for $\gamma < 0.5$ and l odd, we construct $\lfloor l/2 \rfloor$ CS with bits 00 and exactly one CS with bit 0. We denote by $N = \lceil l/2 \rceil$ the number of such CS sequences of the kind 0, 1 ,00, and 11 that have to be added to the RLL codewords to achieve the desired γ.

Next, we need to obtain the locations of these CS within the RLL code to maximize performance. By construction, the standard $4b/6b$ code has codewords in complementary pairs. For such complementary codeword pairs, it is not possible to increase the Hamming distance by addition of only 1s or 0s, i.e., the CS. Therefore, we insert the CS in the same location for such complementary codeword pairs as it preserves the largest Hamming distance between the complementary codeword pairs. We divide the N CS sequences into groups of $S = \lceil \frac{N}{2} \rceil$ and $T = N - S$ sequences. We insert S and T in the existing codewords to reduce their number of nearest neighboring codewords. Let P be a sequence of 11 or 1 for $\gamma > 0.5$ and 00 or 0 for $\gamma > 0.5$. Then, we insert T, P, and S as shown in Step 6 of Algo. 2 [26]. This pattern ensures that the number of nearest neighbors for any codeword are minimized without compromising on the run length.

Note that by construction, the codebook $\mathscr{C}$ corresponding to a γ is equal to the complement of the codebook corresponding to (1-γ) because the CS when $\gamma < 0.5$, i.e. 0s are complements of CS when $\gamma > 0.5$, i.e., 1s.

For illustration, let us consider two complementary codewords
$\underline{C}_1$ = 00 11 10 and
$\underline{C}_2$ = 11 00 01.
Then, for $\gamma = 0.25$, $l = 6$ which using Algo. 2 results in output codewords as
$\underline{C'}_1$ **00** 00 11 10 **00 00** and
$\underline{C'}_2$ **00** 11 00 01 **00 00**.
Thus, maintaining the original maximum Hamming distance. Similarly, for $\gamma = 0.25$, consider two nearest neighboring codewords
$\underline{C}_1$ = 00 11 10 and
$\underline{C}_2$ = 00 11 01.
Then, the output codewords are
$\underline{C'}_1$ = **00** 00 11 10 **00 00** and

Algorithm 2 Proposed algorithm for generating the codebook given γ and the standard $4b/6b$ VLC code

```
INPUT : $\mathscr{C}$ corresponding to $\gamma$=0.5 and desired $\gamma$.
OUTPUT : $\mathscr{C}$ corresponding to desired $\gamma$.
Compute $l$ using (3.2) for $\gamma > 0.5$ and (3.3) for $\gamma < 0.5$.
Construct $l/2$ sequences of CS, each element of the sequence with two bits 11 or 00 for $\gamma > 0.5$ and $\gamma < 0.5$, respectively. In case of $l$ odd, one of these sequences will be only one bit. Let $P$ denote a sequence of CS that can be 11 or 1 for $\gamma > 0.5$ and 00 or 0 for $\gamma < 0.5$. Thus, total CS sequences will always be $N = \lceil l/2 \rceil$.
We divide the $N$ CS sequences into groups of $S = \lceil \frac{N}{2} \rceil$ and $T = N - S$ sequences.
Given $S$ and $T$ CS sequences, we generate the proposed RLL code from the $4b/6b$ by inserting them as follows:
    $T$ 00 11 10 $S$
    00 11 01 $T$ $S$
    $T$ $S$ 01 00 11
    01 01 $T$ $S$ 10
    $T$-1 01 $P$ 01 $P$ 01 $S$-1
    10 $S$ $T$ 00 11
    $S$ 10 01 10 $T$
    $S$-1 10 $P$ 01 $P$ 01 $T$-1
    and the remaining codewords are obtained by complementing all the above bits except $T$, $P$, and $S$.
Specifically, for $N = 1$, we append N CS on either side alternately to obtain the following.
    $N$ 00 11 10
    00 11 01 $N$
    $N$ 01 00 11
    01 01 10 $N$
    $N$ 01 01 01
    10 00 11 $N$
    $N$ 10 11 00
    10 01 10 $N$
The above 8 codewords with their complements (only in bits and not $N$) form the codebook with the required $\gamma$.
```

$\underline{C'}_2$= 00 11 01 **00 00 00**.

Thus, the proposed Algo. 2 can increase the Hamming distance between some of the nearest neighboring codewords for desired γ.

The codes generated by the proposed algorithm for $\gamma = 0.75$ and $\gamma = 0.625$ are shown in Table 3.2 and Table 3.3, respectively. Note that for $\gamma = 0.25$ and $\gamma = 0.375$, the codes can be obtained by complementing the codes in Table 3.2 and Table 3.3 for $\gamma = 0.75$ and $\gamma = 0.625$, respectively. The codebooks for $\gamma = 0.25$ and $\gamma = 0.375$ are shown in Table 3.4 and Table 3.5, respectively. The various performance metrics for comparison of the proposed codes with the existing ones are explained next.

3.3 Performance metrics

We consider the following metrics to compare the performance of the proposed codes with the existing codes for VLC with various dimming factors.

- Average Hamming distance

Table 3.1: Exsiting 4b/6b code.

Message Vector	Existing for γ=0.5
0000	001110
0001	001101
0010	010011
0011	010110
0100	010101
0101	100011
0110	100110
0111	100101
1000	011001
1001	011010
1010	011100
1011	110001
1100	110010
1101	101001
1110	101010
1111	101100

- Run length
- Mutual information
- Fraction of codeword pairs at minimum Hamming distance
- Bit error rate
- Coderate

The proposed algorithm results in codes with increased d_{avg} as compared to the existing codes for various dimming factors as shown in Table 3.6. The d_{min} is same for the proposed and existing codes. We denote the fraction of codeword pairs at minimum Hamming distance by $f^{cw}_{d_{min}}$. This fraction of codewords at d_{min} is equal to

$$\begin{aligned} f^{cw}_{d_{min}} &= number\ of\ codeword\ pairs\ with \\ &\quad distance\ equal\ to\ d_{min} / \binom{2^k}{2}. \end{aligned} \tag{3.5}$$

Then, $\binom{2^k}{2}$ denotes that a pair of codewords that are choosen from 2^k codewords which constitute the codebook. It is observed from Table 3.6 that the proposed algorithm for nearest neighbor reduction results in significant decrease in the $f^{cw}_{d_{min}}$ of the proposed codes as compared to the existing codes for various dimming factors.

The run length is the maximum number of continuous 0s in the binary data. The run length of the proposed codes is always less than or equal to the run length of the existing codes with dimming control in [13] for various γ.

Table 3.2: $4b/6b$ code for γ=.75.

Message Vector	Codeword
0000	110011101111
0001	001101111111
0010	111111010011
0011	010111111110
0100	011101110111
0101	101111110011
0110	111110011011
0111	111011011101
1000	111101100111
1001	110111101110
1010	011111111100
1011	111100011111
1100	110010111111
1101	101011111101
1110	101110111011
1111	111111101100

The log likelihood ratio (LLR) for any message bit, m_i, is given as

$$LLR\left(m_i\right) \quad = \quad log_e\left(\frac{P\left[m_i=0|\underline{Y}\right]}{P\left[m_i=1|\underline{Y}\right]}\right). \tag{3.6}$$

The vector Gaussian distribution is given as

$$f(x) = \frac{1}{\left(2\pi\Sigma\right)^{1/2}} exp\left((x-\mu)\Sigma^{-1}(x-\mu)^T\right), \tag{3.7}$$

where, x is a row vector and μ is a mean row vector. For AWGN, (3.6) simplifies to

$$LLR\left(m_i\right) = log_e\left(\frac{\sum\limits_{\underline{C}\in\mathscr{C},m_i=0} e^{\left(\frac{-1}{2}(\underline{Y}-\underline{C})\Sigma^{-1}(\underline{Y}-\underline{C})^T\right)}}{\sum\limits_{\underline{C}\in\mathscr{C},m_i=1} e^{\left(\frac{-1}{2}(\underline{Y}-\underline{C})\Sigma^{-1}(\underline{Y}-\underline{C})^T\right)}}\right), \tag{3.8}$$

where, $\underline{C}$ denotes the codeword, $\mathscr{C}$ denotes the complete codebook, and Σ denotes the covariance matrix of the AWGN noise. The detection rule decides in favor of $m_i = 0$ if $LLR\left(m_i\right) > 0$ and $m_i = 1$, otherwise. We use this detection rule to compute the BER plots for various dimming levels in the following section.

Mutual information is a quantity which is used to measure the relationship between two random variables that are sampled simultaneously. It measures how much information is communicated, on an average, in one random variable with respect to another random variable. Let $I(\underline{X};\underline{Y})$ be the mutual information between $\underline{X}$ and $\underline{Y}$. For single channel use it is given as

$$I(\underline{X};\underline{Y}) = h\left(\underline{Y}\right) - h\left(\underline{Y}|\underline{X}\right).$$

Table 3.3: $4b/6b$ code for γ=.625.

Message Vector	Codeword
0000	11001110
0001	00110111
0010	11010011
0011	01011011
0100	11010101
0101	10001111
0110	11100110
0111	10010111
1000	11011001
1001	01101011
1010	01110011
1011	11110001
1100	11001011
1101	10100111
1110	11101010
1111	11101100

Then, for n channel uses it is given as

$$I(\underline{X};\underline{Y}) = \frac{1}{n} h\left(\underline{Y}\right) - \frac{1}{n} h\left(\underline{Y}|\underline{X}\right) \ , \tag{3.9}$$

where, $h(.)$ denotes the differential entropy as in [27],

$$\begin{aligned} h(Y|X) &= h(X + N|X), \\ &= h(N). \end{aligned}$$

Entropy of a random variable is the amount of uncertainty associated with it. Given $f(y)$, the probability density function of $\underline{Y}$ and $\underline{X}$, $\underline{Y}$ being vectors of length n, (3.9) simplifies to

$$I(\underline{X};\underline{Y}) = \frac{1}{n} \left\{ E\left[log_2 \left(\frac{1}{f(y)} \right) \right] - \frac{1}{2} log_2((2\pi e)^n |\Sigma|) \right\} \ , \tag{3.10}$$

where, E[.] is the expectation function. Assuming uniform distribution for the input, the weight of the Gaussians in the Gaussian mixture at the output is equal. Hence, $f(y)$ becomes

$$f(y) = \frac{1}{(2\pi)^{n/2} 2^k |\Sigma|^{1/2}} \sum_{\underline{C} \in \mathscr{C}} e^{\left(\frac{-1}{2} (\underline{Y} - \underline{C}) \Sigma^{-1} (\underline{Y} - \underline{C})^T \right)} .$$

We present the simulation results for BER and mutual information computed numerically using (3.10).

3.4 Numerical results

For the simulation results, we consider standard $4b/6b$ code and present the results for various dimming factors.

Table 3.4: $4b/6b$ code for γ=.25.

Message Vector	Codeword
0000	001100010000
0001	110010000000
0010	000000101100
0011	101000000001
0100	011101110111
0101	010000001100
0110	000001100100
0111	000100100010
1000	000010011000
1001	001000010001
1010	100000000011
1011	000011100000
1100	001101000000
1101	010100000010
1110	010001000100
1111	000000010011

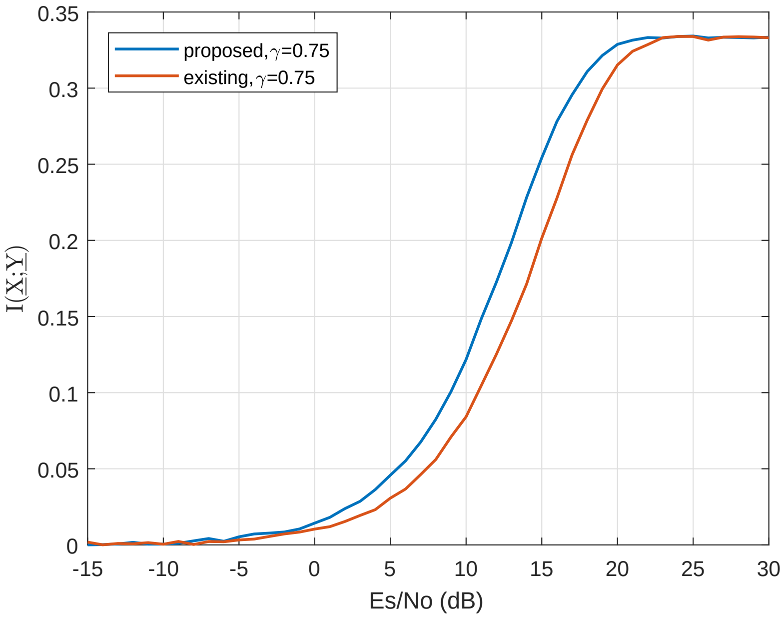

Figure 3.2: Mutual Information of existing and proposed methods for $4b/6b$ code for γ=0.75.

In Fig. 3.2, Fig. 3.3, Fig. 3.4, and Fig. 3.5, the numerically computed mutual information for the proposed and existing codes is presented for dimming factor γ equal to 0.75, 0.25, 0.375, and 0.625, respectively. The proposed code has higher mutual information than the existing code as observed from Fig. 3.2, Fig. 3.3, Fig. 3.4, and Fig. 3.5. It is seen that the mutual information plots saturates to coderate as the SNR increases.

Dimming factor effects coderate also. Given a perfectly DC balanced code with γ=0.5, there will be a decrease in the coderate as γ increases or decreases. This happens because of the addition

Table 3.5: $4b/6b$ code for γ=.375.

Message Vector	Codeword
0000	00110001
0001	11001000
0010	00101100
0011	10100100
0100	00101010
0101	01110000
0110	00011001
0111	01101000
1000	00100110
1001	10010100
1010	10001100
1011	00001110
1100	00110100
1101	01011000
1110	00010101
1111	00010011

Table 3.6: The d_{avg} and $f^{cw}_{d_{min}}$ for the proposed and existing codes.

Dimming Factor γ	d_{avg} proposed	d_{avg} existing	$f^{cw}_{d_{min}}$ proposed	$f^{cw}_{d_{min}}$ existing
0.250	4.8000	3.2000	0.2333	0.9333
0.375	3.7917	3.2000	0.6167	0.9333
0.625	3.7917	3.2000	0.6167	0.9333
0.750	4.8000	3.2000	0.2333	0.9333

of CS which increase the codeword length to achieve the desired dimming. For γ=0.75 and 0.25, coderate is 0.333 and for γ=0.375 and 0.635 it is 0.5.

The BER results for the existing and proposed codes using the LLR based detection rule presented in the previous section were generated for 10^6 realizations. The Monte Carlo simulations were performed in MATLAB. The BER performance of the proposed codes is significantly better than the existing codes as shown in Fig. 3.6. For VLC system with outer [15,11] Reed Solomon (RS) and inner 1/3 convolutional code (CC) as presented in [10] for γ=0.75, the BER performance of the proposed code is significantly better than the existing code. It is also be observed that if the CS length is smaller, increase in the d_{avg} is also smaller whose effect is observed in the BER performance. Even in the mutual information plots, the gap between the proposed and existing codes is smaller for γ=0.625 than for γ=0.75 as the length of CS for γ=0.625 is less when compared to γ=0.75.

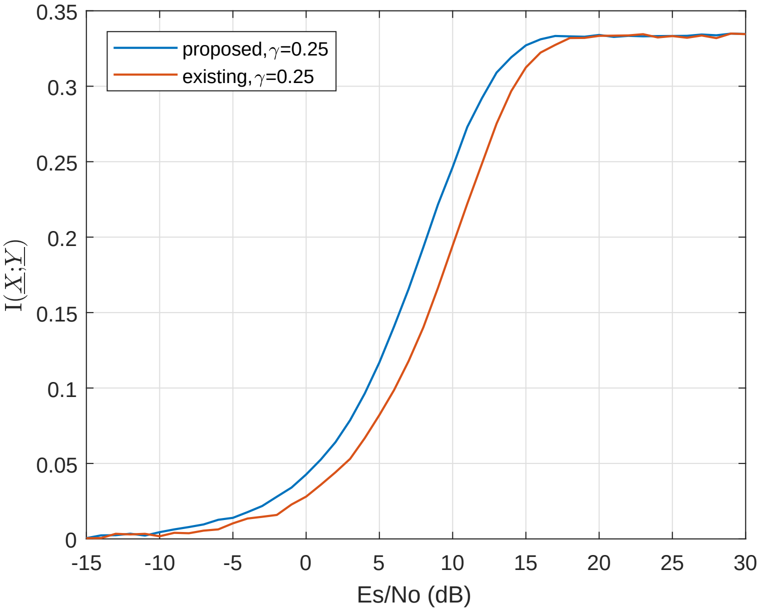

Figure 3.3: Mutual Information of existing and proposed methods for $4b/6b$ code for γ=0.25.

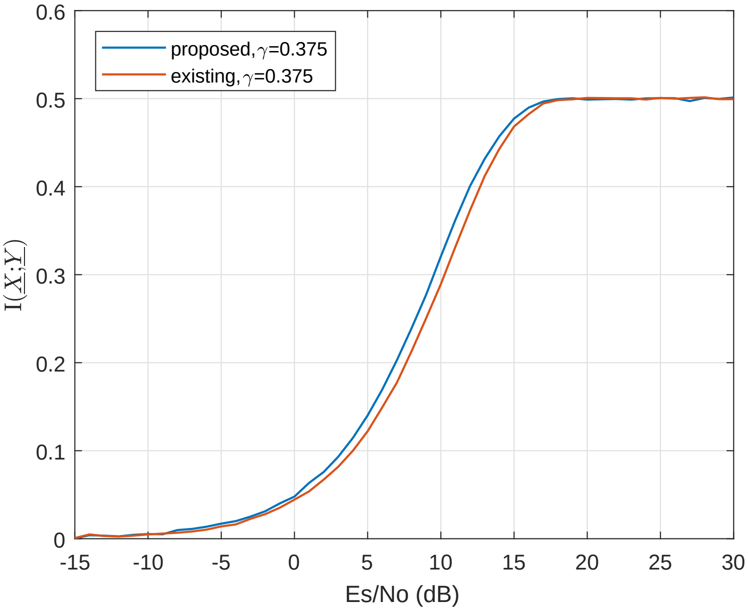

Figure 3.4: Mutual Information of existing and proposed methods for $4b/6b$ code for γ=0.375.

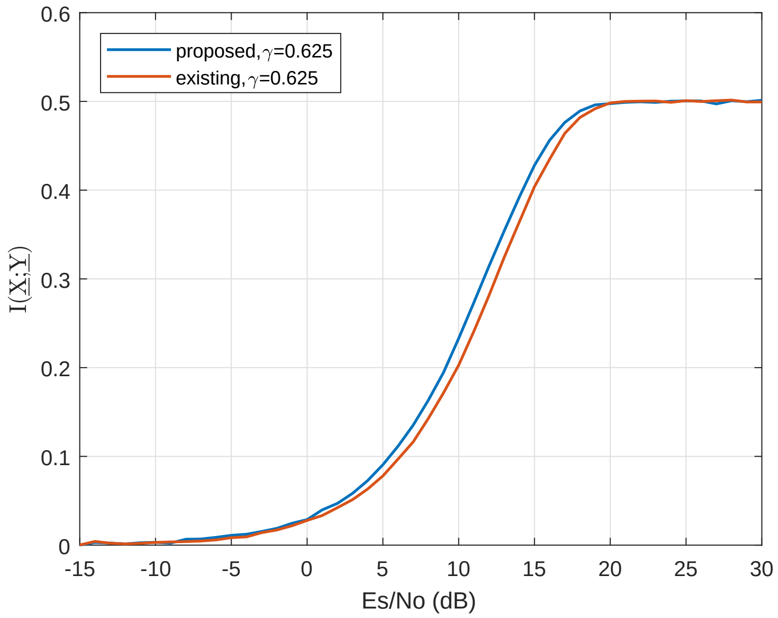

Figure 3.5: Mutual Information of existing and proposed methods for $4b/6b$ code for γ=0.625.

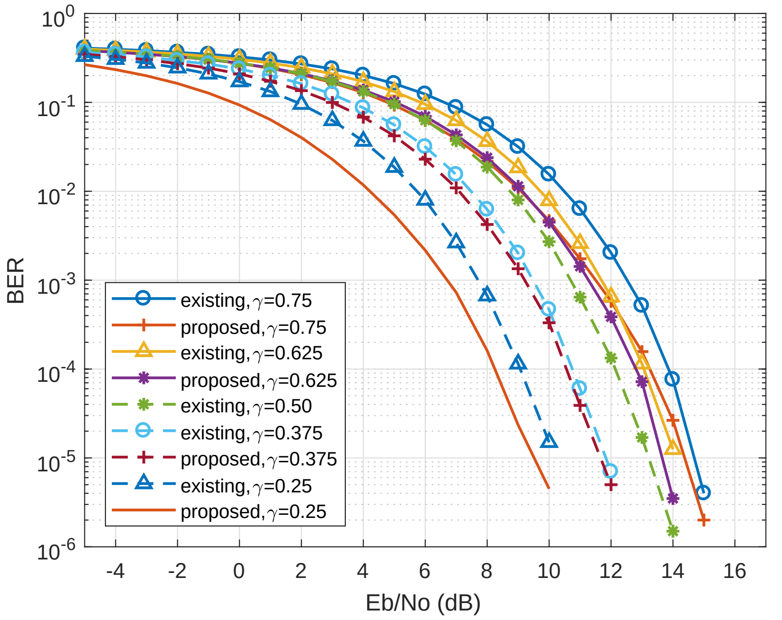

Figure 3.6: BER performance of $4b/6b$ code with existing and proposed methods by varying SNR.

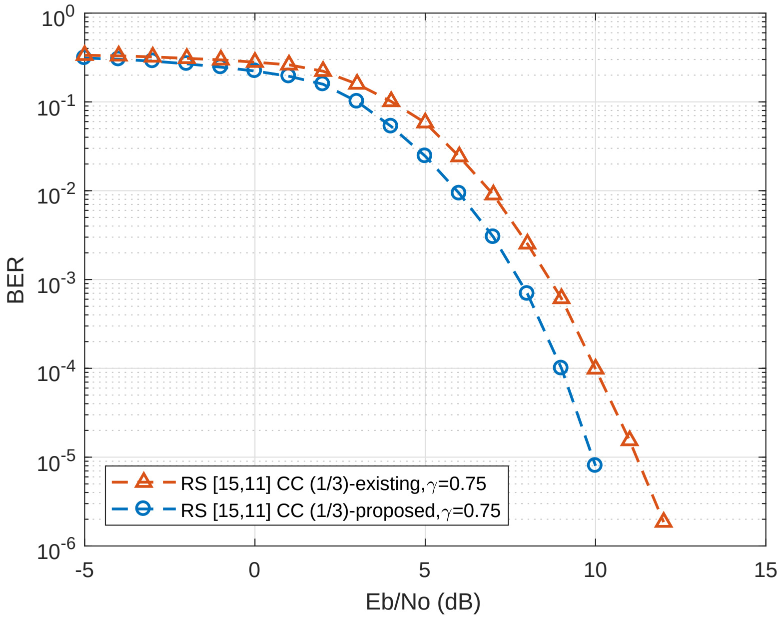

Figure 3.7: BER performance of $4b/6b$ code with existing and proposed methods for γ=0.75.

Chapter 4

High Rate RLL Codes

In this chapter, the proposed high rate RLL codes for VLC are discussed. The proposed codes are good in terms of rate, simplicity in encoding and decoding at the cost of BER performance. In this chapter, we will discuss system model considered to implement the proposed codes, algorithms for encoding and decoding of these proposed codes, performance metrics and numerical results.

4.1 System model

Let the high rate RLL code be of the format kb/nb where k is the length of the message vector and n is the length of the codeword. For the proposed high rate codes, n is equal to $k+1$. Hence, code rate denoted by R is given as

$$R = \frac{k}{k+1}. \tag{4.1}$$

In Table 4.1, the values of R for different values of k are presented for the proposed codes. We use variable pulse position modulation (VPPM) scheme for the proposed codes to show FER performance for different dimming factors. The system model implemented for UMLBI/CMLBI algorithm is presented in Fig. 4.1.

For a given k, the encoder of the proposed algorithm gives the RLL coded output which is then VPPM modulated. This VPPM modulated output is transmitted through the LEDs. At the receiver, the received signal is RLL decoded after hard threshold to get the binary data back. Here we assumed that code book is not available at the receiver. If it is assumed that the code book is available at the receiver, ML decoding can be performed which gives an improved performance than the method for decoding where hard thresholding is used.

4.2 Code design

A new method for high rate RLL code design for digital recording systems is proposed in [22] based on last bit inversion (LBI) method. These can be used in VLC systems. However, for the proposed method in [22], there exists message vectors which violate the maximum run length constraint. For such message vectors, block coding tables are provided which are to be kept at the transmitter and receiver whose size increases as the value of k increases. The method proposed in this thesis is a

Table 4.1: High rate codes.

k	Code	R
4	$4b/5b$	0.8000
5	$5b/6b$	0.8333
6	$6b/7b$	0.8571
7	$7b/8b$	0.8750
8	$8b/9b$	0.8889
9	$9b/10b$	0.9000
10	$10b/11b$	0.9091

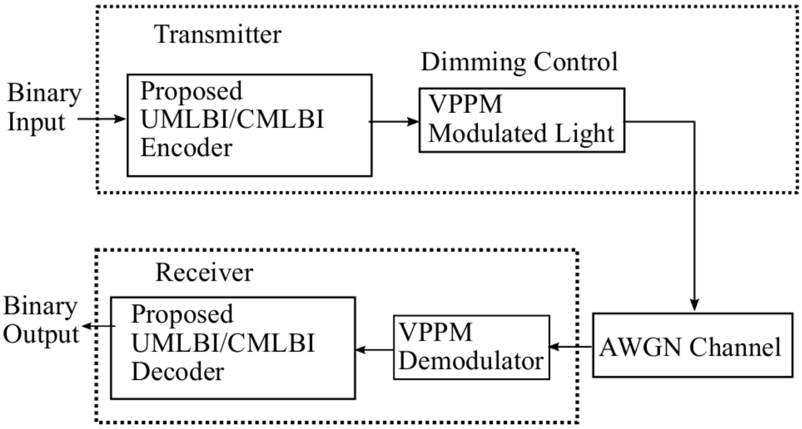

Figure 4.1: System model for high rate RLL codes

combination of UMLBI method [22] and CMLBI method. In this method, we reduce the run length of the codewords by increasing the weight of the codewords by putting a constraint on the weight of the message vectors. The range of the weight constraint, W_c on the codewords is given as

$$\left\lceil \frac{k}{2} \right\rceil \leq W_c \leq k \; for \; k \; odd \tag{4.2}$$

$$\frac{k}{2} \leq W_c \leq k \; for \; k \; even \tag{4.3}$$

Given the value of k we can select the value of W_c using (4.2) and (4.3). If the weight of the message vector is greater than or equal to W_c, then the codeword is message vector appended by the complement of the last bit at the tail of the message vector. For example, for a k length message vector $[m_1, m_2, m_3 \ldots, m_k]$, it is encoded as $[m_1, m_2, m_3 \ldots, m_k, \overline{m_k}]$. If the weight of the message vector is less than W_c that is for weight violating message vectors (WVM), the codeword is the complement of the message vector appended by the complement of the last bit at the end of the message vector. For example, for a k length message vector $[m_1, m_2, m_3 \ldots, m_k]$, it is encoded as $[\overline{m_1}, \overline{m_2}, \overline{m_3} \ldots, \overline{m_k}, \overline{m_k}]$. At the decoder after hard threshold, if the last two bits are same, it means

Algorithm 3 Proposed UMLBI/CMLBI algorithm for High rate RLL code generation with weight constraint

```
INPUT : Format of the code, $kb/(k+1)b$.
OUTPUT : Codebook of $kb/(k+1)b$ code.
Select the value of $W_c$ considering the constraint (4.2) and (4.3).
If the weight of the message vector is greater than or equal to $W_c$, then the codeword is message vector appended by the complement of the last bit at the tail of the message vector.
$Weight\ violation\ encoding(WVE)$: If the weight of the message vector is less than $W_c$ that is for weight violating message vectors (WVM), the codeword is the complement of the message vector appended by the complement of the last bit at the end of the message vector.
At the decoder after hard threshold, if the last two bits are same, it means that we have encoded it as per weight violation encoding. To decode this kind of received vector, discard the last bit ($(k+1)^{th}$ bit) and complement the first $k$ bits.
Else discard the last bit and take the remaining bits as decoded vector.
```

Table 4.2: Code book of the Proposed $4b/5b$ code using UMLBI/CMLBI algorithm.

Message vector	Codeword	Encoding principle
0 0 0 0	1 1 1 1 1	CMLBI
0 0 0 1	1 1 1 0 0	CMLBI
0 0 1 0	1 1 0 1 1	CMLBI
0 0 1 1	0 0 1 1 0	UMLBI
0 1 0 0	1 0 1 1 1	CMLBI
0 1 0 1	0 1 0 1 0	UMLBI
0 1 1 0	0 1 1 0 1	UMLBI
0 1 1 1	0 1 1 1 0	UMLBI
1 0 0 0	0 1 1 1 1	CMLBI
1 0 0 1	1 0 0 1 0	UMLBI
1 0 1 0	1 0 1 0 1	UMLBI
1 0 1 1	1 0 1 1 0	UMLBI
1 1 0 0	1 1 0 0 1	UMLBI
1 1 0 1	1 1 0 1 0	UMLBI
1 1 1 0	1 1 1 0 1	UMLBI
1 1 1 1	1 1 1 1 0	UMLBI

that we have encoded it as per weight violation encoding. To decode this kind of received vector, discard the last bit ($(k+1)^{th}$ bit) and complement the first k bits. For example, for the received vector after hard threshold be $[r_1, r_2, r_3, \ldots, r_k, r_k]$, then the decoded vector is $[\overline{r_1}, \overline{r_2}, \overline{r_3}, \ldots, \overline{r_k}]$. Else discard the last bit and take the remaining bits as decoded vector. For example, for the received vector after hard threshold be $[r_1, r_2, r_3, \ldots, r_k, \overline{r_k}]$, then the decoded vector is $[r_1, r_2, r_3, \ldots, r_k]$. Consider the codebook of $4b/5b$ code with W_c equal to 2 as given in Table 4.2. For encoding, the weight of the k length message vector is compared with the desired W_c. If it is greater than or equal to W_c then the codeword is formed using UMLBI method else it is formed with CMLBI method. For decoding the last two bits of the received vector are compared. If they are complements of each other, discarding the last bit, remaining bits are taken to be the decoded message. If they are same, discarding the last bit, remaining bits are complemented and taken to be the decoded message.

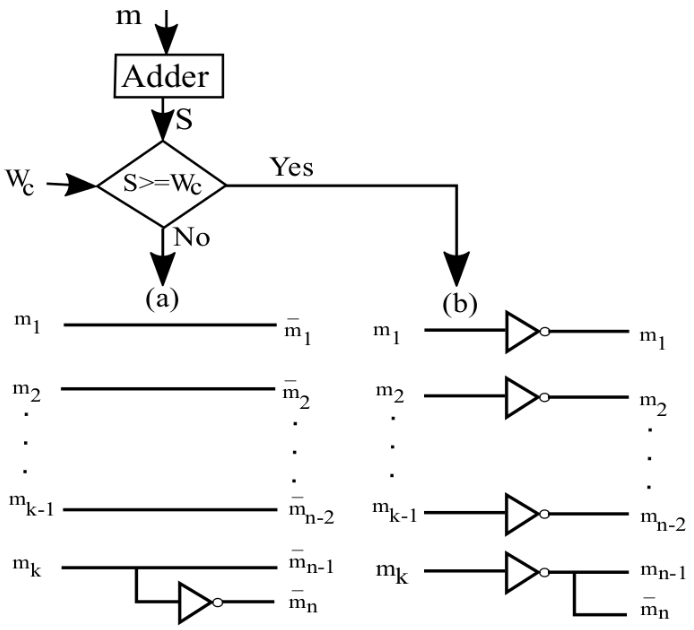

Figure 4.2: Encoder structure of the proposed UMLBI/CMLBI algorithm.

4.3 Performance metrics

In this chapter, the codes generated by the proposed CMLBI/UMLBI algorithm are compared using the following metrics

- Minimum Hamming distance
- Energy deviation in codewords
- Run-length
- Coderate
- Frame error rate

The d_{min} of the proposed high rate RLL codes is 1. This can be proved as follows. Let $\underline{m} = [m_1, m_2, m_3, \ldots, m_k]$ be a WVM be of length k. Then, based on the proposed algorithm, it is encoded as $\underline{C}_1 = [\overline{m_1}, \overline{m_2}, \overline{m_3}, \ldots, \overline{m_k}, \overline{m_k}]$. There will be a unique message vector satisfying W_c which is exactly same as the first k bits of the codeword corresponding to the WVM of length k and it's encoded vector is $\underline{C}_2 = [\overline{m_1}, \overline{m_2}, \overline{m_3}, \ldots, \overline{m_k}, m_k]$. Let $d(\underline{C}_i, \underline{C}_j)$ denote the Hamming distance between any arbitrary pair of codewords, then., $d(\underline{C}_1, \underline{C}_2) = d([\overline{m_1}, \overline{m_2}, \overline{m_3}, \ldots, \overline{m_k}, \overline{m_k}], [\overline{m_1}, \overline{m_2}, \overline{m_3}, \ldots, \overline{m_k}, m_k])$. Therefore, d_{min}=1.

The maximum possible run length in a codeword of the proposed codes is k/2 for k even and $\lceil \frac{k}{2} \rceil$ for k odd for W_c= $\lceil k/2 \rceil$. This can be proved as follows. Let a message vector, $\underline{m} = [m_1, m_2, m_3, \ldots, m_k]$ and W_c= $\lceil k/2 \rceil$. Then, any $\underline{m}$ whose weight is less than W_c is encoded by WVE. It means that no codeword has weight less than $k/2$ for the first k bits. Therefore, maximum run length is k/2 for k even. Similarly, we can show that maximum run length is $\lceil \frac{k}{2} \rceil$ for k odd.

Encoder and decoder structures of the proposed codes are given in Fig. 4.2 and Fig. 4.3, respectively. The run length of the proposed codes for different values of k is shown in Fig. 4.4. The

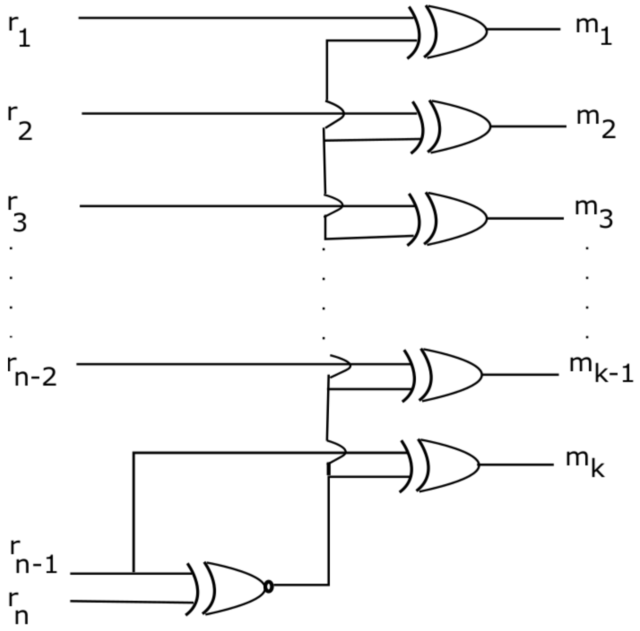

Figure 4.3: Decoder structure of the proposed UMLBI/CMLBI algorithm.

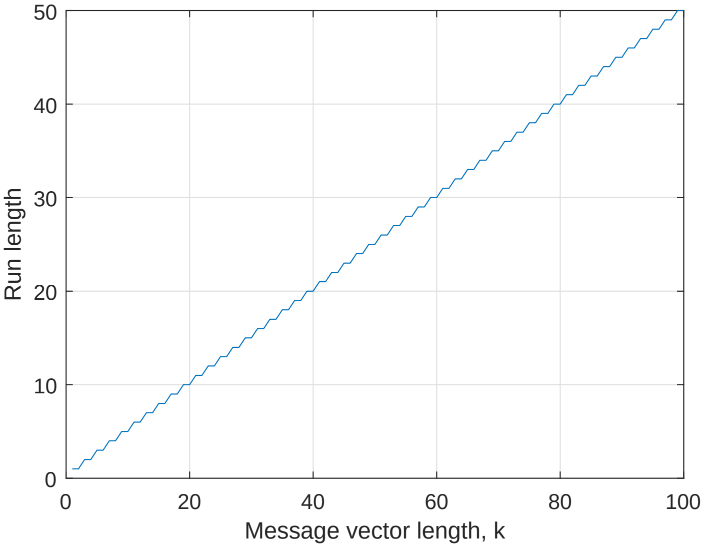

Figure 4.4: Run length of the proposed UMLBI/CMLBI codes for various values of k.

variations in the energy of the codeword instants and the average codeword energy is shown in Fig. 4.5 for k equal to 23 which shows that the deviation is less and can be ignored. From Fig. 4.5, it is seen that the deviation of instantaneous codeword energy form average energy of the codeword is less. For simulation results, In Fig. 4.6, we consider k equal to 4, 23, 100, and 500 to show the FER performance. The FER performance for different dimming levels is shown for k equal to 100 in Fig. 4.7. VLC systems can provide bit rates in the range of 11.67 Kbps to 96 Mbps as seen in [10]. Then, for a codeword length of n bits and bit period T_b, to avoid flickering $nT_b < 5$ ms which is written as $n < R_b$ 5 ms, where, R_b is the bit rate. Then, for R_b=11.67 Kbps, $n < 58$, and for

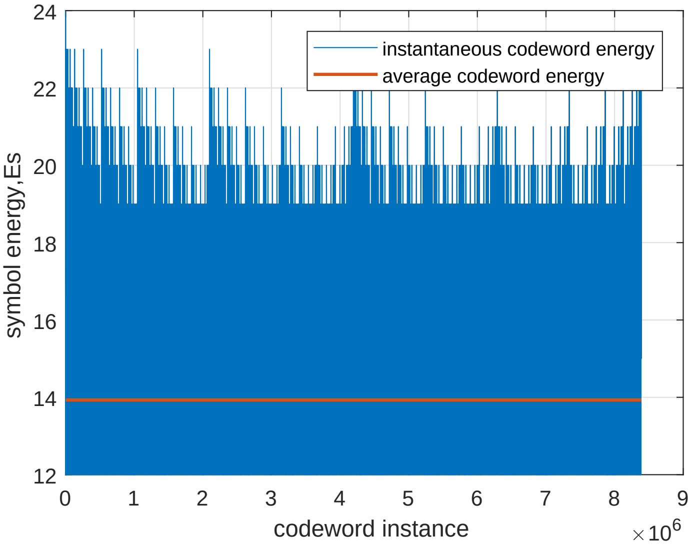

Figure 4.5: Instantaneous versus average codeword energy of proposed UMLBI/CMLBI code for k=23.

R_b=96 Mbps we will have n <480000. Hence, even in the worst case the proposed method can be used till k=116.

4.4 Numerical results

In this section, the FER performance of the proposed codes is presented in Fig. 4.6. it is seen that as the rate increase, the FER performance of the codes is becoming worse. In Fig. 4.7, the performance of the proposed codes is shown for k = 100 for various dimming factors.

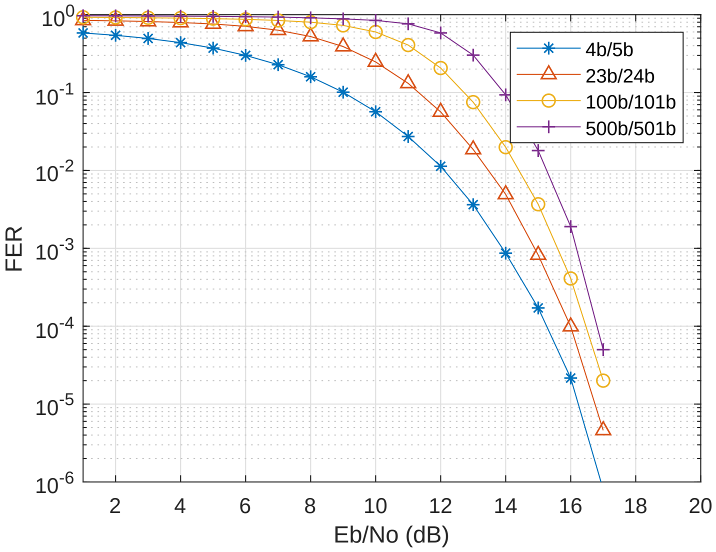

Figure 4.6: FER performance of proposed UMLBI/CMLBI code with varying SNR.

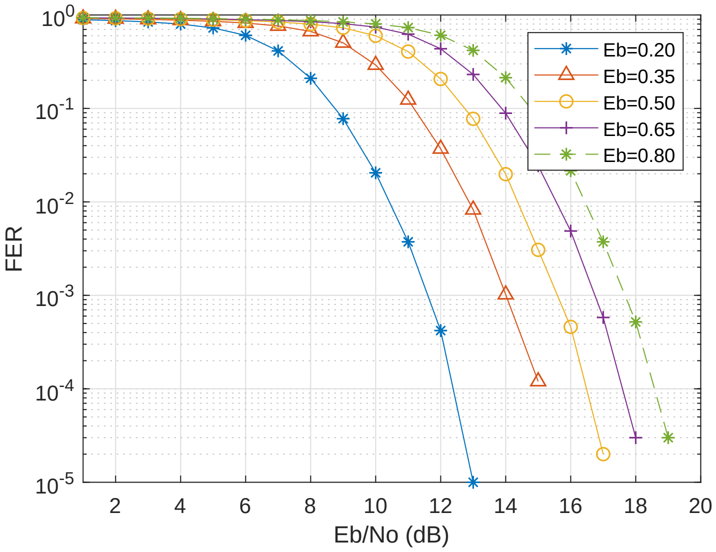

Figure 4.7: FER performance of proposed UMLBI/CMLBI code with varying SNR for k=100.

Chapter 5

Conclusions

In this work, we derived the condition to generate perfectly DC balanced codes of the type kb/nb for various even values of n. We proposed an algorithm to generate the codebook for a perfectly DC balanced kb/nb code. We compared the performance of the proposed perfectly DC balanced codes with some of the existing codes in the literature and showed that the proposed codes provide a better trade-off between FER, BER, and code rate compared to the existing VLC codes. A lower bound on d_{avg} of the proposed has also been derived for the proposed perfectly DC balanced codes. Further, we proposed novel RLL codes for VLC that utilize the CS to improve the BER performance. We showed through extensive Monte-Carlo simulations that the proposed codes have better BER performance and mutual information compared to existing codes in the VLC standard. High rate codes for VLC were proposed with simplified encoding and decoding.

References

[1] D. Karunatilaka, F. Zafar, V. Kalavally, and R. Parthiban, “LED Based Indoor Visible Light Communications: State of the Art,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1649–1678, third quarter 2015.

[2] A. M. Cailean and M. Dimian, “Current Challenges for Visible Light Communications Usage in Vehicle Applications: A Survey” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2681–2703, Fourthquarter 2017.

[3] N. Kumar, N. Loureno, D. Terra, L. N. Alves and R. L. Aguiar, “Visible light communications in intelligent transportation systems,” in *Proc. IEEE Intelligent Vehicles Symposium*, Alcala de Henares, 2012, pp. 748–753.

[4] N. Farr, A. Bowen, J. Ware, C. Pontbriand, and M. Tivey, “An integrated, underwater optical/acoustic communications system,” in *Proc. of IEEE OCEANS*, 2010, pp. 1–6.

[5] R. Shaaban and S. Faruque, “A survey of indoor visible light communication power distribution and color shift keying transmission,” in *Proc. IEEE International Conference on Electro Information Technology (EIT)*, Lincoln, NE, 2017, pp. 149–153.

[6] S.P. Ying, C.-W. Tang, and B.-J. Huang, “Characterizing LEDS for mixture of coloredled light sources,” in *Proc. International Conference on Electronic Materials and Packaging (EMAP)*, pp. 1–5, 2006.

[7] M. Mukherjee, “Visible light communication-A survey of potential research challenges and advancements,” in *Proc. International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Coimbatore, 2017, pp. 1–8.

[8] C.W. Chow, C.H. Yeh, Y.F. Liu, and Y. Liu, “Improved modulation speed of led visible light communication system integrated to main electricity network,”*Electronics Letters*, vol. 47, no. 15, 2011, pp. 867–868.

[9] T. Komine and M. Nakagawa, “Fundamental analysis for visible-light communication system using LED lights,” *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 100–107, Feb. 2004.

[10] R. D. Roberts, S. Rajagopal, and S. K. Lim, “IEEE 802.15.7 physical layer summary,” in *Proc. IEEE GLOBECOM Workshops (GC Wkshps)*, Houston, TX, 2011, pp. 772–776.

[11] J. Duan, A. Shi, and Y. Liu, “A practical indoor visible light communication system,” in *Proc. International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP)*, Manchester, 2014, pp. 1170–1175.

[12] K. Kim, K. Lee, and K. Lee, “Appropriate RLL coding scheme for effective dimming control in VLC,” *Electronics Letters*, vol. 52, no. 19, pp. 1622–1624, Sep. 2016.

[13] H. Wang and S. Kim, “Dimming Control Systems With Polar Codes in Visible Light Communication,” *IEEE Photonics Technol. Lett.*, vol. 29, no. 19, pp. 1651–1654, Oct. 2017.

[14] Harald Burchardt, Nikola Serafimovski, Dobroslav Tsonev, Stefan Videv, and Harald Haas, “VLC: Beyond point-to-point communication,” *Communications Magazine*, vol. 52, no. 7, pp. 98–105, 2014.

[15] John E. Gancarz, H Elgala, and Thomas D.C. Little, “Overlapping PPM for band-limited Visible Light Communication and Dimming”, *Journal of Solid State Lighting*, Springer, 2015.

[16] L Kwonhyung and H Park, “Modulations for visible light communications with dimming control,” *IEEE Photonics Technology Letters*, vol. 23, no. 16, pp. 1136–1138, 2011.

[17] M. Anand and P. Mishra, “A novel modulation scheme for visible light communication,” in *Proc. Annual IEEE India Conference (INDICON)*, Kolkata, pp. 1–3, 2010.

[18] Ron M. Roth, *Introduction to Coding Theory*, Cambridge, United Kingdom, 2006.

[19] J. M. Griffiths, “Binary code suitable for line transmission,” *Electronics Letters*, vol. 5, no. 4, pp. 79–81, Feb. 1969.

[20] T. Uday, A. Kumar, and L. Natarajan, “Generation of perfectly DC balanced codes for visible light communications,” in *Proc. National Conference on Communications (NCC)*, Hyderabad, India, 2018, pp. 1–5.

[21] S. Rajagopal et al., “IEEE 802.15.7 visible light communication: modulation schemes and dimming support,” *IEEE Commun. Mag.*, vol. 50, no. 3, pp. 72–82, Mar. 2012.

[22] L. Scripcariu et al., “A new method for high-rate RLL codes design,” Signals, Circuits and Systems, SCS 2003. International Symposium on, 2003, pp. 645-648 vol.2.

[23] J. H. Yoo, B. W. Kim, and S. Y. Jung, “Modelling and analysis of M-ary variable pulse position modulation for visible light communications,” *IET Optoelectronics*, vol. 9, no. 5, pp. 184–190, Oct. 2015.

[24] S. Kim and S. Y. Jung, “Novel FEC Coding Scheme for Dimmable Visible Light Communication Based on the Modified ReedMuller Codes,” *IEEE Photonics Technology Letters*, vol. 23, no. 20, pp. 1514–1516, Oct. 2011.

[25] S. Kim, “Adaptive FEC Codes Suitable for Variable Dimming Values in Visible Light Communication,” *IEEE Photonics Technology Letters*, vol. 27, no. 9, pp. 967–969, May 2015.

[26] T. Uday, A. Kumar, and L. Natarajan, “Improved run length limited codes for VLC using dimming control compensation symbols,” in *Proc. International Conference on Communication Systems & Networks (COMSNETS)*, Bengaluru, India, 2018, pp. 457–460.

[27] T. Cover and J. Thomas, “Elements of Information Theory,” 1991, Wiley Series in Telecommunications.

[28] Albert X. Widmer,“DC-balanced 6B/8B transmission code with local parity,” US Patent, US6876315 B1, Apr. 2005.