PRACTICAL CHANNEL CODING METHODS FOR CHANNELS WITH INPUT-DEPENDENT NOISE

by

Mehmet Görkem Ülkar

B.S., Electrical and Electronics Engineering, Middle East Technical University, 2010

M.S., Communications Systems, Lund University, 2012

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2020

ACKNOWLEDGEMENTS

I am very grateful to be guided and advised by Assoc. Prof. Ali Emre Pusane. Not only this dissertation is immensely benefited from his technical brilliance; I, myself, gained a lot from his perspective on the academia and life. I am very thankful for his continuous support, patience and motivation during my studies. Definitely, knowing him and working with him is my biggest personal gain from this PhD study.

I would like to thank Asst. Prof. Tunçer Baykaş and Prof. Tuna Tuğcu for their insightful comments and contributions on the milestones of my research. I am very grateful for having Prof. Emin Anarim in my PhD committee as his comments and support have always been very valuable for my studies. I would like to thank Asst. Prof. Farshad Miramirkhani for participating in my jury.

I also thank my colleagues and friends Dr. Batuhan Gündoğdu and Dr. Selda Uyanık who have given great motivation and support other than their invaluable friendship.

I have deep gratitude to my mother Nurhan Ülkar and my father Salih Ülkar for their belief in me, and infinite support. Their values guided me throughout my life. I also thank my beloved brother Göktuğ for his continuous support besides always being the best friend.

Lastly, I am thankful to Seda for her love, patience and motivation.

ABSTRACT

PRACTICAL CHANNEL CODING METHODS FOR CHANNELS WITH INPUT-DEPENDENT NOISE

With the emergence of new applications and needs, communications theory started to be applied to outside of the traditional radio frequency (RF) bands or even in completely different channels. Molecular communication and visible light communication (VLC) are examples of such emerging use cases that communication takes place in a specific channel. Not surprisingly, those channels bring their own challenges and differences compared to the conventional wireless channels. One common point among many of those new channels is that the noise depends on the input signal. This situation is contrary to the prevalent assumption of existence of white noise in the design of wireless communication blocks. Since white noise assumption is not valid, applying directly conventional methods to the new channels yields unsatisfactory performances. In this thesis work, our aim was to develop practical channel coding methods for the channels with input dependent noise. For molecular communication via diffusion (MCvD), we propose 2 novel decoding methods coupled with constant low weight codes. Iterative sorting decoder is a decision-feedback heuristic method that iteratively calculates the intersymbol interference (ISI) from a better estimation at each step. The second proposed method is the super trellis decoder, which is a maximum a posteriori sequence estimator. Iterative sorting and super trellis decoders bring substantially better performances than the existing methods. For VLC, a deep learning based VLCnet is proposed. VLCnet has a novel activation unit, FRAU, to achieve flicker reduction and dimming, which are two main illumination needs. By allowing joint optimization of both the encoder and decoder, VLCnet performs superior compared to the other proposed techniques in the literature.

ÖZET

GİRDİ BAĞIMLI GÜRÜLTÜYE SAHİP KANALLAR İÇİN PRATİK KANAL KODLAMA YÖNTEMLERİ

Yeni uygulamalar ve ihtiyaçların ortaya çıkması ile haberleşme teorisi geleneksel radyo frekans (RF) bantlarının dışında veya tamamen farklı kanallara uygulanmaya başlandı. Moleküler haberleşme ve görünür ışık haberleşmesi (VLC), haberleşmenin özel bir kanalda gerçekleştiği bu tarz yeni kullanımlara örnektir. Şaşırtıcı değildir ki, bu kanallar geleneksel kablosuz kanallara kıyasla kendi zorluklarını ve farklılıklarını beraberinde getirmektedirler. Bu kanalların çoğundaki ortak nokta gürültünün girdi sinyaline bağlı olmasıdır. Bu durum, kablosuz haberleşme blokları tasarımındaki beyaz gürültü varlığı varsayımına zıttır. Beyaz gürültü varsayımı geçerli olmadığı için geleneksel yöntemlerin direkt olarak yeni kanallara uygulanması istenilen düzeyde olmayan performans getirmektedir. Bu tez çalışmasında, amacımız girdi bağımlı gürültüye sahip kanallar için pratik kanal kodlama yöntemleri geliştirmekti. Difüzyon bazlı moleküler haberleşme için sabit düşük ağırlıklı kodlar ile ilgili 2 farklı yeni kod çözme yöntemi önermekteyiz. Özyineli sıralama kodçözücüsü, her adımda daha iyi kestirimden özyineli olarak semboller arası girişimi hesaplayan bir karar geribeslemeli buluşsal yöntemdir. Bir maksimum soncul dizi kestirimi olan süper trellis kodçözücüsü, ikinci önerilen yöntemdir. Özyineli sıralama kodçözücüsü ve süper trellis kodçözücüsü var olan yöntemlerden önemli ölçüde daha iyi performans sağlamaktadır. VLC için de derin öğrenme bazlı VLCnet önerilmektedir. VLCnet iki temel aydınlatma ihtiyacı olan ışık titremesi azaltma ve karartmayı gerçekleştirmek için yeni aktivasyon birimi olan FRAU'ya sahiptir. VLCnet kodlayıcı ve kodçözücünün birlikte optimizasyuna fırsat tanıyarak literatürdeki diğer önerilen tekniklerden üstün performans göstermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

b^i	Bit sequence
c	Number of active neurons in each FRAU block
d_{θ_e}	Dimming of the VLCnet codewords
D	Diffusion coefficient
$\boldsymbol{e_m}$	All zero vector except its m^{th} element being 1
E_b	Bit energy
E_s	Average active symbol energy in a codeword
$f(.)$	Encoder function of the autoencoder
$F_{hit}(t)$	Hitting probability of a molecule to the receiver until time t
$g(.)$	Decoder function of the autoencoder
$\boldsymbol{h}$	Channel impulse response
k	Message word length
K	Number of linear layers in VLCnet decoder
l	Block length of a FRAU
L	ISI channel length
$\mathcal{L}_{CCE}$	Categorical cross entropy loss
m	Codeword weight
M	Number of input messages
n	Codeword length
n_{sh}	Shot noise
n_{th}	Thermal noise
N	Number of released molecules at each $b[i] = 1$
$\mathcal{N}$	Normal distribution
$\mathcal{O}$	Worst case complexity indicator
p_i	Hitting probability of a molecule in i^{th} time slot
P_e	Probability of error
q	Weight state
r_0	Distance between the transmitter and receiver

r_r	Spherical receiver radius
$\boldsymbol{s}$	Input vector of a FRAU
t_s	Sampling period
$\boldsymbol{u}$	Message word
$\boldsymbol{v}$	Codeword
$\boldsymbol{w}$	Equalizer coefficients
x	Super trellis state
y	Received signal
z	Hidden representation in an autoencoder
δ	MAP branch metric
η	Learning rate
γ	Threshold for binary detection
Γ	Set of active neurons in a FRAU
λ	Bias value for active FRAU neurons
∇_θ	Gradient operator with respect to θ
Ω	Neuron count of a single linear layer in VLCnet decoder
σ_c^2	Counting noise variance
θ_e	Encoder network parameters
ς^2	The comparative strength of the input-dependent n_{sh} over the input-independent n_{th}

LIST OF ACRONYMS/ABBREVIATIONS

BCSK	Binary Concentration Shift Keying
BER	Bit Error Rate
CC	Convolutional Codes
CIR	Channel Impulse Response
CLWC	Constant Low Weight Code
CSI	Channel State Information
CSK	Concentration Shift Keying
CWC	Constant Weight Code
DCWC	Doubly Constant Weight Code
FEC	Forward Error Correction
FER	Frame Error Rate
FIR	Finite Impulse Response
FRAU	Flicker Reducing Activation Unit
ISI	Intersymbol Interference
LDPC	Low-Density Parity Check
LED	Light Emitting Diode
LWTA	Locally Winner Takes All
MAP	Maximum a Posteriori
MCvD	Molecular Communication via Diffusion
ML	Maximum Likelihood
MOCSK	Molecular Concentration Shift Keying
MoSK	Molecular Shift Keying
MTSK	Molecular Transition Shift Keying
OOK	On-Off Keying
OWC	Optical Wireless Communications
PD	Photo-Detector
PPM	Pulse Position Modulation
PWM	Pulse Width Modulation

ReLU	Rectified Linear Unit
RF	Radio Frequency
RLL	Run Length Limited
RS	Reed Solomon
SNR	Signal-to-Noise Ratio
Tbps	Terabit-per-second
THz	Terahertz
VLC	Visible Light Communications
VOOK	Variable On-Off Keying

1. INTRODUCTION

As social beings, humankind has sought for ways to improve their ability to exchange information for a long time of its existence. Regarding the modern era communications, most of its interest has been related to increase communication distance, capacity and security. With the invention of communication using radio frequencies, the speed of technical improvements on the area have increased significantly. The communication blocks have evolved and been optimized by the researchers considering the requirements of the wireless channels. For long, those requirements have stayed stable and more superior methods have been developed based on the same models and assumptions.

Communications has been moving to a different era more recently. To be able to realize future technologies, such as revolutionary data speeds, intelligent drugs, or much more accurate indoor localization, communication theory is applied to different contexts and channels, which none of which were researched a couple of decades ago. Although research interest has been focused on other essential communication modules in these emerging channels, error correcting codes specialized to the nature of these channels will likely gain popularity in the future. As stated in [1], only a limited amount of work has been done on molecular communication channel coding, which is an example of the emerging channels.

As they are researched, these channels are found to have some challenges, which make communications harder compared to the familiar ones. This difficulty stems from the fact that these challenges were not considered by the communications researchers before and most of the available systems were optimized considering nature of the conventional wireless channel. The major discrepancy between these channels and the conventional wireless channels is the noise model. The noise contains an input-dependent part unlike the widespread white noise model in conventional wireless channels. The proposed thesis aims to develop implementable, performance increasing

error correcting codes specialized to the challenges caused by the nature of these emerging channels. Molecular communications channel, visible light communications (VLC) channel as well as the terahertz (THz) band channel are some examples of these comparatively new channels with input-dependent noise. These channels and the nature behind the their input-dependent noise are shortly presented in Sections 1.1, 1.2, 1.3, respectively. Other channel specific requirements that must be met by custom channel codes are also mentioned in these sections. The detailed related literature reviews are left to the corresponding chapters.

1.1. Molecular Communications Channel

Molecular communication is a type of communication where information is transmitted with molecules [2]. The information can be encoded in the form of molecule count, molecule release time or molecule type [1]. The molecules that carry message information in their release pattern propagate through the receiver. The receiver senses the molecules and decodes the pattern so the message. It is seen as a feasible communication scheme in the communications community for small scaled devices in fluid medium, which have short distances in between [3]. Design simplicity compared to the RF modules at nano-scale, being less energy demanding compared to electromagnetic wireless communications, and biocompatibility are the main attractions of molecular communications for researchers [4]. These devices are scaled in nanometers and electromagnetic communication in these scales are difficult due to pyhsical limitations such as the required antenna sizes. Therefore, molecular communication is envisaged to solve the networking problem of the nano-machines. More intelligent drugs or nano-scale diagnosis are likely to be realizable using molecular communications [5].

Many biological systems like microorganisms exchange information using molecular communication [6]. Engineered molecular communication systems are inspired mainly by nature and major methods in literature are achieving transmission using diffusion based architecture [7] or alternatively using microtubules and molecular motors [8]. In both cases, electromagnetic wave based communications paradigms

will need to be reconsidered before directly applying to the communication based on molecules.

Molecular communication via diffisuon (MCvD), also known as communication with Brownian motion, is the most popular and most general architecture in the molecular communications community. In this scheme, the molecules are released to the fluid medium and they are subject to the diffusion process afterwards [5]. In the case of communication via diffusion, the major difference compared to the classical communication is that additive Gaussian noise variance in molecular communications depends on the transmitted sequence [9]. This is because the main source of noise is considered to be random movement of the information particles. Therefore the few molecules are released in the transmitter, the less noise is received by the receiver.

In diffusion, some molecules that have been released at the transmitter side may arrive in receiver in unintended time slots [7]. This causes intersymbol interference (ISI).

Considering the fact that releasing fewer molecules reduces energy consumption by synthesizing fewer molecules at the transmitter and it also decreases input dependent noise and ISI, it seems feasible to map the information bits to codewords which have more zeros compared to the ones. A novel method built upon this idea is further discussed in Chapter 2.

Other challenge is production of energy and maintaining communications with limited energy for these small scaled devices. Energy is a constraint for the proposed methods in this channel. Similarly, computation needs of the methods should be consistent with the capabilities of these small scaled devices.

1.2. Visible Light Communications Channel

Visible light communications is an application of optical wireless communications (OWC) which generally utilizes light emitting diodes (LED) as optical transmitters. Illumination and communication are provided simultaneously. Because of the RF spectrum scarcity and the developments in LED technology, VLC has been increasingly appealing for communications community. Some of its other advantages are the fact that the frequency is license free, both photodetectors (PDs) and LEDs are low-cost and this type of communication does not cause radio frequency interference. Electronic medical equipments is especially adversely affected by electromagnetic interference and VLC is envisaged to be used in hospitals in the future.

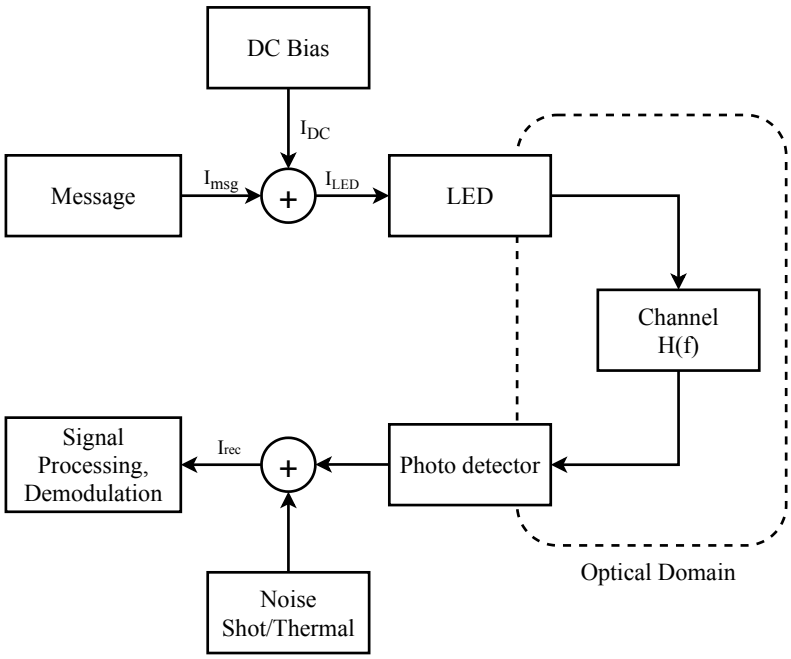


Figure 1.1: VLC Link

For having practical high data rate VLC systems, a technique called intensity modulation/direct detection (IM/DD) is used. In IM/DD, the signal is modulated into the the power of the LED. A typical VLC link model with IM/DD is seen in Figure 1.1. The message signal, I_{msg}, is summed with DC level, I_{DC}, to generate LED driving current, I_{LED}. This current is transformed to amplitude modulated optical power. This power is traveled through the free space optical channel and then transformed back to

electrical current by the photodetector. At PD, the received signal is mixed with the shot and thermal noise. Shot noise is due to random nature of photon collection in the LED and depends on the input itself [10]. Thermal noise is a kind of electronic noise and considered to be input-independent. Thus, the noise model of VLC consists of both input-dependent and independent noise terms.

Other main challenge in VLC is to achieve light dimming [11]. This is generally achieved by either compensation symbols or pulse width modulation (PWM) if on-off keying (OOK) is used as data modulation scheme. Another illumination related challenge is to mitigate flickers. Flicker means the situation when light intensity change is noticeable by human eye [12]. It is stated in [13] that flicker can cause harmful physiological changes to humans, therefore it should be avoided. Long runs of bit-1 values or bit-0 values make humans notice the change between these levels. A technique to avoid flickers is explained in 1.4.2.

1.3. Terahertz Band Channel

With the increase in demand for network speed and capacity, terahertz band communication is gaining more popularity. Researching the terahertz band and utilizing other unused parts of the spectrum is seen as one of the key enabler of future's ultra fast wireless technology, since spectral efficiency of used spectrum has already been increased to a great extend. In very near future, terabit-per-second (Tbps) speeds will be needed to satisfy increased demand. Frequencies below 0.1 THz are not able to support Tbps speeds [14]. Therefore communications systems working on the terahertz band will be the only way to realize these large bandwidths. Another use case of the terahertz band is nanonetworks. Due to limitations of these devices, the terahertz band is considered as a viable solution for the networking of these devices [14].

Although increasing bandwidth for satisfying high capacity may seem to be a trivial solution, challenges caused by the nature of these high frequencies have prevented the band from the usage until now.

Besides the hardware difficulties [14] and the severe path loss, the peculiarities of the channel at these high frequencies cause trouble for the communications. In addition to the spreading loss, which means traveling wave's expansion as its propagation, molecular absorption loss exists as path loss in the terahertz band. Excited molecules absorb some portion of the energy of the electromagnetic waves in terahertz band and radiate this energy later. This radiated signal is denoted as molecular absorption noise. This form of noise only exists when channel is excited; if no user is transmitting, molecules remain still and noise becomes negligible. Thus, by being silent, the energy consumption is reduced (assuming OOK modulation), and the probability of incorrect symbol detection is also lowered. Considering the nano-machine communications being one of the target applications for the terahertz band, diminishing energy consumption is obviously a merit for any designed communication block.

1.4. Error Correcting Codes for Challenging Channels

Well known channel coding methods optimized for decades assume the existence of white Gaussian noise at the channel while considering less about the energy or computation power if not none at all. Because of the channel peculiarities and capability issues of the targeted devices, there is a need to develop special channel codes for the challenging channels described earlier. Firstly, designed codes should not be complex to operate if molecular channel is aimed since devices are generally nano-scaled in this channel. Similarly, decoding time and decoding power are constraints in the terahertz band. It is known that for small distances, decoding power is sometimes larger than transmit power [15].

For the energy constraint of nanonetworks, OOK is proposed as a modulation method as bit-0 is represented with silence in [16], [17]. Encoding bit-0 with silence is further advantageous considering the fact that molecular absorption loss is negligible when there is no transmission in terahertz band [16]. In a similar fashion, the use of low weight channel codes have been proposed in [16], [17], [18]. In this way, probability of error occurring is diminished. So, these codes are defined as error preventing rather

than error correcting by the authors. The main principle for achieving interference and error mitigation is to diminish the number of 1s transmitted. Channel coding proposed in [16] controls the weight of the codewords and by choosing the low weight ones as valid codewords, the interference power is reduced. Unlike existing channel codes that are composed of different weight codewords, authors claim that weights of the codewords should be limited to low values to reduce interference. In [19], the low weight channel code suggested originally for the terahertz band [17] is proposed for molecular channel for its energy efficiency.

In addition to high weight codewords, the authors in [20] point out that consecutive bit-1s are also detrimental as they do not allow channel relaxation, thus increasing molecular absorption noise in the terahertz band. This reasoning is valid for MCvD as well. Long runs of same symbols are also avoided in VLC in order to reduce flicker. In the following sections, the short descriptions of two different coding methods are summarized as they are referred often in the thesis.

1.4.1. Constant Weight Coding

A binary code is defined as constant weight code, if all of its codewords have the same total number of bit-1 samples. Constant weight codes have numerous applications as error detection or correction codes. Most of the research has been done to find bounds of number of constant weight codes for given weight and Hamming distance values.

In the previous part, it is mentioned that having low weight codes is favorable for the interested channels. By using a longer sequence of bits, the number of ones in the uncoded stream can be reduced. Total number of possible k-bit words is 2^k. If we have $n > k$, we can have $\frac{n!}{(n-m)!m!}$ different words with weight m. For instance, for $k = 32$ bit input words, $n = 35$ long sequences are needed to have 2^{32} words with 17 weight. Then the probability of ones is $\frac{17}{35}$, which is slightly smaller than 0.5. By decreasing the m parameter, the count of bit-1's in a codeword can be reduced to have low weight

codewords.

Figure 1.2 shows how the codeword length changes as the fixed weight of codewords, m parameter, changes. If very low weight is desired, the required codeword length to encode messages increases dramatically. For $k = 12$, n increases to 92 if m is lowered to be 2.

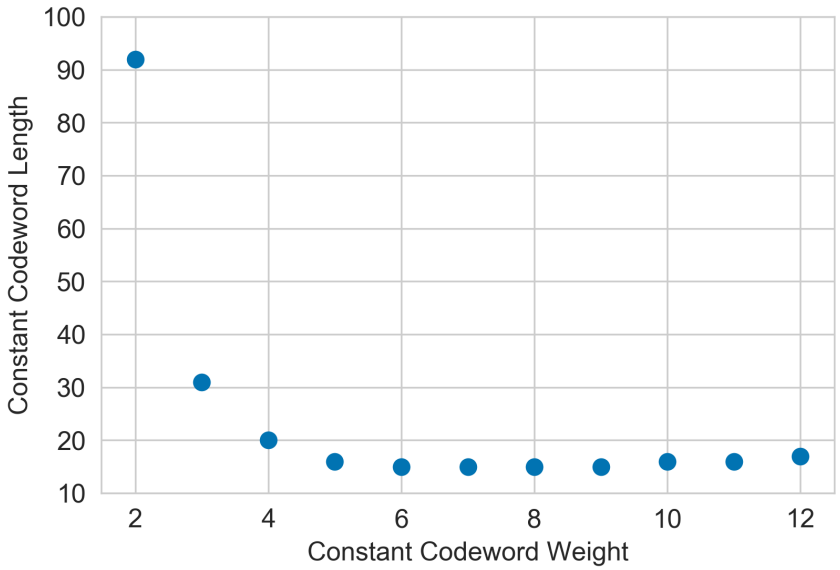


Figure 1.2: Constant codeword weight and associated codeword length to encode 12 information bits

As expected, the amount of useful information decreases the as the codewords are lengthened due to having lower weights. Thus, lowering weights constitutes a trade-off between decreased spectral efficiency and the advantages of low weight codes, namely less input-dependent noise, less energy.

Effect of constant weight codewords has been tested in molecular communications channel. For the n and m pairs shown in 1.2, molecular communication decision thresholds were found by simulations. Table 1.1 shows these values for different number of molecules released when 1 is transmitted. Threshold receiver is explained in 2.3.1 in detail. As expected, the thresholds needed to separate bit-0 and bit-1 increase as constant weight increases since more ISI occurs.

Table 1.1: Calculated Threshold Values

n	m	Threshold for Pw=100	Threshold for Pw=200	Threshold for Pw=400
20	4	14.54601	29.38945	59.55139
16	5	15.46568	31.27603	63.14824
15	6	16.18741	32.80973	66.01291
15	7	16.77731	34.07377	68.41711
15	8	17.34247	35.30765	70.84247
15	9	17.83612	36.44875	73.32761
16	10	18.1771	37.34136	75.18312

By using the values in Table 1.1, the codeword bit error rates (BER) of different constant weight codes have been calculated by simulations. Power is normalized here so lower weighted codes have higher power so higher molecules for each transmitted 1. It is evident that the low weight constant codewords suffer small error rates. However, it must be noted that the sampling time is not normalized in this experiment, therefore the less amount of information content is delivered in longer codewords.

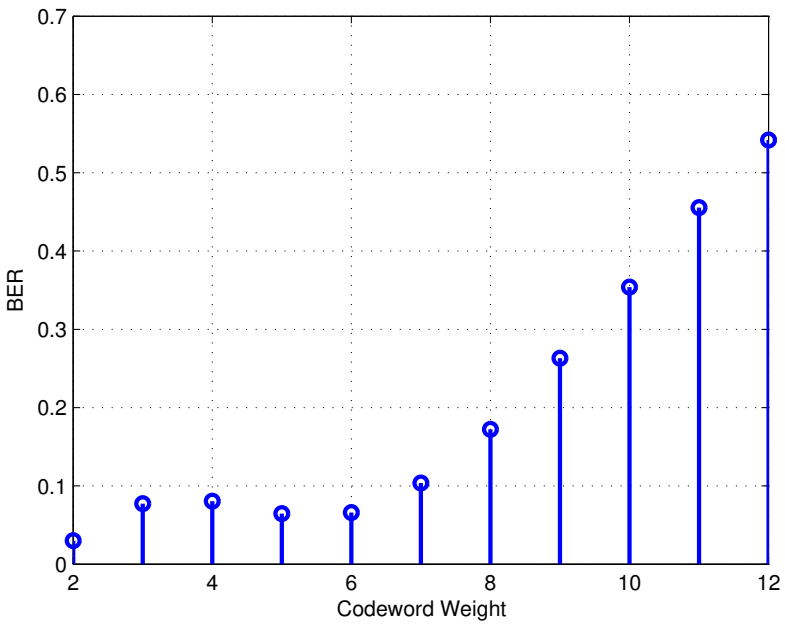


Figure 1.3: Length vs. codeword BER graph

1.4.2. Run Length Limited Codes

A run length limited (RLL) code is actually a line coding technique that guarantees run length of 0s or 1s being within limits. It has been widely used especially in storage systems and telecommunications. In VLC, flickers cause harmful effects on humans and in the quality of illumination as aforementioned. In literature, RLL codes are widely used for avoiding this problem.

RLL codes are defined with 4 parameters, namely m, n, d, and k. First two parameters give the rate of the code, $\frac{m}{n}$, and last two ones refer to the minimum number of zeros and maximum number of zeros between consecutive ones. One of the most basic examples of RLL is Manchester Coding. It ensures transition in each data symbol and both symbols occupy same time so it guarantees having no DC component.

In [13], IEEE 802.15.7 standard is reviewed by the authors. RLL coding is used as a flicker avoidance method in the standard. Both Manchester coding, 4B6B RLL, and 8B10B RLL, which are all defined by earlier standards, are used in the article. RLL is used together with Reed Solomon (RS) and convolutional codes (CC) for error correction. It is noted that this kind of low complexity and hard decision coding schemes are preferred in VLC instead of advanced ones like low-density parity check (LDPC) codes.

4B6B RLL code takes random 4-bits and encodes it to 6-bit codewords with weight 3. The properties of 4B6B RLL code are [13]:

- 50 percent duty cycle during one encoded symbol,
- Balanced DC,
- Capability of error detecting,
- Run length limited to four,
- Reasonable clock recovery.

Output of all investigated RLL codes in [13] have 50 percent duty cycle like the 4B6B RLL code. In the standard, OOK light dimming is achieved by inserting another stream in a frame which has different brightness level and able to satisfy aimed brightness after averaging. This process is called compensation. Not surprisingly, the data rate is lowered as the compensation block does not carry any information but only used for dimming [13]. Instead, by adjusting weight of codewords or using RLL codes with different duty cycle, desired brightness level could be guaranteed. An approach based on this idea is presented in Chapter 3.

1.5. Motivations

We had several motivations behind the research of channel coding methods for these channels. First of all, we found carrying out a research on the topic important, since most of the communication blocks in the literature assume existence of white Gaussian noise. However, this assumption is no longer valid in the aforementioned communication environments. For a researcher, it is tempting to design systems for challenging cases. Moreover, as these channels have recently been popular, the contributions could be valuable and attractive for the increasing audience of the subject.

Secondly, numerous possible applications and use cases have been identified for the same channels. Therefore, the designed techniques focusing practicality could be deployed to the real applications. To be practical, we aimed to cover channel specific requirements besides the common ground of having input dependent noise. To illustrate, codewords for VLC should satisfy the dimming and flicker reduction needs while performing well in terms of error rates. Similarly, the codewords for molecular channel should be sparse enough to decrease severe ISI due to the diffusion process. We take into account those channel specific needs in our proposed methods. Apart from these essential needs, we also tried to cover other nice to have features like requiring low computational power by using short block lengths and paying attention to complexity of the proposed methods.

Lastly, we tried to design the systems that are jointly optimized as much as possible. Contrary to widespread approach of developing separate blocks, we proposed compound blocks as this allows joint optimization and yields better results in our experiments. The techniques presented in Chapter 2 provide joint equalization and channel decoding. The VLCNet, presented in Chapter 3, replaces whole VLC communication link and achieves a jointly optimized system. With the advance of neural networks that can model highly complex functions, the approach of joint optimization is likely to gain more attention in communications.

In the proposed techniques, we tried to present frameworks to allow fulfilling the needs of the channels rather than optimizing a set of parameters for a given scenario. For example, we propose using low weight constant weight codes for molecular communications channel and present our designed methods for decoding these codes. However, we are not interested in calculating what the weight values should exactly be as this depends on the bit rates, the channel states, etc. Rather, our motivation was to design generic techniques that can be used with different parameters. We believe the value of developing such generic architectures are higher than parameter optimization for a given case.

1.6. Contribution of the thesis

The studies and the contributions of this thesis can be divided into two parts. The first part focuses on molecular communications and proposes two different methods for it, while the second part focuses on VLC and proposes an end-to-end communications solution.

Considering the challenges explained in Section 1.1, we have proposed two novel decoding techniques for constant weight codes in energy limited nano- and micro-scale devices. Our aim is to attain higher communication quality by lowering codeword bit error rates. Both of the methods assume that channel state information (CSI) is partially or fully available at the receiver. This information can easily be acquired via

transmitting the pilot sequence, a training sequence which is already known by the receiver. Our motive behind implementing ISI mitigation at the receiver part is that the transmitters are envisaged to be simpler than the receivers in many applications of molecular communications. One example of such a transmission from nano- to micro-scale devices is the communication from the nano-scale position beacons to the micro-scale drug delivery robots [21]. Our proposed methods utilize the information that the codewords have a predetermined weight at the decoding process. Therefore, the CLWC not only decreases ISI, the randomness of the transmission, and the energy usage, but also increases the communication performance if it is used together with our proposed decoders.

Our first proposed algorithm is the iterative sorting decoder. It is a heuristic approach, which requires less computation power compared to our second proposed method. Iterative sorting decoder uses its decision for better interference cancellation in the subsequent iterations. After each iteration, likelihood of the decoded sequence is calculated and the algorithm halts if the likelihood is not increased compared to the previous iteration. Therefore, performance increase at each iteration is guaranteed and error propagation is prevented. The algorithm iterates until the same sequence is obtained as iteration output, or the maximum number of iterations is reached. Our second proposed method embeds the information that the codewords have the same weight into the trellis graph together with ISI. By forming a super trellis, which takes both the channel information and coding information into account, the maximum a posteriori sequence can be determined by using one of the maximum a posteriori sequence estimation (MAPSE) methods in the literature. We employ the Viterbi algorithm for our path selection technique in the trellis. The super trellis method is computationally more expensive compared to the iterative sorting method, yet offers better performance at the expense of this cost. Both proposed methods bring substantial performance enhancement compared to the alternative molecular communication decoding schemes in the literature. To summarize the contributions of the first part of thesis work are as follows.

- We introduce heuristic iterative sorting decoder for constant weight codes, which is a computationally lightweight method designed for micro-scale devices.
- We propose super trellis methods, which provides maximum a posteriori sequence estimation.
- We evaluate our methods for different set of parameters and show their performance superiority over the existing methods in the literature.

In the second part, we have proposed a novel autoencoder designed to satisfy lighting requirements as a practical solution to VLC. To best of our knowledge, this is the first deep learning based system that both provides both flicker reduction and dimming capability, which are the two main constraints of VLC. We introduce a novel competitive activation unit, FRAU. This unit is located at the end of the learned transmitter transformation to map neuron activations to favorable values. FRAU divides the activations into blocks in which only a certain number of nodes could be active. While the values of inactive nodes are zeroed out, values of the active nodes are fed into a modified sigmoid function to guarantee flicker mitigation. The contributions of this part are as follows.

- We introduce VLCnet with a novel activation layer that guarantees flicker mitigation and show that it can be efficiently learnt.
- We explore the effect of different VLCnet parameters on the error rate performance.
- We show that training with a certain amount of noise yields the best error rate performance even under changing noise levels. This is an important result that can be used in another deep learning based communication system designs.
- By comparisons with other methods, we show that VLCnet provides superior FER performance while having moderate complexity.

1.7. Organization of the thesis

This chapter discusses the thesis topic and our research motivations. The products of research contributions of the thesis are presented in two separate chapters, in Chapter II and Chapter III. In Chapter II, we introduce two different novel decoding methods for constant weight codes in the molecular communications environment. In Chapter III, we propose a novel autoencoder that is able to address lighting requirements of VLC as well as communications requirements. Both Chapter II and III contain detailed literature reviews, system models and their own evaluation results. Finally Chapter IV concludes the thesis and discusses the future work.

2. METHODS FOR MOLECULAR COMMUNICATIONS CHANNEL

2.1. Related Literature

Molecular communication systems are inspired mainly by the nature and major methods in the literature utilize a diffusion based architecture [22], or alternatively use microtubules and molecular motors [23]. In both cases, electromagnetic wave based communications paradigms need to be reconsidered before directly applying to the molecular communications. Molecular communication via diffusion, also known as communication with Brownian motion, is the method considered in this study. In this scheme, messenger molecules are released to the fluidic medium and they move in the medium subject to the diffusion process. As they reach to the receiver, the information, which is embedded in the release pattern of the molecules, is decoded by the receiver. A problem of MCvD is that some of the released molecules arrive at the receiver very late and in unintended time slots. This delayed arrival causes severe intersymbol interference.

The nature of molecular communication is different than electromagnetic wireless communications on the grounds that the transmission is based on Brownian motion. The stochastic behavior of the transmission can be modeled as noise [24]. In addition, the variance of the received number of molecules at the receiver depends on the transmitted sequence and is proportional to the amount of the transmitted molecules [25]. This is different from wireless communications, where the noise variance is generally considered to be constant and independent of the transmitted data. Since the variance of the randomly arriving molecules depends on the number of transmitted molecules, the fewer molecules are released by the transmitter to the medium, the less this randomness is experienced by the receiver. Besides mitigating noise, releasing fewer molecules reduces energy consumption by decreasing the need for synthesizing molecules at the transmitter. This motivates a channel code designer to map the information bits to

codewords with more zeros compared to the ones in an on-off keying scheme. Such a coding technique also decreases ISI, hence it cures another problem of the molecular communication. In addition to the randomness specific to molecular communications, the imperfection in the operation of the nano-devices, such as miscounting of received molecules, can also be modeled as noise.

Various modulation techniques have been proposed in the literature to cope with ISI. Molecular concentration shift keying (MOCSK) is a modulation technique in which information bits are encoded as two different molecule concentrations [26]. For each even and odd time slots, molecule types A and B are transmitted respectively in order to reduce the interference due to the previous symbol. Contrary to molecular shift keying (MoSK) [27], the molecule types are not used for signalling but to reduce ISI in MOCSK. Molecular transition shift keying (MTSK) is another ISI-mitigating modulation technique proposed in [28]. In this method, zeros of the information symbols are encoded by the absence of transmitted molecules. For ones, either molecule type A or B is released depending on the following symbol. These methods require at least two molecule types, which increases the complexity of the nano-devices that are used as transmitters and receivers.

Apart from modulation techniques, receiver-based methods are also proposed for ISI mitigation in the molecular communications literature. Kilinc *et al.* [25] proposed implementing minimum mean squared error equalizer and decision feedback equalizer designed specifically for the MCvD. In [29], memory-1 cancellation, which calculates and removes the expected number of molecules from the previous symbol, is proposed. Some works utilize the change of received molecule count within a symbol interval. In [30], the receiver takes the difference of received molecules between two observation durations in each sampling interval. First observation is the point where the transmitter just emits the molecules, and the second point is where the most fraction of current symbol molecules arrive at the receiver. The receiver subtracts the molecule count at the first point from the count at the second point. In [31], weighted sum detectors are proposed for the diffusive molecular communications with flow. The receiver sums

the observations taken in a bit interval after multipliying them with their associated weights. The authors show that particular selection of the weights is equivalent to having a matched filter, where the matched filter performance is the same as the optimal detector when there is no ISI in the channel. To calculate the weights of the weight sum detector, acceleration aided particle swarm optimization is proposed in [32]. As optimum sequence detection method for ISI experiencing MCvD, Viterbi algorithm is suggested in [25], [33], [31]. In [30], reduced-state Viterbi algorithm is proposed for the ease of computation in the nano-scale devices.

Using forward error correction methods is also proposed in the literature as another approach for mitigating ISI. Implementation of Hamming codes in molecular channels are proposed in [34]. The authors show that even the simple channel coding method increases transmission performance while benefiting energy consumption. In another study [21], Hamming codes are compared with more complex codes, such as Euclidean low density parity check codes and cyclic Reed Muller codes, for the molecular communications channel. According to the study, Hamming codes can be a viable solution for nano-device to nano-device communications while LDPC codes can be preferred when one side is a macro sized device. Bit crossover errors are considered when only one molecule is sent as the information symbol in each sample time in [35]. In the same study, error coding techniques specialized for this type of errors are presented. However, only crossovers at consecutive time slots are covered in the study, and the data rates are expected to be very low if level-2 or more crossovers are not allowed for the diffusion channel. In [36] and [37], channel codes mitigating crossover errors in molecule shift keying modulated systems are studied and compared.

Bai *et al.* [19] propose implementing minimum energy channel codes (MEC), originally proposed in [38], in molecular communications. They claim that this scheme, which is based on minimizing the average code weight for a given minimum distance among codewords, is better than Hamming codes both in terms of energy consumption and bit error rate (BER). MEC is also implemented in terahertz band [17]. Another energy efficient low weight coding scheme is proposed in [18], where only weight zero and

weight one codewords are used. This strict weight constraint makes the information rate decrease drastically. The performance of MEC and MTE are compared in [20] together with other low weight codes. It is stated in the study that the performance depends heavily on the metrics used and there are no clear winners.

Energy and computation power scarcity of the nano-machines, severe ISI experiencing channel conditions, and the input dependent noise are unavoidably in the nature of molecular communications, and they need to be taken into account in the design of error correcting codes. To satisfy all of these needs, we propose using constant weight codes with low weights as channel codes. Low weight constant weight codes do not only lower ISI, but they also facilitate error control and correcting mechanisms at the receiver side. Another obvious advantage of constant low weight codes is lower energy consumption. In the literature, constant weight codes are used for their aforementioned advantages in similar channels. In [16], the authors propose using constant low weight channel codes in the terahertz band to prevent multiuser interference in nano-sensor networks. Jamali *et al.* propose constant weight codes for molecular communication [39]. They state that constant weight coding (CWC) allows channel state information free maximum likelihood (ML) detection if symbol intervals are chosen large enough so that ISI does not occur. However, this lowers the information rate substantially in the molecular communication channel where the transmitted molecules can arrive at the receiver very late as a result of their random movement. Therefore, we consider that ISI exists in our system model to target the plausible data rates.

To decrease ISI and input-dependent noise, we propose using constant and low weight codewords. In this work, we propose two different decoding techniques that utilizes the information that the codeword weights are constant. First approach is heuristic and requiring less resource in terms of computation. The second method is optimal as it provides maximum a posteriori sequence estimation but more complex in exchange. The simulation results show that both methods outperform the existing methods in the literature.

The rest of the chapter is organized as follows: Section 2.2 provides the review of the system model, the molecular communications channel and the constant weight coding. The decoding approaches are presented in Section 2.3, starting with basic decoding methods, threshold and sorting decoders. This section then continues with our proposed decoding techniques, iterative sorting, and super trellis decoders. Performance comparisons and evaluations are presented in Section 2.4. The complexities of the proposed methods are discussed in Section 2.5. Transmitter side approaches are mentioned in Section 2.6. Finally, Section 2.7 gives general discussion about the chapter.

2.2. System Model

2.2.1. Molecular Communications Channel

The considered system consists of a transmitter, a diffusion channel and a receiver. The transmitter is assumed to be a point source that emits molecules, and the receiver is a spherical receiver which is located r_0 away from the transmitter. The receiver is assumed to have perfect molecule reception at its surface and its radius is r_r. The environment surrounding the transmitter and the receiver is a three dimensional unbounded fluidic medium as an example of proper diffusive surrounding. The collisions among the messenger molecules are ignored, as widely assumed in literature [40]. The probability density function of finding molecule at distance r at time t for this environment is formulated as [41, Eq. (3.104)]

$$p(r,t|r_0) = \frac{1}{4\pi r r_0 \sqrt{4\pi D t}} \left(e^{-\frac{(r-r_0)^2}{4Dt}} - e^{-\frac{(r+r_0-2r_r)^2}{4Dt}} \right), \tag{2.1}$$

where D is the diffusion coefficient.

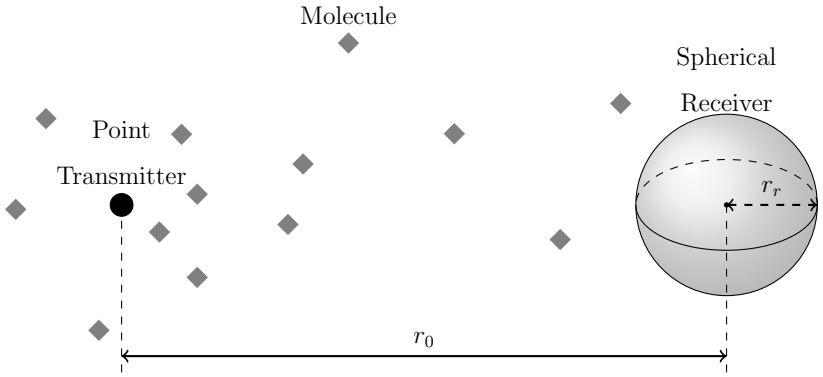


Figure 2.1: MCvD system model

The hitting rate for a molecule is calculated by taking partial derivative of (2.1) and then multiplying with the sphere surface area [41, Eq. (3.108)] as

$$\begin{aligned} f_{hit}(t) &= 4\pi r_r^2 D \partial_r p(r,t|r_0)|_{r=r_r} \\ &= \frac{r_r}{r_0}\frac{1}{\sqrt{4\pi D t}}\frac{r_0 - r_r}{t} e^{-\frac{(r_0-r_r)^2}{4Dt}}. \end{aligned} \tag{2.2}$$

The fraction of molecules that arrive at and are absorbed by the receiver until time t can thus be obtained as

$$F_{hit}(t) = \int_0^t f_{hit}(x)dx = \frac{r_r}{r_0}\,\mathrm{erfc}\left[\frac{r_0 - r_r}{\sqrt{4Dt}}\right]. \tag{2.3}$$

Since $F_{hit}(t)$ provides the fraction of the molecules arriving at the receiver until time t, time can be divided into intervals and the probability of a messenger molecule being absorbed by the receiver at a specific time slot [40] can be calculated as

$$p_k = F_{hit}(kt_s) - F_{hit}([k-1]t_s). \tag{2.4}$$

As the reference modulation technique, binary concentration shift keying (BCSK) is employed in this work. The number of messenger molecules arriving at the receiver

is considered as the signal amplitude in concentration shift keying (CSK) [27]. In the model, the input symbol duration t_s is chosen such that the hitting probabilities, so taps of our diffusion channel, are monotonically decreasing, i.e., $p_1 > p_2 > p_3 > ....$ This assumption about the channel is important for the decoding methods that are covered in the next section.

If N molecules are transmitted with each having p hitting probability for a specific time interval, the number of received molecules at that time interval follows a Binomial distribution [27]. Moreover, it is known that $Binomial(n, p)$ distribution can be aproximated with Gaussian distribution $\mathcal{N}(np, np(1-p))$ if np is large enough and p is not close to zero or one [27]. Hence, for a large number of emitted molecules, the number of received molecules during time slot i can be approximated as [40]

$$y[i] \mid \boldsymbol{b}^i \sim \mathcal{N}(\mu[i],\, \sigma^2[i]), \tag{2.5}$$

where $y[i]$ is the number of received molecules at time slot i and $\boldsymbol{b}^i$ is the binary message symbol history, $\boldsymbol{b}^i = \{b[1], b[2], \ldots, b[i]\}$. The parameters of the distribution can be written as

$$E\Big[y[i] \mid \boldsymbol{b}^i\Big] = \mu[i] = N \sum_{k=1}^{i} p_k\, b[i-k+1], \tag{2.6}$$

$$Var\Big[y[i] \mid \boldsymbol{b}^i\Big] = \sigma^2[i] = N \sum_{k=1}^{i} p_k\,(1-p_k)\, b[i-k+1] + \sigma_c^2, \tag{2.7}$$

where N is the number of released molecules at each $b_i = 1$, and σ_c^2 is the counting noise variance that is independent of the input sequence. Counting process errors are modeled by the noise with constant variance.

2.2.2. Constant Weight Coding for Molecular Communications

Constant weight codes have been widely studied in the literature and they are used in various applications [42, 43].

Let $U = \{0,1\}^k$ be the set of all binary sequences of length k and each element of U be a valid sequence of binary input symbols. Then, by constant weight coding, these k-long input sequence are transformed to n-long codewords that have the same weight, i.e., $\boldsymbol{u} \xrightarrow{CWC(k,n,m)} \boldsymbol{v}$, where $\boldsymbol{u} \in U$ and $\sum_{i=1}^{n} v_i = m$. If the codebook contains all n-long words that have m weights, then this codebook is denoted as a full codebook.

Most of the research has been done for finding bounds and code constructions for constant weight codes with particular minimum distances [44, 45]. Having higher minimum distance increases the correctability of the codewords, yet it also increases the codeword length and complexity of the mapper/demapper between input words and codewords. For such a mapper, a dissection based geometric approach is proposed in [46]. In [45], CWC constructions for Hamming distances 6, 8, 10 are proposed by generalizing the class of CWC suggested by Bose and Rao [47]. Considering the limitations of the nano-scale devices, mapping messages to codewords should be simple. Hence, enumerative approach, as explained in [46], is applied in the system model for the mapper and demapper. The codewords are ordered lexicographically and stored in a look-up table. The mapper computes the table index from the input message by calculating the decimal value of the message and maps the message to a codeword in the table with the calculated index. At the receiver side, our proposed decoders estimate the codeword from the noisy received series of molecules. Afterwards, the lexicographic index from the codeword is calculated, as in [48], and demapping is performed by converting the index to binary to have the received message.

The minimum possible Hamming distance among the constant weight codewords is two. In the system model, CWC with minimum distance of two is implemented to have the simple aforementioned mapper/demapper pair. Another advantage of

not having a Hamming distance requirement is that ISI is not worsened much by over-lengthening the codewords. The bandwidth expansion due to channel coding is minimum thanks to not having a minimum distance requirement among the codewords.

It should be noted that the probabilities of having 0-bit and 1-bit in symbols are equal at these input sequences but not in the codewords. By controlling the m value, CWC allows diminishing the number of ones in the coded stream at the expense of using longer sequences. The total number of possible binary k-bit input messages is 2^k. For $n > k$, we can have $\frac{n!}{(n-m)!m!}$ different words with all of them having weight m. For instance, for $k = 12$ bits, $n = 16$ bits are required to have at least 2^{12} words with weight 5. Then, the probability of ones is $\frac{5}{16}$, which is smaller than the uncoded weight ratio of 0.5. By controlling the m parameter, probability of 1-bit can be controlled and reduced to diminish input dependent noise and energy expenditure. In the system model, low-weight CWC is employed as the channel coding method as a consequence of these benefits.

2.3. Decoding Approaches

We consider four different decoding approaches. The first two, namely the threshold and sorting based methods, have already been proposed for constant weight decoding in the literature. The other two methods, iterative sorting decoder and super trellis based decoder, are the contributions of this work.

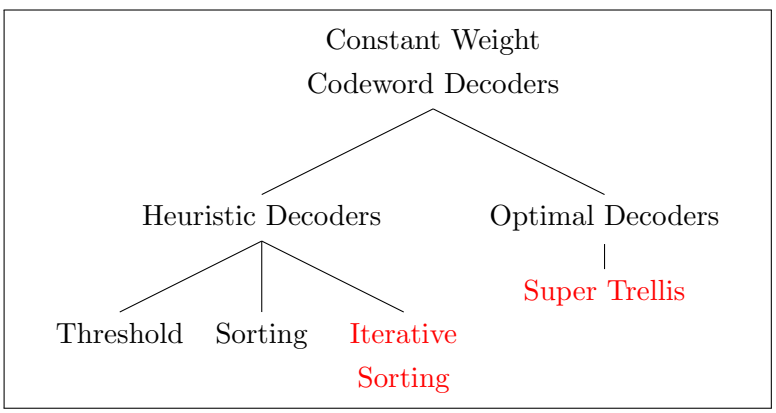


Figure 2.2: Constant weight decoding methods

Our proposed methods address the detector block of the overall system as shown in Figure 2.3.

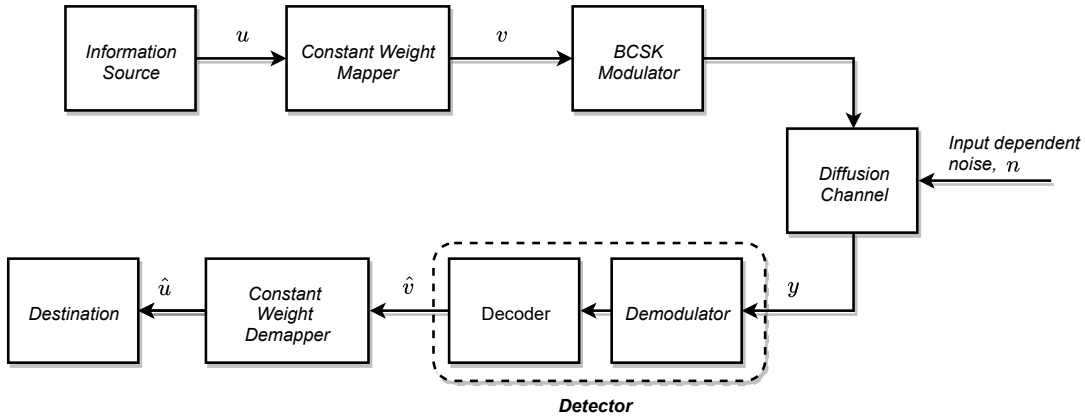


Figure 2.3: System model block diagram

2.3.1. Threshold Decoder

From a computational complexity point of view, the simplest decoding method is to compare the number of received molecules with a predetermined threshold. Because of this convenience, threshold decoders are the most conventional methods for BCSK in the literature [19, 27, 28]. Despite the fact that detection based on a given threshold is easy, determining the threshold is not a trivial task at all. This is because the parameters of the distributions of the received molecules depend on the previously transmitted bits as well as the current one, as shown in (2.6), (2.7). Threshold decoding for bit v_i is given as

$$y[i] \underset{\hat{v}_i=0}{\overset{\hat{v}_i=1}{\gtrless}} \gamma(\boldsymbol{v}^{i-1}), \tag{2.8}$$

where $\gamma(\boldsymbol{v}^{i-1})$ is the optimal threshold for the bit sequence history $\boldsymbol{v}^{i-1}$. (2.8) means the current symbol estimation $\hat{v}_i$ is set to one if the received molecule count exceeds the threshold; otherwise, it is set to zero. For optimal detection of bit v_i given the previous bits $\boldsymbol{v}^{i-1}$, the maximum a posteriori (MAP) probability decision rule can be

written as

$$\begin{aligned} &P[v_i = 1 \mid \boldsymbol{v}^{i-1}]\; p(y[i] \mid \boldsymbol{v}^{i-1}, v_i = 1) \underset{\hat{v}_i=0}{\overset{\hat{v}_i=1}{\gtrless}} \\ &P[v_i = 0 \mid \boldsymbol{v}^{i-1}]\; p(y[i] \mid \boldsymbol{v}^{i-1}, v_i = 0), \end{aligned} \tag{2.9}$$

where $P[.]$ denotes the probability of having bit 1 or 0 at the current time slot i given the bit history, and $p(.)$ is the Gaussian distribution of the received molecule count, $y[i]$, that has mean and variance given in (2.6), (2.7). The reader should note that the prior probability of having 1-bit or 0-bit at the current time slot is not equiprobable. Since all codewords have constant weight, the prior probability of the current bit depends on previous bits in the same codeword.

The optimal threshold for v_i, given $\boldsymbol{v}^{i-1}$, is calculated by equating the left- and the right-hand sides of (2.9) for $y[i] = \gamma(\boldsymbol{v}^{i-1})$. Then, (2.9) can be rewritten as

$$\frac{p(\gamma(\boldsymbol{v}^{i-1})|\boldsymbol{v}^{i-1}, v_i = 1)}{p(\gamma(\boldsymbol{v}^{i-1})|\boldsymbol{v}^{i-1}, v_i = 0)} = \frac{P[v_i = 0|\boldsymbol{v}^{i-1}]}{P[v_i = 1|\boldsymbol{v}^{i-1}]}. \tag{2.10}$$

The optimal threshold for a given bit history v_i can be calculated using (2.10). As the actual transmitted bit sequence cannot be known by the receiver, all possible v_i's must be considered to calculate the optimal threshold for time slot i. This problem is intractable to solve analytically, so numerical methods are used to find the optimal threshold for time slot i in [40].

To have a single threshold value for all bit history v_i, the probability of error can be written in terms of the threshold γ, and then minimized with respect to it as

$$\begin{aligned} P_e(\gamma) = &\sum_{\forall \boldsymbol{v}^{i-1}} P(v_i = 0, \boldsymbol{v}^{i-1}) Q\Big(\frac{\gamma - \mu_{0,\boldsymbol{v}^{i-1}}}{\sigma_{0,\boldsymbol{v}^{i-1}}}\Big) \\ &+ P(v_i = 1, \boldsymbol{v}^{i-1}) Q\Big(\frac{\mu_{1,\boldsymbol{v}^{i-1}} - \gamma}{\sigma_{1,\boldsymbol{v}^{i-1}}}\Big), \end{aligned} \tag{2.11}$$

where $\mu_{\theta,\boldsymbol{v}^{i-1}}$ denotes $\mu[i \mid v_i = \theta, \boldsymbol{v}^{i-1}]$ while $\sigma_{\theta,\boldsymbol{v}^{i-1}}$ is $\sigma[i \mid v_i = \theta, \boldsymbol{v}^{i-1}]$. Minimizing this function with respect to γ yields

$$\begin{aligned}\frac{\partial P_e(\gamma)}{\partial \gamma} &= \sum_{\forall \boldsymbol{v}^{i-1}} P(v_i = 1, \boldsymbol{v}^{i-1}) \mathcal{N}(\gamma^*; \mu_{1,\boldsymbol{v}^{i-1}}, \sigma^2_{1,\boldsymbol{v}^{i-1}}) \\ &\quad - P(v_i = 0, \boldsymbol{v}^{i-1}) \mathcal{N}(\gamma^*; \mu_{0,\boldsymbol{v}^{i-1}}, \sigma^2_{0,\boldsymbol{v}^{i-1}}) \\ &\quad = 0\end{aligned} \tag{2.12}$$

As explained in [40], the expression above is again hard to solve analytically considering all different $\boldsymbol{v}^{i-1}$. Alternatively, an empirical method can be followed to estimate the threshold by transmitting training sequences, which would yield

$$\bar{\gamma} = E\{\gamma(\boldsymbol{v}^{i-1})\} = \sum_{\forall \boldsymbol{v}^{i-1}} \gamma(\boldsymbol{v}^{i-1}) P(\boldsymbol{v}^{i-1}). \tag{2.13}$$

In the provided simulation results presented in Section 2.4, the averaged threshold is used considering its practicality for the devices operating in MCvD.

2.3.2. Sorting Decoder

As aforementioned, the input symbol duration t_s in the system is chosen such that $p_1 > p_2 > ... > p_L$. The symbol duration can be increased so much that the ISI is not experienced in the channel, i.e., $p_1 > 0$ and $p_2 = p_3 = ... = p_L = 0$. The authors in [39] consider such an ISI-free molecular communications channel and propose decoding constant weight codes by sorting the number of received molecules. This decoder labels the topmost m time slots as 1-bit with respect to the received molecule counts. Let the indices of the sorted time slots be $j_1, j_2, j_3, .., j_n$ such that $y[j_1] > y[j_2] > y[j_3] > .. > y[j_n]$. Then, the decoded sequence by the sorting decoder

is given as

$$\begin{aligned} \hat{v}[j_1] = \hat{v}[j_2] = ...\hat{v}[j_{m-1}] = \hat{v}[j_m] = 1, \\ \hat{v}[j_{m+1}] = \hat{v}[j_{m+2}] = ...\hat{v}[j_{n-1}] = \hat{v}[j_n] = 0. \end{aligned} \tag{2.14}$$

Although the sorting decoder achieves maximum likelihood sequence detection in ISI-free channels, it is no longer optimal in channels with symbol durations considered in our system, since all of these channels have ISI. However, it is a suitable option for CWC decoding in MCvD as it is simple to implement at nano-scale, and it does not require CSI. Moreover, the sorting decoder does not require sending long sequences of training codewords unlike the threshold decoder.

2.3.3. Iterative Sorting Decoder

As explained in the previous part, sorting is an optimal solution at extremely low communication rates. However, for realistic rates, the effects of the ISI is greatly amplified in MCvD channels. This substantially degrades the decoding performance of the sorting method. As it has been shown by (2.6) and (2.7), previous codeword symbols affect the mean and the variance of the received molecule count for current symbols. Sorting method determines a set of cardinality m as bit-1 indices in the constant weight codeword. This selection can be used to compute ISI and then remove it from the received signal. So, the derived problem gets closer to the case where there is no ISI and the sorting decoder is optimal. Therefore, it is expected that sorting the signal after subtracting the estimated ISI effects results in better estimation. Surely, this depends on the accuracy of the ISI estimation, thus the accuracy of the initial sorting. Accuracy of each estimation can be calculated using the MAP distance metric for molecular communication, which is given as

$$\hat{\boldsymbol{v}} = \underset{\boldsymbol{v}}{\operatorname{argmax}}\, p(\boldsymbol{y}|\boldsymbol{v})p(\boldsymbol{v}). \tag{2.15}$$

Here, $\boldsymbol{v}$ denotes a valid codeword from the codebook. The constant codewords that are in a full codebook are equally likely, so we can remove $p(\boldsymbol{v})$ to obtain

$$\begin{aligned}
\hat{v} &= \underset{v}{\operatorname{argmax}}\, p(\boldsymbol{y} \mid \boldsymbol{v}) \\
&\overset{(a)}{=} \underset{v}{\operatorname{argmax}} \prod_{i=1}^{n} p(y[i] \mid \boldsymbol{y}^{i-1}, \boldsymbol{v}) \\
&\overset{(b)}{=} \underset{v}{\operatorname{argmax}} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2[i]}} \exp\left(-\frac{(y[i]-\mu[i])^2}{2\sigma^2[i]}\right) \\
&\overset{(c)}{=} \underset{v}{\operatorname{argmax}} \sum_{i=1}^{n} \frac{-(y[i]-\mu[i])^2}{2\sigma^2[i]} - \frac{1}{2}\ln\left(2\pi\sigma^2[i]\right) \\
&= \underset{v}{\operatorname{argmin}} \sum_{i=1}^{n} \frac{(y[i]-\mu[i])^2}{\sigma^2[i]} + 2\ln\left(\sigma[i]\right).
\end{aligned} \tag{2.16}$$

In equality (a) of (2.16), the chain rule is used. Although $y[i]$ depends on the past, $y[i]$ becomes independent of $\boldsymbol{y}^{i-1}$ when $\boldsymbol{v}$ is known, as explained in [49]. This reduces the expression to the equality (b). In equality (c) of (2.16), the monotonicity of the natural logarithm is used and the constants that are not dependent on the hypothesis sequence $\boldsymbol{v}$ are removed. It should be noted that the mean and the variances in (2.16) are conditioned on $\boldsymbol{v}$ and computed by (2.6) and (2.7). Consequently, MAP distance or improbability between any received sequence $\boldsymbol{y}$ and a codeword $\boldsymbol{v}_A$ is given as

$$d(\boldsymbol{y}, \boldsymbol{v_A}) = \sum_{i=1}^{n} \frac{(y[i]-\mu_A[i])^2}{\sigma_A^2[i]} + 2\ln\left(\sigma_A[i]\right). \tag{2.17}$$

Here it must be noted that $\boldsymbol{v}_A$ is not any codeword but a constant weight codeword as we already dropped the prior term in (2.15).

As long as the estimation quality is improved through this operation, it can be repeated until two successive iterations produce the same estimation for the sequence or the maximum number of iterations is reached. Thus, the stopping conditions are either

(i) having increased distance in consecutive iterations, (ii) having same estimation in consecutive iterations, or (iii) reaching maximum iteration number. The algorithm initializes with equating the array that will be sorted to the received molecules counts. The whole iterative sorting decoder algorithm is provided in Figure 2.4.

Input: $\boldsymbol{y}$, m, CSI, $maxIter$
Output: $\hat{v}$
1: $Initialisation : \hat{y} \leftarrow y$
2: **for** $i = 1 : maxIter$ **do**
3: $\hat{v}' \leftarrow SortingCWDecoder(\hat{y}, m)$
4: $d' \leftarrow Distance(r, \hat{v}')$
5: **if** $i == 1$ **or** $(\hat{v}' \neq \hat{v}$ **and** $d' \leq d)$ **then**
6: $ISI \leftarrow CalculateISI(\hat{v}', CSI)$
7: $\hat{y} \leftarrow y - ISI$
8: $d \leftarrow d'$
9: $\hat{v} \leftarrow \hat{v}'$
10: **else**
11: **break**
12: **end if**
13: **end for**
14: **return** $\hat{v}$

Figure 2.4: Algorithm for Iterative Sorting Decoder

Instead of the stopping conditions defined on line 5 in the algorithm in Figure 2.4, a fixed number of iterations may also be applied. However, this fixed iteration number would be dependent on the channel. An example with a severe ISI condition with $t_s = 0.1s$, is given in Figure 2.6. In this example, the frame error rate starts to converge after 5 iterations. However, for less challenging channels, even two iterations are generally sufficient. Stopping conditions in the algorithm in Figure 2.4 avoid unnecessary iterations that do not improve performance.

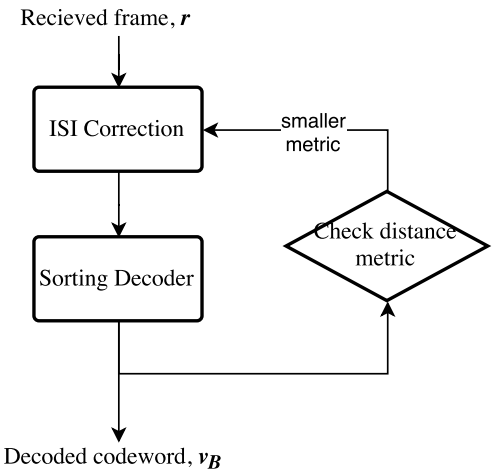


Figure 2.5: Iterative Sorting Flowchart

Figure 2.5 illustrates the iterative sorting flowchart. Sorting and ISI correction phases repeat as distance metric decreases until the same sequence is obtained in consecutive iterations.

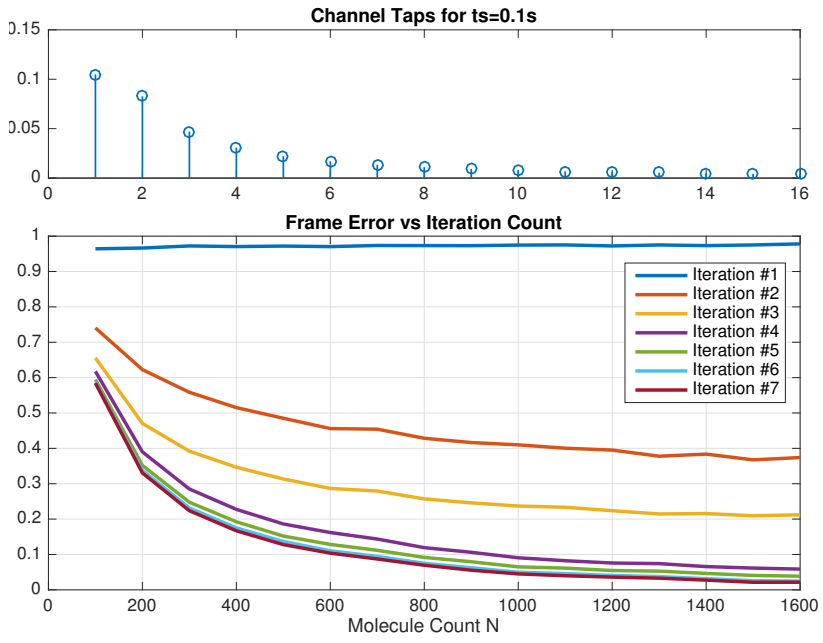


Figure 2.6: Frame error rate plots for different fixed number of ISI removal and sorting stages

Since all estimated sequences of regular sorting have weight m, any bit-0 transmission and bit-1 detection ($0 \rightarrow 1$ error) means that there is a corresponding $1 \rightarrow 0$ error in the estimated codeword as well. The probability of $0 \rightarrow 1$ type error at time

slot i by the initial sorting decoding is denoted by $p(v[i] = 0,\ \hat{v}_s[i] = 1)$, where $\hat{\boldsymbol{v}}_{\boldsymbol{s}}$ is the sorting decoder estimation of iteration s.

Lemma 2.1. *Provided that $0 \rightarrow 1$ error occurs at time slot i for $i > 1$ and at any sorting iteration s, the probability that codeword symbol at $(i-1)$ is also decoded as 1 by the sorting decoder of the same iteration is greater than or equal to 0.5 for all transmitted codeword sequences until index $(i-1)$.*

Proof. The proof starts with defining a lower bound for the probability in Lemma 2.1.

$$\begin{aligned} & p(\hat{v}_s[i-1] = 1 \mid v[i] = 0,\ \hat{v}_s[i] = 1,\ \boldsymbol{v}_1^{i-1}) \\ & \overset{(a)}{\geq} p(y[i-1] > y[i] \mid v[i] = 0,\ \hat{v}_s[i] = 1,\ \boldsymbol{v}_1^{i-1}) \\ & = p(y[i-1] - y[i] > 0 \mid v[i] = 0,\ \hat{v}_s[i] = 1,\ \boldsymbol{v}_1^{i-1}) \\ & = p(z[i] > 0 \mid \xi),\ \forall i > 1,\ \forall \boldsymbol{v}_1^{i-1} \end{aligned} \tag{2.18}$$

Here, the transmitted codeword bit sequence from index 1 to $(i-1)$ is denoted by $\boldsymbol{v}_1^{i-1}$ and the joint condition set is denoted by ξ. Inequality in (a) is a direct consequence of the sorting decoder. Since the noise samples are independent for a given codeword sequence and Gaussian distributed, their difference, denoted by $z[i]$, is also a Gaussian random variable. The mean and the variance of this distribution can be written in terms of (2.6) and (2.7) as

$$\mu_{z[i \mid \xi]} = \mu[i-1] - \mu[i], \tag{2.19}$$

$$\sigma^2_{z[i \mid \xi]} = \sigma[i-1]^2 + \sigma[i]^2. \tag{2.20}$$

Since $v[i] = 0$, there is no contribution of the p_1 term in $\mu[i]$ calculation. Thus, the difference of consecutive means can be computed as

$$\begin{aligned} \mu[i-1] &= p_1 v_{i-1} + p_2 v_{i-2} + ... + p_{i-1} v_1 \\ \mu[i] &= p_2 v_{i-1} + p_3 v_{i-2} + ... + p_i v_1 \\ \mu[i-1] - \mu[i] &= (p_1 - p_2) v_{i-1} + (p_2 - p_3) v_{i-2} + ... \\ &\quad ... + (p_{i-1} - p_i) v_1. \end{aligned} \tag{2.21}$$

All of the coefficients of $\mu[i-1] - \mu[i]$ are positive due to the decreasing channel taps, which is a result of the chosen sampling interval. Considering $\mu_{z[i \mid \xi]} \geq 0$ and $p(z[i] \mid \xi)$ is a Gaussian, $p(z[i] > 0 \mid \xi) > 0.5$ for any $\boldsymbol{v}_1^{i-1}$ except all-zero vector. For the all-zero $\boldsymbol{v}_1^{i-1}$, this probability is equal to 0.5. Thus, $p(\hat{v}_s[i-1] = 1 \mid \xi) \geq 0.5$ for all $\boldsymbol{v}_1^{i-1}$ as well. This concludes the proof.

Theorem 2.2. *The received number of molecules of the codeword bits that have* $0 \rightarrow 1$ *type sorting errors are decreased,* $\hat{y}_{s+1}[i] < \hat{y}_s[i]$*, by the next iteration's ISI removal phase of the iterative sorting with a probability greater or equal to 0.5 for all codeword sequences until index* $(i-1)$*.*

Proof. The statement in the Theorem 2.2 is written mathematically as follows:

$$p(\hat{y}_{s+1}[i] < \hat{y}_s[i] \mid \; v[i] = 0, \; \hat{v}_s[i] = 1, \boldsymbol{v}_1^{i-1}) \geq 0.5, \quad \forall i > 1, \forall \boldsymbol{v}_1^{i-1} \tag{2.22}$$

Equation (2.22) can easily be proven by using Lemma 2.1. Lemma 2.1 states in case of $0 \rightarrow 1$ error at time slot i, the probability of detection of preceding bit as 1 is greater or equal to 0.5 for all bit sequences until $(i-1)$. Then, with the same probability, the received molecule count at index i is lowered in the next iteration due to the removal of estimated ISI caused by decoded 1-bit at $(i-1)$.

It has been shown that the molecule count that yields $0 \rightarrow 1$ error at sorting iteration s is decreased at least by $p_1 * N$ at the ISI removal phase of the next iteration with a probability greater or equal to 0.5. After this removal of molecules, the sorting at next iteration has higher probability of avoiding $0 \rightarrow 1$ error at time slot i.

Corollary 2.3. *By combining Lemma 1 and Theorem 1, a more accurate lower bound is given as*

$$\begin{aligned} &p(\hat{y}_{s+1}[i] < \hat{y}_s[i] \mid \xi) \geq \\ &\frac{1}{2}\operatorname{erfc}\left(\frac{\mu[i-1]-\mu[i]}{\sqrt{2(\sigma^2[i-1]+\sigma^2[i])}}\right) \geq 0.5, \end{aligned} \tag{2.23}$$

where ξ is the same joint condition set in (2.18).

Proof. It was shown that $p(\hat{y}_{s+1}[i] < \hat{y}_s[i] \mid \xi) \geq p(z > 0 \mid \xi)$ and $p(z > 0|\xi) = \int_0^\infty \frac{1}{\sqrt{2\pi(\sigma[i-1]^2+\sigma[i]^2)}} \exp\left(\frac{-(z-\mu[i-1]+\mu[i])^2}{2(\sigma[i-1]^2+\sigma[i]^2)}\right)$. By equating t to $\frac{z-\mu[i-1]+\mu[i]}{\sqrt{2(\sigma[i-1]^2+\sigma[i]^2)}}$ in the complementary error function definition, $\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$, the above statement is proven.

An example of sorting decoder outputs at each iteration is provided in Figure 2.7. The colored bits at the first subplot are the actually transmitted ones and the received molecule numbers are shown in the second subplot. The first iteration of the sorting decoder detects colored bits of second subplot as 1-bit and others as 0-bit. Then, at each iteration, ISI is calculated using the decision of previous iteration and is removed from the received sequence of molecules. Decisions after iteration 4 and 5 are the same, which concludes the algorithm and outputs the latest decisions.

2.3.4. Super Trellis Decoder

Computationally inexpensive suboptimal and heuristic methods seem feasible for nano-machines, considering computation capabilities of these devices. Micrometer sized machines, which have higher computation abilities and power, can have better error

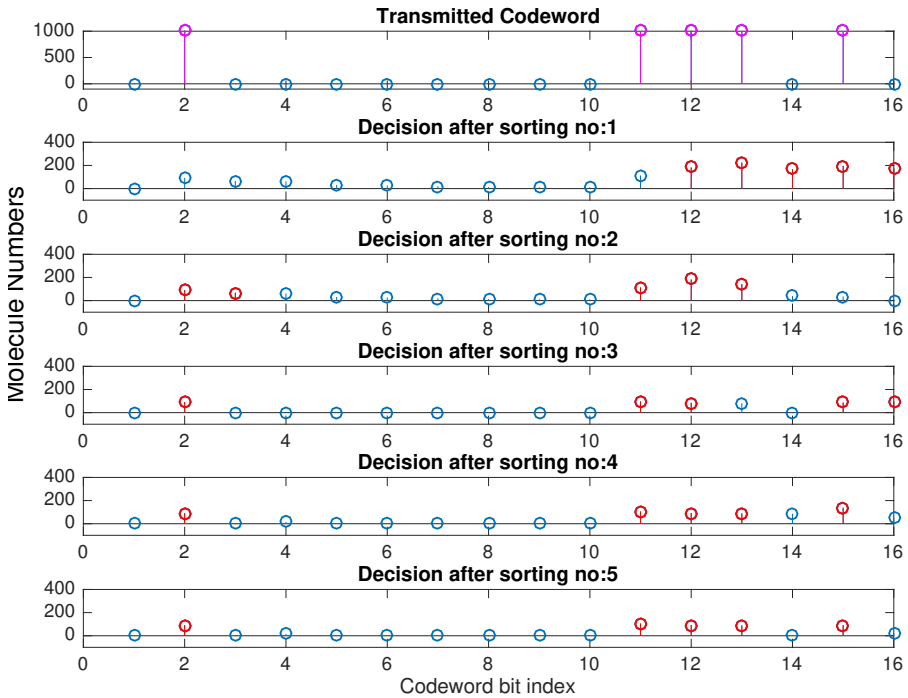


Figure 2.7: An example of iteration sorting process where $t_s = 0.1s$

rate values by deploying optimal decoders. MAP state sequence estimation is a well known optimal decoding method that would be useful in such a setting.

Consider the state sequence $\boldsymbol{x}$ to have a one-to-one correspondence with codeword $\boldsymbol{v}$. In this case, codeword that maximizes $p(\boldsymbol{y}|\boldsymbol{v})$ can be found by maximizing $p(\boldsymbol{r}|\boldsymbol{x})$. Thanks to this one-to-one correspondence between codewords and state sequences, the MAP estimation problem in (2.15) can be reformulated as

$$\hat{\boldsymbol{v}} \Leftrightarrow \hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, p(\boldsymbol{y}, \boldsymbol{x}). \tag{2.24}$$

Furthermore, if the observations at the channel output depend only on the current state transition and the state sequence is a Markov process, the joint probability of the

states and the received samples can be simplified to

$$p(\boldsymbol{y}, \boldsymbol{x}) = \prod_{i=1}^{k} p(y[i] \,\Big|\, x[i+1], x[i]) \prod_{i=1}^{k} p(x[i+1] \mid x[i]). \tag{2.25}$$

Note that the product on the left in (2.25) is the likelihood and the one on the right is the prior probability. Arguments that maximize (2.25) minimize its negative natural logarithm. This term can be written as the sum of the MAP branch metric of the state transitions [50] as

$$-\ln p(\boldsymbol{y}, \boldsymbol{x}) = \sum \delta(x[i], x[i+1]), \tag{2.26}$$

$$\delta(x[i], x[i\ +\ 1]) \quad = \quad -\ln p(y[i] \ \mid \ x[i\ +\ 1], x[i]) \ - \ \ln p(x[i\ +\ 1] \ \mid \ x[i]), \tag{2.27}$$

where $\delta(x[i], x[i+1])$ is the MAP branch metric.

In the conventional Viterbi equalization case where ISI channel has length L and transmitted bits are equally likely, x holds previous $L-1$ codeword symbols, $\boldsymbol{v}_{i-L+1}^{i-1}$, that affect the current received symbol. In the conventional case, this is enough to form a Markov model since $\prod_{\forall i} p(y[i] \mid \boldsymbol{v}) = \prod_{\forall i} p(y[i] \mid x[i+1], x[i])$ and $\prod_{\forall i} p(v[i] \mid \boldsymbol{v}) = \prod_{\forall i} p(x[i+1]|x[i])$. The latter equation holds because the symbols are equally likely in the ordinary case, so are the state transitions. However, to be able to construct a Markov model in our system model, the states should also contain the necessary information to calculate the likelihood of state transitions. This is because symbol probabilities of constant codewords change according to weight realization until the time instance i inside the codeword. Once the states incorporate $L-1$ codeword symbols and weight realization information, current and next state become sufficient to calculate both the likelihood and prior probability. This enables formulating $p(\boldsymbol{y}, \boldsymbol{v})$ as in (2.25). The aim of the work done is to construct a super trellis diagram of joint states that incorporate

both ISI and the constant weight constraints so that Viterbi algorithm is applicable on this trellis. The proposed technique provides joint equalization and channel decoding.

The main constraint in constant weight codes is the unvarying number of ones in each codeword. The total number of ones seen up to a particular time slot i in a codeword should always be less than or equal to the codeword weight. Similarly, the total number of ones seen up to a particular time slot i is less than or equal to the total number of ones seen up to the next time slot $i+1$, i.e.,

$$w_s[i] = \sum_{j=1}^{i} v[j] \leq \sum_{j=1}^{i+1} v[j] \leq m, \ \forall i < n. \tag{2.28}$$

If a trellis diagram is drawn only for this weight constraint, the total number of ones until time slot i together with i define the state of that trellis. The number of states at each i in this trellis is one more than the constant codeword weight, since zero is another possible total weight until time slot i, thus a valid state. We call the coordinates of weight trellis, $(i, w_s[i])$, as weight states, $q[i]$. Having defined the weight states, the possible weight state transitions from a particular weight state can be determined by whether the maximum allowed number is reached or not. If the codeword weight is already reached, the state is (i, m), and the only possible state transition is to state $(i+1, m)$. The weight trellis diagram for codeword length $n = 7$ and codeword weight $m = 3$ is shown in Figure 2.8. A possible state sequence in such a trellis is illustrated as a red path in Figure 2.8. The frame is decoded as $[0, 1, 0, 1, 1, 0, 0]$ if this is path is chosen by the proposed detector.

The trellis explained above forces all decoded codewords to have the same weight. In a full codebook, all of such constant codewords are included with equal probability, therefore the product on the right in (2.25) can be removed from the MAP calculation of all possible codewords. If the Viterbi branch elimination takes place inside the trellis instead of evaluating all possible constant codewords at the end, the prior probabilities need to be taken into account.

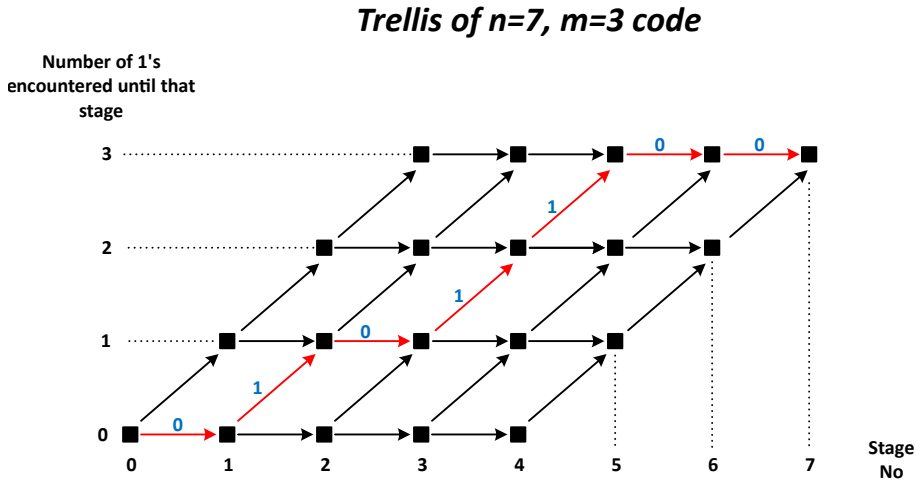


Figure 2.8: An example of decoding on the weight trellis

Mathematically, the joint state $x[i]$ can be written as $(\boldsymbol{v}_{i-L+1}^{i-1}, q[i])$. Then the branch metric in (2.27) can be reformulated as

$$\begin{aligned}\delta(x[i], x[i+1]) = \frac{\big(y[i] - \mu[i \mid \boldsymbol{x}]\big)^2}{\sigma^2[i \mid \boldsymbol{x}]} + 2\ln\left(\sigma[i \mid \boldsymbol{x}]\right) \\ - ln(P(x[i+1] \mid x[i])),\end{aligned} \tag{2.29}$$

where $\mu[i\,|\,\boldsymbol{x}] \coloneqq \mu\big[y[i]\,|\,x[i], x[i+1]\big]$ and $\sigma[i\,|\,\boldsymbol{x}] \coloneqq \sigma\big[y[i]\,|\,x[i], x[i+1]\big]$. $P(x[i+1]\,|\,x[i]$ is the codeword symbol probability at i and it can easily be calculated thanks to weight state $(i,\, w_s[i])$. The probability of having 1-bit at $i+1$ is $\frac{m-w_s[i]}{n-i}$.

A portion of the complete trellis diagram is illustrated in Figure 2.9. The joint states are composed of weight states and ISI states. We illustrate the ISI states inside the weight states to form joint states. Path elimination of the Viterbi algorithm takes place when two paths combine in the same joint state. Branch metrics in (2.29) are accumulated to find likelihood of the paths. Some examples of such intersections are shown in Figure 2.9 with red circles.

In practice, the proposed decoding can be implemented by defining a joint state such as $state \coloneqq \{weight\_state, ISI\_state\}$. A path is composed of joint state transitions from the beginning. Paths array keeps record of all possible paths until time slot

i. Likewise, the sum of those state transitions' branch metrics are recorded in another array named $Costs$. At each transition from $i \rightarrow i+1$, each path in the $Paths$ array is updated by adding a new state or states, if the last weight state allows. Then, the last states of the paths of updated $Paths$ array are checked. If there are paths that have exactly the same state in their last node, the one that has higher cumulative cost in $Costs$ array is eliminated. For the $n = 7, m = 3$ constant weight code and $L = 3$, the $Paths$ array for $i = 0, 1, 2$ and 3 are shown as

$$
\begin{aligned}
Paths[0] &= \Big[\{0,00\}\Big], \\
Paths[1] &= \begin{bmatrix} \{0,00\} & \rightarrow & \{0,00\} \\ \{0,00\} & \rightarrow & \{1,10\} \end{bmatrix}, \\
Paths[2] &= \begin{bmatrix} \{0,00\} & \rightarrow & \{0,00\} & \rightarrow & \{0,00\} \\ \{0,00\} & \rightarrow & \{0,00\} & \rightarrow & \{1,10\} \\ \{0,00\} & \rightarrow & \{1,10\} & \rightarrow & \{1,01\} \\ \{0,00\} & \rightarrow & \{1,10\} & \rightarrow & \{2,11\} \end{bmatrix}. \\
Paths[3] &= \begin{bmatrix} \{0,00\} & \rightarrow & \{0,00\} & \rightarrow & \{0,00\} & \rightarrow & \{0,00\} \\ \{0,00\} & \rightarrow & \{0,00\} & \rightarrow & \{0,00\} & \rightarrow & \{1,10\} \\ \{0,00\} & \rightarrow & \{0,00\} & \rightarrow & \{1,10\} & \rightarrow & \{1,01\} \\ \{0,00\} & \rightarrow & \{0,00\} & \rightarrow & \{1,10\} & \rightarrow & \{2,11\} \\ \{0,00\} & \rightarrow & \{1,10\} & \rightarrow & \{1,01\} & \rightarrow & \{1,00\} \\ \{0,00\} & \rightarrow & \{1,10\} & \rightarrow & \{1,01\} & \rightarrow & \{2,10\} \\ \{0,00\} & \rightarrow & \{1,10\} & \rightarrow & \{2,11\} & \rightarrow & \{2,01\} \\ \{0,00\} & \rightarrow & \{1,10\} & \rightarrow & \{2,11\} & \rightarrow & \{3,11\} \end{bmatrix}.
\end{aligned}
\tag{2.30}
$$

Please note that the possible paths increased exponentially until $i = 3$; however, the path collisions take place and eliminations are performed based on *costs* starting from the $i = 4$. This facilitates the computational cost of the algorithm. The whole Viterbi

based sequence estimation algorithm on the super trellis is given in Appendix A.

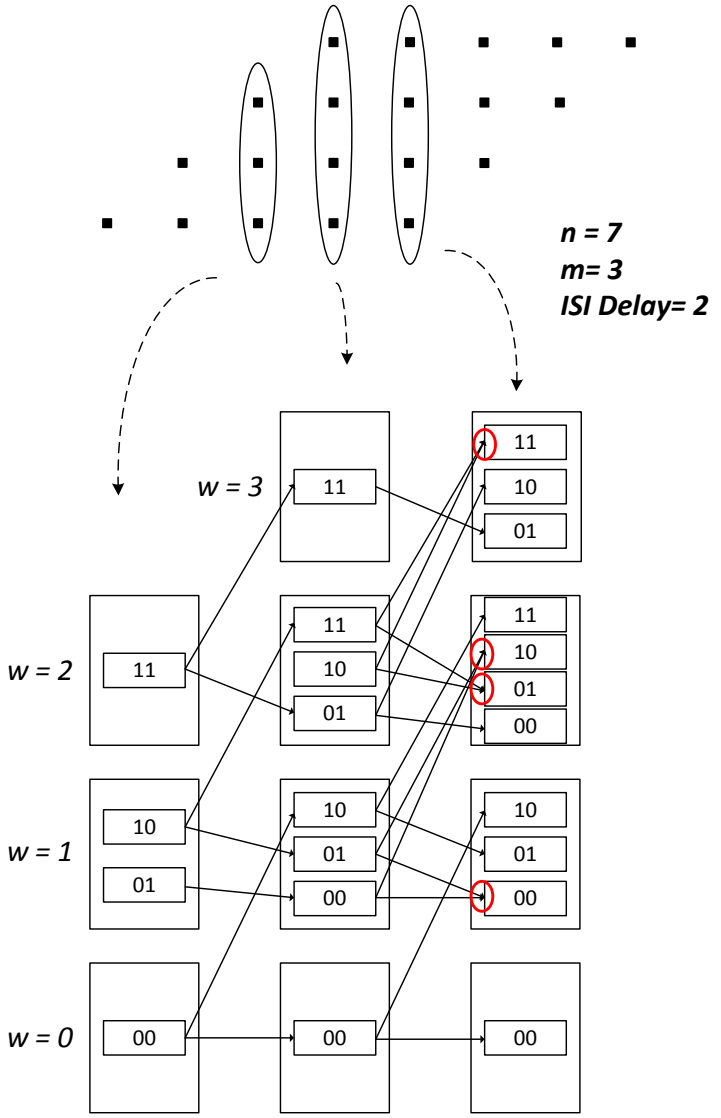
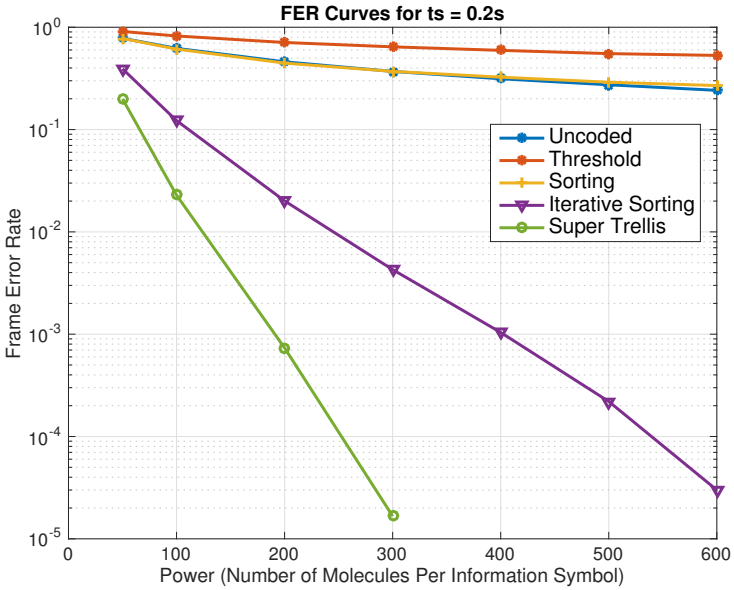
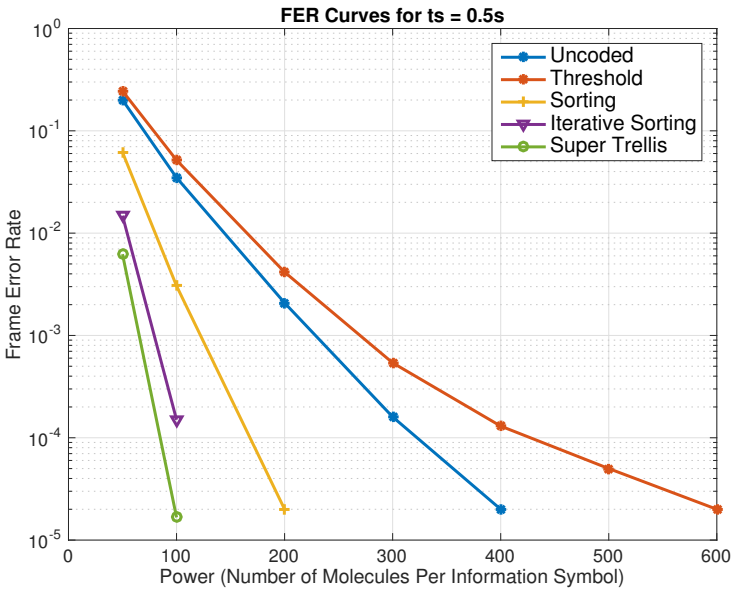


Figure 2.9: A part of the complete trellis diagram for $n = 7$, $m = 3$ and $L = 3$

Both iterative sorting and super trellis decoders can be considered as a combination of forward error correction and receiver side ISI mitigation techniques, since the proposed receivers utilize the information that the codewords have constant weight. Super trellis decoder performs joint ISI cancellation and constant codeword decoding by finding the constant codeword that maximizes the probability in (2.15). Since super trellis decoder method provides MAP sequence estimation, it is the optimal solution.



(a) FER for $t_s = 0.2s$



(b) FER for $t_s = 0.5s$

Figure 2.10: Frame error rate curves for different decoding methods

2.4. Performance Evaluation

2.4.1. Performance of CWC Decoders

The effectiveness of the proposed decoding methods is evaluated via computer simulation. First set of evaluations are carried out to exhibit the frame error rate performance of alternative CWC decoding methods. The environment is modeled as an unbounded 3-dimensional space. As aforementioned, collisions among the messenger molecules are ignored and the parameters for the channel are chosen as $D = 79.4\mu m^2/s$, $r_r = 5\mu m$, $r_0 = 10\mu m$. The Monte Carlo simulations are carried out for two different input symbol duration values, $t_s = 0.2s$ and $t_s = 0.5s$. The frame error rates (FER) are obtained for different numbers of transmitted molecules at 1-bit for each method. The results are averaged by transmitting 10^5 frames. Frames are generated by mapping $k = 12$ long input words to $n = 16$ long codewords by using CWC ($k = 12$, $n = 16$, $m = 5$) for all decoding methods. The sampling rate and the transmitted molecule counts for coded transmissions are normalized with a factor of $\frac{k}{n}$. The channel taps for two different symbol durations after the normalization are shown in Figure 2.11.

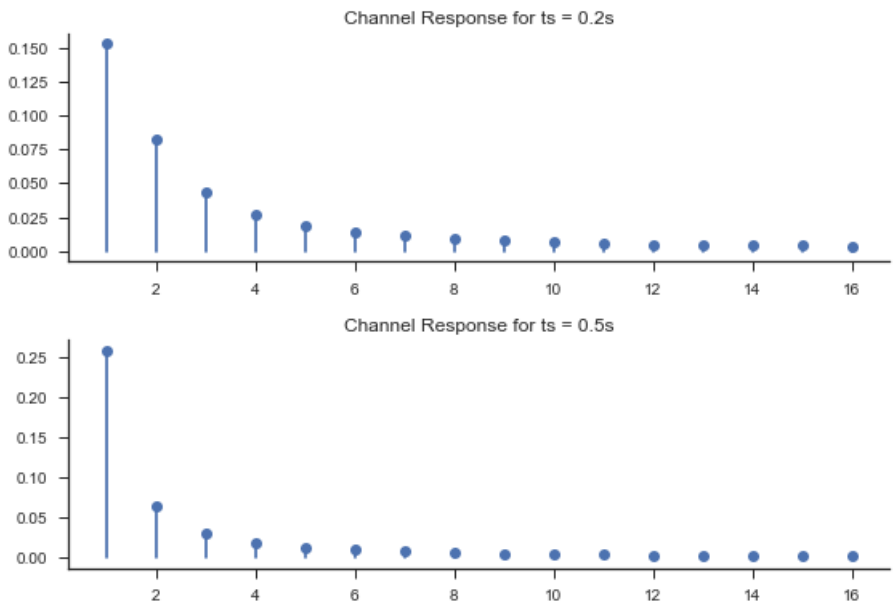
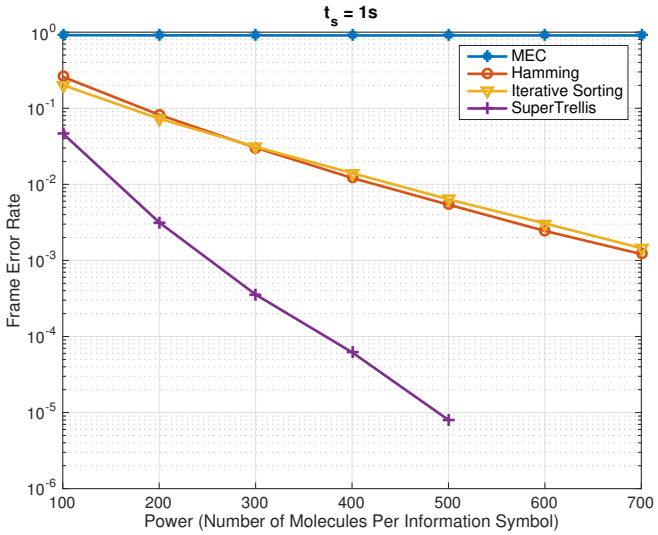
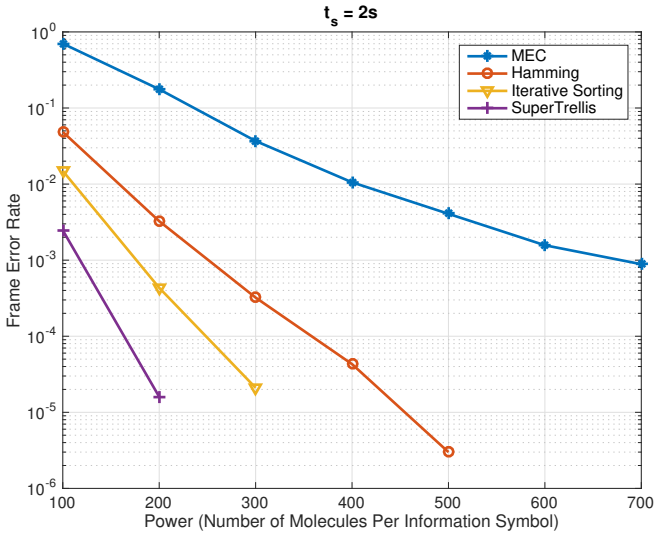


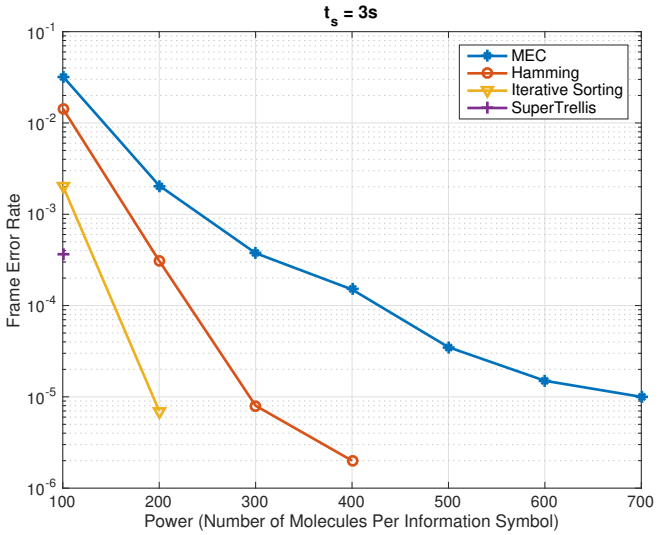
Figure 2.11: Channel taps for uncoded $ts = 0.2s$ and $ts = 0.5s$



(a) FER for $t_s = 1s$



(b) FER for $t_s = 2s$



(c) FER for $t_s = 3s$

Figure 2.12: Frame error rate curves for different channel coding methods

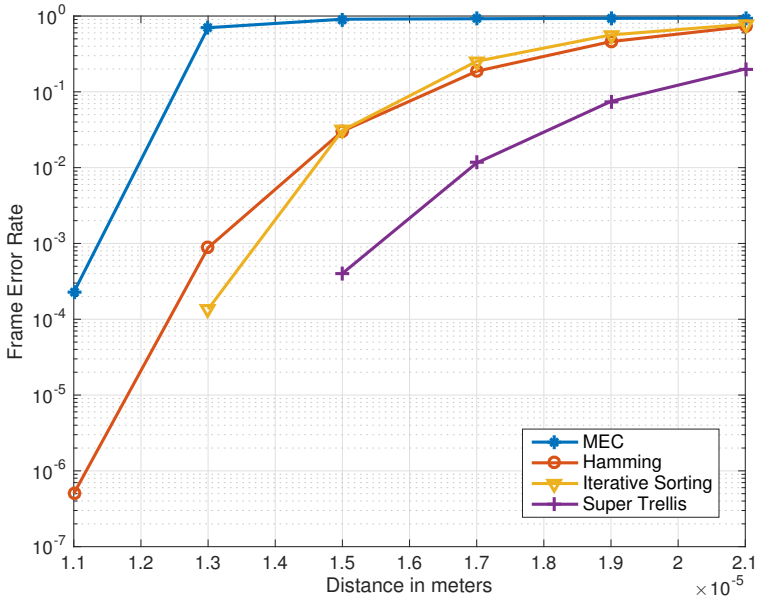


Figure 2.13: Frame error rate for different distances, r_0

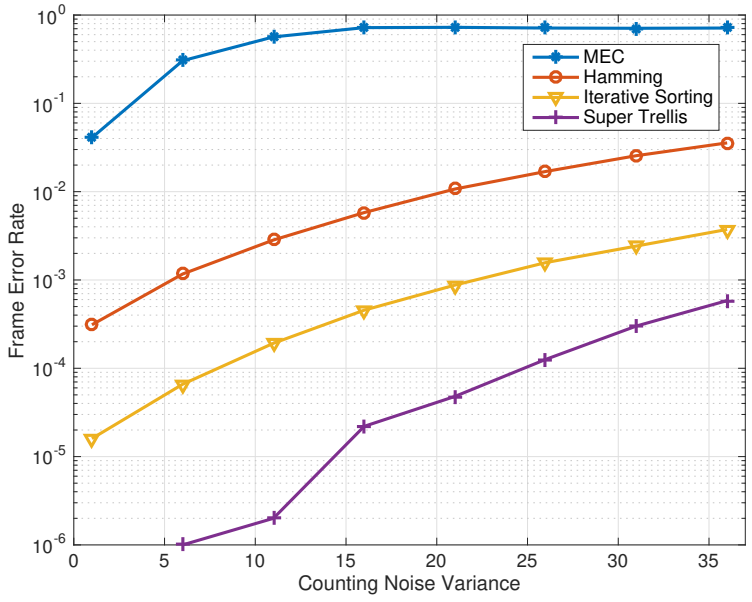


Figure 2.14: Frame error rate for different counting noise variances

The performance superiority of the proposed decoding schemes over the conventional decoders is shown in Figure 2.10a and Figure 2.10b. For $t_s = 0.2s$, ISI is severe and coding degrades the channel further because of the sample duration normalization. For this channel, CLWC does not perform better than the uncoded case if conventional decoders are used. This is a consequence of the noise reduction achieved by employing CWC not being able to compensate for the adverse effect of normalization on the hitting probabilities. Another remark is that increasing the power does not change FER considerably for both uncoded and CLWC with conventional decoder cases. Conversely, utilizing suboptimal iterative sorting decoding at the receiver side decreases FER substantially. The effect of ISI is lowered, and the error reduces notably with increased power. As expected, using optimal super trellis decoder at the receiver decreases FER further compared to the iterative sorting method. It approximately achieves the same FER values of iterative sorting requiring only half the power.

For $t_s = 0.5s$, the channel experiences less ISI compared to the previous channel, so CLWC with sorting method produces lower FER compared to the uncoded transmission. Threshold decoding detection is still worse, since ending up with a valid codeword is not guaranteed after thresholding unlike the sorting decoding if the used codebook is full. By applying iterative sorting, fewer frame errors are encountered compared to only sorting once. It is again verified in this channel that super trellis decoder yields the best FER results.

2.4.2. Performance of Forward Error Correction Methods

The error rate performance of joint usage of CLWC and proposed decoding methods are compared with Hamming and MEC codes, which are suggested to be used together with CSK modulation in literature [19]. Environment parameters are same as described in 2.4.1, except $r_0 = 15\mu m$. CWC ($k = 4$, $n = 7$, $m = 2$), Hamming (7, 4) and MEC($M = 16$, $d = 3$) are simulated in the described environment by transmitting 10^6 frames for $t_s = 1s$, $t_s = 2s$, and $t_s = 3s$. For Hamming and MEC, zero forcing equalizer is added to the receiver due to the presence of ISI. The results of

these channel coding methods are confirmed to be much poorer without equalization. As seen in Figure 2.12a, the performances of CWC with iterative sorting decoding and Hamming coding are similar at $t_s = 1s$ while CWC with super trellis results in superior performance. Having longer symbol intervals, in other words experiencing less ISI, benefits more to CWC with iterative sorting decoding compared to Hamming coding as shown in Fig 2.12b and 2.12c. For $t_s = 3s$ and number of molecules per information symbol greater or equal to 200, no frame error is observed in the simulation when the CWC decoding is performed by super trellis method. In all simulated sampling time configurations, CWC with super trellis decoding yields the best error performance whereas MEC results in the poorest error figures. All evaluated channel codes map input messages of length 4 to codewords of length 7 for Hamming and CWC, length 31 for MEC as described in 2.4.1. Due to the fact that codewords of MEC are lengthy, MEC experiences more ISI compared to CWC and Hamming after sample duration normalization. This explains the inferior performance values of MEC.

2.4.3. Effect of Distance on Performance

Another simulation is performed to evaluate the effect of distance between transmitter and receiver on the performance of alternative channel coding techniques. The environment parameters and the evaluated alternative channel coding methods are the same as explained in Section 2.4.2. The number of molecules per information symbol is set to 300 and $t_s = 1s$ in the analysis. For all simulated distances, CWC with super trellis decoding achieves the best error performances whereas MEC gives the poorest results which is in agreement with the results in Section 2.4.2. For r_0 shorter than $15\mu m$, CWC with iterative sorting achieves better performance than Hamming coding while Hamming coding give similar but slightly better performance for longer distances (see Figure 2.13).

2.4.4. Effect of Counting Noise on Performance

Apart from the input dependent noise, counting noise exists in the system model to represent the receiver molecule counter imperfection, (see (2.7)). Under different counting noise variances, forward error correction methods in Section 2.4.2 are compared. The environment variables are the same as in Section 2.4.2 except $t_s = 2s$ and the information sample molecule count is set to 300. As expected, increasing counting noise adversely affects the error performances of all methods. Proposed CWC with iterative sorting and super trellis achieve superior error performance in all counting noise levels (see Figure 2.14).

The proposed methods outperform the conventional threshold and sorting decoders for CWC. CWC with proposed decoding techniques also outperform Hamming and MEC as alternative channel coding methods in terms of error rates. The cost of this improvement is CSI and the computational complexity of the methods at the receiver.

2.5. Complexity Discussion

As well as performance, computational complexity is another important criterion for the deployment of the proposed methods. Iterative sorting method requires selecting m largest elements in an n-long array of received molecule counts at each iteration. This could be done by sorting the whole array and selecting the maximum m values. In this case, the time complexity is $\mathcal{O}(n \log n)$. Alternatively, a more efficient method, selection in worst case linear time [51], can be utilized to find m^{th} greatest element, and then all elements can be compared to find remaining $(m-1)$ greatest elements. Time complexity of this selection method, which is also known as BFPRT algorithm, is $\mathcal{O}(n)$ and comparing all other elements with a single value is another $\mathcal{O}(n)$. Therefore, total time complexity of the proposed iterative method is $\mathcal{O}(|s|n)$, where $|s|$ is the number of iterations. Space complexity of BFPRT selection is $\Theta(\log n)$ [52]. However, iterative selection method also stores previous iteration's estimated sequence. Therefore, total

space complexity of iterative decoding method becomes $\Theta(n + \log n)$, hence $\Theta(n)$. Second proposed method, super trellis, uses Viterbi algorithm, which is known to have time complexity proportional to total edge count [50, 53] and space complexity proportional to state count [50]. In the proposed super trellis algorithm, there are two edges incoming and outgoing from each joint state, x; one for 0-bit and the other for 1-bit. Therefore, the time complexity is $\mathcal{O}(2|x|n)$ so $\mathcal{O}(|x|n)$, and the space complexity is $\mathcal{O}(|x|n)$ where $|x|$ is the total number of joint states. The total number of joint states is determined by the constant weight of the codewords and the number channel taps. Consequently, $|x|$ is equal to m times $2^{(L-1)}$ in the worst case, where L is the number of channel taps. Although this computational complexity is not feasible for nanoscale devices, this method can be implemented in the nodes that have higher computational capability in a multi-scale communications scenario. Moreover, optimal super trellis constitutes a bound for the performance of other receivers, which is useful in judging how they perform.

2.6. Transmitter Side Approaches

Both of the proposed iterative sorting decoder and super trellis decoder methods are receiver side methods as shown in Figure 2.3. Alternatively, transmitter side approaches can be performed to increase error rate performances.

2.6.1. Molecule Count Adjustment at the Transmitter

The significant amount of molecules that are sent by the transmitter reach to the receiver in the following time slots of the allotted time. When the consecutive ones are sent, those late arriving molecules add up and may lead to false one for a zero following them. To solve this, number of molecules that represent an input one can be decreased if it is preceded by other ones. Aim is to result same number of molecules in the receiver for each one sent by the transmitter. Thus, sorting receiver would not be deceived by ISI. Rather than using the full channel information, a simplistic model of the channel can be assumed to be possessed by the transmitter.

For modeling the channel first linear regression has been investigated. However the goodness of fit is not satisfying. Alternatively, exponential function fitting has been carried on and root mean square error is seen to be decreased to significantly compared to the linear fit. Figure 2.15 show the exponential modeling and goodness of fit values.

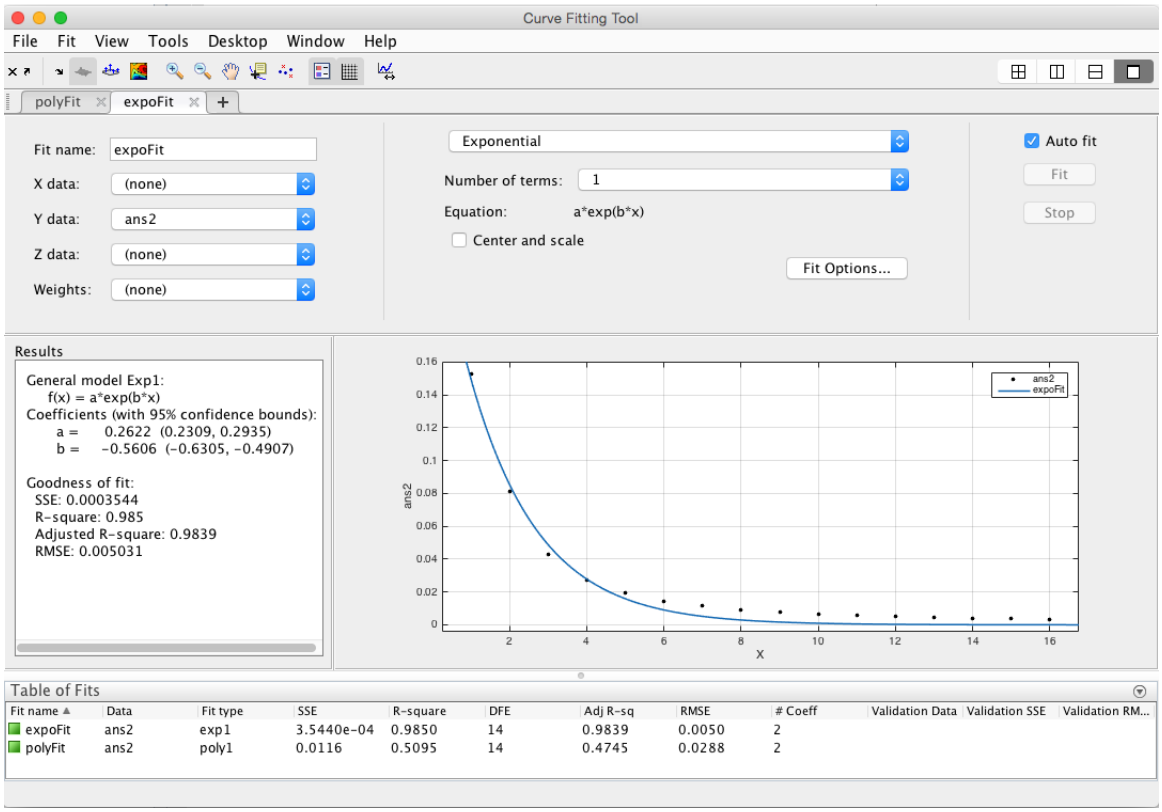


Figure 2.15: Exponential curve fit for $t_s = 0.2s$ channel

If the channel is modeled as $h[n] = \sum_n^l \alpha e^b$, where $b < 0$ for decreasing channel taps, the interference of the value one to the following symbol becomes $A = e^b$. Then if input one is wanted to be sent in the next interval, this one needs to represented by $(1 - A)$ number of molecules. Thus the interference is transformed from destructive to constructive for the communication.

For example, let 3 consecutive ones are sent and each one is represented by N molecules. Without power adjustment 3N molecules are needed and they will not received with the similar number of molecules at the receiver due to the effect of ISI. However with exponential channel modeled power adjustment, the needed number of

molecules for transmission becomes

$$N + N(1 - A) + N - (NA^2 + N(1 - A) * A) = 3N - 2A. \tag{2.31}$$

Both destructive effect of ISI is converted to constructive interference and power efficiency is achieved. The remaining power, 2A, can be used to normalize molecule levels that represent ones so the codewords become more robust to the noise.

For a given b parameter of the channel, the values of rows of the constant weight channel coding construction matrix are updated once and these codewords are used for mapping then.

The proposed method is simulated in the same noise and ISI constrained channels with sorting decoder. Figure 2.16 shows that BER decreases significantly when this power adjustment is applied compared to the case where same number of molecules are sent for each ones.

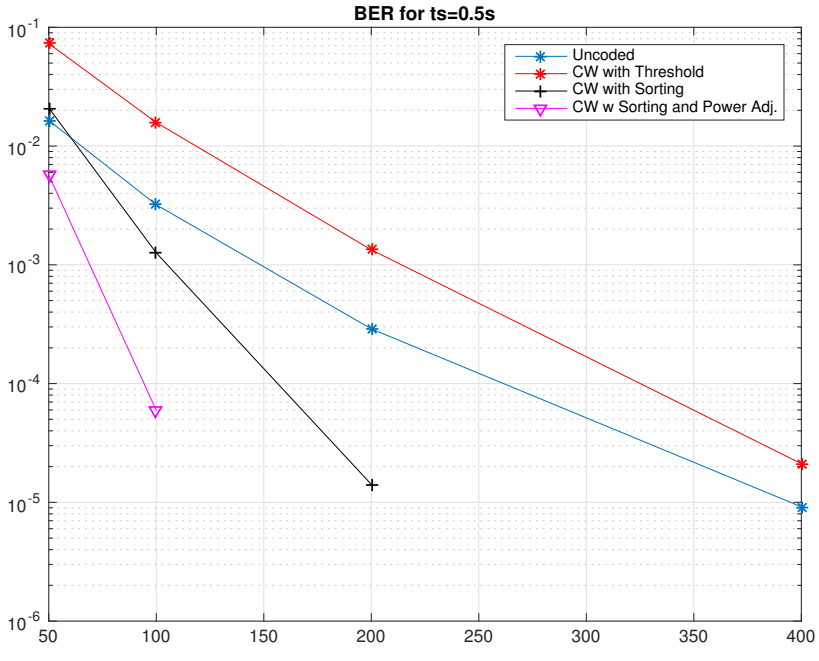


Figure 2.16: BER graph of CWC molecule count adjustment for $t_s = 0.5s$.

2.6.2. Doubly Constant Weight Coding

Doubly constant weight codes are defined as binary constant weight codes with length $n_1 + n_2$, where the first partition of n_1 coordinates has constant weight m_1 and the second partition of n_2 coordinates has constant weight m_2. Let us denote such codes with minimum Hamming distance d as DCWC(n_1, m_1, n_2, m_2, d). If minimum distance is 2, all vectors satisfying the explained weight requirements are valid codewords and such a code is denoted as DCWC(n_1, m_1, n_2, m_2). Doubly constant weight codes are useful in finding the bounds of the regular constant weight codes [54]. Since the weight of all codewords of doubly constant weight codes are $m_1 + m_2$, all doubly constant weight codes are indeed a constant weight code with an extra property. We experimented whether this additional property of the code structure can be utilized for better decoding assuming that the receiver possesses the information of this property.

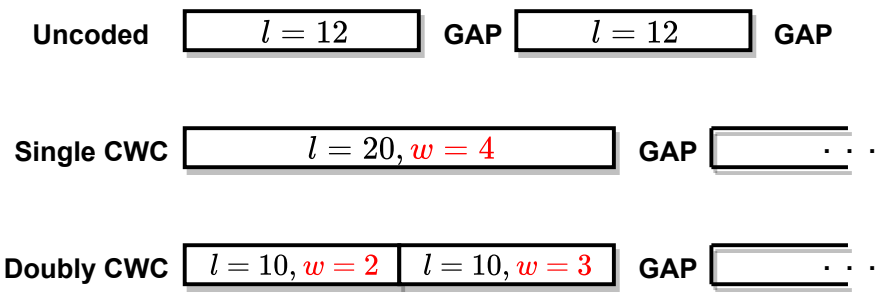


Figure 2.17: Compared CWC and DCWC configurations.

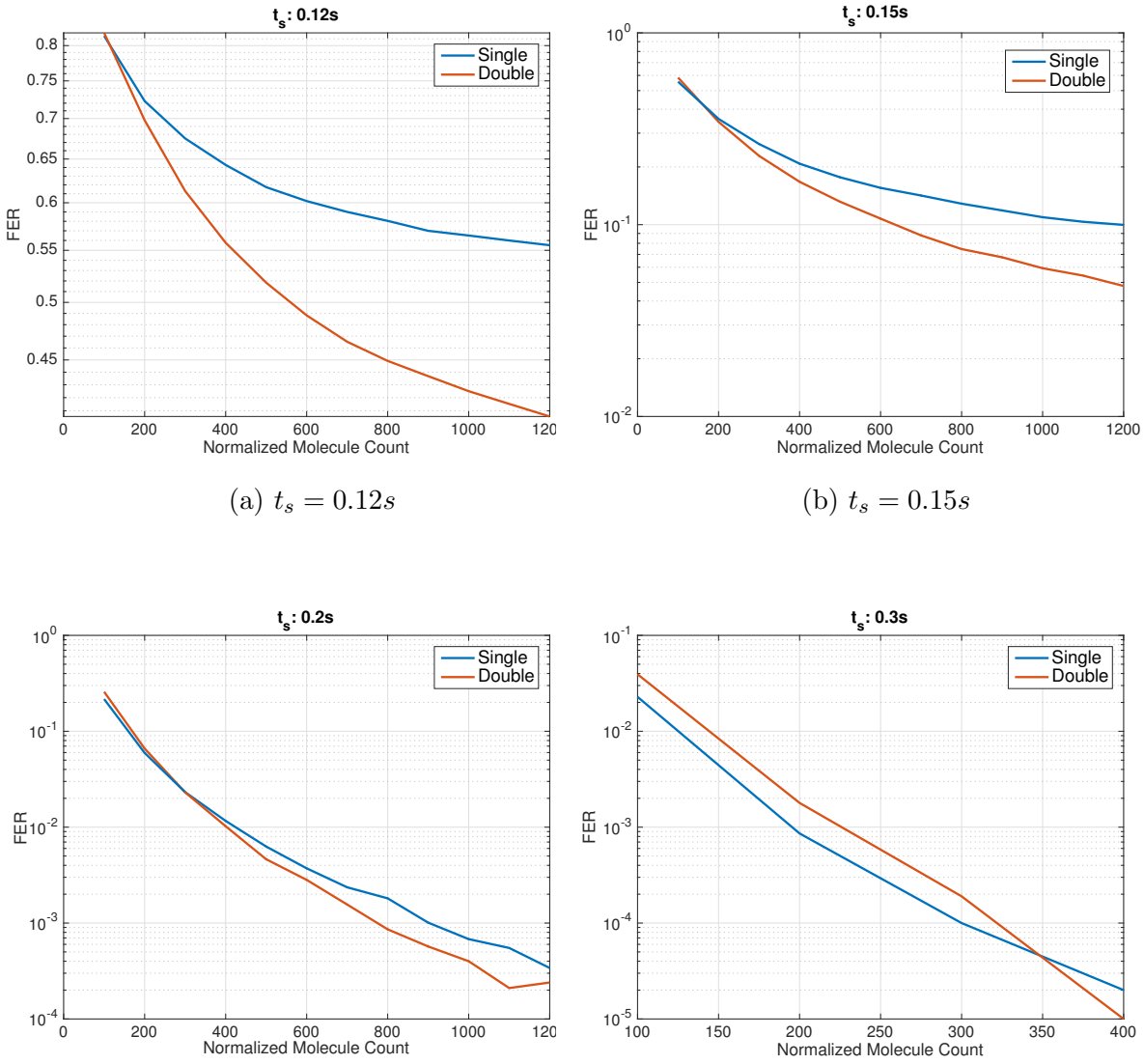
As an example, consider the case where the length of uncoded information blocks are 12. Since $2^{12} \leq \binom{20}{4}$, these input sequences can be coded by codewords with the length 20 and the constant weights of 4. Alternatively, constant weight codewords can be splitted into 2 parts, where the first partition has constant weight 2 and the second partition has constant weight 3, as $2^{12} \leq \binom{20}{4} \leq \binom{10}{2} * \binom{10}{3}$. Both partitions have less number of ones compared to the whole codeword in the classical CWC case. The likelihood of the large number of consecutive ones is diminished by this partitioning. However, since some of the codewords of the classical CWC cammot be used, the total weight increases in order to span same number of codewords as the classical CWC. As a result of this, the total number of weights increases from 4 to 5. That

increase of weight affects ISI adversely but provides additional information for the receiver. Furthermore, since the model assumes gaps between the codewords, having high weight partition at the second part is more logical in terms of ISI. Therefore, a comparison is designed between single and doubly constant weight codes, formally defined as CWC($n = 20$, $m = 4$) and DCWC($n_1 = 10$, $m_1 = 2$, $n_2 = 10$, $m_2 = 3$), respectively. The experimented CWC and DCWC configurations are illustrated in Figure 2.17. As a decoding algorithm, sorting decoder is utilized in the experiment for the both coding schemes. For the DCWC, the sorting decoder is performed in the both partitions for the weights 2 and 3, respectively.

Figure 2.18 shows FER for both CWC($n = 20$, $m = 4$) DCWC($n_1 = 10$, $m_1 = 2$, $n_2 = 10$, $m_2 = 3$) for different sampling periods. and The severe ISI case, where uncoded $t_s = 0.12s$, the superiority of DCWC over CWC in terms of FER is seen in Figure 2.18a. As the uncoded t_s increases so ISI eases, the advantage of DCWC disappears and eventually the performance of DCWC becomes worse than CWC due to increased number of weights. These results experimentally shows the trade-off of increased weights and additional information for the decoding.

2.7. Discussion

Two novel CLWC decoding methods are proposed for the MCvD channels in this chapter. In the literature, mostly low weight codewords have been proposed as channel coding for MCvD due to their power efficiency for nanodevices in case they are used together with OOK. Due to the fact that only transmission is done at ones in OOK, using low weight codewords is advantageous for diminishing ISI in MCvD channel as well. For channel coding CLWC are proposed in this study, since the information that codewords have the same weight can be utilized by the decoder. For the first time in the literature, specialized decoding algorithms are proposed for CLWC in MCvD channel with ISI. First, the MCvD channel model is described and CLWC is presented as coding strategy. Then, two different decoding methods, threshold based and sorting decoders are summarized. Afterwards, a heuristic iterative sorting decoder is presented, and



(a) $t_s = 0.12s$ (b) $t_s = 0.15s$

(c) $t_s = 0.2s$ (d) $t_s = 0.3s$

Figure 2.18: FER results for doubly and single constant weight coding.

the probability of interference reduction at each iteration is examined. Lastly, optimal super trellis decoder is proposed. Constant weight of the codewords are taken into account in addition to ISI, and combined states are formed. Viterbi algorithm is then applied for maximum likely sequence estimation.

The simulation results show substantial reductions in frame errors by applying the proposed decoding methods. Both methods perform better than the decoding methods in the literature for two different channels with uncoded $t_s = 0.2s$ and $t_s = 0.5s$. In addition, CLWC with proposed decoding methods show superior performance

compared to the alternative channel codes for MCvD with CSK under different sample durations, distances and counting noise levels. As expected, the performance of the optimal super trellis method is superior to the other methods at the expense of increased computational complexity at the decoding stage. Depending on channel condition and the computation power at the decoder, one of the two proposed methods can be employed for CLWC decoding at MCvD for improving error performance significantly.

In addition to decoding approaches, transmitted sequence molecule count adjustment and doubly constant weight coding experiments have been carried out. Although some performance enhancement can be achieved with these methods, we prioritize decoder side approaches as the transmitter devices are assumed to be simpler in nano environments [21] and also in our system model.

Our work explained in this chapter is published in [55].

3. METHODS FOR VISIBLE LIGHT COMMUNICATIONS CHANNEL

3.1. Introduction

VLC is a short range optical wireless communication technology that uses visible light for data transmission. In VLC, information is transmitted with an intensity modulation of visible light at a higher rate than a human can perceive [56]. Visible light spectrum offers hundreds of terahertz of unlicensed bandwidth [10, 56] whereas radio frequency spectrum is becoming more scarce each day with the increasing traffic demand. This is one of the motivations for researchers to study VLC.

As transmitters, mostly LEDs are used in VLC. Fast switching ability of LEDs and their increasing prevalence in indoor lighting enable VLC to attain high data rates with the existing illumination infrastructure. The receiving end devices are either photodetectors or image sensors. Photodetectors can transform received light into current with high sampling rate [10]. Since LEDs are already installed for illumination and photodetectors are comparatively cheap communication receivers [57], the cost of device investment is low in VLC. In addition, energy for propagation is already spent for illumination, therefore no extra energy is consumed for communication. Due to those advantages, VLC is considered as a strong complementary or substitute to RF based communication for certain applications.

Although the main use case for VLC is considered to be high-speed indoor downlink communications, VLC is envisaged to have numerous applications such as indoor localization, device to device communications and vehicular communications [56], [10], [58]. Since VLC is used with as a side benefit of existing luminaries in indoor applications, it has to satisfy the requirements of perceived light. This dual use of LEDs results in challenges for VLC. First, VLC methods should support dimming since a user may want to change illumination of the environment in practice. Proposed VLC systems

should support at least a set of different illumination levels. In the literature, different modulation methods are proposed to be able to support dimming as well. Well known OOK is modified in the Standard IEEE 802.15.7 [59] such that either compensation symbols are added or ON amplitude of OOK is adjusted to provide desired brightness. OOK with non-data compensation symbols is called Variable OOK (VOOK) [60] in the literature. Despite OOK is known to be a simple technique to implement, it is shown that more than 200Mbps can be achieved with Non-Return-to-Zero OOK [61] [62].

Second challenge of VLC is the flicker mitigation. Flicker is a noticeable fluctuation in light brightness. Flicker caused by modulation must be mitigated so that the harmful physiological effects of flicker is not experienced by the users. To mitigate flicker, changes in brightness should be faster than a human can perceive. Some modulation techniques such as pulse-position modulation (PPM) or return-to-zero OOK (RZ-OOK) help mitigating flicker since average power over time is constant. However, these methods are not spectral efficient, therefore achievable data rates are lower [58], [10]. To avoid long runs of same brightness level, utilizing coding is proposed in the literature. Run length limited, such as Manchester codes, 4B6B, codes are proposed to avoid long runs of 1s and 0s in the VLC standard [59]. As forward error correction (FEC) codes, shorter and low complexity codes are preferred over advanced codes [63]. In the VLC standard, Reed Solomon (RS) or Convolutional Codes (CC) are proposed with respect to different application areas [59]. FEC codes are combined with RLL codes to support also flicker mitigation. In [64], constant weight codes for dimming control is proposed.

Channel modeling is another research topic in the VLC literature [65], [66]. Having realistic VLC channel models is critical for system design, therefore IEEE802.15.7r1 Task Group has endorsed reference channel models for indoor environments [67]. Using ray tracing methods, it is determined that there are large number of reflections besides the line of sight (LOS) component. Thus, intersymbol interference is experienced when symbol period is less than the multipath delay spread. This study utilizes these realistic reference channel impulse responses to simulate the performance of the

proposed method. Another integral part of channel modeling is noise modeling. The main noise types in VLC are ambient light noise, shot noise and thermal noise. Shot noise is generated by the random nature of photon collection in the photodetector. Due to Poisson process of photon collection [10], the variance of collected photons depends on the average value. This constitutes the input-dependent nature of the shot noise. On the other hand; thermal noise, which is caused by the signal pre-amplification in photodetector, and the ambient noise are considered to be input-independent. Many publications assume only the input-independent part of the noise in their models, but we employ both terms in our channel model for being more accurate as in [68].

Considering dimming support, flicker mitigation, reducing error rate and low complexity constraints of VLC communication system, we advocate optimizing the whole system as an end-to-end solution rather than separate modules. This challenging task requires modeling transmitter and receiver as complex transformations and optimizating many parameters of the transformations to decrease defined loss. Deep learning models are proven methods to approximate complex transformations through composition of simple operations [69]. Deep learning eliminates the need of hand-crafted feature engineering by automatically extracting high-level features from data directly. Neural networks and specifically deep learning has gained significant popularity in communications literature as well. Network optimization, network traffic control, indoor localization and signal processing are some application areas where researchers make use of deep learning [69]. Closer to the focus of our work, deep learning is also used for channel decoding and detection in the literature. In [70], an improved version of belief propagation is proposed. The authors unfold Tanner Graph into a dense neural network and extend the BP algorithm by assigning weights to the edges. Then, they applied conventional training methods of deep learning. In [71], a convolutional neural network (CNN) is used to estimate noise more accurately in the presence of correlated noise to increase the performance of the BP decoder. In [72], the authors suggest using an autoencoder as an end-to-end communication system. The transmitter and receiver, which are modeled as dense neural networks, are jointly optimized [72]. The authors also proposes radio transformer networks (RTN), which provides correction for

estimated channel impairments. Channel impairment parameters are learned with a neural network and then compensated by RTN. In [73], the authors extend the idea of RTN and show with software defined radios that the practical performance of their proposed method over the air is close to the baseline.

As deep learning based end-to-end communication systems for VLC, only a few number of studies exist in the literature. In [74], the whole VLC communication system is modeled as an autoencoder. The paper also provides a gradual binarization method for codewords that alleviates vanishing gradient problem. For the dimming, the authors modify the loss function by adding a term that penalizes when the sum of codewords differ from the desired value. However, adding such a penalty does not guarantee aimed level of sparsity, as also described in [75]. On the other hand, flicker reduction and ISI in channel model are not considered in [74], which are covered in our study. In [76], use of autoencoder as an end-to-end solution for optical wireless communications is proposed. Since the paper does not directly aim VLC, flicker reduction and dimming are not considered. In [77], authors propose using convolutional autoencoder for image sensor communication. Channel imperfections as random image rotations, blurs and signal-dependent noise are considered in the study. An annealing based multi-stage training strategy is employed to obtain OOK symbols. A deep learning system model producing dimming control OOK codes is investigated in [78]. Target dimming level is fed as a part of input for the encoder and the decoder of the proposed system. Thus, multiple dimming constraints can be trained at once. In addition, a probabilistic method is proposed for the binarization of the codewords unlike [74] and [77].

The all VLC communication blocks are modeled as a single novel autoencoder in this work. This autoencoder provides the lighting requirements as well as its main communication duties. We call this autoencoder that is specialized to VLC as VLCnet. As far as we know, this is the first work in the literature that an autoencoder both performs the flicker reduction and dimming. To achieve these illumination needs, VLCnet includes a novel activation unit, flicker reducing activation unit (FRAU). FRAU is a member of locally competitive activation units. The neurons are divided into blocks in

FRAU and only a certain number of them could be active. The active neuron values are then mapped to favorable values with a modified sigmoid function for reducing flicker. Therefore, FRAU is located at the end of the encoder to generate desired codewords. In this work, it is shown that VLCnet with FRAU can effectively be trained to obtain message at the receiver. The effects of FRAU parameters on the FER are analyzed. The error rate performance of VLCnet is compared with other methods and it performs demonstrably superior. Despite different noise levels in the inference stage, it is shown that the models trained with a certain noise level performs decently. This results increases implementability of the proposed method by eliminating the need of storing many models in the real use case. This outcome can be valuable for other deep learning studies in communication context.

The rest of the chapter is organized as follows: Section 3.2 provides review of the system model, the autoencoder concept, assumed channel model, novel FRAU layer, VLCnet in overall and the training method. Simulation setup and experimental performance evaluations are presented in Section 3.3. Practicality of the proposed system is discussed in Section 3.4. A further discussion on FRAU is provided in Section 3.5. Finally, Section 3.6 concludes the chapter.

3.2. System Model

3.2.1. Autoencoders

Autoencoders are neural networks that are trained to reconstruct its input at the output. The inputs, $\boldsymbol{u}$, are mapped to a hidden representation, $\boldsymbol{z} = f(\boldsymbol{u}, \theta_f)$, where f is the neural network transformation and θ_f denotes the parameters of the f. Then this hidden representation, $\boldsymbol{z}$, is mapped to the output $\hat{\boldsymbol{u}} = g(\boldsymbol{z}, \theta_g)$, where θ_g denotes the parameters of the network, g. Commonly the first part of the autoencoder, which is $f(.)$, is called encoder; where $g(.)$ is called decoder [79]. The learning process minimizes

a loss function

$$L(\boldsymbol{u}, g(f(\boldsymbol{u}, \theta_f), \theta_g)) \tag{3.1}$$

between the input and output of the autoencoder over the trainable autoencoder parameters θ_f, θ_g. In Figure 3.1, a shallow autoencoder with one hidden layer is shown. Nodes with 1's denote bias values and the dissimilarity loss between u and $\hat{u}$ is minimized. For adding more nonlinearity, deeper autoencoders can be utilized.

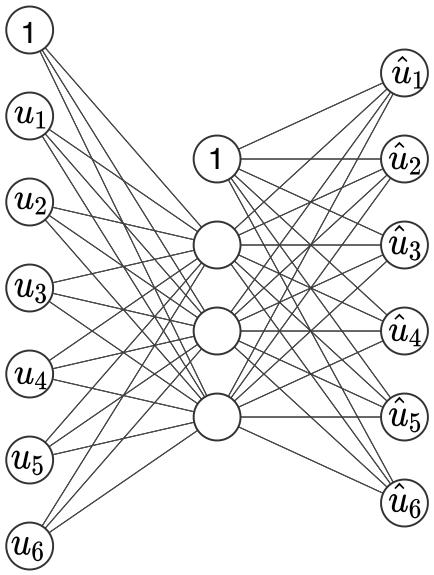


Figure 3.1: A sample autoencoder

Constraining the whole autoencoder network or hidden units, $\boldsymbol{z}$, helps exploring the patterns in the input data, $\boldsymbol{u}$. Most common usage of it is dimensionality reduction. Dimension of z is forced to be smaller than the dimension of u, so that the autoencoder learns to extract only the most important features. Another widely used implementation is sparse autoencoders. Having larger dimension for z than u but imposing sparsity constraint on z makes autoencoder learn features that can be useful for other tasks like classification. Sparsity is generally achieved by adding a sparsity penalty to (3.1).

The proposed VLCnet is an autoencoder and it consists of 3 main blocks. Additional to the encoder and decoder blocks of the traditional autoencoders, VLCnet also involves a channel block to model channel effects on the transmitted codewords. The parameters of the encoder and decoder modules are learned in the training phase.

3.2.2. Channel Model

For the channel block of VLCnet, a multipath channel is assumed to be as close as possible to the realistic VLC indoor channels in [67]. From the available indoor scenarios, manufacturing cell is selected for the simulations since it exhibits strong ISI behavior. Specifically, the selected channel impulse response (CIR) for the simulations is scenario 4 CIR at receiver location 6 from LED 1-6 [80]. Then this channel is downsampled with respect to the symbol rate. In Figure 3.2, the CIR from [80] and its sampled version at 6 ns are shown.

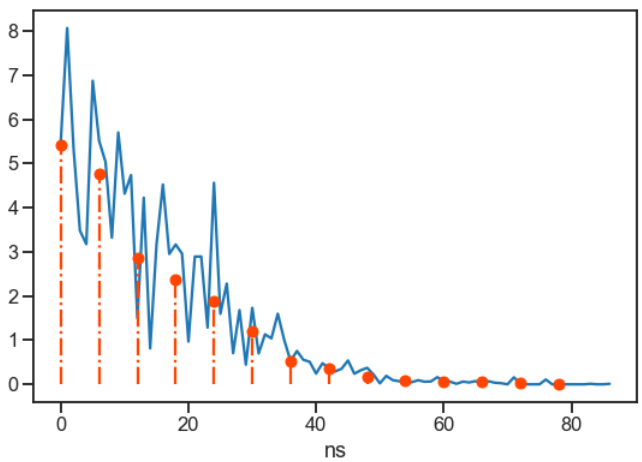


Figure 3.2: CIR for Scenario 4 - LED1-6 to D6 [80]

Transmitted codewords, $\boldsymbol{v}$, are convolved with the CIR, $\boldsymbol{h}$. The resultant vector is $\boldsymbol{r}$, which is

$$r_i = \sum_{k=1}^{i} h_k v_{i-k+1}. \tag{3.2}$$

Input-dependent shot noise and input-independent thermal noise can be modeled as Gaussian noise [68]. With both noise terms, the received vector $\boldsymbol{y}$ becomes

$$y_i = r_i + \sqrt{r_i} n_{sh}[i] + n_{th}[i] \tag{3.3}$$

where n_{th} is the thermal noise and $n_{th} \sim \mathcal{N}(0, \sigma^2)$; n_{sh} is the shot noise and $n_{sh} \sim \mathcal{N}(0, \varsigma^2\sigma^2)$. n_{sh} and n_{th} are assumed to be independent. The ς^2 term describes the strength of the input-dependent noise over the input-independent n_{th}.

3.2.3. FRAU: Flicker Reducing Activation Unit

FRAU is the activation layer that is placed at the end of encoder part of the VLCnet. It enforces some portion of the output to be zero. The remainder nodes are the "active" or non-zero ones. The selection of active and passive nodes depends on the result of the competition among them. The neural networks containing these type of activation functions are called competitive networks or locally competitive networks [81]. These activation functions are different than widely used *sigmoid* or *tanh* functions such that they are not continuously differentiable [81]. However, these networks can effectively be trained by backpropagating the error only through the active nodes [75, 82]. In locally winner takes all (LWTA) activation layer [82], the neurons are divided into blocks and only the maximum valued neuron in each block is forwarded while the others are zeroed out. In k-sparse autoencoder [75], the layer is not divided into blocks and instead of one neuron, k largest neurons of the layer are permitted to keep their value while others are forced to be zero. In both layers, any other non-linear function is not employed to the active neurons, since there is a non-linearity originated from the selection of the active neurons [75]. In this chapter, we extend the concepts introduced by k-sparse and LWTA layers with FRAU. FRAU separates the whole set of neurons, which are length of n, into blocks of length l. Instead of only one element, the largest c elements of each block are labeled as active and the rest is set to zero.

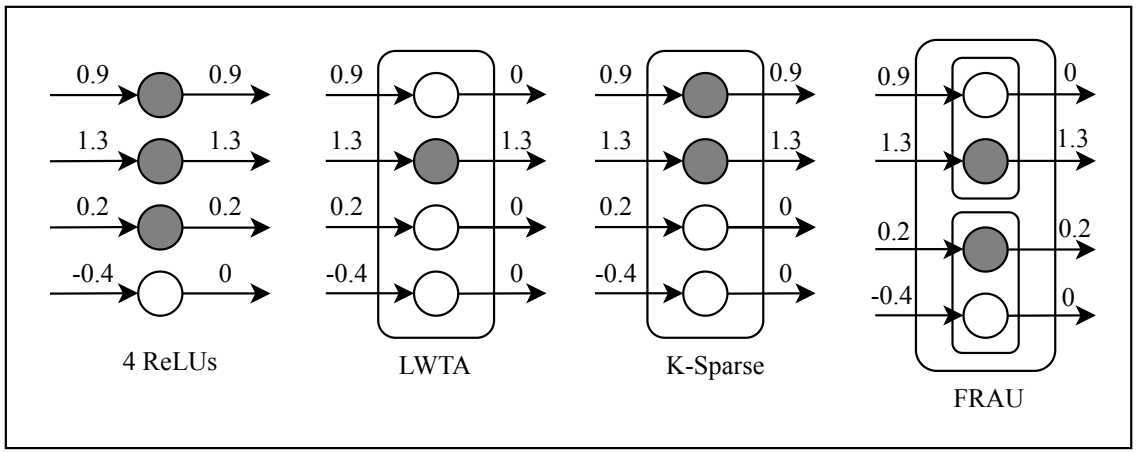


Figure 3.3: Comparison of ReLU, LWTA with block length is 4, K-sparse with $k = 2$ and FRAU with $(l = 4, c = 2)$

As seen in Figure 3.3, locally competitive layers are different than ordinary activation functions such as rectified linear unit (ReLU). In competitive layers, the inputs are not subject to individual functions, instead they are competing against each other to be activated.

Another difference of FRAU is that a modified sigmoid function is applied to the active neurons. Therefore the whole activation function becomes

$$frau_{\lambda,l,c}(x) = \begin{cases} mSigmoid_{\lambda}(x), & \text{if } x \in \Gamma \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

where $\Gamma = supp_{l,c}(\boldsymbol{x})$ and $supp_{l,c}(\boldsymbol{x})$ is an operator that returns the set of elements that are the c largest in each block of consecutive l neurons. $mSigmoid_{\lambda}$ denotes the modified sigmoid and can be expressed as

$$mSigmoid_{\lambda}(x) = \frac{1 + \lambda e^{-x}}{1 + e^{-x}}. \tag{3.5}$$

Modified sigmoid function is differentiable and its derivative for using in backward calculations can be written as

$$\frac{\partial\, mSigmoid_{\lambda}(x)}{\partial x} = \frac{\lambda + e^{x} - \lambda(1 + e^{x})}{(1 + e^{x})^{2}}. \tag{3.6}$$

This expression can be simplified to $(1 - \lambda)$ times the first order derivative of the ordinary sigmoid function.

The certain number of active nodes guarantees the sparsity throughout the layer. The reason constraining them to be separated into blocks is that the maximum number of consecutive zeros or active nodes need to be limited. As aforementioned, both k-sparse and LWTA layers do not employ any function to the neurons besides the competition. However, having identity function after selecting max c in each l does assure that the active nodes are not close to zero, thus any sparsity constraint might not be satisfied all the time. To illustrate, the input neurons may all have negative or close to zero values, so choosing c active neurons in such a block would result in values that are again very close to zero, or passive neurons. Since that can cause violation of the flicker constraint, the active neurons are employed to a sigmoid function which is modified such that its output range is not (0, 1) but (λ, 1). The reason of this operation to guarantee the maximum run-length by separating the value ranges of active and passive nodes. An illustrative diagram of FRAU with $l = 5$, $c = 3$, $\alpha = 0.6$ can be seen in Figure 3.4.

The maximum number of consecutive active or passive neurons defines the maximum run length of the produced codewords by the VLCnet. Since two neighbor blocks can have c active or $(l - c)$ passive neurons, the maximum run length of a FRAU layer can be written as

$$MaximumRunLength = max(2c,\ 2(l - c)). \tag{3.7}$$

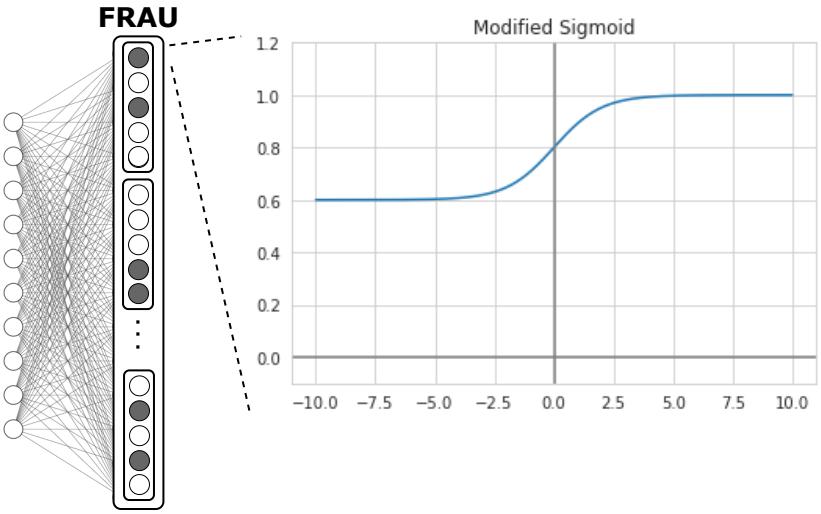


Figure 3.4: An example FRAU layer of $l = 5$, $c = 3$, $\lambda = 0.6$

The maximum run length is shortened to $max(2c,\ 2(l-c))$ from n thanks to FRAU. The average amplitude of the FRAU outputs, the transmitted codewords, over the codebook determines the dimming. Dimming is expressed as

$$d_{\theta_e} = \frac{1}{Mn} \sum_{M} \sum_{i=1}^{n} v[i], \tag{3.8}$$

$0 \geq d_{\theta_e} \geq 1$, $\boldsymbol{v}$ is the output of FRAU and M is the number of input messages. The input messages, thus the codewords are considered to be equiprobable. Dimming both depends on c/l and λ in VLCnet as the amplitude of $\boldsymbol{v}$ depends on these. VLCnet encoder parameters, θ_e, can be regarded as random variables mainly caused by the random initialization of the weights and biases at the beginning of the training. This causes the input of the FRAU, $\boldsymbol{s}$, being sampled from a dependent distribution. Since no assumption can be made about the distribution of $\boldsymbol{s}$, $\mathbb{E}[\boldsymbol{v}]$ cannot be computed in closed form for a given l, c and λ. FRAU function in (3.4) sets the upper and lower bounds for d_{θ_e} as

$$\frac{c\lambda}{l} \leq d_{\theta_e} \leq \frac{c}{l}, \tag{3.9}$$

for $0 \leq \lambda \leq 1$. These bounds are utilized as dimming targets before training, however the actual d_{θ_e} is known once the training is completed.

3.2.4. VLCnet & Training Procedure

VLCnet consists of trainable transmitter, static channel and trainable receiver blocks. At the transmitter block, first the binary input message $\boldsymbol{m} \in \mathbb{R}^k$ is mapped to $\boldsymbol{e_m}$ of length 2^k, the all-zero vector except m^{th} element being 1. Then this vector passes through a fully-connected (FC) layer, where it is multiplied with weights and added with bias. The FC layer at the transmitter does not consist any activation function other than FRAU. The output of this linear operation is fed into a batch normalization layer, where statistically inferred mean of values of this layer is subtracted from the outputs of FC layer and these values are divided by the inferred standard deviation. Batch normalization is performed in VLCnet to regulate the distributions of the inputs neurons of the FRAU [79]. This operation facilitates the training of VLCnet. To add non-linearity and generate the codewords that are able to satisfy certain dimming and flicker constraints, batch normalization output, $\hat{s}$, is fed into a FRAU layer. The activations of FRAU layer are the codewords to be transmitted. The channel layer of VLCnet consists of a convolution with ISI taps and an addition of random noise. Random noise can include input-dependent term, as explained in 3.2.2. The channel impulse response is assumed to be known at the training phase. Received codewords, impaired by the channel, are first processed by a 1-D convolutional layer. The pattern generated by channel impulse response is expected to be learned from the filters of the convolutional layer since the channel itself imposes a convolution. Therefore, the convolutional layer as the first layer of the receiver side is an essential block of the VLCnet. Depending on the message length and complexity of the receiver, a number of FC layers follows the convolutional layer. After the last FC layer, a softmax function [79] is performed to transform neuron outputs to the probabilities of classes. Lastly, the bit with highest estimated probability is chosen with the argmax operation. Hence, the decoded one hot vector is obtained. Figure 3.5 shows the components of the VLCnet.

In the training of the VLCnet, channel impulse response h and statistical parameters of the random channel, namely σ and ς are assumed to be known. A random batch of input messages are generated and passed through the aforementioned layers of

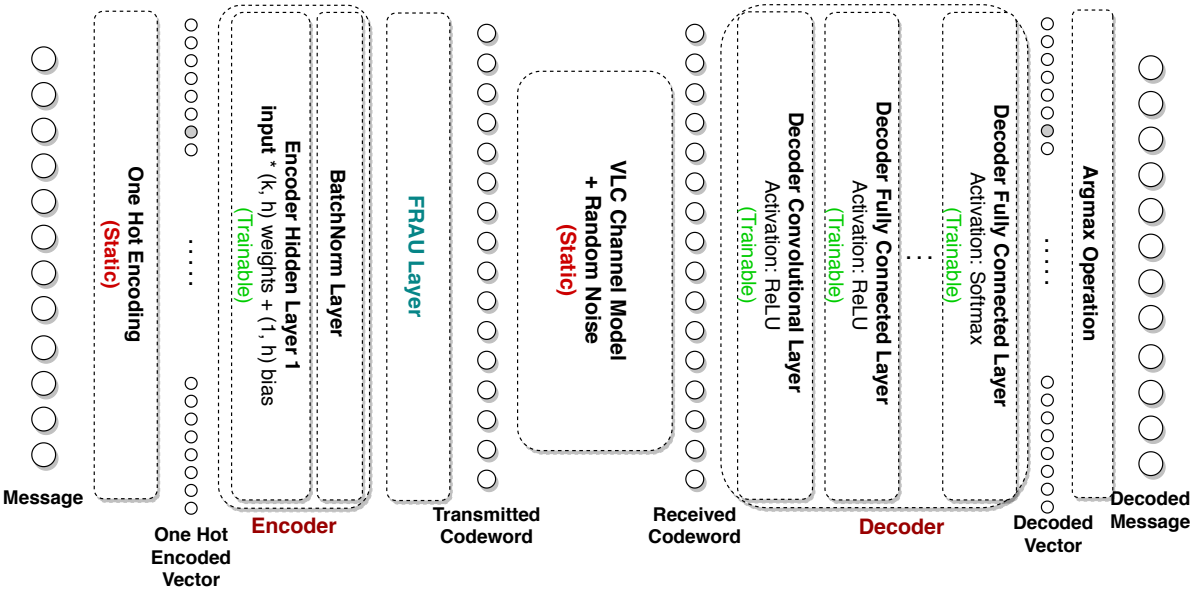


Figure 3.5: Diagram of VLCnet system model

the VLCnet. Just for the training, an extra operation before FRAU is performed. This operation is dropout, which zeroes out some of the neuron values and their connections with a given probability. This operation is done only in training to reduce over-fitting and increase generalization of the network [79]. Weights and biases are randomly initialized in VLCnet and that might cause setting same neurons active, independent from the inputs. Since gradients only flow through the active neurons, the connections belong to the other neurons might not find a chance to update themselves and the advantage of the those neurons maintain. This lowers the representation capability of the network. Although, batch normalization layer alleviates this problem, we suggest randomly shutting some neurons to enable the training of all neurons. By simulations, having dropout with rate of around 0.2 is found to increase the error rate performances. In the literature, dropout is found to be useful in other locally competitive layers as well [81]. Once a forward pass is performed, the loss that penalizes the mismatch between the input message and the decoded message is calculated. Categorical cross entropy loss, expressed as (3.10), is used in VLCnet.

$$\mathcal{L}_{CCE} = -\sum_{i=1}\sum_{j=1}^{2^k} e_m[j] \log \hat{y}[j] \tag{3.10}$$

In (3.10), the first summation is for adding the loss of the different one-hot encoded message vectors $\boldsymbol{e_m}$, and predicted outputs $\hat{\boldsymbol{y}}$ in a batch.

Error minimization subject to flicker reduction problem is formulated as an optimization by

$$\begin{aligned} &\min_{\theta} \mathcal{L}_{CCE} \\ &s.t.\ 0 \leq v_i \leq \lambda,\ v_i \in \Gamma, \\ &\quad v_j = 0, \qquad v_j \in \widetilde{\Gamma}, \\ &\quad 0 \leq \lambda \leq 1, \end{aligned} \tag{3.11}$$

where $\widetilde{\Gamma}$ is the complement set of Γ. The constraints in (3.11) ensures the limitation of maximum run length and bounds the codeword symbol energy. Note that FRAU provides a built-in support for these constraints, consequently VLCnet can be trained with conventional categorical cross entropy loss.

The trainable parameters of the encoder and the decoder are updated using stochastic gradient descent (SGD) or any other optimization algorithm like Adam [83]. In SGD, the parameters of VLCnet are updated at each batch as

$$\theta[t] = \theta[t-1] - \eta \nabla_{\theta} \mathcal{L}_{CCE}\big(\theta[t-1]\big) \tag{3.12}$$

where $\theta[t]$ denotes the network parameters at step t, η denotes the learning rate and $\nabla_{\theta} \mathcal{L}_{CCE}\big(\theta[t-1]\big)$ denotes the gradient of the cost function with respect to the network parameters. Training continues iteratively until the end of predetermined epoch count or loss in validation set, which can be formed as all possible 2^k input messages, is no longer decreasing. Consequently, the model for given noise statistics is obtained. This process is explained in detail in the algorithm in Figure 3.6. Yet, this is very inconvenient in terms of practicality to have different models for changing noise variances in the real deployment. Therefore, we propose deployment of only one model, which is trained with a noise variance that produces best results for whole noise variance

range in overall. Indeed, simulations show that the model works best in overall even works better compared to using individual models that are trained specifically for that noise level. We propose using grid search for the selection of the noise variance that is to be deployed. For [0, 40] dB $E_b/N0$ range, different models are trained with 5 dB intervals. Next, frame error rate (FER) of those models are evaluated for whole signal-to-noise ratio (SNR) range. The model that has minimum sum of logarithm of FER values is selected. Before this operation; if FER values are zero, they are replaced with multiplicative inverse of number of transmitted input frames for the sake of numerical stability. The whole training process for choosing the best VLCnet is explained in the algorithm in Figure 3.7. Although we use grid search for the optimization of noise level hyper-parameter, other methods like random search [84] or bayesian optimization [85] can also be used. However, these are left for future investigations.

A set of signal outputs generated during a sample VLCnet operation is illustrated in Figure 3.8. Here, the parameters are set as $k = 12$, $n = 20$, $l = 5$ and $c = 2$. The bias for the active led is set to 0.6. The message in the top-left figure is encoded into a codeword in top-mid figure by the encoder part of the VLCnet. Despite the ISI and noise, the VLCnet achieves successful decoding of the original message as shown in the bottom-left figure.

3.3. Performance Evaluation

The effectiveness of proposed VLCnet on error rate performances is evaluated via computer simulation. As aforementioned, the channel is modeled as Scenario 4 - D6 in [67]. The gap at the beginning of the channel impulse response is removed since receiver is considered to be synced accordingly to maximize received energy. ISI taps are obtained from this CIR for input symbol duration value $t_s = 10ns$. Results are both generated for the channels

- $\varsigma = 0 \rightarrow$ **without** input-dependent noise,
- $\varsigma = 1 \rightarrow$ **with** input-dependent noise.

Input: $N0$, h, ς, θ_{Conv}, θ_{FC}, l, c, λ, $maxIter$

Output: VLCnet with trained parameters

Initialisation : random initialization of weights and biases
for $i = 1 : maxIter$ **do**
  $\boldsymbol{m} \leftarrow$ Generation of random batch of messages
  $(\boldsymbol{e_m} \leftarrow OneHotEncoding(\boldsymbol{m})$
  $\boldsymbol{s} = FC(\boldsymbol{e_m}, \theta_{FC_{enc}}, \phi = None)$
  $\hat{\boldsymbol{s}} \leftarrow BatchNorm(\boldsymbol{s})$
  $\hat{\boldsymbol{s}} \leftarrow Dropout(\hat{s}, rate = 0.2)$
  $\Gamma \leftarrow$ Set of the largest c in each block of length l of $\hat{\boldsymbol{s}}$
  for $j = 1{:}n$ **do**
    if $v_j \in \Gamma$ **then**
      $v_j = fms(v_j)$
    else
      $v_j = 0$
    end if
  end for
  $\boldsymbol{y} \leftarrow Channel(\boldsymbol{v}, \boldsymbol{h}, N0, \varsigma)$
  $\hat{\boldsymbol{y}} \leftarrow Conv1D(\boldsymbol{y}, \theta_{Conv}, \phi = ReLU)$
  for $f = 1{:}FCLayerCount$ **do**
    $\hat{\boldsymbol{y}} \leftarrow FC(\boldsymbol{y}, \theta_{FC}, \phi = ReLU)$
  end for
  $\hat{\boldsymbol{y}} \leftarrow Softmax(\hat{\boldsymbol{y}})$
  Compute the loss between $\boldsymbol{e_m}$ and $\hat{\boldsymbol{y}}$
  Backpropagate the error only through the x in Γ and update $\theta_{Conv}, \theta_{FC_{enc}}, \theta_{FC_{dec}}$
end for
return $\hat{v}$

Figure 3.6: Algorithm for Training VLCnet for a given N0: *trainVLCwithN0(h, N0, ς, θ_{Conv}, θ_{FC})*

Input: h, ς, θ_{Conv}, θ_{FC}, l, c, λ, $maxIter$

Output: VLCnet with trained parameters

for $\alpha = 0{:}5{:}40$ **do**
  $N0 = E_b/10^{\alpha/10}$
  VLCnet(α) = $trainVLCwithN0(N0, h, \varsigma, \theta_{Conv}, \theta_{FC}, l, c, \lambda, maxIter)$
end for
for $\alpha = 0{:}5{:}40$ **do**
  for $\gamma = 0{:}5{:}40$ **do**
    $E_b/N0 \leftarrow \gamma$
    $FER(\alpha, \gamma) \leftarrow$ Compute FER for each $E_b/N0$ by sending 10^7 input frames to
    if $FER(\alpha, \gamma) = 0$ **then**
      $FER(\alpha, \gamma) = 10^{-7}$
    end if
  end for
end for
$\alpha^* \leftarrow \text{argmin}_\alpha \sum_{\forall\gamma} log(FER(\alpha, \gamma))$
return $VLCnet(\alpha^*)$

Figure 3.7: Algorithm for training VLCnet: $trainVLC(h, \varsigma, \theta_{Conv}, \theta_{FC})$

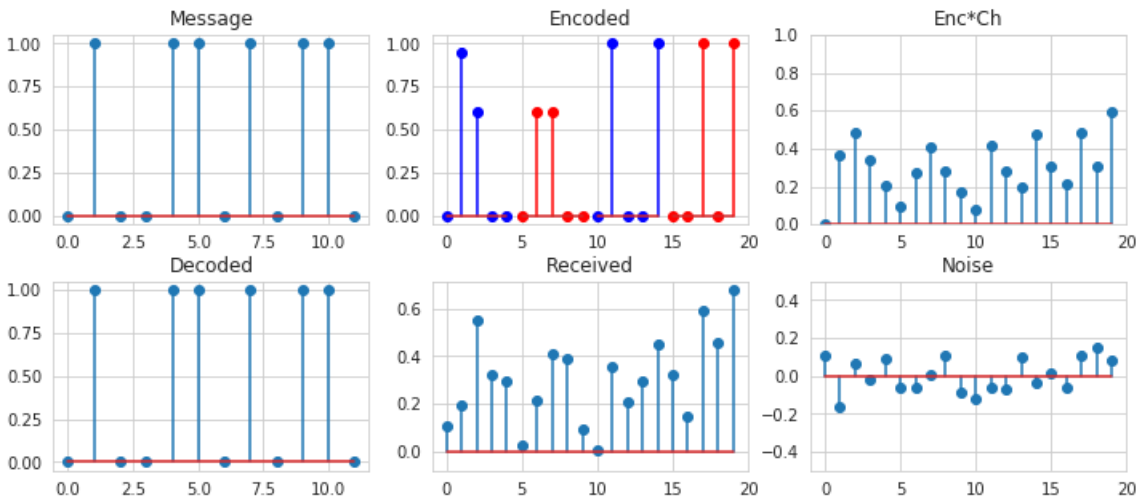


Figure 3.8: Sample Operation of VLCnet for $(k = 12, n = 20, l = 5, c = 2, \lambda = 0.6)$

The FER values are obtained for different information bit energy levels. The Monte Carlo simulations are carried out and the results are averaged by transmitting 10^7 input frames. Input frame length, k, and codeword length, n are kept constant through simulations, where $k = 12$ and $n = 20$. Different simulations are performed for various values of block length (l), number of active neurons in a block (c) and led turn-on level (λ).

The sampling of the channel is normalized with $\frac{k}{n}$. As in (3.9), average codeword active symbol energy, E_s, bounds are written as

$$E_b \frac{k}{n} \frac{l}{c} \leq E_s \leq E_b \frac{k}{n} \frac{l}{c} \frac{1}{\lambda^2}. \tag{3.13}$$

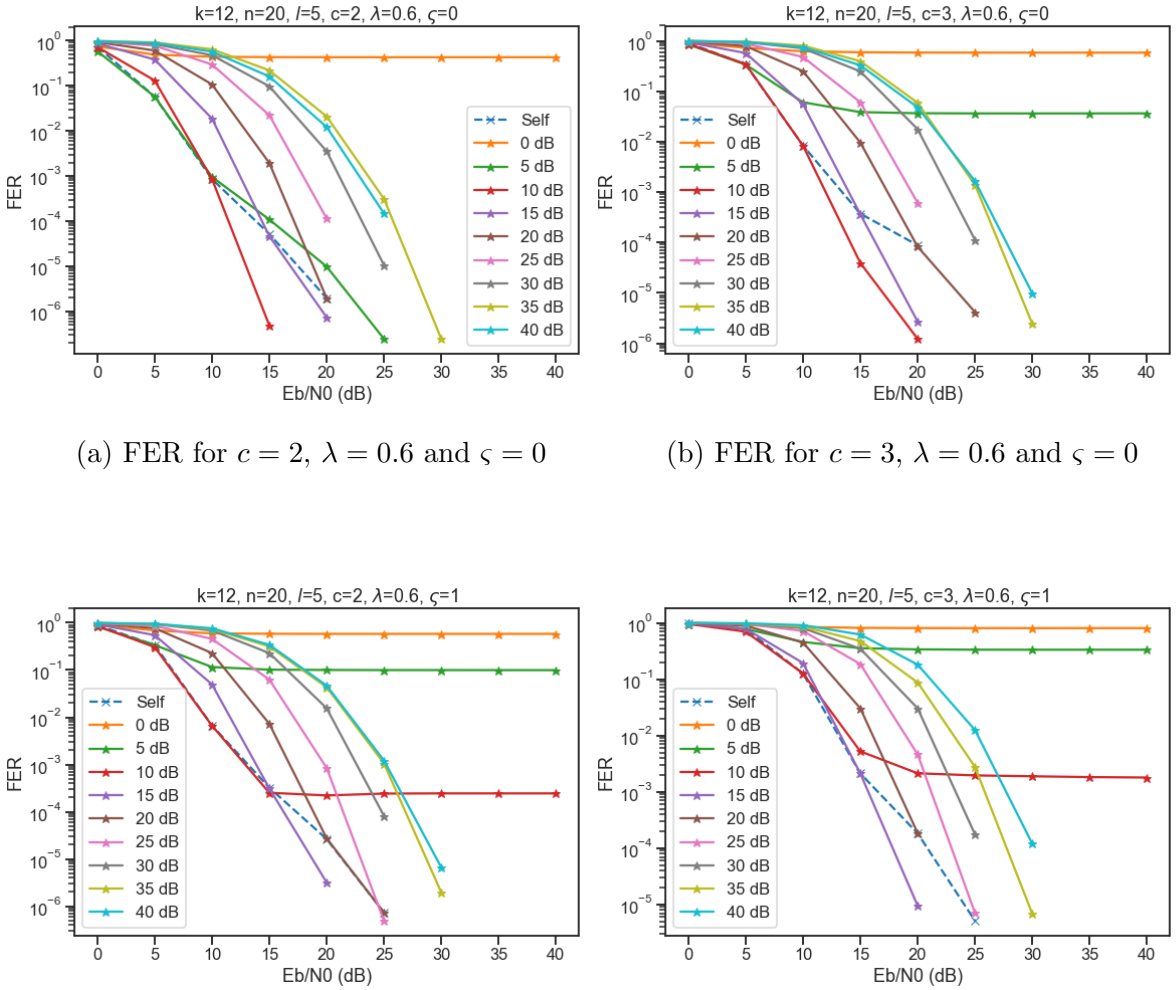
Despite it brings a disadvantage for VLCnet, E_s is assumed to be equal to $E_b \frac{k}{n} \frac{l}{c}$ throughout the simulations. However, the exact value of E_s can be calculated for the whole codebook once the training is completed, and the output of the VLCnet encoder can be scaled accordingly to utilize the unused energy. The variance of the input-independent thermal noise is set to $N0/2$. System configurations of the encoder and the decoder sides of the simulated VLCnet is given in the Table 3.1. Adam [83] method is used as optimizer and learning rate is set to 0.005 during the simulations. Training with a specific noise variance takes 35000 epochs, where 2^{12} random input messages are generated in each epoch. These selections of parameters are found to perform well. Further optimization could be done through neural architecture search [86] or hyperparameter optimization [79] but these are very costly in terms of computation and beyond the scope of this work.

Table 3.1: Simulated VLCnet architecture

Layer	**Type**	**Output Shape**	**Kernel Size**
E1	Dense	20	-
E2	BatchNorm	20	-
E3	FRAU	20	-
D1	Conv. + ReLU	16 x 30	5
D2	Flatten	480	-
D2	BatchNorm	480	-
D3	Dense + ReLU	100	-
D4	BatchNorm	100	-
D5	Dense + ReLU	50	-
D6	BatchNorm	50	-
D7	Dense + ReLU	50	-
D8	BatchNorm	50	-
D9	Dense	4096	-

3.3.1. Effect of Number of Active Nodes on FER

One way to control the illumination level is to change number of active nodes in a block. FER results for $c = 2$ and $c = 3$ are simulated where $l = 5$. As described in the algorithm in Figure 3.7, different models are trained with 9 different levels of noise variance such that $E_b/N0$ is one of $0, 5, 10, 15, 20, 25, 30, 35, 40$ in dB. For $c = 2$ and in input-independent noise condition ($\varsigma = 0$), trained model with 0 dB $E_b/N0$ does not work well at any SNR region, as seen in Figure 3.9a. On the other hand, other trained models are able to decrease the FER as noise variance decreases compared to the information bit energy. The model trained with 10 dB $E_b/N0$ is the best model in terms of the selection criteria in the algorithm in Figure 3.7. It performs better than other models in all $E_b/N0$ values but 5 dB. However; the performance difference is small in 5 dB and all models cannot perform well under that much noise, therefore it can be stated that only 10 dB model is adequate for all noise values under given parameters. In all sub-figures of Figure 3.9, curves with "Self" label denote the



(a) FER for $c = 2$, $\lambda = 0.6$ and $\varsigma = 0$

(b) FER for $c = 3$, $\lambda = 0.6$ and $\varsigma = 0$

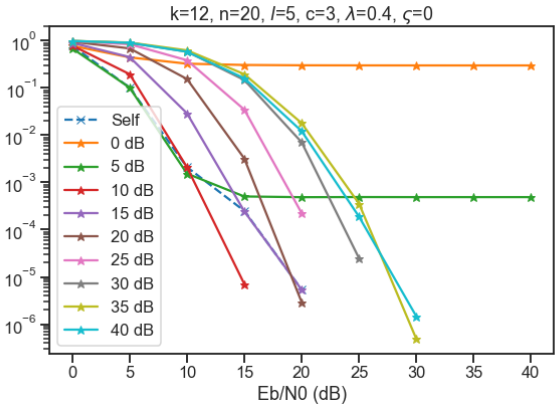
(c) FER for $c = 2$, $\lambda = 0.6$ and $\varsigma = 1$

(d) FER for $c = 3$, $\lambda = 0.6$ and $\varsigma = 1$

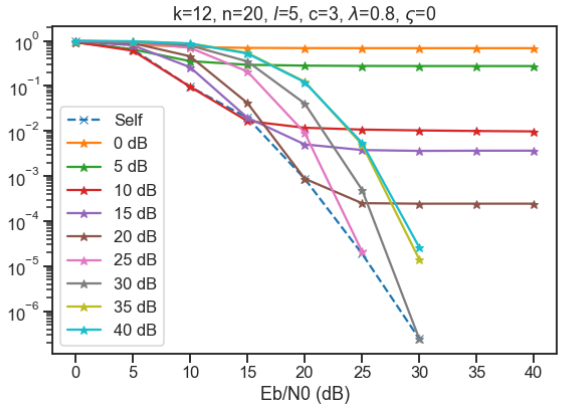
Figure 3.9: The effect of c on FER both with $\varsigma = 0$ and $\varsigma = 1$

evaluation of the models at the noise values where each of them are trained with. Thus, it utilizes all 9 models. Surprisingly, the model trained with 10 dB performs better than the combination of all models in terms of overall performance. For example, the model trained with 10 dB outperforms the model trained with 15 dB in $E_b/N0 = 15$ dB. Training with extra noise makes the model achieve better error correction results. It works as a regularization and increases the generalization ability of the model. This is an important observation and it justifies deploying only one model.

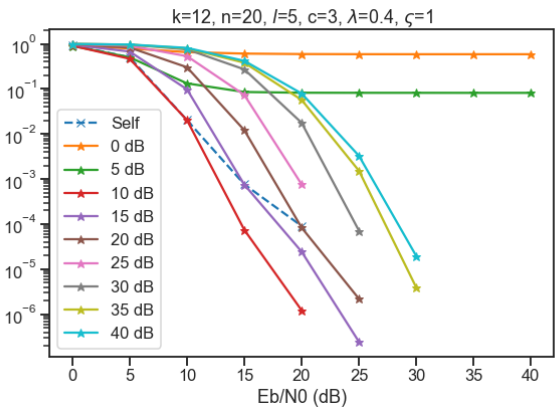
In Figure 3.9b, the FER values for $c = 3$ and input-independent noise are shown. Here training with 5 dB does not work as well as 0 dB. For these models, FER



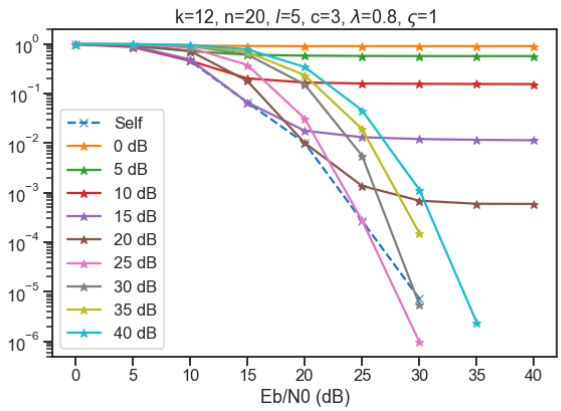
(a) FER for $c = 3$, $\lambda = 0.4$ and $\varsigma = 0$



(b) FER for $c = 3$, $\lambda = 0.8$ and $\varsigma = 0$



(c) FER for $c = 3$, $\lambda = 0.4$ and $\varsigma = 1$

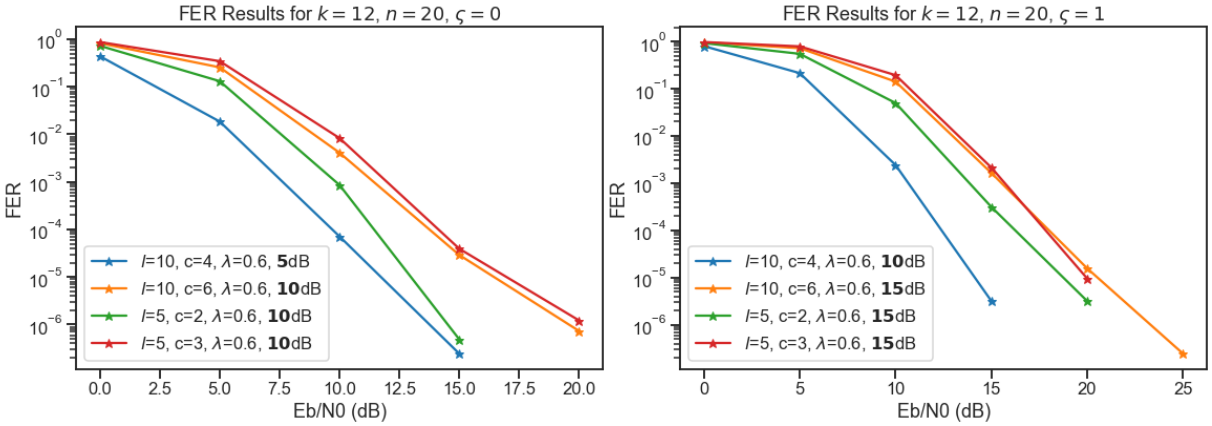


(d) FER for $c = 3$, $\lambda = 0.8$ and $\varsigma = 1$

Figure 3.10: The effect of λ on FER both with $\varsigma = 0$ and $\varsigma = 1$

values stall as $E_B/N0$ increases. The model trained with 10 dB again outperform the other models as well as the self evaluation of all together. When power consumption is constant, increasing c diminishes the average energy per active neurons. The FER performance is declined compared to $c = 2$ case.

In Figure 3.9c, the FER values for $c = 2$ and $\varsigma = 1$ are illustrated. With the existence of input-dependent noise, the error rates are deteriorated compared to $\varsigma = 0$ in Figure 3.9a. The models that are trained with 0, 10 and 15 dB $E_b/N0$; the error floor occurs and the performance does not become better as $E_b/N0$ increases. This is due to additional error term in (3.3) compared to the $\varsigma = 0$ condition. Error rates



(a) FER for $\varsigma = 0$ (b) FER for $\varsigma = 1$

Figure 3.11: FER effect of l on FER when c/l is constant

below 10^{-6} can only be seen for $E_b/N0 > 20\ dB$ although these rates can be achieved at 15 dB in the case for $\varsigma = 0$. The model that achieves best error rate performance for the given parameters is the one which is trained with 15 dB.

Simulation results for $c = 3$ and $\varsigma = 1$ are demonstrated in Figure 3.9d. As expected, the error rate performance is decreased with the addition of input-dependent noise. Similar to Figure 3.9c, the best performing model is 15 dB. Both Figure 3.9c and Figure 3.9d show that VLCnet is able to function even with input-dependent noise.

3.3.2. Effect of λ on FER

The λ parameter in (3.5) is used to control the minimum value of active nodes in FRAU. It directly affects dimming. For $c = 3$, 3 different λ values are simulated and error rate performances are demonstrated in Figure 3.10 and Figure 3.9b-d. In Figure 3.10a, FER performances when $\lambda = 0.4$ and input-independent noise case is shown. Lowering the λ from 0.6 (see Figure 3.9b) to 0.4 increases the error rate performances as the dynamic range of the FRAU output is increased. The model trained with 10 dB $E_b/N0$ produces the best error rate performances. For $E_b/N0 > 15\ dB$, $FER < 10^{-5}$. In contrast, the error performances degrade significantly for $\lambda = 0.8$. None of the models trained with different amount of noise levels can achieve $FER < 10^{-5}$ for

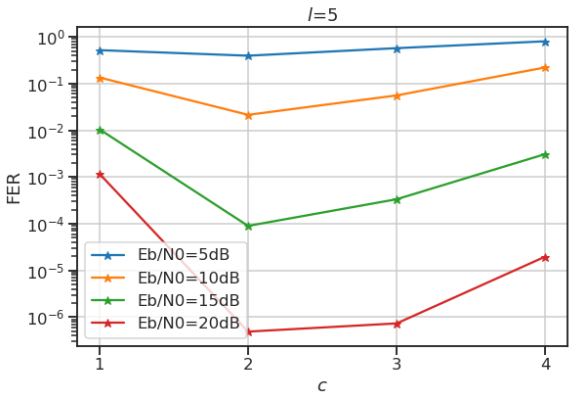
$E_b/N0 < 25\ dB$. In the moderate and high levels of noise, $\lambda = 0.8$ cannot produce decent error rate performances due to very narrow dynamic range of activations that are not zero. The model trained with $E_b/N0 = 25\ dB$ produces the best results where noise level is low ($E_b/N0 > 25\ dB$). Figures 3.10c and 3.10d show the FER in the presence of input-dependent noise, $\varsigma = 1$. The best achieving models are same as the $\varsigma = 1$, the only difference is that the FER values are worsened due to additional noise term. As expected, the results show that $FER_{\lambda=0.4} < FER_{\lambda=0.6} < FER_{\lambda=0.8}$ when other parameters are kept constant.

3.3.3. Effect of Block Length, l, on FER

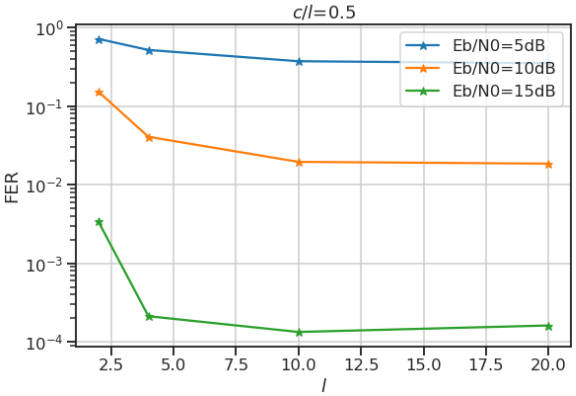
Provided that $\frac{c}{l}$ is kept constant, changing the block length does not alter the dimming level but changes the maximum run length, thus the flicker. The effect of this parameter on error rates, when $\frac{c}{l}$ is constant, are simulated. The longer block sizes increases relaxes the constraints of VLCnet, therefore enables learning better representations. Both for the case without input-dependent noise ($\varsigma = 0$) and with input-dependent noise ($\varsigma = 1$), it is verified that having longer block sizes improves FER performances.

The Figure 3.11a shows the effect of l on FER for the case when no input-dependent noise exists. The labels of the plots explains the parameters of the VLCnet and how much noise they are trained with. These noise values for training are selected as explained in the algorithm in Figure 3.7. As seen in Figure 3.11a, the error rate values for ($l = 10$, $c = 4$) are less than ($l = 5$, $c = 2$). Same results are also valid for the paramater sets of ($l = 10$, $c = 6$) and ($l = 5$, $c = 3$) although the performance difference is less. Figure 3.11b exhibits the same behavior for the case when input-dependent noise exists and $\varsigma = 1$.

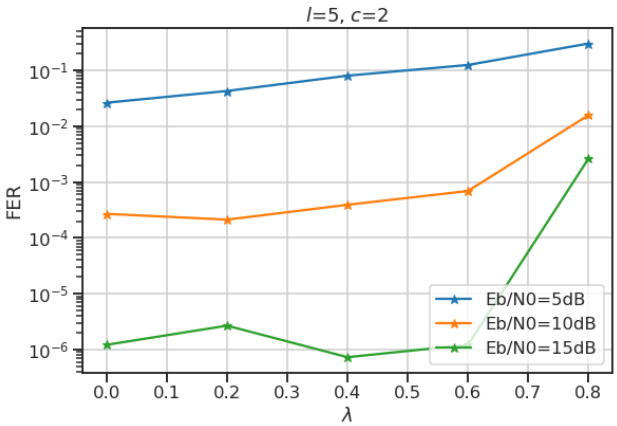
For better visualization, FER curves are plotted against c, l and λ in Figure 3.12 for different $E_b/N0$ values. For $l = 5$, moderate c values 2 or 3 perform better compared to the 1 or 4 because of the higher freedom of the network caused by increased number



(a) FER for c values



(b) FER for l values



(c) FER for λ values

Figure 3.12: Frame Error Rate Curves for different l, c and λ values

of codeword combinations. As l increases when c/l is kept constant, dimming bounds for the model does not change although higher number of codeword combinations is obtained. This relaxation of the constraints by increasing l improves FER accuracy. Lastly, as λ increments, the error performance is worsened due to lowered dynamic range of the FRAU.

3.3.4. Performance Comparison with Other Methods

The error rate performance of proposed VLCnet is compared with Reed Solomon (RS) (11, 15) codes, which are suggested to be used for VLC in the standard [59]. Since short codes and simple to decode codes are suggested for VLC [59], Hamming codes (11, 15) are also added to simulations for comparison. Channel model and

(k, n) parameters of VLCnet are same as described in Section 3.3. OOK is used as modulation for the uncoded transmission, and with Hamming (11, 15) and RS (11, 15) channel codes. The channel taps and symbol energy are normalized in the simulations. Due to the presence of ISI, 5-tap minimum mean-squared error (MMSE) equalizer is added to the receiver for Uncoded, Hamming and RS cases. Without equalization, the error rate performances of simulated conventional methods are confirmed to be much poorer. The effectiveness of the proposed VLCnet in case is no input-dependent noise is shown in Figure 3.13. The labels for the curves explain the parameters of different VLCnet configurations and how much $E_b/N0$ they are trained with in bold. For low SNR region until 10 dB, coding does not enhance the error performance compared to the uncoded case, whereas VLCnet models can provide much better performances. At 15 dB, the RS gives better performance compared to Hamming and uncoded case and many configurations of VLCnet can perform better or close to RS at that SNR value. The only configuration that does not improve FER performance compared to the uncoded case is the one with $\lambda = 0.8$, which is caused by very limited dynamic range as discussed in 3.3.2. The simulation results for the channel including input-dependent noise is illustrated in Figure 3.14. For this case, the performances of both VLCnet and conventional channel code methods are worse because of the additional noise term.

RS and Hamming performs slightly worse than uncoded case for low SNR regions; however, they achieve lower error rate than uncoded at 20 dB. The performance of RS is much better compared to Hamming at that SNR. On the other hand, VLCnet again provides similar or better performance especially when SNR is low. Both for $\varsigma = 0$ and $\varsigma = 1$, the best performing VLCnet configurations are where λ is small or l is big while $\frac{c}{l}$ is kept constant.

Secondly, VLCnet is compared with another channel coding method proposed for VLC [87]. Authors suggest using polar codes concatenated with RLL codes to achieve both error correction and flicker reduction. Compensation symbols are further added to adjust the dimming. FER values for $k = 12$, $n = 20$, $\varsigma = 0$ is simulated

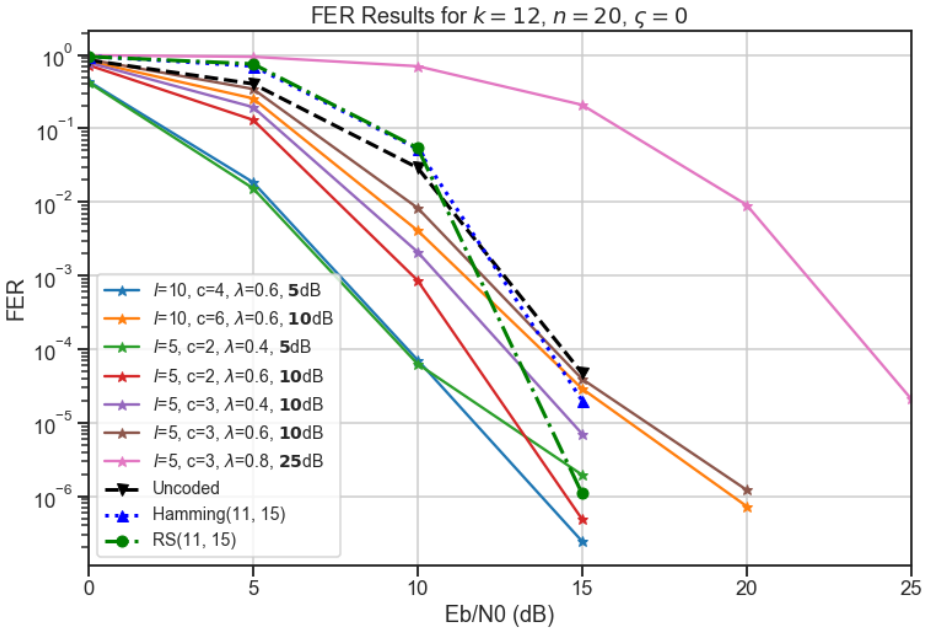


Figure 3.13: FER comparison of VLCnet with channel coding methods, without input-dependent noise

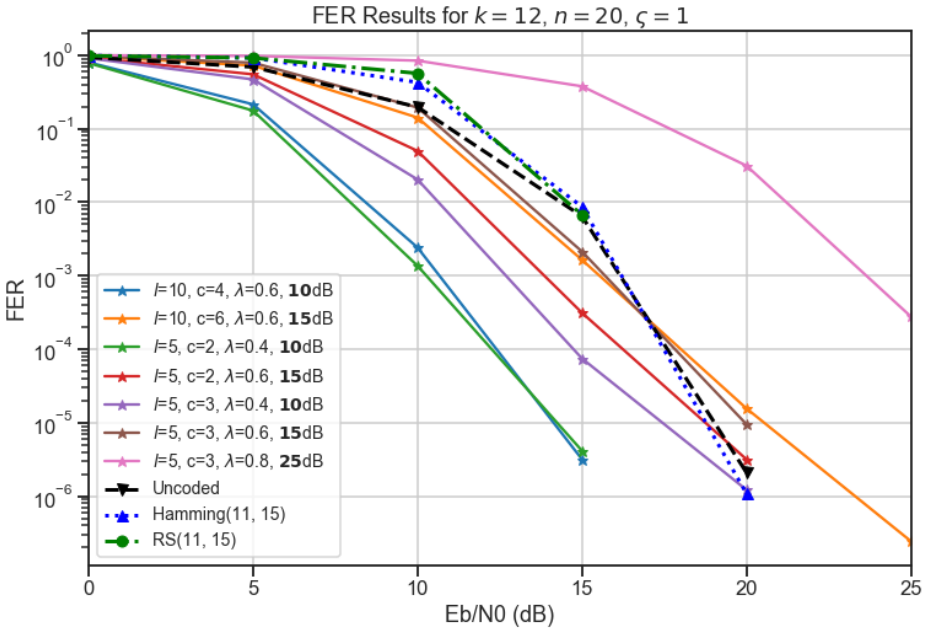


Figure 3.14: FER comparison of VLCnet with channel coding methods, with input-dependent noise

and shown in Figure 3.15. 11-tap MMSE equalizer is added to the receiver of [87] due to the presence of ISI in Figure 3.2. As dimming target approaches to either 0 or 1, baseline model does not success any decent performance because of decreased spectral efficiency by long compensation blocks. For dimming levels 0.4, 0.5, and 0.6; baseline model performs similar and achieves 10^{-5} FER at 15 dB. VLCNet model with $l = 5$, $c = 3$ and $\lambda = 0.8$ performs better than those baseline models. Dimming of that VLCnet model is $0.48 \leq d \leq 0.6$ as it is calculated with (3.9). At the expense of dimming bound precision, a much better performing VLCnet model can be obtained by changing $\lambda = 0.6$ from $\lambda = 0.8$. Dimming of this model has a higher range before training, $0.36 \leq d \leq 0.6$, however performs better thanks to increased dynamic range of the FRAU activations.

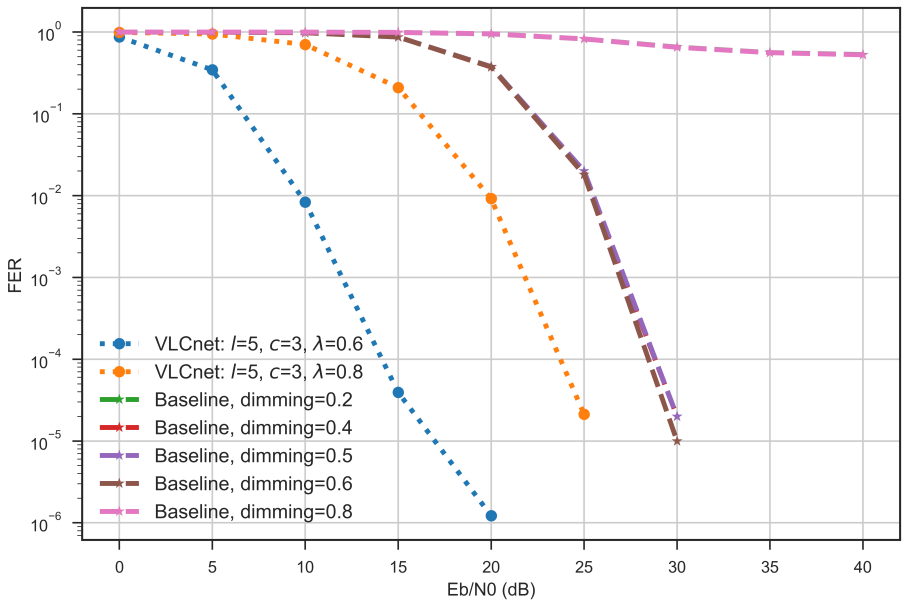


Figure 3.15: FER comparison of VLCnet with [87]

3.4. Practicality Discussion

Besides the performance, the feasibility of the proposed system needs to be assessed well for the practical deployment. Having feasible computational complexity at the inference time is important for the practicality. The encoder side of VLCnet only consists of 1 hidden linear layer and since encoding is a simple mapping for each input message, encoder operation can be realized as codebook based modulation. There-

fore, the complexity of the VLCnet stems from the receiver side. Presented VLCnet receiver in Section 3.3 incorporates 1 convolutional layer and 4 fully connected linear layers. The kernel length and channel count of the convolutional layer, Ω_0 depends on the ISI level of the channel. Therefore, the computational complexity of the convolutional layer of VLCnet decoder is $\mathcal{O}(\Omega_0 n)$. The neuron counts of the following fully connected layers are multiple of n by $\Omega_{1,2,..,K}$, where K is the number of VLCnet decoder side linear layers. Thus, the total complexity of the VLCnet decoder is $\mathcal{O}(\Omega_0 n)+\sum_{l=1}^{K-1}\mathcal{O}(\Omega_l n^2)+\mathcal{O}(\Omega_K nM)$, which can be simplified to $\mathcal{O}(n^2)+\mathcal{O}(nM)$. This complexity is plausible for moderate n and M values considering $\mathcal{O}(n^2M)$ complexity of the maximum likelihood decoders.

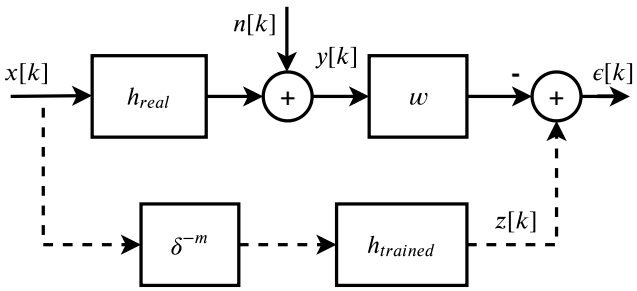


Figure 3.16: Proposed MMSE diagram

Another key issue is the channel information mismatch between the training and operation. As shown in the algorithm in Figure 3.6, the training procedure of VLCnet assumes the channel is known. Utilization of CSI for the optimization of communication modules also exist in some other neural network based communication literature [73,78]. Although this allows model to be tuned and optimized for the most likely case; the trained models can still be used in a different channel by some modifications. Once the real CSI is obtained through pilot symbols, a modified MMSE equalizer can be used before the VLCnet decoder if $\varsigma = 0$. This MMSE equalizer is a finite impulse response (FIR) filter which minimizes the square of the error between the codewords convolved with expected channel and the real channel. Thus this filter does not only equalize the real channel, h_{real}, but also transforms the received codewords such that if they were transmitted through the trained channel, $h_{trained}$. Block diagram of the proposed MMSE equalizer is shown in Figure 3.16. The coefficients of this FIR filter w can be calculated by solving the quadratic function, $\frac{\partial E\{\epsilon[k]\epsilon[k]^*\}}{\partial w} = 0$, which yields

$$
\begin{aligned}
w^H &= h^H_{trained} R_{xx} H^H_{real} (H_{real} R_{xx} H^H_{real} + R_{nn})^{-1} \\
&= h^H_{trained} (R^{-1}_{xx} + H^H_{real} R^{-1}_{nn} H_{real})^{-1} H^H_{real} R^{-1}_{nn}.
\end{aligned}
\tag{3.14}
$$

The effect of channel mismatch on FER is shown in Figure 3.17. A trained model with channel of the scenario4/location6 is tested with scenario4/location1 [80]. Although the model does not work when directly applied to the new channel, using described MMSE equalizer before the decoder gets the FER performance approach to the ideal case where the model is trained and tested with the new channel.

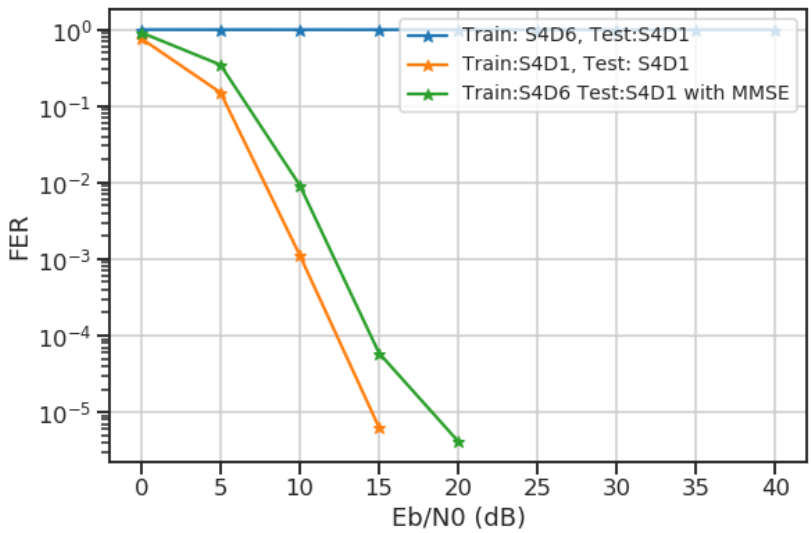


Figure 3.17: FER in channel mismatch

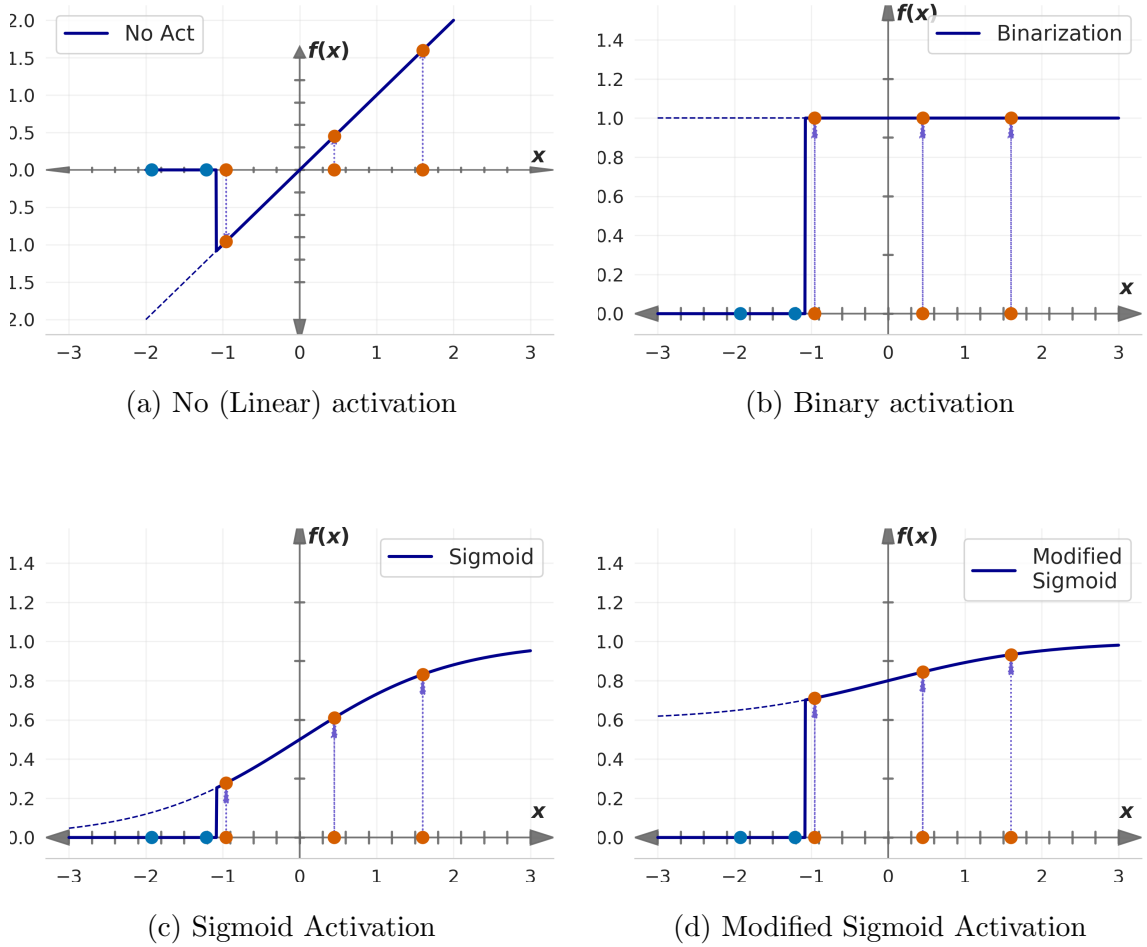
Another method that can work even with input-dependent noise case is to adapt the receiver side of VLCnet to the new CSI. This can be done by resuming learning at the receiver side by calculating losses for the known transmitted sequences. This operation takes much fewer iterations than learning from the scratch. This technique is called transfer learning [79] and widely used in computer science domain to easily modify neural networks to changed conditions or new tasks. Utilizing transfer learning for channel mismatches is also proposed for the channel autoencoders [73].

3.5. FRAU Discussion

To better depict the reason of using modified sigmoid function at the active nodes of FRAU, consider the case where $l = 5$ and $c = 3$. K-sparse autoencoders [75] do not employ any activation function to the top-k active nodes. So, their activation function could be conceived as a linear function. However, having a linear function for active nodes is not possible in VLC as these nodes could be negative, as shown in Figure 3.18a. Directly binarizing based on the selections, as in Figure 3.18b, is not viable since this function is not differentiable. A straightforward solution is to use sigmoid to have differentiability while being similar to a binary function. However, when the active nodes are small as in Figure 3.18c, sigmoid returns values that are close to 0. This situation violates run-length calculations. Therefore, the active nodes should be separate enough from the passive ones. As seen in Figure 3.18d, the modified sigmoid function provides a practical solution to the explained problem.

Flicker mitigation is the main feature of FRAU. As the IEEE 802.15.7 standard [59] and most studies in the literature like [88, 89], we aim to reduce flicker by limiting maximum run length. FRAU limits the maximum run length to $max(2c, 2(l-c))$ value as described in Section 3.2.3. To further illustrate the effectiveness of the proposed method on the flicker, a simulation was performed.

Although flicker definition of Illumination Engineering Association (IES) [90] only considers deterministic pulses (not the codebooks or VLC domain), we found their metric, which is the ratio of low frequency component to high frequency component of pulses, useful. Therefore, we calculated the average energy spectrum of the uncoded case, VLCnet codebook with ($n = 20$, $l = 5$, $c = 2$) and VLCnet codebook with ($n = 20$, $l = 20$, $c = 10$). More flicker, so more powerful low frequency energy component is expected in ($n = 20$, $l = 20$, $c = 10$) since we force FRAU to produce 10 consecutive active and 10 consecutive passive codeword symbols. Likewise, less flickers is expected in ($n = 20$, $l = 5$, $c = 2$) since active and passive symbols change frequently, thus codewords have more powerful higher frequency component.



(a) No (Linear) activation

(b) Binary activation

(c) Sigmoid Activation

(d) Modified Sigmoid Activation

Figure 3.18: Comparison of different activations for a sample set of activated neurons

Figure 3.19 shows average energy spectrum of the codewords in frequency range normalized to the Nyquist frequency. As it is seen, all codebooks have DC components so a powerful first component in their Fourier Transforms. Apart from that DC, ($l = 5$, $c = 2$) code has more powerful high frequency component, while ($l = 20$, $c = 10$) has more powerful low frequency component as expected. If we take the ratios of low half frequency to the high half frequency, we obtain the 1.85 for ($l = 5$, $c = 2$), while this value is 3.88 for the ($l = 20$, $c = 10$) model. Thus, FRAU is an effective tool to mitigate flicker by adjusting l and c parameters.

By fixing l, c and λ parameters of the VLCnet, target dimming bounds can be set as the amplitude of $\boldsymbol{v}$ depends on these parameters. VLCnet encoder parameters, θ_e, can be regarded as random variables mainly caused by the random initialization of

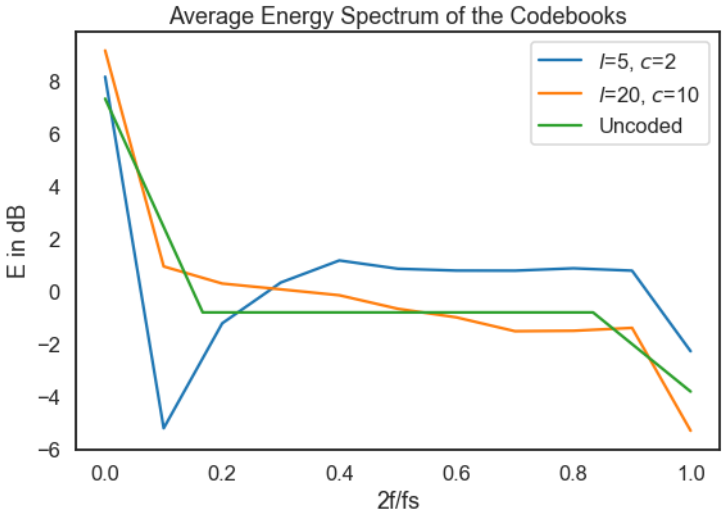


Figure 3.19: Average energy spectrums of the different codebooks

the weights and biases at the beginning of the training. This causes the input of the FRAU, $\boldsymbol{s}$, being sampled from a dependent distribution. Since no assumption can be made about the distribution of $\boldsymbol{s}$, $\mathbb{E}[\boldsymbol{v}]$ cannot be computed in closed form for a given l, c and λ. The dimming bounds for these parameters is given in 3.9. These bounds are utilized as dimming targets before training, however the actual d_{θ_e} is known once the training is completed. As λ approaches to 1, the uncertainty at d_{θ_e} reduces at the expense of decreased output range for active nodes.

A sample histogram of FRAU activations for ($l = 5$, $c = 3$ and $\lambda = 0.6$) is demonstrated in Figure 3.20. As shown, the distribution of the FRAU activation is not a parametric distribution. With the given FRAU parameters, the dimming bounds are calculated as $0.36 \leq d \leq 0.6$ using (3.9). Actual dimming is found to be 0.482 after training.

3.6. Discussion

In order to solve the complex problem of optimizing a VLC system in overall, a deep learning based method, VLCnet is proposed in this chapter. VLCnet addresses several key issues such as decreasing error rates, reducing flicker and controlling dimming with the help of novel activation layer, FRAU. To the best of our knowledge, a

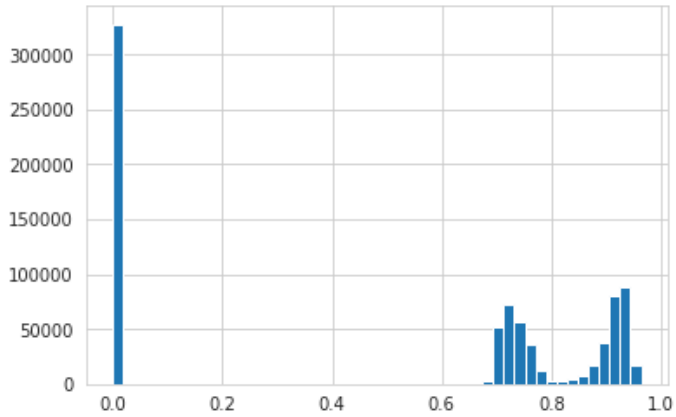


Figure 3.20: Distribution of FRAU activations for ($l = 5$, $c = 3$, $\lambda = 0.6$)

deep learning based flicker mitigating end-to-end system is proposed for the first time in the literature.

First, VLCnet is explained as proposed system model. Then, FRAU is introduced as the unit which would be located at the end of the transmitter. FRAU effectively creates blocks with limiting the number of elements which could be active. The active nodes are applied to a modified sigmoid function while inactive nodes are zeroed out. With this procedure, flicker mitigation is achieved since run length is limited. Furthermore, how VLCnet can be trained effectively using grid search is explained. Through Monte Carlo simulations, the effects of the number of active nodes, minimum level of active nodes and block length on FER are analyzed. An important outcome of the performance evaluations is to show how training with a certain noise level yields decent error rate performance even under altering noise levels.

The performance of our proposed system is examined with different configurations and the results are better than the conventional methods. Depending on the configuration, our proposed VLCnet system generally does an impressive job in decreasing error rate while providing flicker mitigation. Practical considerations are also addressed in our research. VLCnet is able to provide a practical solution as an end-to-end VLC system.

Our work explained in this chapter is published in [91].

4. CONCLUSION

In this thesis, practical and effective channel coding methods for the channels with input-dependent noise are proposed. Many new applications have created a need of applying communications theory to different channels that are less researched compared to the conventional wireless channel. These channels have their own challenges that needs to be addressed with custom designed communication blocks. Directly applying conventional methods Having input-dependent noise is one common property that exits in many of these channels, therefore any practical channel coding algorithm with input-dependent noise could easily find a place in different applications in those channels.

Molecular communications and visible light communications are some examples of the channels that have input dependent noise. Besides this noise type, these channels have their own additional requirements as well. For molecular communications, any proposed method should be energy-efficient and has low complexity as the nano-devices have limited resources in terms of both energy and computation power. For visible light communications, the illumination related requirements namely dimming and flicker reduction need to be satisfied. ISI is a major problem for molecular communications as the diffusion process results in a random delay in molecule transmission time. When high data rates are preferred for indoor applications, VLC also suffers from ISI. In the this thesis work, novel channel coding/decoding methods are proposed addressing all of these requirements.

The main contributions of the thesis can be divided into two parts with respect to the communication channels. In the first part, we propose two different novel methods for molecular communications channel. In the second part, we propose a novel way of end-to-end optimization of communication blocks by addressing the VLC needs.

For molecular communications, we propose using CLWC as channel codes since the low weights are favorable considering energy limitation and presence of ISI. Fur-

thermore, the information of constant weights can be utilized effectively to increase communication quality at the receiver side. In the first part of the thesis, two different decoding methods that take advantage of the constant weight information are proposed. When coupled with our proposed decoders, CLWC is also able to increase communication quality in addition to its aforementioned advantages. First proposed algorithm in this part is the iterative sorting decoder. It is a heuristic yet simpler method in terms of computation compared to the our second method.

Iterative sorting decoder estimates ISI based on its detection in each iteration and renews its selection after removal of estimated ISI from the original received sequence. At each iteration, likelihood of the decoded sequence is calculated and traced, therefore the algorithm halts when the likelihood is not increased at the last iteration. This prevents error propagation. As long as the likelihood is increased , the algorithm iterates until the same sequence is obtained at consecutive iteration or the maximum number of iterations is reached.

As a second method in the first part, a trellis based maximum a posteriori method is proposed for the decoding of CLWC. Together with the past symbols, the current weight status inside constant weight codeword is embed in trellis graph states. By forming a super trellis, which takes both the channel information and coding information into account, the maximum likely sequence can be determined by using Viterbi algorithm. Although this method is computationally expensive compared to the iterative sorting decoder, it provides better error rate performance. The simulation results shows the error rate performance superiority of the proposed methods compared to the alternative molecular communication decoding schemes in the literature.

In the second part, we propose VLCnet, an autoencoder based end-to-end communication system, for VLC. Similar to other autoencoder based communication systems, VLCnet allows joint optimization of both encoder and decoder. The proposed network consists of novel FRAU layer to map encoder side activations to favorable values for illumination requirements. It has a built-in support for run-length-limitation, so for

flicker reduction. In addition, it allows to adjust dimming bounds. The inputs of FRAU are divided into blocks and the neurons in each block compete each other as only certain number of them can be active. The passive nodes are zeroed out and do not be used in backpropagation step. Active nodes are fed into modified sigmoid function to guarantee the run-length limitation. To best of our knowledge, VLCnet is the only method that provides both flicker reduction and dimming capability, which are the fundamental illumination needs. From the practicality point of view, the VLCnet does not need the deployment of many models trained with different noise levels. It has been shown that a model trained with a certain amount of noise works decent for different signal to noise levels. Although VLCnet is trained with a given channel, it is shown that a modified MMSE equalizer can be applied before the decoder side. By comparisons with other methods, we show that VLCnet provides superior FER performance while having moderate complexity.

Both of the methods assume that CSI is partially or fully available at the receiver. This information can easily be acquired via transmitting the pilot sequence, a training sequence which is already known by the receiver.

4.1. Future Work

As future work, our primary goal is to increase the practicality of our proposed methods. Super trellis decoder, which is explained in Section 2.3.4, provides maximum a posteriori sequence estimation with the Viterbi algorithm. Considering the presence of long channel taps, its complexity is not manageable for nanoscale devices as the complexity of Viterbi algorithm increases exponentially with the number of channel taps. There are many researches in the literature decreasing the computation cost of Viterbi algorithm by sacrificing only small amount of performance. Adaptive Viterbi algorithm is one of these methods [92]. Similar method can be designed for our super trellis diagram to decrease the complexity.

VLCnet offers jointly optimized end-to-end solution. Only exception is the extra equalizer that is needed to adapt the decoder side to the changed channel state. However, this part could be embedded to the VLCnet as well. If the channel state information is fed to the receiver side as another input, the whole autoencoder can be trained with different channels. Hence, the decoder is expected to perform equalization as well. Since the whole system will be optimized with the input-dependent noise, the performance is likely to increase compared to using MMSE equalizer.

Lastly, we think that the developed methods for molecular communications channel and VLC can be adapted to terahertz channel. To see their performances alongside the benchmark models for the terahertz band could be interesting.

REFERENCES

1. Farsad, N., H. B. Yilmaz, A. W. Eckford, C. Chae and W. Guo, "A Comprehensive Survey of Recent Advancements in Molecular Communication", *CoRR*, Vol. abs/1410.4258, 2014, `http://arxiv.org/abs/1410.4258`.

2. Hiyama, S., Y. Moritani, T. Suda, R. Egashira, A. Enomoto, M. Moore and T. Nakan, "Molecular communication", *in Proc. 2005 NSTI Nanotechnol. Conf., Anaheim, CA*, pp. 391–394, 2005.

3. Kadloor, S., R. Adve and A. Eckford, "Molecular Communication Using Brownian Motion With Drift", *NanoBioscience, IEEE Transactions on*, Vol. 11, No. 2, pp. 89–99, June 2012.

4. Akyildiz, I. F., F. Brunetti and C. Blázquez, "Nanonetworks: A New Communication Paradigm", *Comput. Netw.*, Vol. 52, No. 12, pp. 2260–2279, Aug 2008.

5. Pierobon, M. and I. Akyildiz, "A physical end-to-end model for molecular communication in nanonetworks", *Selected Areas in Communications, IEEE Journal on*, Vol. 28, No. 4, pp. 602–611, May 2010.

6. Brown, S. P. and R. A. Johnstone, "Cooperation in the dark: signalling and collective action in quorum-sensing bacteria", *Proceedings of The Royal Society B: Biological Sciences*, Vol. 268, pp. 961–965, 2001.

7. Tepekule, B., A. Pusane, H. Yilmaz and T. Tugcu, "Energy efficient ISI mitigation for communication via diffusion", *Communications and Networking (BlackSeaCom), 2014 IEEE International Black Sea Conference on*, pp. 33–37, May 2014.

8. Enomoto, A., M. Moore, T. Nakano, R. Egashira, T. Sud, A. Kayasuga, H. Kojima, H. Sakakibara and K. Oiwa, "A molecular communication system using a network

of cytoskeletal filaments", *2006 NSTI Nanotechnology Conference and Trade Show - NSTI Nanotech 2006 Technical Proceedings*, Vol. 1, pp. 725–728, January 2006.

9. Kilinc, D. and O. Akan, "Receiver Design for Molecular Communication", *Selected Areas in Communications, IEEE Journal on*, Vol. 31, No. 12, pp. 705–714, December 2013.

10. Pathak, P. H., X. Feng, P. Hu and P. Mohapatra, "Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges", *IEEE Communications Surveys Tutorials*, Vol. 17, No. 4, pp. 2047–2077, Fourthquarter 2015.

11. Minh, H. L., Z. Ghassemlooy, D. O'Brien and G. Faulkner, "Indoor Gigabit optical wireless communications: Challenges and possibilities", *Transparent Optical Networks (ICTON), 2010 12th International Conference on*, pp. 1–6, June 2010.

12. Yu, Z., R. Baxley and G. Zhou, "Dynamic range constrained clipping in visible light OFDM systems with brightness control", *Global Communications Conference (GLOBECOM), 2013 IEEE*, pp. 2461–2465, December 2013.

13. Rajagopal, S., R. Roberts and S.-K. Lim, "IEEE 802.15.7 visible light communication: modulation schemes and dimming support", *Communications Magazine, IEEE*, Vol. 50, No. 3, pp. 72–82, March 2012.

14. Akyildiz, I. F., J. M. Jornet and C. Han, "Terahertz Band: Next Frontier for Wireless Communications", *Phys. Commun.*, Vol. 12, pp. 16–32, September 2014, `http://dx.doi.org/10.1016/j.phycom.2014.01.006`.

15. Grover, P., K. Woyach and A. Sahai, "Towards a Communication-Theoretic Understanding of System-Level Power Consumption", *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 8, pp. 1744–1755, 2011.

16. Jornet, J. M. and I. F. Akyildiz, "Low-Weight Channel Coding for Interference Mitigation in Electromagnetic Nanonetworks in the Terahertz Band", *2011 IEEE*

International Conference on Communications (ICC), pp. 1–6, June 2011.

17. Kocaoglu, M. and O. B. Akan, "Minimum Energy Channel Codes for Nanoscale Wireless Communications", *IEEE Transactions on Wireless Communications*, Vol. 12, No. 4, pp. 1492–1500, April 2013.

18. Chi, K., Y. hua Zhu, X. Jiang and X. zhong Tian, "Energy optimal coding for wireless nanosensor networks", *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 998–1002, April 2013.

19. Bai, C., M. S. Leeson and M. D. Higgins, "Minimum energy channel codes for molecular communications", *Electronics Letters*, Vol. 50, No. 23, pp. 1669–1671, 2014.

20. Zainuddin, M. A., E. Dedu and J. Bourgeois, "Low-Weight Code Comparison for Electromagnetic Wireless Nanocommunication", *IEEE Internet of Things Journal*, Vol. 3, No. 1, pp. 38–48, February 2016.

21. Lu, Y., M. D. Higgins and M. S. Leeson, "Comparison of Channel Coding Schemes for Molecular Communications Systems", *IEEE Transactions on Communications*, Vol. 63, No. 11, pp. 3991–4001, November 2015.

22. Nakano, T., T. Suda, M. Moore, R. Egashira, A. Enomoto and K. Arima, "Molecular communication for nanomachines using intercellular calcium signaling", *5th IEEE Conference on Nanotechnology, 2005.*, Vol. 2, pp. 478–481, July 2005.

23. Enomoto, A., M. Moore, T. Nakano, R. Egashira, T. Suda, A. Kayasuga, H. Kojima, H. Sakakibara and K. Oiwa, "A molecular communication system using a network of cytoskeletal filaments", *2006 NSTI Nanotechnology Conference and Trade Show - NSTI Nanotech 2006 Technical Proceedings*, Vol. 1, pp. 725–728, January 2006.

24. Atakan, B., "Optimal Transmission Probability in Binary Molecular Communica-

tion", *IEEE Communications Letters*, Vol. 17, No. 6, pp. 1152–1155, June 2013.

25. Kilinc, D. and O. B. Akan, "Receiver Design for Molecular Communication", *IEEE Journal on Selected Areas in Communications*, Vol. 31, No. 12, pp. 705–714, December 2013.

26. Arjmandi, H., A. Gohari, M. N. Kenari and F. Bateni, "Diffusion-Based Nanonetworking: A New Modulation Technique and Performance Analysis", *IEEE Communications Letters*, Vol. 17, No. 4, pp. 645–648, April 2013.

27. Kuran, M. S., H. B. Yilmaz, T. Tugcu and I. F. Akyildiz, "Modulation Techniques for Communication via Diffusion in Nanonetworks", *2011 IEEE International Conference on Communications (ICC)*, pp. 1–5, June 2011.

28. Tepekule, B., A. E. Pusane, H. B. Yilmaz and T. Tugcu, "Energy efficient ISI mitigation for communication via diffusion", *2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 33–37, May 2014.

29. Yeh, P., K. Chen, Y. Lee, L. Meng, P. Shih, P. Ko, W. Lin and C. Lee, "A new frontier of wireless communication theory: diffusion-based molecular communications", *IEEE Wireless Communications*, Vol. 19, No. 5, pp. 28–35, October 2012.

30. Meng, L., P. Yeh, K. Chen and I. F. Akyildiz, "On Receiver Design for Diffusion-Based Molecular Communication", *IEEE Transactions on Signal Processing*, Vol. 62, No. 22, pp. 6032–6044, November 2014.

31. Noel, A., K. C. Cheung and R. Schober, "Optimal Receiver Design for Diffusive Molecular Communication With Flow and Additive Noise", *IEEE Transactions on NanoBioscience*, Vol. 13, No. 3, pp. 350–362, September 2014.

32. Ntouni, G. D., A. E. Paschos, V. M. Kapinas, G. K. Karagiannidis and L. J. Hadjileontiadis, "Optimal detector design for molecular communication systems

using an improved swarm intelligence algorithm", *IET Micro Nano Letters*, Vol. 13, No. 3, pp. 383–388, 2018.

33. ShahMohammadian, H., G. G. Messier and S. Magierowski, "Optimum receiver for molecule shift keying modulation in diffusion-based molecular communication channels", *Nano Communication Networks*, Vol. 3, No. 3, pp. 183 – 195, 2012, `http://www.sciencedirect.com/science/article/pii/S187877891200035X`.

34. Leeson, M. S. and M. D. Higgins, "Error correction coding for molecular communications", *2012 IEEE International Conference on Communications (ICC)*, pp. 6172–6176, June 2012.

35. Shih, P. J., C. h. Lee and P. C. Yeh, "Channel codes for mitigating intersymbol interference in diffusion-based molecular communications", *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 4228–4232, December 2012.

36. Shih, P., C. Lee, P. Yeh and K. Chen, "Channel Codes for Reliability Enhancement in Molecular Communication", *IEEE Journal on Selected Areas in Communications*, Vol. 31, No. 12, pp. 857–867, December 2013.

37. Haselmayr, W., N. Varshney, A. Taufiq Asyhari, A. Springer and W. Guo, "On the Impact of Transposition Errors in Diffusion-Based Channels", *ArXiv e-prints arXiv:1701.02971*, January 2017.

38. Erin, C. and H. H. Asada, "Energy optimal codes for wireless communications", *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, Vol. 5, pp. 4446–4453, December 1999.

39. Jamali, V., A. Ahmadzadeh, N. Farsad and R. Schober, "SCW codes for optimal CSI-free detection in diffusive molecular communications", *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 3190–3194, June 2017.

40. Tepekule, B., A. E. Pusane, H. B. Yilmaz, C. B. Chae and T. Tugcu, "ISI Mitiga-

tion Techniques in Molecular Communication", *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, Vol. 1, No. 2, pp. 202–216, June 2015.

41. Schulten, K. and I. Kosztin, *Lectures in Theoretical Biophysics*, Vol. 117, p. 39, Department of Physics and Beckman Institute, University of Illinois, 2000.

42. Etzion, T. and A. Vardy, "A new construction for constant weight codes", *2014 International Symposium on Information Theory and its Applications*, pp. 338–342, October 2014.

43. Skachek, V. and K. A. S. Immink, "Constant Weight Codes: An Approach Based on Knuth's Balancing Method", *IEEE Journal on Selected Areas in Communications*, Vol. 32, No. 5, pp. 909–918, May 2014.

44. Brouwer, A. E., J. B. Shearer, N. J. A. Sloane and W. D. Smith, "A new table of constant weight codes", *IEEE Transactions on Information Theory*, Vol. 36, No. 6, pp. 1334–1380, November 1990.

45. Lin, M.-C., "Constant weight codes for correcting symmetric errors and detecting unidirectional errors", *IEEE Transactions on Computers*, Vol. 42, No. 11, pp. 1294–1302, November 1993.

46. Tian, C., V. A. Vaishampayan and N. J. A. Sloane, "A Coding Algorithm for Constant Weight Vectors: A Geometric Approach Based on Dissections", *IEEE Transactions on Information Theory*, Vol. 55, No. 3, pp. 1051–1060, March 2009.

47. Bose and Rao, "Theory of Unidirectional Error Correcting/Detecting Codes", *IEEE Transactions on Computers*, Vol. C-31, No. 6, pp. 521–530, June 1982.

48. Cover, T., "Enumerative source encoding", *IEEE Transactions on Information Theory*, Vol. 19, No. 1, pp. 73–77, January 1973.

49. Troesch, F. and A. Wittneben, "MLSE Post-Detection for ISI Mitigation and Syn-

chronization in UWB Low Complexity Receivers", *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, pp. 2915–2919, April 2007.

50. Forney, G. D., "The Viterbi algorithm", *Proceedings of the IEEE*, Vol. 61, No. 3, pp. 268–278, 1973.

51. Cormen, T. H., C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms (Third Edition)*, pp. 220–222, MIT Press, 2009.

52. Lai, T. W. and D. Wood, "Implicit selection", R. Karlsson and A. Lingas (Editors), *SWAT 88*, pp. 14–23, Springer Berlin Heidelberg, Berlin, Heidelberg, 1988.

53. Ho, M. W., *Maximum Likelihood Sequence Estimation from the Lattice Viewpoint*, Master's Thesis, The Chinese University of Hong Kong, Hong Kong, 1991.

54. Etzion, T., "Optimal doubly constant weight codes", *Journal of Combinatorial Designs*, Vol. 16, No. 2, pp. 137–151, 2008, `https://onlinelibrary.wiley.com/doi/abs/10.1002/jcd.20160`.

55. Ulkar, M. G., A. E. Pusane and T. Tugcu, "Novel decoding methods of constant weight coding for molecular communications", *Nano Communication Networks*, Vol. 19, pp. 157 – 167, 2019, `http://www.sciencedirect.com/science/article/pii/S1878778918300425`.

56. Karunatilaka, D., F. Zafar, V. Kalavally and R. Parthiban, "LED Based Indoor Visible Light Communications: State of the Art", *IEEE Communications Surveys & Tutorials*, Vol. 17, pp. 1–30, September 2015.

57. Jovicic, A., J. Li and T. Richardson, "Visible light communication: opportunities, challenges and the path to market", *IEEE Communications Magazine*, Vol. 51, No. 12, pp. 26–32, December 2013.

58. Ergul, O., E. Dinc and O. B. Akan, "Communicate to illuminate: State-of-the-

art and research challenges for visible light communications", *Physical Communication*, Vol. 17, pp. 72 – 85, 2015, http://www.sciencedirect.com/science/article/pii/S1874490715000385.

59. "IEEE Standard for Local and Metropolitan Area Networks–Part 15.7: Short-Range Wireless Optical Communication Using Visible Light", *IEEE Std 802.15.7-2011*, pp. 1–309, September 2011.

60. Kim, J., K. Lee and H. Park, "Power efficient visible light communication systems under dimming constraint", *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, pp. 1968–1973, September 2012.

61. Vučić, J., C. Kottke, S. Nerreter, K. Habel, A. Búttner, K. Langer and J. W. Walewski, "230 Mbit/s via a wireless visible-light link based on OOK modulation of phosphorescent white LEDs", *2010 Conference on Optical Fiber Communication (OFC/NFOEC)*, pp. 1–3, March 2010.

62. Fujimoto, N. and H. Mochizuki, "477 Mbit/s visible light transmission based on OOK-NRZ modulation using a single commercially available visible LED and a practical LED driver with a pre-emphasis circuit", *2013 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, pp. 1–3, March 2013.

63. Rajagopal, S., R. D. Roberts and S. Lim, "IEEE 802.15.7 visible light communication: modulation schemes and dimming support", *IEEE Communications Magazine*, Vol. 50, No. 3, pp. 72–82, March 2012.

64. Siddique, A. B. and M. Tahir, "Joint error-brightness control coding for LED based VLC link", *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 400–404, April 2014.

65. Lee, K., H. Park and J. R. Barry, "Indoor Channel Characteristics for Visible Light Communications", *IEEE Communications Letters*, Vol. 15, No. 2, pp. 217–219, February 2011.

66. Nan, X., P. Wang, L. Guo, L. Huang and Z. Liu, "A Novel VLC Channel Model Based on Beam Steering Considering the Impact of Obstacle", *IEEE Communications Letters*, Vol. 23, No. 6, pp. 1003–1007, June 2019.

67. Uysal, M., F. Miramirkhani, O. Narmanlioglu, T. Baykas and E. Panayirci, "IEEE 802.15.7r1 Reference Channel Models for Visible Light Communications", *IEEE Communications Magazine*, Vol. 55, No. 1, pp. 212–217, January 2017.

68. Moser, S. M., "Capacity Results of an Optical Intensity Channel With Input-Dependent Gaussian Noise", *IEEE Transactions on Information Theory*, Vol. 58, No. 1, pp. 207–223, January 2012.

69. Zhang, C., P. Patras and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey", *IEEE Communications Surveys Tutorials*, pp. 1–1, 2019.

70. Nachmani, E., Y. Be'ery and D. Burshtein, "Learning to decode linear codes using deep learning", *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 341–346, September 2016.

71. Liang, F., C. Shen and F. Wu, "Exploiting Noise Correlation for Channel Decoding with Convolutional Neural Networks", *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2018.

72. O'Shea, T. and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer", *IEEE Transactions on Cognitive Communications and Networking*, Vol. 3, No. 4, pp. 563–575, December 2017.

73. Dorner, S., S. Cammerer, J. Hoydis and S. t. Brink, "Deep Learning Based Communication Over the Air", *IEEE Journal of Selected Topics in Signal Processing*,

Vol. 12, No. 1, pp. 132–143, February 2018.

74. Lee, H., I. Lee, T. Quek and S. Lee, "Binary signaling design for visible light communication: a deep learning framework", *Optics Express*, Vol. 26, No. 14, pp. 18131–18142, July 2018.

75. Makhzani, A. and B. J. Frey, "k-Sparse Autoencoders", *2nd International Conference on Learning Representations, ICLR*, April 2014, `http://arxiv.org/abs/1312.5663`.

76. Soltani, M., W. Fatnassi, A. Aboutaleb, Z. Rezki, A. Bhuyan and P. Titus, "Autoencoder-Based Optical Wireless Communications Systems", *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, December 2018.

77. Lee, H., S. H. Lee, T. Q. S. Quek and I. Lee, "Deep Learning Framework for Wireless Systems: Applications to Optical Wireless Communications", *IEEE Communications Magazine*, Vol. 57, No. 3, pp. 35–41, March 2019.

78. Lee, H., T. Q. S. Quek and S. H. Lee, "A Deep Learning Approach to Universal Binary Visible Light Communication Transceiver", *IEEE Transactions on Wireless Communications*, Vol. 19, No. 2, pp. 956–969, 2020.

79. Goodfellow, I., Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

80. Uysal, M., T. Baykas, M. Farshad, N. Serafimovski and V. Jungnickel, *TG7r1 Channel Model Document for High-rate PD Communications*, Technical Report IEEE 802.15.7r1/2015/746r1, September 2015, `https://https://mentor.ieee.org/802.15/dcn/15/15-15-0746-01-007a-tg7r1-channel-model-document-for-high-rate-pd-communications.pdf`.

81. Srivastava, R. K., J. Masci, F. Gomez and J. Schmidhuber, "Understanding Locally Competitive Networks", *arXiv e-prints arXiv:1410.1165*, October 2014.

82. Srivastava, R. K., J. Masci, S. Kazerounian, F. Gomez and J. Schmidhuber, "Compete to Compute", *Advances in Neural Information Processing Systems 26*, pp. 2310–2318, Curran Associates, Inc., 2013, `http://papers.nips.cc/paper/5059-compete-to-compute.pdf`.

83. Kingma, D. P. and J. Ba, "Adam: A Method for Stochastic Optimization", *arXiv e-prints arXiv:1412.6980*, December 2014.

84. Bergstra, J. and Y. Bengio, "Random Search for Hyper-Parameter Optimization", *The Journal of Machine Learning Research*, Vol. 13, pp. 281–305, 03 2012.

85. Klein, A., S. Falkner, S. Bartels, P. Hennig and F. Hutter, "Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets", *arXiv e-prints arXiv:1605.07079*, May 2016.

86. Elsken, T., J. Hendrik Metzen and F. Hutter, "Neural Architecture Search: A Survey", *arXiv e-prints arXiv:1808.05377*, August 2018.

87. Wang, H. and S. Kim, "Dimming Control Systems With Polar Codes in Visible Light Communication", *IEEE Photonics Technology Letters*, Vol. 29, No. 19, pp. 1651–1654, October 2017.

88. u. Thummaluri, A. Kumar and L. Natarajan, "Flicker Mitigating High Rate RLL Codes for VLC with Low Complexity Encoding and Decoding", *2018 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, pp. 209–214, December 2018.

89. Han, Y., Y. Kim and B. W. Kim, "Bit-Shuffle Coding for Flicker Mitigation in Visible Light Communication", *IEEE Access*, Vol. 7, pp. 150271–150279, 2019.

90. Billmeyer Jr., F. W., "IES Lighting Handbooks, 1981 Reference Volume and 1981 Applications Volume, John E. Kaufman, Ed. Illuminating Engineering Society of North America, New York, 1981, 500 pp. each. Price: $50.00 each,

$90.00 set", *Color Research & Application*, Vol. 6, No. 4, pp. 253–253, 1981, https://onlinelibrary.wiley.com/doi/abs/10.1002/col.5080060417.

91. Ulkar, M. G., T. Baykas and A. E. Pusane, "VLCnet: Deep Learning Based End-to-End Visible Light Communication System", *Journal of Lightwave Technology*, pp. 1–1, 2020, early Access.

92. Chan, F. and D. Haccoun, "Adaptive Viterbi decoding of convolutional codes over memoryless channels", *IEEE Transactions on Communications*, Vol. 45, No. 11, pp. 1389–1400, 1997.

APPENDIX A: SUPER TRELLIS DECODER PSEUDOCODE

Input: $\boldsymbol{y}$, m, CSI, $maxIter$
Output: $\hat{v}$

$Initialisation: L \leftarrow len(CSI)$, $Paths \leftarrow 000..0$
for $i = 1 : codeWordLength - 1$ **do**
  $tempPaths = [\,]$, $tempCosts = [\,]$
  for $ind = 1 : N_{paths}$ **do**
    $currentPath = Paths[ind, :]$
    $currentState = Paths[ind, end]$
    $currentWeightState = currentState[0]$
    $tempTempPaths = [\,]$, $tempTempCosts = [\,]$
    if $codewordLength - i > m - currentWeightState$ **then**
      $tempState = circshift(currentState, 1, 2)$ {// if bit-0 is possible}
      $tempState[0] = currentWeightState$
      $tempState[1] = 0$
      $tempTempPaths = [currentPath, tempState]$
      $tempTempCosts = Cost(\boldsymbol{y}, CSI, currentState, tempState, costs)$
    end if
    if $currentWeightState < m$ **then**
      $tempState = circshift(currentState, 1, 2)$ {// if bit-1 is possible}
      $tempState[0] = currentWeightState + 1$
      $tempState[1] = 1$
      $tempTempPaths = [tempTempPaths; [currentPath, tempState]]$
      $tempTempCosts = [tempTempCosts; Cost(\boldsymbol{y}, CSI, currentState, tempState, costs)]$
    end if
    $tempPaths = [tempPaths; tempTempPaths]$
    $tempCosts = [tempCosts; tempTempCosts]$
  end for
  $Paths = tempPaths$
  $Costs = tempCosts$
  $inds2remove = findCollidedHighCostIndeces(Paths)$ {// Viterbi Path Elimination}
  if $inds2remove != None$ **then**
    $Paths[inds2remove] = [\,]$
    $Costs[inds2remove] = [\,]$
  end if
end for
$\hat{v} = Backtracing(Paths)$
return $\hat{v}$

Figure A.1: Algorithm for Super Trellis Decoder