NOVEL ERROR-CONTROL CODING TECHNIQUES FOR MOLECULAR COMMUNICATION

by

Ahmet Oğuz Kışlal

B.S., Electronics-Communication Engineering, Istanbul Technical University, 2016

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

Graduate Program in Electrical & Electronics Engineering
Boğaziçi University
2019

ACKNOWLEDGEMENTS

There are no proper words to convey my deep gratitude and respect to my advisor Assoc. Prof. Ali Emre Pusane for the continuous support, motivation, enthusiasm and immense knowledge. I could not have imagined having a better advisor and mentor for my study.

I owe sincere thanks and regards to Prof. Tuna Tuğcu. His insightful comments, passionate participation and encouragement widen my research greatly from various perspectives.

I would like to thank Dr. H. Birkan Yılmaz for his invaluable ideas and contributions throughout my M.Sc. study. I have never met a person who has such a great desire for research. His desire was a great example for me and it always will be.

It is a pleasure to thank my friends from WCL and NRG, for the wonderful times we shared. In addition, I would like to express my gratitude to Can Tok for his endless support. This journey was memorable and interesting thanks to you.

Finally, my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support. They selflessly encouraged me to explore new directions in life and seek my own destiny. This journey would not have been possible if not for them, and I dedicate this milestone to them.

This thesis was partially supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under grant number 116E916.

ABSTRACT

NOVEL ERROR-CONTROL CODING TECHNIQUES FOR MOLECULAR COMMUNICATION

In molecular communication via diffusion, information molecules diffusing in the environment are subject to Brownian motion. Due to probabilistic propagation, the arrival of molecules at the receiver is spread in time, leading to the reception of some molecules belonging to the previous symbol(s) during the upcoming symbol duration. Known as inter-symbol interference (ISI), this problem has been extensively studied in the literature by applying a large spectrum of techniques, mostly inspired by approaches in the wireless communication domain including channel coding techniques. Unfortunately, many known channel codes do not perform well in the molecular communications domain, since the diffusion channel features a significant memory component. In this thesis, novel methods for channel coding by incorporating the effect of ISI in the design of the channel codes for the molecular diffusion channel are proposed first. The results show that the proposed methods provide significant improvements in performance in terms of codeword error rate. Second, a novel channel coding method is proposed for molecular communication. The design of the proposed channel code takes into account the capability of a nano-device and the characteristics of the molecular communication channel. Simulation results show that the proposed method provides a significant improvement in terms of bit error rate. Moreover, a proof-of-concept implementation of the proposed coding scheme has been done on a macro-scale testbed. The reliability of the communication link is shown to be significantly increased.

ÖZET

MOLEKÜLER İLETİŞİM İÇİN HATA DÜZELTEN KODLAMA TEKNİKLERİ

Difüzyonla moleküler iletişimde, bilgi taşıyan moleküller Brownian hareketine uygun olarak ortamda difüzyon ile hareket eder. Moleküllerin alıcıya ulaşması, hareketlerinin rastgeleliğinden dolayı, zamana yayılmaktadır. Bu durum bir önceki simgeye ait moleküllerin, gönderilmek istenen moleküller ile karışmasına sebep olmaktadır. Simgeler arası girişim (SAG) olarak da bilinen bu fenomen literatürde yoğun bir ilgi ile çalışılmış ve farklı bakış açılarına sahip bir çok çözüm üretilmiştir. Bu çözümlerin birçoğu olağan kablosuz iletişimde kullanılan tekniklerden esinlenilmiştir. Bilinen kanal kodlama teknikleri, kanal hafızasının çok yüksek olmasından dolayı, istenen başarımı sağlayamamaktadır. Bu tezde ilk olarak, kanalın SAG etkisini göz önünde bulundurularak, kanal kod tasarımı yapmak için yeni yöntemler önerilmiştir. Başarım sonuçları göstermiştir ki önerilen yöntemler hata başarımında önemli iyileştirmeler sağlamaktadır. İkinci olarak moleküler iletişim için yeni bir kanal kodlama ailesi önerilmiştir. Bu yeni kanal kodlama ailesi bir nano-makinenin yapabilecekleri göz önünde bulunarak geliştirilmiş olup, moleküler iletişim kanalının doğasını göz önünde bulundurarak başarım artışı sağlanması hedeflenmiştir. Çalışmanın genel kavramını doğrulamak amacıyla, makro skaladaki bir deney seti üzerinde bu kodlama ailesi gerçeklenmiştir. Sonuçlardan gözlemlendiği üzere iletişim ağının kalitesi ve güvenilirliği önemli ölçüde artmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	A scaling factor that is used to determine threshold
a_an	Analytically calculated scaling factor
B	Binomial distribution
$\mathbf{C}$	Codeword transition probability matrix
C_i	A codeword
$C_{i,j}$	The probability of receiving j-th sequence when i-th codeword is transmitted
CER_i	The error probability when i-th codeword is transmitted
CL_i	Coset leader indicatior
CM_{i_1,i_2}	Coset member indicator
d_H	Hamming distance
d_0^i	Minimum number of received molecules in i-th codeword
d_1	Maximum number of received molecules in i-th codeword
D	Diffusion coefficient
$\mathbf{D}$	Symmetric probability transition matrix
$F_{hit}(t)$	Probability of a molecule's absorption by receiver until t
I	Channel Memory
K	A scaling constant
M	Number of molecule released from transmitter for a transmission of bit-1
$\mathcal{N}$	Normal distribution
$N_{i,k}^{Rx}$	The number of molecules that are released at the k-th symbol slot and received at i-th symbol slot
N_i^{Rx}	The received number of molecules at the end of the i-th symbol slot
$N_{i,j}$	The predicted decoding region indicator
p_k	Channel coefficients
p_μ	Mutation probability
$\mathbf{P}$	Probability transition matrix
$P_{e_{l,i}}$	Probability of error for the l-th bit of the i-th binary sequence

$P_{i,j}$	Probability of the receiving w_j when w_i is transmitted
P_0	Transmission probability of bit-0
P_1	Transmission probability of bit-1
$PoCT$	Probability of correct transmission
Q	Tail distribution function of the standard normal distribution
r_0	Distance between the transmitter and the receiver
r_r	Radius of the receiver
$S_{i,y}$	The received number of molecules that were transmitted previously
t_s	Symbol duration
w_j	Received vector

σ_n^2	Additive white gaussian noise variance
λ	A pre-determined threshold for detection
$\mu_{l,i}$	Mean of the number of received molecules for the l-th bit of the i-th binary sequence
$\sigma_{l,i}$	Variance of the number of received molecules for the l-th bit of the i-th binary sequence
γ	A threshold to determine approximate decoding region
τ^i	The threshold for the i-th codeword
τ_{opt}^i	The optimum threshold for the i-th codeword

LIST OF ACRONYMS/ABBREVIATIONS

AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhure-Hocquenghem
BCSK	Binary Concentration Shift Keying
BER	Bit Error Rate
CER	Codeword Error Rate
CRCTG	Crossover Resistant Coding with Time Gap
CSK	Concentration Shift Keying
C-RM	Cyclic Reed-Muller
EM	Electromagnetic
EG-LDPC	Euclidean geometry low density parity check
GA	Genetic Algorithm
IP	Integer Programming
ISI	Inter-symbol Interference
MC	Molecular Communication
MCvD	Molecular Communication via Diffusion
MIP	Mixed Integer Programming
MM	Messenger Molecule
MoCo	Molecular Coding
MSK	Molecule Shift Keying
RS	Reed-Solomon
SOCC	Self-orthogonal convolutional code

1. INTRODUCTION

For most nanoscale applications, it is not a viable option to communicate via electromagnetic (EM) signal due to challenges including bio-compatibility, power, and possible health hazards. Nonetheless, molecular communications provides many desirable features, including bio-compatibility and nano-scale implementation. Nature uses a similar scheme and over billions of years has perfected it. Transferring information via molecules can be done through a variety of methods. Some of these are communication via micro-tubules [1], bacteria [2], calcium ions [3], or diffusion [4, 5]. Due to its low energy consumption and simplicity, molecular communication via diffusion (MCvD) is one of the most preferred methods for sending data from one nano-machine to another one [4]. Different features of the molecules can be used to encode the information. The most common features used to convey information are the type and the concentration of the molecules. [4, 6]. It is clear that by using different types of molecules, the one have two (or more depending on how many types of molecules used) distinct channels. In other words, when different types of molecules are available, building a viable communication link is easier for MCvD.

Four fundamental steps define an MCvD system. These are encoding of the information, emission and propagation, detection, and decoding with respect to the occurrence order. Encoding, as its name indicates, is the process of encoding the information ta a physical feature of molecules. Commonly used features are the number and the types of molecules. Emission and propagation is the process in which the molecules are released in an generally fluidic environment (which stands for emission) and diffuse in it (which stands for the propagation). At the receiver side, the molecules that reach the receiver are detected by either absorbing or simply sensing. Depending on the feature used in the encoding step, the detection could differ. For example, when number of molecule is used as the physical encoding feature, counting the molecules could be enough. While types of molecules are also used, detecting which molecule has arrived to the receiver gain a significant importance as well. Finally, the detected signal (i.e. number of absorbed molecules) should be decoded accordingly to finalize transmission.

In MCvD, there are two main challenges that are focused on in the literature. Probabilistic movement of the molecules is the first problem that should be handled. In classical communications, while the noise is probabilistic, the channel itself and the transmission can be modelled with a deterministic model in general. On the other hand, in MCvD, even when noise does not exist, the received molecules cannot be modelled with a deterministic model. Of course it is not solely random, but still the deterministic information is limited. For example, when number of molecules is used as the encoding feature of molecules, the receiver has to have a pre-determined threshold, so that it can decode the transmitted information. This pre-determined threshold is commonly selected by assuming the transmitted sequence is known at the receiver, and the receiver calculates the threshold that minimizes the bit error rate. Obviously, assumption of the receiver having the knowledge of transmitted sequence is not a practical one. Such an assumption is indeed viable when pilot symbols are transmitted. However, this means that the whole communication is redundant in the first place. A more practical approach is sending a long sequence of pilot symbols, determine the threshold from this sequence, and use it for the subsequent transmissions. This is clearly a better approach yet it still fails to resist the variations over channel parameters. To follow the variations, one may send pilot symbols with a period but naturally this is a waste of rate (In classical communication, this phenomen is also expressed with a waste of bandwidth. Since bandwidth is not defined for MCvD, we use the word rate). One of the important contributions of this thesis is that a channel coding technique, and a detection strategy are introduced such that pre-determined threshold is not needed.

Second challenge in MCvD systems is the power constraint. A nano machine, due to its naturally small size, is very limited in many ways. Power is one of them and a practical method should consider this constraint. There are two sides of power consumption: the power expanded to synthesize the messenger molecules, and the power consumption for encoding/decoding, detection and many other operations for communication. As a result, synthesizing fewer messenger molecules (MMs) corresponds to transmitting less power, which degrades the viability of communication link. Hence what designers focus on in general is the second part. In this thesis, the proposed methods consider this limitation. In particular, a detection algorithm that is simple enough to be implemented on a nano-scale device is introduced.

In its most basic form the MCvD, the transmitter releases molecules into the environment for each symbol. When these released molecules enter the vicinity of the receptors, they are absorbed by the receiver. MMs have a probabilistic movement pattern that is subject to the Brownian motion. Hence, some MMs take a more direct path towards the receiver and are absorbed within the communication interval while others follow a longer path and are absorbed later, which is another important challenge of MCvD. This causes inter-symbol interference (ISI), and if handled improperly, it can easily cause incorrect symbol detection. ISI can be reduced by increasing the symbol duration, yet this also decreases the data rate. Since the data rate is already lower compared to other form of communication like classical electromagnetic or visible light communication, this is not a desired solution. The introduced novel methods and schemes are developed to combat ISI and counter its effect. Unfortunately, getting rid of the ISI completely is almost impossible, yet the introduced schemes have a desirable performance even when the channel conditions are poor.

For a communication system, there is inevitably going to be some signal distortion that leads to incorrect detection. Increasing the power or decreasing the data rate are some of the techniques to improve the reliability of the communication. From a system designer's perspective, however, neither of these are desirable. By adding redundancy to the transmitted information, channel coding improves a communication link's reliability, but it also decreases the data rate. Redundant information is utilized at the receiver to recover the errors of the distorted signal. Researchers have proposed many different channel codes for numerous applications [7]. However, their effectiveness is significantly constrained by the heavy ISI effect. In classical communication, this problem is solved via equalizers that are aimed to neutralize the effect of the channel. Unfortunately, designing an equalizer is not an easy task for MCvD since the channel is not linear as we are used to from the classical communication channel. Hence, what is needed is a new perspective.

In the literature, both new channel codes are designed and classic techniques have been tried to improve the performance of molecular communication (MC). Error-detection techniques are considered in [8]. The error performance of the system and how it is affected by rate are also shown. The authors in [9] studied coding gain of Hamming codes, obtaining the optimal coding rate. The authors also analyzed energy consumption at the encoder and the decoder. The authors in [10] considered a more complicated coding scheme Reed-Solomon (RS) codes. The authors showed that RS codes could achieve higher coding gains, though implementing them is more complex than that of Hamming codes. In [11] RS codes were also studied for a large-scale MC system. As expected, the bit error rate (BER) performance was better than the uncoded case. Constructing zero-error codes for an MC system scenario with one type of molecule is studied in [12]. The authors of [13] derive an upper bound for zero-error capacity, and introduce a new technique to use zero-error codes for MC systems which more than one molecule type is available. Minimum energy channel (MEC) codes were considered for MC in [14]. The authors showed that a reduction in average energy consumption for per bit and an improvement in the BER performance over the Hamming codes. In [15], Euclidean geometry low-density parity check (EG-LDPC) and cyclic Reed-Muller (C-RM) codes are considered in terms of BER performance and energy consumption. The authors in [16, 17] introduced ISI-free codes, which aim to completely eliminate the ISI between codewords completely with a low decoding complexity. At first glance, the idea seems very useful; upon close inspection however, the ISI-free codes are for cases where the data rate is very low. Crossover-resistant coding with time gap was introduced for MCvD in [18], yet it is also considered for a very low data rate. The authors in [19] compared BER performance of ISI-free codes, MoCo codes, RM codes, and distinct Hamming codes for mobile robots equipped with molecular communication transceivers that release and detect alcohol (or any similar chemical) molecules. The authors in [20] considered self-orthogonal convolutional codes (SOCC) for MC, showing their coding gain is higher than the Hamming codes. The authors in [21] showed that Hamming distance is not the optimal metric for MCvD and introduce a new distance metric called molecular coding distance (MoCo). The authors also showed that their coding gain can be increased by re-defining according to this distance metric, the decoding region of Hamming codes. This is a significant study since it basically shows that classical coding techniques are not good choices for MCvD. The drawback of this study is that a simple system model with very low transmission power and very long symbol duration is assumed, which is not considered as a practical approach today.

In this thesis, we provide a technique to design effective channel codes. First, a new distance-like metric (inspired by [21]) is introduced for realistic MC scenario. After that, we derive the average error probability for a given coding strategy. With these derivations and founding, new ways to design efficient codebooks are introduced. To be more specific, a heuristic solution, a genetic algorithm, and optimization methods are provided to have a better channel code. Finally, codebook design has been done by modeling the problem with a linear equation set and solving it with a mixed integer programming (MIP) solver. A MIP solver guarantees an optimum solution when enough time and source are given. In other words, optimum codebooks are designed with this method.

Second, we design a new channel code family and a detection algorithm, that is specifically tailored for the introduced channel code, by considering the characteristics of the molecular communication channel. Compared to the other code families in the literature, our proposed scheme has higher coding gain, even for high data rates. Furthermore, our code family is applicable for all different kinds of diffusive molecular communication channels without the requirement of the knowledge of the channel state information. This property is an important asset, as it leads to work for time-varying channels, a realm not extensively considered in the literature. Moreover, the scheme is simple enough to be implemented on a nano-machine. To show both its simple nature and independence from channel, this work is implemented on a macro-scale testbed.

The rest of this thesis is organized as follows: system model of MCvD is discussed in Chapter 2. The ISI-aware channel coding design methods are introduced in Chapter 3. In Chapter 4, ISI-minimizing code family is introduced and analyzed. Finally, Chapter 5 concludes the thesis.

2. SYSTEM MODEL FOR MOLECULAR COMMUNICATION via DIFFUSION

2.1. MCvD Channel Model

In MCvD, a channel with a point transmitter and a fully absorbing spherical receiver are assumed to be fixed in a three-dimensional fluidic environment and, as shown in Fig. 2.1, to be communicating with each other via MCvD.

A function depends on time can model every molecule's displacement, independently from the other molecules' positions. Moreover, molecule movement in each dimension may be shown independently, using normal distribution. The change in a molecule's position in Δt seconds can be modelled as

$$\begin{aligned} x(t+\Delta t) &= x(t) + \Delta x \\ y(t+\Delta t) &= y(t) + \Delta y \\ z(t+\Delta t) &= z(t) + \Delta z \end{aligned} \tag{2.1}$$

where Δx, Δy, and Δz are the position change at x, y, and z dimensions, respectively, which are normally distributed with mean 0 and variance $2D\Delta t$.

One of the most basic forms of MCvD is given in Fig. 2.1. Here, the transmitter releases a number of MMs at the beginning of a symbol duration t_s and waits until the next transmission. Please note that the molecules diffuse randomly while they are following a movement pattern according to Brownian motion [22]. This operation is repeated for every symbol with a period of symbol duration at the transmitter side. At the receiver, MMs are absorbed and counted along the symbol durations. At the end of the symbol duration, the counter is reset and the receiver restarts counting. The

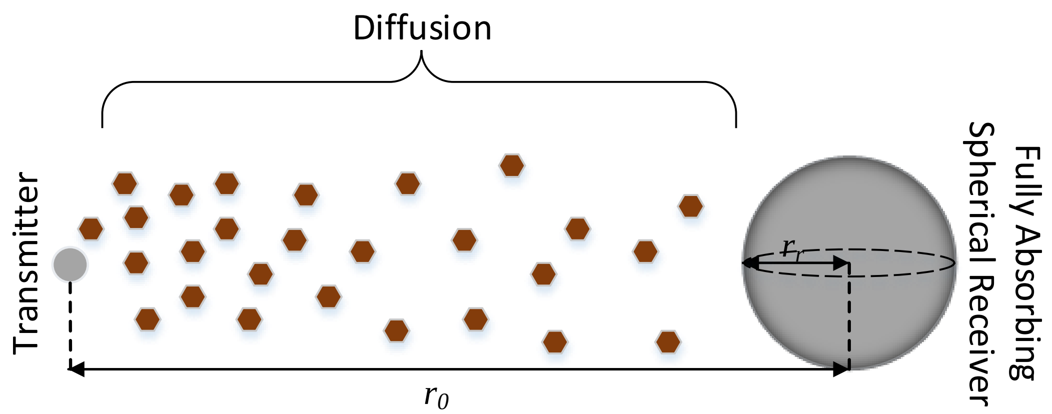

Figure 2.1. MCvD system model.

absorption probability of a released MM until time t is

$$F_{hit}(t) = \frac{r_r}{r_0} \text{erfc}\left(\frac{r_0 - r_r}{\sqrt{4Dt}}\right) \tag{2.2}$$

where D is the diffusion coefficient, r_0 is the distance between the point transmitter and the center of the receiver, r_r is the radius of the receiver, and erfc$(\cdot)$ is the complementary error function [23]. Assume that the time is divided into evenly placed slots, and each slot's period is considered as symbol slot. After a molecule is released, the probability of absorption at a symbol slot is defined as channel coefficients. These are given as

$$p_k = F_{hit}(kt_s) - F_{hit}((k-1)t_s) \quad k = 1, 2, ...I \tag{2.3}$$

where I is the channel memory. The channel coefficients for $t_s = 200\,\text{ms}$ is given in Fig. 2.2. The probability for a molecule to be absorbed in the intended symbol slot is shown with p_1 in (2.3), while the remaining channel coefficients correspond to the probability that the released molecules are absorbed after the symbol slot in which they were released and were intended for. Following the assumption that is typically made in the literature, we assume that the channel coefficients are in descending order (i.e. $p_1 > p_2 ... > p_I$). For the transmission of a single symbol, assuming M molecules are released from the transmitter, molecules that are absorbed in the intended time

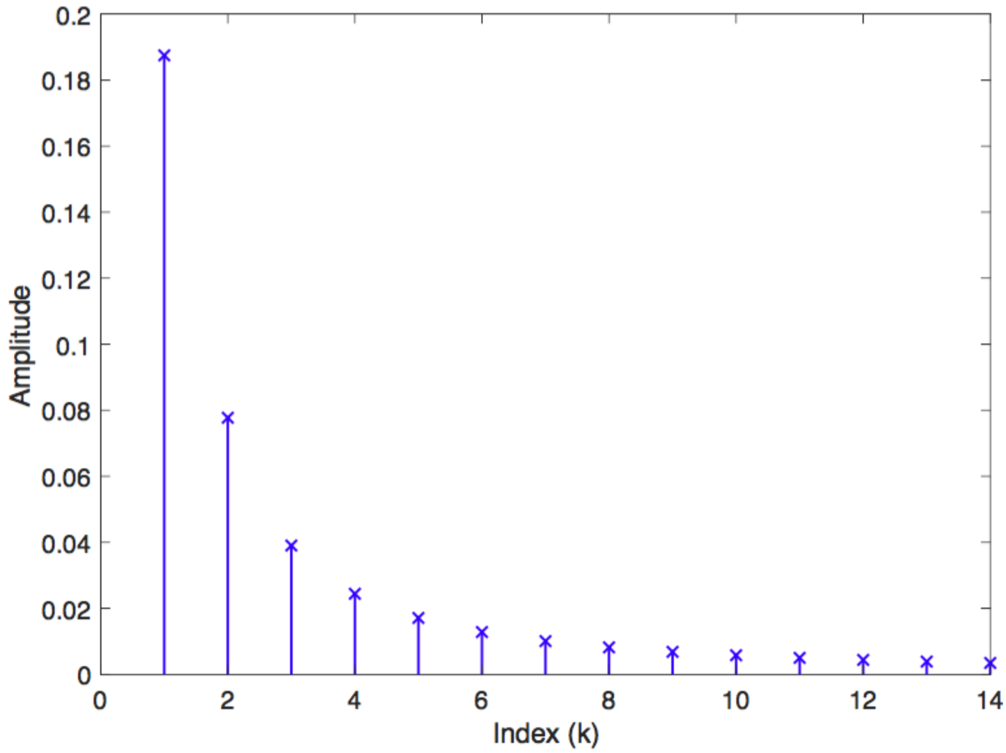

Figure 2.2. Channel coefficients for $t_s = 200\,\text{ms}$.

slot is shown as

$$N_1^{Rx} \sim B(M, p_1) \tag{2.4}$$

where $B(n, p)$ is the binomial distribution with n and p being the trial number and success probability, respectively. In a realistic communication scenario, numerous symbols should be transmitted consecutively. For this case, let $N_{i,k}^{Rx}$ denote the number of molecules that are released at the k-th symbol slot and received at i-th symbol slot, which is distributed according to a binomial distribution. $B(M_k, p_{i-k+1})$, where M_k is the number of released molecules at the k-th symbol slot. The received number of molecules at the end of the i-th symbol slot is given as

$$N_i^{Rx} = \sum_{k=1}^{i} N_{i,k}^{Rx}. \tag{2.5}$$

For $M > 50$, $N_{i,k}^{Rx}$ can be approximated by a normal distribution [24] and (2.5) can be restated as

$$N_i^{Rx} \sim \mathcal{N}\left(\sum_{k=1}^{i} M_k p_{i-k+1}, \sum_{k=1}^{i} M_k p_{i-k+1}\left(1 - p_{i-k+1}\right) + \sigma_n^2\right) \tag{2.6}$$

where σ_n^2 is the variance of the distribution of the counting (or environmental) noise, which is assumed to be additive white Gaussian noise (AWGN).

Equation (2.6) shows that the number of received molecules in a symbol slot is associated with 2 parameters: the number of released molecules M_k and channel coefficients p_i. M_k is similar to the average power in classical communication and as the average power increases, the reliability of communication also increases. Channel coefficients are expressed in a closed form in (2.3). Since the transmitter, the receiver, and the diffusive fluidic environment is fixed. Equations (2.3) and (2.2) show that the only parameter that affects the channel coefficients is the symbol duration. As t_s increases, p_1 also increases, which means ISI decreases, as also does the date rate. In other words, for a desired or arbitrarily low error rate, the lowest symbol duration (fastest data rate) is what a designer wants to obtain. A similar trade-off is also exist in classical communication. This optimization problem has been solved by maximizing the channel capacity or outage capacity depending on the design. For MCvD, the channel capacity is discussed for memorryless channels ($I = 1$) in [25, 26]. An expression of the mutual information is given in [27] yet authors failed to analyze the effect of the receiving structure. In [28, 29], authors express channel capacity with a 1-D environment assumption. Unfortunately, none of these are useful for the case analyzed in this thesis. A promising study on calculating the achievable rate in MCvD is proposed in [30]. However, the calculation of achievable rate is not practical. In a nutshell, the expression of the channel capacity is an open problem and without it, it is hard to determine the optimal symbol duration. In this study, the selection of symbol duration is not optimized; but instead, all results are presented for a low and high symbol durations to show their superiority compared to other methods.

2.2. Modulation Schemes for MCvD

For MCvD, different aspects of molecules can be used to modulate the information. There are two popular and effective types of modulation: Molecule Shift Keying (MSK) and Concentration Shift Keying (CSK). The other types of modulations are in general extension of these two.

One of the most basic and intuitive types of modulations in MCvD is using different type of molecules for different symbols. This type of modulation is called molecule shift keying [31]. For example, in binary MSK, while A-type of molecules are emitted for the transmission of bit-0, B-type of molecules are used for bit-1. In between each symbol interval, the receiver counts both types of molecules. If A type of molecule is higher, the data is detected as bit-0 otherwise it is detected as bit-1.

The other simple scheme is using the quantity of the molecules, a type of modulation known as concentration shift keying. For binary CSK (BCSK), when bit-0 is sent bt the transmitter, the transmitter emits no molecule at. On the other hand, M molecules are released to represent bit-1. At the receiver, if the number of the absorbed molecules in a symbol slot is lower than the pre-determined threshold λ, bit-0 is detected. If the absorbed molecules are higher than λ, the receiver detects a bit-1.

In this work, we mainly use BCSK; this is for simplicity and because it is widely used in the literature [32–34]. Understanding of the concept of different molecule types will be useful ın Chapter 4.

3. ISI-AWARE CHANNEL CODE DESIGN

3.1. Introduction to Channel Coding

Channel coding (or error-control coding) techniques improve the reliability of the overall data transmission (or communication in short). Noise is a part of the communication system that can almost never be avoided. Due to the noise and many different channel effects, it is expected that the transmitted data may have been received erroneously. While error detection codes are used to detect these errors, error correction codes are used to reconstruct the originally transmitted data in the receiver. There are two reasons for preferring error-detection codes over error-correction codes: the complexity issue and the error-detection and correction capacity of a code [35, 36]. The complexity issue is obvious; while detecting the error is the main objective of error-detection codes, error-correction codes should first detect it and then correct it. Hence, their complexity is higher. The second reason is that the error detection capability of code is lower bounded by the error correction capacity. For example, (15,11) Bose–Chaudhuri–Hocquenghem (BCH) codes can detect up to two bits of errors while they can only correct when the received sequence has one bit error [37].

k-bit data sequences, by adding redundant bits, can be extended to n-bit binary sequences which are referred as codewords are the basis of channel codes. The codewords, which are shown $C_i, i = 1, 2, ...2^k$, are sent by the transmitter with a selected modulation type. In a communication scenario, a distortion in the channel is inevitable and the received bit sequence may not be the same as the transmitted one (the transmitted codeword). The received sequence, which is also referred to as received vector and shown by w_j, $j = 1, 2, ..., 2^n$ may or may not be same with any other codeword in the codebook. Hence, the receiver decodes the received vector by mapping the receiving sequence to the most probable codeword in the codebook. Please note that the sequence is still n bit. This should also be mapped back to its k bit form to finalize the decoding operation. An example is given in Fig. 3.2.1.

For two binary sequences, the Hamming distance between these sequences is defined as the number of bits that are not equal to each other. This distance metric is a useful tool when designing channel codes for classical communication. For instance, $(7,4)$ Hamming code, which means that it has a block length $n = 7$ and message length $k = 4$, guarantees the minimum Hamming distance of $d_H = 3$ from one codeword to another. The minimum Hamming distance is important because it shows that when $\lfloor (d_{Hmin} - 1)/2 \rfloor = 1$ bit is wrongfully detected, the received sequence can be successfully mapped back to the transmitted codeword. When classic additive white Gaussian noise (AWGN) channel is used for communication, the bit error performance has a direct dependence with minimum Hamming distance. In an AWGN channel, detection errors are not dependent on the transmitted symbols. Hence, the Hamming distance is the optimal metric for decoding purposes and effectively used. For an MC scenario, this is not correct as the errors caused mostly by the previously transmitted bits. As shown in [18], the Hamming distance is far from optimal and not a decent choice to design channel codes for MC. Moreover, in [38], some classical coding structures have been used in MC. The results show that these coding techniques are not preferable for MC when high data rates are needed.

3.2. Probability Transition Matrix and Decoding Region

3.2.1. Code Distance in Molecular Communication

A code with n block length and k message length has 2^k codewords with n bits. These codewords are selected in between 2^n sequences. To put in different way, Designing a codebook is the selection of 2^k codewords, in between 2^n possible (candidate) binary sequences. The probability of a sequence to be detected as itself or another one is the main concept to design a channel code. For the transmitted sequence w_i and the received sequence w_j, this probability of detection can be shown as $P_{i,j} = P(w_j|w_i)$, where $i = 1, 2, ..., 2^n$ and $j = 1, 2, ..., 2^n$. The transition probabilities can be expressed

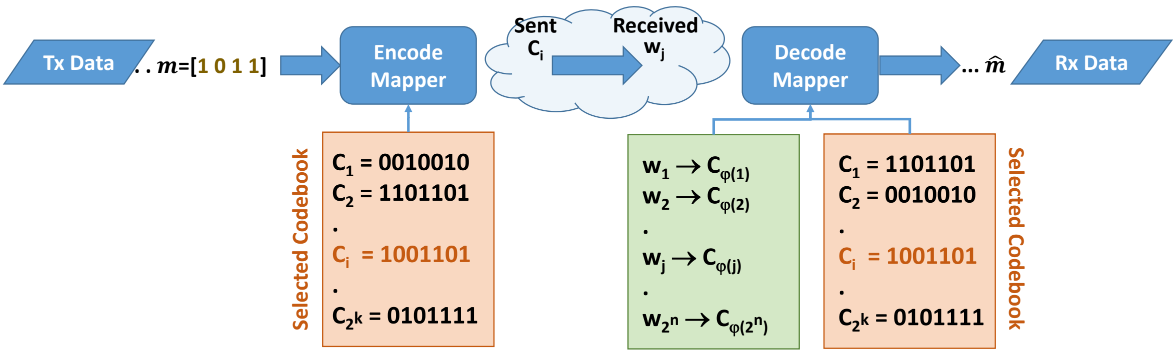

Figure 3.1. The structure of channel coding for a scenario with (7, 4)-coding. The transmitter wants to send the message m, which is four bits in the given example. The encoder maps m to C_i, which is one of the 2^k (i.e., 16 for the example scenario) codewords of length n (i.e., 7 for the example scenario). Then, 7-bits long W_j is received and mapped to one of the codewords in the selected codebook. Note that the indexing function $\varphi(\cdot)$ may map some of the W_j's to the same codeword. In the example scenario, if $\varphi(j) = i$ then the decoded message ($\hat{m}$) should be same as the sent message m.

in matrix form as

$$\mathbf{P} = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,2^n} \\ P_{2,1} & P_{2,2} & \dots & P_{2,2^n} \\ \vdots & \vdots & & \vdots \\ P_{2^n,1} & P_{2^n,2} & \dots & P_{2^n,2^n} \end{bmatrix}. \tag{3.1}$$

Calculating the probability transition matrix $\mathbf{P}$ is not a trivial problem. One possibility is by filling the matrix $\mathbf{P}$ with an exhaustive simulation, but as its name suggests, it takes a huge amount of time. The analytical calculation of $P_{i,j}$ is also possible when every codeword is considered independent from each other. Such a scenario can be practically implemented by using different types of molecules for consecutive codewords. The current codeword is not affected by the previous codeword. Please note that different types of molecules may have very similar diffusion coefficient considering their Stoke radii are very similar and they are in the same environment [6].

For the derivation of the $P_{i,j}$ values, firstly, the received number of molecules is represented with a Gaussian random variable, as in (2.6). After evaluating the number of received molecules, in BCSK, the decision is made by comparing the number of received molecules with a pre-determined threshold. Thus, the probability of error for the l-th bit of the i-th binary sequence can be stated as

$$Pe_{l,i} = \begin{cases} Q(\frac{\lambda - u_{l,i}}{\sigma_{l,i}}) & b_{l,i} = 0, \\ Q(\frac{u_{l,i} - \lambda}{\sigma_{l,i}}) & b_{l,i} = 1, \end{cases} \tag{3.2}$$

where $u_{l,i}$ and $\sigma_{l,i}$ are mean and standard deviation of the number of received molecules for the l-th bit of the i-th binary sequence, respectively which can be evaluated by utilizing (2.6), $b_{l,i}$ is the binary indicator for the l-th bit of the i-th binary sequence, λ is the pre-determined threshold, and $Q(\,\cdot\,)$ is the tail distribution function of the standard distribution. Therefore, the probability of correct detection is $Pc_{l,i} = 1 - Pe_{l,i}$, and finally

$$\begin{aligned} P_{i,j} &= \prod_{l=1}^{n} O_l^{i,j}, \\ O_l^{i,j} &= \begin{cases} Pc_{l,i}, & b_{l,i} = b_{l,j} \\ Pe_{l,i}, & b_{l,i} \neq b_{l,j} \end{cases}. \end{aligned} \tag{3.3}$$

3.2.2. Decoding Region

Since only 2^k codewords out of the 2^n binary sequences are used for transmission, without loss of generalization, binary sequences can be reordered to place 2^k codewords into the first 2^k rows and eliminate others to obtain

$$\mathbf{C} = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,2^n} \\ C_{2,1} & C_{2,2} & \dots & C_{2,2^n} \\ \vdots & \vdots & & \vdots \\ C_{2^k,1} & C_{2^k,2} & \dots & C_{2^k,2^n} \end{bmatrix}, \tag{3.4}$$

where $C_{i,j} = P(w_j|C_i)$. When a vector is received at the receiver, it should be mapped back to the codeword that was most likely sent. The decoding region for a codeword is defined as the set of binary sequences that are mapped back to the corresponding codeword after decoding. It is defined as

$$\hat{C}_j = \arg\max_i (C_{i,j}), \forall j. \tag{3.5}$$

With codewords C_i, decoding region $\hat{C}_j$, and transition probability matrix $\mathbf{P}$, the codeword error rate (CER) can be found. For a codeword to be detected erroneously, it has to be mapped outside of its decoding region. Hence, a codeword's probability of error can be expressed as

$$CER_i = \sum_{j=1, \hat{C}_j \neq i}^{2^n} C_{i,j}, \tag{3.6}$$

and the average CER can be obtained as

$$CER = \frac{\sum_i^{2^k} CER_i}{2^k}. \tag{3.7}$$

3.3. Codebook Design

As explained in the last section, a codebooks decoding region can be found from (3.5). From its decoding region, CER can also be found with (3.7). The problem is now determining a codebook with a low error rate. A brute-force algorithm is not feasible as the possible number of the combination is huge ($\binom{2^n}{2^k}$ to be exact). To find a feasible solution, heuristics, optimization methods or different types of solvers are needed. Since different solutions have their own benefits and drawbacks, having different types of solutions is beneficial. We propose three different methods: a greedy algorithm, a genetic algorithm, and a mixed integer programming model. While the greedy algorithm is the best one in terms of computational complexity yet has the worst performance, mixed integer programming has the best performance yet the worst in terms of computational complexity. The genetic algorithm may be considered as the midway. While it has the advantage of searching the sample space with great depth, its computation requirement is higher than the greedy algorithm and its performance is not as good as mixed integer programming.

3.3.1. Greedy Algorithm

For an optimization problem, choosing the best option for every step and never tracing back is the basis for greedy algorithms. The main drawback of these type of algorithms is that there isn't any way to escape from a local maxima. When a solution that corresponds to a local maxima is found, the algorithm will be stuck there. They are still useful for some applications as their complexity is low.

The codeword selection problem can be seen as selecting the most scattered set among the possible sets for minimizing the probability of wrong decoding of the transmitted codewords. Many heuristics are introduced in the literature as this is a heavily studied problem over the years. Unfortunately, Euclidean space is used in most of these works, and transition probability matrix $\mathbf{P}$ may not be symmetrical or satisfied the triangular inequality. In other words, these algorithms may not suitable for our problem. A slightly modified version of the heuristic introduced in [39] is given in Algorithm 3.2. Please note that the decoding probability of the selected and candidate codewords should be considered in both directions (i.e., $1-P_{CW_c,CW_s}$ and $1-P_{CW_s,CW_c}$, where CW_s and CW_c stand for a selected and a candidate codeword, respectively) for the distance metric. Thus, $1-\mathbf{P}$ is not a good choice for this algorithm. Instead, the symmetric distance matrix $\mathbf{D} = (1 - \mathbf{P}) + (1 - \mathbf{P})'$ is used.

The performance of the codebooks designed by the proposed algorithm, unfortunately, is not desirable. The codebook does not perform better than the Hamming codes which are not tailored specifically for MC. At this stage, the heuristic cannot consider the possible decoding regions. Having a codebook with low error rate depends on having scattered decoding regions, while the greedy algorithm selecting the scattered codewords and do not consider the decoding regions of these codewords. The problem is that the decoding region can only be found after the codebook is set. Nevertheless, the decoding region for every binary sequence may be guessed with an error margin, and by considering these regions a set of a codeword with better performance can be found.

An n, k Hamming code may be examined to have a better understanding of the decoding region. For Hamming codes, there are 2^{n-k} binary sequences of any codewords decoding region. The selection of codewords has been done in a way that no decoding region has been intersected, and satisfies the number of sequences in a decoding region. When Hamming distance is used as a metric, decoding region prediction for each sequence is not challenging. It can be done by selecting the closest 2^{n-k} binary sequences to that codeword. It is clear that such an approach would be very useful for MC as well. However, for MC, the number of sequences that a codeword have in its decoding region may not be fixed. Thus a modification is needed. We propose to use the decoding region a threshold to predict the decoding region. Please note that this is not the optimal solution in any means, yet it is a better solution than having a fixed number of sequence for a decoding region. The threshold is determined as

$$\gamma = \frac{\sum_{i=1}^{2^n} P_{i,i}}{2^n} K, \tag{3.8}$$

where $P_{i,i}$ stands for the i-th diagonal element of $\mathbf{P}$ which is defined in (3.1), and K is a scaling constant that determines the size of clusters in the greedy heuristic. The

Require $\mathbf{P}, (n, k)$

let w $\in \{w_1, w_2, ..., w_{2^n}\}$ and w_i stand for a binary sequence. ;

let $CW = \{CW_1, CW_2...CW_{2^k}\}$ and CW_i stand for the i-th codeword. ;

let $B = \{B_1, B_2, ...B_{2^k}\}$ and B_i stand for the decoding region of CW_i. ;

let $CW_1 = w_1$ and $B_1 = \{w_1, w_2, ...w_{2^n}\}$;

let $\mathbf{D} \leftarrow (1 - \mathbf{P}) + (1 - \mathbf{P})'$ be the symmetric distance matrix ;

for $l = 1$ to $2^k - 1$ **do**

  $d_{max} \leftarrow max(D_{CW_j,w_i}), w_i \in B_j$ and $1 \leq j \leq l$;

  let w_x be one of the binary sequences whose distance to the codeword of the cluster it belongs to is d_{max} ;

  $CW_{l+1} \leftarrow w_x$;

  for each w_t **do**

    let j be such that $w_t \in B_j$;

    if $P_{CW_{l+1},w_t} \geq P_{CW_j,w_t}$ **then**

      move w_t from B_j to B_{l+1} ;

    end if

  end for

end for

Figure 3.2. The heuristic for codeword selection [39].

predicted decoding region is determined as

$$N_{i,j} = \begin{cases} 1, P_{i,j} \geq \gamma, \\ 0, \text{otherwise}, \end{cases} \tag{3.9}$$

where $N_{i,j}=1$ if the j-th binary sequence is in the i-th codeword's predicted decoding region. CER is determined by the average of all codewords' transition probabilities to the decoding regions of other codewords, while $\mathbf{P}$ contains only the transition probabilities for binary sequences to another. Hence, a modified version of $\mathbf{P}$ is needed. The transition probability of receiving one of the binary sequences that is in the j-th binary

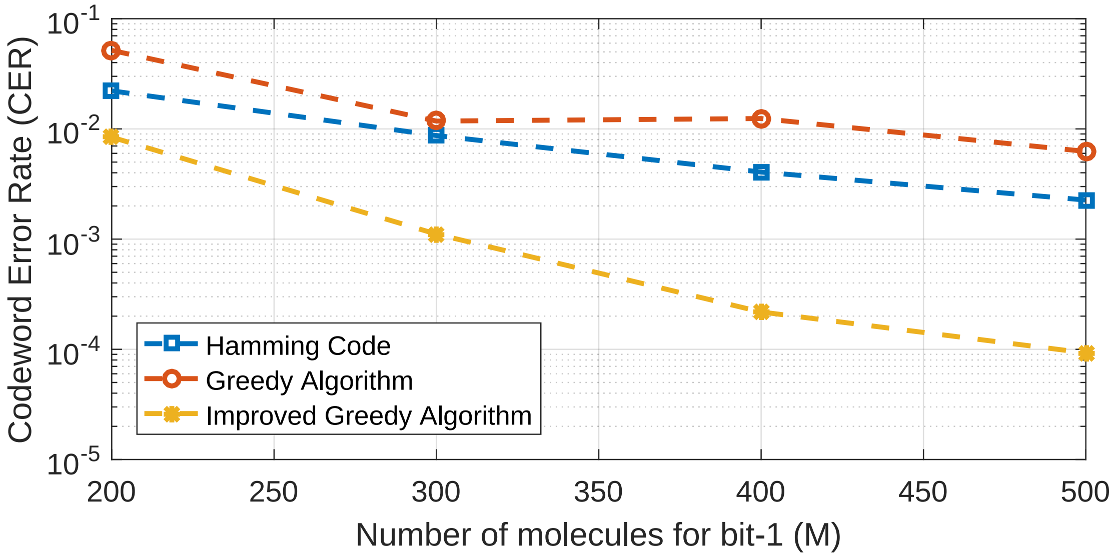

Figure 3.3. Greedy algorithm and Hamming code performance analysis with $(n, k) = (7, 4)$, $K = 0.03$, $t_s = 400\,\mathrm{ms}$, $D = 79.4 \times 10^{-12} \mathrm{\mu m}^2/\mathrm{s}$, $r_r = 5\,\mathrm{\mu m}$, and $r_0 = 10\,\mathrm{\mu m}$.

sequence's predicted decoding region when the i-th binary sequence is sent is given as

$$\widetilde{P}_{i,j} = \sum_{k=1}^{2^n} N_{j,k} P_{i,k}, \tag{3.10}$$

and $\widetilde{\mathbf{P}}$ is a matrix with $\widetilde{P}_{i,j}$ as the i, j-th element. Using $\widetilde{\mathbf{P}}$ instead of $\mathbf{P}$ as the proposed heuristic's input, it is possible to enhance the performance of the algorithm significantly.

The results for the proposed algorithm, its improved version, and Hamming codes with optimum λ are given in Fig. 3.3. The results show that when $\mathbf{P}$ is used in the heuristic algorithm, the designed codebooks do not perform as desired. On the other hand, having $\widetilde{\mathbf{P}}$ improves the performance significantly.

3.3.2. Genetic Algorithm

As an optimization method, genetic algorithm (GA) is inspired by nature itself. The main idea of GA is creating an initial population with diversity to allow for parallel search threads in the search space, and then repetitively combining the good merits of the individuals (i.e., the candidate solutions) via crossovers and optionally mutation steps, and finally applying natural selection for each generation. Although the initial population has a certain importance, the performance of GA does not heavily depend on it.

An individual for GA is represented as a vector of length 2^n (for an (n, k) coding) that is showing the codeword assignments for each binary sequence. The reader should note that the number of unique binary sequences of length 2^n in a candidate solution vector is 2^k, which is composed of the selected codewords from all possible binary sequences. After determining the selected codewords, it is trivial to assign each binary sequence to codewords by considering the probability of error. Therefore, we can represent an individual candidate solution by the set of selected codewords.

For GA to evolve new candidate solutions, it needs an initial population. Even though a totally random initialization would not be a problem, a better approach is to use different heuristics to increase the diversity in the initial population. We used two different heuristics for this purpose: K-means clustering and non-intersected decoding region search heuristics algorithms, given in Fig. 3.4 and 3.5, respectively.

The iteration of GA mainly consists of crossover and mutation operations followed by fitness evaluation. Please recall that an individual is represented by the set of selected codewords. Hence, for the crossover operation, half of the selected codewords come from one parent and the rest come from the other. After the crossover operation, a mutation operation is applied with probability p_μ. For mutation, a random codeword is selected and replaced with the best option in that locality without changing other codewords. Representative GA operations are depicted in Fig. 3.6. The main idea behind it, step by step improving the population while keeping some weak individuals, since a weak codebook might have a strong gene (codeword) that should not be omitted while improving the population. After every crossover and possible mutation, the new born child (codebook) will take place one of the weaker successor of the population. At the end of the generation, the children (codebooks) with same genes are eliminated as they do not bring any variations to the population. Generation after generation, the

Require $\mathbf{P}, IterLim$

let $CrntCWS$ be a randomly selected set of codewords ;

let $\mathbf{D} \leftarrow (1 - \mathbf{P})$ as distance matrix ;

for $counter = 1$ to $IterLim$ **do**

  find $\hat{C}$, the decoder map for $CrntCWS$;

  consider each codeword and its decoding region as clusters ;

  find the center for each cluster from $\mathbf{D}$;

  assign new centers as new codewords and build new codeword set $NewCWS$;

  if $NewCWS = CrntCWS$ **then**

    Break ;

  end if

  $CrntCWS \leftarrow NewCWS$;

end for

Figure 3.4. K-means clustering inspired heuristic.

population gets stronger and smaller. After either a fixed number of generation passed, or the population gets very small, the evolution is stopped and the resulting codebook is used. This idea is similar how the nature improves/adapts itself generation after generation through ages.

3.3.3. Mixed Integer Programming

Integer programming (IP) stands for the mathematical optimization problem with constraints that are either linear, or integer constraints with a linear objective function together with integer variables. If some of the variables are continuous, then it is called an MIP problem. Our problem can be formulated using an IP model; however, from a processing time perspective, a MIP model is preferred for the problem. Please note that, linear programming (LP) is a much faster technique compared to IP or MIP. The reasons behind that the solution for a linear model is easier compared to IP (or MIP). In its basic form, LP finds the set of points, which the conditions are contended. After that, the points that maximizes the cost function (objective function) are selected as

Require N

let CW_{init} be a randomly chosen codeword ;

let DR be the predicted decoding regions of all selected codewords ;

let SC be the set of possible binary sequences whose decoding region has no intersection with DR ;

$CW_1 \leftarrow CW_{init}$;

refresh DR and SC accordingly ;

for $z = 2$ to 2^{k-1} **do**

  randomly select one of the binary sequences in SC, assume SC_x is chosen ;

  $CW_z = SC_x$; refresh DR and SC accordingly ;

  if $SC = \emptyset$ **then**

    recalculate **N** with greater γ

    restart the algorithm with new **N**

  end if

end for

Figure 3.5. Non-intersected decoding region search.

the solution. Its complexity is in polynomial time (P), which means that it can be efficiently solvable or tractable in general (not for every single case!). The key idea here solution space may be represented by a single convex polyhedron for a linear programming model. On the other hand, the continuous solution space is not available for IP, instead there are many disjoint solution spaces. There is no way to have a convex solution space as we have for LP in which much easier to optimize. IP is an Non-deterministic-polynomial time complete (NP-complete) problem.

For both IP and MIP, if there is an optimal solution, it is guaranteed that it can be found with IP or MIP solvers given enough time and memory. The main challenge is that the required time and memory may not be feasible depending on the model. Unfortunately, as we aim designing more powerful and longer codebooks, in other words as n and k increase, the search space grows exponentially and finding optimal solutions with MIP becomes almost impossible. Still, with reasonable time

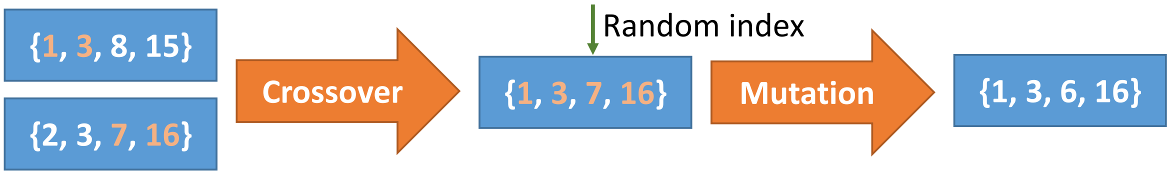

Figure 3.6. Representative genetic operations where the selected codewords are represented by numbers. For the crossover operation, 1 and 3 are selected from the first parent while 7 and 16 are selected from the second parent. For the mutation operation, the third gene is replaced with the best option locally.

and memory constraints, MIP provides solutions that are close to the optimal solution. Hence, it can be conveniently stated that MIP is useful for channel code design.

Before modeling the problem, it is important to recall that we are trying to minimize the CER, which is given in (3.7) (or equivalently, maximizing the probability of correct transmission $PoCT = 1 - CER$). The first observation about the codebook design problem is that the decoding region is decided after the codebook is designed. This is a sequential operation and is not supported by IP models. Thus, a different approach is needed. Secondly, the transition from $\mathbf{P}$ to $\mathbf{C}$ is an index operation, and it is not considered as a linear operation for MIP solvers. Hence, instead of eliminating the rows, we need a binary indicator to represent the decoding regions. The IP model is given as

$$CL_i \in \{0,1\} \tag{3.11}$$

$$CM_{i_1,i_2} \in \{0,1\} \tag{3.12}$$

$$\sum_i CL_i = 2^k \tag{3.13}$$

$$\sum_{i_1} CM_{i_1,i_2} = 1 \tag{3.14}$$

$$\sum_{i_2} CM_{i_1,i_2} <= CL_{i_1} 2^n \tag{3.15}$$

$$\sum_{i_2} CM_{i_1,i_2} >= CL_{i_1} \tag{3.16}$$

$$PoCT = \frac{\sum_{i_1} \sum_{i2} CM_{i_1,i_2} P_{i_1,i_2}}{2^k} \tag{3.17}$$

where i, i_1, and $i_2 \in [1, 2, ...2^n]$ with possibly distinct values, CL stands for coset leader, which is the indicator of whether the corresponding binary sequence is a codeword and CM_{i_1,i_2} stands for the coset member, whether a given binary sequence i_2 belongs to decoding region of the corresponding binary sequence's i_1, i.e. its coset. Condition (3.13) states that there are 2^k coset leaders to be selected. Condition (3.14) guarantees that each coset member can be a member for exactly one coset leader. If a binary sequence is selected as a coset leader, then its coset may have members. Otherwise, there cannot be any member of that coset; hence, the corresponding row of **CM** has to be filled with zeros. This condition is stated with (3.15). Equation (3.16) requires each coset leader to have at least one coset member (each coset leader is expected to be a member of its own coset). Finally, the objective function is defined in (3.17), which is the probability of correct transmission and maximizing it is the objective. For processing time purposes, equation (3.12) can be restated as

$$0 \leq CM_{i_1,i_2} \leq 1. \tag{3.18}$$

Since the best solution for decoding for a received vector is mapping back the codeword that is most likely to have been transmitted, the solver does not assign continuous values to CM_{i_1,i_2} even though it is allowed. Hence, restating the condition does not

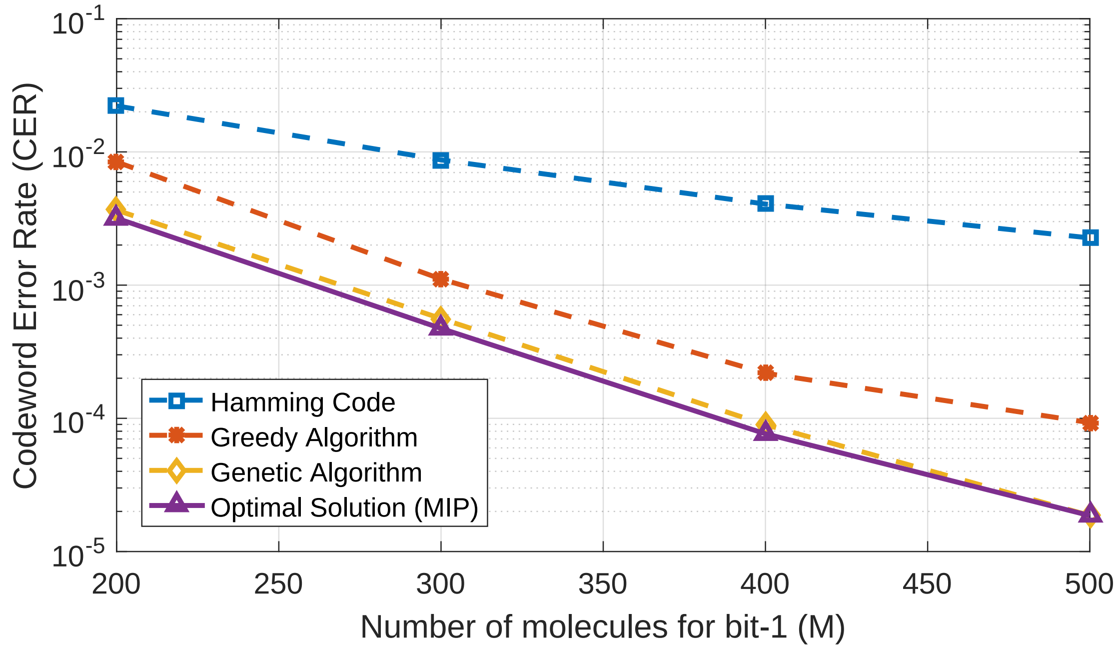

Figure 3.7. CER comparison for all considered methods and Hamming code with $(n, k) = (7, 4)$, $t_s = 400\,\text{ms}$, $D = 79.4 \times 10^{-12} \mu\text{m}^2/\text{s}$, $r_r = 5\,\mu\text{m}$, $K = 0.03$, and $r_0 = 10\,\mu\text{m}$.

harm the performance and decreases the processing time significantly. Since CM_{i_1,i_2} is a continuous variable, the problem is stated with a MIP model.

The results for Hamming code, greedy algorithm, GA, and the optimal solution found with the MIP solver are given in Fig. 3.7. As expected, the Hamming code performs worst, the greedy algorithm performs better, GA is even better and slightly worse than the optimal solution except for $M = 500$. For $M = 500$, the GA also finds the optimal solution.

3.4. Performance Evaluation

One of the crucial points for fair comparison of the uncoded case with coded cases is that the data rate and the average power should be normalized. When channel codes are used, the symbol duration t_s and the number of molecules for bit-1 M should thus be decreased to $\frac{k}{n}t_s$ and $\frac{k}{n}M$, respectively. In parallel to the explanation in Section 2, as t_s decreases, the channel suffers from ISI even more. Hence, the performance severely degrades. Moreover, power normalization also decreases the error performance of the system. Thus, the designed channel code should perform better than the uncoded case even under these poor conditions.

Please note that, when noise is present, the only difference is that the received number of molecules should be stated with an additive white Gaussian noise as

$$\hat{N}_i^{Rx} = N_i^{Rx} + \mathcal{N}\left(0, \sigma_n^2\right) \tag{3.19}$$

where σ_n^2 is the variance of the noise distribution. In classical communication, signal-to-noise ratio (SNR) is used as a metric to represent the ratio of the power of the desired signal to the background noise, and it has a one-to-one relation with BER. Eventhough there are different SNR definitions available in the MC's literature, none of them, to the best of our knowledge, has a one-to-one relation with BER. Hence, we believe that using σ_n^2 as a metric is more insightful than SNR.

The performance evaluation of the designed codebooks is obtained using (3.7) with the parameters given in Table 3.1. It is assumed that after every codeword transmission, the channel is cleared. Moreover, finding the optimal solution with MIP might be time consuming; therefore, a time limit is set. Another crucial point is that the threshold λ is selected such that the received vector has the least error while every binary sequence is transmitted independently, and the channel is cleared after every codeword transmission, which is not optimal. The threshold λ is obtained when any binary sequence can be transmitted from the transmitter; however, only codewords are transmitted when coding is used. What should be done is to recalculate λ after each codebook design, which means that $\mathbf{P}$ should be recalculated. With new $\mathbf{P}$, the codebook search methods should be repeated and new codebooks should be designed. The same steps should be repeated until the threshold converges to a certain λ value. Not only this procedure is time consuming, but also the performance gain is negligible. Hence, λ is only calculated once considering the transmission of all possible binary sequences. In addition, for any combination of (n, k) where $k < n$, it is possible to design channel codes with the proposed methods. We choose $k = 7$ and $n = 10$ for

Table 3.1. Simulation Parameters.

Parameter	Variable	Value
Diffusion Coefficient	D	79.4, 61 μm²/s [4,6]
Receiver Radius	r_r	5 μm
Tx-Rx distance	r_0	10 μm
Channel Memory	I	7
Symbol Duration	t_s	250 ms, 400 ms
Released MM (per bit-1)	M	200, 300, 400, 500, 600
Variance of noise distribution	σ_n^2	$80 - 0$
Message and block length	(n, k)	$(10, 7)$

simulation purposes only. Finally, the parameters that are given in Table 3.1 are for the uncoded case. For coded cases, the symbol duration t_s and the number of molecules M per bit-1 are normalized, and the channel memory I is increased to $\frac{n}{k}I$.

The results of the $(5, 2, 2)$ ISI-free code[1] [17], (31,4,3) minimum energy code [14], (2,1,6) SOCC[2] [20], (15,11) and (7,4) Hamming codes, the proposed solutions, and the uncoded case for different symbol durations, D, and σ_n^2 are given in Fig. 3.8 - Fig. 3.12. Please note that, M and t_s values indicated in the results are all for uncoded cases.

The results show that, while other coding schemes perform worse than the uncoded case, MIP solution has a clear advantage for all presented different cases, except when noise is severe at $t_s = 400$ ms. The greedy algorithm may be considered as a feasible solution under several conditions. Finally, the genetic algorithm performs similarly to MIP. The coding gain is higher for short symbol duration (e.g. at $t_s = 250$ ms)

[1]Since ISI-Free codes are designed to neglect the effect of ISI caused by the previous codeword, it is considered that the channel is cleared after each 10-bit transmission, which corresponds to twice the codeword length for the (5,2,2) ISI-free code and the codeword length for proposed (10,7) codes.

[2]Since codewords are not defined for convolutional codes, BER is shown instead of CER. Moreover, the channel is cleared after each 10-bit transmission, which corresponds to codeword length for the proposed (10,7) codes.

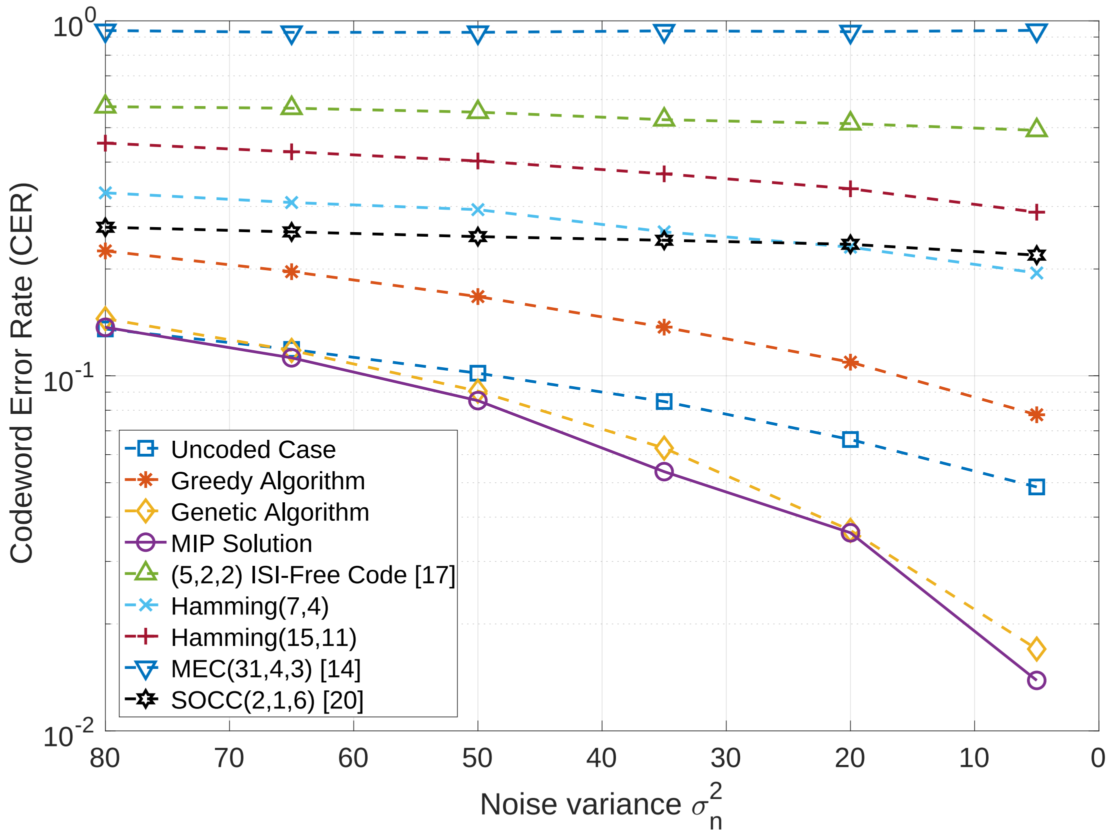

Figure 3.8. CER of the coded and uncoded transmissions in the presence of system noise for $t_s = 250\,\mathrm{ms}$, $D = 79.4\,\mathrm{\mu m^2/s}$, $M = 400$.

as expected, since the ISI is more severe for short symbol durations, and our channel coding techniques are designed to combat ISI. The purpose behind channel coding is achieving a low error rate at a high data rate (short symbol duration). For lower symbol durations, we expect that the coding gain increases up to a peak point. After that, the error rate starts to increase to the point where channel coding is meaningless due to the data rate and power penalties it brings. Our trials show that, at that point, the communication link (coded or uncoded) has a considerably higher error rate, in which a reliable communication link cannot be established. Consequently, for very low data rates, using a coding scheme may not be reasonable since uncoded communication already has a low error rate; hence, the coding gain is limited.

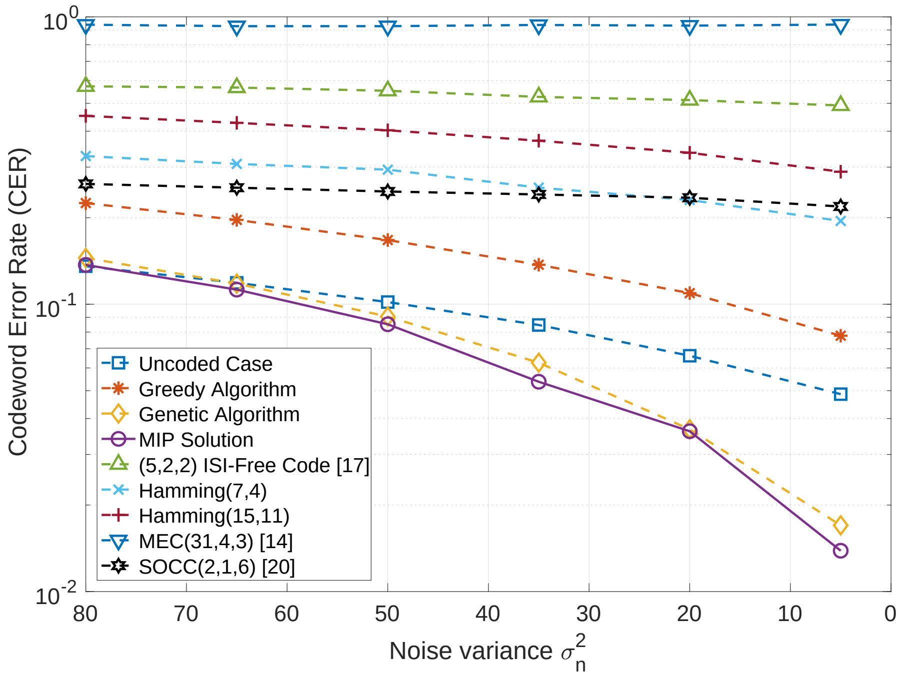

Figure 3.9. CER of the coded and uncoded transmissions in the presence of system noise for $t_s = 400\,\text{ms}$, $D = 79.4\,\mu\text{m}^2/\text{s}$, $M = 400$.

As a final remark, the block length is not quite high considering the other known coding schemes. Due to the complex nature of integer programming, finding a solution via MIP could be very time consuming. However, with greedy algorithm, a longer block code design could be done. Even with the genetic algorithm, such a design could be possible. One of the most time consuming part of the genetic algorithm is the mutation step, which is based on finding the best codeword for that codebook. Such an operation's complexity grows exponentially with respect to the block length, n. Hence, designing channel codes with longer block length is not an easy task. Another crucial limiter for all the methods is the size of the probability transition matrix $\mathbf{P}$. Since its size $2^n \times 2^n$, the needed memory would be extremely high for $n > 15$. What is needed then having a simpler distance metric (like Hamming distance). However, defining such a metric is not an easy task and there is none available in the literature to the best of our knowledge. When such a metric is found, the provided methods and algorithms should be updated to take advantage of it.

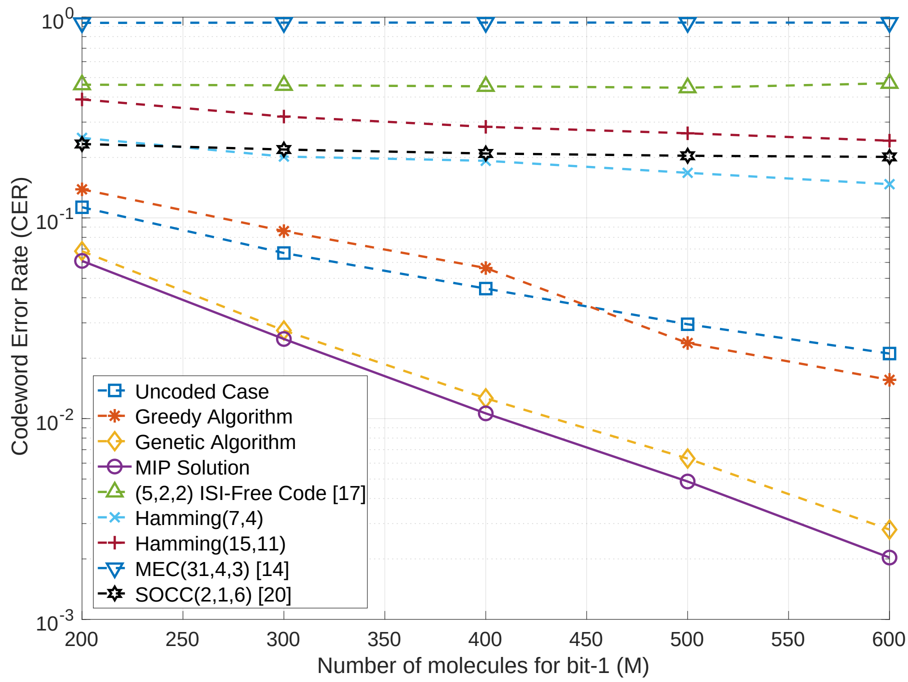

Figure 3.10. CER of the coded and uncoded transmissions in the presence of system noise for $t_s = 250\,\text{ms}$, $D = 79.4\,\mu\text{m}^2/\text{s}$, $\sigma_n^2 = 0$.

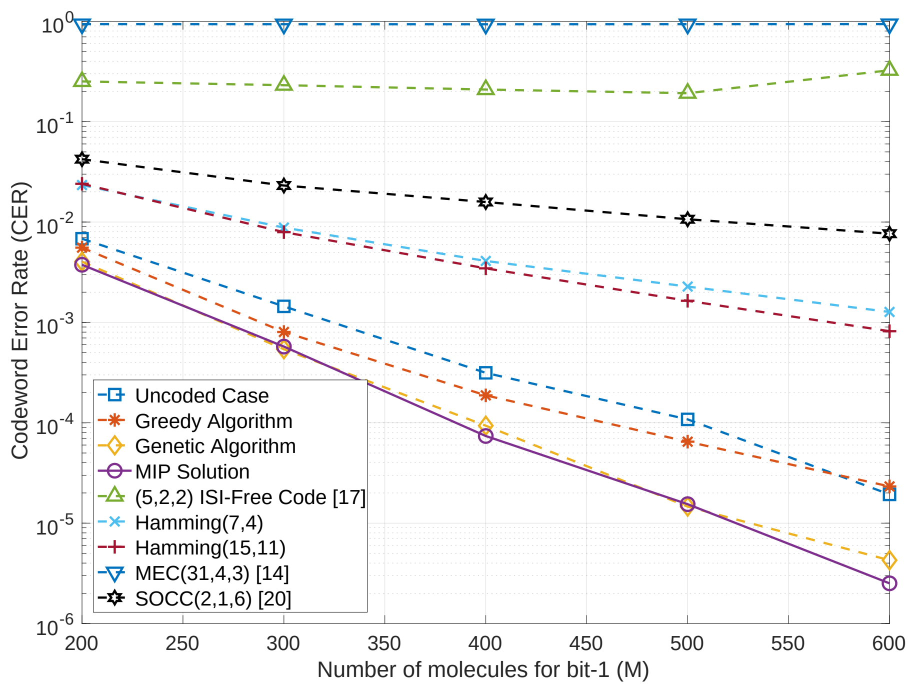

Figure 3.11. CER of the coded and uncoded transmissions for $t_s = 400\,\text{ms}$, $D = 79.4\,\mu\text{m}^2/\text{s}$, $\sigma_n^2 = 0$.

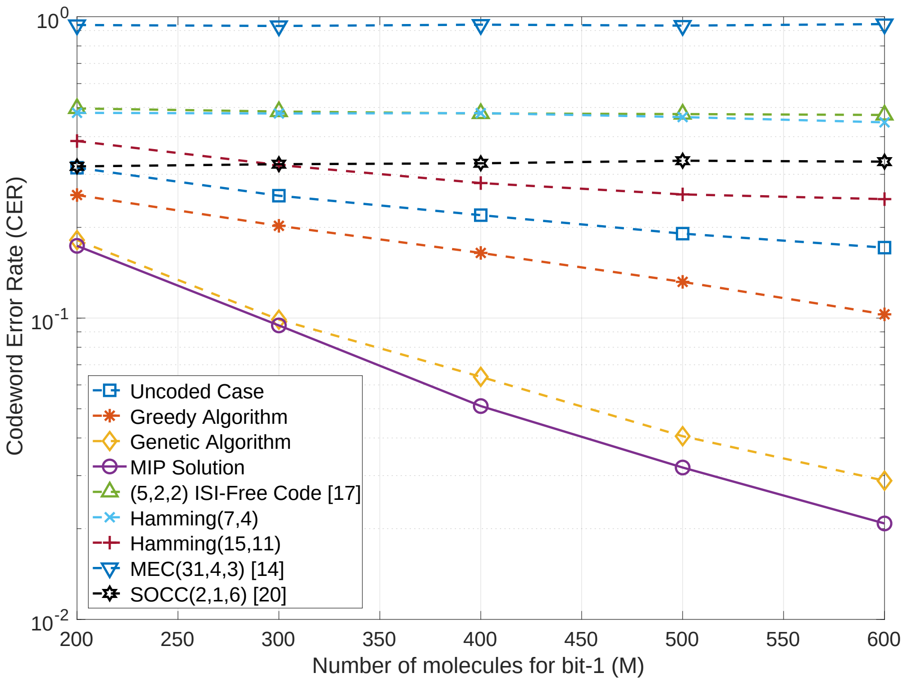

Figure 3.12. CER of the coded and uncoded transmissions for $t_s = 250\,\text{ms}$, $D = 79.4\,\mu\text{m}^2/\text{s}$, $\sigma_n^2 = 0$.

4. ISI-MINIMIZING CODES

4.1. ISI-Minimizing Code Design

In this chapter, we proposed ISI-minimizing codes, which are denoted as (n,k) ISI-min, are based on codewords that have no consecutive bit-1s. This coding scheme also allows a simple yet very effective adaptive coding scheme that can be updated for every received bit sequence. In particular, each codeword (CW) in the codebook has three properties:

- it cannot have consecutive bit-1s
- it starts with bit-0 to avoid consecutive bit-1s between two neighbors CWs
- it has to have at least one bit-1

The first and second properties minimize the ISI between consecutive bits. The third property is used for the proposed detection algorithm. Please note that the proposed code family is a block code where k-bit information sequences are mapped to n-bit CWs.

The optimum threshold for a CW is the threshold that minimizes the bit error rate after decoding. The proposed detection has been done with a threshold that adapts itself for each codeword. Let $\mathbf{r}^i = (r_2^i, r_3^i, ...r_n^i)$ be the received number of molecules of each bit of the i-th transmitted codeword, except for the first bit (r_1^i corresponds to bit-0, certainly). Since a codeword consists of bit-1s and bit-0s, the optimum threshold has to be in between $d_1 = \max \mathbf{r}^i$ and $d_0 = \min \mathbf{r}^i$. By using this trivial observation, the threshold for the i-th codeword can be estimated as

$$\tau^i = ad_0 + (1-a)d_1 \tag{4.1}$$

where a is the scaling factor. With the correct selection of a, the optimum threshold can be found. Nonetheless, determining a is a challenging task.

The following section explains the construction of a codebook for a desired n. We then demonstrate how the erroneously detected binary sequences are corrected and mapped back to a codeword. Finally, two methods are introduced to determine the scaling factor a.

4.1.1. Codebook Construction

A codebook with a desired length n can be constructed by following a set of simple steps. First, we need the set of codewords that satisfy the properties of the proposed code family. For $n = 3$, the set of codewords is

$$\mathbf{CW}_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.2}$$

where each row is a codeword. Note that the first column of $\mathbf{CW}_3$ is filled with zeros, since the first bit of each codeword has to be a bit-0. Let $\mathbf{W}_3$ be the matrix that contains bit sequences that start with bit-1 and do not have consecutive bit-1s for $n = 3$. The $\mathbf{W}_3$ can be stated as

$$\mathbf{W}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \tag{4.3}$$

By concatenating $\mathbf{CW}_3$ and $\mathbf{W}_3$, $\mathbf{CW}_4$ can be constructed as

$$\mathbf{CW}_4 = \left[\begin{array}{c:ccc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \hdashline 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right] = \left[\begin{array}{c:c} \mathbf{0} & \mathbf{CW}_3 \\ \hdashline \mathbf{0} & \mathbf{W}_3 \end{array}\right] \tag{4.4}$$

In a similar manner, $\mathbf{W}_4$ can be constructed as follows:

$$\mathbf{W}_4 = \left[\begin{array}{c:ccc} 1 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{c:c} 1 & \mathbf{0} \\ \hdashline \mathbf{1} & \mathbf{CW}_3 \end{array}\right]. \tag{4.5}$$

In other words, for $n > 3$, with the knowledge of $\mathbf{CW}_n$ and $\mathbf{W}_n$, $\mathbf{CW}_{n+1}$ and $\mathbf{W}_{n+1}$ can be constructed as

$$\mathbf{CW}_{n+1} = \left[\begin{array}{c:c} \mathbf{0} & \mathbf{CW}_n \\ \hdashline \mathbf{0} & \mathbf{W}_n \end{array}\right] \tag{4.6}$$

$$\mathbf{W}_{n+1} = \left[\begin{array}{c:c} 1 & \mathbf{0} \\ \hdashline \mathbf{1} & \mathbf{CW}_n \end{array}\right]. \tag{4.7}$$

Starting from $n = 3$, the codebook for $n > 3$ can be constructed recursively, using equations in (4.6) and (4.7).

To prove the validity of this codebook building method, we first need some observations on $\mathbf{CW_n}$ matrix. As explained before its first column is filled with zeros. Second observation is, when you remove this column, the remaining binary sequences has to follow the first rule and the third rule. The first rule, which is a codeword cannot have consecutive bit-1s, is satisfied; since, there is not any bit (column) concatenated with the second column, hence this rule cannot be broken with this operation. The third rule, which is a codeword has to have at least one bit-1, also satisfied after removing the first column, because the first column filled with zeros. None of the bit-1s are removed with this operation. Now the remaining matrix can be categorized into two groups. The first group is the bit sequences that starts with bit-0, the second group is the bit sequences that starts with bit-1 (Please note that there cannot be any other possibility here! We have covered every binary sequence in the remaining matrix). As we already explained while both group satisfy the first and the third rule, only the first group satisfies the first rule also. Hence, we can now safely indicate the first group with $\mathbf{CW}_{n-1}$ and the second group with $\mathbf{W}_{n-1}$ from their definition. We have now one final thing to prove: construction of $\mathbf{W}_n$ from $\mathbf{CW}_{n-1}$. Note that, by definition, the first column of $\mathbf{W}_n$ contains only bit-1s. If we remove the first column, the remaining binary sequences will follow the every three rule of $(n-1), k$ ISI-min codes with one exception, one of the binary sequence does not have bit-1 as shown in (4.5). The observation here, $1, 0, 0, ...$ is a viable binary sequence for $\mathbf{W}_n$ as it does not have any consecutive bit-1s and starts with bit-1. However, when the first bit-1 is removed, the remaining sequence is not a viable codeword due to the third rule, there has to be a bit-1 in a codeword. Then, while constructing $\mathbf{W}_n$ this sequence has to be handled exclusively as done in (4.7). This concludes the explanation of how the codebook can be generated recursively.

4.1.2. Detection and Error Correction

Due to the properties of the proposed coding scheme, we know that there are at least one bit-1 and one bit-0 present in the codeword. Moreover, the first bit of a codeword is definitely bit-0. By using these facts, the decoding algorithm for (n,k) ISI-min codes is given in Algorithm 4.1.

The decoding algorithm is mostly straightforward and simple. The part that needs further explanation is the last step. When two consecutive bits are detected as bit-1, at least one of them should be bit-0. Hence, there are three possibilities: 00, 01 and 10. To determine which one is the most probable, we need to consider the effects of previously transmitted bits, which can be stated for the i-th bit by using (2.6) as

$$S_{i,y} \sim \mathcal{N}\left(\sum_{k=1}^{i-1} M_k p_{i-k+y+1}, \sum_{k=1}^{i-1} M_k p_{i-k+y+1}\left(1 - p_{i-k+y+1}\right)\right) \tag{4.8}$$

Require $\mathbf{r}^i, \tau^i$

let $\mathbf{z}^i = (z_2^i, ..., z_n^i)$ be the binary indicator of i-th received binary sequence ;

let $l = 2, 3, ..., n$;

$\mathbf{z}_l^i \gtrless \tau^i$; {make a decision on $\mathbf{z}_l^i$ for all l }

$mp = \arg\max \mathbf{r}^i$. ; {find the bit position with the highest number of molecules}

$z_{mp}^i = 1$; {mp is the most likely position that bit-1 is sent}

$z_{mp-1}^i \leftarrow 0$, $z_{mp+1}^i \leftarrow 0$; {consecutive bit-1s are not allowed}

if $z_l^i = 1$ and $z_{l+1}^i = 1$ **then**

  $z_{l+1}^i \leftarrow 0$; {this part is explained in (4.8)-(4.9)}

end if

Figure 4.1. Proposed decoding algorithm.

where the subscript y stands for the delayed slot compared to i. The probabilities of detecting both z_v^i and z_{v+1}^i for any $v = 2, 3, .., n-1$ as bit-1 can be stated as

$$P(z_v^i = 1, z_{v+1}^i = 1 | CW_v^i = 1, CW_{v+1}^i = 0) = P(N_{iv,iv}^{Rx} + S_{iv,0} > \tau^i) P(N_{iv+1,v}^{Rx} + S_{iv,1} > \tau^i) \tag{4.9a}$$

$$P(z_v^i = 1, z_{v+1}^i = 1 | CW_v^i = 0, CW_{v+1}^i = 1) = P(S_{iv,0} > \tau^i) P(N_{iv+1,iv+1}^{Rx} + S_{iv,1} > \tau^i) \tag{4.9b}$$

$$P(z_v^i = 1, z_{v+1}^i = 1 | CW_v^i = 0, CW_{v+1}^i = 0) = P(S_{iv,0} > \tau^i) P(S_{iv,1} > \tau^i) \tag{4.9c}$$

where $iv = n(i-1)+v$. It is clear that (4.9c) is the least likely one, as the probability of error is assumed to be higher than the probability of correct transmission. Comparing (4.9a) and (4.9b), is unfortunately no trivial task and calls for more observations. First, the terms that stand for the error probability give more information than the terms that stand for the correct transmission. This is because the error probability is, as stated before, lower than the probability of correct transmission. Second, the error terms at (4.9a) and (4.9b) are $P(N_{iv+1,iv}^{Rx} + S_{iv,1} > \tau^i)$ and $P(S_{iv,0} > \tau^i)$, respectively. While $N_{iv+1,iv}^{Rx}$ includes the p_2 term, $S_{iv,0}$ might not have it. When they both have the p_2 term, $S_{iv,1}$ has the p_3 term and $S_{iv,0}$ cannot have a p_3 term since, consecutive bit-1s are not allowed ($S_{iv,0}$ cannot have both p_2 and p_3). Thus, the probability calculated by (4.9a) is higher in general, and when two consecutive bit-1s are detected, the last bit is corrected as bit-0.

When three consecutive bit-1s are detected, there are five different combinations that might have been transmitted: 101, 100, 010, 001, and 000. Since the error probability is much lower than the probability of correct transmission, 101 is the most probable sequence. While the other combinations have at least 2-bit errors, 101 has a single bit error. This result is consistent with the two consecutive bit-1s problem. When 111 is detected, the first two consecutive bit-1s are resolved by setting the second bit as 0 and the result becomes 101, which is the same result when two consecutive bit-1s are corrected as 10. The solution of these two cases, as expected, is consistent with each other.

Even though it's very unlikely, it is possible to have four, five or more consecutive bit-1s at the detection stage. When odd number of bit-1s (e.g. three (111), five (11111), seven (1111111) etc.) it is clear that putting zero in between bit-1s (e.g. 111 − > 101, 11111 − > 10101, 1111111 − > 1010101, etc.) yields the most likely solution as in the case of having three consecutive bit-1s, since the probability of correct transmission is much higher than the probability of erroneous transmission. When even number of consecutive bit-1s are detected (e.g. four (1111), six (1111111), etc.) determining the most probable bit sequence exactly is a greater challenge. Still as in the case of having two consecutive bit-1s, setting evenly indexed positions as bit-0 (e.g. 1111 − > 1010, 111111 − > 101010, etc.) in general yields the most likely sequence. In other words, for longer consecutive bit-1 sequences, the correction rule should stay the same. Starting from the beginning, correcting the 11 as 10 is a consistent correction method for any number of consecutive bit-1s.

As explained in 2, the molecules that moves randomly in the environment is the released molecules that when bit-1 is transmitted. Moreover, their effect as ISI are the greatest to the following symbol duration, since the channel taps are assumed to be ordered. By correcting the detected two consecutive bit-1s as 10 is actually negates the effect of the greatest ISI term caused by transmitting bit-1, as it negates the ISI for the symbol duration following to transmitting bit-1.

4.1.3. Threshold Scaling Constant Estimation

The optimum threshold for each codeword varies between the minimum and the maximum R^i values as indicated in (4.1). In fact, a very challenging task consists of estimating the optimum a for every codeword is a very challenging task. To determine a, we propose two methods.

The first method is by using pilot codewords (coded pilot symbols). Since the pilot codewords are also known at the receiver, the optimum threshold for each codeword can be found. For every pilot codeword, the required equation can be stated as

$$ad_0^i + (1-a)d_1^i = \tau_{opt}^i. \tag{4.10}$$

This equation can also be written as

$$(d_0^i - d_1^i)a = \tau_{opt}^i - d_1^i \tag{4.11}$$

where $i = 1, 2, ...\hat{L}$. This equation set can be stated in matrix form as

$$\begin{bmatrix} d_0^1 - d_1^1 \\ d_0^2 - d_1^2 \\ \vdots \\ d_0^{\hat{L}} - d_1^{\hat{L}} \end{bmatrix} \begin{bmatrix} a \end{bmatrix} = \begin{bmatrix} \tau_{opt}^1 \\ \tau_{opt}^2 \\ \vdots \\ \tau_{opt}^{\hat{L}} \end{bmatrix}. \tag{4.12}$$

Because there is only one variable and $\hat{L}$ equations, a unique solution does not exist, but least square (LS) solution can be found as

$$a_{LS} = (\mathbf{d}^T\mathbf{d})^{-1}\mathbf{d}^T\tau_{opt} \tag{4.13}$$

where $\mathbf{d} = [d_0^1 - d_1^1, d_0^2 - d_1^2, ..., d_0^{\hat{L}} - d_1^{\hat{L}}]^T$ and $\tau_{opt} = [\tau_{opt}^1, \tau_{opt}^2, ..., \tau_{opt}^{\hat{L}}]^T$. Please note that, the pilot symbols are sent once to determine a at the beginning of the communication.

To determine a using the second method, we propose an analytic solution. As explained in the previous section, after the thresholding, we correct 11 as 10. In other words, an erroneously detected bit-0 that comes after a bit-1 is most likely corrected. That's why, the possible strongest ISI term that causes an error on bit-0 may come from the two previous bits, which is distributed as $ISI_3 \sim \mathcal{N}(M_3p_3, M_3p_3(1-p_3))$ (instead of ISI_2). Please note that ISI terms on bit-1s are not destructive. In contrast, they help the symbol to be detected as bit-1. Assume that the distribution of the molecules if bit-1 is transmitted at the current slot, which can be modeled as $X_{signal} \sim \mathcal{N}(M_1p_1, M_1p_1(1-p_1))$. Obviously, the number of received molecules at x-th slot can be represented as $N_x^0 = \sum_{x=3}^{\infty} b_x ISI_x$ if the transmitted symbol is bit-0, and $N_x^1 = X_{signal} + \sum_{x=3}^{\infty} b_x ISI_x$ if the transmitted symbol is bit-1. Our approach for deriving an analytic threshold consists of determining the weakest N_x^1 and the strongest N_x^0 that form the most susceptible case for erroneous decoding. For this case, we derive the probability of error and minimize this function with respect to the threshold. In particular, ISI_x for $x > 3$ is ignored, number of received molecules at i-th slot for transmitting bit-0 and bit-1 are $N_x^1 = X_{signal}$ and $N_x^0 = ISI_3$ respectively. The error probability for decoding N_x can be written as

$$P_E = P_0 Q\left(\frac{\tau^i - Mp_3}{\sqrt{Mp_3(1-p_3)}}\right) + P_1\left(1 - Q\left(\frac{\tau^i - Mp_1}{\sqrt{Mp_1(1-p_1)}}\right)\right) \quad (4.14)$$

where P_0 and P_1 are the transmission probability of bit-0 and bit-1, respectively. By replacing τ^i with (4.1), taking the derivative and equating it to 0 yields

$$P_1 \frac{e^{\frac{-\left(-Mp_1 + ad_0 + (1-a)d_1\right)^2}{2Mp_1(1-p_1)}}}{\sqrt{Mp_1(1-p_1)}} = P_0 \frac{e^{\frac{-\left(-Mp_3 + ad_0 + (1-a)d_1\right)^2}{2Mp_3(1-p_3)}}}{\sqrt{2Mp_3(1-p_3)}}. \quad (4.15)$$

Finally, the solution of (4.15) can be obtained as

$$a_{an} = \frac{Mp_1R - Mp_3U + d_1U - d_0R - \sqrt{R \quad U(Mp_1 - Mp_3)^2 - 2RU \log \sqrt{\frac{U}{R}}(R-U)}}{(U-R)(d_1-d_0)}. \tag{4.16}$$

where $U = \frac{Mp_1(1-p_1)}{P_1}$ and $R = \frac{Mp_3(1-p_3)}{P_0}$. Please note that, while the LS solution determines the value of a once and uses it for subsequent transmissions, the analytic solution is updated a for every codeword.

As a final note, the ISI for a bit-0, can be upper bounded with $\sum_{i=1}^{\lfloor (I-1)/2 \rfloor} ISI_{2i-1}$. For classical coding techniques (or coding techniques that are not specifically tailored for MCvD), this kind of an upper bound would be higher, in general, $\sum_{j=1}^{I} ISI_i$. Unfortunately, this upper bound does not have a one-to-one relation with error rate; hence, cannot be used as a performance metric. We still believe that it is nice to have it to show the superior side of the proposed coding technique.

4.2. Simulation and Experimental Results

4.2.1. Simulation Results

While comparing coding strategies, it is very important to normalize the average power and symbol duration according to the coding rate. We use the uncoded communication data rate and average power as a reference, and normalize accordingly. For an (n, k) block code, the symbol duration t_s and the number of molecules released for bit-1 M are decreased to $\frac{k}{n}t_s$ and $\frac{k}{n}M$, respectively[3] . Decreasing the power and the symbol duration affects the BER performance significantly. Thus, channel coding strategies should perform at least better than the uncoded case under normalized condition. Otherwise, it is not meaningful to use that channel coding strategy at all.

[3] Please note that, when sparse codes are used, M would change according to k, and the total number of bit-1s and bit-0s in the codebook.

Table 4.1. Simulation Parameters.

Parameter	Variable	Value
Diffusion Coefficient	D	$79.4\,\mu m^2/s$
Receiver Radius	r_r	$5\,\mu m$
Tx-Rx distance	r_0	$10\,\mu m$
Channel Memory	I	100
Symbol Duration	t_s	160 ms, 200 ms, 300 ms, 400 ms
Released MM (per bit-1)	M	$400 - 10000$
Message and block length	(n, k)	$(18, 12)$

In general, channel coding strategies are based on correcting up to a number of errors in an erroneously detected binary sequence. If the received sequence has more errors than the channel code can correct, then the received sequence cannot be decoded correctly. As a result, classical channel codes perform better when the error rate is low, and have a worse BER performance than the uncoded communication otherwise. However, such a coding strategy is not suitable for MCvD due to the heavy ISI channel. Our proposed coding scheme also does not follow such a traditional strategy; hence, it is possible to have a desirable performance even when the channel conditions are harsh. The simulation parameters for the uncoded case are given in Table 3.1.

Please note that, while a pre-determined threshold is needed in the BCSK, our proposed scheme has an adaptive threshold. Thus, a pre-determined threshold is needed for comparison. The BER performance of other coding strategies and the uncoded case are obtained according to a threshold that minimizes the bit error rate at the receiver (this is also called the genie threshold).

The simulation results for the uncoded BCSK, (15,11) Hamming codes, (31,4) MEC codes [14, 40], (5,2,2) ISI-free codes [16, 17], (2,1,6) SOCC codes [20], and the proposed (18,12) ISI-min coding scheme with pilot symbol aided and analytically determined thresholds for different conditions are given in Fig. 4.2 - Fig. 4.7.

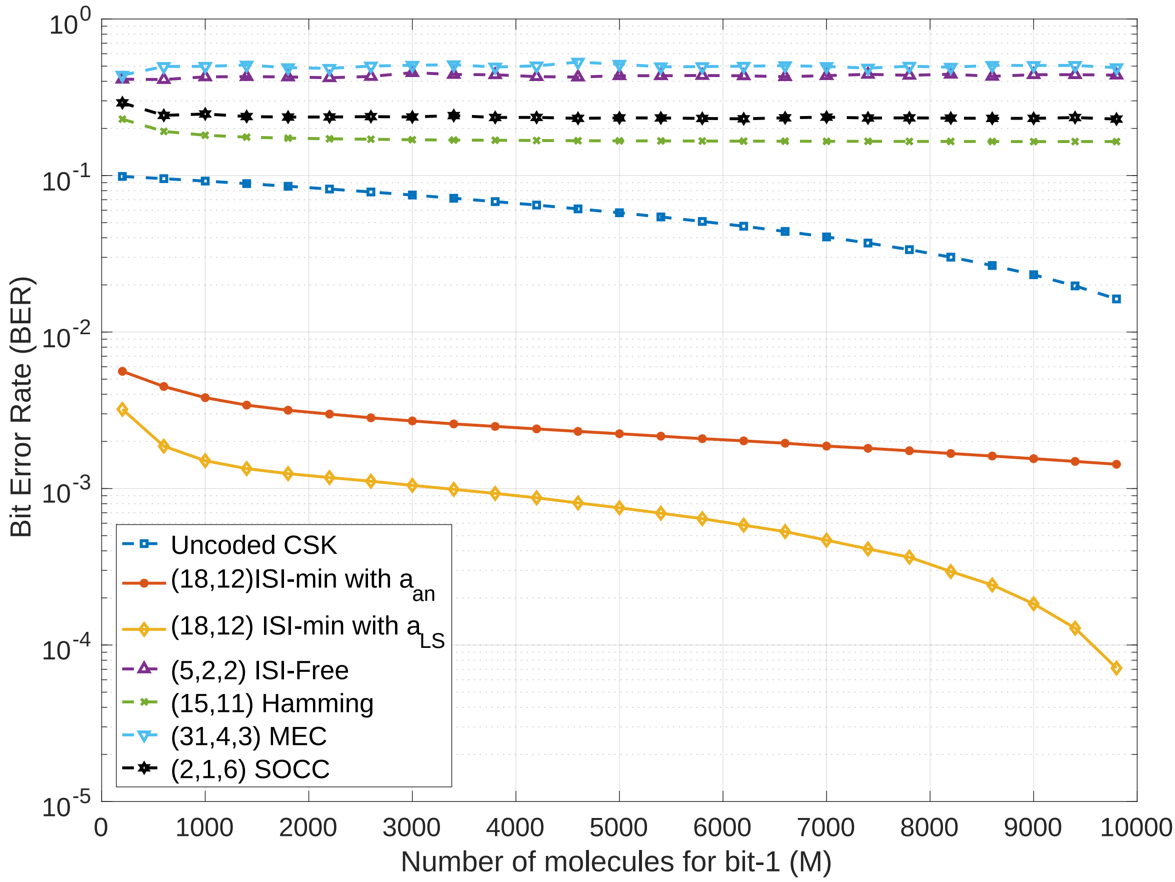

Figure 4.2. Comparison of different coding strategies and uncoded transmissions for $t_s = 200\,\mathrm{ms}$, $\sigma_n^2 = 0$.

First observation about the results is that the analytical solution has a desirable BER performance for high symbol durations, yet the same is not correct for low symbol durations. This is expected since the analytical solution is based on the assumptions that hold for high symbol durations. On the other hand, the pilot symbol aided method performs well, independent of the symbol duration. The results when noise is present are also given in Figs, where the coding gain is significantly high for both analytic and pilot aided method. Please note that the proposed pilot symbol aided coding scheme has a better performance than the analytic solution in general; however, the need for pilot symbols at the start of the communication might be a challenge for this method. Fortunately, the pilot symbols should be transmitted only once at the start of the communication. It is important to realize that the analytical solution has a desirable performance for low symbol durations, and has a similar (or better) performance for

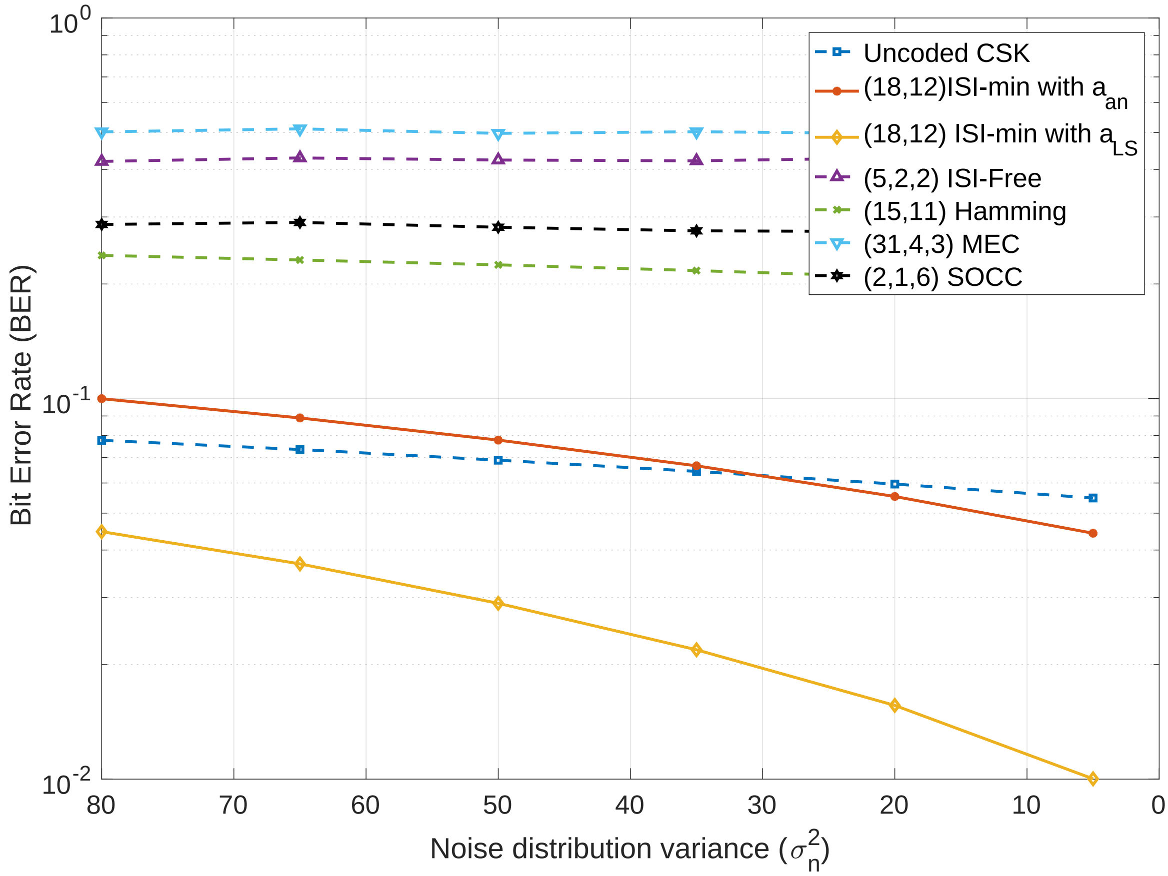

Figure 4.3. Comparison of different coding strategies and uncoded transmissions for $t_s = 200\,\text{ms},\ M = 500$.

high symbol durations without having any pilot symbols at the start.

The second observation is that our coding scheme has a clear advantage over other coding schemes, and our scheme performs better than the uncoded case even for low symbol durations, like $t_s = 200\,\text{ms}$. A more important result is that at low symbol durations and high transmitted number of molecules, while the other coding techniques cannot benefit from high power, due to their error floor, our scheme does not reach an error floor. In other words, a fast data rate communication with a considerably low probability of error is possible at high power with the proposed schemes.

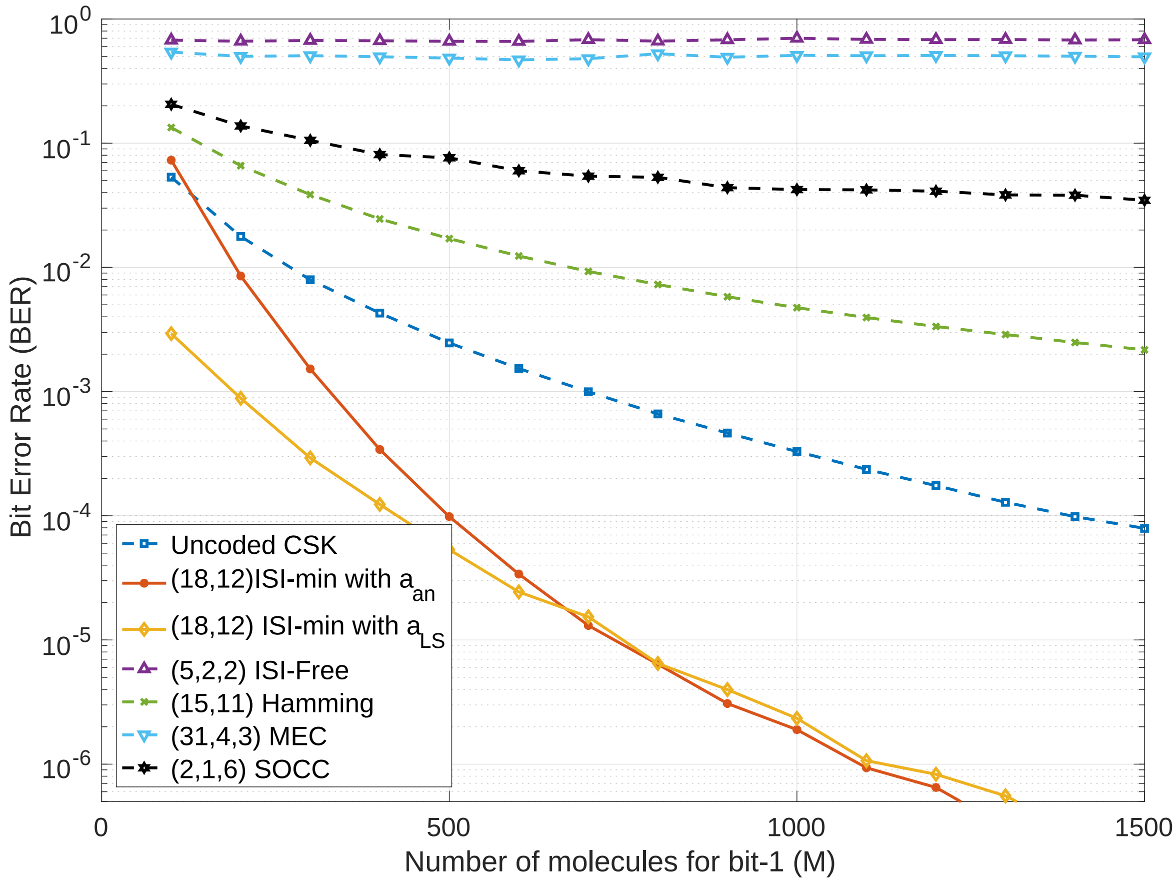

Figure 4.4. Comparison of different coding strategies and uncoded transmissions for $t_s = 300\,\text{ms}$, $\sigma_n^2 = 0$.

4.2.2. Experimental Results on a Macro-Scale MIMO Testbed

The experimental aspect of the MC gains significant interest in last years. One of the first testbed introduced in the literature can be found in [41]. This first testbed is at macro-scale and consist of an electrically controllable spray and a micro-controller at the transmitter side. At the receiver side, there is an alcohol sensor and a micro-controller. When bit-1 is sent, transmitter spray alcohol to the environment, and the receiver make a decision according to the density of the alcohol. In [42], the channel and the noise models for the introduced testbed is analyzed. The authors of [43], improve this testbed, by designing two transmitter two receiver multiple input multiple output system. The system still at macro-scale yet it is abilities and the date rate is significantly increased. The same system (two transmitter and two receiver) with the

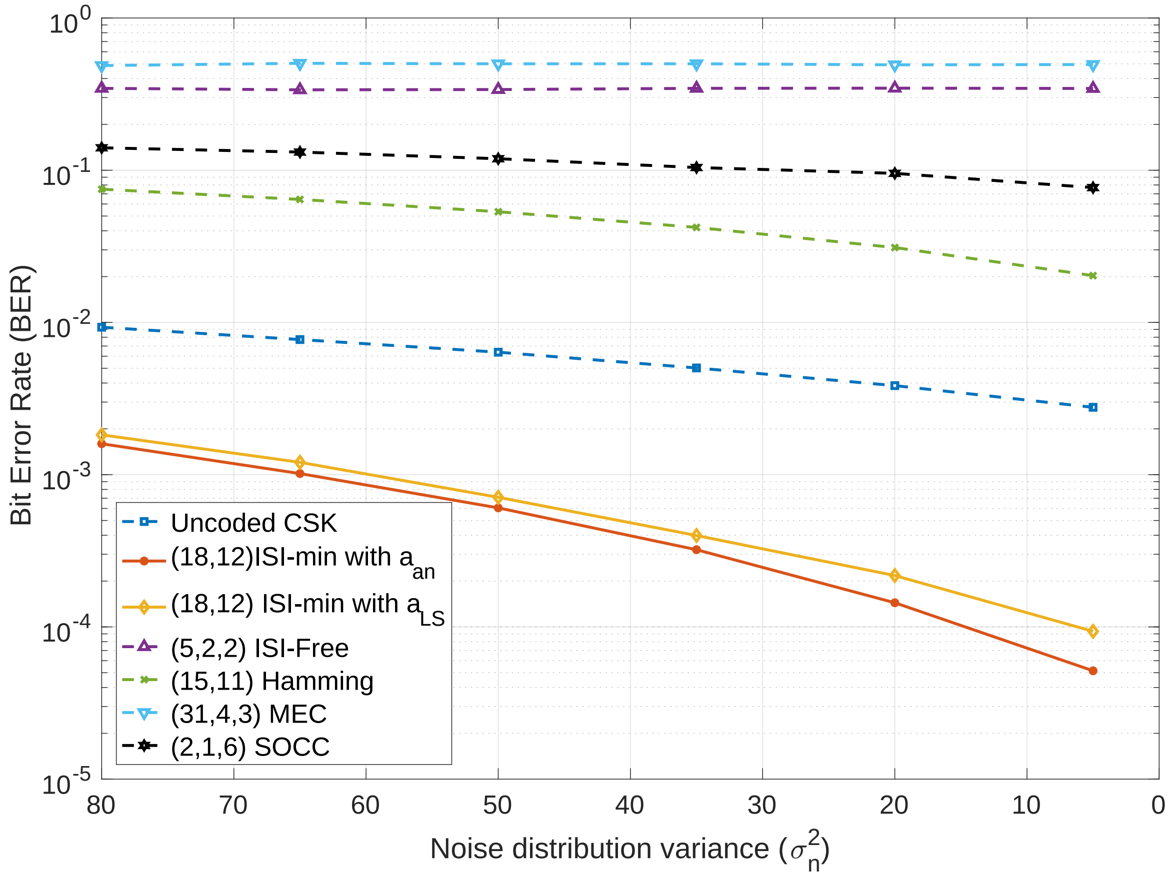

Figure 4.5. Comparison of different coding strategies and uncoded transmissions for $t_s = 300\,\text{ms},\ M = 500$.

addition of fans behind the transmitters is introduced by the same authors in [44]. Thanks to this addition, the authors manage to analyze the effect of drift.

To verify our proposed coding and detection scheme, we used the macro scale testbed presented in [43], which is shown in Fig. 4.8. The system mainly consists of a transmitter part and a receiver part. At the transmitter part, a microcontroller controls the two spray nozzles separately, and at the receiver side, two alcohol sensors are driven by two separate micro-controllers. Please note that the coding at the transmitter and the detection at the receiver are also handled by these micro-controllers. While transmitting a bit sequence, it is evenly divided into two-bit sequences that are sent in parallel channels by the two spray nozzles, and detected in parallel by alcohol detectors.

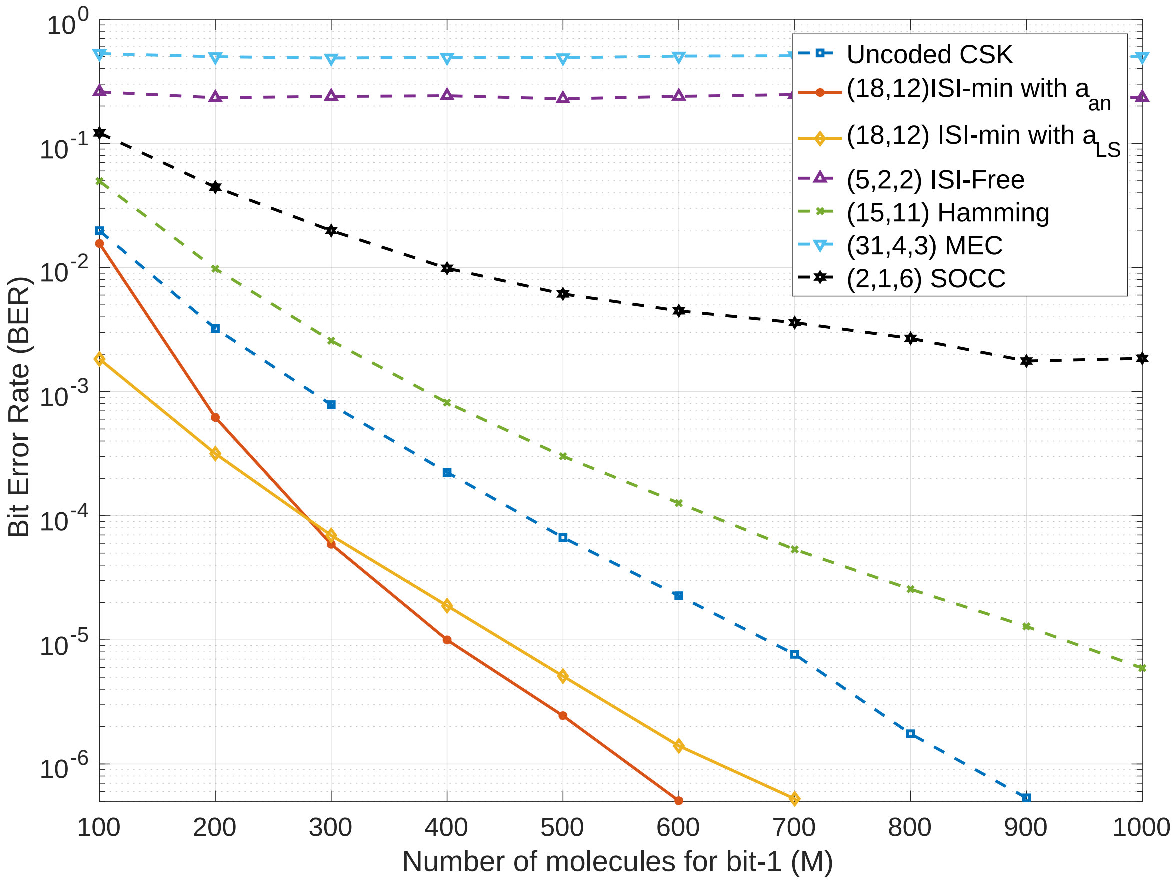

Figure 4.6. Comparison of different coding strategies and uncoded transmissions for $t_s = 400\,\text{ms}$, $\sigma_n^2 = 0$.

In [43], International Telegraph Alphabet No. 2 (ITA-2) is used to represent 26 different characters with five bits. Moreover, the authors propose three different detection algorithms: adaptive thresholding, practical zero forcing, and genie-aided zero-forcing. While genie-aided zero-forcing performs best, it is not practical as its name indicates. Authors show that the other the two detection algorithms perform almost the same. Hence, the adaptive algorithm is used with ITA-2 for comparison.

In our system, we use eight bits to represent 26 different characters. For eight bits, there are 33 different binary sequences that obey the rules of the proposed coding scheme. We choose 26 codewords that have fewer number of bit-1s. For detection, we send 100 pilot symbols and use LS solution to determine the threshold scaling constant a. Symbol duration $t_s = 4000\,\text{ms}$ is chosen for our system, and the average power

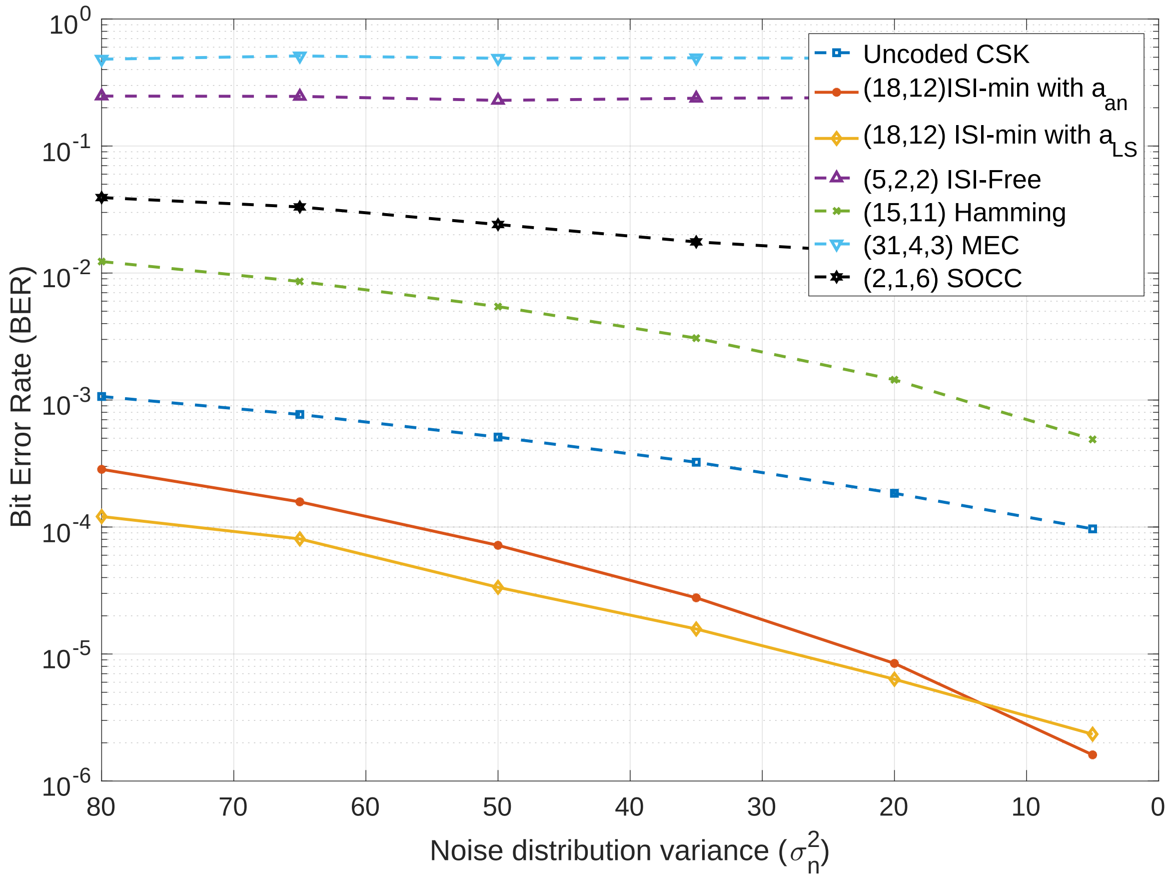

Figure 4.7. Comparison of different coding strategies and uncoded transmissions for $t_s = 400\,\text{ms},\ M = 500$.

and data rate are normalized as explained in Section 3. Since counting the number of released molecules is not possible in the testbed, we use spray duration to adjust the power. Character error rates (CER) for different distances between the transmitter and the receiver are given in Fig. 4.9, 4.10, and the results show that our scheme has a clear advantage in terms of CER.

As a final remark, please note that this testbed is in macro-scale. Which means, it is not exactly characterizing the conditions of a nano-scale system. There are several studies to implement a testbed for MC [45–48]. In particular, [45, 46] are very exciting and promising studies which are based on magnetic nano-particles. Even though their capability is limited for now, it will be eventually a great reference for the literature, since it is a testbed in nano-scale. Such different perspective and methods to analyze

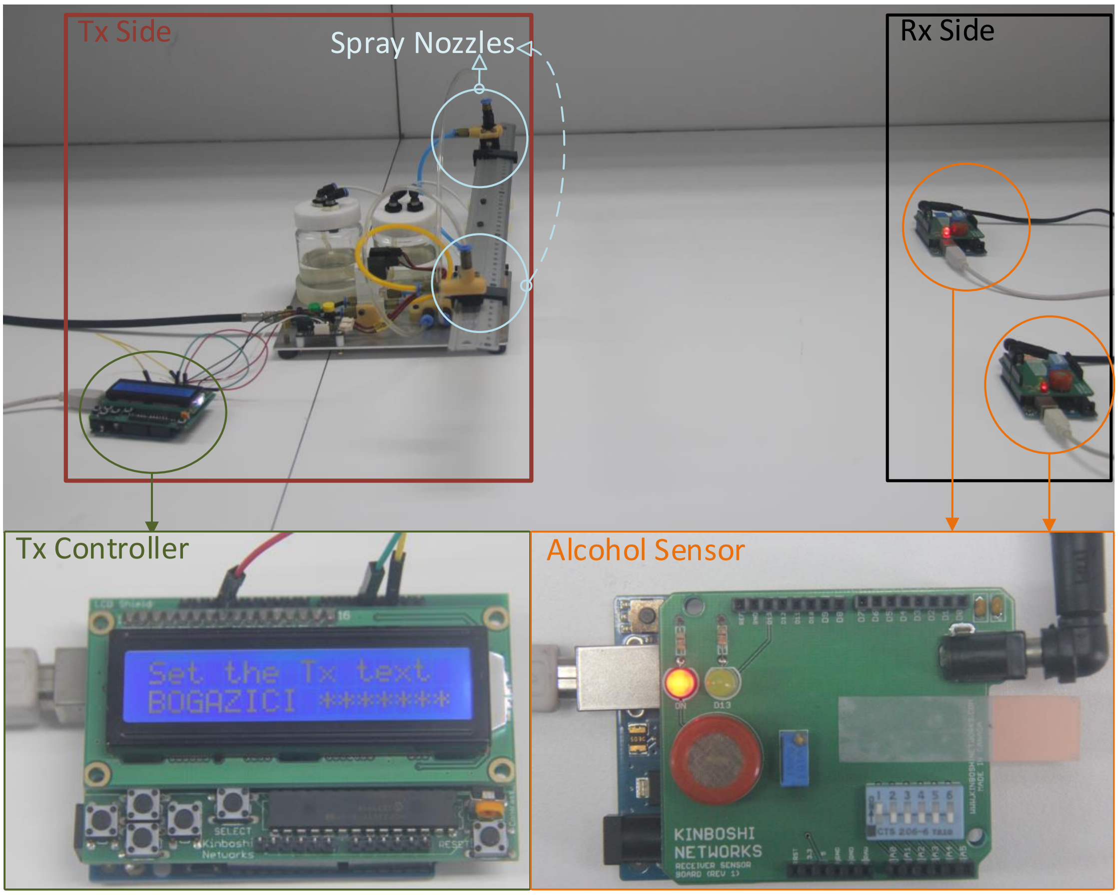

Figure 4.8. Testbed Structure.

nano-scale communication is definitely needed and should have more visibility. The downside of that testbed is that it is challenging implementation step. Reproducing such a testbed is definitely not a trivial task. In the future, we believe that, it will be more accessible.

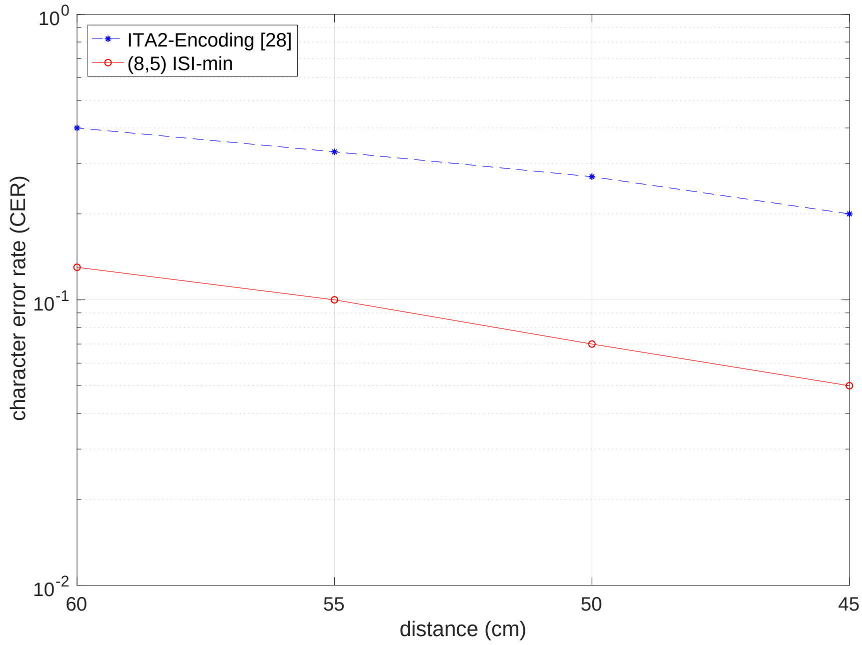

Figure 4.9. CER performances of the proposed scheme and the ITA-2 encoding scheme [43].

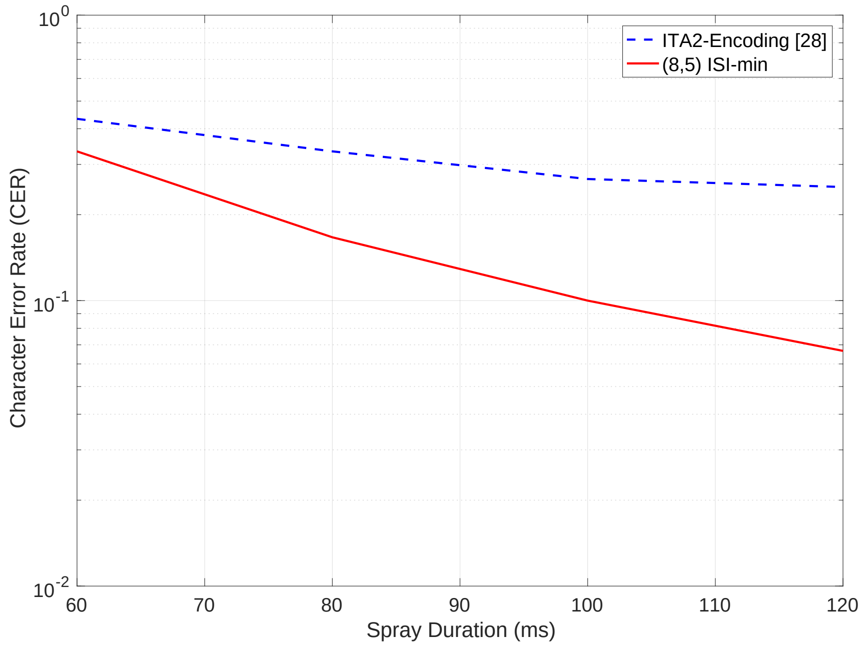

Figure 4.10. CER performances of the proposed scheme and the ITA-2 encoding scheme [43].

5. CONCLUSION

In this thesis, first, channel code design techniques for MCvD are proposed. The probability of the transition of a binary sequence to another is defined, and it is shown that this probability can be calculated analytically under certain conditions. Thereafter, the decoding regions are defined for all codewords according to the obtained transition probabilities, which performs better than the classical technique relying on the Hamming distance. Then, an analytical expression for the codeword error rate is found, and new channel code design algorithms are proposed. Three methods are proposed to design codebooks: greedy algorithm, genetic algorithm, and mixed integer programming solver. The results show that all of them perform better than the uncoded case for good channel conditions even with penalized symbol duration and average power due to the coding ratio. It is also shown that the genetic algorithm performs very close to mixed integer programming, and it is a feasible algorithm to design long channel codes. The encoder and the decoder of the non-linear codes produced by our proposed algorithms consist of lookup tables. As a future work, we plan to improve the proposed algorithms to design linear block codes to ease the use in nanomachines. As another future work, we also plan to tackle the impotency of IP for long block lengths due to time and memory limitations.

Second, a channel coding scheme and a detection technique that is specifically tailored for this coding scheme are proposed. To begin with, without having consecutive bit-1s, a new codebook is introduced. For detection purposes, the codebook consists of codewords that have at least one bit-1. After that, the steps to correct the received vectors are proposed so that the erroneously detected bits are corrected. Finally, a new detection algorithm, which takes advantage of the information of the maximum and minimum number of received molecules for bits of every transmitted codeword, is proposed. The results show that the proposed scheme has a significant improvement in terms of bit error rate. Finally, the proposed scheme is implemented on a macroscale testbed to show its practical use, and improvement in the character error rate. Due to the non-ideal structure of the testbed, the channel conditions are constantly changing with time. One of the most crucial aspect of ISI-min code family is that it does not require any a constant channel to work efficiently. Moreover, it also does not require any form of channel state information. Similar with the codebooks that are designed with the previously introduced methods, ISI-min codes are not a linear code family. Hence, the channel encoder and decoder can only be implementable with look-up tables. We will further study to find a method or algorithm for implementation of the encoder and decoder.

REFERENCES

1. Moore, M., A. Enomoto, T. Nakano, R. Egashira, T. Suda, A. Kayasuga, H. Kojima, H. Sakakibara and K. Oiwa, "A design of a molecular communication system for nanomachines using molecular motors", *Proc. Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, pp. 6 pp.–559, March 2006.

2. Cobo, L. C. and I. F. Akyildiz, "Bacteria-based communication in nanonetworks", *Nano Communication Networks*, Vol. 1, No. 4, pp. 244–256, 2010.

3. 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.*, pp. 478–481 vol. 2, July 2005.

4. Kuran, M. S., H. B. Yilmaz, T. Tugcu and B. Ozerman, "Energy model for communication via diffusion in nanonetworks", *Nano Communication Networks*, Vol. 1, No. 2, pp. 86 – 95, 2010.

5. Pierobon, M. and I. F. Akyildiz, "Diffusion-based noise analysis for molecular communication in nanonetworks", *IEEE Transactions on Signal Processing*, Vol. 59, No. 6, pp. 2532–2547, 2011.

6. Farsad, N., H. B. Yilmaz, A. Eckford, C. Chae and W. Guo, "A comprehensive survey of recent advancements in molecular communication", *IEEE Communications Surveys Tutorials*, Vol. 18, No. 3, pp. 1887–1919, thirdquarter 2016.

7. Shu, L. and D. J. Costello, "Error control coding, fundamentals and applications", *Prentice Hall*, 2003.

8. Einolghozati, A. and F. Fekri, "Analysis of error-detection schemes in diffusion-based molecular communication", *IEEE Journal on Selected Areas in Communi-*

cations, Vol. 34, No. 3, pp. 615–624, March 2016.

9. Lu, Y., M. D. Higgins and M. S. Leeson, "Diffusion based molecular communications system enhancement using high order Hamming codes", *Proc. 9th International Symposium on Communication Systems, Networks Digital Sign (CSNDSP)*, pp. 438–442, July 2014.

10. Dissanayake, M. B., Y. Deng, A. Nallanathan, E. M. N. Ekanayake and M. Elkashlan, "Reed Solomon codes for molecular communication with a full absorption receiver", *IEEE Communications Letters*, Vol. 21, No. 6, pp. 1245–1248, June 2017.

11. Dissanayake, M. B., Y. Deng, A. Nallanathan, M. Elkashlan and U. Mitra, "Enhancing the reliability of large-scale multiuser molecular communication systems", *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, June 2018.

12. Kovačević, M. and P. Popovski, "Zero-error capacity of a class of timing channels", *IEEE Transactions on Information Theory*, Vol. 60, No. 11, pp. 6796–6800, Nov 2014.

13. Abadi, N., A. A. Gohari, M. Mirmohseni and M. Nasiri-Kenari, "Zero-error codes for multi-type molecular communication in random delay channel", *2018 Iran Workshop on Communication and Information Theory (IWCIT)*, pp. 1–6, April 2018.

14. 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.

15. 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, Nov 2015.

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

17. Shih, P. J., C. H. Lee, P. C. Yeh and K. C. 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.

18. Akhkandi, P., A. Keshavarz-Haddad and A. Jamshidi, "A new channel code for decreasing inter-symbol-interference in diffusion based molecular communications", *Proc. 8th International Symposium on Telecommunications (IST)*, pp. 277–281, Sept 2016.

19. Qiu, S., T. Asyhari and W. Guo, "Mobile molecular communications: Positional-distance codes", *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, July 2016.

20. Lu, Y., M. D. Higgins and M. S. Leeson, "Self-orthogonal convolutional codes (SOCCs) for diffusion-based molecular communication systems", *2015 IEEE International Conference on Communications (ICC)*, pp. 1049–1053, June 2015.

21. Ko, P.-Y., Y.-C. Lee, P. C. Yeh, C. han Lee and K. C. Chen, "A new paradigm for channel coding in diffusion-based molecular communications: Molecular coding distance function", *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 3748–3753, Dec 2012.

22. Nakano, T., A. W. Eckford and T. Haraguchi, *Molecular communication*, Cambridge University Press, 2013.

23. Yilmaz, H. B., A. C. Heren, T. Tugcu and C.-B. Chae, "Three-dimensional channel characteristics for molecular communications With an absorbing receiver", *IEEE Communications Letters*, Vol. 18, No. 6, pp. 929–932, June 2014.

24. Yilmaz, H. B., C.-B. Chae, B. Tepekule and A. E. Pusane, "Arrival modeling and error analysis for molecular communication via diffusion with drift", *Proc. Nanoscale Computing and Communication (NANOCOM)*, ACM, 2015.

25. Einolghozati, A., M. Sardari and F. Fekri, "Capacity of diffusion-based molecular communication with ligand receptors", *2011 IEEE Information Theory Workshop*, pp. 85–89, Oct 2011.

26. Atakan, B. and O. B. Akan, "An information theoretical approach for molecular communication", *2007 2nd Bio-Inspired Models of Network, Information and Computing Systems*, pp. 33–40, Dec 2007.

27. Meng, L., P. Yeh, K. Chen and I. F. Akyildiz, "A diffusion-based binary digital communication system", *2012 IEEE International Conference on Communications (ICC)*, pp. 4985–4989, June 2012.

28. Srinivas, K. V., A. W. Eckford and R. S. Adve, "Molecular communication in fluid media: The additive inverse gaussian noise channel", *IEEE Transactions on Information Theory*, Vol. 58, No. 7, pp. 4678–4692, July 2012.

29. Nakano, T., Y. Okaie and J. Liu, "Channel model and capacity analysis of molecular communication with Brownian motion", *IEEE Communications Letters*, Vol. 16, No. 6, pp. 797–800, June 2012.

30. Genc, G., Y. E. Kara, H. B. Yilmaz and T. Tugcu, "ISI-aware modeling and achievable rate analysis of the diffusion channel", *IEEE Communications Letters*, Vol. 20, No. 9, pp. 1729–1732, Sep. 2016.

31. Kuran, M. S., H. B. Yilmaz, T. Tugcu and I. F. Akyildiz, "Modulation techniques for communication via diffusion in nanonetworks", *Proc. IEEE International Conference on Communications (ICC)*, pp. 1–5, June 2011.

32. Mahfuz, M. U., D. Makrakis and H. T. Mouftah, "A comprehensive study of

sampling-based optimum signal detection in concentration-encoded molecular communication", *IEEE Transactions on NanoBioscience*, Vol. 13, No. 3, pp. 208–222, Sept 2014.

33. Shi, L. and L. Yang, "Error performance analysis of diffusive molecular communication systems with on-off keying modulation", *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, pp. 1–1, 2018.

34. Tepekule, B., A. E. Pusane, M. S. Kuran and T. Tugcu, "A novel pre-Equalization method for molecular communication via diffusion in nanonetworks", *IEEE Communications Letters*, Vol. 19, No. 8, pp. 1311–1314, Aug 2015.

35. Costello, D. J. and G. D. Forney, "Channel coding: The road to channel capacity", *Proceedings of the IEEE*, Vol. 95, No. 6, pp. 1150–1177, June 2007.

36. Hamming, R. W., "Error detecting and error correcting codes", *Bell System Technical Journal*, Vol. 29, No. 2, pp. 147–160, 1950, `https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1950.tb00463.x`.

37. Proakis, J. G., M. Salehi, N. Zhou and X. Li, *Communication systems engineering*, Vol. 2, Prentice Hall New Jersey, 1994.

38. Kışlal, A. O., A. E. Pusane and T. Tuğcu, "A comparative analysis of channel coding for molecular communication", *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, May 2018.

39. Gonzalez, T. F., "Clustering to minimize the maximum intercluster distance", *Theoretical Computer Science*, Vol. 38, pp. 293–306, 1985.

40. 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.

41. Farsad, N., W. Guo and A. W. Eckford, “Tabletop molecular communication: text messages through chemical signals”, *PLOS ONE*, Vol. 8, No. 12, pp. 1–13, 12 2013.

42. Farsad, N., N. Kim, A. W. Eckford and C. Chae, “Channel and noise models for nonlinear molecular communication systems”, *IEEE Journal on Selected Areas in Communications*, Vol. 32, No. 12, pp. 2392–2401, Dec 2014.

43. Koo, B., C. Lee, H. B. Yilmaz, N. Farsad, A. Eckford and C.-B. Chae, “Molecular MIMO: from theory to prototype”, *IEEE Journal on Selected Areas in Communications*, Vol. 34, No. 3, pp. 600–614, March 2016.

44. Lee, C., B. Koo, N.-R. Kim, H. B. Yilmaz, N. Farsad, A. Eckford and C.-B. Chae, “Demo: molecular MIMO with drift”, *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom ’15, pp. 201–203, ACM, 2015.

45. Unterweger, H., J. Kirchner, W. Wicke, A. Ahmadzadeh, D. Ahmed, V. Jamali, C. Alexiou, G. Fischer and R. Schober, “Experimental molecular communication testbed based on magnetic nanoparticles in duct flow”, *CoRR*, 2018.

46. Wicke, W., A. Ahmadzadeh, V. Jamali, H. Unterweger, C. Alexiou and R. Schober, “magnetic nanoparticle based molecular communication in microfluidic environments”, *CoRR*, Vol. abs/1808.05147, 2018, `http://arxiv.org/abs/1808.05147`.

47. Shakya, P., E. Kennedy, C. Rose and J. K. Rosenstein, “Correlated transmission and detection of concentration-modulated chemical vapor plumes”, *IEEE Sensors Journal*, Vol. 18, No. 16, pp. 6504–6509, Aug 2018.

48. McGuiness, D. T., S. Giannoukos, A. Marshall and S. Taylor, “Experimental results on the open-air transmission of macro-molecular communication using membrane inlet mass spectrometry”, *IEEE Communications Letters*, Vol. 22, No. 12, pp. 2567–2570, Dec 2018.