

# ISI-aware Channel Code Design for Molecular Communication via Diffusion

A. Oguz Kislal, *Student Member, IEEE*, H. Birkan Yilmaz, *Member, IEEE*, Ali E. Pusane, *Member, IEEE*, Tuna Tugcu, *Member, IEEE*

**Abstract**—In molecular communication via diffusion, information molecules diffusing in the environment are subject to Brownian motion. Due to probabilistic propagation, the arrival of the 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 paper, 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. The results show that the proposed methods provide significant improvements in performance in terms of codeword error rate.

**Index Terms**—Molecular communications, nanonetworking, diffusion channel, ISI-aware channel coding

## I. INTRODUCTION

MOLECULAR communications (MC) has been gaining more popularity as a promising approach to converge the recent advancements in the nanotechnology and biology domains to enable future health applications. In the molecular communications literature, there are numerous different means of transferring information, including communication via calcium ions [1], micro-tubules [2], bacteria [3], or diffusion [4], [5]. Specifically, molecular communication via diffusion (MCvD) is one of the most prominent means of carrying information between nanomachines [4], which are possibly (engineered) living cells. In the literature, one can observe that many features of the messenger molecules (MMs) are used for communication, including the concentration or type of the molecule, and sometimes a combination [4], [6].

The probabilistic movement pattern of the MMs subject to Brownian motion causes inter-symbol interference (ISI), since some of the MMs released for a specific symbol take a longer path to the destination nanomachine, arriving during

the following symbol durations. Such delayed MMs may easily cause incorrect symbol detection.

All communication systems incur the possibility of incorrect detection at the receiver as a consequence of signal distortion in the communication channel. Several different approaches have been developed to combat erroneous reception of information. Channel coding addresses this problem, by adding redundancy to the transmitted information and by utilizing the redundant information at the receiver to recover errors due to the channel. Although many channel codes have been proposed for a wide variety of applications over the years [7], unfortunately most of them are not suitable for MCvD due to the memory property of the MCvD channel.

In the literature, either some well-known channel codes are considered or new codes are designed for MC. In [8], error-detection schemes are analyzed, and the trade-off between the rate and the probability of error is shown for MC. The bit error rate (BER) performance and energy consumption of the encoder and decoder for Hamming codes have been considered in [9], where the authors show that there is an optimal number of parity bits for which the coding gain is maximized. Reed-Solomon (RS) codes have also been considered for MCvD in [10]. This study shows that RS codes achieve better coding gain than Hamming codes. However, their decoder complexity is much higher than that of Hamming codes. RS codes are also considered for large-scale MC in [11], and the authors show that RS coding improves the BER performance of such systems. Zero-error capacity of the MC channel and how to construct a zero-error code are discussed in [12] when a single type of molecule is used. In [13], the authors extend this work and introduce an upper bound on the zero-error capacity and a method to construct zero-error codes when multiple types of molecules are available for transmission. Minimum energy channel codes are investigated in [14], where the authors show that both the performance and the average energy per bit consumption are improved, compared to Hamming codes when the channel memory is very short (two taps). 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. ISI-free codes for MC are introduced in [16], [17]. The latter designs two families of these codes whilst also focusing on achieving low decoding complexity. ISI-free codes consider the ISI caused by the previous codewords as the main source of error, and they are proposed for low data rates. For higher data rates, the ISI within the codeword becomes important, and the error performance of ISI-free codes degrades. Another channel code

A. O. Kislal and A. E. Pusane are with the Dept. of Electrical and Electronics Engineering, Bogazici University, Istanbul, Turkey. e-mail: {ahmet.kislal, ali.pusane}@boun.edu.tr

H. B. Yilmaz is with the Dept. of Network Engineering, Polytechnic University of Catalonia, 08034 Barcelona, Spain. e-mail: birkan.yilmaz@upc.edu

T. Tugcu is with NETLAB, Dept. of Computer Engineering, Bogazici University, Istanbul, Turkey. e-mail: tugcu@boun.edu.tr

This work is partially supported by the Government of Catalonia's Secretariat for Universities and Research via the Beatriu de Pinos postdoctoral programme, the Scientific and Technical Research Council of Turkey (TUBITAK) under grant number 116E916, and the State Planning Organization (DPT) of Turkey under TETAM (2007K120610).

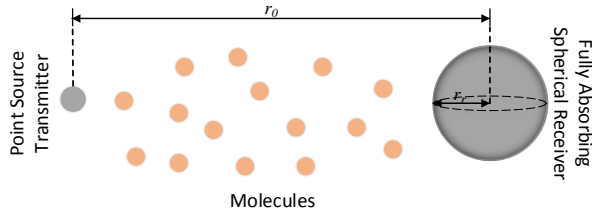


Fig. 1. MCvD system model.

family, crossover resistant coding with time gap (CRCTG), is introduced for MC to improve the error performance [18]. In [19], the BER performance of positional-distance code is compared with classical Hamming-distance coding strategies on mobile robots equipped with MC transceivers, and the results show that positional-distance codes have a clear advantage. The authors of [20] compare the coding gain of self-orthogonal convolutional codes (SOCC) with the Hamming codes, and show that SOCC have a higher coding gain when the transmitter and the receiver are very close to each other.

As a final remark, due to the heavy ISI, it is claimed that Hamming distance is far from being the optimal metric for channel code design in MC. The authors of [21] introduce a new distance function, called molecular coding (MoCo) distance, for MC. It is shown that, just by modifying the decoding region of Hamming codes according to the introduced distance function, the performance can be improved gradually.

In this paper, we mainly aim to introduce new methods for designing efficient channel codes for MC. In this respect, we first introduce a slightly modified version of MoCo distance and consider a more realistic system model than the one considered in [21]. Then, the average error probability for a selected codebook is obtained. With these foundations, a new paradigm and a greedy algorithm are introduced to design efficient channel codes. Moreover, optimization methods are considered and a genetic algorithm is introduced to achieve better performance. Finally, the problem is modeled as a linear optimization problem and solved with mixed integer programming (MIP) solver to find the optimum codebook designs. Error performances of the designed codebooks are obtained to demonstrate the superior performance of codebooks.

## II. SYSTEM MODEL

In this work, diffusion based communication systems are considered in a three dimensional space. Moreover, it is assumed that a point transmitter and a fully absorbing spherical receiver are at a fixed position and are communicating via MCvD as shown in Fig. 1.

The MMs move according to Brownian motion while they diffuse randomly in the fluid environment [22]. Each molecule's position can be modeled as a time dependent random variable. For a time step of size  $\Delta t$ , the movement in a dimension is independent from the movement in other dimensions and can be modeled with the normal distribution. The updated position of a molecule after  $\Delta t$  seconds is

$$\begin{aligned} \vec{r}(t + \Delta t) &= \vec{r}(t) + (\Delta r_1, \Delta r_2, \Delta r_3) \\ \Delta r_i &\sim \mathcal{N}(0, 2D\Delta t) \end{aligned} \quad (1)$$

where  $\vec{r}(t)$  is the location vector of a molecule,  $\Delta r_i$  is the displacement in  $i$ -th dimension,  $D$  is the diffusion coefficient, and  $\mathcal{N}(\mu, \sigma^2)$  stands for the normal distribution with mean  $\mu$  and variance  $\sigma^2$ . In this work, the movements of the MMs are assumed to be independent from each other.

The molecules released from the transmitter start to diffuse in the environment. As time advances, some of the molecules are absorbed by the receiver and others are not. For a molecule in transition, the absorption probability by the receiver until time  $t$  is given as

$$F_{hit}(t) = \frac{r_r}{r_0} \operatorname{erfc}\left(\frac{r_0 - r_r}{\sqrt{4Dt}}\right), \quad (2)$$

where  $r_0$  is the distance between the point transmitter and the center of the receiver,  $r_r$  stands for the diameter of the receiver,  $t$  is the duration after the release of the molecule, and finally  $\operatorname{erfc}(\cdot)$  is the complementary error function [23]. So, the channel coefficients can be found by utilizing (2) as

$$p_k = F_{hit}(kt_s) - F_{hit}((k-1)t_s), \quad k = 1, 2, \dots, I \quad (3)$$

where  $t_s$  is the chosen symbol duration and  $I$  is the channel memory. Please note that  $p_1$  corresponds to the probability that a molecule is absorbed by the receiver in its current symbol duration. More specifically,  $p_k$  stands for the probability that a molecule is absorbed by the receiver in the  $k$ -th symbol slot starting from the slot in which it was released. If  $M$  molecules are released from the transmitter without any successive transmissions, the number of received molecules at the end of the first symbol slot ( $N_1^{Rx}$ ) can be modeled as

$$N_1^{Rx} \sim B(M, p_1), \quad (4)$$

where  $B(n, p)$  is the binomial distribution with  $n$  trials and the success probability  $p$ . When we have consecutive transmissions, we denote the number of received molecules during the  $i$ -th symbol duration with  $N_i^{Rx}$  and denote the number of molecules that are emitted at the start of the  $k$ -th symbol slot and absorbed during the  $i$ -th symbol slot as  $N_{i,k}^{Rx}$ . Under this condition, the number of absorbed molecules in the  $i$ -th symbol duration can be obtained as

$$N_i^{Rx} = \sum_{k=1}^i N_{i,k}^{Rx}. \quad (5)$$

Please note that  $N_{i,k}^{Rx}$  is distributed according to a binomial distribution  $B(M_k, p_{i-k+1})$ , where  $M_k$  is the number of emitted molecules at the start of the  $k$ -th symbol duration. For  $M > 50$ , this binomial distribution can be approximated by a normal distribution [24] as

$$N_i^{Rx} \sim \mathcal{N}\left(\sum_{k=1}^i M_k p_{i-k+1}, \sum_{k=1}^i M_k p_{i-k+1} (1 - p_{i-k+1})\right). \quad (6)$$

The number of absorbed molecules is directly related to the channel coefficients ( $p_i$  values) as shown in (6). Moreover, since the position of the transmitter and the receiver, and the receiver's diameter are fixed,  $p_i$  values are only affected by the symbol duration,  $t_s$ , which is shown in (3). As  $t_s$  increases, more molecules are absorbed and the channel causes less ISI, but the data rate decreases at the same time. On the contrary,

as  $t_s$  decreases, the data rate is increased, but ISI also increases and the quality of communication is degraded.

Finally, binary concentration shift keying (BCSK) is used due to its wide use in the literature [25]–[28]. The transmitter releases  $M$  molecules at the beginning of the symbol duration to send bit-1, while it does not release any molecules for bit-0. At the receiver side, if the number of absorbed molecules at the end of the symbol duration exceeds a pre-determined threshold, bit-1 is detected; otherwise bit-0 is detected.

### III. CODE DISTANCE AND DECODING REGION

In channel codes,  $k$ -bit information sequences are mapped to  $n$ -bit sequences that are called codewords, and are shown as  $C_i, i = 1, 2, \dots, 2^k$ . Then, the codewords are transmitted using the chosen modulation technique and detected at the receiver. Due to the adverse effects of the channel, the detected sequence can be different from the transmitted codeword; it might even not match any other codeword. Hence, the received  $n$ -bit sequence is called the received vector, whose elements are denoted by  $w_j, j = 1, 2, \dots, 2^n$ . After detection, the decoder maps the received vector back to the most likely codeword. Then, the codeword is mapped back to the corresponding  $k$ -bit information sequence (Fig. 2).

#### A. Code Distance in Molecular Communication

In RF communications, Hamming distance, which is the number of different bits between two binary sequences, is effectively used to design channel codes. For example, a Hamming code with message length  $k = 4$  and block length  $n = 7$  (shown as  $(n = 7, k = 4)$  or  $(7, 4)$ ) has a minimum Hamming distance of  $d_H = 3$  between its codewords. This implies that if  $\lfloor (d_{Hmin} - 1)/2 \rfloor = 1$  bit out of  $n = 7$  bits is detected erroneously, it can be corrected. The minimum Hamming distance is directly related to the error probability for the conventional additive white Gaussian noise (AWGN) channel. While detecting a binary sequence in the AWGN channel, bit errors are assumed to be independent, and thus the Hamming distance between the original codeword and the received vectors are highly correlated, which is different in the case of MC, where bit errors depend on the value of subsequent bits. Hence, Hamming distance is not a good choice for designing MC codes as demonstrated in [18].

For an  $(n, k)$  code family, there are  $2^k$  codewords of length  $n$  bits, which are selected out of  $2^n$  possibilities. In other words, we select  $2^k$  codewords out of  $2^n$  candidate binary sequences to design the codebook. The basis for the channel code design depends on the probability  $P_{i,j} = P(w_j|w_i)$ , where  $w_i$  and  $w_j$  are the transmitted and detected binary sequences, respectively, with  $i, j = 1, 2, \dots, 2^n$ , and the probability transition matrix is given as

$$\mathbf{P} = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,2^n} \\ P_{2,1} & P_{2,2} & \dots & P_{2,2^n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{2^k,1} & P_{2^k,2} & \dots & P_{2^k,2^n} \end{bmatrix}. \quad (7)$$

Calculating  $P_{i,j}$  for all  $i, j$  is a challenging task. It is only possible to fill the P matrix with exhaustive simulations.

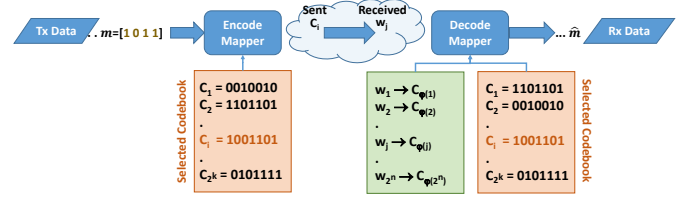


Fig. 2. The structure of channel coding for a scenario with  $(7, 4)$ -coding. The transmitter wants to send the message  $m$ , which is four bits in the given example. The encoder maps  $m$  to  $C_i$ , which is one of the  $2^k$  (i.e., 16 for the example scenario) codewords of length  $n$  (i.e., 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  $m$ .

However, assuming that the channel is cleaned after every codeword transmission,  $P_{i,j}$  can be calculated analytically.<sup>1</sup>

For the derivation of the  $P_{i,j}$  values, firstly, the received number of molecules is represented with a Gaussian random variable, as in (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

$$P_{el,i} = \begin{cases} Q\left(\frac{\lambda - u_{l,i}}{\sigma_{l,i}}\right) & b_{l,i} = 0, \\ Q\left(\frac{u_{l,i} - \lambda}{\sigma_{l,i}}\right) & b_{l,i} = 1, \end{cases} \quad (8)$$

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 which can be evaluated by utilizing (6),  $b_{l,i}$  is the binary indicator for the  $l$ -th bit of the  $i$ -th binary sequence,  $\lambda$  is the pre-determined threshold, and  $Q(\cdot)$  is the tail distribution function of the standard distribution. Therefore, the probability of correct detection is  $P_{cl,i} = 1 - P_{el,i}$ , and finally

$$P_{i,j} = \prod_{l=1}^n O_l^{i,j}, \quad (9)$$

$$O_l^{i,j} = \begin{cases} P_{cl,i} & b_{l,i} = b_{l,j} \\ P_{el,i} & b_{l,i} \neq b_{l,j} \end{cases}.$$

#### B. 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

$$\mathbf{C} = \begin{bmatrix} C_{1,1} & C_{1,2} & \dots & C_{1,2^n} \\ C_{2,1} & C_{2,2} & \dots & C_{2,2^n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{2^k,1} & C_{2^k,2} & \dots & C_{2^k,2^n} \end{bmatrix}, \quad (10)$$

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

<sup>1</sup>This assumption can be justified when two different types of molecules are used to send consecutive codewords so that the ISI of last codeword can be neglected. Also note that, choosing two different molecules with similar Stoke's radii would make their diffusion coefficient almost the same assuming that the environmental conditions are not changing [29].

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

$$\hat{C}_j = \arg \max_i (C_{i,j}), \forall j. \quad (11)$$

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

$$CER_i = \sum_{j=1, \hat{C}_j \neq i}^{2^n} C_{i,j}, \quad (12)$$

and the average CER can be obtained as

$$CER = \frac{\sum_i^{2^k} CER_i}{2^k}. \quad (13)$$

#### IV. CODEBOOK DESIGN

Designing a channel code for MC can be considered to be identical to the selection of  $2^k$  codewords among  $2^n$  candidate binary sequences. When the codebook is determined, the decoding region and CER can be easily found from (11) and (13), respectively. However, there are  $\binom{2^n}{2^k}$  combinations for the selection, and even for short codes (small  $n$  and  $k$ ) the number of possible codebooks is huge. Hence, using a brute-force algorithm is off the table and some heuristics should be considered to find efficient and effective solutions. We propose three different methods: a greedy algorithm, a genetic algorithm, and a mixed integer programming model. The greedy algorithm is the least computationally complex approach, its performance is also the worst. The genetic algorithm has the advantage of searching the sample space with great depth but it is much more time consuming than the greedy algorithm. Finally, MIP is the most computationally complex one among the three methods, but the optimal solution is guaranteed given enough time and memory resources.

##### A. Greedy Algorithm

Greedy algorithms are based on iteratively selecting the best option at each step. However, selecting the best option step-by-step can lead to getting stuck at local maxima. Yet, greedy algorithms still have their own use, since they are straightforward, easy to implement, and computationally cheap.

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. This problem is widely studied in the literature, and many heuristics are available. However, most of them propose solving this problem in the Euclidean space. Due to the transition probability matrix  $\mathbf{P}$  not being symmetrical and not necessarily satisfying the triangular inequality, these solutions would not perform as desired. A slightly modified version of the heuristic introduced in [30] is given in Algorithm 1. Please note that the decoding probability of the selected and candidate codewords should be considered in both directions

---

#### Algorithm 1: The heuristic for codeword selection [30]

---

**Data:** a transition probability matrix  $\mathbf{P}$ , block and message bit length  $(n, k)$

**Result:**  $CW$

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

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

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

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

let  $\mathbf{D} \leftarrow (1 - \mathbf{P}) + (1 - \mathbf{P})'$  be the 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**

**end**

**end**

---

(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. Therefore,  $\mathbf{D}$  is used in Algorithm 1 as a distance matrix instead of  $1 - \mathbf{P}$ . Unfortunately, the considered heuristic could not design codebooks with desirable performance. The solution does not even surpass the Hamming codes (that are not initially designed for MC). The problem is that the algorithm is neglecting the effect of the decoding region. In other words, not only the codewords should be scattered, but also the decoding regions for each codeword should be as far away from each other as possible. The challenging part is that the decoding region cannot be determined without determining all the codewords. To overcome this issue, we propose to predict the possible decoding regions for each binary sequence and then perform codeword selection.

For a better understanding of the decoding region concept, one can consider any  $(n, k)$  Hamming code as an example. Each codeword has  $2^{n-k}$  binary sequences in its decoding region, and the codewords are selected in a manner that decoding regions do not intersect with each other. Since the channel is binary symmetric, predicting which decoding region the binary sequence belongs to is trivial; the closest (according to the Hamming distance)  $2^{n-k}$  binary sequences would be the decoding region for that codeword. A similar approach may be used for MC. However, in general, each codeword may have a different number of elements in its decoding region; hence, a modification is needed. Instead of assuming a fixed number of binary sequences in the decoding region, we propose determining a threshold to predict whether a binary sequence is in the decoding region of a particular codeword or not. Clearly, this is only an approximation and is not optimal.

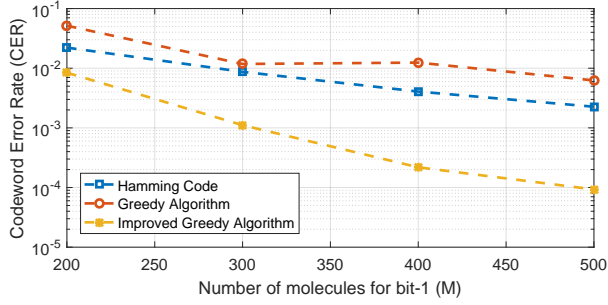


Fig. 3. Greedy algorithm and Hamming code performance analysis with  $(n, k) = (7, 4)$ ,  $K = 0.03$ ,  $t_s = 400$  ms,  $D = 79.4 \times 10^{-12} \mu\text{m}^2/\text{s}$ ,  $r_r = 5 \mu\text{m}$ , and  $r_0 = 10 \mu\text{m}$ .

The threshold is determined as

$$\gamma = \frac{\sum_{i=1}^{2^n} P_{i,i} K}{2^n}, \quad (14)$$

where  $P_{i,i}$  stands for the  $i$ -th diagonal element of  $\mathbf{P}$  which is defined in (7), and  $K$  is a scaling constant that determines the size of clusters in the greedy heuristic. The predicted decoding region is determined as

$$N_{i,j} = \begin{cases} 1, & P_{i,j} \geq \gamma, \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

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

$$\tilde{P}_{i,j} = \sum_{k=1}^{2^n} N_{j,k} P_{i,k}, \quad (16)$$

and  $\tilde{\mathbf{P}}$  is a matrix with  $\tilde{P}_{i,j}$  as the  $i, j$ -th element. Using  $\tilde{\mathbf{P}}$  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  $\lambda$  are given in Fig. 3. The results demonstrate that the proposed heuristic does not perform well, and the results do not even consistently improve with the number of molecules when using  $\mathbf{P}$ . However, the performance gradually improves when  $\tilde{\mathbf{P}}$  is used. The authors have tried steepest ascent climbing and simulated annealing methods, but have observed that they do not provide a significant improvement. The results of the mentioned approaches are not provided here for the sake of brevity of the manuscript.

## B. Genetic Algorithm

Genetic algorithm (GA) is an optimization method that is inspired by evolution. 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

---

## Algorithm 2: K-means clustering inspired heuristic

---

**Data:** a transition probability matrix  $\mathbf{P}$ , an iteration limit

$IterLim$

**Result:**  $NewCWS$

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**

$CrntCWS \leftarrow NewCWS$

**end**

---

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.

1) *Initial Population:* 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 Algorithm 2 and 3, respectively.

2) *Genetic Operations:* 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  $p_\mu$ . 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. 4.

## C. Mixed Integer Programming

Integer programming (IP) stands for the mathematical optimization problem with constraints that are either linear, or

**Algorithm 3:** Non-intersected decoding region search

---

**Data:** a predicted decoding region  $\mathbf{N}$   
**Result:**  $CW = CW_1, CW_2, \dots, CW_{2^k}$   
 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  $\mathbf{N}$  with greater  $\gamma$   
     restart the algorithm with new  $\mathbf{N}$   
   **end**  
**end**

---

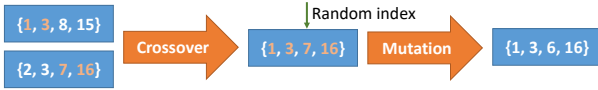


Fig. 4. Representative genetic operations where the selected codewords are represented by numbers. For the crossover operation, 1 and 3 are selected from the first parent while 7 and 16 are selected from the second parent. For the mutation operation, the third gene is replaced with the best option locally.

integer constraints with a linear objective function together with integer variables. If some of the variables are continuous, then it is called a 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.

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 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 (13) (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

$$CL_i \in \{0, 1\}, \quad (17)$$

$$CM_{i_1, i_2} \in \{0, 1\}, \quad (18)$$

$$\sum_i CL_i = 2^k, \quad (19)$$

$$\sum_{i_1} CM_{i_1, i_2} = 1, \quad (20)$$

$$\sum_{i_2} CM_{i_1, i_2} \leq CL_{i_1} 2^n, \quad (21)$$

$$\sum_{i_2} CM_{i_1, i_2} \geq CL_{i_1}, \quad (22)$$

$$PoCT = \frac{\sum_{i_1} \sum_{i_2} CM_{i_1, i_2} P_{i_1, i_2}}{2^k} \quad (23)$$

where  $i$ ,  $i_1$ , and  $i_2 \in [1, 2, \dots, 2^n]$  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 (19) states that there are  $2^k$  coset leaders to be selected. Condition (20) 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  $\mathbf{CM}$  has to be filled with zeros. This condition is stated with (21). Equation (22) 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 (23), which is the probability of correct transmission and maximizing it is the objective. For processing time purposes, equation (18) can be restated as

$$0 \leq CM_{i_1, i_2} \leq 1. \quad (24)$$

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 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. 5. 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.

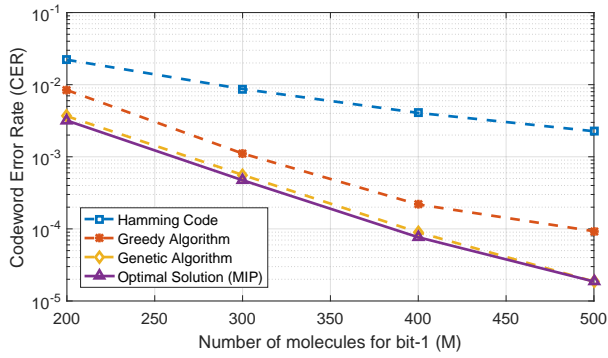


Fig. 5. CER comparison for all considered methods and Hamming code with  $(n, k) = (7, 4)$ ,  $t_s = 400$  ms,  $D = 79.4 \times 10^{-12} \mu\text{m}^2/\text{s}$ ,  $r_r = 5 \mu\text{m}$ ,  $K = 0.03$ , and  $r_0 = 10 \mu\text{m}$ .



Fig. 6. Three different messages are encoded by the codebook constructed with MIP for  $t_s = 400$  ms,  $M = 400$ ,  $(n, k) = (10, 7)$ .

## V. 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 II, 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

$$\hat{N}_i^{Rx} = N_i^{Rx} + \mathcal{N}(0, \sigma_n^2), \quad (25)$$

where  $\sigma_n^2$  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  $\sigma_n^2$  as a metric is more insightful than SNR.

The simulation parameters are given in Table I, and an example of encoding and decoding process is given in Fig. 6. The performance evaluation of the designed codebooks is obtained using (13). 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  $\lambda$  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  $\lambda$  is obtained when any binary

TABLE I  
SIMULATION PARAMETERS

Parameter	Variable	Value
Diffusion Coefficient	$D$	79.4, 61 $\mu\text{m}^2/\text{s}$ [4], [29]
Receiver Radius	$r_r$	5 $\mu\text{m}$
Tx-Rx distance	$r_0$	10 $\mu\text{m}$
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	$\sigma_n^2$	80 - 0
Message and block length	$(n, k)$	(10, 7)

sequence can be transmitted from the transmitter; however, only codewords are transmitted when coding is used. What should be done is to recalculate  $\lambda$  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  $\lambda$  value. Not only this procedure is time consuming, but also the performance gain is negligible. Hence,  $\lambda$  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 simulation purposes only. Finally, the parameters that are given in Table I 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<sup>2</sup> [17],  $(31, 4, 3)$  minimum energy code [14],  $(2, 1, 6)$  SOCC<sup>3</sup> [20],  $(15, 11)$  and  $(7, 4)$  Hamming codes, the proposed solutions, and the uncoded case for different symbol durations,  $D$ , and  $\sigma_n^2$  are given in Fig. 8 and Fig. 7. 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) 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.

<sup>2</sup>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.

<sup>3</sup>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.

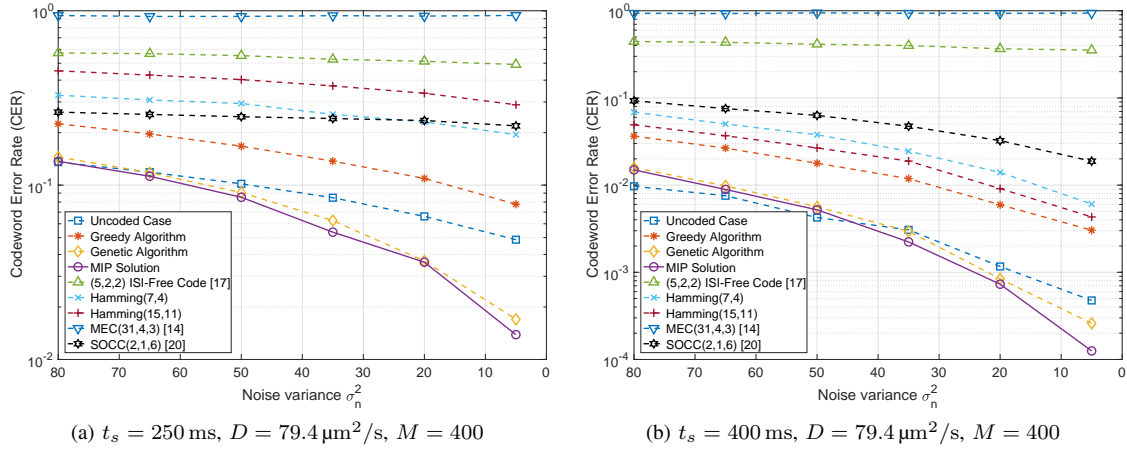


Fig. 7. CER of the coded and uncoded transmissions in the presence of system noise

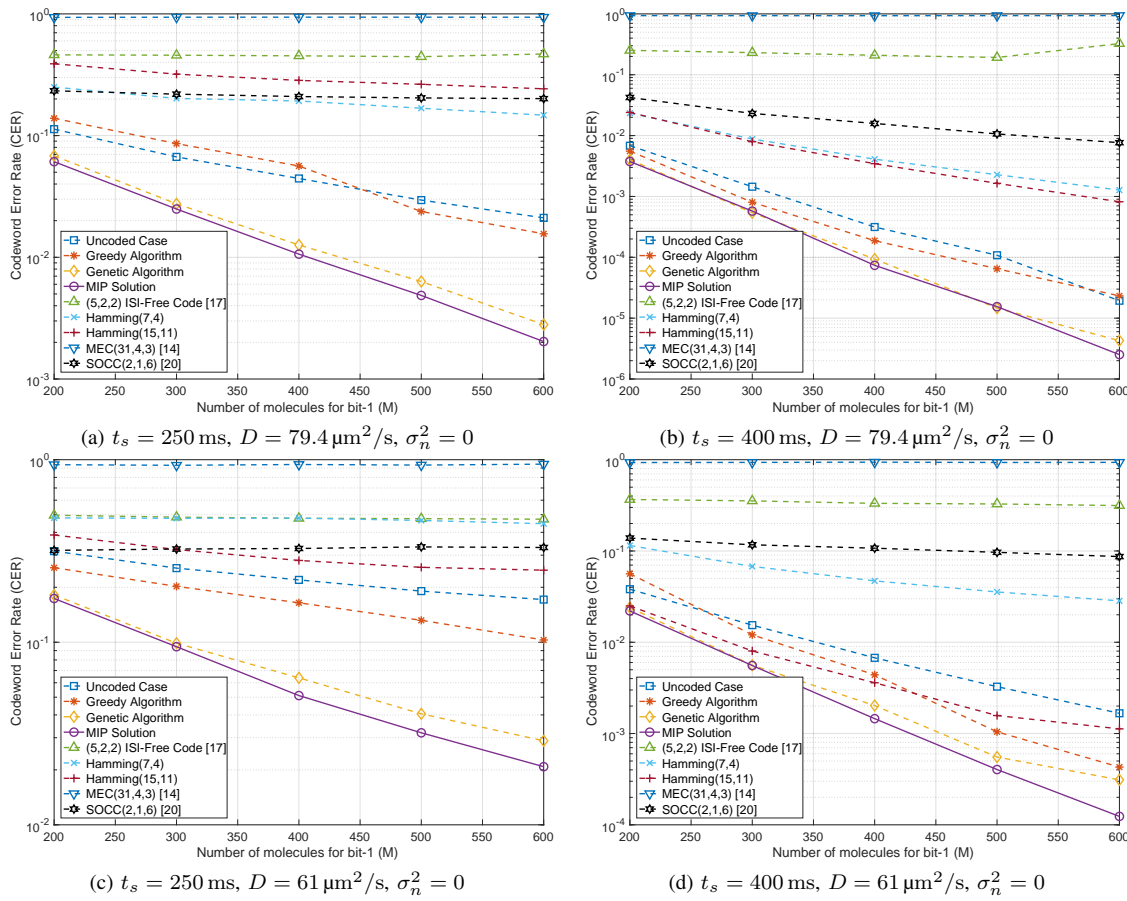


Fig. 8. CER of the coded and uncoded transmissions for different symbol durations and diffusion coefficients.

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.

## VI. CONCLUSION AND FUTURE WORKS

In this paper, 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, GA, and MIP 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 GA performs very close to MIP, 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.

## REFERENCES

- [1] T. Nakano, T. Suda, M. Moore, R. Egashira, A. Enomoto, and K. Arima, "Molecular communication for nanomachines using intercellular calcium signaling," in *5th IEEE Conference on Nanotechnology.*, July 2005, pp. 478–481 vol. 2.
- [2] M. Moore, 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," in *Proc. Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, March 2006, pp. 6 pp.–559.
- [3] L. C. Cobo and I. F. Akyildiz, "Bacteria-based communication in nanonetworks," *Nano Communication Networks*, vol. 1, no. 4, pp. 244–256, 2010.
- [4] M. S. Kuran, 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] M. Pierobon 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] M. S. Kuran, H. B. Yilmaz, T. Tugcu, and I. F. Akyildiz, "Modulation techniques for communication via diffusion in nanonetworks," in *Proc. IEEE International Conference on Communications (ICC)*, June 2011, pp. 1–5.
- [7] L. Shu and D. J. Costello, "Error control coding, fundamentals and applications," *Prentice Hall*, 2003.
- [8] A. Einolghozati and F. Fekri, "Analysis of error-detection schemes in diffusion-based molecular communication," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 615–624, March 2016.
- [9] Y. Lu, M. D. Higgins, and M. S. Leeson, "Diffusion based molecular communications system enhancement using high order Hamming codes," in *Proc. 9th International Symposium on Communication Systems, Networks Digital Sign (CSNDSP)*, July 2014, pp. 438–442.
- [10] M. B. Dissanayake, 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] M. B. Dissanayake, Y. Deng, A. Nallanathan, M. ElKashlan, and U. Mitra, "Enhancing the reliability of large-scale multiuser molecular communication systems," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2018, pp. 1–5.
- [12] M. Kovaevi 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] N. Abadi, A. A. Gohari, M. Mirmohseni, and M. Nasiri-Kenari, "Zero-error codes for multi-type molecular communication in random delay channel," in *2018 Iran Workshop on Communication and Information Theory (IWCIT)*, April 2018, pp. 1–6.
- [14] C. Bai, 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] Y. Lu, 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] P. J. Shih, C. h. Lee, and P. C. Yeh, "Channel codes for mitigating intersymbol interference in diffusion-based molecular communications," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 4228–4232.
- [17] P. J. Shih, 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] P. Akhkandi, A. Keshavarz-Haddad, and A. Jamshidi, "A new channel code for decreasing inter-symbol-interference in diffusion based molecular communications," in *Proc. 8th International Symposium on Telecommunications (IST)*, Sept 2016, pp. 277–281.
- [19] S. Qiu, T. Asyhari, and W. Guo, "Mobile molecular communications: Positional-distance codes," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2016, pp. 1–5.
- [20] Y. Lu, M. D. Higgins, and M. S. Leeson, "Self-orthogonal convolutional codes (SOCCs) for diffusion-based molecular communication systems," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 1049–1053.
- [21] P.-Y. Ko, 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," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 3748–3753.
- [22] T. Nakano, A. W. Eckford, and T. Haraguchi, *Molecular communication*. Cambridge University Press, 2013.
- [23] H. B. Yilmaz, 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] H. B. Yilmaz, C.-B. Chae, B. Tepekule, and A. E. Pusane, "Arrival modeling and error analysis for molecular communication via diffusion with drift," in *Proc. Nanoscale Computing and Communication (NANOCOM)*. ACM, 2015.
- [25] M. U. Mahfuz, D. Makrakis, and H. T. Mouftah, "A comprehensive study of sampling-based optimum signal detection in concentration-encoded molecular communication," *IEEE Transactions on NanoBio-science*, vol. 13, no. 3, pp. 208–222, Sept 2014.
- [26] L. Shi 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.
- [27] B. Tepekule, A. E. Pusane, M. . 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.
- [28] B. C. Akdeniz, A. E. Pusane, and T. Tugcu, "Optimal reception delay in diffusion-based molecular communication," *IEEE Communications Letters*, vol. 22, no. 1, pp. 57–60, Jan 2018.
- [29] N. Farsad, 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.
- [30] T. F. Gonzalez, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.