Error Detection in Diffusion-based Molecular Communication

Arash Einolghozati, Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
Email:{einolghozati, fekri}@ece.gatech.edu

***Abstract*—Despite the recent research activities in molecular communication among bio agents, the design of reliable schemes remains an open problem. One of the challenges is to develop suitable coding schemes which meet the molecular communication specific constraints in terms of reliability and complexity. In this paper, we consider diffusion-based molecular communication in which the information is encoded into the concentration (e.g., on/off keying). Such a communication system operates over a completely asymmetric channel where one of the bits can be transmitted without any error while the other can undergo a random error by the channel. Because of the limitations of bio agents, we focus on error-detecting schemes which require far less complexity at the receiver relative to error-correction codes. We model the detection process at the receiver via an erasure channel and propose an algorithm that obtains the optimal codewords efficiently. Then, we consider an error-free sub-family of such codes, namely constant weight codes, and propose an implementation specific to the molecular communication. We analyze the rate of the constant-weight coding scheme and specify the optimal weights and lengths of such codes. We also show that this coding scheme, by design, would enable the nodes to synchronize their communications.**

I. Introduction

Recent developments in nano/bio technology have motivated the researchers from various disciplines to study the fundamentals of molecular communication systems [1]–[4]. Among the remaining challenges in design of such systems is the communication reliability, in particular developing codings schemes compatible with molecular communication requirements [5]. In this paper, we are interested in a molecular communication setting in which the information is encoded in the concentration of molecules at the receiver. In particular, we use a model in which the transmitter sends a pulse of molecules for transmitting the bit “1” and shuts off for the bit “0” [6]. The molecules traverse the diffusion channel freely and their concentration is sensed by the receiver. By choosing the appropriate pulse lengths and shut-off periods in conjunction with methods introduced to expedite the communication, such as the use of enzymes in [7], one can assume that the symbols are decoded independently [6]. Furthermore, in the absence of other transmitters in the vicinity, the molecules cannot be produced in either the channel or the receiver and can only be lost. In other words, unlike the bit “1”, the bit “0” can be transmitted almost without any error. Hence, we use a completely asymmetric channel model (i.e., the z-channel) for the binary communication setting explained above and develop practical error-detecting schemes for it.

We focus on error-detecting schemes for two reasons. First, the envisioned biological communication networks require that a node acts as sensor that reports its readings (e.g., existence of a chemical in the environment [5]) to a destination. In such a context, we need a reliable error-detection capability but missing some transmissions can be tolerated as the nodes are perpetually sensing the environment and sending the information of interest. The second reason for considering the error-detecting schemes is to obviates the complexity imposed by error-correcting codes, especially at the receiver. The primitive bio nodes only have limited built-in complexity and hence, it is more practical for a receiver node to discard the erroneous codewords than trying to correct them as an optimal receiver would do.

We first, study the general problem of error-detecting coding schemes in completely asymmetric channels. We show the trade-off between the rate and probability of error in such detection codes. While using all the possible codewords achieves the maximum information rate, it also incurs the maximum probability of error. Here, we want to limit the probability of error and optimize the chosen codewords under such constraint. In order to find the optimal codewords, we model the detection process at the receiver by an erasure channel. We show that choosing the optimal codewords, in general, is an intractable problem and hence, employ dynamic programming to find the optimal codes efficiently under certain assumptions. We then discuss how constant-weight codewords, a sub-family of the above constructed codewords, can be implemented in molecular communication. Constant-weight codes, notably the four out of eight codes, have been commonly used for error detection in completely asymmetric channels in the context of conventional communication systems [8]. In such channels, the fixed weight of the codewords is the perfect error detection mechanism as the codeword weights can only decrease by error and hence, any error can be detected.

Coding schemes for molecular communication has been briefly discussed in the literature. In [9], a new distance measure for molecular communication has been studied. Authors in [10] apply convolutional coding schemes, by making use of memory units and XOR gates, to improve the probability of error in molecular communication. The use of Hamming codes

This material is based upon work supported by the National Science Foundation under Grant No. CNS-111094

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

for molecular communication is discussed in [11] in which logic gates are used for encoding and decoding. The field of synthetic biology is still in its infancy [12], and it is unclear how practical it would be to develop stable bio circuits for the coding purposes. Unlike the above schemes, we introduce a scheme that does not use logic gates or any storage. Hence, it is suitable for molecular communication in terms of both complexity and functionality of the nodes.

The rest of the paper is organized as follows. In Section II, we present the model for the general problem of finding the optimal codewords for an error-detecting scheme in a completely asymmetric channel. Section III introduces an algorithm based on dynamic programming to solve the problem of finding the optimal codewords. In Section IV, we introduce and analyze a perfect error-detecting scheme specific for molecular communication . Finally, Section V concludes the paper.

II. Optimal Error-detecting Coding Scheme for Completely Asymmetric Channels

We consider a molecular communication setting in which the diffusion channel inputs and outputs are the binary set $\{0, 1\}$. The transmitter releases a pulse of molecules in order to transmit the bit "1" and shuts off for the bit "0". By using appropriate waiting times and/or expediting the reseting process through use of specific enzymes [7], one could assume the independent decoding of the symbols. As shown in [6], the probability of error in the reception of the molecules approaches zero when the concentration of molecules is close to zero (i.e., sending bit "0"). On the other hand, there exists a non-zero probability of error p_e in transmission and decoding of the bit "1", which depends on the duration length and magnitude of the pulse. Note that p_e is due to both the communication channel and the uncertainty induced by the reception itself. As such, we use a completely asymmetric channel model (i.e., z-channel) with the parameter p_e to account for the molecular communication from the transmitter to the receiver.

Our objective is to find the optimal codewords for an error-detecting scheme in the above z-channel. We denote by $\mathbf{S}$ our codeword set where $\mathbf{S} \subseteq \{0,1\}^n$, and the codewords by $X^n \in \mathbf{S}$ where X^n are length n binary vectors. In order to capture the transition between the codewords in the aforementioned completely asymmetric channel, we define a codeword transition matrix $\mathbf{P}$ where $\mathbf{P}_{i.j} = Prob(X_i^n \to X_j^n)$. As an example, the general transition matrix is as follows for $n = 2$ (illustrated for the hypothetical case of $\mathbf{S} = \{0,1\}^2$):

$$\begin{array}{c} \\ (0,0) \\ (0,1) \\ (1,0) \\ (1,1) \end{array} \begin{array}{c} \begin{array}{cccc} (0,0) & (0,1) & (1,0) & (1,1) \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ p_e & 1-p_e & 0 & 0 \\ p_e & 0 & 1-p_e & 0 \\ {p_e}^2 & p_e(1-p_e) & p_e(1-p_e) & (1-p_e)^2 \end{pmatrix} \end{array}$$

At the receiver side, any vector that is not in $\mathbf{S}$ is erroneous and hence, discarded. In order to capture this behavior, we model all such received vectors as an erasure symbol E. Hence, the received set would be $\{S'\} = \{S\} \cup E$. The average probability of error in such a setting can be written as $P_{\text{avg}} = \sum_{i=1}^{K} p_i \sum_{j:X_j^n < X_i^n} \mathbf{P}_{i,j}$ where p_i is the probability of sending X_i^n, and $X_j^n < X_i^n$ if and only if $i \neq j$ and for every "1" in X_j^n, the corresponding position in X_i^n is "1" as well. The new transition matrix by considering the E symbol can be obtained by reducing the general transition matrix and removing the rows corresponding to the vectors $X_i \notin \mathbf{S}$ and adding the corresponding column to the E column. For example, the reduced transition matrix resulting from removing the vector $(0, 0)$ from $\mathbf{S} = \{0,1\}^2$ is given by:

$$\begin{array}{c} \\ (0,1) \\ (1,0) \\ (1,1) \end{array} \begin{array}{c} \begin{array}{cccc} (0,1) & (1,0) & (1,1) & E \end{array} \\ \begin{pmatrix} 1-p_e & 0 & 0 & p_e \\ 0 & 1-p_e & 0 & p_e \\ p_e(1-p_e) & p_e(1-p_e) & (1-p_e)^2 & {p_e}^2 \end{pmatrix} \end{array}$$

TABLE I
The effect of removing each codeword for $n = 2$

	(0, 0)	(0, 1)	(1, 0)	(1,1)
$R(bits/codeword)$	1.2076	1.2118	1.2118	1.2592
P_{avg}	0.0436	0.0615	0.0615	0.0647

Our goal is to find the detection code set $\mathbf{S}$ (and equivalently the set $\mathbf{S}'$) that is optimal in the sense that a given probability of error is satisfied and the information rate of the detection scheme is maximized. We model the detection process as a channel where the input symbols are the codewords $X^n \in \mathbf{S}$, the outputs are the received words $Y^n \in \mathbf{S}'$ and the channel is represented by the codeword transition matrix $P(Y^n|X^n)$ described above. With the above setup, the problem of finding an optimal error detection code resembles Shannon's channel capacity problem with the difference that we are not given the channel input symbols. It is worth clarifying that by information rate of the detection scheme, we mean $I(X^n; Y^n)$. This quantity would be the theoretical limit for the rate of any error detection coding scheme. Note that without the probability of error constraint, using all the codewords (i.e., $\mathbf{S} = \{0,1\}^n$) would achieve the maximum rate objective.

In order to obtain the optimal distribution and hence, the maximum rate for each choice of $\mathbf{S}$, we use the Blahut-Arimoto (BA) algorithm [13] on the corresponding codeword transition matrix. As a toy example, for $n = 2$ and $\mathbf{S} = \{0,1\}^2$, the maximum rate and the corresponding average probability of error are obtained as $R = 1.525$ bits/codeword and $P_{\text{avg}} = 0.089$. The resulting average probability of error P_{avg} and optimal rate R achieved by removing only one vector from the general transition matrix, are shown in Table I. Here, the vector in the column is removed from $\mathbf{S} = \{0,1\}^2$. As we observe in this Table, there exists a trade-off between the rate and the probability of error. We also observe that perturbing the error constraint may result in a completely different subset of codewords and hence, greedy algorithms that remove the most "suitable" codeword at each step won't

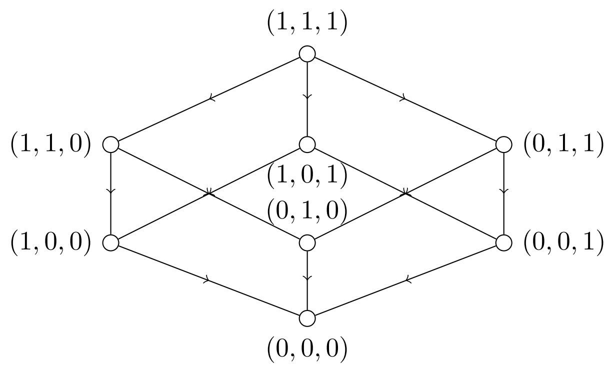

Fig. 1. The codeword graph for n=3

necessarily result in the final optimal codewords. In other words, different constraints on P_{avg} may result in different orders that the vectors are removed. It can be shown that finding the optimal codewords in the above general scenario is an NP-hard problem. In order to solve the problem for a more specific case, we model it on a directed graph. We have shown the codeword transition graph for $n = 3$ in Fig. 1. In such a directed graph, each potential codeword is represented by a vertex and the vectors are arranged from the highest weight to the lowest such that vectors with the same weights are on the same level (i.e., drawn on the same horizontal level). The edges of the codeword graph represent the transition of a vector to another with only a single bit error. A vector can be transformed to another if and only if there exists a directed path between input and the output vectors. Due to the asymmetric nature of the channel, the transformations can only happen from a codeword to the ones with lower weights located on lower levels of the graph. The probability of such transformation p_t can be written as:

$$p_t = p_e^h (1 - p_e)^{w-h} \tag{1}$$

where w is the transmitted vector weight and h is the number of hops in the directed path to the received vector.

We can also observe that the transformation probability is unique regardless of the path. The reason is that at each hop, the vector weight decreases by 1. Since traversing to above or across nodes is not possible, the number of hops is always equal to the difference in vector weights and independent of the chosen path. Based on the above description, the problem of finding the optimal code would map to finding the optimal subset of vertices of the above graph that satisfies the error probability constraint.

III. Dynamic-programming based Algorithm to Obtain the Optimal codeword Subset

In order to provide an algorithm rather than brute-force search to find the optimal codewords, we make two simplifying assumptions:

1) We relax the average probability of the detection error constraint and instead we use an upper limit on the maximum probability of the detection error. The advantage of this assumption is that we will be able to set a minimum distance h^* between each pair of chosen codewords and hence, can decide whether a subset is a viable code set before running the BA algorithm.
2) We confine ourselves to small enough p_e such that the transition probability between two codewords of the distance more than h^* is negligible. This assumption, as shown later, makes choosing the optimal codewords of two subsets with distance more than h^* independent of each other.

In the first phase of the algorithm, we need to obtain the minimum edge distance of the codewords in order to satisfy the maximum error probability constraint. Edge distance of two codewords is the directed distance between their corresponding vertices on the codeword graph. Since we only consider small p_e, it suffices to only consider the transition to the immediate neighbor of each codeword as the sole cause of error.

Lemma 1 *In order to comply with the maximum error probability p_{max}, the codewords must be at least in the distance h^* from each other where h^* is the minimum integer that satisfies $\max_w \binom{w}{h}\left(\frac{p_e}{1-p_e}\right)^h (1-p_e)^w \leq p_{max}$ for $0 \leq w \leq n$.*

Proof. Consider the codeword c_w with the weight w. Since the weight of the vectors at the distance h from c_w is $w - h$, there are $\binom{w}{h}$ vectors at distance h that c_w can be transformed to. Using (1), in order to satisfy the error constraint for all the graph, we need $\max_w \binom{w}{h}\left(\frac{p_e}{1-p_e}\right)^h (1-p_e)^w \leq p_{\max}$ where $0 \leq w \leq n$. Note that for the small values of p_e, $w = n$ maximizes the above criterion. ∎

In the second phase of the algorithm, we employ dynamic programming [14] to find the optimal codewords given the minimum edge distance h^*.

Lemma 2 *If one vector with weight w is chosen for the optimal code, all the vectors with the same weight must be chosen.*

Proof. First, we observe that since all the vectors with the same weight are on the same level in the codeword graph, the minimum edge distance remains intact by exhausting all the codewords in that level. Hence, the maximum probability of error constraint would still hold. Secondly, adding more codewords to a subset of codewords can only increase the information rate of the detection scheme. The reason is that the optimal distribution for the original subset can be achieved by setting the probability of the new codewords to zero. Hence, the rate obtained by the old set is always sub-optimal to the one obtained by the new set. ∎

Hence, the problem of finding the optimal code amounts to finding the optimal codeword weights. Note that this alone reduces the number of viable sets, i.e., number of BA algorithm runs, from $O(2^{(2^n)})$ to $O(2^n)$. Now, we resort to dynamic programming to obtain the optimal weights with $O(n)$ runs of the BA algorithm. We assume C_k to be the set of weight-k vectors, S_k to be the optimal codeword set using only the weights lower than or equal to k, and R_k^* to be the corresponding optimal rate. Moreover, we denote by $R(\Pi)$ the rate of the set Π. Our objective is to obtain S_n and R_n^*. The algorithm works as follows:

- Initialize S_0 by C_0 and S_k by the empty set for $k < 0$.
- Initialize R_k^* to zero for $k \leq 0$.
- for $k = 1, 2, 3, \ldots, n$, do: $R_k^* = \max(R(C_k \cup S_{k-h^*}), R(C_k \cup S_{k-h^*-1}), R_{k-1}^*, R_{k-2}^*..., R_{k-h^*+1}^*)$ and update S_k accordingly to either $C_k \cup S_{k-h^*}$, or $C_k \cup S_{k-h^*-1}$, or S_{k-1},..., or S_{k-h^*+1}.

The above algorithm ensures that the minimum edge distance between the codewords is always held at h^* and builds upon the solutions to the smaller sets in order to obtain the optimal solution of the larger ones. Note that at the maximization step, the average probability of error can be calculated since the optimal distribution of the codewords is already obtained by the BA algorithm at that step. Hence, one could discard the choices that yield average probability of error greater than a threshold with no additional cost. In order to obtain R_n^* and S_n, in each iteration, we need two runs of the BA algorithm and $O(h^*)$ comparisons which is negligible compared with the BA algorithm cost. Hence, we have reduced the total number of BA algorithm runs from $O(2^n)$ to $O(n)$. It is worth noting that the overall performance of the above algorithm is not linear in n as the BA algorithm itself has exponential complexity with respect to n. Before we prove the correctness of the above algorithm, we need the following lemma.

Lemma 3 *Assume the codeword subsets S_1 and T_1 are the optimal subsets of the sets S and T, respectively, which satisfy the maximum error detection probability constraint. If there exists no transition probability between the codewords of the sets S and T, $S_1 \cup T_1$ would be the optimal codewords for $S \cup T$ under the same constraints.*

Proof. First, notice that since there is no transition probability between the codewords of S_1 and T_1, the overall transition matrix is in the form of $C = \left[\begin{array}{c|c} T_1 & 0 \\ \hline 0 & S_1 \end{array}\right]$. Hence, this transition matrix can be viewed as two independent parallel channels with overall rate of $R(S_1 \cup T_1) = R(S_1) + R(T_1)$. Now imagine there were subsets $S_2 \subset S$ and $T_2 \subset T$ which $S_2 \cup T_2$ would give the optimal rate for the set $S \cup T$, i.e., $R(S_1 \cup T_1) < R(S_2 \cup T_2)$. Since the transition matrix of $S_2 \cup T_2$ has a similar form as above, we would have $R(S_1) + R(T_1) < R(S_2) + R(T_2)$ which would mean that either $R(S_1) < R(S_2)$ or $R(T_1) < R(T_2)$, or both. Imagine the first case was true. Since $S_2 \cup T_2$ is assumed to satisfy the maximum error probability, each one of the individual sets must satisfy the constraint individually. Hence, there would exist a subset $S_2 \subset S$ that achieves a higher rate than the optimal subset S_1 under the same constraint, which is a contradiction. ■

Theorem 4 *The dynamic algorithm described above, ensures finding the optimal codewords under the maximum error probability constraint.*

Proof. In order to show the correctness of the algorithm, we need to show that given the optimal answer for the sub-problems $1, 2, \ldots, k-1$, the algorithm gives the optimal answer for the k^{th} problem. First, assume the weight-k vectors C_k are not included in S_k. Since the minimum edge distance of the codewords must be h^*, R_k^* would be $\max(R_{k-1}^*, R_{k-2}^*, \ldots, R_{k-h^*+1}^*)$ and S_k would be the corresponding subset. Note that $R_{k-h^*}^*$ cannot be the optimal answer as adding C_k to S_{k-h^*} would increase the optimal rate while we assumed C_k is not included in S_k. Now consider the case where C_k is indeed in the optimal codeword set S_k. We consider two sub-cases:

First, consider the case that C_{k-h^*} is not included in S_{k-h^*}, i.e., $S_{k-h^*} = S_{k-h^*-1}$. Here, the set $S_{k-h^*} \cup C_k$ would be the optimal set because the minimum edge distance between C_k and S_{k-h^*} is greater than h^* and since we ignore the transition between two codewords of edge distance more than h^*, using Lemma 3, choosing C_k does not impact the optimality of S_{k-h^*}. Next, consider the case that C_{k-h^*} is actually included in S_{k-h^*}, and hence, picking C_k could possibly impact the optimality of S_{k-h^*} for the smaller problem. This could possibly result in evicting C_{k-h^*} from the optimal set. If so, the rest of the codewords are in distance at least $h^* + 1$ from C_k, and hence, S_{k-h^*-1} would be the optimal subset for the smaller problem. If C_{k-h^*} is not evicted from the optimal set, since other codewords in S_{k-h^*} are not impacted by the presence of C_k (as their distance is more than h^*), S_{k-h^*} remains optimal. Hence, we need to consider $R(C_k \cup S_{k-h^*})$ and $R(C_k \cup S_{k-h^*-1})$ in the maximization step of the algorithm for the case that C_k is included in the optimal set as well as $R_{k-1}^*, R_{k-2}^*, \ldots, R_{k-h^*+1}^*$ for the case that C_k is not included. ■

In Fig. 2, we have shown the optimal rate at each step of the algorithm for $n = 8$, $p_e = 0.1$, different values of h^*, and the corresponding maximum error probability. Note that $h^* = 1$ corresponds to the case that all the vectors can be used. We can observe the amount of forgone rate in order to achieve lower maximum error probabilities. Further, the incremental value of the higher-weight vectors diminishes at each step. On the other hand, in order to show the negative effect of high-weight codewords, we have plotted the average probability of error for the optimal codewords at each step of the algorithm in Fig. 3. As we can see in this plot, the average probability of error continues to increase even where there is no significant improvement in the rate by including more high-weight vectors in the code set. This suggests stopping the algorithm when the rate improvement is negligible by increasing k.

IV. Coding Schemes for Perfect Error Detection and Synchronization in Molecular Communication

In this section, we study and analyze the constant-weight codes for molecular communication. These codes are sub-family of the general optimal codes discussed in the previous

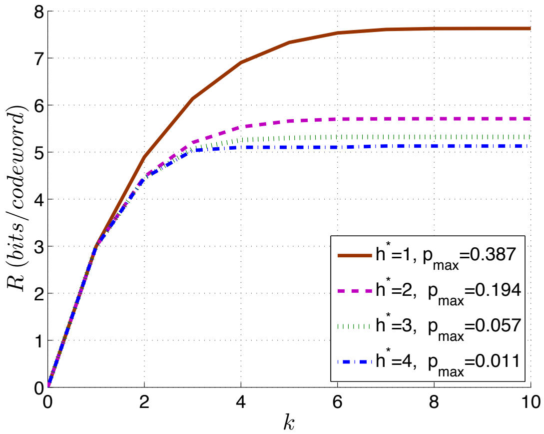

Fig. 2. Optimal rates at each step of the algorithm

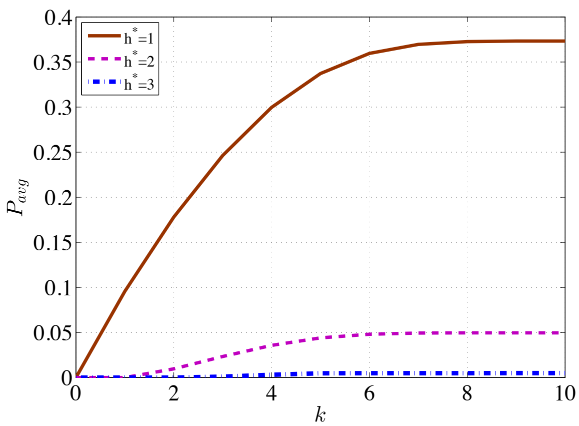

Fig. 3. optimal codewords' average probability of error at each step

section, by constraining the maximum probability of error to zero. This coding scheme is suitable for the low complexity requirements of the molecular communication especially for the communication systems that need perfect error detection. As discussed before, the binary (on/off) molecular communication works as follows: In order to send symbol "1", the transmitter sends pulses of duration T of signal molecules (AHL) and shuts off for the same period of time in order to send symbol "0". The duration T must be long enough to give the receiver sufficient time to sense the concentration of the AHL molecules. It must also provide enough time for the channel to reset from high to low. The optimal value of T is studied in [6]. It is also shown in [7] that specific enzymes can be used to expedite this process. Hence, we assume that the symbols can be decoded independently at the receiver. Hence, as in the previous section, symbol "0" can be transmitted impeccably while there is a non-zero probability of error p_e in transmitting "1" (i.e., the completely asymmetric channel).

One of the main constraints in designing a coding scheme for molecular communication is the low complexity requirements in encoding and decoding. In addition, unlike the common case in molecular communication literature, we do not assume the existence of an additional mechanism to synchronize the transmitter and the receiver [15], and instead, we seek to provide synchronization using the proposed code. Moreover, since sending the symbol "0" and sending nothing can seem the same, the receiver cannot differentiate between codewords such as (0,0,1,1) and (0,1,1,0). To mitigate this problem and also synchronizing the communication, each codeword should begin with "1" to signal the beginning of a codeword. Note that we assume that the least significant digit is sent first.

Our objective here, is to implement perfect error-detecting schemes for the above molecular communication by employing constant weight codes, and to compare the rate loss compared with the theoretical limit from the previous section. As an example, considering the synchronization constraint, the allowed codewords with weight two for $n = 4$ are: (0,0,1,1), (0,1,0,1) and (1,0,0,1). Note that since the conversion of the symbol "0" into "1" is not possible, error among the codewords of the same weight cannot happen. At the receiver side, weight of the codewords is examined and in case of a mismatch with a preset weight, the codeword is discarded.

Here, we explain how the decoding process at the receiver works. Arriving of the first "1" symbol indicates an incoming codeword. The following symbols are decoded one by one for the duration of the codeword length which is known to the receiver. The output of the receiver agents (e.g., bacteria) is only produced when symbol "1" is detected and it is in the form of Green Fluorescent Protein (GFP) [16]. Equipped with a circuitry, the receiver chamber should have the capability of detecting the aggregate output produced by the agents during the entire codeword. In order to check the codeword weights, the chamber circuitry accepts a codeword only if the aggregated level of the output surpasses a threshold. This threshold can be programmed beforehand and should be higher than the level obtained where the codeword weight is less than the preset weight. This way, any erroneous codeword can be detected and discarded when the output is below the threshold.

In order to analyze the general length n family of codes described here, we denote by w the constant weight of such codes and by p_e the probability of error for transmitting the symbol "1". Hence, the codeword probability of error ,which results in being rejected by the receiver, is $P_e = 1 - (1 - p_e)^w$. Hence, the expected number of transmissions of each codeword (in order to to be transmitted correctly) would be equal to $\frac{1}{(1-p_e)^w}$; using a Geometric r.v. with parameter $(1 - p_e)^w$. The number of information bits in such a codeword is $\log_2 \binom{n-1}{w-1}$ as the first bit is always "1". Hence, the average rate for such a code would be

$$R = \frac{(1-p_e)^w}{n} \log_2 \binom{n-1}{w-1}. \quad (2)$$

The code rate versus weight is shown in Fig. 4 for $n = 8$ and for different values of the probability of error p_e. As we observe in the plot, and shown to be true for other values of n, the low-weight codes are optimal in high p_e but optimal w approaches $\frac{n}{2}$ for small values of p_e.

In Fig. 5, we have shown the effect of block size n for different channel probability of error when the optimal weight is chosen. Since the probability of error is small, $w = \frac{n}{2}$

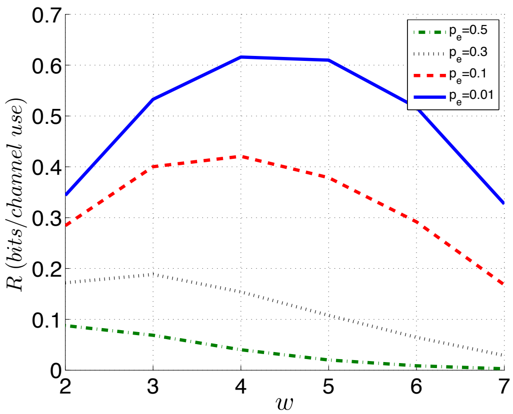

Fig. 4. Code rates versus the codeword weight for different values of p_e

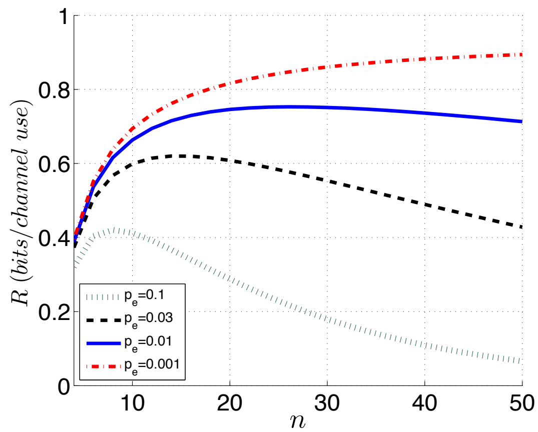

Fig. 5. Optimal code rates versus the code length for different values of p_e

is chosen as the optimal weight. As we see in the plot, the optimal block size increases for a more reliable channel but the rate eventually approaches zero as the code length continues to increase. Moreover, we can observe the rate loss due to perfect error detection by comparing the $p_e = 0.1$ curve in Fig. 4 with the top curve in Fig. 2. We can see that imposing perfect error detection together with synchronization constraint result in about 40% loss in the code rate compared with the maximum theoretical limit obtained by imposing no constraints on probability of error.

In the above coding scheme, the only complexity imposed on the receiver node (i.e., the chamber circuitry) is the ability to read consecutive bio agents' output, aggregate and compare it with a threshold, and report the decoded output if it passes the threshold.

V. Conclusion

In this paper, we studied the error-detecting coding schemes for the molecular communication channel. We considered diffusion-based molecular communication in which the information is encoded into the binary concentration of molecules. and the communication process was modeled as a completely asymmetric channel. We showed the trade-off between the rate and probability of error for such coding schemes. In order to obtain the theoretical limits, we introduced a dynamic programming algorithm that finds the optimal codewords under the maximum error probability constraint. We then studied constant-weight codewords, a sub-family of above codes, which meet the molecular communication specific needs in terms of both perfect error detection and low complexity. We showed that through the use of these codes, synchronization can also be achieved. At the receiver side, the weight of the codewords is assessed and discarded in case of a mismatch with a default weight. We also analyzed the optimal weight and length of such codes and compared the rate with the theoretical limit obtained earlier.

References

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

[2] C. Rose and I. S. Mian, "Signaling with identical tokens: Upper bounds with energy constraints," in *Information Theory (ISIT), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 1817–1821.

[3] M. Pierobon, "A molecular communication system model based on biological circuits," in *Proceedings of ACM The First Annual International Conference on Nanoscale Computing and Communication*. ACM, 2014, pp. 1–8.

[4] N. Farsad, A. W. Eckford, S. Hiyama, and Y. Moritani, "On-chip molecular communication: analysis and design," *NanoBioscience, IEEE Transactions on*, vol. 11, no. 3, pp. 304–314, 2012.

[5] T. Nakano, T. Suda, Y. Okaie, M. Moore, and A. Vasilakos, "Molecular communication among biological nanomachines: A layered architecture and research issues," 2014.

[6] A. Einolghozati, M. Sardari, and F. Fekri, "Design and analysis of wireless communication systems using diffusion-based molecular communication among bacteria," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 12, pp. 6096–6105, 2013.

[7] A. Noel, K. C. Cheung, and R. Schober, "Improving receiver performance of diffusive molecular communication with enzymes," *arXiv preprint arXiv:1305.1926*, 2013.

[8] M. Blaum, *Codes for detecting and correcting unidirectional errors*. IEEE Computer Society Press, 1993.

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

[10] M. Mahfuz, D. Makrakis, and H. Mouftah, "Performance analysis of convolutional coding techniques in diffusion-based concentration-encoded pam molecular communication systems," *BioNanoScience*, vol. 3, no. 3, pp. 270–284, 2013.

[11] M. Leeson and M. Higgins, "Error correction coding for molecular communications," in *Communications (ICC), 2012 IEEE International Conference on*, 2012, pp. 6172–6176.

[12] N. Nandagopal and M. B. Elowitz, "Synthetic biology: integrated gene circuits," *science*, vol. 333, no. 6047, pp. 1244–1248, 2011.

[13] R. W. Yeung, *A first course in information theory*. Springer, 2002, vol. 1.

[14] R. E. Bellman and S. E. Dreyfus, *Applied dynamic programming*. Rand Corporation, 1962.

[15] H. ShahMohammadian, G. Messier, and S. Magierowski, "Blind synchronization in diffusion-based molecular communication channels," *Communications Letters, IEEE*, vol. 17, no. 11, pp. 2156–2159, November 2013.

[16] P. Mehta, S. Goyal, T. Long, B. L. Bassler, and N. S. Wingreen, "Information processing and signal integration in bacterial quorum sensing." *Mol Syst Biol*, vol. 5, p. 325, 2009.