Polar Coding with Chemical Reaction Networks for Molecular Communications

Matsumine, Toshiki; Koike-Akino, Toshiaki; Wang, Ye

TR2020-160 December 11, 2020

Abstract

In this paper, we propose a new polar coding scheme with molecular programming, which is capable of highly parallel implementation at a nano-scale without the need for electrical power sources. We designed chemical reaction networks (CRN) to employ either successive cancellation (SC) or maximumlikelihood (ML) decoding schemes for short polar codes. From differential equation analysis of the proposed CRNs, we demonstrate that SC and ML decoding achieve accurate computations across fully-parallel chemical reactions. In terms of the number of required chemical reactions, we verify the superiority of ML decoding over SC decoding for very short block lengths

IEEE Global Communications Conference (GLOBECOM)

Polar Coding with Chemical Reaction Networks for Molecular Communications

Toshiki Matsumine[‡†], Toshiaki Koike-Akino[‡], and Ye Wang[‡]
[‡] Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA.
[†] Department of Electrical and Computer Engineering, Yokohama National University,
79-5 Tokiwadai, Hodogaya, Yokohama, Kanagawa, Japan
Email: toshiki.matsumine.jp@ieee.org, {koike, yewang}@merl.com

***Abstract*—In this paper, we propose a new polar coding scheme with molecular programming, which is capable of highly parallel implementation at a nano-scale without the need for electrical power sources. We designed chemical reaction networks (CRN) to employ either successive cancellation (SC) or maximum-likelihood (ML) decoding schemes for short polar codes. From differential equation analysis of the proposed CRNs, we demonstrate that SC and ML decoding achieve accurate computations across fully-parallel chemical reactions. In terms of the number of required chemical reactions, we verify the superiority of ML decoding over SC decoding for very short block lengths.**

I. Introduction

Thanks to recent advancements of bio-molecular technologies, molecular programming has been rapidly grown as an emerging topic. Chemical reaction networks (CRNs) are a useful descriptive programming language for modeling complex chemical systems. New possible applications with CRNs have been widely studied so far. For example, chemical implementation of neural networks has been studied in [1–3]. Also, digital logic [4–8] and belief propagation (BP) with CRNs have been proposed in [9, 10]. CRNs are Turing-universal models, which can be used to perform arbitrary computation [11]. Although it is beyond our scope in this paper, deoxyribonucleic acid (DNA) has been extensively studied to translate CRNs in the literature [12]. In fact, designing DNA strands can realize the entire dynamic behaviors of CRNs [13].

In this paper, we develop CRNs to implement polar coding for molecular communications. Molecular communications are emerging paradigms used in various biomedical and healthcare applications such as nanoscale lab-on-a-chip, in-situ physiological sensing, targeted drug delivery, artificial morphogenesis, and neural signal transduction [14–19]. It is known that molecular communications are inherently unreliable due to stochastic molecular propagation, molecule-to-molecule collisions, chemical degradation, and environmental noise. Accordingly, the molecular channels experience extremely long latency, large jitters, high erasure rate, and low capacity [20–23]. To deal with the unreliable molecular channels, several error control schemes have been investigated, including feedback-based rate control [24–26], automatic repeat request [27, 28], and forward error correction [29–34].

T. Matsumine conducted this research when he was an intern at MERL.

For in-vivo applications, the transmitted data size is expected to be small. Hence, short channel coding that has powerful error correcting capability may be required for such applications. As shown in the literature [35, 36], polar codes have an advantage over state-of-the-art low-density parity-check (LDPC) codes in the short block-length regime, which indicates that polar codes may be a potential candidate for error-correcting codes in molecular communication systems.

In this paper, we propose several chemical implementations of short polar codes. We investigate two decoding schemes for short polar codes: successive cancellation (SC) decoding and maximum-likelihood (ML) decoding. Our implementation does not rely on the specific rate constant of chemical reactions, and thus parallel implementation is possible. In simulation results, we show that the performances of the two decoding schemes, in terms of the trade-off between computation accuracy and speed, are comparable, whereas the ML decoder can be realized by using much fewer chemical species and reactions when the code length is very short. The main contributions of this paper are summarized as follows:

- We propose a novel application of molecular computing to polar encoding and decoding;
- We design two CRN-based decoding schemes of polar codes: SC and ML decoding; and
- We evaluate the performance trade-off between accuracy and speed of the proposed CRNs, as well as the number of required chemical species and reactions.

Although such a molecular SC decoder was discussed in [37], our preliminary report in [38] is earlier than their preprint. Moreover, no molecular ML decoder was studied for polar codes to the best of our knowledge.

II. Preliminaries

A. Molecular Programming

In order to implement computational abstractions with biomolecules, abstract computations shall be "compiled" into a CRN. Fig. 1 illustrates this compiling step of molecular programming. The first step is to translate abstract computation, which is polar coding in our case, into a CRN. Note that this translation is always possible from the fact that CRNs are universal Turing machines [11]. The second step is to translate CRNs into actual molecules such as DNA molecules.

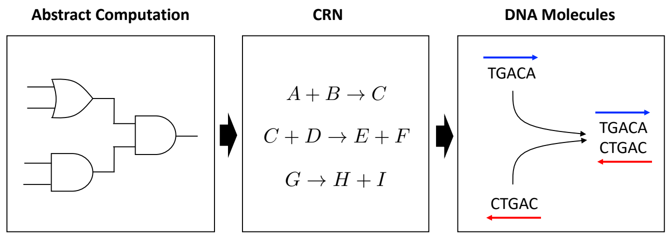

Fig. 1. Compiling steps of molecular programming with CRNs [12]: 1) abstract computation, 2) CRN implementation of a computation (represented by a set of chemical reactions), 3) DNA representation for generating CRN (A, G, C, T are nitrogenous nucleobases).

A DNA strand is a sequence of nucleotide bases each of which can be either adenine (A), cytosine (C), guanine (G), or thymine (T). This step should also be always possible for any CRN as described in [13], where the general construction of CRN-DNA translation is studied. In this work, we consider translation of polar decoding for discrete variables into CRNs. Basic notations and examples of probability calculation using CRNs are described in the subsequent subsection.

B. Chemical Reaction Networks (CRNs)

Consider an example of CRNs involving three chemical species A, B, and C, which follow a chemical reaction:

$$A+B\xrightarrow{k}C, \tag{1}$$

where A and B are *reactants*, C is *products*, and k is the rate constant that indicates how fast the reaction occurs. Species that participate in a reaction, while not being consumed or produced, are called *catalysts*. For example, the species A in the following reaction is called catalyst

$$A+B\xrightarrow{k}A+C. \tag{2}$$

For CRN equations, the empty set ϕ is used to denote null reactants or products. For example, when ϕ is used in place of the reactants, e.g.,

$$\phi\xrightarrow{k}C, \tag{3}$$

it denotes that the products are generated from a large or replenishable source. When ϕ appears as the product, e.g.,

$$A+B\xrightarrow{k}\phi, \tag{4}$$

it denotes that species A and B cancel out equal concentrations by transferring them to an external sink. We design CRNs so that the *equilibrium* concentration of some species shows the result that we want to compute. Throughout this paper, we assume a reaction rate of 1, since our implementation does not depend on the specific rate.

C. Basic Operations

Both bit and probability computations with CRNs are required for the chemical implementation of polar encoding and decoding. In this subsection, we review how bits and probabilities are represented with molecular concentrations and how some basic computations are performed with CRNs [10].

1) Bit Representation: In order to represent a bit with CRNs, we use a *complementary representation* [6], where two molecules A^0 and A^1 are used for a single bit A. The presence of molecule A^0 indicates that $A=0$, and vice versa. For this reason, molecules A^0 and A^1 should not be present at the same time. This is implemented by the following reaction set

$$A^0+A^1\to S, \tag{5}$$
$$S+A^0\to 3A^0, \tag{6}$$
$$S+A^1\to 3A^1, \tag{7}$$

where in (5), A^0 and A^1 are mutually consumed so that only one may be present in steady state, whereas S is an intermediate compound.

2) Probability Expression: The probability is expressed by a ratio of concentrations of two molecules A^0 and A^1,

$$P_A=\frac{[A^1]}{[A^0]+[A^1]},\qquad P_A^{\mathrm{c}}=1-P_A=\frac{[A^0]}{[A^0]+[A^1]}, \tag{8}$$

where $[\cdot]$ denotes the concentration of the argument molecule.

3) Probability Multiplication: Let us consider multiplication of two probabilities $P_C=P_A\times P_B$. Letting initial concentrations $[C^0]=[C^1]=0.5$, we transfer $[C^0]$ to $[C^1]$ when $AB=1$ and $[C^1]$ to $[C^0]$ when $AB=0$. This is performed by the following set of reactions

$$A^0+C^1\to A^0+C^0, \tag{9}$$
$$A^1+B^0+C^1\to A^1+B^0+C^0, \tag{10}$$
$$A^1+B^1+C^0\to A^1+B^1+C^1, \tag{11}$$

where in (9), concentration $[C^1]$ is transferred to $[C^0]$ for $A=0$. Similarly, (10) corresponds to $A=1,B=0$, and (11) corresponds to $A=1,B=1$. We denote this probability multiplication by

$$P_C=\mathsf{multiply}(P_A,P_B). \tag{12}$$

Our implementation is different from that in [10] in that we continuously calculate $P_C=P_A\times P_B$, whereas the calculation in [10] is performed only when an auxiliary molecule S exists.

4) Probability Division: To perform division of two probabilities, $P_C=P_A/(P_A+P_B)$, we initialize concentrations of two molecules as $[C^0]=[C^1]=0.5$. Then, division is performed by transferring $[C^0]$ and $[C^1]$ such that their ratio is equal to that of $[A^1]$ and $[B^1]$, i.e., $[A^1]:[B^1]=[C^0]:[C^1]$. The following set of reactions performs division of two probabilities, $P_C=[C^1]/([C^0]+[C^1])=P_A/(P_A+P_B)$:

$$A^1+C^1\to A^1+C^0, \tag{13}$$
$$B^1+C^0\to B^1+C^1. \tag{14}$$

For simplicity, we use the following notation for this function

$$P_C=\mathsf{divide}(P_A,P_B). \tag{15}$$

III. Chemical Polar Encoder

In this section, our proposed chemical implementation of polar encoding is described. The generator matrix of polar codes with a block length of $N = 2^n$ is expressed as follows:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes n}, \tag{16}$$

where $[\cdot]^{\otimes n}$ denotes the nth Kronecker power.

For general codes, encoding, i.e., multiplication by a generator matrix, may be performed based on two basic binary operations, copy and exclusive-OR (XOR). Suppose A and B are input bits, a chemical implementation of the XOR operation $C = \mathsf{XOR}(A, B)$ is proposed in [6] as below:

$$A^0 + B^1 \to A^0 + B^1 + C^{1\prime}, \tag{17}$$
$$A^1 + B^0 \to A^1 + B^0 + C^{1\prime}, \tag{18}$$
$$C^{1\prime} \to \phi, \tag{19}$$
$$C^{1\prime} + C^0 \to C^1, \tag{20}$$
$$A^0 + B^0 \to A^0 + B^0 + C^{0\prime}, \tag{21}$$
$$A^1 + B^1 \to A^1 + B^1 + C^{0\prime}, \tag{22}$$
$$C^{0\prime} \to \phi, \tag{23}$$
$$C^{0\prime} + C^1 \to C^0, \tag{24}$$

where reactions in (17)–(20) will generate the molecule C^1 when $A = 0, B = 1$ and $A = 1, B = 0$, and reactions (21)–(24) produce the molecule C^0 when $A = 0, B = 0$ and $A = 1, B = 1$. The copy operation can be performed by reusing the XOR implementation. Specifically, we perform a copy of A by taking the XOR with 0, i.e., $C = \mathsf{copy}(A) = \mathsf{XOR}(A, 0)$.

IV. Chemical SC Decoding

We here describe the proposed implementation of a polar decoder based on the factor graph. Fig. 2 shows the factor graph of polar codes with a block length of $N = 4$. As shown in this figure, SC decoding consists of two fundamental computations; the f-function and g-function [39]. In what follows, we review these functions to propose efficient implementations with CRNs. Although f- and g-functions for LDPC BP decoders are proposed in [10], our approach is different from [10] in that our polar decoder is asynchronous and does not require chemical clock CRNs for circuit synchronization.

A. The f-Function

The f-function in SC decoding takes two probabilities P_A and P_B as its input, and outputs $P_C = f(P_A, P_B)$. As shown in [39], the probability P_C and its complementary probability $P_C^{\mathrm{c}} = 1 - P_C$ is calculated as

$$P_C = P_A P_B^{\mathrm{c}} + P_B P_A^{\mathrm{c}}, \tag{25}$$
$$P_C^{\mathrm{c}} = 1 - P_C = P_A^{\mathrm{c}} P_B^{\mathrm{c}} + P_A P_B. \tag{26}$$

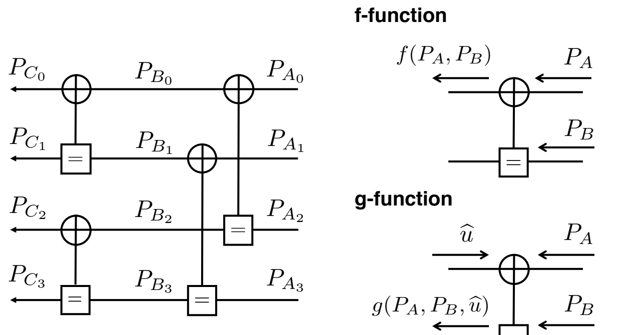

Fig. 2. SC decoder with block length $N = 4$. Probability is represented by the concentration ratio of two molecules $P_{A_i} = [A_i^1]/([A_i^1] + [A_i^0])$.

Initializing molecular concentrations as $[C^0] = [C^1] = [D^0] = [D^1] = [E^0] = [E^1] = [F^0] = [F^1] = [G^0] = [G^1] = 0.5$, the f-function is implemented as follows:

$$P_D = \mathsf{multiply}(P_A, P_B), \tag{27}$$
$$P_E = \mathsf{multiply}(P_A^{\mathrm{c}}, P_B), \tag{28}$$
$$P_F = \mathsf{multiply}(P_A, P_B^{\mathrm{c}}), \tag{29}$$
$$P_G = \mathsf{multiply}(P_B^{\mathrm{c}}, P_B^{\mathrm{c}}), \tag{30}$$
$$F^1 + C^0 \to F^1 + C^1, \tag{31}$$
$$E^1 + C^0 \to E^1 + C^1, \tag{32}$$
$$D^1 + C^1 \to D^1 + C^0, \tag{33}$$
$$G^1 + C^1 \to G^1 + C^0, \tag{34}$$

where P_D, P_E, P_F, and P_G correspond to $P_A P_B$, $P_A^{\mathrm{c}} P_B$, $P_A P_B^{\mathrm{c}}$, and $P_A^{\mathrm{c}} P_B^{\mathrm{c}}$, respectively. Note that we have $P_D = [D^1]/([D^0] + [D^1]) = [D^1]$ since we initialize as $[D^0] + [D^1] = 1$, and similarly, $P_E = [E^1]$, $P_F = [F^1]$, and $P_G = [G^1]$. After that, reactions (31)–(34) transfer C^0 to C^1 based on the ratio of $[F^1] + [E^1]$ to $[D^1] + [G^1]$, and the resulting concentrations can realize the computations (25) and (26).

B. The g-Function

In addition to two input probabilities P_A and P_B, the g-function depends on the decision $\widehat{u}$ after the f-function. We denote g-function by $P_C = g(P_A, P_B, \widehat{u})$:

$$P_C = \begin{cases} \frac{P_A P_B}{P_A P_B + P_A^{\mathrm{c}} P_B^{\mathrm{c}}}, & \text{if } \widehat{u} = 0, \\ \frac{(1-P_A) P_B}{P_A^{\mathrm{c}} P_B + P_A P_B^{\mathrm{c}}}, & \text{otherwise.} \end{cases} \tag{35}$$

The following set of reactions shows the computation of the g-function when $\widehat{u} = 0$:

$$P_D = \mathsf{multiply}(P_A, P_B), \tag{36}$$
$$P_G = \mathsf{multiply}(P_A^{\mathrm{c}}, P_B^{\mathrm{c}}), \tag{37}$$
$$P_C = \mathsf{divide}(P_D, P_G). \tag{38}$$

Analogously, the g-function with $\widehat{u} = 1$ can be performed by replacing P_A and P_A^{c} in reactions (36) and (37).

Letting A^0 and A^1 denote outputs from the f-function, we make a decision by setting $[U^0]=1,[U^1]=0$ if $[A^0]>[A^1]$ and otherwise $[U^0]=0,[U^1]=1$. This is implemented as:

$$A \to A + X, \tag{39}$$
$$A^{\mathrm{c}} \to A^{\mathrm{c}} + Y, \tag{40}$$
$$X + Y \to \phi, \tag{41}$$
$$X + U^0 \to X + U^1, \tag{42}$$
$$Y + U^1 \to Y + U^0, \tag{43}$$

where initial concentrations of U^0 and U^1 are $[U^0]=[U^1]=0.5$. Reactions (39) and (40) copy A^0 and A^1 to X and Y, respectively, and (42) and (43) divide U^0 and U^1 according to the ratio $[A^0]:[A^1]$. Reaction (41) consumes molecules X and Y, such that both of them together are completely consumed. The reactions (39)–(43) are not required for frozen bits, where we set $[U^0]=1,[U^1]=0$.

As mentioned earlier, since the g-function depends on the previous decision, we implement this as follows:

$$U^0 + g(P_A,P_B,\widehat{u}=0) \to U^0 + g(P_A,P_B,\widehat{u}=0), \tag{44}$$
$$U^1 + g(P_A,P_B,\widehat{u}=1) \to U^1 + g(P_A,P_B,\widehat{u}=1). \tag{45}$$

Since either U^0 or U^1 is completely consumed in (5), either of (44) and (45) is computed, e.g., only the reaction (44) occurs when $[U^0]>[U^1]$ and (45) occurs otherwise.

C. Parallel Asynchronous SC Decoding

For traditional SC decoding of polar codes, parallel implementation is difficult, since the f-function should be calculated before the g-function since the g-function depends on the decision result from the f-function. On the other hand, our chemical SC decoder performs asynchronous decoding operations fully in parallel, i.e., all the f and g-functions are continuously calculated. To do this, we continuously perform bit decision and polar encoding at the same time as decoding. This enables the g-function to be automatically updated based on the intermediate decision result of the f-function, and eventually the decoding results converge to the target results equal to the conventional synchronous-circuit SC decoding.

V. Chemical ML Decoding

We next describe the CRN design for ML decoding, which tries to maximize the following probability

$$p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|x_i), \tag{46}$$

where x_i and y_i are the ith transmitted and received codeword bits, respectively. The optimal ML decoder finds the most-likely codeword that maximizes (46), however the efficient implementation of a max function of multiple variables with CRNs is difficult. Instead, we propose the sub-optimal ML decoding that maximizes the bit-wise likelihood, rather than symbol-wise likelihood in this paper.

The probability that the ith bit is b is calculated as follows:

$$P_i^b = \frac{\sum_{\mathbf{x}\in\mathcal{X}_i^b} p(\mathbf{y}|\mathbf{x})}{\sum_{\mathbf{x}\in\mathcal{X}_i^0} p(\mathbf{y}|\mathbf{x}) + \sum_{\mathbf{x}\in\mathcal{X}_i^1} p(\mathbf{y}|\mathbf{x})}, \tag{47}$$

where $\mathcal{X}_i^b$ is a set of codewords whose ith element is $b\in\{0,1\}$. The below describes how to calculate (47) with CRNs.

Here we take (4, 2) polar codes as an example for simplicity of explanation, whose generator matrix is given by

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \tag{48}$$

and hence possible codewords are $\{0000, 0011, 1100, 1111\}$. We first consider computing (46). In order to compute multiplication of multiple variables (more than two), we recursively perform multiplication of two variables. More specifically, multiplication of N variables consists of $\log_2 N$ layers of multiplications of two variables, i.e., $\mathsf{multiply}(P_{A_0},P_{A_1},P_{A_2},P_{A_3}) = \mathsf{multiply}(\mathsf{multiply}(P_{A_0},P_{A_1}),\mathsf{multiply}(P_{A_2},P_{A_3}))$. In this way, letting $[B^0]=[B^1]=[C^0]=[C^1]=[D^0]=[D^1]=[E^0]=[E^1]=0.5$ as initial concentrations, (46) is implemented as follows

$$P_B = \mathsf{multiply}(P^{\mathrm{c}}_{A_0},P^{\mathrm{c}}_{A_1},P^{\mathrm{c}}_{A_2},P^{\mathrm{c}}_{A_3}), \tag{49}$$
$$P_C = \mathsf{multiply}(P^{\mathrm{c}}_{A_0},P^{\mathrm{c}}_{A_1},P_{A_2},P_{A_3}), \tag{50}$$
$$P_D = \mathsf{multiply}(P_{A_0},P_{A_1},P^{\mathrm{c}}_{A_2},P^{\mathrm{c}}_{A_3}), \tag{51}$$
$$P_E = \mathsf{multiply}(P_{A_0},P_{A_1},P_{A_2},P_{A_3}), \tag{52}$$

where P_B,P_C,P_D,P_E correspond to the probability of each codeword, $0000, 0011, 1100, 1111$, respectively. Note that $P_B=[B^1]/([B^0]+[B^1])=[B^1]$ due to our initialization, and also, $P_C=[C^1]$, $P_D=[D^1]$, and $P_E=[E^1]$.

Let L_i^b correspond to the probability that ith data bit is b. Setting initial $[L_0^0]=[L_0^1]=[L_1^0]=[L_1^1]=0.5$, the bit-wise likelihood in (47) is calculated by the following reactions,

$$B^1 + L_0^1 \to B + L_0^0, \tag{53}$$
$$C^1 + L_0^1 \to C + L_0^0, \tag{54}$$
$$D^1 + L_0^0 \to D + L_0^1, \tag{55}$$
$$E^1 + L_0^0 \to E + L_0^1, \tag{56}$$
$$B^1 + L_1^1 \to B + L_1^0, \tag{57}$$
$$C^1 + L_1^0 \to C + L_1^1, \tag{58}$$
$$D^1 + L_1^1 \to D + L_1^0, \tag{59}$$
$$E^1 + L_1^0 \to E + L_1^1, \tag{60}$$

where reactions (53)–(56) correspond to the first data bit and those in (57)–(60) correspond to the second data bit. More specifically, a set of reactions (53)–(56) calculate $P_0^0=(P_B+P_C)/(P_B+P_C+P_D+P_E)$, and (57)–(60) calculate $P_1^0=(P_B+P_D)/(P_B+P_C+P_D+P_E)$.

VI. Simulation Results

In this section, we compare the performances of our chemical SC and ML decoders for short polar codes in terms of the number of required chemical species and reactions, and also the trade-off between computation accuracy and speed. To do so, we analyze the time evolution of the proposed chemical reactions governed by associated ordinary differential equations (ODE) [40]. We consider decoding polar codes defined by

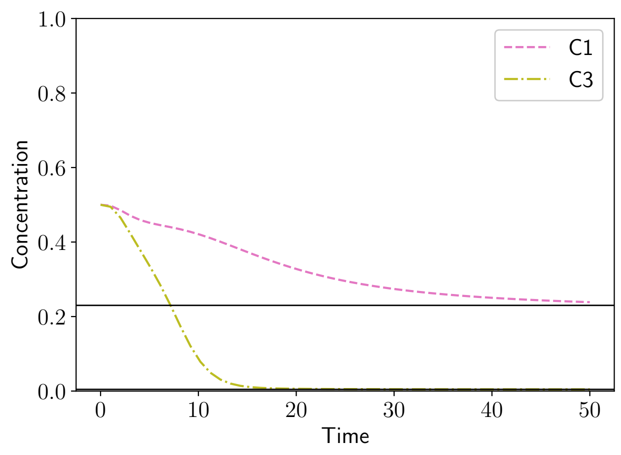

Fig. 3. Convergence behavior of the molecular concentration over time with chemical SC decoder. Only the molecular concentration corresponding to the probability of 1 is shown. Horizontal lines indicate the idealistic values.

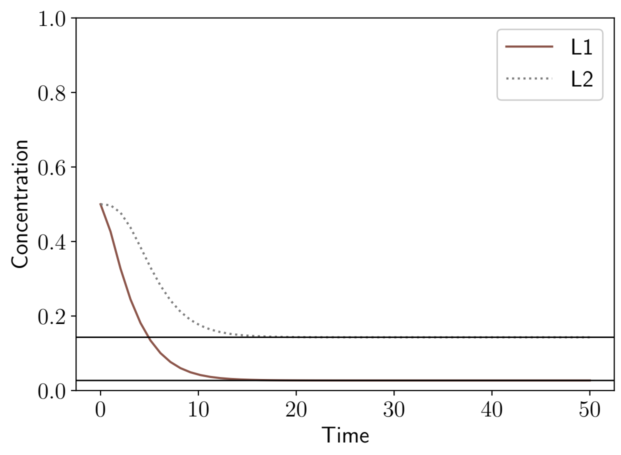

Fig. 4. Convergence behavior of the molecular concentration over time with chemical ML decoder. Only the molecular concentration corresponding to the probability of 1 is shown. Horizontal lines indicate the idealistic values.

the generator matrix given in (48). This code can be derived from (16) by setting the first and third rows to frozen indices.

A. Accuracy vs. Speed Trade-off

Fig. 3 shows the result with the proposed SC decoder, where the probability vector $A_1 = 0.2, A_2 = 0.4, A_3 = 0.1, A_4 = 0.2$ is fed into the decoder. In this work, we do not assume any specific molecular channel model and the decoder input is randomly generated. The expected decoder output in this case is $C_1 = 0.23, C_1^c = 0.77, C_3 = 0.0005, C_3^c = 0.9995$. From this figure, it is observed that molecular concentrations are converging to expected values, and hence it was demonstrated that our chemical polar decoder works well as intended. Note that the convergence speed towards the equilibrium depends on chemical reaction rates.

TABLE I
EVOLUTION OF MOLECULAR CONCENTRATION OVER TIME AND COMPARISON WITH EXPECTED VALUES.

	Molecule	0 [s]	10 [s]	20 [s]	30 [s]	Expected
SC	C1	0.500	0.420	0.325	0.272	0.249
	C3	0.500	0.078	0.006	0.005	0.005
ML	L1	0.500	0.042	0.027	0.027	0.027
	L2	0.500	0.174	0.143	0.143	0.143

TABLE II
NUMBER OF CHEMICAL REACTIONS AND SPECIES REQUIRED IN THE PROPOSED CRN-BASED SC AND ML DECODING OF HALF-RATE POLAR CODES WITH CODE LENGTH N.

	# of Reactions ($N = 4, 8, 16$)	# of Species ($N = 4, 8, 16$)
SC	222, 640, 1704	124, 356, 912
ML	44, 224, 4352	36, 152, 2608

Fig. 4 shows the result with the proposed chemical ML decoder. From (46) and (47), the expected output vector is calculated as $L_0 = 0.143, L_0^c = 0.857, L_1 = 0.027, L_1^c = 0.973$. We can verify that the molecular concentrations converge to the expected values in Fig. 4. Rigorous stability analysis and feasibility for in-vivo scenarios are left as future work.

Table I summarizes the evolution of molecular concentration over time in SC and ML decoding. From this table, we observe that for both SC and ML decoding CRNs, molecular concentrations can converge to expected values as time proceeds. It is also observed that the convergence of molecule C_2 of SC decoding is slower than that of ML decoding, which stems from the delay associated with the decision of the previous bit used in the g-function.

B. Number of Chemical Species and Reactions

Finally, we briefly compare two decoding schemes in terms of the required number of chemical species and reactions. Table II shows the number of chemical reactions and species required in the proposed SC and ML decoding for half-rate polar codes with code lengths of $N = 4, 8, 16$. From these results, we can see that the ML decoder requires much fewer chemical reactions and species even with the relatively higher-speed convergence when the code length is very short $N = 4$, whereas it increases exponentially as the code length increases. Note that if more species are involved, designing the DNA strand will be more challenging due to fundamental errors. We conclude from this result that ML decoding is better when the code length is very small or faster processing speed is required, however, SC decoding may be suited when the code length is longer, e.g., greater than $N = 8$ in terms of the CRN-based circuit size.

VII. CONCLUSION

In this paper, we have proposed a new polar coding scheme based on molecular programming. Our polar coding does not rely on timing, and thus enables highly parallel implementation. Two decoding schemes for short polar codes based on SC and ML decoding have been investigated. ODE simulation results demonstrated that our proposed CRNs for SC and

ML decoders can realize fully-parallel and accurate decoding through chemical reactions without needing an external power supply, while the ML decoder requires much fewer chemical species and reactions when the code length is very short. To the best of our knowledge, our research is the first attempt to implement polar encoding and decoding via CRNs, which can be used for intra-body molecular communications without electric power supplies. More evaluations over realistic channel models will be required for feasibility analysis. We have focused on a specific short polar code and hence generalization to longer codes should be studied. Future work also includes the design of DNA strands that realize our chemical polar decoder. The *cello* [41] may be useful for programming our chemical polar decoders via synthetic genetic circuits.

References

[1] A. Hjelmfelt, E. D. Weinberger, and J. Ross, "Chemical implementation of neural networks and turing machines," *Proceedings of the National Academy of Sciences*, vol. 88, no. 24, pp. 10 983–10 987, 1991.

[2] J. Kim, J. Hopfield, and E. Winfree, "Neural network computation by in vitro transcriptional circuits," in *Advances in neural information processing systems*, pp. 681–688, 2005.

[3] L. Qian, E. Winfree, and J. Bruck, "Neural network computation with DNA strand displacement cascades," *Nature*, vol. 475, no. 7356, p. 368, 2011.

[4] A. Shea, B. Fett, M. D. Riedel, and K. Parhi, "Writing and compiling code into biochemistry," in *Proc. the Pacific Symposium on Biocomputing*, pp. 456–464, 2010.

[5] H. Jiang, M. Riedel, and K. Parhi, "Synchronous sequential computation with molecular reactions," in *Proceedings of the 48th Design Automation Conference*, pp. 836–841. ACM, 2011.

[6] H. Jiang, M. D. Riedel, and K. K. Parhi, "Digital logic with molecular reactions," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 721–727, 2013.

[7] P. Senum and M. Riedel, "Rate-independent constructs for chemical computation," *PloS one*, vol. 6, no. 6, 2011.

[8] L. Cardelli, M. Kwiatkowska, and M. Whitby, "Chemical reaction network designs for asynchronous logic circuits," *Natural computing*, vol. 17, no. 1, pp. 109–130, 2018.

[9] N. E. Napp and R. P. Adams, "Message passing inference with chemical reaction networks," in *Advances in Neural Information Processing Systems*, pp. 2247–2255, 2013.

[10] Z. Xingchi, G. Lulu, Y. Xiaohu, and Z. Chuan, "Synthesizing LDPC belief propagation decoding with molecular reactions," in *Proc. 2018 IEEE International Conference on Communications (ICC)*, Jul. 2018.

[11] D. Soloveichik, M. Cook, E. Winfree, and J. Bruck, "Computation with finite stochastic chemical reaction networks," *natural computing*, vol. 7, no. 4, pp. 615–633, 2008.

[12] S. W. Shin, "Compiling and verifying DNA-based chemical reaction network implementations," Ph.D. dissertation, California Institute of Technology, 2012.

[13] D. Soloveichik, G. Seelig, and E. Winfree, "DNA as a universal substrate for chemical kinetics," *Proceedings of the National Academy of Sciences*, vol. 107, no. 12, pp. 5393–5398, 2010.

[14] T. Nakano, M. J. Moore, F. Wei, A. V. Vasilakos, and J. Shuai, "Molecular communication and networking: Opportunities and challenges," *IEEE Trans. Nanobiosci.*, vol. 11, no. 2, pp. 135–148, 2012.

[15] T. Suda, M. Moore, T. Nakano, R. Egashira, and A. Enomoto, "Exploratory research on molecular communication between nanomachines," in *ACM Genetic and Evol. Computat. Conf.*, 2005.

[16] T. N. annd A. Eckford and T. Haraguchi, *Molecular Communication*. Cambridge University Press, 2013.

[17] I. Akyildiz, F. Brunetti, and C. Blazquez, "Nanonetworks: A new communication paradigm," *Comput. Netw.*, vol. 52, 2008.

[18] Y. Moritani, S. Hiyama, and T. Suda, "Molecular communication for health care applications," in *IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2006.

[19] J. Suzuki, D. H. Phan, and H. Budiman, "A nonparametric stochastic optimizer for tdma-based neuronal signaling," *IEEE Trans. NanoBioscience*, vol. 13, no. 3, pp. 244–254, 2014.

[20] M. Pierobon and I. F. Akyildiz, "Diffusion-based noise analysis for molecular communication in nanonetworks," *IEEE Trans. SP*, vol. 59, no. 6, 2011.

[21] K. V. Srinivas, R. S. Adve, and A. W. Eckford, "Molecular communication in fluid media: The additive inverse gaussian noise channel," *IEEE Trans. IT*, vol. 58, no. 7, 2012.

[22] N. Farsad, A. Eckford, S. Hiyama, and Y. Moritani, "A simple mathematical model for information rate of active transport molecular communication," in *IEEE INFOCOM*, 2011.

[23] M. J. Moore and T. Suda, "Molecular communication: Modeling noise effects on information rate," *IEEE Trans. NanoBiosci.*, vol. 8, no. 2, 2009.

[24] J. S. Mitzman, B. Morgan, T. M. Soro, J. Suzuki, and T. Nakano, "A feedback-based molecular communication protocol for noisy intrabody environments," in *IEEE Int. Conf. E-health Networking Applications and Services*, 2015.

[25] T. Nakano, Y. Okaie, and A. V. Vasilakos, "Transmission rate control for molecular communication among biological nanomachines," *IEEE JSAC*, vol. 31, no. 12, pp. 835–846, 2013.

[26] L. Felicetti, M. Femminella, G. Reali, T. Nakano, and A. V. Vasilakos, "Tcp-like molecular communications," *IEEE JSAC*, vol. 32, no. 12, pp. 2354–2367, 2014.

[27] X. Wang, M. D. Higgins, and M. S. Leeson, "Simulating the performance of sw-arq schemes within molecular communications," *Simulation Modelling Practice and Theory*, vol. 42, 2014.

[28] C. Bai, M. S. Leeson, and M. D. Higgins, "Performance of sw-arq in bacterial quorum communications," *Nano Commun. Netw.*, vol. 6, no. 1, 2015.

[29] M. S. Leeson and M. D. Higgins, "Forward error correction for molecular communications," *Nano Commun. Netws*, vol. 3, no. 3, pp. 161–167, 2012.

[30] ——, "Error correction coding for molecular communications," in *IEEE ICC*, pp. 6172–6176, 2012.

[31] Y. Lu, M. D. Higgins, and M. S. Leeson, "Self-orthogonal convolutional codes (soccs) for diffusion-based molecular communication systems," in *IEEE ICC*, pp. 1049–1053, 2015.

[32] C. Bai, M. S. Leeson, and M. D. Higgins, "Minimum energy channel codes for molecular communications," vol. 50, no. 23, pp. 1669–1671, 2014.

[33] H. Egashira, J. Suzuki, J. S. Mitzman, T. Nakano, and H. Fukuda, "Robust directional-diffusive hybrid molecular communication with parity-check erasure coding," in *IFSA-SCIS*, 2017.

[34] H. Egashira, J. Suzuki, T. Koike-Akino, T. Nakano, H. Fukuda, and P. Orlik, "Molecular fountain: A robustness enhancement framework for diffusive molecular communication," in *IEEE GLOBECOM*, pp. 1–7. IEEE, 2017.

[35] G. Liva, L. Gaudio, T. Ninacs, and T. Jerkovits, "Code design for short blocks: A survey," *arXiv:1610.00873*, Oct. 2016.

[36] M. Shirvanimoghaddam, M. S. Mohamadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, Y. Li, S. Johnson, and B. Vucetic, "Short block-length codes for ultra-reliable low-latency communications," *arXiv:1802.09166*, Sep. 2018.

[37] Z. Zhong, L. Ge, Z. Zhang, X. You, and C. Zhang, "Molecular polar belief propagation decoder and successive cancellation decoder," in *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*, pp. 236–241. IEEE, 2019.

[38] T. Matsumine, T. Koike-Akino, and Y. Wang, "Polar coding with chemical reaction networks," *arXiv preprint arXiv:1903.03709*, 2019.

[39] S. S. Tehrani, W. J. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 10, pp. 716–718, 2006.

[40] B. Dahlgren, "ChemPy: A package useful for chemistry written in Python," *Journal of Open Source Software*, vol. 3, no. 24, p. 565, 2018.

[41] A. A. Nielsen, B. S. Der, J. Shin, P. Vaidyanathan, V. Paralanov, E. A. Strychalski, D. Ross, D. Densmore, and C. A. Voigt, "Genetic circuit design automation," *Science*, vol. 352, no. 6281, p. aac7341, 2016.