

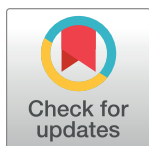
RESEARCH ARTICLE

# A concatenated LDPC-marker code for channels with correlated insertion and deletion errors in bit-patterned media recording system

Tianbo Xue  \*

Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

\* [tianbo.xue@connect.polyu.hk](mailto:tianbo.xue@connect.polyu.hk)



## Abstract

Most synchronization error correction codes deal with random independent insertion and deletion errors without correlation. In this paper, we propose a probabilistic channel model with correlated insertion and deletion (CID) errors to capture the data dependence applicable to the bit-patterned media recording (BPMR) system. We also investigate the error performance and decoding complexity of a concatenated LDPC-marker code over the CID channel. Furthermore, we modify the forward backward decoding algorithm to make it suitable for the CID channel, and elaborate it based on a two-dimensional state transition diagram. Compared with the conventional marker coding scheme dealing with random errors, the concatenated LDPC-marker code takes into account the dependence between synchronization errors, improves the error performance, and reduces the decoding complexity. The BER performance of the concatenated LDPC-marker code is improved by more than 50% on average, and the decoding time is reduced by nearly 35% when the LDPC code ( $n = 4521$ ,  $k = 3552$ ) and the marker code ( $N_m = 2$ ,  $N_c = 30$ ) are used over the CID channel.

## OPEN ACCESS

**Citation:** Xue T (2022) A concatenated LDPC-marker code for channels with correlated insertion and deletion errors in bit-patterned media recording system. PLoS ONE 17(7): e0270247. <https://doi.org/10.1371/journal.pone.0270247>

**Editor:** Muhammad Zeeshan, National University of Sciences and Technology, PAKISTAN

**Received:** March 9, 2021

**Accepted:** June 7, 2022

**Published:** July 8, 2022

**Copyright:** © 2022 Tianbo Xue. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** All relevant data are within the paper.

**Funding:** The work was supported in part by the Hong Kong PhD Fellowship PF14-13269. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors have declared that no competing interests exist.

## 1. Introduction

Channels impaired by insertion, deletion and substitution errors were first considered in 1961 [1]. Insertions and deletions are categorized as synchronization errors because they affect the synchronization of a communication channel. Recently, channels with synchronization errors have attracted renewed interest because of applications such as DNA-based data storage [2], image watermarking [3], and video digital watermarking [4]. Conventional error correction codes such as low-density parity-check (LDPC) codes [5] and Reed-Solomon codes [6], which are designed to combat noise and substitutions, cannot correct insertions and/or deletions.

Many coding schemes have been proposed for channels impaired by random insertion, deletion and substitution errors in the past few decades [7]. Synchronization error correction mechanisms have shown promising results by adopting a concatenated scheme. An LDPC code with inserted markers can correct multiple insertions and deletions in [8]. RS codes are

concatenated with marker codes in [9] to correct residual errors in the output from the marker decoder. The authors in [10] constructed a family of synchronization error correction codes through concatenation, and provided efficient encoding and decoding algorithms. A concatenated LDPC-trellis code with a high information rate is constructed in [11]. In [12], the authors proposed a coding scheme consisting of a 2-D marker code and an irregular LDPC code to achieve excellent decoding performance under a non-iterative scenario against insertions and deletions in racetrack memories.

However, the concatenated coding schemes mentioned above are developed for channels with random independent errors. Very few papers focus on codes correcting correlated errors. In fact, correlated insertion and deletion errors exist in some practical applications, such as BPMR systems. In BPMR, deletions and insertions often occur in pairs (i.e., a deletion is followed by an insertion in a latter position, and vice-versa) [13]. In particular, experiments are conducted to provide written-in error statistics and experimental data is provided, where synchronization is first lost due to an insertion and then resynchronize following a deletion, and vice-versa. It is further noted that the subsequent synchronization error is more likely to occur at the position closer to the preceding one and the error probability decreases with the distance. From these circumstances, insertions and deletions are correlated rather than independent and random in the writing process of BPMR.

Several channel models suitable for the writing process of BPMR have been proposed. Dependence of written-in errors was considered by Mazumdar et al. in [14], but the only case studied was a bit erroneously changed to the preceding value when binary data was written on the medium. The authors in [15] proposed a write channel model with input-dependent noise for BPMR, and theoretically analyzed the lower and upper bounds of the information rate. A recent paper [16] presented a channel model with correlated insertion and deletion errors adapted to BPMR systems. The channel model is fully described by a finite-state machine, i.e., the current event (insertion, deletion or transmission) is completely dependent on the previous event. Although the occurrence of consecutive insertion/deletion errors is prohibited, it does not guarantee the insertion event and the deletion event to appear in pairs. Moreover, the insertion and deletion error probability after a synchronization error is fixed to 0.5, which does not fit the fact that the subsequent synchronization error is more likely to occur at a closer position of the preceding one in BPMR.

In BPMR writing process, 1) a deletion is followed by an insertion in a latter position, and vice-versa; 2) the subsequent synchronization error probability is more likely to occur at the position closer to the preceding one and decreases with the distance. As far as we know, no channel models have been proposed for BPMR systems that considers both of the above error statistics. For the above reasons, we propose a more accurate channel model that takes into account all error statistics in BPMR writing process.

In this paper, we first propose a probabilistic channel model with correlated insertion and deletion errors, in addition to substitution errors. The channel model captures, for example, the data dependence adapted to the write channel in BPMR more accurately. Then, we modify the forward-backward algorithm to make it suitable for the proposed channel. Finally, we compare the error performance of the concatenated LDPC-marker coding scheme with the conventional marker coding scheme without considering the dependency between synchronization errors over the proposed channel. The paper is organized as follows. Sect. II introduces the proposed channel model, the concatenated LDPC-marker code, and the encoding and decoding steps. Sect. III describes the details of the modified forward-backward algorithm based on a two-dimensional transition diagram. Sect. IV presents simulation results and discussion. Concluding remarks and future work are given in Sect. V.

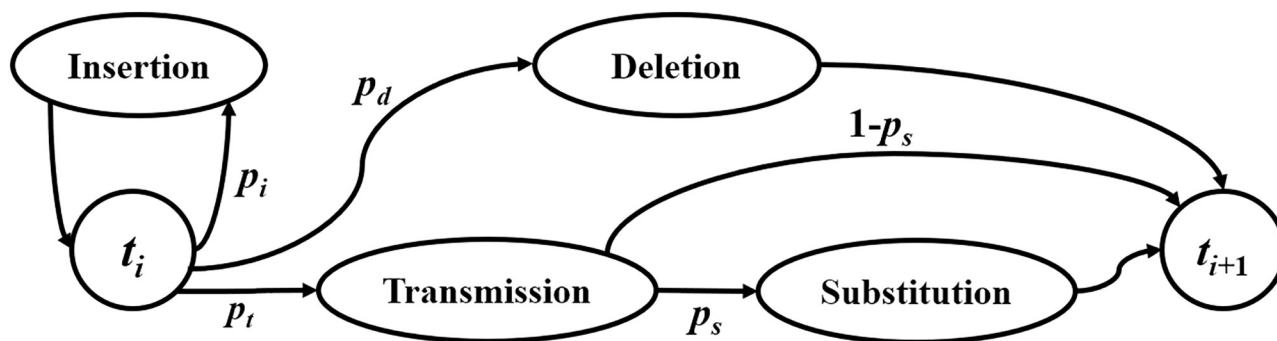


Fig 1. BSID channel model.

<https://doi.org/10.1371/journal.pone.0270247.g001>

## II. Concatenated LDPC-marker code

This section starts with a binary channel with random errors and the proposed probabilistic channel model with correlated insertion and deletion errors, followed by a description of the encoding and decoding of the concatenated LDPC-marker code.

### BSID channel with random errors

A channel with random independent insertion, deletion and substitution errors proposed by Davey and MacKay [17] is depicted in Fig 1. For the binary case, this channel is referred to as binary substitution/insertion/deletion (BSID) channel. It can be regarded as a binary symmetric channel (BSC) with synchronization errors. For each transmitted bit  $x_k$ , three possible events may occur: (1)  $x_k$  is deleted with probability  $p_d$ ; (2) one more bit is transmitted before  $x_k$  with probability  $p_i$ ; (3)  $x_k$  is transmitted with probability  $p_t = 1 - p_d - p_i$ . Moreover, the transmitted bit may be substituted with probability  $p_s$ . The operation repeats until all bits are sent. Insertions and deletions are random without correlation in this channel.

### Proposed CID channel with correlated errors

Here, we propose a probabilistic channel with correlated insertion and deletion (CID) errors, i.e., an insertion and a deletion occur in pairs while substitution errors occur randomly within a sequence. Such a channel is referred to as a correlated insertion/deletion (CID) channel.

Given a transmitted sequence  $\mathbf{x}_1^t = (x_1, x_2, \dots, x_t) \in \{0, 1\}^t$ , we assume synchronization errors occur to  $x_{a_1}, x_{b_1}, x_{a_2}, x_{b_2}, x_{a_3}, x_{b_3}, \dots$  where  $a_1 < b_1 < a_2 < b_2 < a_3 < b_3 < \dots$ . We also call  $(x_{a_n}, x_{b_n})$  ( $n = 1, 2, \dots$ ) a synchronization error bit pair.

The characteristics of the proposed probabilistic channel model are as follows.

1. The synchronization error probability of  $x_{a_n}$  is determined by the blackBSID channel.
2. We assume  $x_{a_n}$  and  $x_{b_n}$  suffer from different synchronization errors. In other words, if  $x_{a_n}$  suffers from an insertion error,  $x_{b_n}$  will suffer from a deletion error; and vice versa.
3. The synchronization error probability of  $x_{b_n}$  is determined by the synchronization error type of  $x_{a_n}$  and the bit separation  $l$  between  $x_{a_n}$  and  $x_{b_n}$ . The synchronization error probability of  $x_{b_n}$  decreases as the separation  $l$  increases.

We consider the synchronization error bit pair  $(x_{a_n}, x_{b_n})$ . After a synchronization error has occurred to  $x_{a_n}$ , we denote the probability of the synchronization error occurring at  $x_{a_n+l}$

by  $\Pr(x_{b_n} = x_{a_n+l})$ . Since  $\Pr(x_{b_n} = x_{a_n+l})$  is a decreasing function of  $l$ , we assume that  $\Pr(x_{b_n} = x_{a_n+l})$  ( $l = 1, 2, \dots$ ) forms a geometric progression with a common ratio of  $r < 1$ , i.e.,

$$\Pr(x_{b_n} = x_{a_n+l}) = Ar^{l-1}; l = 1, 2, \dots \quad (1)$$

where  $A$  is a constant. After a synchronization error has occurred to  $x_{a_n}$ , we use  $p'_d$ ,  $p'_i$  and  $p'_t$  to represent the new deletion probability, new insertion probability and new transmission probability, respectively, for  $x_{a_n+l}$ . Moreover, they are given by Eqs (2) to (4). The probabilities will be reset to  $p_d$ ,  $p_i$  and  $p_t$  once a synchronization error occurs to  $x_{b_n}$ . We further assume that random substitution errors always exist with an error probability  $p_s$ . The above operation repeats until all transmitted bits are sent.

$$p'_d = \begin{cases} Ar^{l-1} & \text{if an insertion error has occurred to } x_{a_n} \\ 0 & \text{if a deletion error has occurred to } x_{a_n} \end{cases} \quad (2)$$

$$p'_i = \begin{cases} 0 & \text{if an insertion error has occurred to } x_{a_n} \\ Ar^{l-1} & \text{if a deletion error has occurred to } x_{a_n} \end{cases} \quad (3)$$

$$p'_t = 1 - p'_i - p'_d = 1 - Ar^{l-1}. \quad (4)$$

### Encoder

The presented scheme consists of an outer LDPC code and an inner marker code, as shown in Fig 2. The outer code is used to correct errors while the role of the inner marker code is to maintain synchronization. Encoding is implemented in two steps: the message is first encoded into an outer LDPC code and then regular markers [18] with length  $N_m$  are inserted periodically every  $N_c$  LDPC code bits. For an LDPC code with length  $N$ , the regularly inserted markers

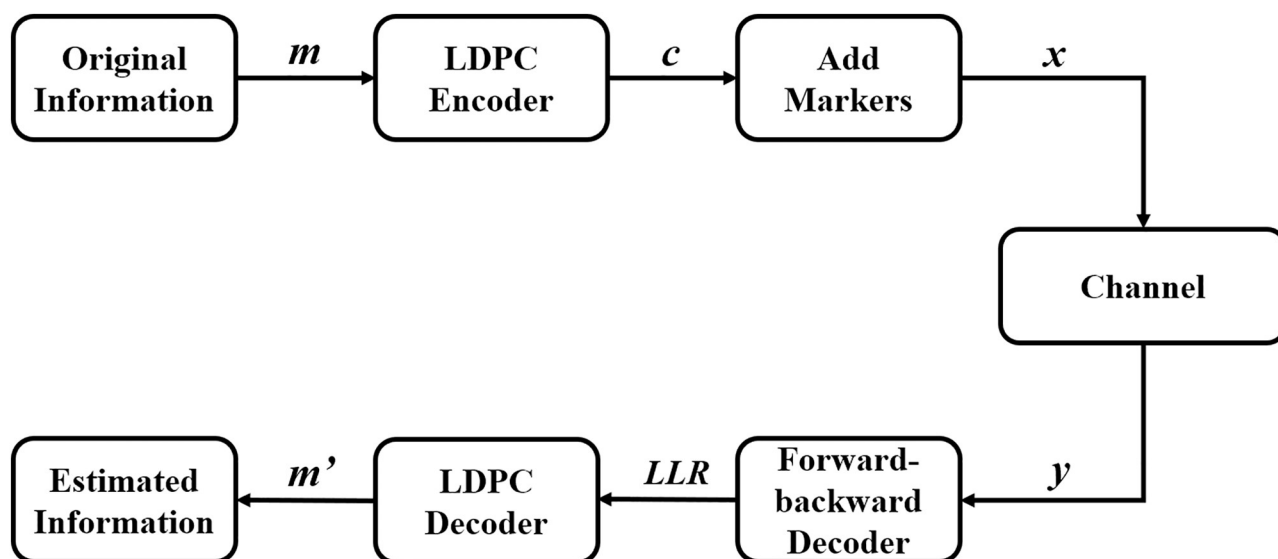


Fig 2. Flow chart of the encoding and decoding of the concatenated LDPC-marker code.

<https://doi.org/10.1371/journal.pone.0270247.g002>

divide the LDPC code into  $N/N_c$  segments (assuming  $N/N_c$  is an integer). Each of the segment consists of  $N_c + N_m$  bits where the first  $N_m$  bits are the marker bits and the last  $N_c$  bits are the LDPC code bits. The content and position of the periodical markers in the transmitted sequence are known to and used by the receiver to maintain synchronization. The encoded sequence after the two encoding steps is ready to be transmitted through the channel. The code rate of the concatenated code is  $R = R_C \cdot R_M$ , where  $R_C$  and  $R_M = \frac{N_c}{N_c + N_m}$  are the code rates of the LDPC code and the marker code, respectively.

### Inner marker decoder

We denote  $\mathbf{x}_1^t = (x_1, x_2, \dots, x_t)$ ,  $x_i \in \{0, 1\}$  for  $i = 1, 2, \dots, t$  and  $\mathbf{y}_1^r = (y_1, y_2, \dots, y_r)$ ,  $y_i \in \{0, 1\}$  for  $i = 1, 2, \dots, r$  as the transmitted and the received sequences, respectively. At the receiver, the corrupted sequence  $\mathbf{y}_1^r$  is first passed into the inner marker decoder. Given the received vector  $\mathbf{y}_1^r$ , the inner marker decoder makes full use of the information provided by the markers to compute the likelihood  $p(\mathbf{y}_1^r | x_k)$  for  $x_k \in \{0, 1\}$  and  $k = 1, 2, \dots, t$ . To derive the likelihood  $p(\mathbf{y}_1^r | x_k)$ , we modify the forward-backward algorithm [19], which will be explained in Sect. IV. The log-likelihood ratio (LLR) value  $L(x_k | \mathbf{y}_1^r)$  is the ratio between the conditional probabilities  $\Pr(x_k = 0 | \mathbf{y}_1^r)$  and  $\Pr(x_k = 1 | \mathbf{y}_1^r)$ . Let  $\mathcal{L}$  be the set of positions where the outer LDPC code bits are located in the transmitted sequence. Applying Bayes' rule, the LLR  $L(x_k | \mathbf{y}_1^r)$  at bit position  $k \in \mathcal{L}$  can be computed using

$$L(x_k | \mathbf{y}_1^r) = \ln \frac{\Pr(x_k = 0 | \mathbf{y}_1^r)}{\Pr(x_k = 1 | \mathbf{y}_1^r)} = \ln \frac{p(\mathbf{y}_1^r | x_k = 0)}{p(\mathbf{y}_1^r | x_k = 1)} \quad k \in \mathcal{L} \quad (5)$$

which are used as soft inputs to the outer LDPC decoder for further decoding.

### Outer LDPC decoder

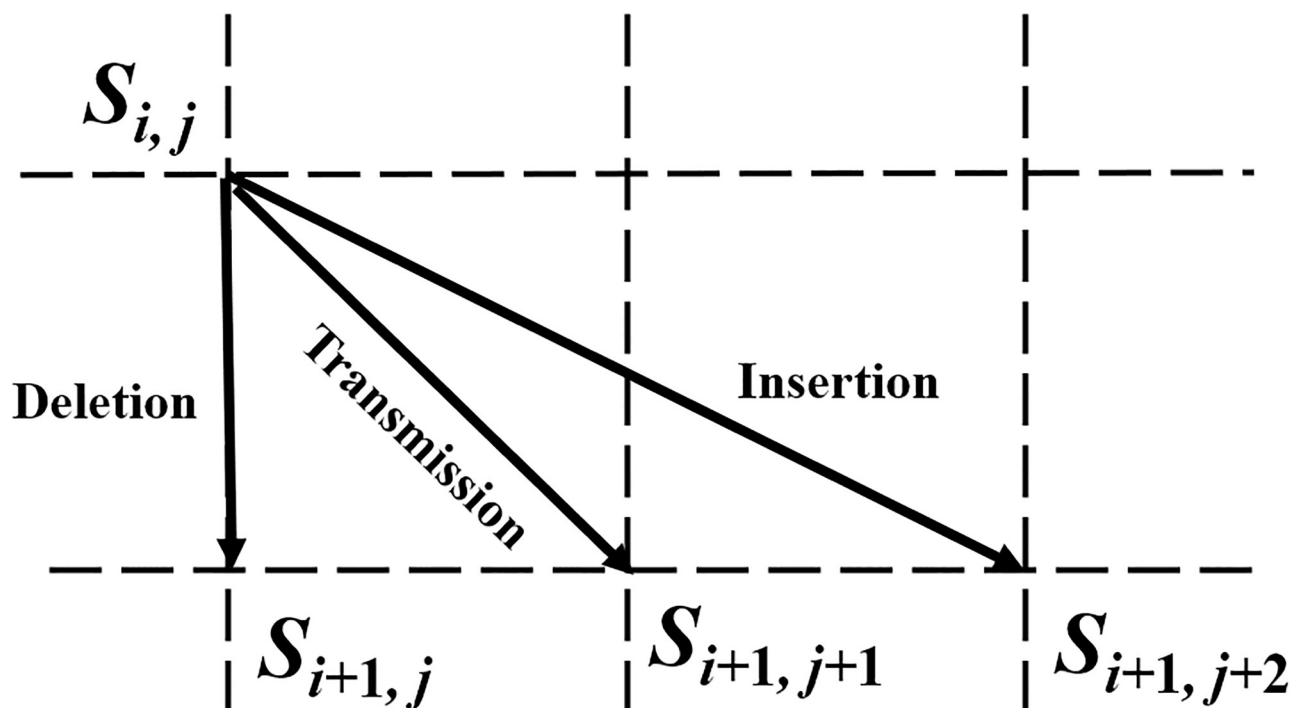
LDPC code is a linear block code with a sparse parity-check matrix  $H$  in which the number of non-zero entries is small. We apply the sum-product decoding algorithm (also called belief propagation algorithm) to decode the LDPC code by iteratively passing updated LLR messages along the edges between variable nodes and check nodes in the Tanner graph corresponding to  $H$ . At the end of each iteration, the updated *a posteriori* LLR for each transmitted bit is calculated and an estimated codeword is obtained by making hard decisions based on the LLRs. The LDPC decoder subsequently checks whether the estimated codeword satisfies all check equations. The codeword is successfully decoded if all check equations are satisfied; otherwise, the iteration continues. If the maximum number of iterations is reached and not all check equations are satisfied, the decoding fails.

## III. Modified forward-backward algorithm

In this section, we modify the forward-backward algorithm to make it suitable for the proposed CID channel, taking into account the correlated insertion and deletion errors. We further elaborate a computationally efficient marker decoding algorithm based on a two-dimensional state transition diagram. Finally, the inner marker decoder implements the modified FB algorithm to compute the LLR to initialize the outer LDPC decoder.

### State transition diagram

We define  $S_{0,0}$  as the initial state when the transmission begins; and  $S_{t,r}$  as the end state when  $t$  bits have been sent and  $r$  bits have been received. We also define an intermediate state  $S_{i,j}$  when  $i$  bits ( $i = 1, 2, \dots, t$ ) have been sent and  $j$  bits ( $j = 0, 1, \dots, r$ ) have been received. State  $S_{i,j}$



**Fig 3.** State transition paths for insertion, deletion and transmission when  $x_{i+1}$  is transmitted on a two-dimensional grid diagram.  $i$  increases downwards and  $j$  increases rightwards.

<https://doi.org/10.1371/journal.pone.0270247.g003>

may transit to  $S_{i+1,j}$ ,  $S_{i+1,j+1}$  and  $S_{i+1,j+2}$  if there is a deletion error, no deletion/insertion error, and an insertion error, respectively, for the transmitted bit  $x_{i+1}$ . The transitions are illustrated in Fig 3. When the state transition paths corresponding to individual transmitted bits are connected together, a complete transmission path like the one shown in Fig 4 is formed. For our CID channel, the complete state transmission path is bounded between the lower boundary ( $i - j = 1$ ) and the upper boundary ( $j - i = 1$ ).

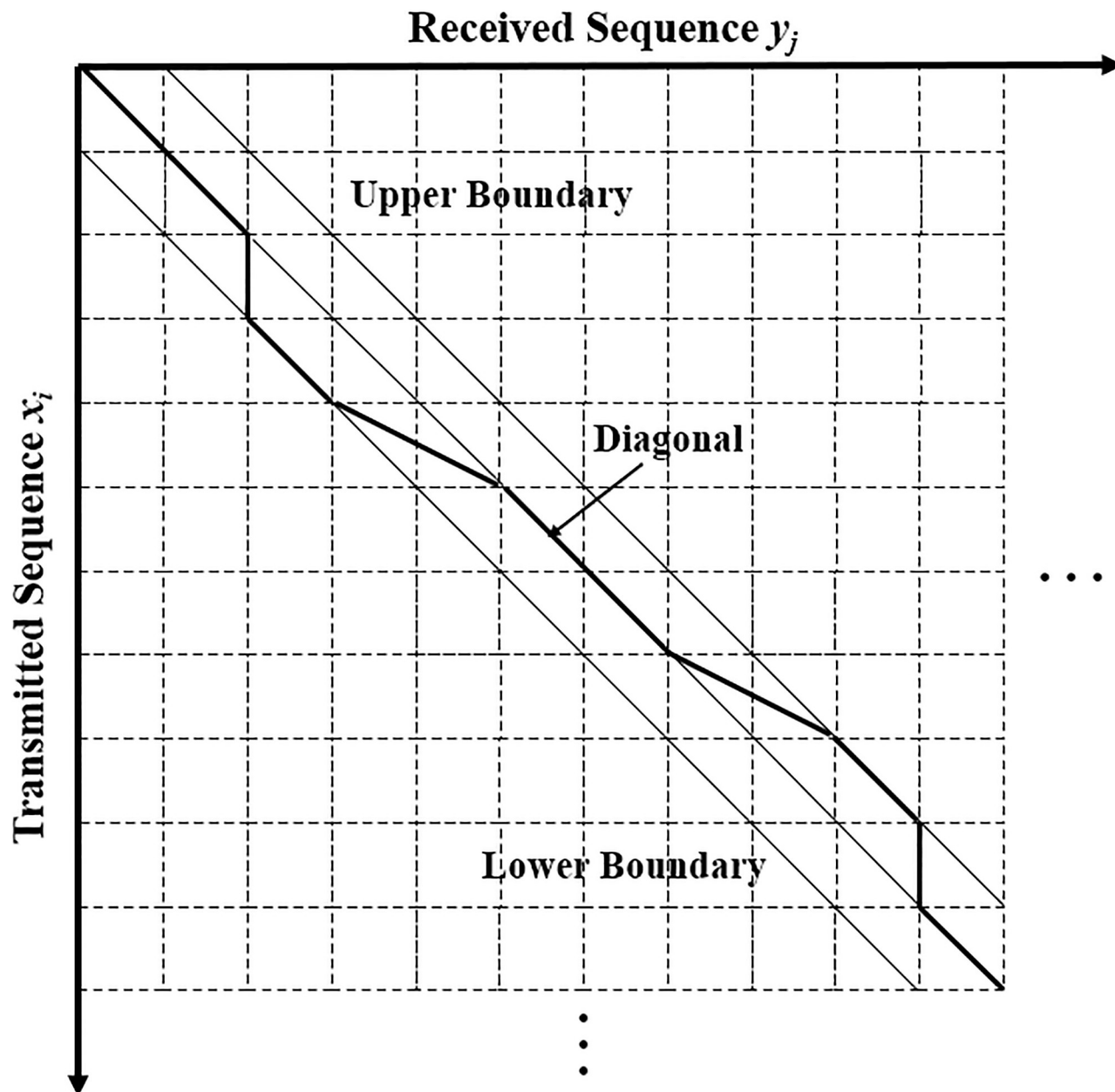
### Forward algorithm

A transmitted bit  $x_i$  is either an LDPC code bit or a marker bit. For an LDPC code bit  $x_i$ ,  $\Pr(x_i = 0) = \Pr(x_i = 1) = 0.5$  because no information about an LDPC code bit  $x_i$  is provided to the decoder and the receiver does not know the value of  $x_i$ . However, when  $x_i$  is a bit from the marker,  $\Pr(x_i)$  is a determined value, i.e., (a)  $\Pr(x_i = 0) = 0$  and  $\Pr(x_i = 1) = 1$ ; or (b)  $\Pr(x_i = 0) = 1$  and  $\Pr(x_i = 1) = 0$ . The reason is that the exact value and position of a marker bit  $x_i$  are accurately known to the receiver.

We define  $T(x_i, y_j)$  as the transmission probability that  $x_i$  is transmitted (with/without insertion) and the corresponding received bit is  $y_j$ , i.e.,

$$T(x_i, y_j) = \Pr(x_i) \cdot C(x_i, y_j), \quad (6)$$

where  $C(x_i, y_j)$  denotes the probability that the transmitted bit  $x_i$  and the corresponding received bit  $y_j$  have or do not have the same value. If  $x_i = y_j$ ,  $x_i$  has been transmitted without any substitution error and thus  $C(x_i, y_j) = 1 - p_s$ ; otherwise,  $x_i$  is substituted and  $C(x_i, y_j) = p_s$ .



**Fig 4.** A complete transmission path representing a one-to-one mapping between the received and transmitted sequence on a two-dimensional grid diagram. The actual transmission path represented by the bold line over the CID channel is between the lower and upper boundaries.

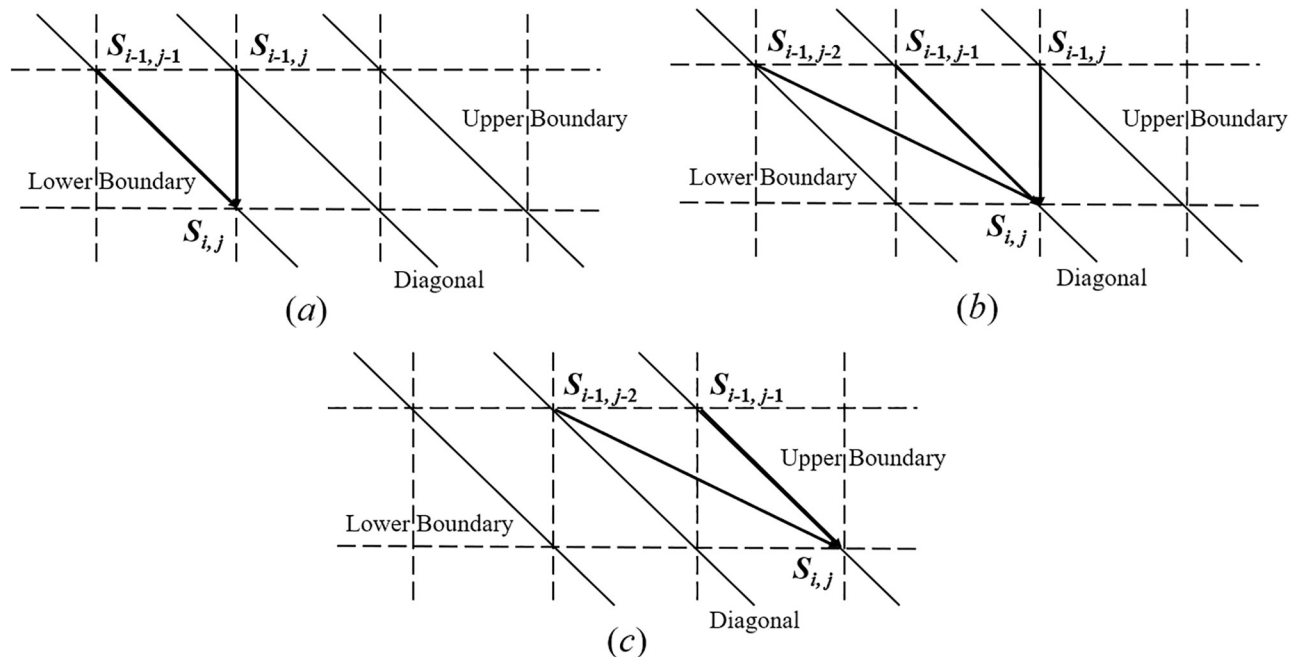
<https://doi.org/10.1371/journal.pone.0270247.g004>

Thus, we have

$$C(x_i, y_j) = \begin{cases} 1 - p_s & \text{if } x_i = y_j \\ p_s & \text{if } x_i \neq y_j \end{cases}. \quad (7)$$

The forward quantity  $\alpha_{i,j}$  is the joint probability that  $i$  bits are transmitted and the  $j$  bits  $y_1^j = (y_1, y_2, \dots, y_j)$  are received. In other words, it is the probability that  $y_1^j$  has been received





**Fig 5. Possible state transitions when  $x_k$  is transmitted on condition that  $S_{i,j}$  occurs.** (a)  $S_{i,j}$  is located on the lower boundary ( $i - j = 1$ ); (b)  $S_{i,j}$  is located on the diagonal ( $i - j = 0$ ); (c)  $S_{i,j}$  is located on the upper boundary ( $j - i = 1$ ).

<https://doi.org/10.1371/journal.pone.0270247.g005>

when the transmission state transits from  $S_{0,0}$  to  $S_{i,j}$ . The forward quantity  $\alpha_{i,j}$  is given by

$$\alpha_{i,j} = \Pr(y_1^j, S_{i,j}). \quad (8)$$

Suppose  $x_i$  is transmitted over the CID channel. According to the relationship between  $i$  and  $j$  and the location of the previous state, three scenarios are to be considered in order to calculate  $\alpha_{i,j}$  as shown in Fig 5.

**Case 1:  $S_{i,j}$  is located on the lower boundary when  $i - j = 1$  and it can be reached from  $S_{i-1,j-1}$  and  $S_{i-1,j}$ , as shown in Fig 5(a).**

$$\alpha_{i,j} = p_d \alpha_{i-1,j} + p'_i \alpha_{i-1,j-1} T(x_i, y_j). \quad (9)$$

**Case 2:  $S_{i,j}$  is located on the diagonal when  $i - j = 0$  and it can be reached from  $S_{i-1,j-2}$ ,  $S_{i-1,j-1}$  and  $S_{i-1,j}$ , as shown in Fig 5(b).**

$$\begin{aligned} \alpha_{i,j} &= p'_d \alpha_{i-1,j} + p_i \alpha_{i-1,j-1} T(x_i, y_j) + p'_i \alpha_{i-1,j-2} T(x_i, y_j) \Pr(\text{inserted bit} = y_{j-1}) \\ &= p'_d \alpha_{i-1,j} + p_i \alpha_{i-1,j-1} T(x_i, y_j) + \frac{p'_i}{2} \alpha_{i-1,j-2} T(x_i, y_j). \end{aligned} \quad (10)$$

**Case 3:  $S_{i,j}$  is located on the upper boundary when  $j - i = 1$  and it can be reached from either  $S_{i-1,j-2}$  or  $S_{i-1,j-1}$ , as shown in Fig 5(c).**

$$\begin{aligned} \alpha_{i,j} &= p'_i \alpha_{i-1,j-1} T(x_i, y_j) + p_i \alpha_{i-1,j-2} T(x_i, y_j) \Pr(\text{inserted bit} = y_{j-1}) \\ &= p'_i \alpha_{i-1,j-1} T(x_i, y_j) + \frac{p_i}{2} \alpha_{i-1,j-2} T(x_i, y_j). \end{aligned} \quad (11)$$



In summary, given the initial conditions  $\alpha_{0,0} = 1$  and  $\alpha_{0,1} = 0$ , all other values of  $\alpha_{i,j}$  within the boundaries in Fig 4 can be calculated iteratively.

### Backward algorithm

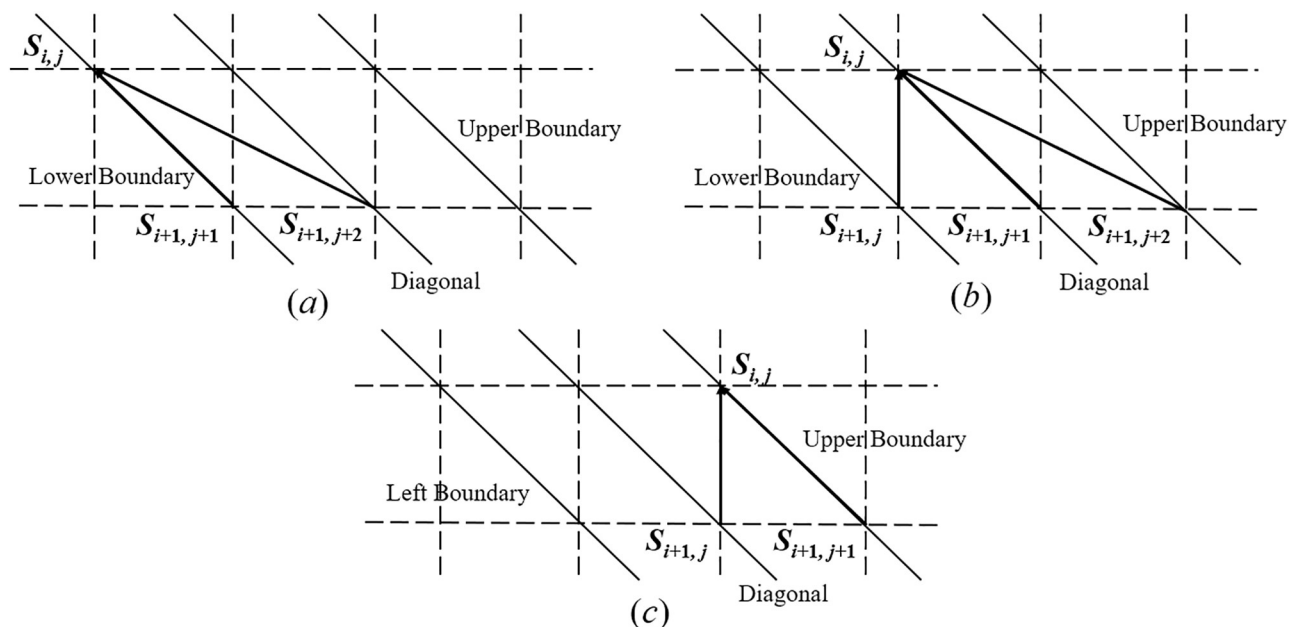
Similarly, the backward quantity  $\beta_{i,j}$  denotes the probability that the remaining  $r-j$  received bits are  $\mathbf{y}_{j+1}^r = (y_{j+1}, \dots, y_r)$  given the transmission state  $S_{i,j}$  has occurred. The backward quantity  $\beta_{i,j}$  is therefore denoted by

$$\beta_{i,j} = \Pr(\mathbf{y}_{j+1}^r | S_{i,j}). \quad (12)$$

Suppose  $x_{i+1}$  is transmitted through CID channel. According to the relationship between  $i$  and  $j$ , three scenarios are to be considered in order to calculate  $\beta_{i,j}$  as shown in Fig 6. Moreover, the channel error types and accompanying synchronization error probabilities of each transition over the CID channel depends on the location of  $S_{i,j}$ .

**Case 1:  $S_{i,j}$  is located on the lower boundary when  $i - j = 1$  and it can transit to  $S_{i+1,j+1}$  or  $S_{i+1,j+2}$ , as shown in Fig 6(a).**

$$\begin{aligned} \beta_{i,j} &= p'_i \beta_{i+1,j+1} T(x_{i+1}, y_{j+1}) + p'_i \beta_{i+1,j+2} \cdot T(x_{i+1}, y_{j+2}) \Pr(\text{inserted bit} = y_{j+1}) \\ &= p'_i \beta_{i+1,j+1} T(x_{i+1}, y_{j+1}) + \frac{p'_i}{2} \beta_{i+1,j+2} T(x_{i+1}, y_{j+2}). \end{aligned} \quad (13)$$



**Fig 6. Possible state transitions when  $x_{i+1}$  is transmitted on condition that  $S_{i,j}$  has occurred.** (a)  $S_{i,j}$  is located on the lower boundary ( $i - j = 1$ ); (b)  $S_{i,j}$  is located on the diagonal ( $j - i = 0$ ); (c)  $S_{i,j}$  is located on the upper boundary ( $j - i = 1$ ).

<https://doi.org/10.1371/journal.pone.0270247.g006>

**Case 2:**  $S_{i,j}$  is located on the diagonal when  $i - j = 0$  and it can transit to  $S_{i+1,j}$ ,  $S_{i+1,j+1}$  or  $S_{i+1,j+2}$ , as shown in Fig 6(b).

$$\begin{aligned}\beta_{i,j} &= p_d \beta_{i+1,j} + p_i \beta_{i+1,j+1} T(x_{i+1}, y_{j+1}) + p_i \beta_{i+1,j+2} T(x_{i+1}, y_{j+2}) \Pr(\text{inserted bit} = y_{j+1}) \\ &= p_d \beta_{i+1,j} + p_i \beta_{i+1,j+1} T(x_{i+1}, y_{j+1}) + \frac{p_i}{2} \beta_{i+1,j+2} T(x_{i+1}, y_{j+2}).\end{aligned}\quad (14)$$

**Case 3:**  $S_{i,j}$  is located on the upper boundary when  $j - i = 1$  and it can transit to  $S_{i+1,j}$  or  $S_{i+1,j+1}$ , as shown in Fig 6(c).

$$\beta_{i,j} = p'_d \beta_{i+1,j} + p'_i \beta_{i+1,j+1} T(x_{i+1}, y_{j+1}).\quad (15)$$

In summary, given the initial conditions  $\beta_{t,r} = 1$  and  $\beta_{t,r-1} = 0$ , all other values of  $\beta_{i,j}$  within the boundaries in Fig 4 can be calculated iteratively.

## Likelihood

The likelihoods  $p(y_1^r | x_i)$  are first derived in terms of  $\alpha_{i,j}$  and  $\beta_{i,j}$  under three cases, i.e.,  $x_i$  is deleted;  $x_i$  is transmitted without deletion or insertion error; and  $x_i$  suffers from an insertion error. They are derived as follows.

**Case 1: Only two scenarios are possible when  $x_i$  is deleted.**

$$p(y_1^r | x_i \text{ is deleted}) = p_d \alpha_{i-1,j} \beta_{i,j} |_{j=i-1} + p'_d \alpha_{i-1,j} \beta_{i,j} |_{j=i}.\quad (16)$$

**Case 2: Three scenarios are possible when  $x_i$  is transmitted without synchronization errors.**

$$\begin{aligned}p(y_1^r | x_i \text{ is transmitted}) &= p'_i \alpha_{i-1,j-1} \beta_{i,j} \cdot C(x_i, y_j) |_{j=i-1} \\ &\quad + p_i \alpha_{i-1,j-1} \beta_{i,j} C(x_i, y_j) |_{j=i} \\ &\quad + p'_i \alpha_{i-1,j-1} \beta_{i,j} C(x_i, y_j) |_{j=i+1}.\end{aligned}\quad (17)$$

**Case 3: Only two scenarios are possible when  $x_i$  is transmitted with an insertion error.**

$$p(y_1^r | x_i \text{ is inserted}) = \frac{p'_i}{2} \alpha_{i-1,j-2} \beta_{i,j} C(x_i, y_j) |_{j=i} + \frac{p_i}{2} \alpha_{i-1,j-2} \beta_{i,j} C(x_i, y_j) |_{j=i+1}.\quad (18)$$

Because all the cases are independent, the overall  $p(y_1^r | x_i)$  is the sum of Eqs (16), (17) and (18), where  $x_i \in \{0, 1\}$ . The LLRs are subsequently computed using Eq (5) and are used as soft inputs to the LDPC decoder.

## IV. Results and discussion

Simulations are conducted to investigate the error performance of the concatenated LDPC-marker code over the CID channel. We have implemented the simulation using Matlab program under the Windows operating system and the simulation is carried out as follows. A random sequence of  $k$  message bits is first generated and encoded into an LDPC codeword of  $n$  bits. Then markers of length  $N_m$  are inserted into an LDPC codeword at a fixed interval  $N_c$  to form a block. The encoding procedure is repeated for every  $k$  message bits as shown in Fig 2. A total of  $N = 10^5$  blocks are generated for simulation. The encoded sequence with  $N$  blocks is transmitted through the CID channel with insertion probability  $p_i$ , deletion probability  $p_d$ , substitution probability  $p_s$ , constant  $A$  and common ratio  $r$ . For all simulations, the  $(n = 4521,$

$k = 3552$ ) LDPC code [20] with rate 0.79 is used as the outer code. The code rate of the concatenated code can be determined when the values of  $N_c$  and  $N_m$  are given.

The received corrupted sequence is first decoded by the inner marker decoder. The inner marker decoder uses the modified forward-backward algorithm described in Sect. IV to derive the LLR of each LDPC code bit. All LLRs are sent to the outer LDPC decoder. The LDPC decoder further decodes the codeword through the sum-product algorithm. At the end of each iteration, the LDPC decoder calculates the updated *a posteriori* LLR for each transmitted bit and make a hard decision on the LLRs to estimate a decoded codeword. The LDPC decoder immediately checks whether the estimated codeword satisfies all check equations. The block is decoded successfully if it satisfies all check equations; otherwise, the iteration continues. In our simulations, the maximum number of iterations is set to 60. If the decoded codeword cannot satisfy all check equations after the maximum number of iterations, the decoding fails. We perform the decoding procedure and count the number of decoded bits and blocks in error. The bit error rate (BER) and block error rate (BLER) are used as metrics to evaluate the error performance of the concatenated LDPC-marker code.

### BER/BLER error performance

In the first set of simulations, regular markers '10' of length  $N_m = 2$  are inserted at the beginning of each LDPC codeword, and also periodically every  $N_c = 18$  LDPC code bits. In other words, each marker is followed by 18 LDPC code bits, except the last marker which is followed only by 3 code bits. The total length of the transmitted sequence is 5023 bits per block and the

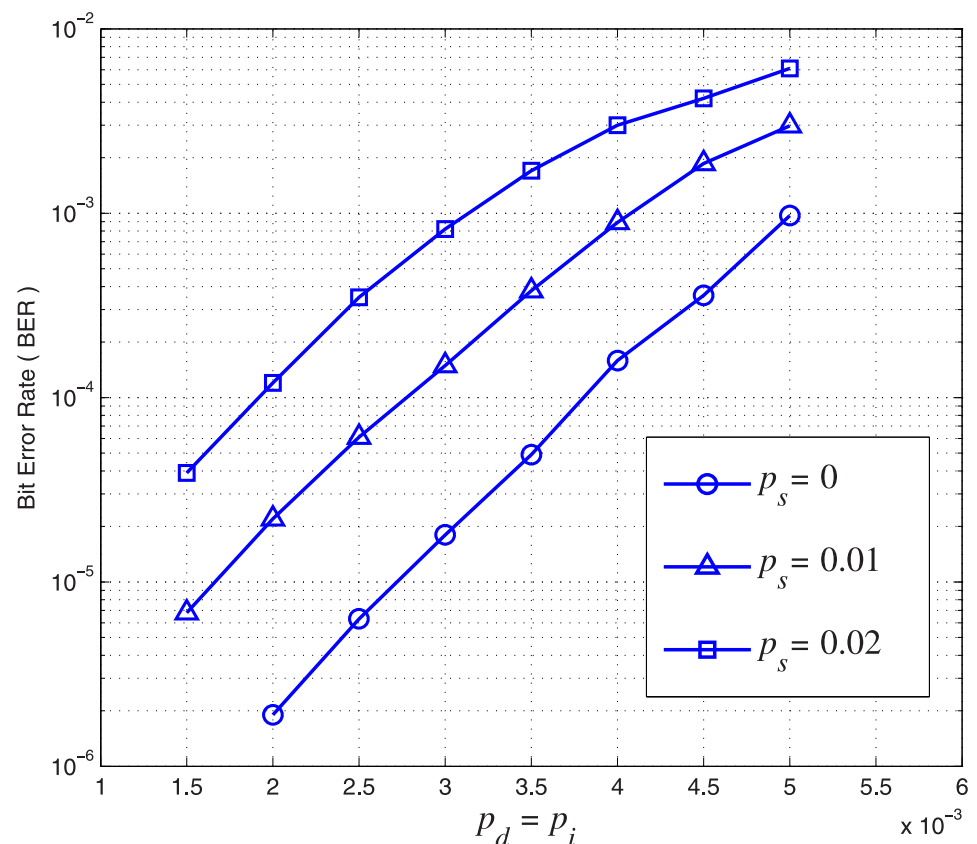
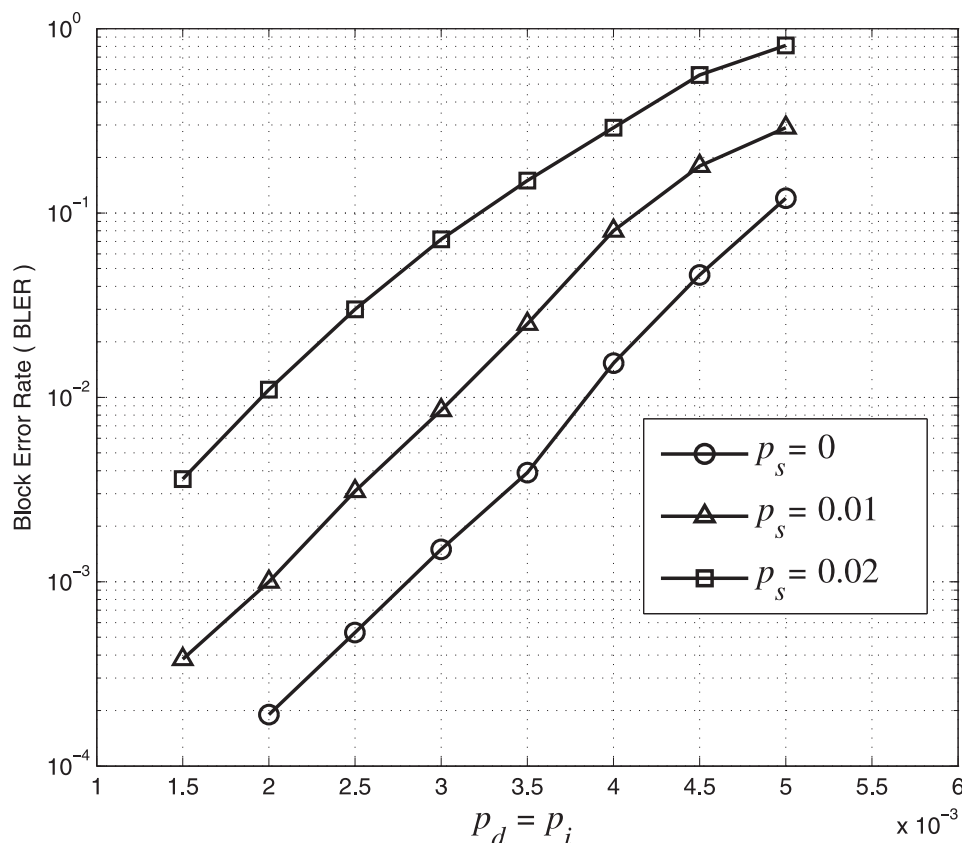


Fig 7. BER of the concatenated LDPC-marker code over the CID channel.  $N_c = 18$  and  $N_m = 2$ .

<https://doi.org/10.1371/journal.pone.0270247.g007>



**Fig 8. BLER of the concatenated LDPC-marker code over the CID channel.**  $N_c = 18$  and  $N_m = 2$ .

<https://doi.org/10.1371/journal.pone.0270247.g008>

overall code rate is  $R = 3552/5023 = 0.71$ . For this simulation, we consider  $p_i = p_d$  and set  $A = r = 0.5$  in Eq (1). The BER and BLER performance of the concatenated code over the CID channel are shown in Figs 7 and 8, respectively, under different channel insertion, deletion and substitution error rates. As expected, both BER and BLER increase as  $p_i$ ,  $p_d$  and  $p_s$  increase. We first investigate the case when there are only insertion and deletion errors, i.e.,  $p_s = 0$ . The BER of the concatenated LDPC-marker code can be as low as  $2 \times 10^{-6}$  when  $p_i = p_d = 2 \times 10^{-3}$ . The BER of the concatenated LDPC-marker code increases as  $p_i$  and  $p_d$  increase. For example, the BER of the concatenated LDPC-marker code increases from  $2 \times 10^{-6}$  to  $10^{-3}$  when  $p_i$  and  $p_d$  increase from  $2 \times 10^{-3}$  to  $5 \times 10^{-3}$  in Fig 7. In Fig 8, the BLER of the concatenated LDPC-marker code can be as low as  $2 \times 10^{-4}$  when  $p_i = p_d = 2 \times 10^{-3}$ . The BLER of the concatenated LDPC-marker code also increases as  $p_i$  and  $p_d$  increase. For example, the BLER of the concatenated LDPC-marker code increases from  $2 \times 10^{-4}$  to 0.12 when  $p_i$  and  $p_d$  increase from  $2 \times 10^{-3}$  to  $5 \times 10^{-3}$  as shown in Fig 8. The result shows that the inner decoder can effectively maintain synchronization over the CID channel with the help of the modified forward-backward algorithm.

Furthermore, we investigate and compare the error performance in terms of BER and BLER with different  $p_s$ . The simulation results show that the decoder can correctly decode the corrupted block when  $p_s$  is relatively low. For example, when  $p_i = p_d = 3 \times 10^{-3}$  and  $p_s = 0.01$ , BER and BLER are as low as  $1.7 \times 10^{-4}$  and  $8 \times 10^{-3}$ , respectively. However, more bits suffer from substitution errors as  $p_s$  increases and hence both BER and BLER increases. For example, the BLER increases rapidly from  $2 \times 10^{-4}$  to  $1.2 \times 10^{-2}$  when  $p_s$  increases from 0 to 0.02 ( $p_i = p_d$ ).

$= 2 \times 10^{-3}$ ). The corrupted block cannot be decoded at a higher substitution error rate because the number of substitution errors exceeds the correction capability of the LDPC code.

### Impact of markers on error performance

We first investigate the marker length  $N_m$  on the error performance of the concatenated code. We fix  $N_c = 18$  and use  $N_m = 2, 3, 4$ . The markers used are '10', '101' and '1010' while the corresponding overall code rates are 0.71, 0.67 and 0.64, respectively. For this simulation, we set  $A = r = 0.5$  in Eq (1). The BLER performance of concatenated codes with different marker length  $N_m$  is shown in Fig 9. The results show that when  $N_c$  is fixed, the BLER decreases as the marker length  $N_m$  increases. For example, when  $p_d = p_i = 4 \times 10^{-3}$  and  $p_s = 0.01$ , the BLER decreases from  $8 \times 10^{-2}$  ( $N_m = 2$ ) to  $2 \times 10^{-2}$  ( $N_m = 4$ ). When  $p_d = p_i = 3 \times 10^{-3}$  and  $p_s = 0$ , the BLER decreases from  $7 \times 10^{-4}$  ( $N_m = 3$ ) to  $4.5 \times 10^{-4}$  ( $N_m = 4$ ). In other words, the error performance of the concatenated code is improved by increasing the marker length  $N_m$ . However, the code rate of the concatenated code is decreased as the marker length  $N_m$  increases. We also notice that the error performance of the concatenated code in terms of BLER is seriously affected by  $p_s$ . For example, when  $p_i = p_d = 3 \times 10^{-3}$ , the BLER for  $N_m = 4$  increases by about 50 times, rapidly from  $4.5 \times 10^{-4}$  ( $p_s = 0$ ) to  $2.1 \times 10^{-2}$  ( $p_s = 0.02$ ). The results show that even though markers help to maintain synchronization, corrupted blocks cannot be corrected due to a large number of substitution errors.

We also study the effect of marker interval  $N_c$  on the error performance of the concatenated code. We fix  $N_m = 2$  and use  $N_c = 18, 24, 30$ . The corresponding overall code rates are 0.71,

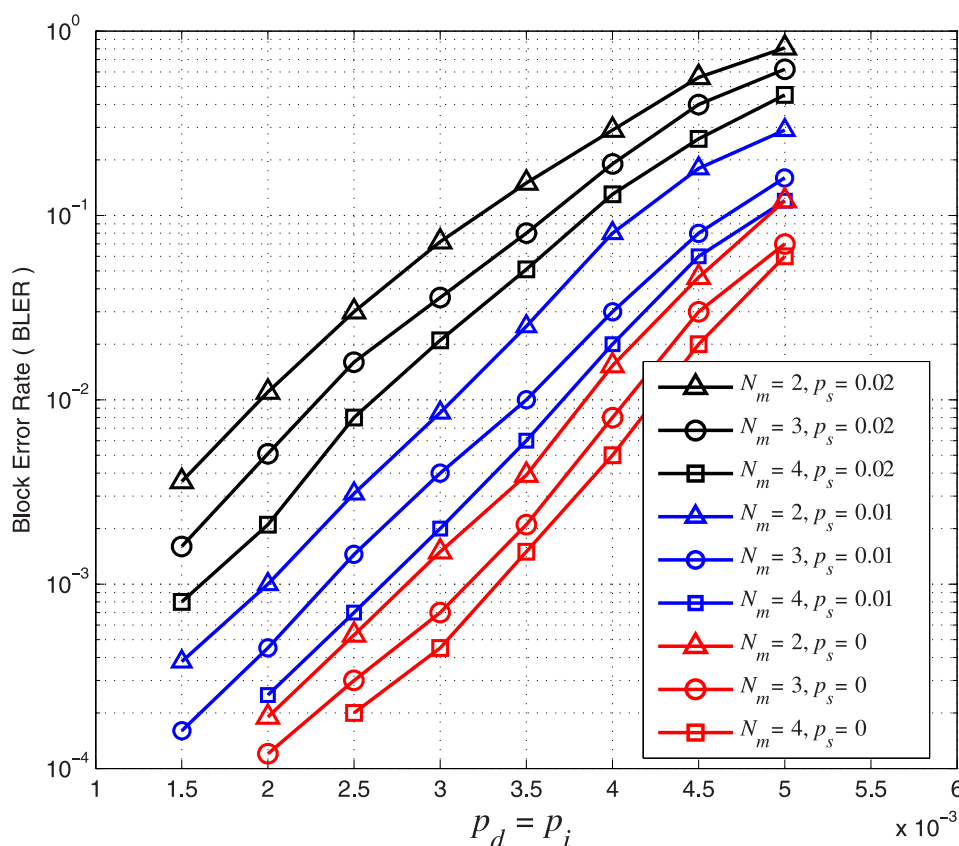
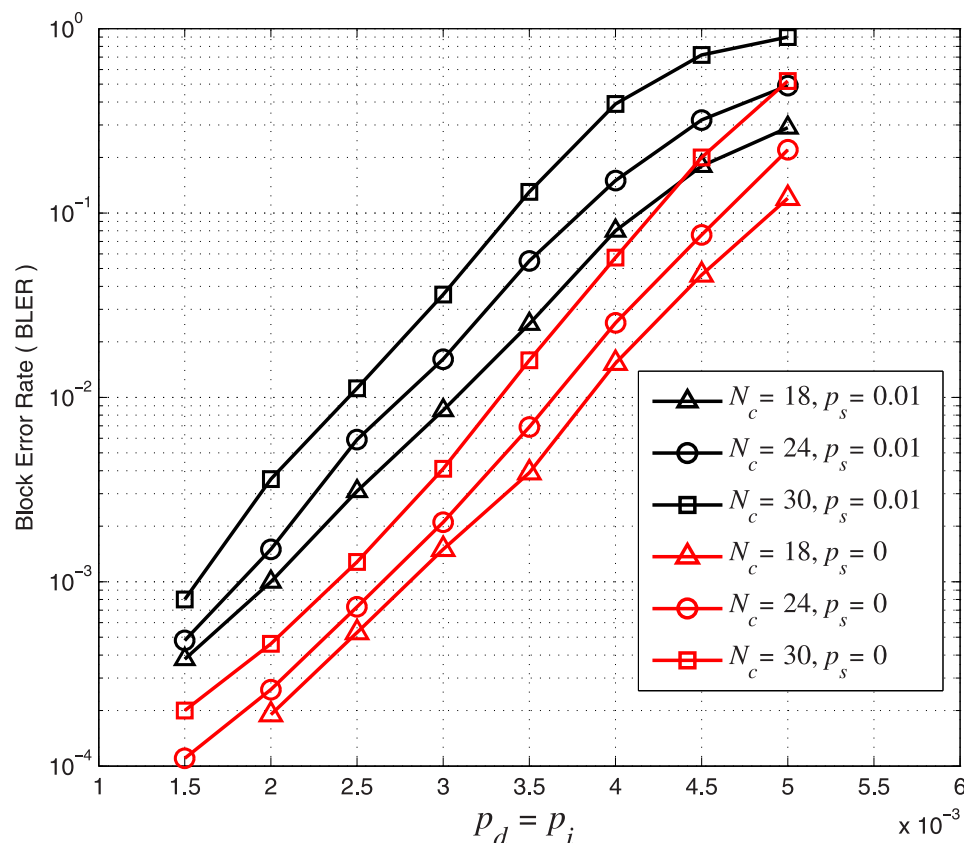


Fig 9. BLER of the concatenated LDPC-marker code over the CID channel.  $N_c = 18$  and  $N_m = 2, 3, 4$ .

<https://doi.org/10.1371/journal.pone.0270247.g009>



**Fig 10.** BLER of the concatenated LDPC-marker code over the CID channel.  $N_m = 2$  and  $N_c = 18, 24, 30$ .

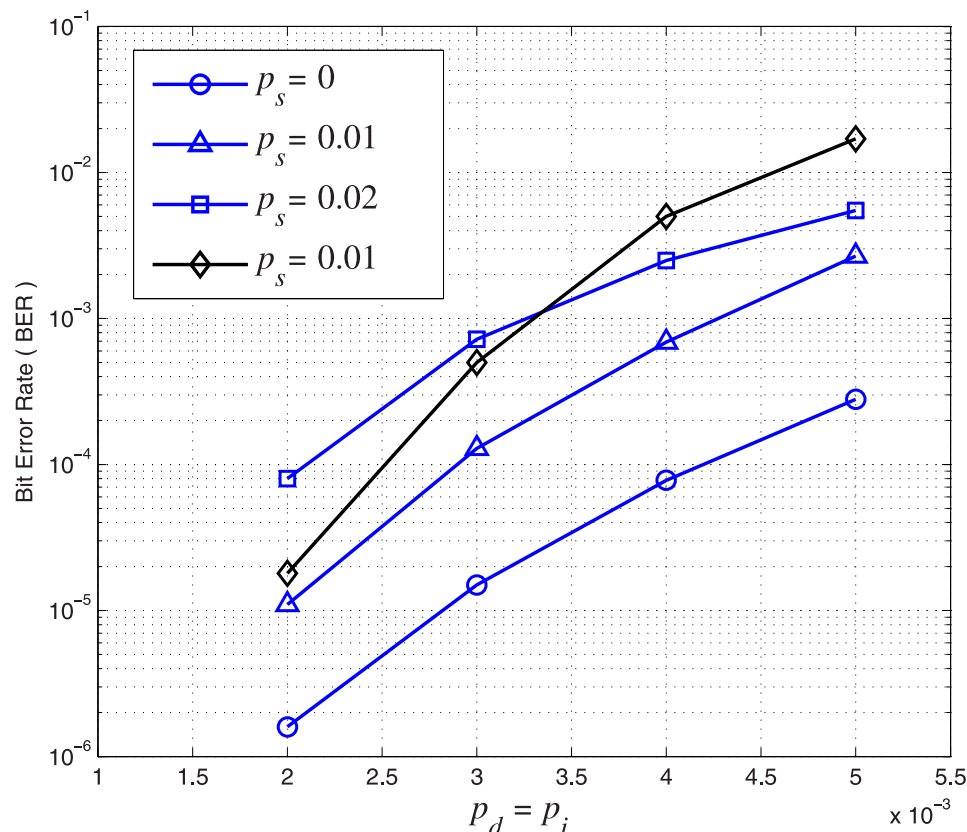
<https://doi.org/10.1371/journal.pone.0270247.g010>

0.73 and 0.74, respectively. The marker interval  $N_c$  is related to the number of markers. A larger marker interval means fewer inserted markers in a block. For this simulation, we set  $A = r = 0.5$  in Eq (1). In Fig 10, we plot BLER curves for concatenated codes with different marker intervals  $N_c$  under different channel parameters. It can be observed that the BLER error performance is improved with a smaller marker interval or more markers. For example, when  $p_d = p_i = 3 \times 10^{-3}$  and  $p_s = 0$ , the BLER increases from  $2.1 \times 10^{-3}$  ( $N_c = 24$ ) to  $4 \times 10^{-3}$  ( $N_c = 30$ ). When  $p_d = p_i = 2.5 \times 10^{-3}$  and  $p_s = 0.01$ , the BLER increases from  $3 \times 10^{-3}$  ( $N_c = 18$ ) to  $5.8 \times 10^{-3}$  ( $N_c = 24$ ). In other words, reducing the marker interval or inserting more markers in a block improves the error performance of the concatenated LDPC-marker code.

From the simulation results in Figs 9 and 10, we conclude that codes with larger marker length  $N_m$ , smaller marker interval  $N_c$  and more markers have a better synchronization capability and hence error performance. Although increasing  $N_m$  and/or reducing  $N_c$  improve the error performance of the concatenated code, the information transmission efficiency and code rate is reduced.

### Comparison with existing schemes

Simulations are further carried out to compare the error performance of the concatenated LDPC-marker code with the result in [16]. In order to make a fair comparison, we set  $A = 0.5$  and  $r = 1$  in Eq (1). Thus,  $p'_d = p'_i = 0.5$  are consistent with the error rates in [16] for subsequent bits after a synchronization error. Furthermore, the inner marker code with  $N_m = 2$  and



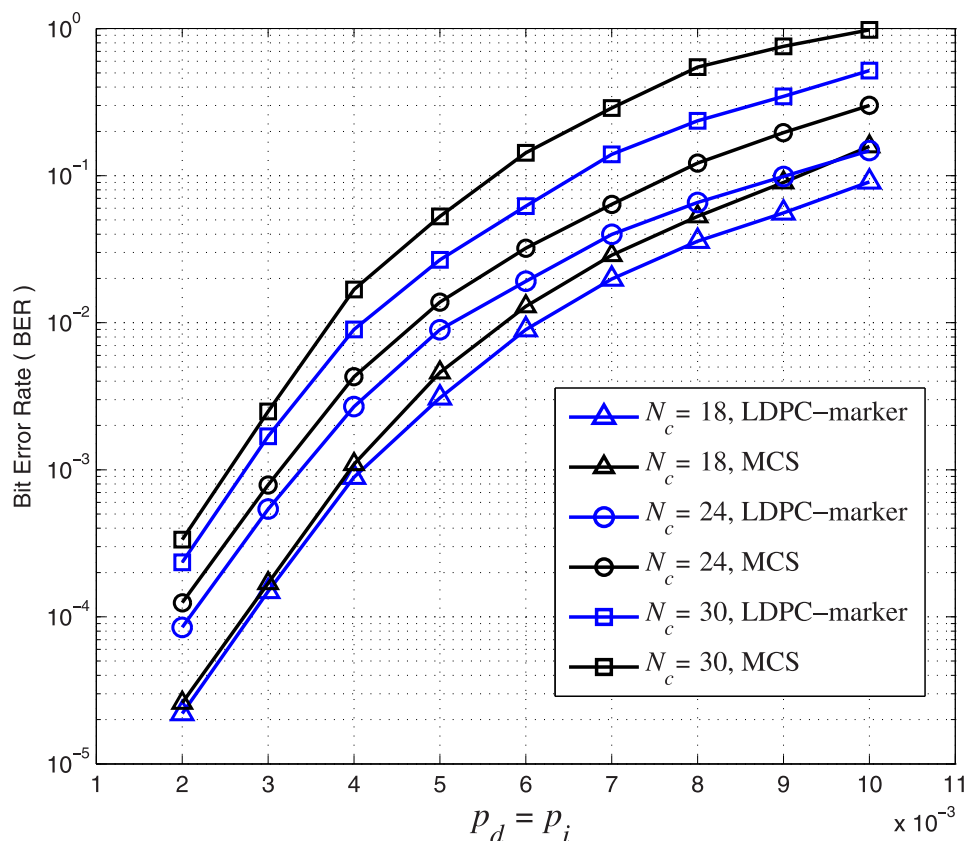
**Fig 11. Comparison performance under different channel parameters.**  $p_d = p_i$  increase from  $2 \times 10^{-3}$  to  $5 \times 10^{-3}$  and  $p_s$  increases from 0 to 0.02. Blue curves represent concatenated LDPC-marker codes under different  $p_s$  and the black curve represents the best code in [16] when  $p_s = 0.01$ .

<https://doi.org/10.1371/journal.pone.0270247.g011>

$N_c = 18$  is adopted so that both codes have the same code rate 0.71 for a fair comparison. The BLER curves of the concatenated LDPC-marker code over the CID channel and the BLER curve with the best performance in [16] under the same channel parameters are compared in Fig 11. The simulation result shows that the concatenated LDPC-marker code performs much better than the best code in [16] when  $p_s = 0.01$ . This improvement is more significant with the increase of insertion and deletion error rates. For example, when  $p_d = p_i = 5 \times 10^{-3}$  and  $p_s = 0.01$ , the BER of the concatenated code is  $2.8 \times 10^{-3}$  while the BER of the best code in [16] is more than  $1 \times 10^{-2}$ . We also observe that the concatenated code at  $p_s = 0.02$  even performs better than the best code in [16] at  $p_s = 0.01$  when  $p_d$  and  $p_i$  are greater than  $3.3 \times 10^{-3}$ . This proves that the concatenated LDPC-marker code has a better resistance to substitution errors. Compared with [16], the CID channel model is more accurate for BPMR systems and the error performance is improved under the same channel parameters.

We further investigate and compare the error performance of the concatenated LDPC-marker code considering correlated synchronization errors with the conventional marker coding scheme (MCS) [8] without considering the dependency between synchronization errors over the CID channel. In the following, we use the (4521, 3552) LDPC code as the outer code. For the inner marker code, we fix  $N_m = 2$  and use  $N_c = 18, 24, 30$ . The corresponding overall code rates are 0.71, 0.73 and 0.74, respectively. For this simulation, we consider  $p_i = p_d$ ,  $p_s = 0.01$  and set  $A = r = 0.5$  in Eq (1). We simulate the error performance of the two coding



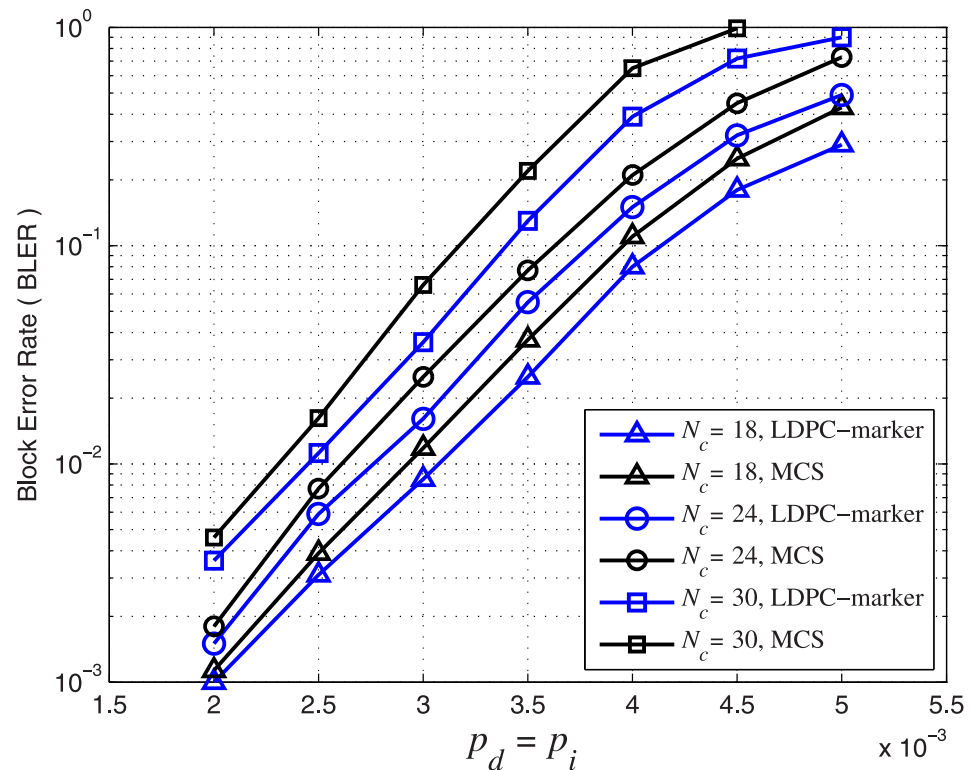


**Fig 12. BER performance comparison over the CID channel.**  $p_d = p_i$  increase from  $2 \times 10^{-3}$  to  $10^{-2}$  and  $p_s = 0.01$ . The blue curve represents the concatenated LDPC-marker code and the black curve represents the conventional marker coding scheme.

<https://doi.org/10.1371/journal.pone.0270247.g012>

schemes. The BER and BLER error performance over the CID channel are shown in Figs 12 and 13, respectively.

The simulation result shows that the concatenated LDPC-marker code produces a lower BER/BLER error rate than the conventional MCS over the CID channel. The improvement is intuitive and can be explained as follows. The conventional MCS assumes that insertion, deletion and substitution errors occur randomly without correlation, and it does not consider the dependency between the synchronization errors in the CID channel. On the contrary, the concatenated LDPC-marker code takes into account the correlation to compute LLRs by means of the modified forward-backward algorithm. As a result, LLRs provided by the inner decoder of the concatenated LDPC-marker code are more accurate. In addition to error performance, we compare the decoding complexity of the two coding schemes. Compared with the MCS, the decoding time of the concatenated LDPC-marker code is greatly reduced by 35% on average for each BLER curve in Fig 13. The explanation is as follows. The width of the decoding trellis is proportional to insertion and deletion error rates. As insertion and deletion error rates increase, the scale of the decoding trellis becomes larger and the computational complexity becomes higher. For the concatenated LDPC-marker code, the decoding trellis is limited to a small range (between upper boundary and lower boundary) because insertion and deletion errors always occur in pairs. Many transitions on the transition diagram are not allowed due to the correlation between synchronization errors. For example, consecutive



**Fig 13. BLER performance comparison over the CID channel.**  $p_d = p_i$  increase from  $2 \times 10^{-3}$  to  $5 \times 10^{-3}$  and  $p_s = 0.01$ . The blue curve represents the concatenated LDPC-marker code and the black curve represents the conventional marker coding scheme.

<https://doi.org/10.1371/journal.pone.0270247.g013>

insertions/deletions are not allowed in the CID channel. However, the decoding scale of the MCS is relatively large and the decoder has to find the mostly likely transmission path among all possible candidates.

## V. Conclusion and future work

In this paper, we have proposed a probabilistic channel model suitable for BPMR systems with frequent correlated insertion/deletion errors. We have further modified the forward-backward algorithm and investigated the error performance of a concatenated code over this channel. The simulation results show that the inner marker code can effectively maintain synchronization, while the outer LDPC code can provide sufficient error correction capability for the BPMR system. The error performance can be further improved when the marker interval is reduced and/or more markers are inserted in the transmitted sequence. The simulation result shows that the concatenated LDPC-marker code performs much better than the code in [16]. The improvement is more significant with the increase of insertion and deletion error rates. We also investigate and compare the error performance of the concatenated LDPC-marker code considering correlated synchronization errors with the existing conventional MCS without considering the dependency between synchronization errors over the CID channel. The simulation result shows that the concatenated LDPC-marker code produces a lower BER/BLER error rate than the conventional MCS over the CID channel. Compared with existing methods, our coding scheme improves the error performance and reduces the decoding complexity. We have assumed binary data transmission and the fixed LDPC code in this study. In

the future, we plan to investigate  $M$ -ary data transmission and different LDPC codes over the CID channel.

## Author Contributions

**Conceptualization:** Tianbo Xue.

**Data curation:** Tianbo Xue.

**Formal analysis:** Tianbo Xue.

**Methodology:** Tianbo Xue.

**Software:** Tianbo Xue.

**Validation:** Tianbo Xue.

**Writing – original draft:** Tianbo Xue.

## References

1. Gallager, RG. Sequential decoding for binary channels with noise and synchronization errors. Group Report. 1961.
2. Dong Yiming and Sun Fajia and Ping Zhi and Ouyang Qi and Qian Long. DNA storage: research landscape and future prospects. National Science Review. 2020; 7(6):1092–1107. <https://doi.org/10.1093/nsr/nwaa007> PMID: 34692128
3. Begum Mahbuba and Uddin Mohammad Shorif. Digital image watermarking techniques: a review. Information. 2020; 11(2):110. <https://doi.org/10.3390/info11020110>
4. Saito Hidetoshi. Concatenated Coding Schemes for High Areal Density Bit-Patterned Media Magnetic Recording. IEEE Transactions on Magnetics. 2018; 54(2):1–10.
5. Gallager Robert. Low-density parity-check codes. IRE Transactions on Information Theory. 1962; 8(1):21–28. <https://doi.org/10.1109/TIT.1962.1057683>
6. Reed Irving S and Solomon Gustave. Polynomial codes over certain finite fields. Journal of the Society for Industrial and Applied Mathematics. 1960; 8(2):300–304. <https://doi.org/10.1137/0108018>
7. Haeupler B. and Shahrabi A. Synchronization strings and codes for insertions and deletions—A Survey. IEEE Transactions on Information Theory. 2021; 67(6):3190–3206. <https://doi.org/10.1109/TIT.2021.3056317>
8. Ratzer Edward A. Marker codes for channels with insertions and deletions. Annales Télécommunications. 2005; 60:29–44.
9. H. Kaneko. Timing-drift channel model and marker-based error correction coding. Proc. IEEE Int. Symp. Inf. Theory (ISIT). 2017.
10. Liu S., Tjuawinata I. and Xing C. Efficiently List-Decodable Insertion and Deletion Codes via Concatenation. IEEE Transactions on Information Theory. 2021.
11. Shibata Ryo and Hosoya Gou and Yashima Hiroyuki. Concatenated LDPC/Trellis Codes: Surpassing the Symmetric Information Rate of Channels with Synchronization Errors. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. 2020; 103(11):1283–1291. <https://doi.org/10.1587/transfun.2020EAP1019>
12. Shibata Ryo and Hosoya Gou and Yashima Hiroyuki. Concatenated LDPC/2-D-Marker Codes and Non-Iterative Detection/Decoding for Recovering Position Errors in Racetrack Memories. IEEE Transactions on Magnetics. 2020; 56(9):1–9. <https://doi.org/10.1109/TMAG.2020.3011447>
13. Iyengar Aravind Raghava and Siegel Paul H and Wolf Jack Keil. Write channel model for bit-patterned media recording. IEEE Transactions on Magnetics. 2010; 47(1):35–45. <https://doi.org/10.1109/TMAG.2010.2080667>
14. Mazumdar Arya and Barg Alexander and Kashyap Navin. Coding for high-density recording on a 1-D granular magnetic medium. IEEE transactions on information theory. 2011; 57(11):7403–7417. <https://doi.org/10.1109/TIT.2011.2158514>
15. Ghanami Fatemeh and Abed Hodtani Ghosheh. Information Theoretical Analysis of a New Write Channel Model for Bit-Patterned Media Recording. IEEE Transactions on Magnetics. 2020; 56(4):1–9. <https://doi.org/10.1109/TMAG.2020.2970830>

16. Y. Suzuki and H. Kaneko. Correlated insertion/deletion error correction coding for bit-patterned media. 2017 IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW). 2017:7-8.
17. Davey Matthew C and MacKay David JC. Reliable communication over channels with insertions, deletions, and substitutions. IEEE Transactions on Information Theory. 2001; 47(2):687–698. <https://doi.org/10.1109/18.910582>
18. Sellers F. Bit loss and gain correction code. IRE Transactions on Information Theory. 1962; 8(1):35–38. <https://doi.org/10.1109/TIT.1962.1057684>
19. Durbin Richard and Eddy Sean R and Krogh Anders and Mitchison Graeme. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press. 1998.
20. MacKay D. J. C. Good error-correcting codes based on very sparse matrices. IEEE Transactions on Information Theory. 1999; 45(2):399–431. <https://doi.org/10.1109/18.748992>