IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024TAP0007

Publicized:2024/09/09

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER *Special Section on Information Theory and Its Applications*

Double-Stack Erasure-Filled Channel and Level-By-Level Error Correction*

Hironori UCHIKAWA[†], ***Senior Member and*** **Manabu HAGIWARA**[††], ***Member***

SUMMARY Racetrack memory is a new type of high-capacity memory that stores data in magnetic nanowires called racetracks. Data is transferred through the nanowires to the access port for reading and writing. However, the data transfer process is imperfect and can lead to errors. Inspired by racetrack memory array architecture, the authors propose a new channel model in which missing data is filled with erasures at the end of the racetrack. Channel capacity and symmetric information rate for the proposed channel, a double-stack erasure-filled (DSEF) channel, are derived. Since the DSEF channel is a quaternary-input septenary-output channel, constructing good error-correcting codes is not trivial. We decompose the DSEF channel into two binary-input ternary-output channels to overcome this difficulty. This decomposition allowed us to construct an adequate error-correction scheme using existing binary codes, which is a meaningful achievement in terms of implementation.

key words: *racetrack memory, deletion channel, symmetric information rate, error correction, LDPC codes.*

1. Introduction

Racetrack memory (also known as domain wall memory) is expected to be the next generation of high-capacity memory [1]. In racetrack memory, data is stored in arrays of magnetic nanowires (racetracks). Electric currents push the data through the nanowires and past the access port that serves as the interface for reading and writing data.

In contrast to hard disk drives, which move the read/write head over the rotating disk, the data is transferred through the magnetic nanowires to the head in racetrack memory. The data transfer in the nanowires is not perfect and may suffer from positional errors. Data symbols might be shifted by multiple locations or might not be shifted at all. These unique specifications of racetrack memory have led to the creation of new error models known as deletion and sticky-insertion errors [2, 3]. These errors change the data length. Such kinds of errors are also called synchronization errors, and a comprehensive survey on the capacity of synchronization errors, including deletion and insertion errors, is summarized in [4].

However, in general memory systems, the data length is expected to remain unchanged because data is sampled a specified number of times at the access port during each read operation. Assume the data length of racetrack memory is L and all L symbols are read during data access. If E deletion errors occur, the last E symbols will become invalid or are expected to form a random sequence of 0s and 1s. On the other hand, if E sticky-insertion errors occur, each of the E erroneous symbols will be duplicated, resulting in the last E symbols not being read. Thus, the channel model for racetrack memory in a system where the read data length remains unchanged will differ from the literature on this topic. Therefore, it is appropriate to discuss a channel model where the data length does not change for racetrack memory.

In this work, we propose an erasure-filled channel, where missing data due to data-transfer failures on magnetic nanowires is filled with erasures. As previously mentioned, in practice, the end of the racetrack is filled with invalid values corresponding to the number of deleted symbols, making a model with random bits more realistic than one with erasures, we adopted erasures for easier analysis to deepen theoretical understanding. This approach enabled us to analyze the symmetric information rate. This paper focuses solely on deletion errors and does not consider insertion errors. This limitation does not imply an optimistic assumption, as the data transfer speed in racetrack memory can be adjusted by the applied current pulse [5]. Adjusting the data transfer speed makes it realistically possible to configure racetrack memory where deletion errors dominate and insertion errors do not occur. The results of this paper will be beneficial for such devices.

Multiple racetracks are expected to be implemented in parallel to achieve high capacity and high-speed access [6,7]. In such a racetrack memory array, writes and reads are performed in units of the number of parallelized racetracks. Therefore, the number of parallelized racetracks should be equal to or a multiple of the encoding block size for error correction from a practical point of view [8]. Multiple deletion and insertion channels, such as multiple racetrack memory channels, can be categorized into two main models. One model assumes that deletion and insertion errors occur simultaneously and in parallel within each racetrack [9–11]. The other is the case where errors occur independently in each racetrack [3]. In multiple racetrack memory, errors are expected to occur independently for each racetrack due to variations in the physical characteristics of each racetrack, so the latter model is considered more suitable as a channel model for multiple racetrack memory. When deletion and insertion errors occur in each racetrack, as in [3], the read-

[†]Memory Division, Kioxia Corporation, Kanagawa, Japan.
Email: hironori.uchikawa@kioxia.com

[††]Department of Mathematics and Informatics, Chiba University, 263-0022 Chiba, Japan.
Email: hagiwara@math.s.chiba-u.ac.jp

*The research has been partly executed in response to support by KAKENHI 21H03393 and Kioxia Corporation.

out data length for each racetrack is different. The channel model discussed in this paper is similar to [3] in that the errors occur independently for each racetrack, but the difference is that the erasure is filled in at the end to keep the data length constant.

The data length in each nanowire of the racetrack memory is generally longer because higher data density can be achieved by increasing the amount of data held in the racetrack. However, for simplicity of analysis, the case of length two is discussed in this paper. To evaluate the quality of this channel, we derive the channel capacity and symmetric information rate for the proposed deletion and erasure-filled channel of length two.

There is extensive research on constructing powerful error-correcting codes for racetrack memory [12–14]. However, many of these coding schemes store the codeword in a single track, which is not suitable for multi-racetrack memory where multiple tracks are written simultaneously. On the other hand, there has been limited exploration of vertical coding [1, 3], which stores each bit of the codeword across multiple tracks. This paper contributes to the study of vertical coding. Although constructing good error-correcting codes for the proposed channel is not trivial, we construct an adequate error-correction scheme using existing binary codes through a decomposition of the proposed channel into two binary-input ternary-output channels. This results in a meaningful achievement in terms of implementation.

The rest of this paper is organized as follows. In Section 2, we formally define *multi-stack erasure-filled (MSEF)* channels and introduce their simplest channel as *double-stack erasure-filled (DSEF)* channel. In Section 3, the channel capacity of the DSEF channel is discussed. Section 4 discusses the construction of error-correcting codes for the DSEF channel. We especially decompose the DSEF channel into two binary-input ternary-output channels to overcome the difficulty of constructing good error-correcting codes. The level-by-level error correction is also introduced. In Section 5, the error-correction capability of the level-by-level error correction is evaluated by using computer simulations. Finally, Section 6 concludes the paper.

2. Channel Model

Data-transfer failures in racetrack memory cause data errors. For example, if excessive data transfer occurs, data reading at the access port is skipped, and subsequent data is read. If there is insufficient data transfer, data is read in duplicate at the access port [2]. Such data errors are often modeled as deletion-insertion channels. In the delete-insertion channel, the data length changes due to errors [15]. Meanwhile, the data length in a typical memory system is expected to remain unchanged. That is because data is sampled a certain number of times at the access port during read operations.

Therefore, we propose the following channel. The input alphabet $\mathcal{X}$ is the set $\mathbb{F}_2^L$ of a vertical vector of length L, where L corresponds to the length of the data in each nanowire of the racetrack memory. The output alphabet $\mathcal{Y}$ is a subset of $(\mathbb{F}_2 \cup \{?\})^L$, where ? is the erasure symbol. The element of $\mathcal{Y}$ has the form $(y_1, y_2, \dots, y_l, ?, ?, \dots, ?)$ for some $l \leq L$ and $y_i \in \mathbb{F}_2$. In this case, the first l symbols are binary, and the remaining $L - l$ symbols are erasure symbols.

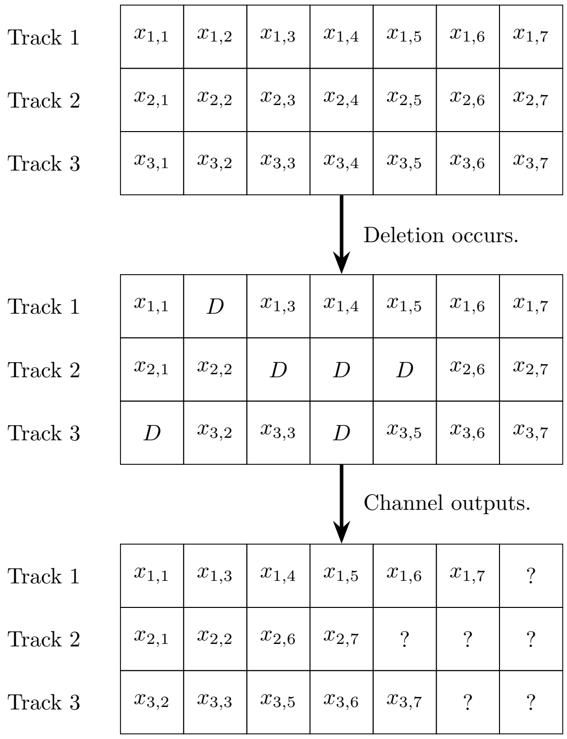
For the input $\boldsymbol{x} = (x_1, x_2, \dots, x_L) \in \mathcal{X}$, each symbol x_i is independently deleted with a fixed probability δ. If $\boldsymbol{x}$ is changed to $\boldsymbol{y} = (y_1, y_2, \dots, y_l) \in \mathbb{F}_2^l$, the output is $(y_1, y_2, \dots, y_l, ?, ?, \dots, ?) \in \mathcal{Y}$. We call this channel a *multi-stack erasure-filled (MSEF) channel* of L levels with deletion probability δ. As a first step in analyzing the MSEF channels, this paper discusses an MSEF channel of two levels called a *double-stack erasure-filled (DSEF) channel.* Because the DSEF channel is the simplest MSEF channel, it is useful for understanding the MSEF channels.

Example 2.1: Consider sending a sequence $(0, 1, 1, 0, 1, 0, 0, 1)$ through an MSEF channel of eight levels. At the deletion step, let us assume the 2nd and 4th entries are deleted. It may happen with the probability $\delta^2(1 - \delta)^6$. The sequence is changed to $(0, 1, 1, 0, 0, 1)$. Then channel outputs $(0, 1, 1, 0, 0, 1, ?, ?)$.

Note that the probability of the input $(0, 1, 1, 0, 1, 0, 0, 1)$ and the output $(0, 1, 1, 0, 0, 1, ?, ?)$ is not $\delta^2(1-\delta)^6$. Because other deletions can change the input to the same output, e.g., the 3rd and 4th deletions. In fact, the probability is $5\delta^2(1 - \delta)^6$.

Since multiple racetracks are expected to be implemented in parallel to achieve high capacity and high-speed access, we consider multi-track MSEF channels. Figure 1 illustrates a three-track MSEF channel of seven levels. Each track in the row direction is an MSEF channel and contains a sequence of data symbols $x_{i,j}$. Deletions occur independently in each track, marked as ‘D’, resulting in a varying number of deletions across different tracks. This independent deletion process causes variability in the number of deletions per track. Subsequently, the channel outputs the remaining symbols, filling the gaps with erasure symbols ‘?’ at the end of the sequence in each track.

In general, in an n-track multi-racetrack memory, the write unit is n bits, meaning that each of the n bits is input to the corresponding track. If the length of each track is L, all nL slots will be stored by L writes. On the other hand, data reads are assumed to be performed in bulk. Specifically, it is assumed that all nL bits are read out at once, rather than reading only specific n bits. This approach is taken because shift errors in racetrack memory accumulate over time. Therefore, to prevent the accumulation of shift errors and maintain data reliability, all data is output at once when a data readout occurs. This type of readout is known as destructive readout [16]. If data needs to be retained, it is rewritten afterward. In the multi-track MSEF channel discussed in this paper, we also assume this kind of readout method. In other words, the readout unit is nL bits, and new data is written after reading all nL bits. It can be considered that the accumulation of shift errors is refreshed every nL bits. Since the write unit is n bits, n is equal to or multiple of the encoding block size for error correction. Thus, the coding



Track 1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$	$x_{1,7}$
Track 2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$	$x_{2,7}$
Track 3	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$x_{3,6}$	$x_{3,7}$

Deletion occurs.

Track 1	$x_{1,1}$	D	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$	$x_{1,7}$
Track 2	$x_{2,1}$	$x_{2,2}$	D	D	D	$x_{2,6}$	$x_{2,7}$
Track 3	D	$x_{3,2}$	$x_{3,3}$	D	$x_{3,5}$	$x_{3,6}$	$x_{3,7}$

Channel outputs.

Track 1	$x_{1,1}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$	$x_{1,7}$	?
Track 2	$x_{2,1}$	$x_{2,2}$	$x_{2,6}$	$x_{2,7}$	?	?	?
Track 3	$x_{3,2}$	$x_{3,3}$	$x_{3,5}$	$x_{3,6}$	$x_{3,7}$	?	?

Fig.1: Three-track MSEF channels of seven levels.

scheme discussed in Section 5 is vertical coding [1]. This scheme uses slots across multiple tracks to store a codeword, i.e., each column $(x_{1,j}, x_{2,j}, x_{3,j})$ in Fig. 1 corresponds to a codeword.

3. Double-Stack Erasure-Filled Channel

This section calculates the channel capacity of the DSEF channel. Due to the destructive readout assumption mentioned in the previous section, memoryless inputs suffice to achieve the channel capacity of the DSEF channel. The channel capacity is obtained by calculating the maximum mutual information between the input and the output of the DSEF channel. The input alphabet consists of four elements $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$, while the output alphabet consists of seven elements $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$, $(0,?)$, $(1,?)$, and $(?,?)$. The transition matrix $P_{Y|X}$ is described as follows:

$$P_{Y|X} = \begin{pmatrix} (1-\delta)^2 & 0 & 0 & 0 \\ 0 & (1-\delta)^2 & 0 & 0 \\ 0 & 0 & (1-\delta)^2 & 0 \\ 0 & 0 & 0 & (1-\delta)^2 \\ 2\delta(1-\delta) & \delta(1-\delta) & \delta(1-\delta) & 0 \\ 0 & \delta(1-\delta) & \delta(1-\delta) & 2\delta(1-\delta) \\ \delta^2 & \delta^2 & \delta^2 & \delta^2 \end{pmatrix}. \tag{1}$$

By denoting the event probabilities for input $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$ by p_{00}, p_{01}, p_{10}, and p_{11} respectively. The mutual information $I(X;Y)$ is

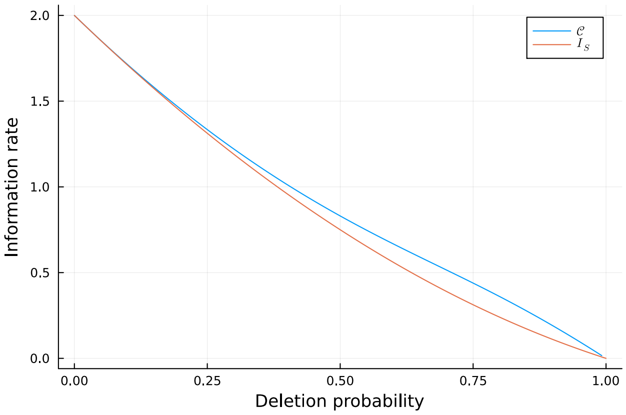


Fig. 2: The channel capacity C and the symmetric information rate I_S.

$$\begin{aligned} &I(X;Y) \\ &= (1-\delta)^2(-p_{01}\log_2 p_{01} - p_{10}\log_2 p_{10}) \\ &+ (1-\delta)^2(-p_{00}\log_2 p_{00} - p_{11}\log_2 p_{11}) \\ &- \delta(1-\delta)(1+p_{00}-p_{11})\log_2(1+p_{00}-p_{11}) \\ &- \delta(1-\delta)(1-p_{00}+p_{11})\log_2(1-p_{00}+p_{11}) \\ &+ 2\delta(1-\delta)(p_{00}+p_{11}). \end{aligned}$$

If p_{00} and p_{11} are fixed, the mutual information is maximized under the case $p_{01} = p_{10}$. If p_{01} and p_{10} are fixed, the mutual information is maximized under the case $p_{00} = p_{11}$. By setting $p_{00} = p_{11} = t/2$ and $p_{01} = p_{10} = (1-t)/2$, where $0 \le t \le 1$, we obtain

$$I(X;Y) = (1-\delta)^2(1+h(t)) + 2\delta(1-\delta)t,$$

where $h()$ is the binary entropy function. It is maximized when $t_0 = 2^{2\delta/(1-\delta)}/(2^{2\delta/(1-\delta)}+1)$, except for $\delta = 1$. For $\delta = 1$, the mutual information is 0.

The mutual information is maximized under the non-uniform distribution. This complicates the design of error-correcting codes for this channel. We found that t_0 approximates $1/2$ if δ is small, i.e. the uniform distribution $p_{00} = p_{01} = p_{10} = p_{11} = 1/4$. The mutual information for $p_{00} = p_{01} = p_{10} = p_{11}$ is called the symmetric information rate I_S. In this case,

$$I_S = 2(1-\delta)^2 + \delta(1-\delta) = (1-\delta)(2-\delta). \tag{2}$$

Figure 2 depicts the channel capacity C and the symmetric information rate I_S. It can be seen that there is no significant gap between C and I_S in high-rate regions[†]. Therefore, in the following section, we evaluate the performance of error correction methods based on the symmetric information rate rather than the channel capacity.

4. Symmetric Information Rates Under Level-By-Level Error Correction

This section discusses the construction of error-correcting

[†] Above 1.6 is generally assumed in storage systems.

codes for the DSEF channel. The size of the input and output alphabets for the DSEF channel are four and seven, respectively. The authors are not aware of any error-correcting codes under such alphabet sizes. We focus on one of the two levels at a time. In other words, instead of considering the DSEF channel for input (x_1, x_2) and output (y_1, y_2), consider the DSEF channel as two channels, the first-level channel $\mathrm{Ch}_1 : x_1 \mapsto y_1$ and the second-level channel $\mathrm{Ch}_2 : x_2 \mapsto y_2$. In this way, the input alphabet of either channel will be $\{0, 1\}$, and the output alphabet will be $\{0, 1, ?\}$.

Consider an input data block of length n. Since each symbol is a pair of bits, the input data block is regarded as a n-by-2 matrix $(c_{i,j})_{1\le i\le n, j=1,2}$ over the binary alphabet. The first and second columns of the input data block are independently encoded codewords so they can be decoded independently. This is the basis for the term *level-by-level error correction*, as each level is decoded separately. The first and second columns go through the first-level channel Ch_1 and the second-level channel Ch_2, respectively. Let us now study the symmetric information rates for the Ch_1 and Ch_2.

4.1 The Second-Level Channel Ch_2

Analyzing the symmetric information rate of the second-level channel Ch_2 is easier than that of the first-level channel Ch_1, so we first study the symmetric information rate of Ch_2. A transition matrix $P_{Y_2|X_2}$ is obtained by

$$\begin{aligned} &P_{Y_2|X_2}(Y_2 = y_2 | X_2 = x_2) \\ &= \sum_{x_1, y_1} P_{Y|X}(Y = (y_1, y_2) | X = (x_1, x_2)) P_{X_1}(X_1 = x_1), \end{aligned}$$

where $P_{Y|X}$ is the transition matrix in (1) and P_{X_1} is the probability distribution for the first-level input. Then the transition matrix $P_{Y_2|X_2}$ is given by

$$\begin{pmatrix} (1-\delta)^2 & 0 \\ 0 & (1-\delta)^2 \\ 2\delta-\delta^2 & 2\delta-\delta^2 \end{pmatrix}.$$

Hence, Ch_2 is an erasure channel with the erasure probability $2\delta - \delta^2$. The symmetric information rate of Ch_2 is

$$I_2 = 1 - (2\delta - \delta^2) = (1-\delta)^2. \tag{3}$$

Since the second-level channel Ch_2 is an erasure channel, the symmetric information rate is identical to the channel capacity. Therefore, for an input data block in matrix form $(c_{i,j})_{1\le i\le n, j=1,2}$, the second column $(c_{i,2})_{1\le i\le n}$ is expected to be an erasure error-correcting codeword.

4.2 The First-Level Channel Ch_1

Next, we study the symmetric information rate of the first-level channel Ch_1. Similar to Ch_2, the transition matrix $P_{Y_1|X_1}$ is obtained by

$$\begin{aligned} &P_{Y_1|X_1}(Y_1 = y_1 | X_1 = x_1) \\ &= \sum_{x_2, y_2} P_{Y|X}(Y = (y_1, y_2) | X = (x_1, x_2)) P_{X_2}(X_2 = x_2), \end{aligned}$$

where $P_{Y|X}$ is the transition matrix in (1) and P_{X_2} is the probability distribution for the second-level input. The transition matrix $P_{Y_1|X_1}$ depends on P_{X_2}. By setting $P_{X_2}(X_2 = 0) = P_{X_2}(X_2 = 1) = 1/2$ due to the symmetric information assumption $p_{00} = p_{01} = p_{10} = p_{11} = 1/4$, then the transition matrix $P_{Y_1|X_1}$ is given by

$$\begin{pmatrix} (1-\delta)^2 + \frac{3}{2}\delta(1-\delta) & \frac{1}{2}\delta(1-\delta) \\ \frac{1}{2}\delta(1-\delta) & (1-\delta)^2 + \frac{3}{2}\delta(1-\delta) \\ \delta^2 & \delta^2 \end{pmatrix},$$

where the input alphabet is $\{0, 1\}$ and the output alphabet is $\{0, 1, ?\}$.

The first-level channel Ch_1 is regarded as a binary symmetric erasure channel with the flip probability $\frac{1}{2}\delta(1-\delta)$ and the erasure probability δ^2. In general, the symmetric information rate, which is actually equal to the channel capacity, of a binary symmetric erasure channel with the flip probability f and the erasure probability e is given as $(1-e)(1-h(\frac{f}{1-e}))$, where $h()$ is the binary entropy function. Hence, the symmetric information rate I_1 for the first-level channel Ch_1 is

$$\begin{aligned} I_1 &= (1-\delta^2)(1 - h(\frac{\delta}{2+2\delta})) \\ &= (1-\delta)(1 + \delta - h(\frac{\delta}{2+2\delta})(1+\delta)). \end{aligned} \tag{4}$$

4.3 The First-Level Channel Ch_1 After Error Correction for Ch_2

Here, we study the symmetric information rate of the first-level channel under the knowledge of error-correction result for Ch_2. First, guess the input pair (x_1, x_2) for Ch_1 and Ch_2, respectively from the output pair (y_1, y_2). If the output pair (y_1, y_2) is one of $(0,0), (0,1), (1,0), (1,1)$, it implies that no deletion error happened. Hence we may conclude $(x_1, x_2) = (y_1, y_2)$. If the output pair (y_1, y_2) is $(?, ?)$ and even if the error correction for Ch_2 is successfully done, the error-correction result does not increase the information for x_1, i.e., it is regarded as an erasure over Ch_1.

The remaining is the cases y_1 is a binary symbol but y_2 is the erasure symbol ?, i.e., $(y_1, y_2) = (0, ?)$ or $(1, ?)$. Assume the error-correcting result z for Ch_2 is different from y_1. The input pair is determined as $(x_1, x_2) = (y_1, z)$ since x_1 or x_2 must be y_1 but $x_2 = z \ne y_1$. Next, assume the error-correcting result z for Ch_2 is equal to y_1. The input pair (x_1, x_2) cannot be determined because both $(0, z)$ and $(1, z)$ are possible for (x_1, x_2).

We introduce symbols $\mathfrak{0}$ and $\mathfrak{1}$ in the case where the input is determined to be 0 and 1, respectively, by using the error-correction result for Ch_2. The transition matrix of the channel Ch_1 is regarded as the following channel Ch_1' with the input alphabet $\{0, 1\}$ and the output alphabet $\{\mathfrak{0}, 0, ?, 1, \mathfrak{1}\}$:

$$\begin{pmatrix} \frac{(1-\delta)(2-\delta)}{2} & 0 \\ \delta(1-\delta) & \frac{\delta(1-\delta)}{2} \\ \delta^2 & \delta^2 \\ \frac{\delta(1-\delta)}{2} & \delta(1-\delta) \\ 0 & \frac{(1-\delta)(2-\delta)}{2} \end{pmatrix}.$$

By the routine calculation of entropies for input, output, and joint distribution, the symmetric information rate for Ch_1' is given as

$$\begin{aligned} I_{2\to1} &= (1-\delta)(1+\delta-(\frac{3\log_2 3}{2}-1)\delta) \qquad (5)\\ &\sim (1-\delta)(1+\delta-1.38\delta). \end{aligned}$$

Let us clarify the mapping procedure from the output pair (y_1, y_2) and the error-correction result z for Ch_2 to the output w of Ch_1' as follows.

- If the output is $(0,0)$ or $(0,1)$, set $w = \mathfrak{0}$ as a reliable symbol.
- If the output is $(1,0)$ or $(1,1)$, set $w = \mathfrak{1}$ as a reliable symbol.
- If the output is $(?,?)$, set $w =?$ as a erasure symbol.
- If the output is $(0,?)$ and the estimated symbol z for Ch_2 is 0, set $w = 0$ as a doubtful symbol.
- If the output is $(0,?)$ and the estimated symbol z for Ch_2 is 1, set $w = \mathfrak{0}$ as a reliable symbol.
- If the output is $(1,?)$ and the estimated symbol z for Ch_2 is 0, set $w = \mathfrak{1}$ as a reliable symbol.
- If the output is $(1,?)$ and the estimated symbol z for Ch_2 is 1, set $w = 1$ as a doubtful symbol.

4.4 The Second-Level Channel Ch_2 After Error Correction for Ch_1

Similar to the previous section, we study the symmetric information rate of the second-level channel under the knowledge of error-correction result for Ch_1.

The cases for (y_1, y_2) is one of $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$ and $(?,?)$ are the same as in the previous section.

Let us consider the cases $(y_1, y_2) = (0,?)$ or $(1,?)$. Assume that the error-correcting result z for Ch_1 is different from y_1. The input is determined as $(x_1, x_2) = (z, y_1)$. Next, assume that the error-correcting result z for Ch_1 is equal to y_1. The input cannot be determined since both $(x_1, x_2) = (z, 0)$ and $(z, 1)$ may happen, i.e., $x_1 = z$ but the information for x_2 is still ?.

As in the previous section, we introduce symbols $\mathfrak{0}$ and $\mathfrak{1}$ in the case where the input is determined to be 0 and 1, respectively, by using the error-correction result for Ch_1.

The transition matrix of the channel Ch_2 is regarded as the following channel Ch_2' with the input alphabet $\{0, 1\}$ and the output alphabet $\{\mathfrak{0}, 0, ?, 1, \mathfrak{1}\}$:

$$\begin{pmatrix} (1-\delta)^2+\frac{1}{2}\delta(1-\delta) & 0 \\ 0 & 0 \\ \delta^2+\frac{3}{2}\delta(1-\delta) & \delta^2+\frac{3}{2}\delta(1-\delta) \\ 0 & 0 \\ 0 & (1-\delta)^2+\frac{1}{2}\delta(1-\delta) \end{pmatrix}.$$

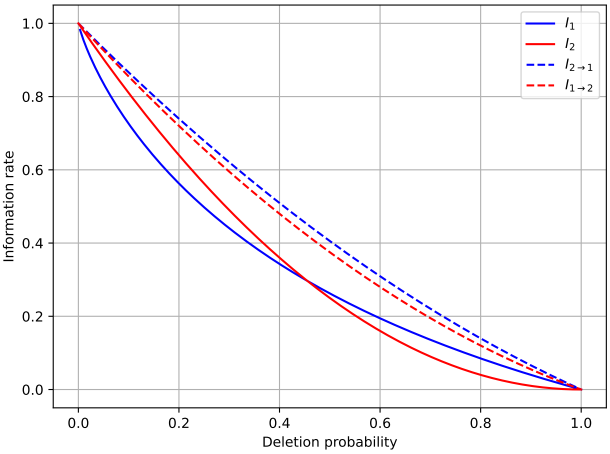


Fig. 3: Symmetric information rate (SIR) comparison. The I_1 is the SIR of the first-level channel Ch_1 in (4). The I_2 is the SIR of the second-level channel Ch_2 in (3). The $I_{2\to1}$ is the SIR of the Ch_1 after error correction for Ch_2 in (5). The $I_{1\to2}$ is the SIR of the Ch_2 after error correction for Ch_1 in (6).

Hence the channel Ch_2' is regarded as an erasure channel with the erasure probability $\delta^2 + \frac{3}{2}\delta(1-\delta)$.

The symmetric information rate, which is equal to the channel capacity, for Ch_2' is

$$\begin{aligned} I_{1\to2} &= 1-(\delta^2+\frac{3}{2}\delta(1-\delta)) \qquad (6)\\ &= (1-\delta)(1+\delta-1.5\delta). \end{aligned}$$

4.5 Symmetric Information Rate Comparison

We have obtained four symmetric information rates $I_1, I_2, I_{1\to2}$ and $I_{2\to1}$. Figure 3 shows these four symmetric information rates for each deletion probability. In the deletion-probability regions less than 0.455, I_2 is higher than I_1. Regarding symmetric information rate after decoding another level, $I_{2\to1}$ is higher than $I_{1\to2}$ for any deletion probabilities.

Let us consider the combination of two symmetric information rates. For example, the sum of I_1 and I_2 means the compound symmetric information rate of independently decoding for two channels. On the other hand, the sum of I_1 and $I_{1\to2}$ means the symmetric information rate by decoding for the first-level channel first and the second-level channel next. The other one is the sum of I_2 and $I_{2\to1}$ that means the symmetric information rate by decoding for the second-level channel first and the first-level channel next.

- The sum of I_1 and I_2 is

$$I_1 + I_2 = (1-\delta)(2-h(\frac{\delta}{2+2\delta})(1+\delta)). \qquad (7)$$

- The sum of I_1 and $I_{1\to2}$ is

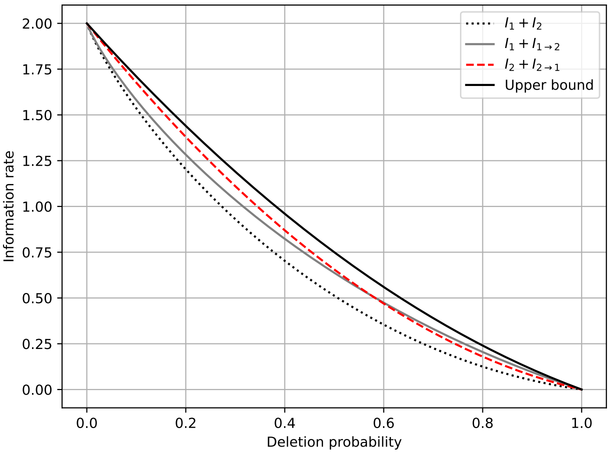


Fig.4: Compound information rates and the upper bound. The upper bound represents the symmetric information rate shown (SIR) in (2). It represents the theoretical maximum of other compound information rates. The $I_1 + I_2$ represents the sum of the respective SIRs when a received codeword in each level is decoded independently. The $I_1 + I_{1\to2}$ represents the SIR when a received codeword at the first level is decoded, followed by the decoding of a received codeword at the second using the results from the first. The $I_2 + I_{2\to1}$ represents the SIR when a received codeword at the second level is decoded, followed by the decoding of a received codeword at the first using the results from the second. The $I_2 + I_{2\to1}$ approaches the upper bound, especially when the deletion probability is small.

$$I_1 + I_{1\to2} = (1-\delta)(2 - h(\frac{\delta}{2+2\delta})(1+\delta) + 0.5\delta). \tag{8}$$

- The sum of I_2 and $I_{2\to1}$ is

$$I_2 + I_{2\to1} = (1-\delta)(2 - (\frac{3\log_2 3}{2} - 1)\delta). \tag{9}$$

Figure 4 depicts compound information rates shown in (7), (8) and (9), and the upper bound, i.e., the symmetric information rate for the double-stack erasure channel shown in (2). The dashed line, representing the sum of $I_2+I_{2\to1}$, and the gray line, representing the sum of $I_1 + I_{1\to2}$, intersect at the deletion probability of $\delta \sim 0.5756$. The sum of $I_2+I_{2\to1}$ approaches the upper bound. The difference between them is

$$\begin{aligned} I_S - (I_2 + I_{2\to1}) &= (\frac{3\log_2 3}{2} - 2)\delta(1-\delta) \\ &\sim 0.38\delta(1-\delta), \end{aligned}$$

and it is small under a small deletion probability δ.

5. Demonstration of Level-By-Level Error Correction

This section demonstrates the error-correcting capability of the level-by-level error correction. Since decoding the second level first and then decoding the first level described in the previous section achieves the highest compound information rate, we demonstrate this method in our experiments.

Figure 5 illustrates the system model used for demonstrating the level-by-level error correction in the multi-track DSEF channel. Two codewords $\boldsymbol{c}_1 = (c_{1,1}, c_{2,1}, \ldots, c_{n,1}) \in C_1$ and $\boldsymbol{c}_2 = (c_{1,2}, c_{2,2}, \ldots, c_{n,2}) \in C_2$ are prepared for input into the DSEF channel. The coding rates of C_1 and C_2 can be identical, however as explained in Section 4, each level of the DSEF channel has different symmetric information rates for any given deletion probability. Therefore, the coding rates of C_1 and C_2 should differ according to each level's SIR. Outputs from the DSEF channel are received as sequences $\boldsymbol{y}_1 = (y_{1,1}, y_{2,1}, \ldots, y_{n,1})$ and $\boldsymbol{y}_2 = (y_{1,2}, y_{2,2}, \ldots, y_{n,2})$ corresponding to the codewords $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, respectively. First, the received sequence $\boldsymbol{y}_2$ is decoded by the L2 decoder, which corrects erasures at the second level, producing an estimated codeword $\hat{\boldsymbol{c}}_2 = (\hat{c}_{1,2}, \hat{c}_{2,2}, \ldots, \hat{c}_{n,2})$. Next, the estimated codeword $\hat{\boldsymbol{c}}_2$ from the L2 decoder is fed into the L1 decoder, which corrects errors in $\boldsymbol{y}_1$ at the first level. The L1 decoder calculates the likelihood of the channel output $\boldsymbol{y}_1$ based on the transition matrix of the DSEF channel discussed in Sect. 4.3 using the estimated codeword $\hat{\boldsymbol{c}}_2$ and the channel outputs $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. The log-likelihood ratio $LLR(y_{i,1}, y_{i,2}, \hat{c}_{i,2})$ of the channel output $y_{i,1}$ is calculated by

$$LLR(y_{i,1}, y_{i,2}, \hat{c}_{i,2}) = \begin{cases} 0 & \text{if } y_{i,1} = \,? \\ \Omega(y_{i,1}) & \text{if } \gamma(y_{i,1}, y_{i,2}, \hat{c}_{i,2}) \\ \Lambda(y_{i,1}) & \text{otherwise,} \end{cases}$$

where the condition $\gamma(y_{i,1}, y_{i,2}, \hat{c}_{i,2})$ is defined as

$$\begin{aligned} &\gamma(y_{i,1}, y_{i,2}, \hat{c}_{i,2}) = \\ &y_{i,2} \neq \,? \vee y_{i,2} = \,? \wedge \hat{c}_{i,2} \neq \,? \wedge y_{i,1} \neq \,? \wedge y_{i,1} \neq \hat{c}_{i,2}, \end{aligned}$$

where $\vee$ and $\wedge$ are logical OR and AND operators, respectively. Functions $\Omega(y_{i,1})$ and $\Lambda(y_{i,1})$ are defined as

$$\Omega(y_{i,1}) = \begin{cases} +\infty & \text{if } y_{i,1} = 0 \\ -\infty & \text{otherwise,} \end{cases}$$

and

$$\Lambda(y_{i,1}) = \begin{cases} \log(2) & \text{if } y_{i,1} = 0 \\ -\log(2) & \text{otherwise,} \end{cases}$$

respectively. Interestingly, the log-likelihood ratio of the channel output $\boldsymbol{y}_1$ here is independent of the deletion probability δ. The L1 decoder then corrects the errors in $\boldsymbol{y}_1$, resulting in the corrected output $\hat{\boldsymbol{c}}_1$. This two-step level-by-level decoding ensures robust error correction.

Low-density parity-check (LDPC) codes [17] are utilized at each level in the DSEF channel since the sum-product algorithm, a decoding method for LDPC codes, effectively corrects erasures and bit-flip errors, even when they occur simultaneously. Other codes, like polar codes, will probably also be applicable, but we believe the evaluation using LDPC codes provides sufficient insights into their performance for our level-by-level decoding. The LDPC codes used in our experiments are regular quasi-cyclic LDPC (QC-LDPC) codes [18] with column weight three. Three different LDPC codes, C_1, C_2, and C_3, with different code rates, are used in our experiments.

Their parity-check matrices are 8×64, 10×64, and 18×64 arrays consisting of 128×128 cyclic-permutation or zero matrices for C_1, C_2, and C_3, respectively. Since all LDPC codes are column weight three, each array column consists of three circulant permutation matrices (CPMs) and zero matrices. The shift values of the CPMs are determined by progressive edge growth (PEG) [19] . CPMs are distributed uniformly so that the row weights of all parity-check matrices are also uniform. The sizes of the parity-check matrices generated in this way are 1024×8192, 1280×8192, and 2304×8192 for C_1, C_2, and C_3, respectively. The coding rates of C_1, C_2, and C_3 are 0.875, 0.84375, and 0.71875, respectively. The maximum number of iterations for the sum-product algorithm is 50. Channel likelihood is calculated by using the transition matrices described in Sect. 4.

Figure 6 shows the experimental results. All decoding curves except 'C_1 L1' begin to fall around the deletion probability of 0.065. Symmetric information rates at the deletion probability of 0.065 are shown in Table 1. The difference between $I_{2\to1}$ and I_1 is 0.112. You can see that the coding-rate difference between C_1 and C_3 is 0.15625. Therefore, we roughly obtain information-rate gain by decoding the second level first and then decoding the first level.

The performance of 'C_3 L1' is worse than both that of 'C_2 L2' and that of 'C_1 L1 after C_2 L2' in Fig. 6 below the decoding-error probability of 10^{-1} even though the parity length of C_3 is equal to the sum of the parity lengths of C_1 and C_2. If the first-level decoding cannot be performed correctly, the performance of decoding the first level first and then decoding the second level cannot be achieved. Therefore, it is not expected that the performance of decoding the first level first and then decoding the second level is better.

Table 2 shows the deletion thresholds for code rates of C_1, C_2, and C_3. The deletion threshold for any given symmetric information rate I' is defined as the maximum deletion probability δ such that the symmetric information rate $I(\delta)$ calculated from the deletion probability δ does not fall below I'. The difference between the threshold of $I_{2\to1} = 0.875$ and that of $I_1 = 0.875$ is equal to 0.058. It is observed that the distance between the decoding curve of 'C_1 L1 after C_2 L2' and that of 'C_1 L1' is 0.05 in Fig. 6. The deletion thresholds in Table 2 can be seen to be larger than the deletion probabilities in Fig. 6. This indicates that there is room for code optimization. The regular LDPC codes used in this experiment are generally not the class of codes that achieve the Shannon limit. It is conceivable that by optimizing the structure of LDPC codes, codes that approach the deletion threshold can be constructed. However, this is beyond the scope of this paper and will be addressed in future work.

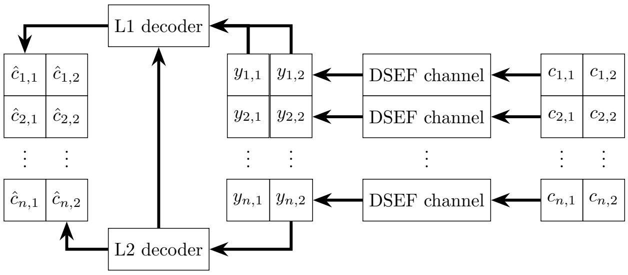


Fig.5: System model of the demonstration involves decoding the second level first, followed by decoding the first level.

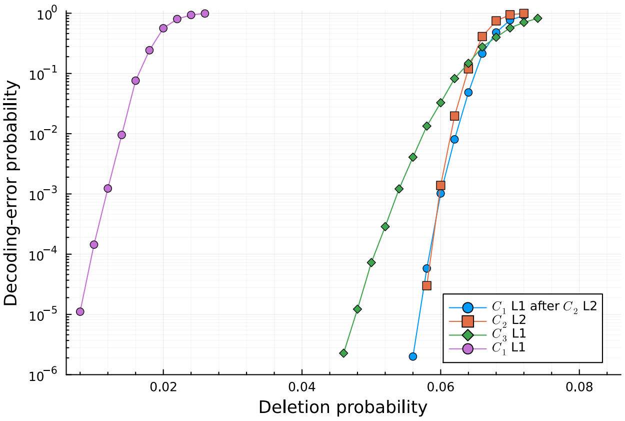


Fig.6: Error-correcting capability of three different LDPC codes C_1, C_2, and C_3. The 'C_1 L1 after C_2 L2' represents the error-correcting capability of C_1 at the first level after the decoding of C_2 at the second level. The 'C_2 L2' represents the error-correcting capability of C_2 at the second level. The 'C_3 L1' represents the error-correcting capability of C_3 at the first level. The 'C_1 L1' represents the error-correcting capability of C_1 at the first level.

Table 1: Symmetric information rates at the deletion probability of 0.065.

$I_{2\to1}$	I_2	I_1
0.912	0.874	0.800

Table 2: Deletion thresholds for code rates of C_1, C_2, and C_3.

$I_{2\to1} = 0.875$	$I_1 = 0.875$	$I_2 = 0.84375$	$I_1 = 0.71875$
0.0931	0.0351	0.0814	0.104

6. Conclusion

A multi-stack erasure-filled (MSEF) channel inspired by racetrack memory with multiple racetracks was introduced. Channel capacity, symmetric information rate, and error-correcting capability for a double-stack erasure-filled (DSEF) channel, an MSEF channel of two levels, were discussed. As verified through Sections 4 and 5, good error-correction performance was obtained by decomposing the DSEF channel into two binary-input ternary-output channels and by decoding the second level first and then decoding the first level. In contrast, the DSEF channel is initially a quaternary-input septenary-output channel. In particular, it can be argued that the error correction for the existing channel was adequate, which is a meaningful achievement in terms of implementation.

As future work, we aim to consider more realistic channels. Although we treated the filled symbols as erasures to facilitate the analysis in this paper, the random-binary-filled channel that fills a sequence of bit values will likely be more realistic. We plan to examine this point in more detail in future research and conduct analyses using the random-binary-filled channel as well.

References

[1] G. Mappouras, A. Vahid, R. Calderbank, and D.J. Sorin, “Greenflag: Protecting 3D-racetrack memory from shift errors,” 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp.1–12, IEEE, 2019.

[2] Y.M. Chee, H.M. Kiah, A. Vardy, V.K. Vu, and E. Yaakobi, “Coding for racetrack memories,” IEEE Transactions on Information Theory, vol.64, no.11, pp.7094–7112, 2018.

[3] A. Vahid, G. Mappouras, D.J. Sorin, and R. Calderbank, “Correcting two deletions and insertions in racetrack memory,” arXiv preprint arXiv:1701.06478, 2017.

[4] M. Cheraghchi and J. Ribeiro, “An overview of capacity results for synchronization channels,” IEEE Trans. Inf. Theory, vol.67, no.6, pp.3207–3232, June 2021.

[5] S. Li, X. Lin, P. Li, S. Zhao, Z. Si, G. Wei, B. Koopmans, R. Lavrijsen, and W. Zhao, “Ultralow power and shifting-discretized magnetic racetrack memory device driven by chirality switching and spin current,” arXiv [cond-mat.mes-hall], May 2023.

[6] A.J. Annunziata, M.C. Gaidis, L. Thomas, C.W. Chien, C.C. Hung, P. Chevalier, E.J. O’Sullivan, J.P. Hummel, E.A. Joseph, Y. Zhu, T. Topuria, E. Delenia, P.M. Rice, S.S.P. Parkin, and W.J. Gallagher, “Racetrack memory cell array with integrated magnetic tunnel junction readout,” 2011 International Electron Devices Meeting, pp.24.3.1–24.3.4, Dec. 2011.

[7] S. Ghosh, A. Iyengar, S. Motaman, R. Govindaraj, J.W. Jang, J. Chung, J. Park, X. Li, R. Joshi, and D. Somasekhar, “Overview of circuits, systems, and applications of spintronics,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol.6, no.3, pp.265–278, Sept. 2016.

[8] S. Ollivier, S. Longofono, P. Dutta, J. Hu, S. Bhanja, and A.K. Jones, “Toward comprehensive shifting fault tolerance for Domain-Wall memories with PIETT,” IEEE Trans. Comput., pp.1–14, 2022.

[9] M. Hagiwara, “Multi Deletion/Substitution/Erasure Error-Correcting codes for information in array design,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E106.A, no.3, pp.368–374, 2023.

[10] L. Welter, R. Bitar, A. Wachter-Zeh, and E. Yaakobi, “Multiple Criss-Cross insertion and deletion correcting codes,” IEEE Trans. Inf. Theory, vol.68, no.6, pp.3767–3779, June 2022.

[11] Y.M. Chee, M. Hagiwara, and V. Van Khu, “Two dimensional deletion correcting codes and their applications,” 2021 IEEE International Symposium on Information Theory (ISIT), pp.2792–2797, IEEE, July 2021.

[12] R. Shibata and H. Yashima, “Symbol-Level detection with matched Non-Binary LDPC codes for position errors in racetrack memories,” IEEE Trans. Magn., vol.59, no.2, pp.1–9, Feb. 2023.

[13] H. Koremura and H. Kaneko, “Insertion/Deletion/Substitution error correction by a modified successive cancellation decoding of polar code,” IEICE Trans. Fundamentals, vol.E103.A, no.4, pp.695–703, 2020.

[14] R. Goto and K. Kasai, “Sparse graph codes for channels with synchronous errors,” IEICE Trans. Fundamentals, vol.E101.A, no.12, pp.2064–2071, 2018.

[15] V.I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” Soviet Physics Dokl., vol.10, no.8, pp.707–710, 1966.

[16] W. Anacker, G.F. Bland, P. Pleshko, and P.E. Stuckert, “On the design and performance of a small 60-nsec destructive readout magnetic film memory,” IBM J. Res. Dev., vol.10, no.1, pp.41–50, Jan. 1966.

[17] R. Gallager, “Low-density parity-check codes,” IRE Transactions on information theory, vol.8, no.1, pp.21–28, 1962.

[18] M.P.C. Fossorier, “Quasi-cyclic low-density parity-check codes from circulant permutation matrices,” IEEE Transactions on Information Theory, vol.50, no.8, pp.1788–1793, 2004.

[19] X.Y. Hu, E. Eleftheriou, and D.M. Arnold, “Progressive edge-growth Tanner graphs,” GLOBECOM’01: IEEE Global Telecommunications Conference, pp.995–1001, IEEE, 2001.

Hironori Uchikawa joined the Research and Development Center at Toshiba Corporation, where he was engaged in research on MIMO-OFDM systems in 2003. He also developed error-correcting codes, especially low-density parity check (LDPC) codes, for flash memory storage systems. He received the Ph.D. degree in electrical engineering from Tokyo Institute of Technology in 2012. He is currently a Chief Specialist in the Memory Division at Kioxia Corporation. His research interests include coding theory, information theory, communication theory and their applications to storage systems. He received the 2008 Young Researcher’s Award of the IEICE. He is also a Senior Member of the IEEE.

Manabu Hagiwara was born in Ashikaga, Japan, on July 26, 1974. He received the B.E. degree in mathematics from Chiba University in 1997, and the M.E. and Ph.D. degrees in mathematical science from the University of Tokyo in 1999 and 2002, respectively. From 2002 to 2005 he was a postdoctoral fellow at IIS, the University of Tokyo. From 2005 to 2012 he was a research scientist with National Institute of Advanced Industrial Science and Technology (AIST). From 2011 to 2012 he was a visiting scholar with the University of Hawaii. From 2013 to 2020 he was an associate professor with Chiba University. He was the general co-chair of the International Symposium on Information Theory and its Application 2020 (ISITA2020), Oahu, Hawaii. Currently, he is a full professor with Graduate School of Science, Chiba University. His current research interests include coding theory and combinatorics.
Prof. Hagiwara was the recipient of the ComEX Top Downloaded Letter Award, from IEICE, in April 2020 and February 2020 and Kioxia Research Grant Program Outstanding Research Award, from Kioxia, in June 2020.