Average Bit Erasure Probability of Regular LDPC Code Ensembles under MAP Decoding over BEC

Takayuki ITSUI†, Nonmember, Kenta KASAI†(a), Ryoji IKEGAYA††, Tomoharu SHIBUYA†††, Members, and Kohichi SAKANIWA††, Fellow

SUMMARY The average bit erasure probability of a binary linear code ensemble under maximum a-posteriori probability (MAP) decoding over binary erasure channel (BEC) can be calculated with the average support weight distribution of the ensemble via the EXIT function and the shortened information function. In this paper, we reformulate the relationship between the average bit erasure probability under MAP decoding over BEC and the average support weight distribution for a binary linear code ensemble. Then, we formulate the average support weight distribution for a binary linear code ensemble under MAP decoding over BEC for regular LDPC code ensembles.

key words: regular LDPC code ensemble, a-posteriori probability decoding, binary erasure channel, EXIT function, shortened information function, support weight distribution

1. Introduction

Low-density parity-check (LDPC) codes are binary linear error-correcting codes originally proposed by Gallager in 1963 [2]. LDPC codes are defined by sparse parity-check matrices. The parity-check matrices are associated with bipartite graphs which are obtained by connecting variable nodes and check nodes sparsely. LDPC codes have been studied through an ensemble-averaged analysis. In this paper, we consider regular LDPC code ensembles and study the average error-correcting performance.

We study the error-correcting performance under maximum a-posteriori probability (MAP) bit decoding over binary erasure channel (BEC). In a study of the error-correcting performance, investigation of MAP decoding performance is meaningful because MAP decoding is optimal with the assumption that each codeword is chosen with the uniform probability, and thereby the performance under MAP decoding indicates the potential capability of the employed code. In the asymptotic case, the average bit extrinsic erasure probability of regular LDPC code ensembles under MAP decoding over BEC was derived by Méasson et al. [4]. In this paper, the exact average bit erasure probability of regular LDPC code ensembles under MAP decoding over BEC are analyzed for the finite case. The bit extrinsic erasure probability of a binary linear code equals the extrinsic information transfer (EXIT) function of the code [5]. It is known that the EXIT function is related to the support weight distribution via the punctured information function [1], which is usually called information function. However, the expression of the punctured information function of a code depends on its dimension [3]. Since an ensemble contains codes with various dimensions in general, it is hard to use the form in terms of the punctured information function to obtain the average EXIT function for an ensemble. Then, we focus on the shortened information function, which has a relationship with the punctured information function [3]. We can calculate the average EXIT function with the average support weight distribution via the shortened information function for a binary linear code ensemble.

In this paper, we reformulate the relationship between the bit erasure probability and the support weight distribution, which does not depend on the dimension of a code, using the shortened information function. This relationship enables us to calculate the average bit erasure probability with the average support weight distribution for a binary linear code ensemble. Then, we formulate the average support weight distribution of regular LDPC code ensembles.

This paper is organized as follows. In Sect. 2, we give the definitions of the EXIT function, the support weight, the punctured information function, the shortened information function, the support weight distribution and the auxiliary shortened information function. In Sect. 3, we formulate the relationship between the average bit erasure probability and the average support weight distribution for a binary linear code ensemble. Then, we formulate the average support weight distribution of regular LDPC code ensembles. In Sect. 4, we conclude this paper. Some proofs are provided in Appendices.

2. Preliminaries

2.1 EXIT Function for BEC

Assume that we transmit over BEC with erasure proba-
In this paper, we define and use the support weight of a binary linear code $C$ with length $n$. Let $X = \{X_1, X_2, \ldots, X_n\}$ be the random variable corresponding to the codeword randomly chosen with the uniform probability from $C$ and $Y = \{Y_1, Y_2, \ldots, Y_n\}$ be the random variable corresponding to the received word after transmission. The **EXIT function** associated to the $\ell$th position of a binary linear code $C$ is defined as

$$h_C(\ell) := H(X_\ell | Y_{\ell+1}, \ldots, Y_n),$$

where $H(\cdot)$ denotes the conditional entropy, $[n] := \{1, 2, \ldots, n\}$ and $Y_{\ell+1} \ldots Y_n = Y_{\ell+1} \cdot Y_{\ell+1} \cdots Y_n$. In [5, §3.14.2], the EXiT function of a binary linear code $C$ is defined as

$$h_C(\ell) := \frac{1}{n} \sum_{\ell=1}^{n} H(X_\ell | Y_{\ell+1}, \ldots, Y_n).$$

Note that in [1], the EXIT function is defined as

$$I_E(\ell) := \frac{1}{n} \sum_{\ell=1}^{n} I(X_\ell : Y_{\ell+1}, \ldots, Y_n) = 1 - h_C(\ell).$$

In this paper, we define and use $h_C(\ell)$ as the EXIT function of a binary linear code $C$ for convenience. Let $P_{h,\ell}(C)$ be the bit erasure probability at the $\ell$th position of a binary linear code $C$ under MAP decoding. It is known [5, §3.14.2] that

$$P_{h,\ell}(C) = \epsilon \cdot h_C(\ell).$$

Furthermore, we define the bit erasure probability $P_{h}(C)$ of a binary linear code $C$ under MAP decoding as

$$P_{h}(C) := \frac{1}{n} \sum_{\ell=1}^{n} P_{h,\ell}(C) = \epsilon \cdot h_C(\ell). \tag{1}$$

### 2.2 Support Weight

The **support** of a binary vector $x$ is defined as

$$\chi(x) := \{\ell : x_\ell = 1\},$$

and the support of a set $C$ of binary vectors is defined as

$$\chi(C) := \bigcup_{x \in C} \chi(x).$$

The **support weight** $w(C)$ of a set $C$ of binary vectors is defined as the number of positions at which the components of one or more vectors in $C$ are 1, namely

$$w(C) := \#\chi(C).$$

For example, the following set of binary vectors

$$C = \{[000], [001], [010], [011]\}$$

has support weight

$$w(C) = \#(\emptyset \cup \{3\} \cup \{2\} \cup \{2, 3\}) = 2.$$
By substituting Eq. (2) into Eq. (3), we have the following lemma.

**Lemma 1:** The EXIT function \( h_C(\epsilon) \) of a binary linear code \( C \) can be written in terms of the unnormalized shortened information function \( \tilde{m}_n(C) \) as

\[
h_C(\epsilon) = \frac{1}{n} \sum_{g=1}^{n} (1 - \epsilon)^{n-1} \epsilon^{n-g} \cdot [g \cdot \tilde{e}_g(1) - (n - g + 1) \cdot \tilde{e}_{g-1}(1)].
\]

(3)

For example, for the code \( C \)

The support weight distribution is as follows.

\[
\text{For a binary linear code } C, \text{ the } r\text{th support weight distribution } A_r(C) (i = 0, 1, \ldots, n) \text{ is defined as}
\]

\[
A_r^i(C) := \# \{ D \subset C : D \text{ is an } r\text{-dimensional subspace of } C \text{ with } w(D) = i \}.
\]

For example, for the code \( C = \{[000], [001], [010], [011]\} \)

\[
A_0^0(C) = \# \{ \text{span}([000]) \} = 1,
A_1^0(C) = \# \{ \text{span}([001]), \text{span}([010]) \} = 2,
A_2^0(C) = \# \{ \text{span}([00], [11]) \} = 1,
A_3^0(C) = \# \{ \text{span}([01], [01]) \} = 1,
\]

where \( \text{span}[] \) denotes the linear space spanned by argument vectors. The 1st support weight distribution \( A_1^i(C) \) is identical to the weight distribution of non-zero codewords of \( C \). Since the dimension of a subspace cannot be larger than its support weight, we have \( A_r^i(C) = 0 \) for \( r > i \). Also, the \( r\text{th auxiliary shortened information function in } h \text{ positions is given by} \]

\[
m_h^r(C) = \sum_{s=0}^{n-r} (-1)^{s/2} \left[ \frac{n-r}{s} \right] A_{r+s}^i(C),
\]

(6)

where \( \left[ \frac{n}{s} \right] = \binom{n}{s} \) is the binomial coefficient. Solving the set of Eq. (6) gives the following theorem.

**Theorem 1:** For a binary linear code \( C \), the auxiliary shortened information function \( m_h^r(C) \) in \( h \) positions is given by

\[
m_h^r(C) = \sum_{s=0}^{n-r} (-1)^{s/2} \left[ \frac{n-r}{s} \right] A_{r+s}^i(C),
\]

and the unnormalized shortened information function \( m_0^r(C) \) in \( h \) positions is given by

\[
m_0^r(C) = \sum_{r=0}^{h} \sum_{s=0}^{n-r} (-1)^{s/2} \left[ \frac{n-r}{s} \right] A_{r+s}^i(C).
\]

From Eq. (1), Lemma 1, Theorem 1 and the linearity of the expectation, we can calculate the average bit erasure probability under MAP decoding over BEC with the average support weight distribution for a binary linear code ensemble.

**3. Average Performance under MAP Decoding over BEC and Average Support Weight Distribution**

In this section, we formulate the relationship between the average bit erasure probability and the average support weight distribution for a binary linear code ensemble. Then, we formulate the average support weight distribution of regular LDPC code ensembles.

**3.1 Average Bit Erasure Probability and Average Support Weight Distribution**

The \( r\text{th support weight distribution } A_r^i(C) (i = 0, 1, \ldots, n) \) and the auxiliary shortened information functions \( m_h^r(C) \) (\( q = 0, 1, \ldots, n \)) in \( h \) positions satisfy the following linear equation for \( r = 0, 1, \ldots, n, h = 0, 1, \ldots, n \) [6, Lemma 1].

\[
\sum_{q=0}^{n-r} \left[ \frac{q}{r} \right] m_h^q(C) = \sum_{i=0}^{\lfloor n/h \rfloor} \left( \frac{n-i}{n-h} \right) A_r^i(C),
\]

(6)

where \( \lfloor s \rfloor = \frac{s}{2} - \frac{s}{2} \) for \( s \geq 1 \) and \( \lfloor 0 \rfloor := 1 \).

**3.2 Average Support Weight Distribution of Regular LDPC Code Ensembles**

Let \( E_C[A_r^i(C)] \) denotes the expectation over an ensemble \( C \).

The formulation of the average support weight distribution \( E_C[A_r^i(C)] \) (\( r = 0, 1, \ldots, n, i = 0, 1, \ldots, n \)) of a binary linear code ensemble \( C \) is not straightforward. Then we will formulate the average support weight distribution by counting the average number of \( r\text{-tuples of support weight } i \) \( (r = 0, 1, \ldots, n, i = 0, 1, \ldots, n) \) for a binary linear code ensemble \( C \) as described in the following theorem. The theorem tells us that we can derive the average support weight distribution through counting an average substitute distribution for a binary linear code ensemble.
Theorem 2: Let $A_C$ be an $(n + 1) \times (n + 1)$ matrix whose column and row indices start from 0, and $(s, i)$ entry is $E[C_i](C)$ for a binary linear code ensemble $C$. $A_C$ is obtained by solving

$$FA_C = Y_C,$$

where $F$ is an $(n + 1) \times (n + 1)$ invertible lower triangular matrix whose column and row indices start from 0, and $(r, s)$ entry $F_{rs}$ is given by

$$F_{rs} = \sum_{t=0}^{r} (-1)^{t} 2^{\binom{t}{2}} \binom{s}{t} 2^{(s-r)t},$$

and $Y_C$ is an $(n + 1) \times (n + 1)$ matrix whose column and row indices start from 0, and $(r, i)$ entry $Y_{ri}$ is

$$Y_{ri} = \begin{cases} 1 & \text{if } r = i = 0, \\
0 & \text{if } r = 0, i > 0, \\
\sum_{C \in \mathcal{C}} P(C) \#(\{x_1, x_2, \ldots, x_r\} : x_p \in C (p = 1, 2, \ldots, r), \\
\text{span}(x_1, x_2, \ldots, x_r) \text{ is an } s\text{-dimensional subspace of } C, \\
u((x_1, x_2, \ldots, x_r)) = i) & \text{otherwise,} \end{cases}$$

where $P(C)$ denotes the probability that a randomly chosen code is $C$.

Proof: Let $r > 0$ and $i$ be fixed. Then, $Y_{ri}$ can be written as follows.

$$Y_{ri} = \sum_{C \in \mathcal{C}} p(C) \sum_{s=0}^{r} \#(\{x_1, x_2, \ldots, x_r\} : x_p \in C (p = 1, 2, \ldots, r), \\
\text{span}(x_1, x_2, \ldots, x_r) \text{ is an } s\text{-dimensional subspace of } C, \\
u((x_1, x_2, \ldots, x_r)) = i)$$

$$= \sum_{C \in \mathcal{C}} p(C) \sum_{s=0}^{r} \sum_{D \in D} \#T^r(D),$$

where we define $T^r(D)$ as the set of $r$-tuples which generate $D$, namely

$$T^r(D) := \{\{x_1, x_2, \ldots, x_r\} : x_p \in D (p = 1, 2, \ldots, r), \\
\text{span}(x_1, x_2, \ldots, x_r) = D\}.$$
randomly. Then, we place the uniform probability distribution on the set of bipartite graphs. This is the ensemble \( \mathcal{G}(n, c, d) \) of bipartite graphs. The cardinality of \( \mathcal{G}(n, c, d) \) is \((cn)!\). We associate a bipartite graph \( G \in \mathcal{G}(n, c, d) \) with an \((nc/d) \times n\) parity-check matrix \( H_G \) where \((i, j)\) entry is 1 if the \( j \)-th variable node is connected to the \( i \)-th check node an odd number of times, and 0 otherwise. Then, we define \( C_G := \{ x \in \{0, 1\}^n : H_G x^T = 0^T \} \). In this thesis, we identify an ensemble of bipartite graphs with an ensemble of codes via the above association. The average support weight distribution \( E_{\mathcal{G}(n, c, d)}[A'_i(C_G)] \) of a \((c, d)\)-regular LDPC code ensemble \( \mathcal{G}(n, c, d) \) is given by

\[
E_{\mathcal{G}(n, c, d)}[A'_i(C_G)] = \sum_{G \in \mathcal{G}(n, c, d)} P(G) A'_i(C_G) = \frac{1}{(cn)!} \sum_{G \in \mathcal{G}(n, c, d)} A'_i(C_G),
\]

where \( P(G) \) denotes the probability that a randomly chosen bipartite graph is \( G \).

The following theorem formulates the average number of \( r \)-tuples of support weight \( i \) \((r = 0, 1, \ldots, n, i = 0, 1, \ldots, n)\) for a regular LDPC code ensemble \( \mathcal{G}(n, c, d) \).

**Theorem 3:** For a regular LDPC code ensemble \( \mathcal{G}(n, c, d) \), \((r, i)\) entry \( Y_{ri} \) of \( Y_{\mathcal{G}(n, c, d)} \) introduced in Theorem 2 is given by

\[
Y_{ri} = \begin{cases} 1 & \text{if } r = i = 0, \\ \frac{1}{(cn)!} \sum_{i_j \in B_0^r} \prod_{b \in B_0^r} \binom{i!}{j_b!} & \text{if } r = 0, i > 0, \\ \text{cof} \left[ F_r \left( \left\{ x_b : b \in B_0^r \right\} \right) \right] & \text{if } r = 0, i = 0, \\ \text{cof} \left[ F_r \left( \left\{ x_b : b \in B_0^r \right\} \right) \right] & \text{if } r > 0, i > 0, \\ \end{cases}
\]


\[
F_r \left( \left\{ x_b : b \in B_0^r \right\} \right) = \frac{1}{2^r} \sum_{x \in \{0, 1\}^n} \sum_{g \in \{0, 1\}^r} \left( 1 + \sum_{b \in B_0^r} (-1)^{\sum_{p=1}^r a_p b_p x_b} \right)^d
\]

where \( F_r \left( \left\{ x_b : b \in B_0^r \right\} \right) \) is the polynomial such that the coefficient of \( \prod_{b \in B_0^r} x_b \) is the number of ways of choosing \( k_b \) sockets for \( b \in B_0^r \) from \( d \) sockets on a check node, i.e., \( \frac{d}{\prod_{b \in B_0^r} \binom{i_b!}{j_b!}} \). In addition, since for each codeword, the variable nodes at which its corresponding components are 1 are connected to each check node an even number of times, the following constraint is satisfied.

\[
\sum_{b \in B_0^r} k_b \text{ is even } (p = 1, 2, \ldots, r)
\]
The derivation of the polynomial $F_r\left(\{x_b : b \in B_0^r\}\right)$ is provided in Appendix A. As the second step, for each way of choosing sockets on the check node side, we count the number of ways of connecting sockets on the variable node side and sockets on the check node side. It does not depend on the way of choosing sockets on the check node side and is given by

$$
\#U_r^{C}_i\left(\{j_b : b \in B_0^r\}\right) \text{ is given by Eq. (14) multiplied by Eq. (16). By substituting } \#U_r^{C}_i\left(\{j_b : b \in B_0^r\}\right) \text{ into Eq. (13), we have}
$$

$$
Y_r = \frac{1}{(cn)!} \left(\begin{array}{c}n \\ i \end{array}\right) \sum_{\sum_{b \in B_0^r, j_b=i}} i! \prod_{b \in B_0^r} \left(\frac{j_b}{b}\right) 
\cdot \text{coe } F_r\left(\left\{x_b : b \in B_0^r\right\}\right)^{j_{b=x}^n} \prod_{b \in B_0^r} x_b^{j_b}
\cdot (c(n-i))! \prod_{b \in B_0^r} \left(\frac{c(j_b)!}{b}\right).
$$

3.3 Numerical Example

In this subsection, we show a numerical example. We draw the graph of the average bit erasure probability under MAP decoding over BEC for a regular LDPC code ensemble $G(6, 2, 3)$ in Fig.1. When we calculate the average bit erasure probability of regular LDPC code ensembles under MAP decoding over BEC for the finite case. There is difficulty in applying this work to calculating the performance of practical code ensembles, because lots of computations and memories are needed. Also, it seems difficult to find some knowledge about the mutuality between code-length and performance because of the complexity of the expression in Theorem 3. However, we suspect that computational effort will be reduced by some improvement of computation scheme, such as use of symmetry in expanding $F_r\left(\left\{x_b : b \in B_0^r\right\}\right)^{j_{b=x}^n}$ in Eq. (11), and simple and/or tight upper and lower bounds of the support weight distribution of regular LDPC code ensembles which are suitable to imply the mutuality between code-length and performance will be derived from the expression in Theorem 3. Maybe, application of this work to irregular LDPC code ensembles is easy, if those problems are overcome. In addition, this work may contribute to development of analysis of finite-length LDPC codes, which is an important but difficult topic. Our future works are to give the exact characterization of the average block erasure probability of regular LDPC code ensembles under MAP decoding over BEC and to analyze the asymptotic average support weight distribution of regular LDPC code ensembles.

References


Appendix A

In this appendix, we derive the polynomial $F_r\left(\left\{x_b : b \in B_0^r\right\}\right)$. Let $b_p^{(p)} \in \{0, 1\}$ be the binary vector whose $p$-th component is 1 and the other components are 0, namely

$$
b_p^{(p)} := (b_s)_{s=1}^r, \text{ where } b_s = \begin{cases} 1 & s = p, \\ 0 & s \neq p. \end{cases}
$$

Whether $k_{b_p^{(p)}}$ (the number of sockets which are connected to sockets on variable nodes whose corresponding components are 1 only in $x_p$) is even or odd depends on whether

$$
K_p := \sum_{b \in B_0^r} k_b \prod_{b \in B_0^r} j=1 \frac{1}{b_b^{(p)}},
$$

(A.1)
is even or odd since Eq. (15) is satisfied. Let \( \mathcal{B} \) be the set of all even non-negative integers and \( \mathcal{B}' := \mathcal{B}_0' \setminus \left( \bigcup_{\mathcal{P}=1}^{r} \{ b^{(\mathcal{P})} \} \right) \).

The probability of erasure, \( F_r \left( \{ x_b : b \in \mathcal{B}'_0 \} \right) \), can be written as follows.

\[
F_r \left( \{ x_b : b \in \mathcal{B}'_0 \} \right) = \sum_{k_b \in \mathcal{B}'_0} \frac{d!}{\prod_{b \in \mathcal{B}'_0} b!} \prod_{b \in \mathcal{B}'_0} x_b^{k_b} = \sum_{k_b \in \mathcal{B}'_0} \frac{d!}{\prod_{b \in \mathcal{B}'_0} b!} \prod_{b \in \mathcal{B}'_0} x_b^{k_b} = \sum_{k_b \in \mathcal{B}'_0} \frac{d!}{\prod_{b \in \mathcal{B}'_0} b!} \prod_{b \in \mathcal{B}'_0} x_b^{k_b}
\]

where the first equality follows from Eq. (A.1). Then, we have

\[
F_r \left( \{ x_b : b \in \mathcal{B}'_0 \} \right) = \frac{1}{2^r} \sum_{\mathcal{P}=1}^{r} \sum_{k_{\mathcal{P}} \in \mathcal{B}'_0} D! \prod_{b \in \mathcal{B}'_0} (k_b!) \cdot (-1)^{\sum_{\mathcal{P}=1}^{r} a_{\mathcal{P}} K_{\mathcal{P}}} \cdot \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \frac{1}{2^r} \sum_{\mathcal{P}=1}^{r} \sum_{k_{\mathcal{P}} \in \mathcal{B}'_0} d! \cdot \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D \cdot \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \frac{1}{2^r} \sum_{\mathcal{P}=1}^{r} \sum_{k_{\mathcal{P}} \in \mathcal{B}'_0} d! \cdot \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D \cdot \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D
\]

where we define \( D := d - \sum_{b \in \mathcal{B}'_0} k_b \). When we transform Eq. (A.2) into Eq. (A.3), we use the following lemma.

**Lemma 2:** For given \( K_1, K_2, \ldots, K_r \), and \( D \), the following equation holds:

\[
\sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D
\]

A proof of Lemma 2 is provided in Appendix B. The exponent of \( -1 \) in Eq. (A.3) can be rewritten as follows.

\[
\sum_{p=1}^{r} a_p K_p = \sum_{p=1}^{r} a_p \sum_{b \in \mathcal{B}'_0} k_b = \sum_{p=1}^{r} a_p k_b
\]

\[
\sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D
\]

**Appendix B**

In this Appendix, we prove Lemma 2 by mathematical induction on \( r \).

For \( r = 1 \), the following equation holds.

\[
\sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D = \sum_{k_{\mathcal{P}}} \left( \prod_{b \in \mathcal{B}'_0} x_b^{k_b} \right)^D
\]

Suppose the lemma is true for \( r = R \). Then, we have
\[
\sum_{k_{j/p}^{(p)}} \frac{D!}{(D - \sum_{p=1}^{R+1} k_{j/p}^{(p)})! \prod_{p=1}^{R+1} (k_{j/p}^{(p)}!)} \prod_{p=1}^{R+1} X_{j/p}^{k_{j/p}^{(p)}}
\]

where the second equality follows from the induction hypothesis, and the proof follows by mathematical induction.

\[\square\]

**Takayuki Itsui** received B.E. and M.E. degrees from Tokyo Institute of Technology in 2005 and 2007, respectively. Since 2007, he has been an engineer in Information Technology R&D Center, Mitsubishi Electric Corporation.

**Kenta Kasai** received B.E., M.E. and Ph.D. degrees from Tokyo Institute of Technology in 2001, 2003 and 2006, respectively. Since April 2006, he has been an associate professor in the Department of Communications and Integrated Systems, Graduate School of Science and Engineering, Tokyo Institute of Technology. His current research interests are error correcting coding and communication theory.

**Ryoji Ikegaya** received B.E., M.E. and Ph.D. degrees from Tokyo Institute of Technology in 2002, 2004 and 2006, respectively. Since October 2006, he has been with R&D Division, Semiconductor Business Group, Sony Corporation. His current research interests are error correcting coding and communication theory.

**Tomoharu Shibuya** received B.E., M.E., and Ph.D. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1992, 1994 and 1999, respectively. He joined Tokyo Institute of Technology in 1994 as a research associate in the Department of Electrical and Electronic Engineering, and served as a research associate in the Department of Communications and Integrated Systems from 1999 to 2003. Since 2003, he has been an associate professor in R&D Department, National Institute of Multimedia Education. He received the Young Engineer Award from the IEICE of Japan in 1997. His research interests are error control coding, information theory and communication theory. He is a member of IEEE, SITA, and IPSJ.
Kohichi Sakaniwa received B.E., M.E., and Ph.D. degrees all in electronic engineering from the Tokyo Institute of Technology, Tokyo Japan, in 1972, 1974 and 1977, respectively. He joined the Tokyo Institute of Technology in 1977 as a research associate and served as an associate professor from 1983 to 1991. Since 1991 he has been a professor in the Department of Electrical and Electronic Engineering, and since 2000 in the Department of Communication and Integrated Systems, Graduate School of Science and Engineering, both in the Tokyo Inst. of Tech. From November 1987 to July 1988, he stayed at the University of Southwestern Louisiana as a Visiting Professor. He received the Excellent Paper Award from the IEICE of Japan in 1982, 1990, 1992 and 1994. His research area includes Communication Theory, Error Correcting Coding, (Adaptive) Digital Signal Processing and so on. Dr. Sakaniwa is a member of IEEE, Information Processing Society of Japan, Institute of Image Information and Television Engineers of Japan, and Society of Information Theory and its Applications.