1

Sum-Product Decoding of Punctured Convolutional Code for Wireless LAN

Toshiyuki Shohon
Kagawa National College of Technology
Japan

1. Introduction

The next generation wireless Local Area Network (LAN) standard (IEEE802.11n) aims for high rate data transmission such as 100Mbps to 600Mbps. In order to implement that rate, high speed decoder for the convolutional code for the wireless LAN standard is necessary. From the viewpoint of high speed decoder, sum-product algorithm is an attractive decoding method, since decoding rule of sum-product algorithm is simple and sum-product algorithm is suit for parallel implementation. Furthermore, sum-product decoding is a soft-in soft-out decoding. The combined use of sum-product algorithm and another soft-in soft-out processing may provide good performance such as turbo equalization (Douillard et al., 1995; Laot et al., 2001). However, sum-product decoding for the convolutional code of the wireless LAN can not provide good performance. To improve the performance, the sum-product decoding method for the non-punctured convolutional code of the wireless LAN has been proposed (Shohon et al., 2009b; 2010). In the wireless LAN, however, punctured convolutional codes are also used. Therefore, this paper proposes sum-product decoding methods for the punctured convolutional codes of the wireless LAN.

A sum-product decoding method for convolutional codes has been introduced in (Kschischang et al., 2001). The sum-product algorithm uses a Wiberg-type graph that represents a code trellis with hidden variables as code states and visible variables as code bits. In this case, the Wiberg-type graph is equivalent to the code trellis and the sum-product algorithm becomes precisely identical to BCJR algorithm (Berrou, C. et al.;C; Kschischang et al., 2001). This method only gives interpretation of BCJR algorithm as sum-product algorithm. To avoid confusion, the method of (Kschischang et al., 2001) is referred to as BCJR. This paper deals with sum-product algorithm that uses a Tanner graph that represents a parity check matrix of the code. This sum-product algorithm is the same as that for Low-Density Parity-Check code (Gallager, 1963; MacKay, 1999). The sum-product decoding method for recursive systematic convolutional codes has been proposed in (Shohon et al., 2009a). In the wireless LAN, the non-systematic convolutional code is used. For the non-punctured convolutional code of the wireless LAN, the sum-product decoding method has been proposed in (Shohon et al., 2009b; 2010). In this paper, for punctured codes of the wireless LAN, sum-product decoding methods are proposed.

This paper is constructed as follows. In section 2, the convolutional codes used in the wireless LAN are explained. In section 3, the sum-product algorithm for convolutional codes is explained. In section 4, the sum-product decoding method for non-punctured convolutional code of the wireless LAN is explained and decoding performance of that

method for punctured codes are shown. In section 5 and section 6, the sum-product decoding methods for punctured codes of the wireless LAN are proposed. In section 7, the decoding complexity is discussed.

2. Convolutional code for wireless LAN

2.1 Non-punctured code

The convolutional code for the wireless LAN is a non-systematic code with rate 1/2 (IEEE Std 802.11, 2007). Let a sequence of information bits be denoted by $x_0, x_1, \cdots, x_{N-1}$, a sequence of parity bits 1 be denoted by $p_{1,0}, p_{1,1}, \cdots, p_{1,N-1}$, and a sequence of parity bits 2 be denoted by $p_{2,0}, p_{2,1}, \cdots, p_{2,N-1}$. Polynomial representation for each sequence is as follows.

$$X(D) = x_0 + x_1 D + x_2 D^2 + \cdots + x_{N-1} D^{N-1} \tag{1}$$

$$P_1(D) = p_{1,0} + p_{1,1} D + p_{1,2} D^2 + \cdots + p_{1,N-1} D^{N-1} \tag{2}$$

$$P_2(D) = p_{2,0} + p_{2,1} D + p_{2,2} D^2 + \cdots + p_{2,N-1} D^{N-1} \tag{3}$$

Parity bit polynomials are given by

$$P_1(D) = G_1(D) X(D), \tag{4}$$

$$P_2(D) = G_2(D) X(D). \tag{5}$$

For the wireless LAN standard, $G_1(D)$ and $G_2(D)$ are given by

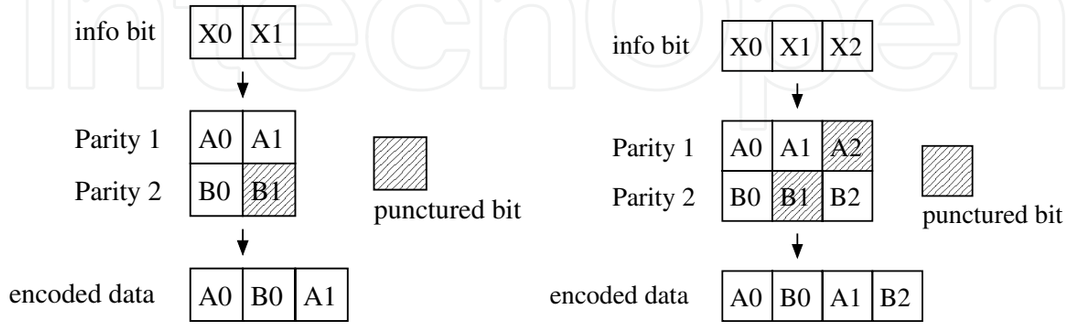
$$G_1(D) = 1 + D^2 + D^3 + D^5 + D^6, \tag{6}$$

$$G_2(D) = 1 + D + D^2 + D^3 + D^6. \tag{7}$$

Polynomials $X(D), P_1(D), P_2(D)$ are also represented by X, P_1, P_2 in this paper.

2.2 Punctured code

In this section, puncturing method for wireless LAN will be explained. Puncturing is a procedure for omitting some of the encoded bits in the transmitter. The effect from puncturing will reducing the number of transmitted bits and increasing the coding rate. Figure 1(a) to Fig.1(b) shows the puncturing pattern for coding rate, $r = 2/3, 3/4$.



(a) Puncturing pattern for code rate 2/3 (b) Puncturing pattern for code rate 3/4

Fig. 1. Puncturing pattern

3. Sum-product algorithm for convolutional codes

Sum-product algorithm is a message exchanging algorithm along with edge of the Tanner graph of the code. Tanner graph is a bipartite graph that represents the parity check matrix of the code. For convolutional code, it is easy to obtain tanner graph from parity check polynomial. This section explains parity check polynomial for convolutional codes, tanner graph and sum-product algorithm.

3.1 Parity check polynomial of convolutional code for wireless LAN

From Equation 4 $\sim$ Equation 5, we can obtain following equations.

$$G_1(D)X + P_1 = 0 \tag{8}$$
$$G_2(D)X + P_2 = 0 \tag{9}$$

Let left parts of Equation 8 and Equation 9 be defined as parity check polynomial.

$$H_{org,1}(X, P_1) = G_1(D)X + P_1 \tag{10}$$
$$H_{org,2}(X, P_2) = G_2(D)X + P_2 \tag{11}$$

A tuple of polynomials (X, P_1, P_2) is a code word if following equations are satisfied.

$$H_{org,1}(X, P_1) = 0 \tag{12}$$
$$H_{org,2}(X, P_2) = 0 \tag{13}$$

The degree of a parity check polynomial is denoted by ν, that is the maximum degree of coefficients of the polynomial. For example, since coefficients of $H_{org,1}(X, P_1)$ are $\{G_1(D), 1\}$, the maximum degree is $\nu = 6$ that is the maximum degree of $G_1(D)$.

3.2 Tanner graph of convolutional code

From Equation 12, parity check equations at k and $k+1$ time slots are given by

$$C_k \ : \ x_{k-6} + x_{k-5} + x_{k-3} + x_{k-2} + x_k + p_{1,k} = 0, \tag{14}$$
$$C_{k+1} \ : \ x_{k-5} + x_{k-4} + x_{k-2} + x_{k-1} + x_{k+1} + p_{1,k+1} = 0. \tag{15}$$

Those equations are corresponding to check nodes C_k and C_{k+1}, of the tanner graph. The part of tanner graph corresponding to those parity check equations is as shown in Fig.2.

3.3 Algorithm

For convenience, bit node is denoted by u_n such that

$$\begin{cases} u_{3n} = x_n \\ u_{3n+1} = p_{1,n} \\ u_{3n+2} = p_{2,n} \end{cases} \tag{16}$$

where information bit is x_n and parity bits are $p_{1,n}$, $p_{2,n}$. Message from bit node, u_n, to check node C_m, is denoted by $V_{m,n}$. Message from check node, C_m, to bit node u_n, is denoted by $U_{m,n}$. Sum-Product algorithm is described as follows.

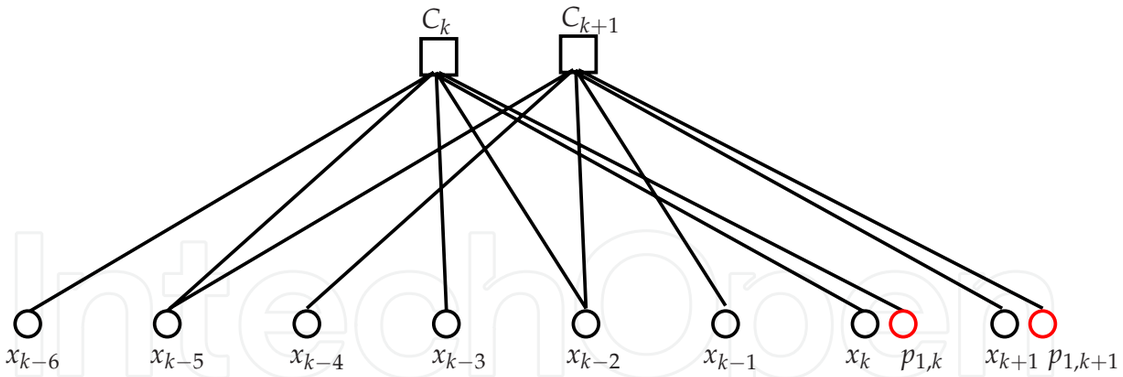


Fig. 2. Part of Tanner graph

Step1. Initialization

Each message $V_{m,n}$ is set to the initial value as follows.

$$V_{m,n} = \lambda_n = \frac{2r_n}{\sigma^2} \tag{17}$$

where, r_n denotes received signal, σ^2 denotes variance of additive white Gaussian noise and λ_n is channel value.

Step2. Message from check node to bit node

Each check node C_m updates the message on bit node u_n by gathering all incoming messages from other bit nodes that connected to check node C_m. Message $U_{m,n}$ is calculated by following equation (Gallager, 1963; Hagenauer, 1996; Richardson et al., 2001).

$$U_{m,n} = 2f_s \tanh^{-1}\left\{\prod_{n' \in \mathcal{N}(m)\backslash n} \tanh\left(\frac{V_{m,n'}}{2}\right)\right\} \tag{18}$$

where, $\mathcal{N}(m)$ denotes the set of bit node numbers that connect to the check node C_m and f_s is a scaling factor. This factor is used in the proposed method described later. When f_s is not specified, $f_s = 1$.

Step3. Message from bit node to check node

Each bit node n propagates its message to all check nodes that connect to it.

$$V_{m,n} = \lambda_n + \sum_{m' \in \mathcal{M}(n)\backslash m} U_{m',n} \tag{19}$$

where $\mathcal{M}(n)$ denotes the set of check node numbers that connect to the bit node, u_n.

Step4. Tentative estimated code word computation

By summing up all the messages from all check nodes connected to a bit node, the a posteriori value Λ_n can be obtained by

$$\Lambda_n = \lambda_n + \sum_{m \in \mathcal{M}(n)} U_{m,n}. \tag{20}$$

The extrinsic value, $L_e(u_n)$, of bit node u_n can be obtained by

$$L_e(u_n) = \sum_{m \in \mathcal{M}(n)} U_{m,n}. \tag{21}$$

The tentative estimated bit u'_n can be obtained by

$$u'_n = \begin{cases} 0 & if \quad \text{sign}(\Lambda_n) = +1 \\ 1 & if \quad \text{sign}(\Lambda_n) = -1 \end{cases} \tag{22}$$

Step5. Stop criterion

Tentative estimated code word $\mathbf{u}'$ obtained in Step 4 is checked against the parity check matrix H. If H multiplied by Tentative estimated code word $\mathbf{u}'^T$ equal to zero vector, the decoder stop and outputs $\mathbf{u}'$, if not, it repeats Steps 2-5.

$$H\mathbf{u}'^T = \mathbf{0} \tag{23}$$

If maximum iteration number of decoding is set, the tentative estimated code word $\mathbf{u}'$ outputs after decoding procedure repeat the process until the maximum iteration is reached.

4. Sum-product decoding for wireless LAN (conventional method)

This section will give summary of (Shohon et al., 2009b; 2010). Sum-product decoding can be performed by using Equation 10 and Equation 11 as parity check polynomials. However, the decoding provides bad performance. Since the code under consideration is a non-systematic code, there are no received signals corresponding to information bits and channel values for information bits are zero. It can be seen from Equation 10, Equation 11 that each check node has more than one information bit connections. Therefore reliability increment at check node cannot be obtained. Consequently, conventional sum-product algorithm cannot realize good performance. To improve the sum-product decoding performance, I have proposed the 2-step decoding method (Shohon et al., 2009b; 2010).

4.1 2-Step decoding

The 2-step decoding method is as follows. (1) Only parity bits are decoded by sum-product algorithm. (2) With decoded parity bits, information bits are regenerated.

4.1.1 Decoding parity bits

The parity check equation is derived from Equation 4 $\sim$ Equation 5 as follows.

$$G_2(D)P_1(D) + G_1(D)P_2(D) = 0 \tag{24}$$

The left part of the equation is defined as parity check polynomial $H(P_1, P_2)$.

$$H(P_1, P_2) = G_2(D)P_1 + G_1(D)P_2 \tag{25}$$

Parity bits P_1 and P_2 can be decoded by sum-product algorithm based on parity check polynomial given by Equation 25. By using the decoded parity bits, information bits can be regenerated.

4.1.2 Decoding information bits

Decoded information bit $\hat{X}$ can be obtained by Equation 26 with decoded parity bits $\hat{P}_1, \hat{P}_2$.

$$\hat{X} = G_{x,1}(D)\hat{P}_1 + G_{x,2}(D)\hat{P}_2 \tag{26}$$

where,

$$G_{x,1}(D) = D^4 + D^2 \tag{27}$$

$$G_{x,2}(D) = D^4 + D^3 + D^2 + D + 1 \tag{28}$$

From Equation 26, Equation 27, and Equation 28, it can be seen that information bit can be regenerated by using a non-recursive convolutional encoder with input $\hat{P}_1$, $\hat{P}_2$ and output $\hat{X}$ as shown in Fig.3.

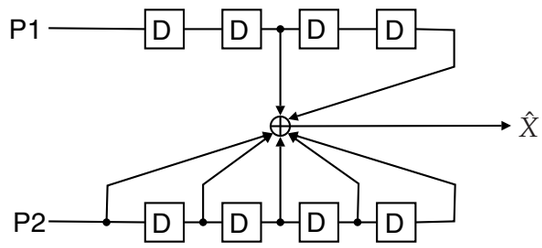


Fig. 3. Information bits regenerator

4.2 Higher degree parity check polynomial

I have proposed to use higher degree parity check polynomial to obtain further performance improvement (Shohon et al., 2009b; 2010).

The method is a sum-product decoding with higher degree parity check polynomial than that of the original parity check polynomial. In this section, the method is applied to improve the sum-product decoding performance for parity bits. The higher degree parity check polynomial is denoted by $H'(P_1, P_2)$, that is given by

$$H'(P_1, P_2) = M(D)H(P_1, P_2) \tag{29}$$

$$= M(D)G_2(D)P_1 + M(D)G_1(D)P_2 \tag{30}$$

$$= G'_2(D)P_1 + G'_1(D)P_2 \tag{31}$$

where $M(D)$ is a non-zero polynomial. Among possible higher degree parity check polynomials, we aim to select the optimum higher degree parity check polynomial by experiments and to use it for sum-product decoding. However, the number of prospective objects becomes too much when we include all possible higher degree parity check polynomials in the experimental objects. Therefore, we limit the range of degree of higher degree parity check polynomials ($\nu \leq 16$). For those higher degree parity check polynomials, we further limit the prospective objects by using n_{fc}, that is the number of four-cycles per one check node (Shohon et al., 2009a). For every degree of higher degree parity check polynomial, we select the higher degree parity check polynomial that has the minimum n_{fc} among higher degree parity check polynomials of object degree and include it in the experimental objects. By this means, Table 1 was obtained.

ν	n_{fc}	$G_2'(oct)$	$G_1'(oct)$
6	29	117	155
7	24	321	267
8	52	563	731
9	17	1067	1405
10	36	3131	2417
11	11	4015	6243
12	28	13103	16111
13	13	21003	30611
14	22	45203	65011
15	17	100001	145207
16	25	221001	322207

Table 1. Examined higher degree parity check polynomials for code rate 1/2

Experimental result shows that higher degree parity check polynomial of degree $\nu = 13$ provides the best performance. The higher degree parity check polynomial is given by

$$H'(P_1, P_2) = G_2'(D)P_1 + G_1'(D)P_2 \tag{32}$$

$$G_1'(D) = 1 + D^3 + D^7 + D^8 + D^{12} + D^{13} \tag{33}$$

$$G_2'(D) = 1 + D + D^9 + D^{13} \tag{34}$$

4.3 Simulation results for non-punctured code

Simulation condition is shown in Table 2. Hereafter, this condition was used, if simulation condition is not specified. Figure 4 shows simulation results. The figure shows that the performance for information bits of 2-Step Decoding with higher degree parity check polynomial (denoted by conventional) is only 0.7[dB] inferior to that of BCJR at bit error rate 10^{-5}.

Number of info bits per block	1024[bit]
Termination	Zero-termination
Channel	Additive white Gaussian noise
Maximum iterations	200

Table 2. Simulation condition

4.4 Simulation results for punctured codes

For non-punctured code, higher degree parity check polynomial with degree $\nu = 13$ provides the best performance. With that higher degree parity check polynomial, for punctured codes with code rates 2/3 and 3/4, the sum-product decoding simulation were executed. The simulation results are shown in Fig.5 and Fig.6.

From Fig.5 and Fig.6, it can be seen that the conventional method, that is sum-product decoding with higher degree parity check polynomial with $\nu = 13$, can not provide good performance for punctured code with code rates 2/3 and 3/4.

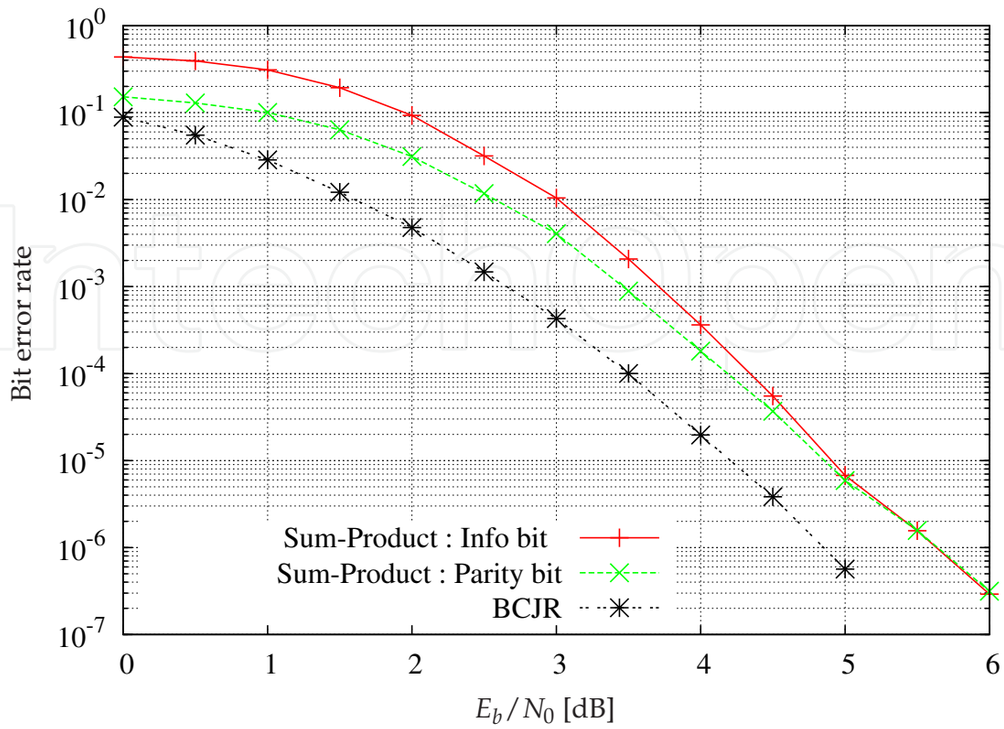


Fig. 4. Bit error rate performance of conventional method for code rate 1/2

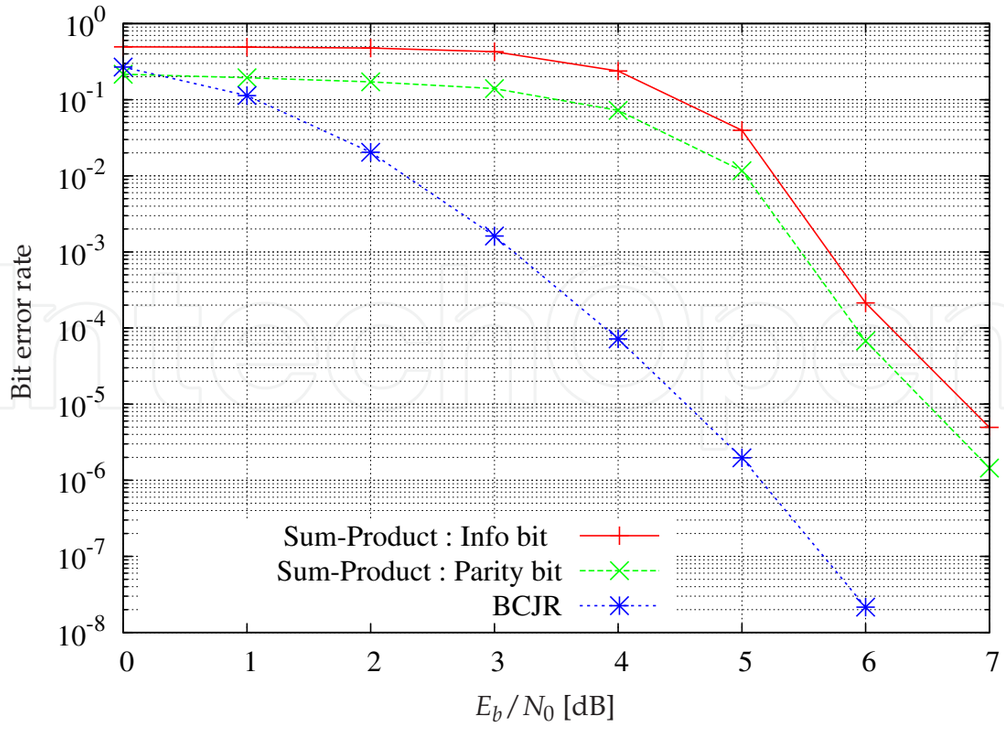


Fig. 5. Bit error rate performance of conventional method for code rate 2/3

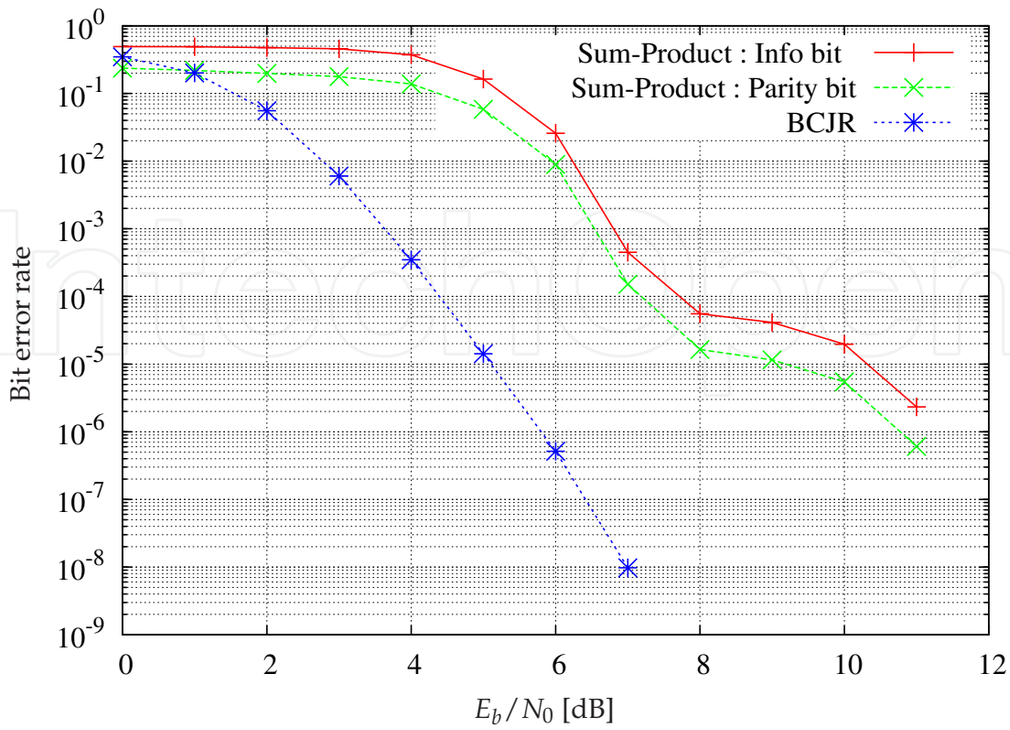


Fig. 6. Bit error rate performance of conventional method for code rate 3/4

5. Single punctured bit method (Proposed decoding method (1))

I inferred that the bad sum-product decoding performance for punctured codes is caused by more than one punctured bits included in the parity check equation at time slot k. The reason is as follows. Since received signals are not available for punctured bits, the channel values for punctured bits are zero. This causes that every messages from punctured bit node to check node are zero. In this case, like stopping set (Di et al., 2002), messages from the check node to bit nodes are zero. Therefore, sum-product algorithm does not work.

In order to improve the sum-product decoding performance, this paper proposes to use parity check equation that includes single punctured bit. The condition to include single punctured bit in parity check equation is referred to as single punctured bit condition. If single punctured bit is included in a parity check equation at time slot k, the message to the corresponding bit node can be obtained from the corresponding check node C_k. In this case, sum-product algorithm can work. Therefore, we expect that using higher degree parity check polynomial such that parity check equation includes single punctured bit, brings performance improvement of sum-product decoding of punctured codes.

5.1 Higher degree parity check polynomial satisfying single puncture bit condition

In this section, single punctured bit condition is derived for higher degree parity check polynomial. A higher degree parity check equation is given by

$$H'(P_1, P_2) = G'_2(D)P_1 + G'_1(D)P_2 \tag{35}$$

Generally, polynomials $G_1'(D)$ and $G_2'(D)$ are given by

$$G_1'(D) = \sum_{i=1}^{d_1} D^{\alpha_i} \tag{36}$$

$$G_2'(D) = \sum_{i=1}^{d_2} D^{\beta_i} \tag{37}$$

If (P_1, P_2) is code word, it satisfies

$$G_2'(D)P_1 + G_1'(D)P_2 = 0 \tag{38}$$

From Equation 36, Equation 37 and Equation 38, parity check equation at time slot k is represented by

$$\sum_{i=1}^{d_1} p_{1,k-\beta_i} + \sum_{i=1}^{d_2} p_{2,k-\alpha_i} = 0 \tag{39}$$

5.1.1 Code rate 2/3

For code rate $2/3$, punctured bits are

$$\{\, p_{2,2n+1} \mid n = 0, 1, 2, \cdots \} \tag{40}$$

From Equation 39 and Equation 40, it can be seen that punctured bits included in parity check equation at time slot k satisfies

$$p_{2,k-\alpha_i} = p_{2,2n+1} \tag{41}$$

Therefore, we obtain

$$k - \alpha_i = 2n + 1 \tag{42}$$

$$\alpha_i = k - (2n + 1). \tag{43}$$

For time slot $k = 2l,\ l = 0, 1, 2, \cdots,$

$$\alpha_i = 2l - (2n + 1) \tag{44}$$

$$= 2(l - n) - 1 \tag{45}$$

Therefore, the set $\{\alpha_i \mid (\alpha_i \bmod 2) = 1\}$ in higher degree parity check polynomial corresponds to punctured bits in the parity check equation at time slot $k = 2l,\ l = 0, 1, 2, \cdots$. If Equation 46 is satisfied, the higher degree parity check polynomial satisfies single punctured bit condition at time slot $k = 2l,\ l = 0, 1, 2, \cdots$.

$$\#\{\alpha_i \mid (\alpha_i \bmod 2) = 1\} = 1 \tag{46}$$

where $\#\{x\}$ denotes the number of elements in the set $\{x\}$.

Similarly, if Equation 47 is satisfied, the higher degree parity check polynomial satisfies single punctured bit condition at time slot $k = 2l + 1,\ l = 0, 1, 2, \cdots$.

$$\#\{\alpha_i \mid (\alpha_i \bmod 2) = 0\} = 1 \tag{47}$$

Therefore, if either Equation 46 or Equation 47 is satisfied, the higher degree parity check polynomial satisfies single punctured bit condition.

5.1.2 Code rate 3/4

For code rate 3/4, punctured bits are

$$\begin{cases} p_{1,3n+2} & n = 0, 1, 2, \cdots \\ p_{2,3n+1} & n = 0, 1, 2, \cdots \end{cases}. \tag{48}$$

From Equation 39 and Equation 48, it can be seen that punctured bits included in parity check equation at time slot k satisfies

$$\begin{cases} p_{1,k-\beta_i} = p_{1,3n+2} & n = 0, 1, 2, \cdots \\ p_{2,k-\alpha_i} = p_{2,3n+1} & n = 0, 1, 2, \cdots \end{cases}. \tag{49}$$

Therefore, we obtain

$$\begin{cases} k - \beta_i = 3n + 2 \\ k - \alpha_i = 3n + 1 \end{cases}. \tag{50}$$

For time slot $k = 3l,\ l = 0, 1, 2, \cdots,$

$$3l - \beta_i = 3n + 2 \tag{51}$$

$$\beta_i = 3(l - n) - 2 \tag{52}$$

$$3l - \alpha_i = 3n + 1 \tag{53}$$

$$\alpha_i = 3(l - n) - 1 \tag{54}$$

From Equation 52, it can be seen that the set $\{\beta_i \mid (\beta_i \bmod 3) = 1\}$ in higher degree parity check polynomial corresponds to punctured bits of parity bit P_1 in the parity check equation at time slot $k = 3l,\ l = 0, 1, 2, \cdots$. From Equation 54, it can be seen that the set $\{\alpha_i \mid (\alpha_i \bmod 3) = 2\}$ in higher degree parity check polynomial correspond to punctured bits of the parity bit P_2 in the parity check equation at time slot $k = 3l,\ l = 0, 1, 2, \cdots$.

Therefore, if either Equation 55 or Equation 56 is satisfied, the higher degree parity check polynomial satisfies single punctured bit condition at time slot $k = 3l,\ l = 0, 1, 2, \cdots$.

$$\begin{aligned} &(\#\{\beta_i \mid (\beta_i \bmod 3) = 1\} = 1) \\ &\quad \wedge (\#\{\alpha_i \mid (\alpha_i \bmod 3) = 2\} = 0) \end{aligned} \tag{55}$$

$$\begin{aligned} &(\#\{\beta_i \mid (\beta_i \bmod 3) = 1\} = 0) \\ &\quad \wedge (\#\{\alpha_i \mid (\alpha_i \bmod 3) = 2\} = 1) \end{aligned} \tag{56}$$

Similarly, if either Equation 57 or Equation 58 is satisfied, the higher degree parity check polynomial satisfies single punctured bit condition at time slot $k = 3l + 1,\ (l = 0, 1, 2, \cdots)$.

$$\begin{aligned} &(\#\{\beta_i \mid (\beta_i \bmod 3) = 2\} = 1) \\ &\quad \wedge (\#\{\alpha_i \mid (\alpha_i \bmod 3) = 0\} = 0) \end{aligned} \tag{57}$$

$$(\#\{\beta_i \mid (\beta_i \bmod 3) = 2\} = 0) \\ \wedge(\#\{\alpha_i \mid (\alpha_i \bmod 3) = 0\} = 1) \quad (58)$$

Similarly, if either Equation 59 or Equation 60 is satisfied, the higher degree parity check polynomial satisfies single punctured bit condition at time slot $k = 3l + 2$, $(l = 0, 1, 2, \cdots)$.

$$(\#\{\beta_i \mid (\beta_i \bmod 3) = 0\} = 1) \\ \wedge(\#\{\alpha_i \mid (\alpha_i \bmod 3) = 1\} = 0) \quad (59)$$

$$(\#\{\beta_i \mid (\beta_i \bmod 3) = 0\} = 0) \\ \wedge(\#\{\alpha_i \mid (\alpha_i \bmod 3) = 1\} = 1) \quad (60)$$

5.2 Search of higher degree parity check polynomial for decoding

In this paper, basically, the higher degree parity check polynomials for decoding are searched as follows.

Step.1 Select higher degree parity check polynomials with degree $\nu \leq 21$ that satisfies single punctured bit condition.

Step.2 Among those higher degree parity check polynomials, select the higher degree parity check polynomial that provides the best sum-product decoding performance by using computer simulation.

5.2.1 Code rate 2/3

In the Step.1, 208 higher degree parity check polynomials satisfy single punctured bit condition. Since many higher degree parity check polynomials are selected, they are limited by n_{fc}. In this paper, among those higher degree parity check polynomials, 9 higher degree parity check polynomials with lower n_{fc} are selected. They are shown in Table 3.

No.	ν	n_{fc}	$G_2'(oct)$	$G_1'(oct)$
1	8	29	755	403
2	9	17	1067	1405
3	11	38	6143	5251
4	14	26	62501	50107
5	16	26	364203	202011
6	16	33	203133	310001
7	16	43	310207	243025
8	17	16	624403	500211
9	17	42	445207	640025

Table 3. Examined higher degree parity check polynomials for code rate 2/3

The simulation results of Step.2 with higher degree parity check polynomials in Table 3 are shown in Fig. 7. Simulation condition is shown in Table 2 and $E_b/N_0 = 5.0$ [dB]. From Fig. 7, it can be seen that higher degree parity check polynomial of No.5 with scaling factor $f_s = 0.9$ provides the best performance.

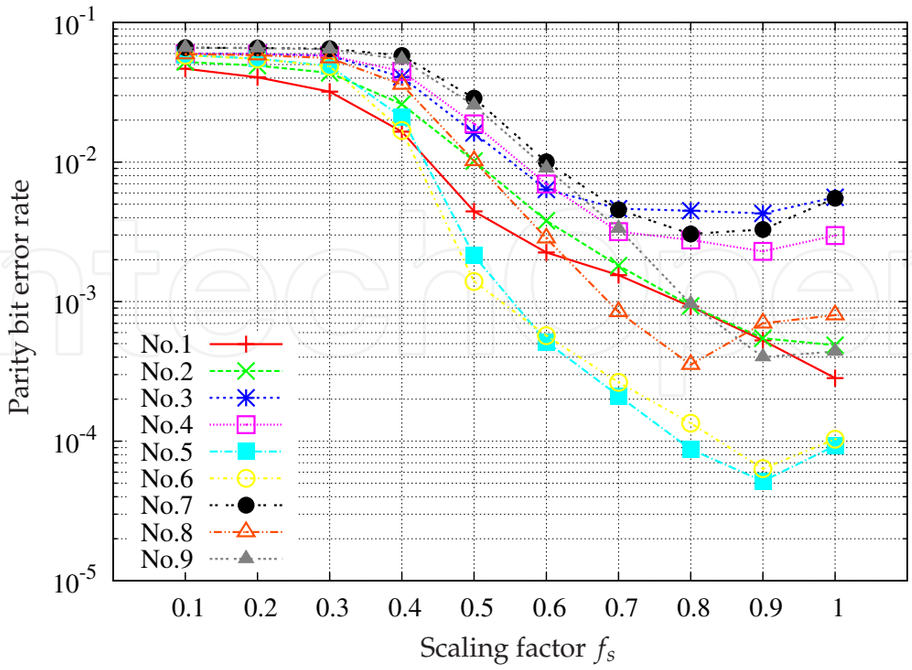


Fig. 7. Simulation results of Step.2 for code rate 2/3 at E_b/N_0 =5[dB]

5.2.2 Code rate 3/4

5.2.2.1 Step.1

For code rate 3/4, there are both puncture bits of parity P_1 and parity P_2. Decodable punctured parity bit by sum-product algorithm with certain parity check equation is either parity P_1 or parity bit P_2. From the viewpoint of decodable parity bit, single punctured bit condition can be arranged as follows.

1. If either Equation 55 or Equation 57, or Equation 59 is satisfied, the higher degree parity check polynomial includes single punctured bit of parity P_1. Therefore, with the higher degree parity check polynomial, punctured bits of parity P_1 can be decoded. That higher degree parity check polynomial is referred to as higher degree parity check polynomial for P_1.
2. If either Equation 56 or Equation 58 or Equation 60 is satisfied, the higher degree parity check polynomial includes single punctured bit of parity P_2. Therefore, with the higher degree parity check polynomial, punctured bits of parity P_2 can be decoded. That higher degree parity check polynomial is referred to as higher degree parity check polynomial for P_2.

Therefore, for code rate 3/4, both higher degree parity check polynomials for P_1 and P_2 are necessary to decode.

For code rate 3/4, there are 16 higher degree parity check polynomials for P_1 and 16 higher degree parity check polynomials for P_2. The number of combination of higher degree parity check polynomial for P_1 and that for P_2 is many. Therefore, they are limited by n_{fc}. Higher degree parity check polynomials that have lower n_{fc} are selected as shown in Table 4 and Table 5.

No.	ν	n_{fc}	$G_2'(oct)$	$G_1'(oct)$
1	12	30	14453	12121
2	21	64	17010055	10212103

Table 4. Examined higher degree parity check polynomials for P_1

No.	ν	n_{fc}	$G_2'(oct)$	$G_1'(oct)$
3	7	24	321	267
4	12	29	11055	15103
5	21	38	10540055	14222103

Table 5. Examined higher degree parity check polynomials for P_2

5.2.2.2 Step.2

A block diagram of the decoder for code rate 3/4 is shown in Fig. 8. It is similar to a turbo decoder. In Fig.8, DEC1 is sum-product algorithm decoder with higher degree parity check polynomial for P_1 and DEC2 is sum-product algorithm decoder with higher degree parity check polynomial for P_2. Channel value is denoted by λ_n. Extrinsic values of DEC1 and DEC2 are denoted by $L_{e1}(u_n)$ and $L_{e2}(u_n)$, respectively. A posteriori value of DEC2 is denoted by $\Lambda_{2,n}$.

In DEC1, $L_{e2}(u_n)$ is added to λ_n as follows.

$$\lambda_n' = \lambda_n + L_{e2}(u_n) \tag{61}$$

The value λ_n' is used as initial value of λ_n in Equation 17.

Similarly, in DEC2, $L_{e1}(u_n)$ is added to λ_n and that value is used as initial value of λ_n. In computer simulation, the number of iteration of sum-product algorithm at each decoder was set to 1. The maximum number of iteration between two decoders was set to 200. Other simulation conditions are the same as shown in Table 2.

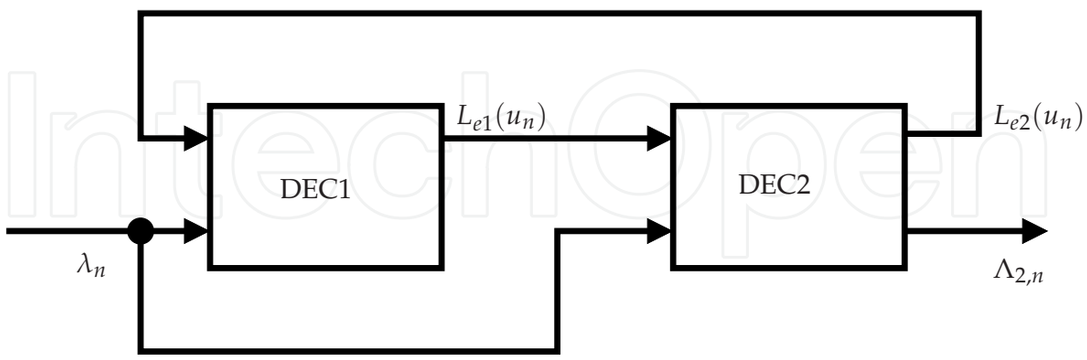


Fig. 8. Block diagram of decoder of single punctured bit method for code rate 3/4

Figure 9 shows the simulation results of Step.2 for code rate 3/4 at $E_b/N_0 = 6$[dB].

From Fig.9, it can be seen that the combination of higher degree parity check polynomials No.2 and No.3 with scaling factor $f_s = 0.5$ provides the best decoding performance.

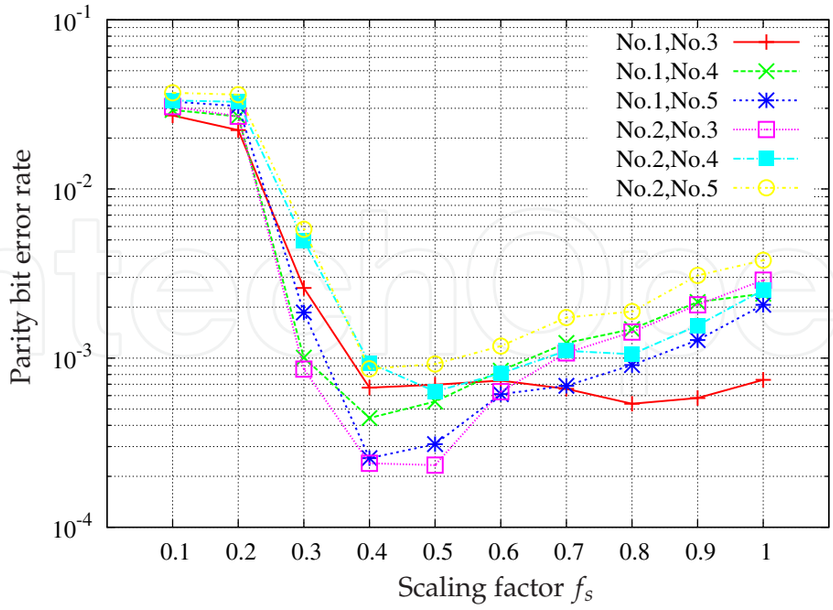


Fig. 9. Simulation results of Step.2 for code rate 3/4 at $E_b/N_0 = 6$[dB]

5.3 Simulation results

5.3.1 Code rate 2/3

Figure 10 shows bit error rate performance of the single punctured bit method for code rate 2/3.

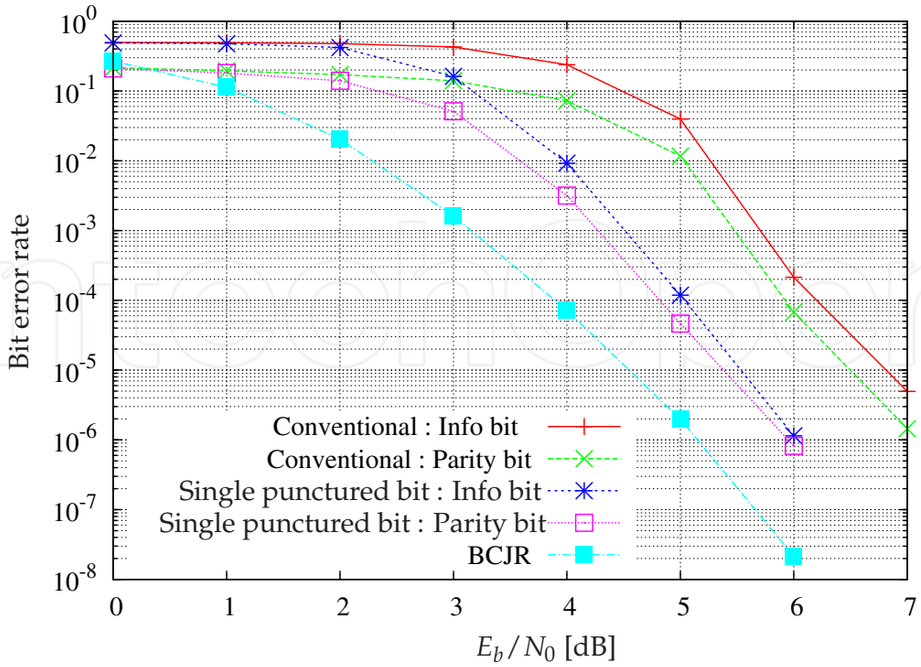


Fig. 10. BER performance of single punctured bit method for code rate 2/3

From Fig.10 it can be seen that the parity bit error rate performance of the single punctured bit method is 1.12[dB] superior to that of the conventional method (higher degree parity check polynomial of $\nu = 13$) at bit error rate 10^{-5}. The parity bit error rate performance of the single punctured bit method is only 0.83[dB] inferior to that of BCJR.

From Fig.10, information bit error performance of the single punctured bit method is 1.28[dB] superior to that of the conventional method at bit error rate 10^{-5}. The information bit error rate performance of the single punctured bit method is only 0.98 [dB] inferior to that of BCJR.

5.3.2 Code rate 3/4

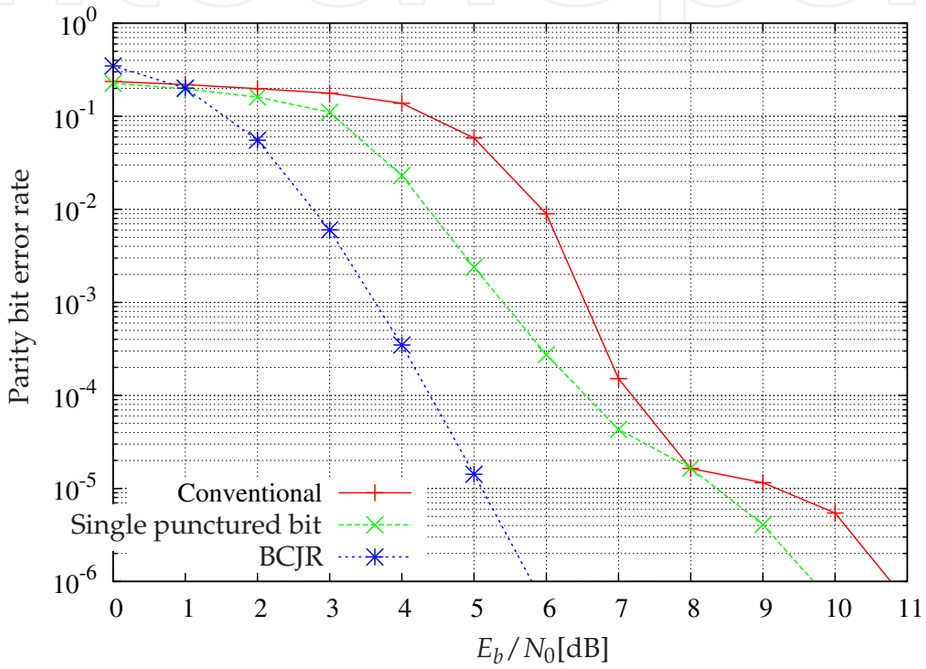


Fig. 11. Parity bit error rate performance of single punctured bit method for code rate 3/4

Figure 11 shows parity bit error rate performance of the single punctured bit method for code rate 3/4. From Fig.11 it can be seen that the parity bit error rate performance of the single punctured bit method is 0.82[dB] superior to that of the conventional method (higher degree parity check polynomial of $\nu = 13$) at bit error rate 10^{-5}. The parity bit error rate performance of the single punctured bit method is 3.24[dB] inferior to that of BCJR.

Figure 12 shows information bit error rate performance of the single punctured bit method. From Fig.12, it can be seen that the information bit error rate performance of the single punctured bit method is 1.11[dB] superior to the conventional method at bit error rate 10^{-5}. The information bit error rate performance of the single punctured bit method is 4.11[dB] inferior to that of BCJR at bit error rate 10^{-5}.

6. Switching parity check method (proposed decoding method (2))

For code rate 3/4, the proposed method (1) can not provide good performance. Therefore, this paper try to improve the sum-product decoding performance for code rate 3/4.

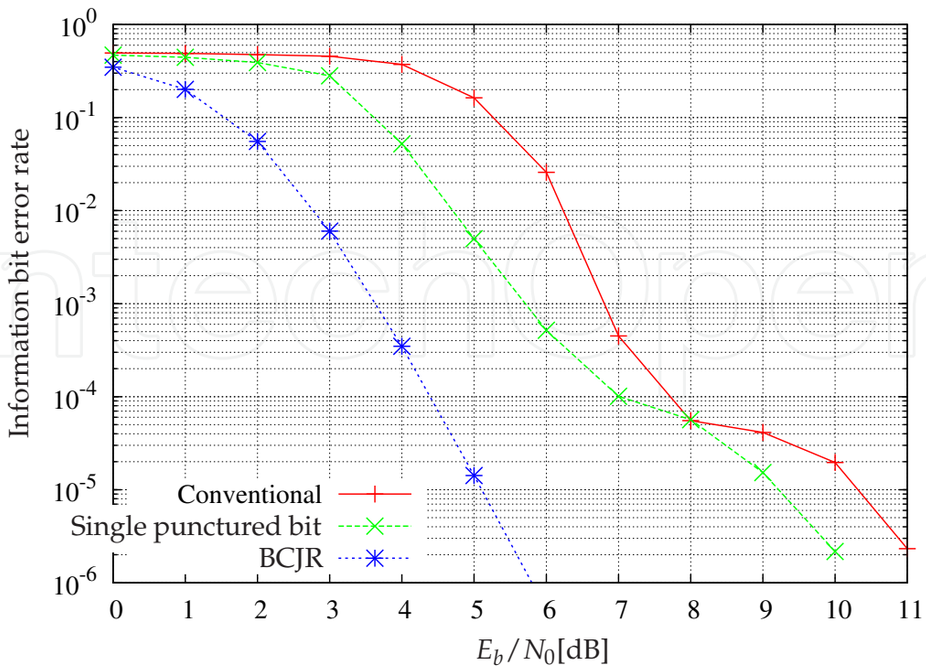


Fig. 12. Information bit error rate performance of single punctured bit method for code rate 3/4

I inferred that the bad decoding performance is caused by the four-cycles of higher degree parity check polynomial, since n_{fc} of higher degree parity check polynomial satisfying single punctured bit condition tends to be larger than n_{fc} of higher degree parity check polynomial that does not satisfy single punctured bit condition. Therefore, this paper proposes following method. Only at first iteration, the higher degree parity check polynomial satisfying single punctured bit condition is used to decode and after first iteration, another higher degree parity check polynomial without single punctured bit condition is used to decode. By decoding, only at first iteration, with higher degree parity check polynomial satisfying single punctured bit condition, the a posteriori values of punctured bits are obtained. After obtaining the a posteriori values of punctured bit, the higher degree parity check polynomial with lower n_{fc} may provide good bit error rate performance.

Figure 13 shows a block diagram of decoder of the switching parity check method. In Fig. 13, DEC1 is a sum-product algorithm decoder with higher degree parity check polynomial for P_1, DEC2 is a sum-product algorithm decoder with higher degree parity check polynomial for P_2 and DEC3 is a sum-product algorithm decoder with higher degree parity check polynomial with lower n_{fc} for iteration. Chanel values for DEC1, DEC2 and DEC3 are $\lambda_{1,n}$, $\lambda_{2,n}$ and $\lambda_{3,n}$, respectively. A posteriori values of DEC1, DEC2 and DEC3 are $\Lambda_{1,n}$, $\Lambda_{2,n}$ and $\Lambda_{3,n}$, respectively. Decoders DEC2 and DEC3 use the a posteriori value of previous decoder as the channel value.

6.1 Search of higher degree parity check polynomial for decoding

This paper searches higher degree parity check polynomials for DEC1, DEC2 and DEC3 by computer simulation.

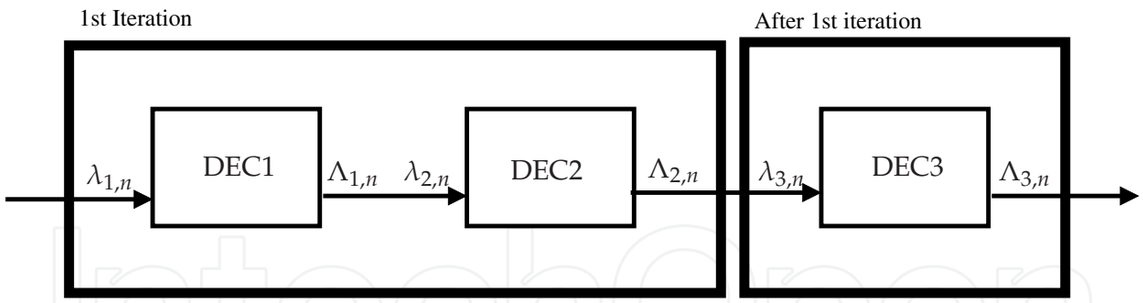


Fig. 13. Block diagram of decoder of switching parity check method for code rate 3/4

In this paper, the higher degree parity check polynomials for DEC1, DEC2 and DEC3 were selected from Table 4, Table 5 and Table 1, respectively. There are many number of combination of the higher degree parity check polynomials. Therefore, this paper searches the higher degree parity check polynomials as follows.

Step.1 At first, the higher degree parity check polynomial for DEC3 is determined by decoding simulation with only DEC3.

Step.2 With the determined higher degree parity check polynomial for DEC3, the higher degree parity check polynomials for DEC1 and DEC2 are determined by decoding simulation with DEC1, DEC2 and DEC3.

Figure 14 shows the simulation results of Step.1 at $E_b/N_0 = 6$[dB]. From Fig.14, it can be seen that the higher degree parity check polynomial with $\nu = 15$ provides the best performance. Therefore, that higher degree parity check polynomial is used.

Figure 15 shows the simulation results of Step.2 at E_b/N_0=7[dB]. From Fig.15, it can be seen that the combination of higher degree parity check polynomials No.2 and No.5 with scaling factor $f_s = 0.1$ provides the best performance, where scaling factor $f_s = 0.1$ is used for DEC1 and DEC2, and DEC3 uses fixed scaling factor $f_s = 1$.

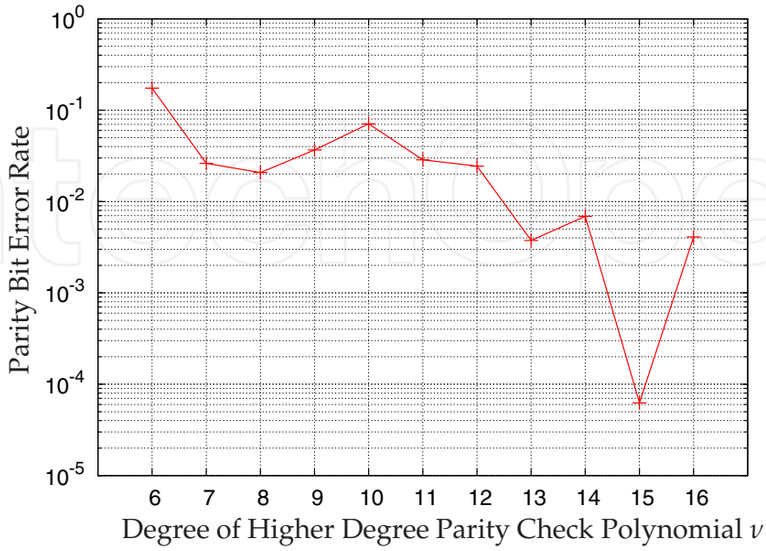


Fig. 14. Simulation results of step.1 in switching parity check method at $E_b/N_0 = 6$[dB]

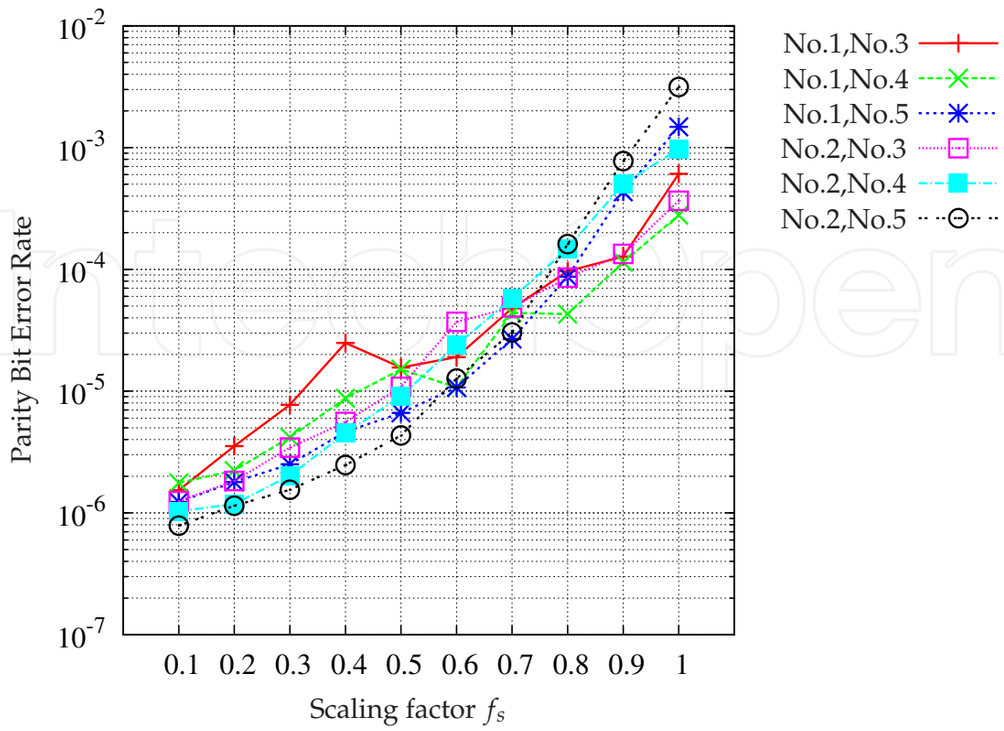


Fig. 15. Simulation results of step.2 in switching parity check method at $E_b/N_0 = 7$[dB]

6.2 Simulation results

Simulation results are shown in Fig.16 and 17. Figure 16 shows parity bit error rate performance. From Fig.16, it can be seen that parity bit error rate performance of the switching parity check method is 3.02[dB] superior to that of the conventional method and 2.2[dB] superior to that of the single punctured bit method. Parity bit error rate performance of the switching parity check method is only 1.04[dB] inferior to that of BCJR.

Figure 17 shows information bit error rate performance. From Fig.17, it can be seen that information bit error rate performance of the switching parity check method is 4.16[dB] superior to that of the conventional method and 3.05[dB] superior to that of the single punctured bit method. Information bit error rate performance of the switching parity check method is only 1.06 [dB] inferior to that of BCJR.

7. Decoding complexity

Table 6 and Table 7 show the numbers of operations per one bit decoding for sum-product algorithm and BCJR, respectively. In both tables, N_{add} denotes the number of additions, N_{mult} denotes the number of multiplications and N_{total} denotes the total number of operations. For sum-product algorithm, N_{sp} denotes the number of operations for $\tanh(\cdot)$, $\tanh^{-1}(\cdot)$. For BCJR, N_{sp} denotes the number of operations for $\exp(\cdot)$, $\log(\cdot)$. In Table 6, for information bits, N_{add} shows the number of XOR's. In Table 6, N_{itr} denotes the average number of iterations, where the number was counted at E_b/N_0=6[dB] by using computer simulation. For code rate 2/3, complexity of the single punctured bit method is shown. For code rate 3/4, complexity of the switching parity check method is shown. It is necessary to notice that iteration of sum-product algorithm is required for parity bits decoding only. For the switching parity

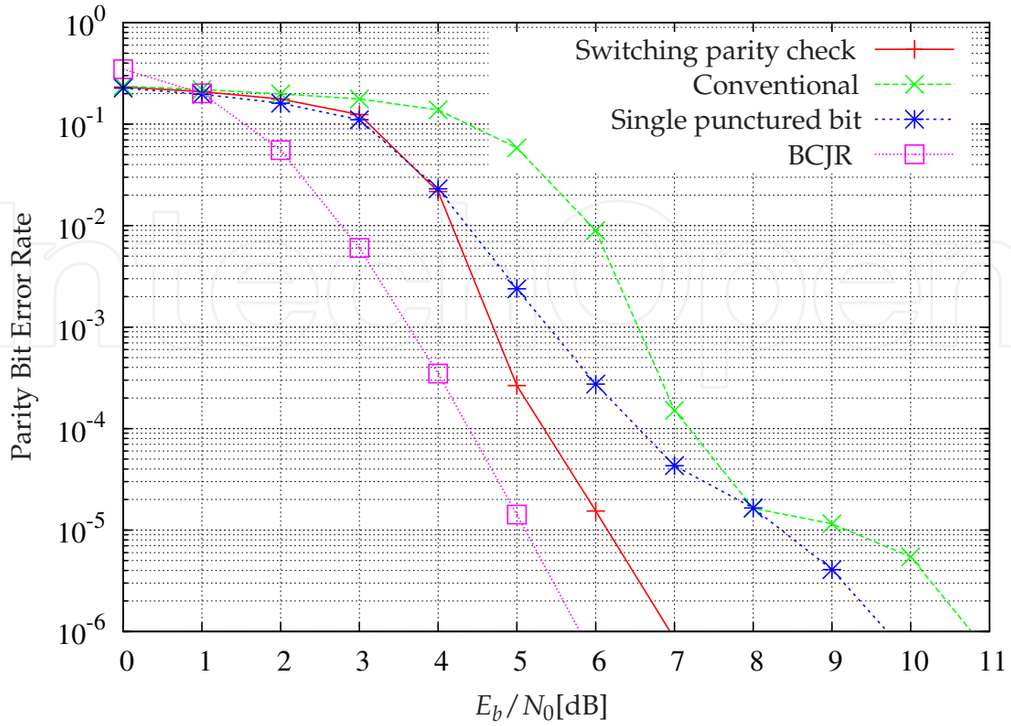


Fig. 16. Parity Bit Error Rate Performance of switching parity check method

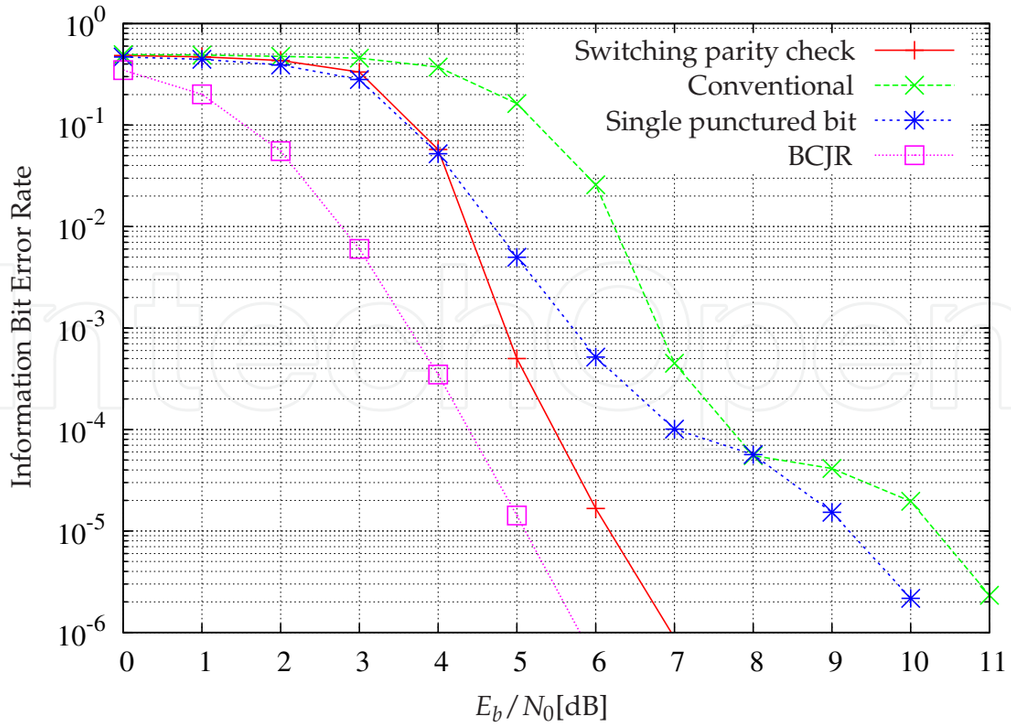


Fig. 17. Information Bit Error Rate Performance of switching parity check method

check method, it is necessary to notice that higher degree parity check polynomials No.2 and No.5 are used at only first iteration and after first iteration, higher degree parity check polynomials with degree $\nu = 15$ is used.

From those tables, for code rate 2/3, it can be seen that the number of operations of the single punctured bit method is 0.1 times of that of BCJR. For code rate 3/4, the number of operations of the switching parity check method is 0.2 times of that of BCJR.

For the both code rates, it can be seen that the number of operations of the proposed method is much less than that of BCJR.

Code rate	Classification	N_{add}	N_{mult}	N_{sp}	N_{itr}	N_{total}
2/3	Parity	10	23	24	3.04	175
	Info	1	0	0	1	
3/4	No.2	14	31	32	1	336
	No.5	14	31	32	1	
	$\nu = 15$	8	19	20	3.85	
	Info	1	0	0	1	

Table 6. Complexity of sum-product algorithm

Code rate	N_{add}	N_{mult}	N_{sp}	N_{total}
2/3, 3/4	640	1044	7	1691

Table 7. Complexity of BCJR

8. Conclusion

This paper proposes sum-product decoding methods for the punctured convolutional codes of wireless LAN. The wireless LAN standard include the punctured convolutional codes with code rate 2/3 and 3/4. This paper proposes to decode with the higher degree parity check polynomial that satisfies single punctured bit condition as the single punctured bit method. Single punctured bit condition is the condition to include single punctured bit in parity check equation. For code rate 2/3, the performance of the single punctured bit method is 1.28[dB] superior to that of the conventional method and only 0.98[dB] inferior to that of BCJR at bit error rate 10^{-5}. For code rate 3/4, the single punctured bit method can not provide good performance. To improve the performance, this paper proposes following method as the switching parity check method. Only at first iteration, the higher degree parity check polynomial satisfying single punctured bit condition is used to decode and after first iteration, another higher degree parity check polynomial with lower n_{fc} without single punctured bit condition is used to decode. For code rate 3/4, the performance of the switching parity check method is 4.16[dB] superior to that of the conventional method, 3.05[dB] superior to that of the single punctured bit method and only 1.06[dB] inferior to that of BCJR. Complexity of the single punctured bit method is 0.1 times of that of BCJR for code rate 2/3. For code rate 3/4, complexity of the switching parity check method is 0.2 times of that of BCJR. For the both code rates, complexity of the proposed method is much less than that of BCJR.

9. Acknowledgment

This work was supported by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (C) 23560444.

10. References

Benedetto, S.; Divsalar, D.; Montorsi, G. & Pollara, F. (1998). Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding. *IEEE Trans. Inf. Theory*, Vol. 44, No.3, 909-926

Berrou, C.; Glavieux, A. & Thitimajshima, P. (1993). Near shannon limit error-correcting coding and decoding : Turbo-codes (1). *IEEE International Conference on Communications (ICC93)*, 1064-1070

Berrou, C. & Glavieux, A. (1996). Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Trans. Commun.*, Vol. 44, No. 10, 1261-1271

Di, C.; Proietti, D.; Telatar, E.; Richardson, T.; & Urbanke, R. (2002). Finite length analysis of low-density parity-check codes, *IEEE Trans. Inf. Theory*, Vol.48, No.6, 1570-1579

Douillard, Catherine; Jézéquel, Michel; Berrou, Claude; Picart, Annie; Didier, Pierre & Glavieux, Alain (1995). Iterative correction of intersymbol interference: Turbo-Equalization. *European Trans. Telecommun.*, Vol.6, No.5, 507–511

Gallager, R. G. (1963). *Low Density Parity Check Codes*, Cambridge, MA: MIT Press

Hagenauer, J.; Offer, E. & Papke, L. (1996). Iterative decoding of binary block and convolutional codes. *IEEE Trans. Inf. Theory*, Vol. 42, No. 2, 429-445

IEEE Computer Society (2007). Part11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, *IEEE Std 802.11-2007*

Kschischang, Frank R.; Frey, Brendan J. & Loeliger, Hans-Andrea. Factor Graphs and the Sum-Product Algorithm, *IEEE Trans. Inf. Theory*, 498-519

Laot, Christophe; Glavieux, Alain & Labat, Joël (2001). Turbo equalization: adaptive equalization and channel decoding jointly optimized, *IEEE J. Selected Areas in Commun.*, Vol.19, No.9, 1744-1752

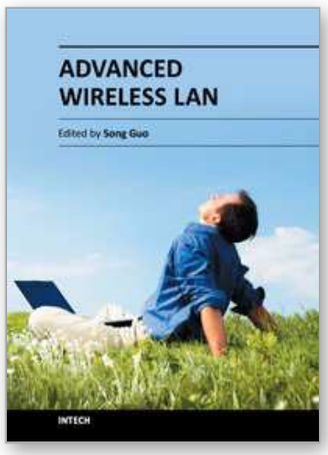
MacKay, D. J. C. (1999). Good error-correcting codes based on very sparse matrices, *IEEE Trans. Inf. Theory*, Vol. 45, No. 3, 399-431

Richardson, T. J. & Urbanke, R.L. (2001). The capacity of low-density parity-check codes under message-passing decoding, *IEEE Trans. Inf. Theory*, Vol. 47, No. 2, 599-618

Shohon, T.; Ogawa, Y. & Ogiwara, H. (2009a). Sum-Product decoding of convolutional codes, *The Fourth International Workshop on Signal Design and its Application in Communications (IWSDA'09)*, 64-67

Shohon, T.; Razi, F.; Ogawa, Y. & Ogiwara, H. (2009b). Sum-Product decoding of convolutional code for wireless LAN standard, *The Fourth International Workshop on Signal Design and its Application in Communications (IWSDA'09)*, 68-71

Shohon, T.; Razi, F. & Ogiwara, H. (2010). Sum-Product decoding of non-systematic convolutional codes, *Far East Journal of Electronics and Communications*, Vol.5, No.1, 25-35



Advanced Wireless LAN

Edited by Dr. Song Guo

ISBN 978-953-51-0645-6
Hard cover, 136 pages
Publisher InTech
Published online 05, June, 2012
Published in print edition June, 2012

The past two decades have witnessed starling advances in wireless LAN technologies that were stimulated by its increasing popularity in the home due to ease of installation, and in commercial complexes offering wireless access to their customers. This book presents some of the latest development status of wireless LAN, covering the topics on physical layer, MAC layer, QoS and systems. It provides an opportunity for both practitioners and researchers to explore the problems that arise in the rapidly developed technologies in wireless LAN.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen