

Performance Analysis of Punctured Turbo Code under An AWGN Channel

Ya-Fen Chen and Cheng-Ying Yang

Graduate Institute Of Networking and Communication Engineering

Choayang University of Technology

168 Gifeng E. Rd., Taichung 413, Taiwan, R.O.C.

Abstract

Turbo code is a coding technique that enhances a reliable data transmission. Because of special structure of encoder and iterative decoder, it could obtain 1dB at low E_b/N_0 db that closed the channel capacity limit. However, the redundant bits in coding would decrease the bandwidth efficiency. The puncturing scheme could improve the bandwidth efficiency. Different puncturing pattern could provide a different level of performance. We focus on the analysis of puncturing pattern to compare those performances. In this paper, we discuss the performance of punctured turbo coding on different location of punctured bits and different number of punctured bit.

keyword : Turbo code, Punctured Turbo Code, Channel coding, Performance analysis

I, Introduction

How to ensure that correct information and transmission efficiency is important in a communication system. Using a digital communication, it economizes on bandwidth and the digital signals could hardly to suffer from noise interference in transmission. Channel coding is used to protect the transmitted data in the digital communication system. The transmitted data are added redundant bits in channel coding. It could function as error correction or error detection. There are many type of channel coding, such that Hamming Code, Linear Block Code, Convolutional Code, Turbo Code..., etc. Each of channel coding has a different performance under different communication environments. Turbo code is proposed as a near to Shannon limit encoding [1]. Because of its structure of encoder and iterative decoding,

it could obtain a lower Bit Error Rate (BER) at the receiver. It would provide a largest data transmission in a finite bandwidth [10]. So, turbo code is proposed to the application on wireless communication system, such as asynchronous satellite communication, W-CDMA system..., etc [1-3]. Because of special structure of encoder and iterative decoder, turbo coding could approach a performance of error floor. However, the coding rate of turbo code is not perfect. Some researchers proposed punctured turbo code which could be applied in high code rate in communication system. The performance of punctured turbo code is worse than that of turbo code dose but coding scheme could improve bandwidth efficiency [3]. The puncturing pattern could be used to adjust code rate to meet the communication system requirement [4][7-8].

Although punctured turbo code could increase bandwidth efficiency, different punctured location could affect a different performance at the receiver. Kouse et al. [4] proposed the puncturing scheme with the codeword weight calculation. When the bit in codeword is with a high weight, it had to avoid puncturing that bit. It could not puncture the systematic sequence because the systematic bit is with a high weight. Consequently, the puncturing scheme should alternative the lower weight bits to be punctured. Hence, how to decide a good pattern in high rate system is an important task.

In this paper, the block size of the mother code is assumed to be 1024. Performance analysis focuses on the case of 4 bits in the puncturing pattern. Different punctured location and different number of punctured bits is provided in the analysis. The performance analysis on different location of punctured bits and number of punctured bits is presented in the following section. In the third section, the simulation results are given. Finally, a conclusion is given.

II、 Analysis of Puncturing pattern

In this section, the structure of punctured turbo code is described. The block diagram of punctured turbo code is given in Fig 1.

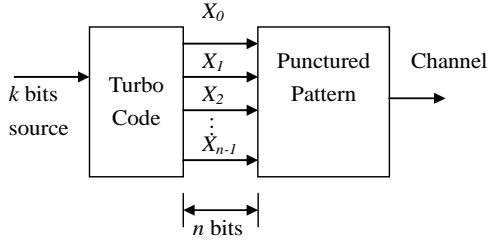


Fig 1 structure of punctured turbo code

At the beginning of turbo encoder, there is a k -bit source data to the encoder. After turbo encoder, there is an n -bit codeword corresponding to the source. This codeword vector $[X_0, X_1, \dots, X_{n-1}]$ is called as mother code [3]. For a systematic encoder, X_0 is represented as the systematic bit and the other bits are referred as the parity bits of the mother code. While the punctured scheme with q bits deletion is applied, the punctured turbo code is generated. The coding rate is defined as

$$R = k / (n - q) \quad (1)$$

and punctured rate is defined as

$$P.R. = q / (\text{total \# of parity bits}) \quad (2)$$

The puncturing scheme could be explained with a puncturing pattern. For example, the puncturing pattern is given as

$$P_{ij} = \begin{bmatrix} P_{11} & \cdots & P_{1p} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{np} \end{bmatrix}, \quad P_{ij} \in \{0,1\} \quad (3)$$

where $1 \leq i \leq n$, $1 \leq j \leq p$, $P_{ij}=0$ if the mother code P_{ij} is without transmission and p is the punctured period.

Different puncture patterns lead different BER performance [9]. In this paper, many types of puncture patterns are discussed in detail. To analyze puncture q -bits on the codeword vector $[X_0, X_1, X_2, \dots, X_{n-1}]$. There are two groups of puncture patterns. One is puncturing parity bits with non-uniformly distributed and the other is puncturing with uniformly distributed. First, puncturing q -bits only on X_i where i is 1 to $(n-1)$. Second, the puncturing q -bits is with uniformly distributed on $X_i, X_j \dots X_k$ simultaneously where $1 \leq i, j, \dots, k \leq (n-1)$. For example, if 1 parity bit is

punctured in the codeword vector $[X_1; X_2; \dots; X_{n-1}]$, there are $(n-1)p$ types patterns. When 2 parity bits are punctured in the codeword vector $[X_1; X_2; \dots; X_{n-1}]$, it may puncture on one of parity bits with uniformly distributed or puncture with non-uniformly distributed. Puncturing with uniformly distributed, there are $C_1^p \times C_1^p \times \frac{(n-1)}{p}$ patterns. There are $C_2^p \times \frac{(n-1)}{p}$ patterns with

non-uniformly puncturing. Furthermore, if q parity bits are punctured with uniformly distributed, there are

$$\left(C_{\frac{(n-1)}{q}}^p \times C_{\frac{(n-1)}{q}}^p \times \dots \times C_{\frac{(n-1)}{q}}^p \right) \times \frac{(n-1)}{p} \text{ patterns.}$$

Besides, uniformly or non-uniformly distributed puncturing patterns in codeword vector, there may be classified same locations patterns or different locations patterns during a puncturing period. It is easily to analyze. For example, when the 2-bits is punctured in a puncturing period, it may be classified with a same location of $[P_{11} \dots P_{1p}; P_{2i}, \dots, P_{2p}; P_{3i}, \dots, P_{3p}]$ where i is deletion bits with $1 \leq i \leq ((n-1)/p)$. Another pattern may continuous 4 bits in $[P_{11} \dots P_{1p}; P_{2i}, P_{2(i+1)}, \dots, P_{2p}; P_{3i}, P_{3(i+1)}, \dots, P_{3p}]$ where $1 \leq i \leq ((n-1)/p)$. The other pattern deletes different location bit during a puncturing period.

According to the above analysis steps, the numerical example is given in the following section.

III、 Simulation results

When systematic bit are punctured, it causes a serious degradation [3-5]. Hence, the analysis of puncturing pattern is given for simplification without puncturing the systematic bit. In order to analyze the puncturing pattern easily, it is defined octal replacement to represent the puncturing pattern. For example, the puncturing pattern is given in [1111;0111;1111], it is represented with a punctured matrix [17,07,17].

In this section, it discusses the puncture parity bits not only on X_1 and X_2 individually, but also on puncture X_1 and X_2 simultaneously. There are two types of puncture patterns with uniformly distributed format and non-uniformly distributed format. Hence, the various puncturing patterns with different bits deletion and a same coding rate are as show in Table 1 to Table 4. The coding rate is 4/11 and the P.R. is 1/8.

Table 1 One-bit deletion on X_1 and X_2

Item	R	P.R	Puncturing Pattern
X_1	4/11	1/8	[17,07,17],[17,13,17],[17,15,17], [17,16,17]
X_2	4/11	1/8	[17,17,07],[17,17,13],[17,17,15], [17,17,16]

When the P.R. is 2/8, there are six puncturing patterns shown in Table 2. The puncturing patterns are chosen the continuous deletion or scattered deletion. Based on the simulation, the pattern has to avoid a continuous deletion. It obtains a better performance when puncturing X_1X_2 is with uniformly distribution. The puncturing rule could be helpful to find out the optimal performance in different communication environment.

Table 2 Two-bit Deletion

Item	R	P.R	Puncturing Pattern
X_1	4/10	2/8	[17,03,17],[17,05,17],[17,06,17], [17,11,17],[17,12,17], [17,14,17]
X_2	4/10	2/8	[17,17,03],[17,17,05],[17,17,06], [17,17,11],[17,17,12],[17,17,14]
X_1X_2	4/10	2/8	Uniform patterns: [17,07,07],[17,07,13],[17,07,15], [17,07,16],[17,13,07],[17,13,13], [17,13,15],[17,13,16],[17,15,07], [17,15,13],[17,15,15],[17,15,16], [17,16,07],[17,16,13],[17,16,15], [17,16,16]

In Table 3, it is classifying with non-uniformly distributed patterns on X_1X_2 . There are three types of patterns that are with deletion on the same location or scattered location in the different patterns. For example, the patterns are [17,07,07] and [17,07,12] and [17,07,11].

Table 3 Three-bit deletion

Item	R	P.R.	Puncturing Pattern
X_1	4/9	3/8	[17,01,17],[17,02,17],[17,04,17],

			[17,10,17]
X_2	4/9	3/8	[17,17,01],[17,17,02],[17,17,04], [17,17,10]
X_1X_2	4/9	3/8	Non-uniform patterns: [17,07,03],[17,07,05],[17,07,06], [17,07,11],[17,07,12],[17,07,14], [17,13,03],[17,13,05],[17,13,06], [17,13,11],[17,13,12],[17,13,14], [17,15,03],[17,15,05],[17,15,06], [17,15,11],[17,15,12],[17,15,14], [17,16,03],[17,16,05],[17,16,06], [17,16,11],[17,16,12],[17,16,14], [17,03,07],[17,05,07],[17,06,07], [17,11,07],[17,12,07],[17,14,07], [17,03,13],[17,05,13],[17,06,13], [17,11,13],[17,12,13],[17,14,13], [17,03,15],[17,05,15],[17,06,15], [17,11,15],[17,12,15],[17,14,15], [17,03,16],[17,05,16],[17,06,16], [17,11,16],[17,12,16],[17,14,16]

In Table 4, the puncturing patterns are with deletion in parity bits X_1 and X_2 simultaneously. There are six patterns, such as [17,03, X_2], [17,05, X_2], [17,06, X_2], [17,11, X_2], [17,12, X_2], [17,14, X_2]. When it punctures four bits, there are two types with puncturing patterns. One is with uniformly distributed puncturing patterns and the other is non-uniformly distributed puncturing patterns shown in Table 4.

Table 4 Four-bit deletion

Item	R	P.R.	Puncturing Pattern
4			
X_1	4/8	4/8	[17,00,17]
X_2	4/8	4/8	[17,17,00]
X_1X_2	4/8	4/8	Uniform patterns: [17,03,03],[17,03,05],[17,03,06], [17,03,11],[17,03,12],[17,03,14], [17,05,03],[17,05,05],[17,05,06], [17,05,11],[17,05,12],[17,05,14], [17,06,03],[17,06,05],[17,06,06], [17,06,11],[17,06,12],[17,06,14],

			[17,11,03],[17,11,05],[17,11,06], [17,11,11],[17,11,12],[17,11,14], [17,12,03],[17,12,05],[17,12,06], [17,12,11],[17,12,12],[17,12,14], [17,14,03],[17,14,05],[17,14,06], [17,14,11], [17,14,12],[17,14,14]
X_1X_2	4/8	4/8	Non-uniform patterns: [17,07,01],[17,07,02],[17,07,04], [17,07,10],[17,13,01],[17,13,02], [17,13,04],[17,13,10],[17,15,01], [17,15,02],[17,15,04],[17,15,10], [17,16,01],[17,16,02],[17,16,04], [17,16,10],[17,01,07],[17,02,07], [17,04,07],[17,10,07],[17,01,13], [17,02,13],[17,04,13],[17,10,13], [17,01,15],[17,02,15],[17,04,15], [17,10,15],[17,01,16],[17,02,16], [17,04,16],[17,10,16]

Assume the mother code in this simulation is with the size of 1024 bits. The structure of Recursive Systematic Convolutional (RSC) encoder is (7,5) encoder. The puncturing pattern has one period of 4 bits and the puncturing length is 4. Fig 2 illustrates the performance with one bit puncturing. Based on simulation result, the pattern [17,13,17] and the pattern [17,17,07] have the best performances. With $R=4/10$ and $P.R.=2/8$, the performance is shown in Fig 3. At $BER=10^{-4}$, the pattern [17,16,13] may get coding gain about 0.2 dB compared with the pattern [17,17,03].

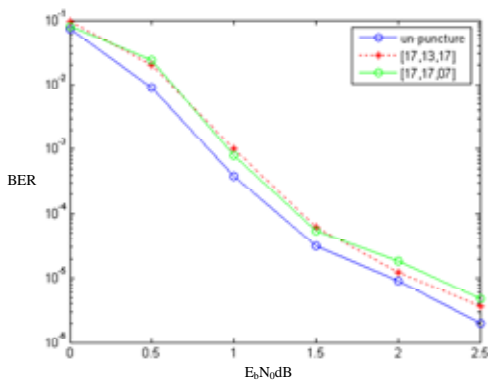


Fig 2 Performance with one bit puncturing when $R=4/11$ and $P.R.=1/8$

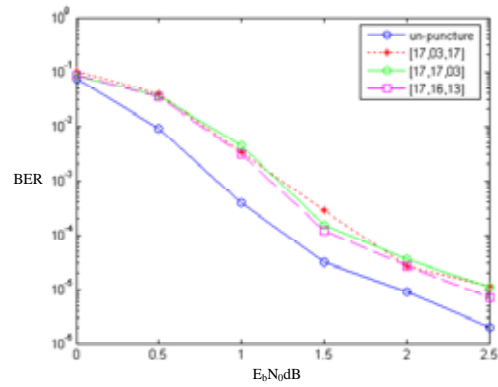


Fig 3 Performance with two bits puncturing when $R=4/10$ and $P.R.=2/8$

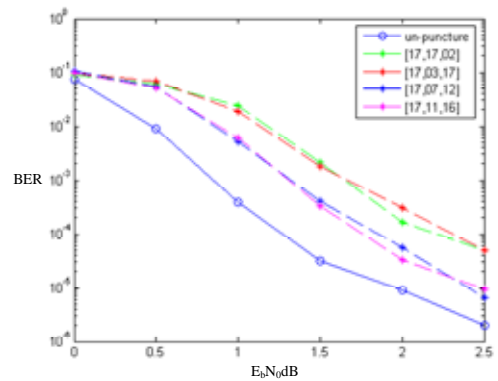


Fig 4 Performance with three bits puncturing when $R=4/10$ and $P.R.=3/8$

In Fig 4 and Fig 5, the simulation results are shown when it increases the deletion bits. As show in Fig 4 and Fig 5, the pattern punctures parity bits with uniformly distributed or non-uniformly better than punctures one of parity bits.

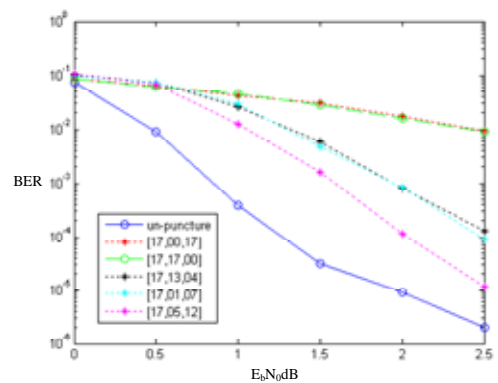


Fig 5 Performance with four bits puncturing when $R=4/8$ and $P.R.=4/8$

Table 5 and Table 6 illustrate the best puncturing pattern comparison with the same coding rate and the same puncturing rate and that with different coding rate

and different puncturing rate, respectively.

Table 5 Best puncturing pattern comparison with the same coding rate and the same puncturing rate

Delection bits	R	P.R.	The best puncturing pattern
puncture one bits	4/11	1/8	$X_1 : [17,13,17]$
	4/11	1/8	$X_2 : [17,17,07]$
puncture two bits	4/10	2/8	$X_1 : [17,03,17]$
	4/10	2/8	$X_2 : [17,17,03]$
	4/10	2/8	X_1X_2 uniformly : $[17,16,13]$
puncture three bits	4/9	3/8	$X_1 : [17,12,17]$
	4/9	3/8	$X_2 : [17,17,10]$
	4/9	3/8	Non-uniformly: Delete 2-bits in X_1 and 1-bit in X_2 : $[17,11,16]$ Delete 2-bits in X_2 and 1-bit in X_1 : $[17,07,12]$
puncture four bits	4/8	4/8	$X_1 : [17,00,17]$
	4/8	4/8	$X_2 : [17,17,00]$
	4/8	4/8	Uniformly : $[17,05,12]$ Non-uniformly: Delete 3-bits in X_2 and 1-bit in X_1 : $[17,13,04]$ Delete 3-bits in X_1 and 1-bit in X_2 : $[17,01,07]$

Table 6 Best puncturing pattern comparison with different coding rate and different puncturing rate

Parity bits	R	P.R.	The best puncturing pattern
X_1	4/11	1/8	Puncture one bit : $[17,13,17]$
	4/10	2/8	Puncture two bit : $[17,03,17]$
	4/9	3/8	Puncture three bit uniformly : $[17,02,17]$ Delete 2-bits in X_1 and 1-bit in X_2 : $[17,11,16]$
	4/8	4/8	Puncture four bits

			uniformly : $[17,00,17]$ Delete 3-bits in X_1 and 1-bit in X_2 : $[17,01,07]$
X_2	4/11	1/8	Puncture one bit : $[17,17,07]$
	4/10	2/8	Puncture two bit : $[17,17,03]$
	4/9	3/8	Puncture three bits uniformly : $[17,17,10]$ Delete 2-bits in X_2 and 1-bit in X_1 : $[17,07,12]$
	4/8	4/8	Puncture four bits uniformly : $[17,17,00]$ Delete 3-bits in X_2 and 1-bit in X_1 : $[17,13,04]$
X_1X_2	4/10	2/8	Puncture one bit : $[17,16,13]$
	4/8	4/8	Puncture two bits uniformly : $[17,05,12]$

In Fig 6, Fig 7 and Fig 8, when it increases the puncturing bits, the pattern $[17,00,17]$ is with a worst BER performance at parity bits, X_1 . When puncturing pattern increases puncturing rate at parity bit, X_1 (or X_2), there are a worst BER performance.

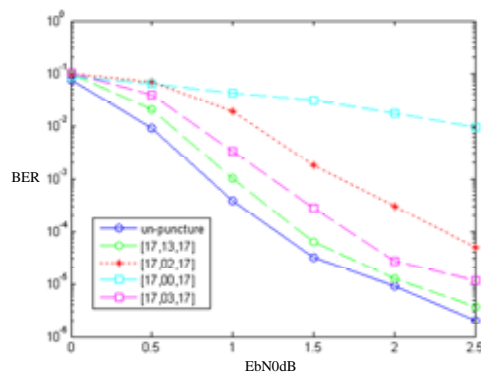


Fig 6 Performance comparison when X_1 is punctured with different number of punctured bits and different coding rate

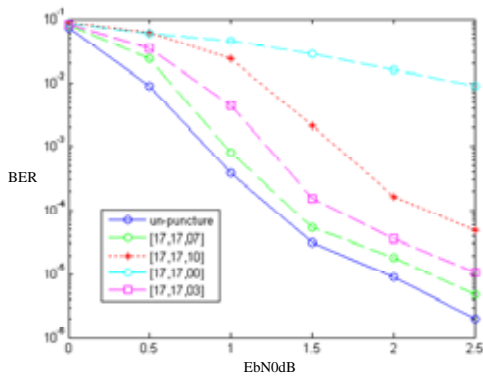


Fig 7 Performance comparison when X2 is punctured with different number of punctured bits and different coding rate

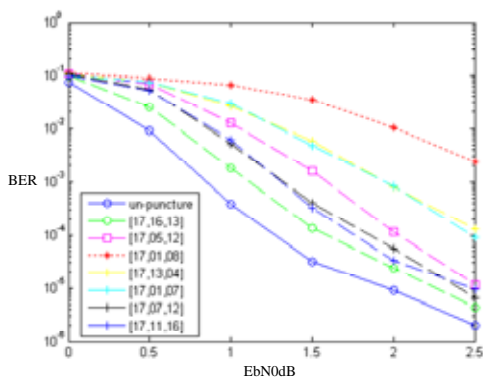


Fig 8 Performance comparison when X₁X₂ are punctured with different number of punctured bits and different coding rate

IV、 Conclusion

In this paper, the performance of turbo code with different puncturing patterns is presented. According to simulation result, when it deletes the same number of parity bits in a puncturing period, the puncturing pattern in composed of codeword vector $[X_1; X_2; \dots; X_{n-1}]$ has a better BER performance than that with the puncturing pattern in codeword vector $[X_1; X_2; \dots; X_{n-1}]$ independently. When the patterns punctured some parity bits with uniformly distributed, it has better performance than that with puncturing non-uniformly distributed. To increase deletion bits gradually, the performance may decrease oppositely with puncturing non-uniformly. To reach a near optimal pattern, it is an important scheme in digital communication system with turbo code application. The appropriate puncturing patterns might provide a high coding rate to meet the QoS requirements and to increase the bandwidth efficiency.

Reference

- [1] Claude Berror and Alain Glavieux, "Near Optimum Error Correcting Coding and Decoding: Turbo -Codes," IEEE Transactions on Communications, Vol.44, No.10, 1996, pp.1261-1271
- [2] Sergio Benedetto, Guido Montorsi, "Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes," IEEE Transactions on Information Theory, Vol. 2, No. 2, March 1996, pp.409-428
- [3] Branka, Jinhong, *Turbo Codes Principles and Applications*, Kluwer, 2000
- [4] M.A. Kousa and A.H. Mugaibel, "Puncturing effects on Turbo Codes," IEE Proceedings on Communications, Vol.149, 2002, pp.132-138
- [5] Fan Mo, Kwatra, S.C., Junghwan Kim, "Analysis of Puncturing Pattern for High Rte Turbo Code," Proceeding of IEEE MILCOM, Vol.1, 1999, pp.547-550
- [6] Bonghoe Kim and Sung Lark Kwon, "A new soft handover scheme using Punctured Turbo Codes in the Wideband CDMA system," Proceeding of IEEE VTC, 2001, pp. 1420-1424
- [7] Joachim Hagenauer, "Rate compatible punctured convolutional codes (RCPC Codes) and their Application," IEEE Transactions on Communications, Vol. 36, 1988, pp.389-400
- [8] F. Baich, G. Montorsi, and F. Vatta, "Some notes on rate-compatible punctured turbo codes (RCPTC) design," IEEE Transactions on Communications, Vol. 52, 2004, pp.681-684
- [9] Bernard Sklar, *Digital Communications Fundamentals And Applications*, Prentice-Hall, 2001
- [10] Claude Berrou, Alain Glavieux and Punya Thitimjshima, "Near Shannon Limit Error-Correcting Coding and Decoding:Turbo Codes," Proceeding of IEEE ICC, Vol.2, 1993, pp.1064-1070