THE EFFECT OF PUNCTURING ON THE CONVOLUTIONAL TURBO-CODES PERFORMANCES

Horia BALTA[1], Lucian TRIFINA[2], Anca RUSINARU[2]
[1]Electronics and Telecommunications Faculty, 2 Bd. V. Parvan, 1900 Timisoara, ROMANIA,
[2]Electronics and Telecommunications Faculty, Bd. Carol I, no. 11, Iasi, ROMANIA,

Abstract: In this paper the desired turbo coding rates providing methods using the puncturing process and also the effect of this process on BER and FER performances are presented. The performances of the RSC turbo codes with puncturing and without puncturing having the same component codes are compared. The methods of the parity bits puncturing are investigated. In all cases we used the same method of the component codes trellis closing. The simulations were performed for K=3, K=4 and K=5 constraint length convolutional codes and the interleaver length of 1784 bits. We assumed an AWGN transmission channel and BPSK modulation. The turbo decoder uses 15 iterations per block and an iterations stopping criterion based on the LLR (Log-Likelihood Ratio). In conclusion, there are indicated the most appropriate solutions of the turbo codes puncturing with K=3, K=4 and K=5 constraint length.
Keywords: turbo codes, puncturing.

Introduction

The block diagram of a punctured turbo code is shown in Fig. 1. The data sequence u is turbo coded by two convolutional coders C1, C2 and the interleaver I. The resulted sequences, c_1 and c_2, have length N, which is equal with the one of the data sequence u. For unpunctured TC, these sequences are transmitted to the channel. The resulting coding rate (without puncturing) is $R_c=1/3$. For obtaining higher rates we use the puncturing of the c_1 and c_2 sequences, which assumes the erasing or the eliminating of some bits from these sequences (this action is shown by P1 and P2 blocks in Figure 1).

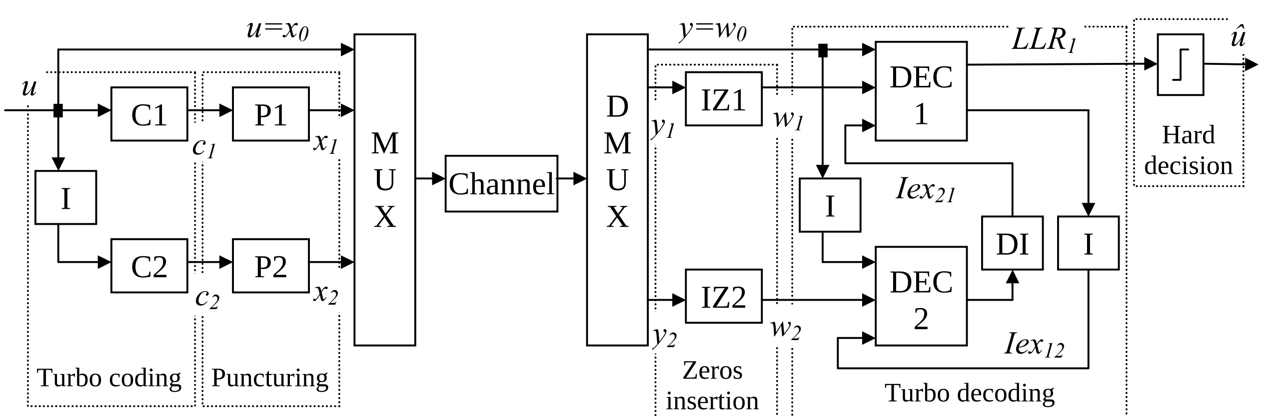


Fig.1. Scheme of a turbo coded transmission system with puncturing

The erased (punctured) bits are established by a puncturing matrix, G_p, which contains zeros and ones, and which can be interpreted like a "mask". For example, let be G_p =[*1 0 1; 0 1 0*]. The first line of G_p constitutes the mask for puncturing of the c_1 sequence and indicates that the second, from three bits, will be erased (the bits one and three are masked for the erasing operation). The puncturing of the c_2 sequences is made on the bits one and three, the second bit being masked.
After puncturing, the length of the resulted sequences, x_1 and x_2, is $2 \cdot N/3$ and $N/3$ respectively, resulting a turbo coding rate $R_{cp}=1/2$. In general, if the G_p dimension is $2 \times p$ and contains N_1 ones, then the turbo coding rate is

$$R_{cp} = \frac{p}{p + N_1} \quad (1)$$

The sequences which result by coding and puncturing, after multiplexing, are transmitted into the channel. At reception, after the reconstruction of the three sequences y_0, y_1 and y_2 by de-multiplexing, the length of N bits is rebuild, by inserting zeros in the place of the erased bits at emission (the blocks labeled IZ1 and IZ2). After that, the turbo decoding process is applied on the w_0, w_1 and w_2 sequences, like in the case without puncturing.

Each component decoder calculates the Log Likelihood Ratio (LLR) for each systematic bit:

$$LLR(u_i) = ln \frac{p(u_i = 1 | w)}{p(u_i = 0 | w)} \quad (2)$$

(in Figure 1 is indicated just LLRs for first component decoder, LLR_1), and also the extrinsic information destined to the second decoder, C2.

Each component decoder takes extrinsic information and, based on it and on soft channel inputs (w_0 and w_1 for C1, interleaved w_0 and w_2 for C2, respectively), provides another extrinsic information. This process iteratively repeated in a number of times (imposed or calculated, in function of turbo codes design). After the effectuation of all iterations, a hard decision is made on LLRs generated in last iteration by one of the two component decoders (in Figure 1 was chosen LLR_1). The resulting sequence by this operation constitutes the out of turbo decoder.

Often half-rate Recursive Systematic Convolutional (RSC) encoders are used. Figure 2 presents a RSC encoder with generator matrix:

$$G(D) = [1, \frac{1 + D^2}{1 + D + D^2}] \Leftrightarrow G_{octal} = [1, 5/7] \quad (3)$$

In contrast with the convolutional encoder, which has a simple hard implementation, shown in Figure 2, a decoder, to be component in turbo decoder, must accept soft input, and also must provide soft output.

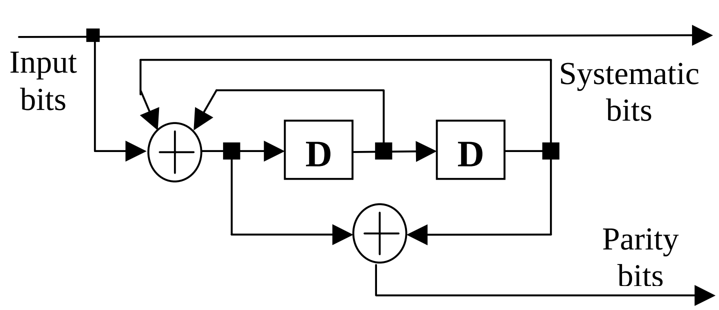


Fig. 2. Recursive Systematic Convolutional encoder, with G=[*1, 5/7*]

The Maximum A-Posteriori (MAP) algorithm

The Maximum A-Posteriori (MAP) algorithm, proposed by Bahl, Cocke, Jelinek and Raviv (1974), is the highest performance algorithm used in turbo decoding. This algorithm calculates the LLR under the form [1]:

$$L(u_k/\underline{w}) = ln\left(\frac{\sum_{\substack{(\hat{s},s)\Rightarrow \\ u_k=+1}} P\left(S_{k-1}=\hat{s} \wedge S_k = s \wedge \underline{w}\right)}{\sum_{\substack{(\hat{s},s)\Rightarrow \\ u_k=-1}} P\left(S_{k-1}=\hat{s} \wedge S_k = s \wedge \underline{w}\right)}\right) = ln\left(\frac{\sum_{\substack{(\hat{s},s)\Rightarrow \\ u_k=+1}} \alpha_{k-1}(\hat{s})\cdot\gamma_k(\hat{s},s)\cdot\beta_k(s)}{\sum_{\substack{(\hat{s},s)\Rightarrow \\ u_k=-1}} \alpha_{k-1}(\hat{s})\cdot\gamma_k(\hat{s},s)\cdot\beta_k(s)}\right) \quad (4)$$

where:

$\alpha_{k-1}(\hat{s}) = P(S_{k-1} = \hat{s} \wedge \underline{w}_{j<k})$ is the probability that the encoder trellis was in $\hat{s}$ state at instant $k-1$ and the received channel sequence, before this moment, is $\underline{w}_{j<k}$,

$$\alpha_k(s) = \sum_{\text{all }\hat{s}} \gamma_k(\hat{s},s)\cdot\alpha_{k-1}(\hat{s}) \quad (5)$$

$\beta_k(s) = P(\underline{w}_{j>k}/ S_k = s)$ is the probability that, having been given the trellis state s at instant k, the received channel sequence, after this moment, to be $\underline{w}_{j>k}$,

$$\beta_{k-1}(\hat{s}) = \sum_{\text{all }\hat{s}} \gamma_k(\hat{s},s)\cdot\beta_k(s) \quad (6)$$

$\gamma_k(\hat{s},s) = P(\{\underline{w}_k \wedge S_k = s\}/ S_{k-1} = \hat{s})$ is the probability that the encoder trellis took the transition from state $\hat{s}$ to state s and the received channel sequence for this transition is $\underline{w}_k$.

$$\gamma_k(\hat{s},s) = C\cdot e^{(u_k L(u_k)/2)}\cdot exp\left(\frac{E_b}{2\sigma^2}2\cdot a\sum_{i=1}^{n} w_{ki}\cdot u_{ki}\right) =$$

$$C\cdot e^{(u_k L(u_k)/2)}\cdot exp\left(\frac{L_c}{2}\sum_{i=1}^{n} w_{ki}\cdot u_{ki}\right), \quad (7)$$

Where $Lc = 2\cdot a\cdot E_b/\sigma^2$ is the channel reliability value and $L(u_k)$ is the interleaved extrinsic information, sometimes referred as a priori.

Experimental results

In the attached diagrams 18 *BER* curves are compared and presented, labeled as in Table I. The curves were obtained by simulating of a TC RSC operating (with the diagram shown in Fig. 1). Seven puncturing matrices labeled P1, … , P7 and four component convolutional codes (C1, …, C4) were used, also presented in Table I. An AWGN noise and a BPSK modulation were supposed. 15 iterations and a stopping criterion of them were proposed [3]. The interleaving was performed on blocks with length $N=1784$, [3], using a S-random interleaver ($S=29$). The number of simulated blocks was correlated with SNR. Thus, for $SNR=0$ dB, 1000 blocks × 1784 information bits per block were transmitted. For SNR values at which *BER* was below 10^{-6}, over 10^6 blocks × 1784 information bits per block were transmitted.

In the diagram 1 (*BER* and *FER*) and 2 (*BER* and *FER*), respectively, a comparison between the different puncturing matrices for the 5/7 and 15/13 codes, having the same coding rate $R_c=1/2$, is presented. Analyzing the results we observe that: i) the best performances were obtained using symmetrical matrices; ii) it must be a correlation between the puncturing matrix and the involved codes. Thus, we obtained poor results with P4 matrix and 5/7 code, but very good results with the same matrix and 15/13 code.

The diagrams 3 and 4 compare the various codes with the same puncturing matrix P1 and P6, respectively (for $R_c = 1/2$ and $R_c = 3/5$, respectively). The performances obtained by the codes with the $K=4$ or 5 constraint length are very similar, at least up to $BER=10^{-6}$. The performances obtained by the 5/7 code at $R_c=3/5$ are very poor.

The diagrams 5 and 6 presents the performances obtained for different coding rates of two codes: 5/7 ($K=3$) and 15/13 ($K=4$). Paradoxically, the performance of 5/7 code at $R_c = 3/5$ is more poor than at $R_c=2/3$. This fact is due to the inadequate choosing of the puncturing matrix. For 15/13 code the performance decrease inverse proportional with the coding rate. At $BER=10^{-5}$ the departure is 0.5 dB approximately.

Conclusions

In this paper the effects of puncturing on the TC performances are presented. The choice of the puncturing matrix has to be done in correlation with the code choice. The codes having $K \geq 4$ presents similar performances up to $BER>10^{-7}$. The coding rate is a decisive factor which influences the performance.

References

[1] C. Berrou, A. Glavieux, P. Thitimajshima – "Near Shannon limit error-correcting coding and decoding: Turbo-codes", Proc.ICC'93, Geneva, Switzerland, pp. 1064 – 1070, May, 1993.

[2] L. Hanzo, T.H. Liew, B.L. Yeap – "Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels", John Wiley & Sons Ltd, England, 2002

[3] L. Trifina, H. Balta, A. Rusinaru – "Decreasing of the Turbo MAP Decoding Time by Using an Iterations Stopping Criterion", in course of publication

Table I

R_c	Puncturing matrix		Component code – octal form			
			$K=3$	$K=4$	$K=4$	$K=5$
	Number	P	5/7	15/13	13/15	23/31
1/2	1	10 01	P1C1	P1C2	P1C3	P1C4
1/2	2	1001 0110	P2C1	P2C2	-	-
1/2	3	10001 01110	P3C1	P3C2	-	-
1/2	4	110 001	P4C1	P4C2	-	-
1/2	5	1110 0001	P5C1	P5C2	-	-
3/5	6	100 001	P6C1	P6C2	P6C3	P6C4
2/3	7	1000 0100	P7C1	P7C2	-	-

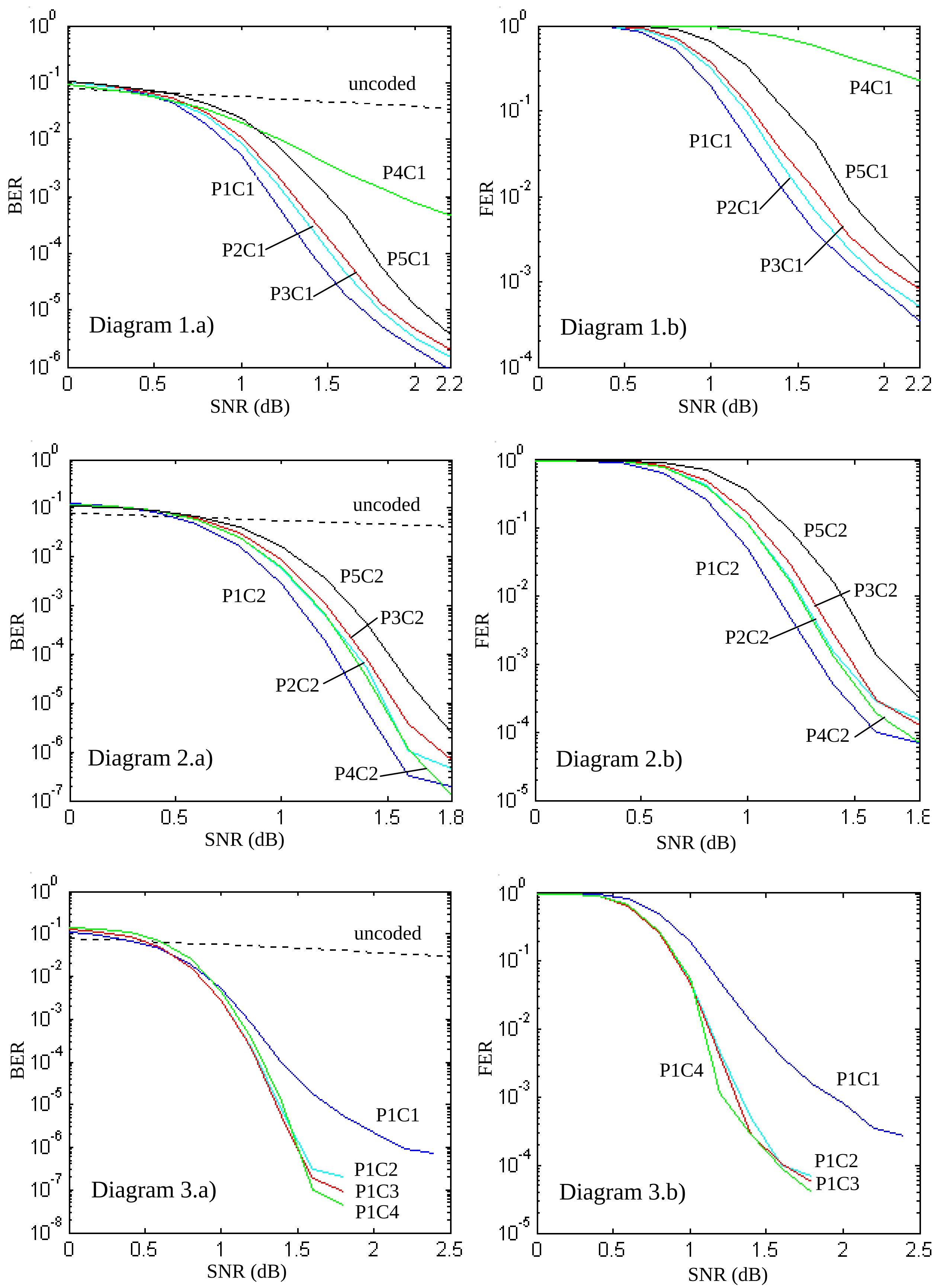
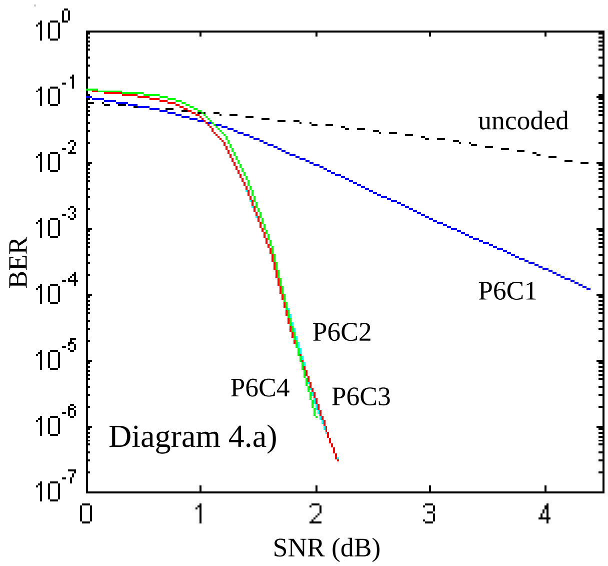
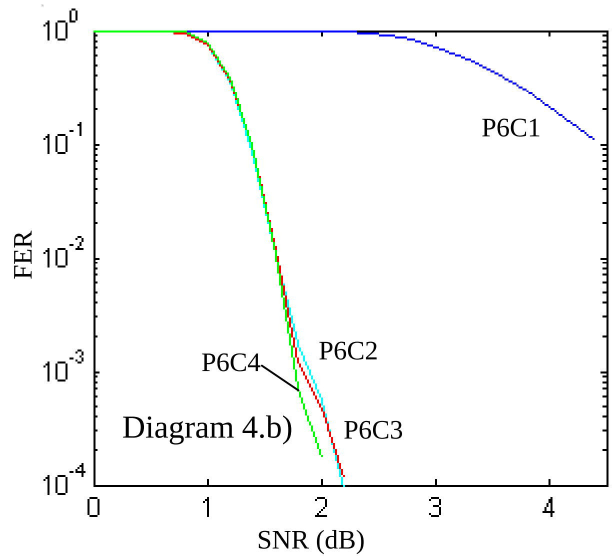
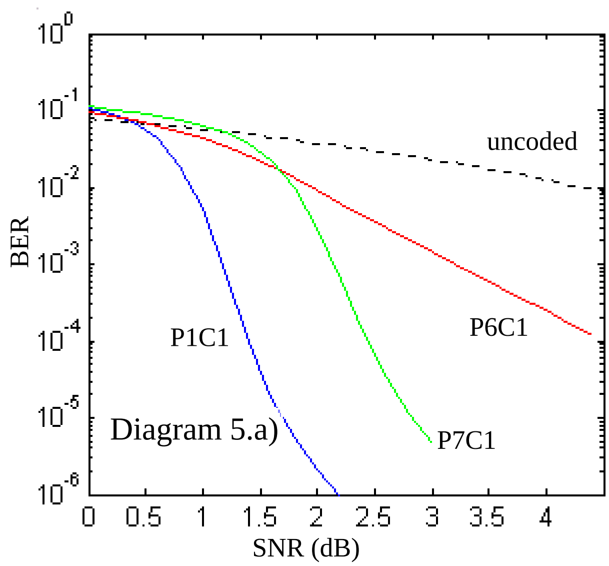
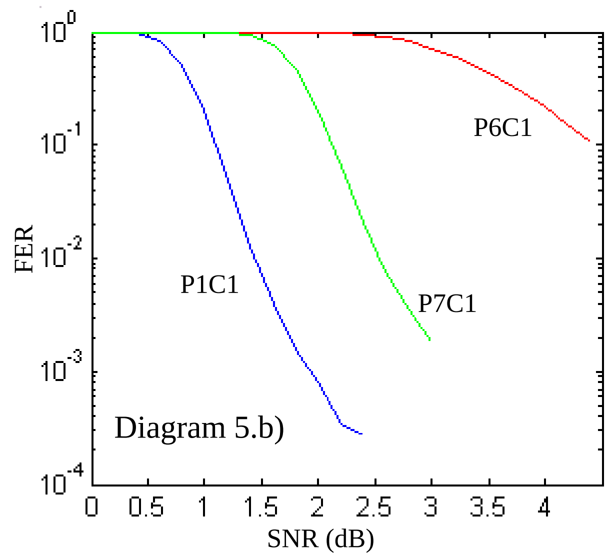


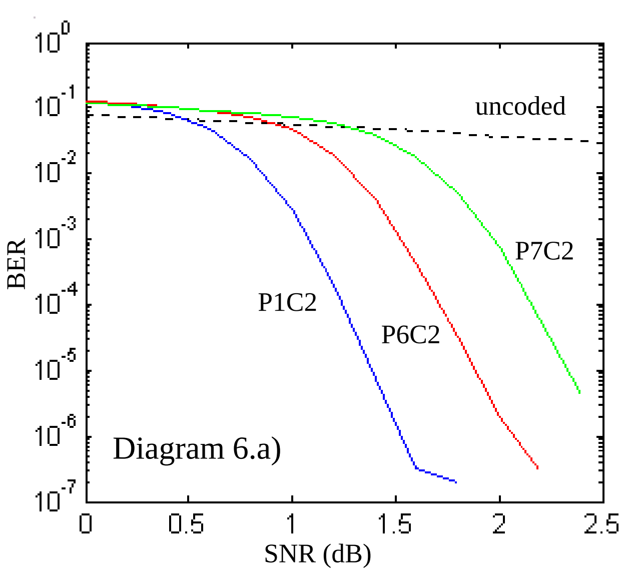
Diagram 1.a)
BER
SNR (dB)
uncoded
P4C1
P1C1
P2C1
P3C1
P5C1
Diagram 1.b)
FER
SNR (dB)
P4C1
P1C1
P5C1
P2C1
P3C1
Diagram 2.a)
BER
SNR (dB)
uncoded
P5C2
P1C2
P3C2
P2C2
P4C2
Diagram 2.b)
FER
SNR (dB)
P5C2
P1C2
P3C2
P2C2
P4C2
Diagram 3.a)
BER
SNR (dB)
uncoded
P1C1
P1C2
P1C3
P1C4
Diagram 3.b)
FER
SNR (dB)
P1C4
P1C1
P1C2
P1C3

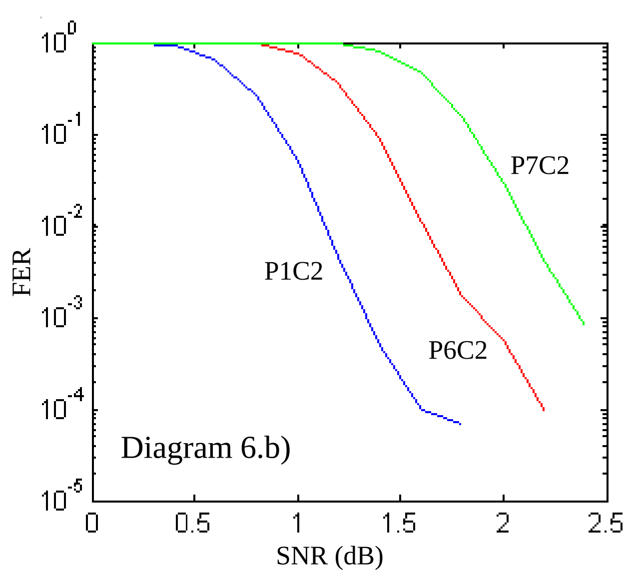

BER
SNR (dB)
uncoded
P6C1
P6C2
P6C4
P6C3
Diagram 4.a)


FER
SNR (dB)
P6C1
P6C2
P6C4
P6C3
Diagram 4.b)


BER
SNR (dB)
uncoded
P1C1
P6C1
P7C1
Diagram 5.a)


FER
SNR (dB)
P6C1
P1C1
P7C1
Diagram 5.b)


BER
SNR (dB)
uncoded
P7C2
P1C2
P6C2
Diagram 6.a)


FER
SNR (dB)
P7C2
P1C2
P6C2
Diagram 6.b)