



Performance Evaluation and Comparison of the BPTC using a new Soft Hamming Decoder

ALAA A. GHAITH

HKS Laboratory - Electronics and Physics Dept.

Faculty of Sciences I - Lebanese University

Lebanon

HAMZÉ H. ALAEDDINE

HKS Laboratory - Electronics and Physics Dept.

Faculty of Sciences I - Lebanese University

Lebanon

Abstract:

The block product turbo code (BPTC) is classified as one of block turbo code concatenation forms. The Hamming code can detect two-bit error and correct one-bit error. The BPTC uses two Hamming codes for "column" coding and "row" coding, it has improved the Hamming code correcting only one error. In addition, the BPTC carries out block interleaving coding for disorganizing the transmission sequence before transmission, so as to avoid burst errors when the signal meets multi-path channel in the channel. In this paper, we are interested by the impact of the soft decoding on the performance of a BPTC system and the comparison of different schemes BPTC system composed from two well-known Hamming codes concatenated together. So we will discuss the decoding mechanism of the BPTC and analyse the efficiency of using a soft decoding algorithm in the decoding process. The main BPTC considered in this paper is the (196, 96), used the Hamming (15,11) and Hamming (13,9) block channel code combinations and a block interleaver in the concept of feedback encoding in turbo code. This BPTC (196,96) coding system is well used in different standard, so as to observe its coding efficiency and compare the coding gain with one-dimension Hamming code and other schemes of two-dimension Hamming code (BPTC). The article presents

a design of soft decoding of the BPTC system, which is composed of two, similar or different, Hamming block code combinations, a block interleaving, and a BPSK modulation, over an AWGN channel. Different schemes were simulated to observe its coding improvement, and compare the bit error rate (BER) for different soft decoding of the BPTC.

Key words: Syndrome based soft decoding, Hamming Code, soft-output, BPTC, AWGN, block interleaver, coding gain, turbo code.

1. INTRODUCTION

Concatenated coding schemes were first proposed by Forney [Forney 1966] as a method for achieving large coding gains by combining two or more relatively simple block or component codes. The resulting codes had the error-correction capability of much longer codes, and they were endowed with a structure that permitted relatively easy to moderately complex decoding. A serial concatenation of codes is most often used for power-limited systems. The most popular of these schemes consists of a Reed-Solomon outer (applied first, removed last) code followed by a convolutional inner (applied last, removed first) code. A turbo code [Berrou 1993] can be thought of as a refinement of the concatenated encoding structure plus an iterative algorithm for decoding the associated code sequence. Turbo codes were first introduced in 1993 by Berrou, and Glavieux, where a scheme is described that achieves a bit-error probability of 10^{-5} using a rate 1/2 code over an AWGN channel and BPSK modulation at an E_b/N_0 of 0.7 dB. The codes are constructed by using two or more component codes on different interleaved versions of the same information sequence. Whereas, for conventional codes, the final step at the decoder yields hard-decision decoded bits (symbols), for a concatenated scheme such as a turbo code to work properly, the decoding algorithm should not limit itself to passing hard decisions among the decoders. To

best exploit the information learned from each decoder, the decoding algorithm must effect an exchange of soft decisions rather than hard decisions. For a system with two component codes, the concept behind turbo decoding is to pass soft decisions from the output of one decoder to the input of the other decoder, and to iterate this process several times so as to produce more reliable decisions. The purpose was to find digital communications systems that have a capacity and a performance close to the limits found by Shannon. For applications that require error correcting codes to operate with much shorter delays, Berrou, Evano, and Battail have advocated block component codes, maintaining turbo coding decoding principle. These codes, called turbo-block codes, exhibit a coding gain that is considerably larger than that of the standalone component block codes [Divsalar 2001]. Moreover, the decoding complexity for these turbo-block codes is quite reasonable, as long as the decoding complexity of the component block codes is so [Divsalar 2004]. The block product turbo code (BPTC) is classified as one of block turbo code concatenation forms. The Hamming code can detect two-bit error or correct one-bit error [Li 2004]. The BPTC uses two Hamming codes for "column" coding and "row" coding, it has improved the Hamming code correcting only one error. In addition, the BPTC carries out block interleaving coding for disorganizing the transmission sequence before transmission, so as to avoid burst errors [Li 2004] and [Huang 2007].

We will introduce here the same concept behind the turbo decoding which is to pass the soft decisions between both decoders, and to iterate this process several times to produce more reliable decisions [Muller 2011]. The major contributions of this work are presented by the proposition of a new soft decoding algorithm which can achieve a linear complexity with a small degradation compared to maximum likelihood decoding [Cho 2011] but it permits to us to consider the error patterns with three errors, and the utilization of this soft decoder in the

turbo principal process which will provide a lot of gain when compared to the turbo hard decision decoder and to the single soft decoder [Chen 2011] and [Megha 2014].

The remainder of this paper is organized as follows. In Section 2, we discuss the background about the Hamming codes. Section 3 presents the syndrome based soft, and the soft decoding technique. The system design of the BPTC coding and decoding schemes are also presented in Section 4. The system performance is investigated in Section 5 through extensive trace-driven simulation. Finally, conclusions are given in Section 6 along with the suggestions for future work.

2. THEORETICAL BACKGROUND

2.1. Encoding and Transmission

The encoding of the message bits' \mathbf{m} can be performed by a modulo 2 vector matrix multiplication of \mathbf{m} and the generator matrix \mathbf{G}

$$\mathbf{c} \equiv \mathbf{m} \cdot \mathbf{G} \quad (1)$$

The expression " \equiv " is equivalent with $\mathbf{c} = (\mathbf{m} \cdot \mathbf{G})$ modulo 2.

Hamming code is an important forward error correction (FEC) in theory and practice so far. It is a sort of binary linear block code. It puts forward an important single-error-correcting code, using parity check matrix (H) to detect and correct errors. It is a simple type of systematic code, described as the following structure.

Block length: $\mathbf{n} = 2^p - 1$

Number of data bits: $\mathbf{k} = 2^p - p - 1$

Number of check bits: $\mathbf{n} - \mathbf{k} = p$

Minimum distance: $\mathbf{d}_{min} = 3 \rightarrow$ Correct single bit error

$$(\mathbf{n}, \mathbf{k}) = (2^p - 1, 2^p - 1 - p)$$

We can use the following $\mathbf{k} \times \mathbf{n}$ array to define the generator matrix (G)

$$G = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_k \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ V_{k1} & V_{k2} & \cdots & V_{kn} \end{bmatrix}$$

Among which, $V_1 \sim V_k$ are linearly independent vectors that can generate all code vectors. The data of the transmitting terminal are usually expressed in column vectors, therefore, the sequence of k message bits, i.e. the message \mathbf{m} is expressed as $1 \times k$ matrix.

The generator matrix of systematic (7, 4)-Hamming code is given by

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & | & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & | & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & | & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & | & 1 & 0 & 1 \end{bmatrix}$$

After the encoding, \mathbf{c} is modulated, so that a logical zero is equivalent to a +1 and a logical one is equivalent to a -1, $x \in \{\pm 1\}$. The modulated signal \mathbf{x} is distorted by the additive white Gaussian noise (AWGN) \mathbf{w} and results in the receive signal \mathbf{y} ,

$$\mathbf{y} = \mathbf{x} + \mathbf{w} \quad (2)$$

2.2. Hard Decision Decoding

In order to decode the received signals, we need to define a parity check matrix and a syndrome. There is a $(n-k) \times n$ matrix \mathbf{H} in each generator matrix \mathbf{G} , so that the columns of \mathbf{G} are orthogonal to the columns of \mathbf{H} , i.e. $\mathbf{G} \cdot \mathbf{H}^T = 0$, the \mathbf{H}^T is the transpose matrix of \mathbf{H} . In order to meet the orthogonality of system coding, the component of matrix \mathbf{H} can be expressed as $\mathbf{H} = [I_{n-k} | P^T]$. Therefore, the matrix \mathbf{H}^T can be expressed as

$$\mathbf{H}^T = \left[\frac{I_{n-k}}{P^T} \right]$$

\mathbf{c} is the code word derived from matrix \mathbf{G} if and only if $\mathbf{c} \cdot \mathbf{H}^T = 0$. Let \mathbf{r} be the vector received by the receiving terminal, so the \mathbf{r} can be described as $\mathbf{r} = \mathbf{c} + \mathbf{e}$ among which, \mathbf{e} is the error vector resulted from the channel. The syndrome is defined as $\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T$,

it is the result of parity check implemented in \mathbf{r} , judge whether \mathbf{r} is an element in the codeword set. Based on development of equation $\mathbf{s} = (\mathbf{c} + \mathbf{e}) \cdot \mathbf{H}^T = \mathbf{c} \cdot \mathbf{H}^T + \mathbf{e} \cdot \mathbf{H}^T$. However, for all elements in codeword set $\mathbf{c} \cdot \mathbf{H}^T = 0$, therefore $\mathbf{s} = \mathbf{e} \cdot \mathbf{H}^T$.

Since the correction capability of Hamming code is 1, meaning the error pattern is one selected from \mathbf{n} . Error patterns with 2 (duets) or 3 errors (triplets) which belong to the same syndrome are not taken into account for the decoding and the distorted code word is corrected as follow:

1. Use $\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T$ to calculate the syndrome of \mathbf{r}
2. Find out common first error pattern \mathbf{e}_j , its syndrome equals $\mathbf{r} \cdot \mathbf{H}^T$
3. This error pattern is supposed to be the error caused by channel
4. The identified receive correction vector or code word equals $\mathbf{c} = \mathbf{r} + \mathbf{e}_j$

In fact, every double error is decoded to a valid but wrong code word. This explains the poor performance of HDD for Hamming codes.

3. SOFT-OUTPUT DECODING

3.1. Syndrome Based Soft Decision Decoding

For the syndrome based soft decision decoding it is required to calculate the log-likelihood ratios (LLR) from the received signal y ,

$$L(x | y) = \ln \frac{P(x = +1 | y)}{P(x = -1 | y)} \quad (3)$$

which finally leads to

$$L(x | y) = \ln \frac{\exp\left(-\frac{E_b}{N_0}(y-1)^2\right)}{\exp\left(-\frac{E_b}{N_0}(y+1)^2\right)} = \frac{4E_b}{N_0} y, \quad (4)$$

assuming that a logical zero and one have the same probability.

Let us assume that the syndrome of the distorted bit sequence of a (7,4)-Hamming code is $\mathbf{s} = (0 \ 0 \ 1)$. The possible error patterns are collected in matrix \mathbf{E} with its elements $e_{j,i}$,

$$E = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

where the second to fourth row bears the duets and the fifth to eighth row bears the triplets. The error patterns for all syndromes are determined in advance and stored in a list. The size of the list rises quadratically for double errors and cubically for triple errors. Every row of \mathbf{E} is multiplied by the absolute value of LLRs of the received signal $L(\mathbf{x}|\mathbf{y})$. Afterwards, the resulting row vector is added up. The vector with the lowest sum of LLR suggests the error pattern with the highest probability of a correct decoding.

3.2. Soft Decoding

The structure of the soft hamming decoder is shown in Fig. 1. In general, soft-output decoding provides output values for iterative or turbo decoding.

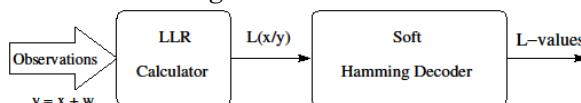


Figure 1: Soft-Output Hamming Decoder

In order to generate soft-outputs, the following algorithm is proposed. The probability values of a code word are given by

$$P(\hat{c}_j = c_j | y_j) = \frac{\exp(|L(x_j | y_j)|)}{1 + \exp(|L(x_j | y_j)|)} \quad (5)$$

Next, the probability values are multiplied column-wise for the given error pattern of every row i .

$$\tilde{P}_i = \prod_j \tilde{P}_j \text{ where } \tilde{P}_j = \begin{cases} P(\hat{c}_j = c_j | y_j) & \text{if } e_{i,j} = 0 \\ 1 - P(\hat{c}_j = c_j | y_j) & \text{if } e_{i,j} = 1 \end{cases} \quad (6)$$

Now is normalized, so that the sum of the normalized

probabilities \mathbf{P}_i over all rows i is equal to 1, $\sum_i P_i = 1$. So the probabilities P_i are given by

$$P_i = \frac{\tilde{P}_i}{\sum_i \tilde{P}_i} \quad (7)$$

P_i can be interpreted as the probability of correct decoding for the given error pattern of row i . In a last step, the probability that $x_j = +1$, for a given received code word y , is calculated by the sum of P_i over all rows i , if $e_{i,j} = \hat{c}_j$, where \hat{c}_j is defined as the logical received bit sequence. The estimation of the new probabilities after the soft decoding.

$$\hat{P}(x_j = +1 | y) = \sum_{\substack{i \\ e_{i,j} = \hat{c}_j}} P_i \quad (8)$$

Due to the normalization, so that $\sum_i P_i = 1$, the probability of

$\hat{P}(x_j = -1 | y)$ can be calculated by

$$\hat{P}(x_j = -1 | y) = 1 - \hat{P}(x_j = +1 | y) \quad (9)$$

In order to exchange the information for turbo decoding it is required to calculate L-values from the derived probabilities.

3.3. Some Simulation Results

For the simulation results of the soft decoding, hamming codes of a code word length for 7 till 127 bit were investigated. Fig. 2, 3, and 4 illustrate the performance of the different decoding strategies for a certain code word length. Tab. 1 summarizes the results for all non-iterative codes.

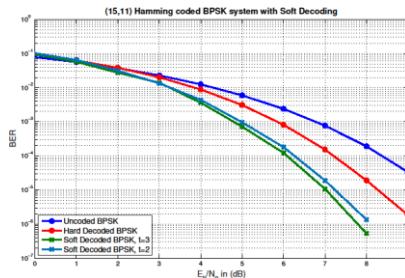


Figure 2: (15,11)-Hamming code

Fig. 2 shows the bit error rate of the (15,11)-Hamming code for different types of decoding. It is shown that the decoding performance of the duet and triplet decoding is very close to the union bound which is an upper bound for the bit error probability after maximum likelihood decoding. For the evaluation we focus on a $\text{BER}=10^{-4}$. The coding gain amounts to 0.9 dB for the HDD and 1.8 dB for the duet decoding. Further 0.25 dB can be gained by triplet decoding.

N_c	K_c	HDD Gain	Duets Gain	Triplets Gain			
7	4	7.80	0.31	6.75	1.36	6.65	1.46
15	11	7.20	0.90	6.30	1.80	6.05	2.05
31	26	7.00	1.21	6.05	2.16	5.80	2.41
63	57	7.00	1.21	6.05	2.16	5.75	2.46
127	120	7.10	1.11	6.35	1.86	6.00	2.21

Table 1: Coding gain for E_b/N_0 in dB for a $\text{BER}=10^{-4}$.

The (31,26)-Hamming code obtained the results for codes, for duets as well as for triplets. Fig. 3 shows that the coding gain amounts to 2.16 dB for the duet decoding and 2.41 dB for the triplet decoding.

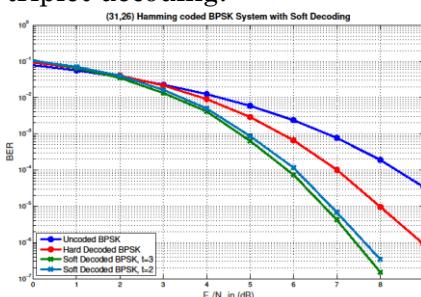


Figure 3: (31,26)-Hamming code

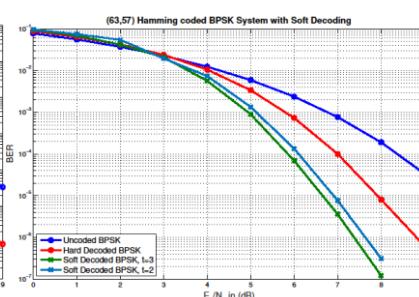


Figure 4: (63,57)-Hamming code

Quite a similar picture can be drawn for the (63,57)-Hamming code as shown in Fig. 4. The coding gain is similar to the (31,26)-Hamming code, but the bit error curve falls sharply. It is also shown that the difference between duet and triplet decoding is higher with 0.3 dB.

4. BPTC CODING AND DECODING SCHEMES

Hamming put forward an important error-correcting code, Hamming code in 1948. It uses parity check matrix (H) to detect and correct errors, however, its ability in detection and correction is limited, it can only detect 2-bit errors or correct 1-bit errors.

The block product turbo code (BPTC) is classified as one of block turbo code concatenation forms. The Hamming code can detect two-bit error or correct one-bit error. The BPTC uses two Hamming codes for "column" coding and "row" coding, it has improved the Hamming code correcting only one error. The encoder starts with the first row of information bits, calculates and appends the parity bits, and then moves on to the second row. This is repeated for each row. Next, the encoder starts with the first column of information bits, calculates and appends the parity bits for that column, and moves to the next column. Once the information block is complete, the encoder calculates and appends parity bits onto rows. It is important to note that different code lengths may be used for the horizontal and vertical blocks. In addition, the BPTC carries out block interleaving coding for disorganizing the transmission sequence before transmission, so as to avoid burst errors.

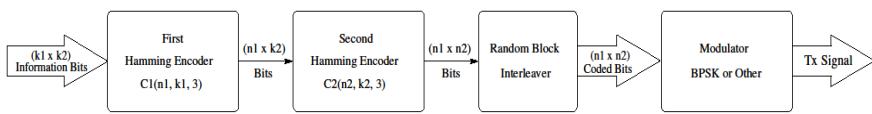


Figure 5: BPTC Encoder

4.1. Encoder System Design

The general structure of a BPTC encoder is shown in Fig. 5. It consists of two systematic hamming encoders C_1 and C_2 . It should be noted that the size of these two hamming codes could be the same and the free distance of any hamming code is always 3 which means it can correct one-bit error. The output sequences, however, are the same for identical input sequences. The N bit data block is first arranged in $(k_1 \times k_2)$ matrix form before encoded by C_1 , an additional zero padding bits are placed at the end of the data block if needed. After encoded by the first encoder, the output block is then $(n_1 \times k_2)$ matrix after adding the corresponding bits to each column. The output data block of the C_1 is also encoded by C_2 giving an output encoded data of $(n_1 \times n_2)$ matrix after adding the corresponding bits to each row. Then this data block will be interleaved by a random interleaver. The main purpose of the interleaver is to randomize bursty error patterns so that it can be correctly decoded. It also helps to increase the minimum distance of the BPTC. The turbo coder obtained here can be described with the following structure.

Block length: $N = n_1 \times n_2$

Number of data bits: $K = k_1 \times k_2$

Number of check bits: $P = (n_1 - k_1) \times k_2 + (n_2 - k_2) \times k_1 + k_1 \times k_2$

Coding Rate: $R = K/N = R_1 \times R_2$.

4.2. Decoder System Design

The decoding procedure described below is generalized by cascading elementary decoders illustrated in Fig. 6. Let us consider the decoding of the rows and columns of a product code described in Section 4.1 and transmitted on a Gaussian channel using BPSK signalling. On receiving $(n_1 \times n_2)$ matrix \mathbf{R} corresponding to a transmitted $(k_1 \times k_2)$ codeword \mathbf{E} , the second decoder, corresponding to the second encoder C_2 , performs the decoding of the rows using the input matrix but first de-interleaved by the block de-interleaver corresponding to the interleaver used at the transmitter. The output $(n_1 \times k_2)$ matrix

of the second decoder is entered to the first decoder, corresponding to the first encoder C_1 , which performs the decoding of the columns.

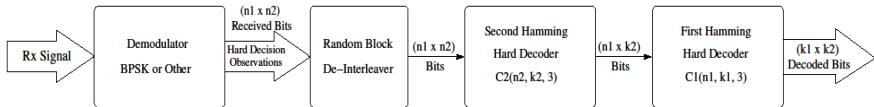


Figure 6: BPTC Hard Decoder

4.3. BPTC coding mechanism analysis

Although the BPTC is composed of Hamming code, its probability of being corrected in the second dimension coding is increased by using the fundamental characteristics of turbo code. So the correction capability will increase normally to more than three-bits error. But we should mention here that it is unfair to compare a simple hamming code with the corresponding product turbo code formed by the concatenation of two from this simple code. For example, we can't compare the performance of the (7,4)-Hamming code with the BPTC code formed from two concatenated (7,4)-Hamming code, because the first one has a coding rate of $4/7$ slightly bigger than half, but the second one has a coding rate of $(4/7)^2$ slightly lower than third. Therefore, in the simulation results presented here we will consider these remark by comparing approximately equally coding rate.

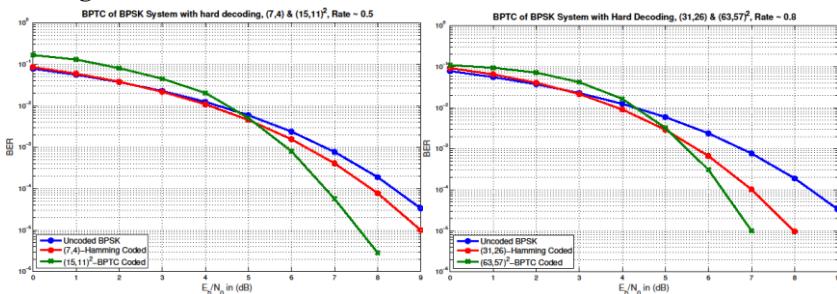


Figure 7: BER of BPTC for different rates with Hard Decoding

Fig. 7 shows the performance of a BPSK system over AWGN channel using a BPTC coding with the conventional hard decoding. We can simply remark that a gain is obtained with

respect to the hard decoding of a simple hamming coding at high SNR (bigger than 5 dB). And the curves of BPTC system are sharper when the codeword length of the coding used increase. Also the different curves in Fig. 7 show that at approximately 6.3 to 6.5 dB we can obtain a $\text{BER} = 10^{-4}$. Finally, these results show that the proposed Soft decoder of a simple Hamming decoder, where we can obtain the 10^{-4} of BER at approximately 6.05 dB (see Tab. 1), can give a better performance than most of the Hard decoded BPTC system, and the gain is around 0.25-0.3 dB (see Fig. 2, 3, and 4).

4.4. BPTC with Soft Decoder

The LLR based decoding procedure described above can be used in the Soft Decoder of the BPTC. The Decoding process is done by cascading the proposed decoders and it is illustrated in Fig. 8. Let us consider the soft decoding of the rows and columns of a product code described in Section A and transmitted on a Gaussian channel using BPSK signalling. On receiving the observations \mathbf{y} corresponding to the message \mathbf{x} transmitted. The LLR calculator compute the $(\mathbf{n}_1 \times \mathbf{n}_2)$ L-values matrix corresponding to these observations, after the block de-interleaver, the second soft decoder performs the decoding of the rows using the input LLR matrix to compute the $(\mathbf{n}_1 \times \mathbf{n}_2)$ L-values output. Only the $(\mathbf{n}_1 \times \mathbf{k}_2)$ portion of the output matrix is taking into account in the first soft decoder which performs the soft decoding of the columns and give at its output $(\mathbf{n}_1 \times \mathbf{k}_2)$ containing the $(\mathbf{k}_1 \times \mathbf{k}_2)$ L-values corresponding to the sent codeword. Finally, a threshold based decision device is needed to obtain the $(\mathbf{k}_1 \times \mathbf{k}_2)$ output decoded bits.

4.5. BPTC (196,96) Example

The 96 messages must be arranged in 11×9 matrix form before BPTC coding, the 3 absent messages are set as 0, placed in the last three positions of the matrix (as shown in Fig. 9), which are R(0), R(1) and R(2), one R(3) is added in the coding process for

replacing the position during interleaving after coding, so it is not placed in the messages to be encoded.

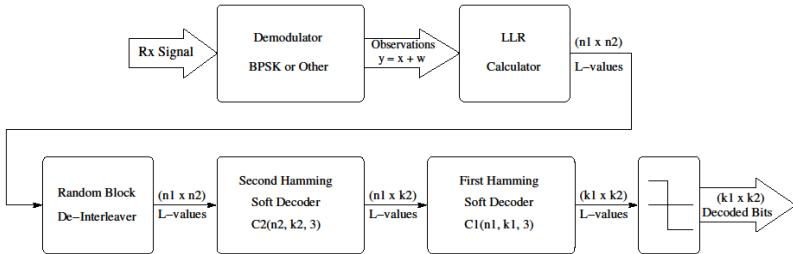


Figure 8: BPTC Soft Decoder

The Hamming (15,11) is encoded according to the above Hamming encode, I(0) to I(10) are a set of messages, $HR_0(0)$, $HR_0(1)$, $HR_0(2)$, $HR_0(3)$, I(11) to I(21) derived from encoding are a set of messages, $HR_1(0)$, $HR_1(1)$, $HR_1(2)$ and $HR_1(3)$ are derived from encoding, till I(88) to R(2) are a set of messages, afterwards, Hamming (15,11) encoding is finished. After Hamming (15,11) encoding, the message array has changed to 15x9 form, and then Hamming (13,9) will be encoded.

Step 3. Hamming (15,11) encoder										Step 1. Message [96]									
$HR_0(3)$	$HR_0(2)$	$HR_0(1)$	$HR_0(0)$	I(10)	I(9)	I(8)	I(7)	I(6)	I(5)	I(4)	I(3)	I(2)	I(1)	I(0)					
$HR_1(3)$	$HR_1(2)$	$HR_1(1)$	$HR_1(0)$	I(21)	I(20)	I(19)	I(18)	I(17)	I(16)	I(15)	I(14)	I(13)	I(12)	I(11)					
$HR_2(3)$	$HR_2(2)$	$HR_2(1)$	$HR_2(0)$	I(32)	I(31)	I(30)	I(29)	I(28)	I(27)	I(26)	I(25)	I(24)	I(23)	I(22)					
$HR_3(3)$	$HR_3(2)$	$HR_3(1)$	$HR_3(0)$	I(43)	I(42)	I(41)	I(40)	I(39)	I(38)	I(37)	I(36)	I(35)	I(34)	I(33)					
$HR_4(3)$	$HR_4(2)$	$HR_4(1)$	$HR_4(0)$	I(54)	I(53)	I(52)	I(51)	I(50)	I(49)	I(48)	I(47)	I(46)	I(45)	I(44)					
$HR_5(3)$	$HR_5(2)$	$HR_5(1)$	$HR_5(0)$	I(65)	I(64)	I(63)	I(62)	I(61)	I(60)	I(59)	I(58)	I(57)	I(56)	I(55)					
$HR_6(3)$	$HR_6(2)$	$HR_6(1)$	$HR_6(0)$	I(76)	I(75)	I(74)	I(73)	I(72)	I(71)	I(70)	I(69)	I(68)	I(67)	I(66)					
$HR_7(3)$	$HR_7(2)$	$HR_7(1)$	$HR_7(0)$	I(87)	I(86)	I(85)	I(84)	I(83)	I(82)	I(81)	I(80)	I(79)	I(78)	I(77)					
$HR_8(3)$	$HR_8(2)$	$HR_8(1)$	$HR_8(0)$	R(2)	R(1)	R(0)	I(95)	I(94)	I(93)	I(92)	I(91)	I(90)	I(89)	I(88)					
$HC_0(0)$	$HC_0(1)$	$HC_0(2)$	$HC_0(3)$	$HC_1(0)$	$HC_1(1)$	$HC_1(2)$	$HC_1(3)$	$HC_0(0)$	$HC_0(1)$	$HC_0(2)$	$HC_0(3)$	$HC_1(0)$	$HC_1(1)$	$HC_1(2)$	$HC_1(3)$	$HC_0(0)$	$HC_0(1)$	$HC_0(2)$	$HC_0(3)$
$HC_1(0)$	$HC_1(1)$	$HC_1(2)$	$HC_1(3)$	$HC_2(0)$	$HC_2(1)$	$HC_2(2)$	$HC_2(3)$	$HC_1(0)$	$HC_1(1)$	$HC_1(2)$	$HC_1(3)$	$HC_2(0)$	$HC_2(1)$	$HC_2(2)$	$HC_2(3)$	$HC_1(0)$	$HC_1(1)$	$HC_1(2)$	$HC_1(3)$
$HC_2(0)$	$HC_2(1)$	$HC_2(2)$	$HC_2(3)$	$HC_3(0)$	$HC_3(1)$	$HC_3(2)$	$HC_3(3)$	$HC_2(0)$	$HC_2(1)$	$HC_2(2)$	$HC_2(3)$	$HC_3(0)$	$HC_3(1)$	$HC_3(2)$	$HC_3(3)$	$HC_2(0)$	$HC_2(1)$	$HC_2(2)$	$HC_2(3)$
$HC_3(0)$	$HC_3(1)$	$HC_3(2)$	$HC_3(3)$	$HC_0(0)$	$HC_0(1)$	$HC_0(2)$	$HC_0(3)$	$HC_3(0)$	$HC_3(1)$	$HC_3(2)$	$HC_3(3)$	$HC_0(0)$	$HC_0(1)$	$HC_0(2)$	$HC_0(3)$	$HC_3(0)$	$HC_3(1)$	$HC_3(2)$	$HC_3(3)$
Step 2. Zero padding [3]																			
Step 4. Hamming (13,9) encoder																			

Figure 9: BPTC (196,96) Matrix

Encode Hamming (13,9), I(0), I(11), I(22),..., I(88) are a set of messages, $HC_0(0)$, $HC_0(1)$, $HC_0(2)$, $HC_0(3)$ are derived from encoding, I(1), I(12), I(23),..., I(89) are a set of messages, $HC_1(0)$, $HC_1(1)$, $HC_1(2)$ and $HC_1(3)$ are derived from encoding, the rest can be deduced accordingly, till $HR_0(3)$, $HR_1(3)$, $HR_2(3)$,..., $HR_8(3)$ are a set of messages, afterwards, Hamming (13,9) encoding is finished. When the row and column encoding

is completed, the matrix has changed to 15×13 form, and then the interleaving is carried out.

The messages received by the receiving terminal are in the form of 15×13 matrix, de-interleaving must be carried out before decoding to restore the original message sequence. Decoding can be carried out after de-interleaving, the decoding should be in such an order that encoded late should be decoded first, that encoded early should be decoded late.

Decode Hamming (13,9) according to the aforesaid Hamming decode, $I(0)$, $I(11)$, $I(22), \dots, HC_0(2)$ and $HC_0(3)$ after sequence restoration are a set of messages, decode and correct $I(0)$, $I(11)$, $I(22), \dots, I(88)$, $I(1)$, $I(12)$, $I(23), \dots, HC_1(2)$ and $HC_1(3)$ are a set of messages, decode and correct $I(1)$, $I(12)$, $I(23), \dots, I(89)$, the rest can be deduced accordingly, till $HR_0(3)$, $HR_1(3)$, $HR_2(3), \dots, HC_{14}(2)$ and $HC_{14}(3)$ are a set of messages, afterwards, Hamming (13, 9) decoding is finished. After Hamming (13,9) decoding, the message array has changed to 15×9 form, and then Hamming (15,11) will be decoded.

Decode Hamming (15,11), $I(0)$, $I(1)$, $I(2), \dots, HR_0(2)$ and $HR_0(3)$ are a set of messages, decode and correct $I(0)$, $I(1), \dots, I(10)$. $I(11)$, $I(12)$, $I(13), \dots, HR_1(2)$ and $HR_1(3)$ are a set of messages, decode and correct $I(11)$, $I(12), \dots, I(21)$, the rest can be deduced accordingly, till $I(88)$ to $HR_8(3)$ are a set of messages, afterwards, Hamming (15, 11) decoding is finished.

5. SIMULATION RESULTS AND ANALYSIS

A key performance index to evaluate the capacity-approaching is the BER given a received SNR over an AWGN channel. We consider a received SNR from 0 to 9 dB and examine the BER, as shown in the performance figures presented previously, Fig. 2, 3, 4, and 7. The received signal to noise ratio is considered here as E_b/N_0 where E_b is the received energy per bit, and N_0 is the noise power spectral density. Monte Carlo simulation by MatLab is used to obtain the results shown in the past and

following figures. For the evaluation, all size of hamming code from code length 7 into 127 was compared together, and the value of coding rate of each product turbo code is considered when compared to a simple hamming code. The code chosen are the (15,11), (31,26), and (63,57)-Hamming code. And the turbo product code chosen are formed by two identical hamming code and they will be compared with the BPTC (196,96) which is formed by the (15,11)-Hamming code concatenated with the (13,9)-reduced Hamming code.

5.1. BPTC vs Simple Hamming code

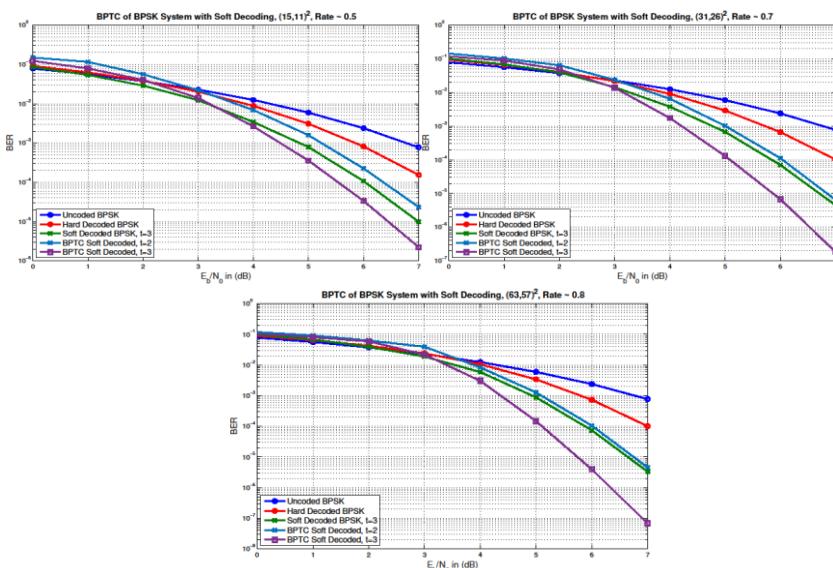


Figure 10: Simple Hamming code vs. BPTC (Identical).

For the (15,11), (31,26), and (63,57)-Hamming codes, which give to each 11, 26, and 57 information bits 4, 5, and 6 additional parity bits, and has a coding rate of approximately 0.7, 0.8, and 0.9 respectively. The $(15,11)^2$, $(31,26)^2$, and $(63,57)^2$ BPTC formed by the concatenation of these two identical hamming will encode each 121, 676, and 3249 information bits to 225, 961, and 3969 coded bits and will have 0.5, 0.7, and 0.8 as code rate, respectively. Fig. 10 shows that the performance of the

BPTC using the soft decoder which corrects the duets bit error or the triplets bit error. the BPTC increases the gain of the simple hamming code soft decoded by respectively 0.5, 0.75, and 0.75 dB and the gain with the hard decoded to 1.5, 2, and 2 dB respectively at BER = 10^{-4} .

5.2. BPTC Comparison

With respect to the performances shown in the last figure, two questions will be responded here, which are: “Which is the better BPTC code to use?”, “Why some standard choose to apply the BPTC (196,96), is it the best BPTC?”.

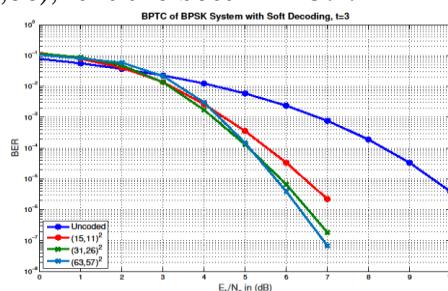


Figure 11: BPTC performances comparison.

Fig. 11 shows a comparison between the three cases presented before, where we remark directly that the $(31,26)^2$ and the $(63,57)^2$ cases show the best performance, where the $(31,26)^2$ BPTC, which has 0.7 as coding rate, slightly outperforms the $(63,57)^2$ BPTC at low SNR (SNR<5 dB) and the $(63,57)^2$ BPTC, with coding rate equal to 0.8, has a very small gain with respect to the $(31,26)^2$ BPTC code. We should not forget the coding rate when we want to compare different schemes of coding. Those two BPTC code, the BPTC(961,676) formed by two identical $(31,26)$ -Hamming code and the BPTC(3969,3249) formed by two identical $(63,57)$ -Hamming code, improve the probability of error, as shown in the Fig. 12, when compared with the BPTC(196,96) which has a coding rate equal to 0.49 and use a concatenation of two different hamming code, the first one the

(15,11)-Hamming code and the second one the reduced (13,9)-Hamming code.

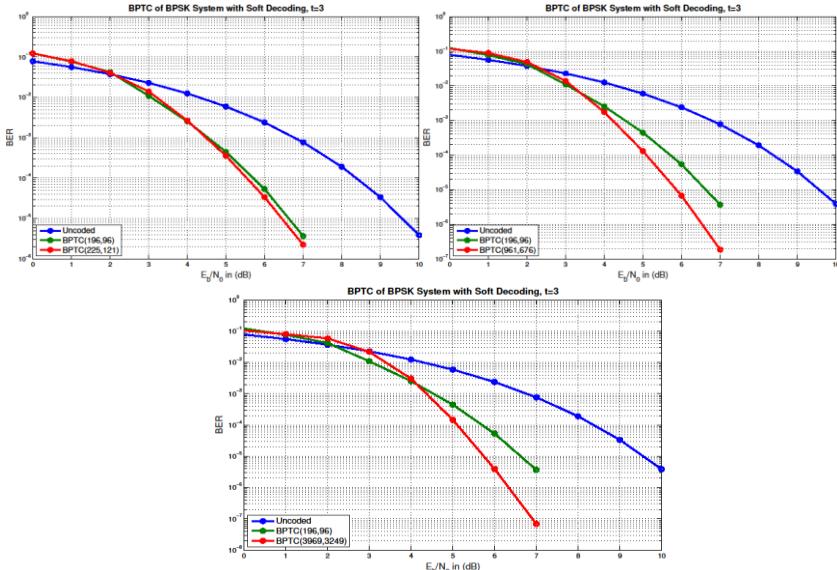


Figure 12: Comparison with the BPTC(196,96).

In the same time, Fig. 12 shows that the BPTC(225,121) formed by two identical (15,11)-Hamming code give approximately the same probability of error as the BPTC(196,96) with a coding rate slightly bigger. As we can see there are other BPTC that outperform the BPTC(196,96) with a coding rate bigger, much bigger. In despite that, the BPTC(196,96) still has the advantage of the block size, which is the smallest possible and these small block size could be necessary for some standards. So we recommend the BPTC(961,676) for standard with a bigger block size.

6. CONCLUSION

In order to observe the efficiency of different combinations of hamming codes to obtain a Block Product Turbo Code (BPTC), we make modularization on purpose, due to this paper uses AWGN channel, one of some editions of BPTC coding, the result

shows an obvious increase in efficiency, if we change the Hamming code used, but there is not one best edition for all the cases.

This paper observes the efficiency of BPTC, and compares the complexity and coding gain with Hamming code, and the physical mechanism of decoding efficiency difference between Hamming code and BPTC is discussed in the paper. This paper uses MatLab code for simulation in order to form a mathematical model of the system. The simulation results show that the soft decoding of the Hamming codes improves well the performance of the coding by more than 1 dB in some case. The BPTC scheme decoded hard doesn't improve the performance with respect to the soft decoding of the simple Hamming code, but when our proposed soft decoder is used in the BPTC scheme, we obtained a gain of approximately 1 dB with respect to the soft decoding of the simple Hamming code. Finally, we have shown a good comparison between different Hamming code size and rate, and we have demonstrated that the (31,26)-Hamming code can be a good compromise when concatenated to give a BPTC(961,676) code when the block size is big. In the other hand, if the block size needed is small, the concatenation of the (15,11)-Hamming code and the (13,9)-Hamming code to give the BPTC(196,96) still the best compromise in despite of the loss in the coding rate.

In the future work, the results of simulations based on the modelling of the system will be compared with the real implementation of the system on a DSP chip to obtain a small software defined radio (SDR) system.

REFERENCES

1. [Forney 1966] Forney, G. D., Jr., Concatenated Codes (Cambridge, MA: MIT Press, 1966).

2. [Berrou 1993] Berrou, C., Glavieux, A., and Thitimajshima, P., "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," IEEE Proceedings of the Int. Conf. on Communications, Geneva, Switzerland, May 1993 (ICC '93), pp. 1064-1070.
3. [Divsalar 2001] Divsalar, D., Dolinar, S., and Pollara, F., "Iterative Turbo Decoder Analysis Based on Density Evolution," IEEE Journal on Selected Areas in Communications, vol. 19, no. 5, pp. 891–907, May 2001.
4. [Divsalar 2004] Divsalar, D., Dolinar, S., "Concatenation of Hamming Codes and Accumulator Codes with High-Order Modulations for High-Speed Decoding," IPN Progress Report 42-156, February 15, 2004.
5. [Li 2004] Li, J, Narayanan, K. R. and Georghiades, C. N., "Product accumulate codes: a class of codes with near-capacity performance and low decoding complexity", IEEE Trans. Inform. Theory, vol. 50, pp. 31-46, 2004.
6. [Huang 2007] Huang, T. D.-H., Chang, C.-Y., Zheng, Y.-X., and Su, Y. T., "Product Codes and Parallel Concatenated Product Codes," IEEE Wireless Communications and Networking Conference, March 2007 (WCNC 2007).
7. [Muller 2011] Muller, B., Holters, M., Zolzer, U., "Low Complexity Soft-Input Soft-Output Hamming Decoder," IEEE proceedings of the 50th FITCE Congress, Palermo, Italy, September 2011 (FITCE 2011), pp. 242-246.
8. [Chen 2011] Chen, Y. H., Hsiao, J. H., Liu, P. F., Lin, K. F., "Simulation and Implementation of BPSK BPTC and MSK Golay Code in DSP Chips", Journal of Communications in Information Science and Management Engineering, CISME Vol. 1, No.4, Pages 46-54, 2011.
9. [Cho 2011] Cho, J., and Sung, W., "Reduced complexity chase-pyndiah decoding algorithm for turbo product codes", Department of Electrical Engineering Seoul National University, 2011.

10. [Megha 2014] Megha, M.S., "Iterative Decoding Algorithm for Turbo Product Codes," International Journal of Innovative Research in Advanced Engineering (IJIRAE) Volume 1 Issue 2, April 2014.