TELKOMNIKA, Vol.16, No.5, October 2018, pp.2040~2047
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v16i5.7463

Reliability Level List Based Iterative SISO Decoding Algorithm for Block Turbo Codes

V Sudharsan, V Vijay Karthik, B Yamuna*
Department of Electronics and Communication Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India
*Corresponding author, e-mail: b_yamuna@cb.amrita.edu

Abstract

An iterative Reliability Level List (RLL) based soft-input soft-output (SISO) decoding algorithm has been proposed for Block Turbo Codes (BTCs). The algorithm ingeniously adapts the RLL based decoding algorithm for the constituent block codes, which is a soft-input hard-output algorithm. The extrinsic information is calculated using the reliability of these hard-output decisions and is passed as soft-input to the iterative turbo decoding process. RLL based decoding of constituent codes estimate the optimal transmitted codeword through a directed minimal search. The proposed RLL based decoder for the constituent code replaces the Chase-2 based constituent decoder in the conventional SISO scheme. Simulation results show that the proposed algorithm has a clear advantage of performance improvement over conventional Chase-2 based SISO decoding scheme with reduced decoding latency at lower noise levels.

Keywords*: error control coding, block turbo code, Iterative SISO decoding, soft decision decoding, reliability level list*

1. Introduction

In communication systems, error control coding techniques are used for reliable and efficient transmission of digital data over noisy channels. P Elias introduced a recursive approach and constructed concatenated codes by combining an inner code with an outer code and achieved exponentially decreasing error performance [1]. Tanner proposed a hard in hard out (HIHO) iterative decoding method for concatenated product block codes and these 'high code rate' concatenated codes found widespread applications in deep space communication systems [2]. Berrou developed a class of Forward Error Correction (FEC) codes called Turbo codes constructed by either serial or parallel concatenation of convolutional codes as constituent codes [3]. These Concatenated Block Codes or Block Turbo Codes (BTCs) proposed by Elias and Convolutional Turbo Codes (CTC) proposed by Berrou revolutionized the field of coding theory since they achieved reliable transmission at a code rate closer to Shannon limit. Since then much research has been done on the decoding algorithms for Turbo codes. Turbo codes are widely used in 3G/4G mobile communication systems, return channel of satellite communication systems and IEEE 802.16 (WiMAX). Decoding algorithms for Turbo codes remains a sought after topic of research in the field of error control coding.

Berrou proposed a soft-input soft-output (SISO) approach for CTC decoding which had a better performance when compared to HIHO decoding algorithm for BTC proposed by Tanner. Ramesh Pyndiah proposed a SISO iterative decoding algorithm for BTC using Chase-2 algorithm [4-5]. It was established that SISO decoding approach to BTC could help achieve a performance similar to CTC with an added advantage of the constituent block decoding not requiring memory. Dave et. al. proposed a decoding algorithm based on Kaneko's algorithm for BTC [6]. Argon et. al. and Al-Dweik et. al. improved the iterative SISO decoding proposed by Pyndiah to reduce the complexity without compromising much on the performance [7-8]. Belkasmi et. al. have proposed a Genetic Algorithm (GA) based iterative decoding for BTC where the decoding of constituent block codes has been approached as an optimization problem [9]. Yuan et. al. had also attempted a GA based decoding for Block Turbo Codes [10]. Askali et. al. have proposed an iterative soft permutation based decoding of product codes [11]. Lu et. al. proposed a syndrome based decoding of BTC and the reduction in complexity of the

Received October 23, 2017; Revised April 11, 2018; Accepted September 6, 2018

same was put forth by Li et. al. [12-13]. The various novel decoding algorithms for BTC proposed in the literature target performance improvement or complexity reduction, trading off one for the other [14-16].

In conventional decoding of BTC, Chase-2 is used as the constituent decoder. The computational complexity of Chase-2 remains the same even at a higher SNR. To overcome the complexity of Chase-2 based SISO decoder at lower noise levels, in this paper, a RLL based decoding algorithm has been proposed for decoding of BTC. Here the constituent codes are decoded using the RLL based decoding algorithm [17]. The proposed algorithm is found to have a better coding gain at lower SNR and has a lower decoding latency at higher SNR when compared to iterative SISO decoding algorithm proposed by Pyndiah, which has been the basis for most of the recent work in the decoding of BTC. The paper is organized as follows: Section 2 discusses about BTC and the traditional SISO iterative decoding. The proposed RLL based decoding algorithm for BTC is explained in detail in Section 3. In Section 4, the simulation results obtained using the proposed algorithm are discussed, followed by conclusion in Section 5.

2. SISO Decoding of Block Turbo Codes

Block Turbo Codes are a class of FEC codes constructed by serial concatenation of Block codes like BCH and RS codes. Consider two systematic BCH codes C_1 (n_1, k_1, d_1) and C_2 (n_2, k_2, d_2) where 'n' is the codeword length, 'k' is the message length and 'd' is the minimum hamming distance. The Block Turbo Code is constructed by encoding the 2-D message block with an encoding scheme C_1. The encoded block is passed to an interleaver and the interleaved code block is subsequently encoded using C_2.the second encoding scheme. Thus, for a message block of dimension k_1 x k_2, the turbo code T=$C_1 \otimes C_2$ can be obtained by:

a. Encoding k_1 rows using the coding scheme C_2.
b. Encoding k_2 columns using the coding scheme C_1.

The resultant BTC can be represented as $T\ (n, k, d)$ where $n = n_1 \times n_2$, $k = k_1 \times k_2$, $d = d_1 \times d_2$. Consider a 2-D message block of bits {0,1} encoded into a codeword block using encoding scheme 'T', modulated using BPSK and transmitted over an Additive White Gaussian Noise (AWGN) channel with mean zero and variance $N_o/2$. The received soft value matrix from the demodulator is given as in equation (1) below:

$$R = C + N \tag{1}$$

where R = {r_{11},r_{12},…r_{1n1}; r_{21},r_{22},...;….r_{n1n2}}. The turbo decoding of BTC involves two main steps namely decoding of constituent Block Code and iterative turbo decoding process.

2.1. Decoding of constituent Block Code

Each row or column in the 2-D received matrix is decoded using the corresponding elementary decoder. The traditional SISO decoding proposed by Pyndiah uses Chase-2 algorithm. The Chase-2 algorithm [5] involves the following steps:

a. Identify the $p = \lfloor d_{min}/2 \rfloor$ least reliable positions (LRPs) using the received soft decision sequence.
b. Generate 2^p error patterns 'e_i' at the LRPs and obtain 2^p distorted sequences 'Z_i' using the hard decision sequence 'y'.

$$Z_i = e_i \oplus y_i \tag{2}$$

c. Decode each of the 2^p distorted sequences using an algebraic decoder or a hard decision decoder and add the decoded codeword to the candidate set Ω.
d. Euclidean distance of each candidate codeword from the original received soft decision sequence is calculated and the codeword with closest Euclidean metric is taken as the decision codeword D.

2.2. Iterative SISO Turbo decoding

In the iterative decoding process, the decoding of the received soft decision matrix is done over a fixed number of iterations. Each iteration consists of two half iterations namely row

decoding and column decoding. An extrinsic information is calculated based on the decision D from the elementary Chase decoder and is passed from one half iteration to the consecutive half iteration. The iterative process follows the steps as given below:

a. The reliability values of the received soft decision sequence is defined in terms of its Log Likelihood Ratio (LLR) as given by equation (3).

$$\Lambda(y_i) = \ln\left(\frac{P_r\{e_j = +1/r_j\}}{P_r\{e_j = -1/r_j\}}\right) = \left(\frac{2}{\sigma^2}\right) r_j. \quad (3)$$

b. In each half iteration, each row/column in the received matrix is passed to the elementary constituent decoder and decision D is obtained. Decision D {0,1} is mapped to {-1, +1}.
c. The soft-output of the current half iteration is computed as in equation (4).

$$r_j' = \left(\frac{|R-C|^2 - |R-D|^2}{4}\right) d_j \quad (4)$$

Here 'C' is the competing codeword in the set Ω which has the second least Euclidean distance to R. In certain cases, the complexity of finding the competing codeword increases exponentially with respect to 'p'. In such cases, soft-output is calculated as in equation (5) given below.

$$r_j' = \beta \times d_j \quad (5)$$

where 'β' is a reliability scaling factor.

d. The extrinsic information for the next half iteration is obtained as in equation (6).

$$w(m+1) = r_j'(m) - r_j(m) \quad (6)$$

e. The extrinsic information is then added to the soft-input of the next half iteration as in equation (7).

$$R(m) = R + \alpha(m) \times w(m) \quad (7)$$

where 'α' is the weighting factor that controls the effect of extrinsic information at the earlier iterations when BER is high. The process is now repeated for the subsequent iterations until the predefined number of iterations is reached. The output from the final iteration is taken as the optimal estimate of transmitted code block.

3. The proposed RLL Based Iterative Decoding Algorithm for Block Turbo Codes

The main components of BTC based decoding scheme are an elementary decoder for each of the constituent block code and iterative SISO turbo decoding, where exchange of extrinsic information between iterations take place. There has been much work in the literature on complexity reduction in iterative SISO decoding and extrinsic information calculations, with Chase-2 algorithm as the decoder for constituent codes. The fact that the computational complexity of Chase-2 decoding algorithm remains the same even at a lower noise level has lead researchers to come up with modifications/alternate solutions to Chase-2. One such algorithm is Reliability Level List based SDD Algorithm [14].

RLL based decoding is a scheme of soft decision decoding based on an efficient and meaningful strategy of exploiting the reliability information available from the demodulator. The algorithm is based on the RLL which forms the framework for identifying the codeword through a directed minimal search. Using this framework of RLL, we propose an iterative algorithm for decoding BTC. For each row/column decoding, the RLL gives the most reliable valid codeword which forms the decision D. The reliability of the decision D is then calculated using equation (4), where the competing codeword 'C' is the second most reliable valid codeword in the RLL. This soft-output of the current iteration is passed as soft-input to the next half iteration. The proposed algorithm gives a better performance at lower SNR and lower decoding latency at higher SNR when compared to iterative Chase-2 based SISO decoding algorithm for BTC.

The two main steps of the proposed algorithm are RLL generation to find decision D, followed by iterative SISO decoding.

3.1. RLL generation for decoding constituent Block Code

Based on the reliability values of the 'n' bit soft decision sequence, a rank 'v' is assigned to all possible codewords of the constituent block code in a structured manner. This is called generation of RLL. Each entry in the RLL generated is given by a set of parameters (v_i,p_i,q_i,m_i) where

i=0 to 2^{n-1}.

v-rank of the entry.

p-set of indices which form the least reliable bit positions.

q-codeword obtained by flipping the hard decision sequence at positions given by p_i.

m-absolute sum of reliability values at positions given by p_i.

The procedure for RLL generation is as follows:

a. By default, first 3 entries of the RLL are hard decision sequence (q_0, v=1, p_0={ }, m=0), codewords obtained by flipping the least reliable position(q_1, v=2, p_1={'1'}, m=r_j; j-1st LRP) and second least reliable position(q_2, v=3, p_1={'2'}, m=r_j; j-2nd LRP) respectively in the hard decision sequence.
b. A pending list 'P' is formed which contains all possible contestants for the next entry to the list.
c. Further the pending list is updated based on the current entry 'v_i' under consideration. Updation of pending list is done as follows:
   i. If the least index in 'p_i' is not '1', add '1' to the index set and add it to the pending list.
   ii. For all non-zero indices in the set 'p_i', add 1 to the index and check if it is less than the next non-zero element in the index set 'q' and add the index set to the pending list.
   iii. For the last non-zero index in the set 'p_i', add 1 to it and check if it is less than the codeword length 'n' and add the entry to the pending list.
   iv. If the index set generated by any of the above rules is already a member in the pending list, remove it from the list.
d. The entry 'q_i' with least value of 'm' from the pending list forms the next entry to the RLL. Once a member from the pending list is chosen to be the next entry, it is checked if it is a valid codeword.
e. If the current entry 'q_i' is a valid codeword, then the decoding is terminated and 'q_i' is taken as the decision D. Else the pending list is updated based on the current entry 'p_i' and the process is repeated until a valid codeword from RLL constituent decoder is obtained.

3.2. RLL based iterative SISO decoder for BTC

With D as the output of the RLL constituent decoder, the process of RLL search is continued to find the next most probable codeword which is the competing codeword C. This is the second most reliable valid codeword in the RLL. The soft-output is then calculated as a function of the competing codeword C as in equation (4) which is then passed as extrinsic information to the next constituent decoder. The soft-input for the next half iteration is then calculated as in equation (6) and equation (7). The process is repeated for the fixed number of iterations and decision 'D' of the final iteration gives the optimal estimate.

In Chase decoders, the complexity of finding the competing codeword increases if the competing codeword is not in the candidate codeword set Ω. This is because the scanning radius has to be extended beyond (d_{min}/2), resulting in more candidate codewords in the set Ω, with a corresponding increase in number of HDD. In the proposed algorithm, since HDD is not involved, the complexity associated with the same is avoided.

The proposed RLL based iterative SISO decoding algorithm for BTC is given below:

1. The reliability value R of the matrix is defined in terms of it's LLR value as

$$\Lambda(y_i) = \ln\left(\frac{P_r\{e_j = +1/r_j\}}{P_r\{e_j = -1/r_j\}}\right) = \left(\frac{2}{\sigma^2}\right) r_j.$$

2. For each row in R:
   Do
   2.1. Index the bit positions in increasing order of magnitudes from 1 to n.

2.2. Define pending list with parameters (v_i, pi, q_i, m_i) where i = 0 to 2^{n-1}.
2.3. Include the hard decision sequence (q_0, v=1, p_0 = {}, m=0), codewords obtained by flipping the least reliable position (q_1, v=2, p_1={'1'}, m=r_1) and second reliable position(q_2, v=3, p_1={'2'}, m=r_2) of the hard decision sequence as the first 3 entries to the pending list.
2.4. For each entry (v_i,p_i,q_i,m_i) in RLL:
2.4.1. Do
If (q_i is a valid codeword) :
q_i = decoded codeword
jump to 3
Else: Remove the current 'p_i' from RLL and Update the pending list
If ('1' ∉ p_i) :
add '1' to indices set
add the new entry to pending list
End
For each non-zero element 'b_j' in p_i:
If ((b_j+1) < b_{j+1} && (b_j+1) ≤ n):
add (b_j+1) to indices set
add the new entry to pending list
End
End
End
2.4.2. For each newly added candidate in the pending list, calculate 'm_i' by adding the soft values in the indices mentioned in 'p_i'.
2.4.3. Choose the candidate with least value of 'm_i' as next entry to the RLL.
3. Map the decoded codeword q_i {0,1} to D {-1, +1}.
4. Calculate the soft-output of the decoder using the formula

$$r_j' = \left(\frac{|R-C|^2 - |R-D|^2}{4}\right) d_j$$

where 'C' is the competing codeword. In practical cases, where the complexity of finding 'C' increases, soft-output is calculated as

$$r_j' = \beta \times d_j$$

where β is reliability scaling factor [5].
5. Calculate the extrinsic information as

$$w(m+1) = r_j'(m) - r_j(m)$$

6. Calculate soft-input to the next half iteration as

$$R(m) = R + \alpha(m) \times w(m)$$

where α is weighting factor.
7. Repeat Steps 2-6 for the column decoding.
8. Repeat Steps 2-7 for each iteration until we reach the fixed number of iterations.
The Decision 'D' of the final half-iteration gives the optimal estimate of the transmitted BTC.

4. Results and Discussion

The performance of the proposed algorithm is evaluated in an AWGN channel under BPSK modulation. The results obtained using this algorithm for BTC $(15,7,5)^2$, BTC $(31,21,5)^2$ and BTC (31,21,5)x(31,26,3) are compared against the traditional iterative SISO decoding proposed by Pyndiah. All simulations have been carried out using Matlab. A 4-iteration simulation has been shown for all the codes considered. The weighting factor α is taken as α(m)

= [0 0.2 0.4 0.5 0.7 0.9 1.0 1.0] and reliability scaling factor is taken to be β(m) = [0.2 0.4 0.6 0.8 1.0 1.0 1.0 1.0] [5]. The weighting and scaling factors are chosen to increase with the iteration number as the decision D obtained from a particular iteration is more reliable than the previous iteration.

In Figure 1, the improvement of the proposed algorithm over iterations is compared against conventional SISO decoding algorithm for a BTC $(15,7,5)^2$. At a BER of $2x10^{-4}$, the proposed RLL based SISO decoding algorithm is found to have a coding gain of 1 dB over the Chase-2 based conventional SISO decoding algorithm proposed by Pyndiah using Chase-2 algorithm (Ch-Py). With the proposed algorithm, the decoding latency reduces at higher SNR. The decoding latency of RLL-SISO decoding algorithm and Chase-2 based conventional SISO at SNR 1 dB and SNR 5 dB are shown in Table 1.

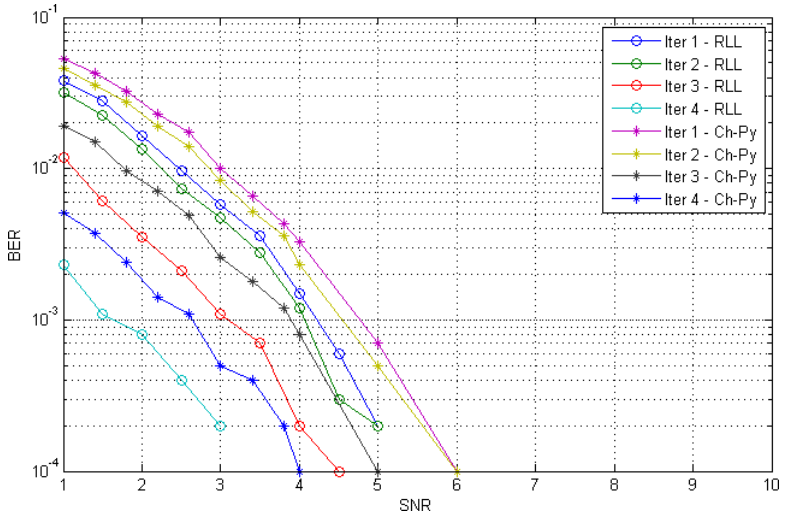

Figure 1. Performance comparison of RLL-SISO decoding algorithm with Conventional SISO for BTC $(15,7,5)^2$

Table 1. Decoding latency of the proposed algorithm for decoding one frame of BTC $(15,7,5)^2$ against conventional SISO algorithm

Decoding algorithm	SNR = 1 dB	SNR = 5 dB
RLL SISO	4.6924 sec	0.0971 sec
Conventional SISO	1.07 sec	1.09 sec

The performance comparison of the proposed algorithm for BTC $(31,21,5)^2$ has been shown in Figure 2. It can be seen that RLL based SISO decoding algorithm has a coding gain of 0.7 dB over conventional algorithm at the end of iteration 4. The proposed algorithm works well for asymmetric codes as well and it has been applied to an asymmetric BTC (31,21,5)x (31,26,3) and the improvement over iterations has been shown in Figure 3.

In traditional SDD algorithms, the search space is generally bounded to a limit. But the RLL algorithm is not a bounded search algorithm and the search space is the entire 2^n codewords. The algorithm is applicable to BTCs of any dimension and adapts well for mobile applications where the coding scheme changes often. Most of the SDD algorithms in literature do not make use of the high-SNR environment in which transmission takes place and have a fixed computations at all SNRs. RLL based iterative algorithm has a faster decoding convergence for high-SNR applications where it is likely to find the optimal codeword in the first few entries itself thus reducing the complexity drastically.

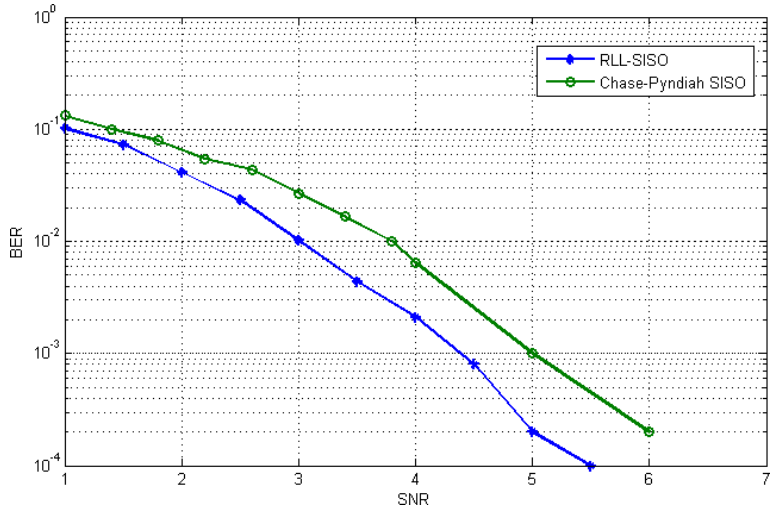

Figure 2. Performance of RLL-SISO algorithm for BTC $(31,21,5)^2$ at 4^{th} iteration

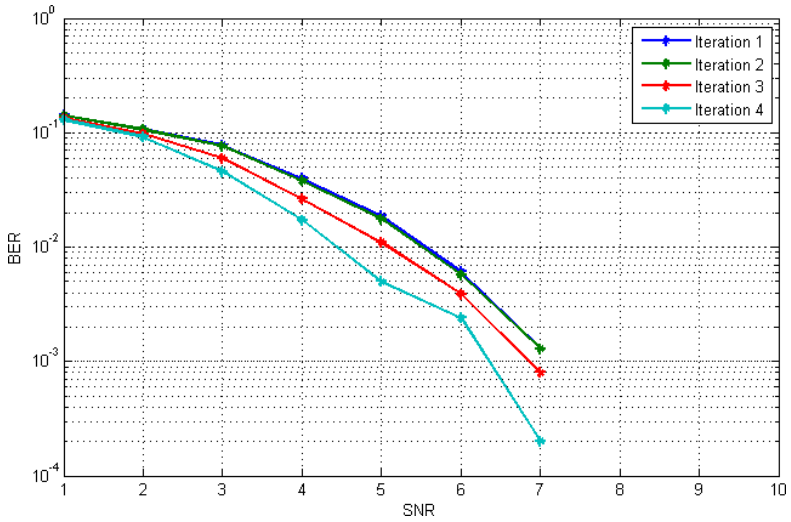

Figure 3. Performance improvement of RLL-SISO algorithm over iterations for BTC (31,21,5) x (31,26,3)

5. Conclusion

The proposed algorithm gives a better performance for BTC decoding and has very low complexity at high SNR applications. This algorithm makes maximum use of the channel reliability information and does a structured search over the entire 2^n codeword space thus giving the most reliable codeword as the decoded codeword. The algorithm is generalized for any Turbo code constructed using cyclic block codes as constituent codes. Much of the SISO decoders for BTC revolve around LRP based Chase-2 decoders and thus the proposed RLL algorithm widens the scope for research in alternate SISO decoders for BTC. Further the algorithm can extended to BTC with non-binary cyclic codes as constituent codes [18].

References

[1] Elias P. Error-free Coding. *Transactions of the IRE Professional Group on Information Theory*. 1954; 4(4): 29-37.
[2] Tanner R. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*. 1981; 27(5): 533-547.

[3] Berrou C, Glavieux A, Thitimajshima P. *Near Shannon limit error-correcting coding and decoding: Turbo-codes*. IEEE Int Conf Communications. 1993; 23: 1064-1071.
[4] Pyndiah R. Near-optimum decoding of product codes: block turbo codes. *IEEE Transactions on Communications*. 1998; 46(8): 1003-1010.
[5] Chase D. Class of algorithms for decoding block codes with channel measurement information. *IEEE Transactions on Information Theory*. 1972; 18(1): 170-182.
[6] Dave S, Junghwan Kim, Kwatra S. An efficient decoding algorithm for block turbo codes. *IEEE Transactions on Communications*. 2001; 49(1): 41-46.
[7] Argon C, McLaughlin S. An Efficient Chase Decoder for Turbo Product Codes. *IEEE Transactions on Communications*. 2004; 52(6): 896-898.
[8] Al-Dweik A, Goff S, Sharif B. A Hybrid Decoder for Block Turbo Codes. *IEEE Transactions on Communications*. 2009; 57(5): 1229-1232.
[9] Belkasmi M, Berbia H, Bouanani F. 7th International ITG Conference on Source and Channel Coding (SCC). 2008: 1-6.
[10] Y Yuan J, Tian Y, Hu X, Huang S, Lin J, Pang Y. A novel BTC decoding algorithm based on the genetic algorithm in optical communication systems. *Optoelectronics Letters*. 2014; 10(2): 123-125.
[11] Askali M, Ayoub F, Chana I, Belkasmi M. Iterative Soft Permutation Decoding of Product Codes. *Computer and Information Science*. 2016; 9(1): 128.
[12] Lu E, Lu P. *A syndrome-based hybrid decoder for turbo product codes*. Tainan: International Symposium on Computer Communication Control and Automation (3CA). 2010: 280-282.
[13] Li L, Zhang F, Zheng P, Yang Z. *Improvements for decoding algorithm of turbo product code*. IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). 2014: 374-378.
[14] Kumar M, Saxena J. Performance Comparison of Latency for RSC-RSC and RS-RSC Concatenated Codes. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*. 2013; 1(3).
[15] Salim M, Yadav R, Narwal K, Sharma A. A New Block S-Random Interleaver for Shorter Length Frames for Turbo Codes. *Bulletin of Electrical Engineering and Informatics*. 2013; 2(4).
[16] Wang J, Li J, Cai C. Two Novel Decoding Algorithms for Turbo Codes Based on Taylor Series in 3GPP LTE System. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(5).
[17] Yamuna B, Padmanabhan TR. A Reliability Level List based SDD Algorithm for Binary Cyclic Block Codes. *International Journal of Computers Communications & Control*. 2012; 7(2): 388-395.
[18] Yamuna B, Padmanabhan TR. A minimal search soft decision list decoding algorithm for Reed-Solomon codes. *International Journal of Information and Communication Technology*. 2014; 6(1): 71-85.