Turbo Like Multi-Stage Threshold Decoding for Self-Orthogonal Convolutional Codes

Muhammad Ahsan Ullah
Department of Electrical and Electronic Engineering
Chittagong University of Engineering and Technology
Chittagong, 4349, Bangladesh
Email: ahsan@cuet.ac.bd

Haruo Ogiwara
Nagaoka University of Technology
1603-1, Kamitomioka, Nagoka, Niigata, 940-2188, Japan
Email: ogiwara@vos.nagaokaut.ac.jp

***Abstract*—This paper presents a new dimension of soft decoding turbo-like multi-stage threshold decoding (TLMTD) for self-orthogonal convolutional codes (SOCCs). The TLMTD uses comparatively shorter constraint length conventional code of multi-stage threshold decoding (MTD) system. The bit error performance is considered for several types of soft decoding algorithms on the additive white Gaussian noise (AWGN) channel. When threshold value used as a priori threshold value for other decoding stage, TLMTD realizes better performance in waterfall and error floor regions. Moreover, the TLMTD gives 0.20 dB more coding gain compared to MTD for equivalent SOCCs at the bit error rate less than 10^{-4}.**

***Index Terms*—rthogonal convolutional code, threshold decoding, turbo code, LDPC coderthogonal convolutional code, threshold decoding, turbo code, LDPC codeo**

I. Introduction

As a low complex decoding method, a threshold decoding (TD) is a first choice in the field of error correcting codes. The threshold decoding algorithm and threshold decodable codes have been presented in [1]. The TD experiences catastrophic errors with some channel error patterns for convolutional codes [2]. The self-orthogonal convolutional codes (SOCCs) give limited error propagation with TD and prevent the catastrophic error flow [3], [4], [5]. In actual case, TD with SOCC makes unavoidable error group in the decoding information bit stream.

The multi-stage threshold decoding with difference register (MTD) for a special SOCC can break down the error group and produces maximum likelihood decoding (MLD) performance in the error floor region [6],[7]. The special kind of SOCC consists accessibly large constraint length that can useful for limited number of system.

This paper proposed ordinary SOCC with a pseudo random interleaver in encoding and in decoding side. This decoding method is called turbo-like multistage threshold decoding with difference register (TLMTD). Similar decoding scenario is presented in [8], [9] which are soft decoding with a priori based algorithm. This paper proposed a combined decoding scheme of weighted bit flipping algorithm which is widely used in low density parity check code [10] and soft decoding MTD in [6]. Hence, the TLMTD uses ordinary SOCC, the constraint length of SOCC contains several bits to several hundred of bits. Thats why, the TLMTD can be useful in wide range with MLD performance.

Rest of the paper is arranged as follows. Section II shows the multi-stage threshold decoding system. Section III gives the concept of TLMTDs as well as the SOCCs. Section IV gives several soft decoding algorithms for TLMTD and provides their bit error performance. Section V gives soft decoding TLMTD serially concatenated with parity check code that improves the bit error rate in the error floor region. Section VI gives the decoding performance of various TLMTD system. Section VII concludes this paper.

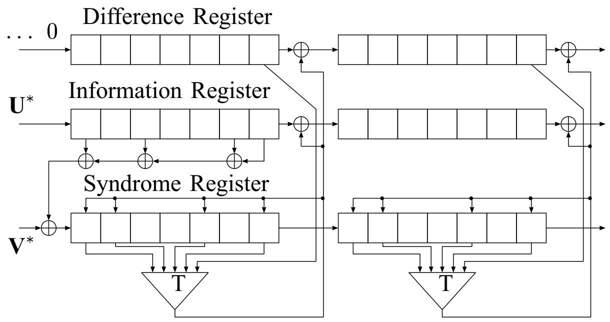

Fig. 1. MTD for SOCC with $J = 4, K = 7$.

II. Multi-Stage Threshold Decoding with Difference Register

A self-orthogonal convolutional code (SOCC) with rate $R = 1/2$, memory length K and tap weight J is considered. We shall consider binary signal and tail bitting termination is used for convolutional code in this paper. Figure 1 shows a multi-stage threshold decoder with $K = 7$ and $J = 4$. A tap connection set $A = \{a_m\},\ m = 1, 2, \ldots, J,\ a_1 < a_2 < \ldots < a_J$, is defined by the code. The encoder generates a parity bit v_i by a set of information bits selecting from an information bit stream $\{u_i\},\ (i = 0, 1, \ldots)$. The i'th parity bit of a given SOCC is defined by

$$v_i = \sum_{m=1}^{J}\!\!\oplus\, u_{i-a_m} \tag{1}$$

where $\sum\!\oplus$ represents the variable-sized modulo-2 sum operator in this paper. The encoder makes a codeword c $\triangleq \{u_0, v_0, u_1, v_1, \ldots\}$. The codeword is converted into a binary phase shift keying (BPSK) signal and transmitted through the AWGN channel. The binary error sequence e $\triangleq \{e_0, \xi_0, e_1, \xi_1, \ldots\}$, e_i and ξ_i are the i'th error bits add with the information and parity bits, respectively, and forms a hard decision received word c* $\triangleq \{u_0^*, v_0^*, u_1^*, v_1^*, \ldots\}$. Receiver first separates the information part U* and parity part V* from the received word. This received information and parity bits generate a syndrome sequence S $\triangleq \{s_0, s_1, \ldots\}$. The i'th syndrome bit is calculated by

$$s_i = v_i^* \oplus \sum_{k=1}^{J}\!\!\oplus\, u_{i-a_k}^* = \xi_i \oplus \sum_{k=1}^{J}\!\!\oplus\, e_{i-a_k} \tag{2}$$

978-1-4673-1436-7/12/$31.00 ©2012 IEEE

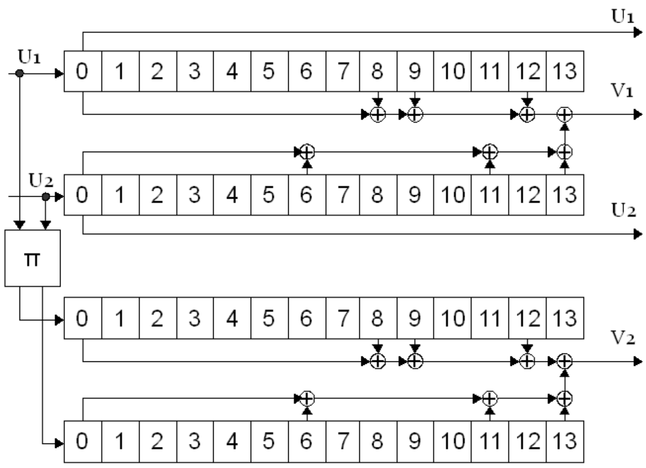

Fig. 2. Turbo like threshold encoder for SOCC with $J = 4,\ K = 14$.

where $\oplus$ represents the modulo-2 sum operator in this paper. For decoding each received information bit, a set of syndrome bits, collected by the tap connection in syndrome register, are necessary. The collected set of syndrome bits, with J elements, is called checking syndrome ant its summation is defined as checksum value. The j'th checksum value for decoding j'th information bit is defined by

$$L_j = \sum_{k=1}^{J} s_{j+a_k} + d_j \tag{3}$$

where d_j is the j'th bit of difference register. At the initial stage of MTD, DR contains all zero signal. The threshold value of a given code be $T = \lfloor \frac{J+1}{2} \rfloor$, where the mark $\lfloor x \rfloor$ represents the largest integer not greater than x. When L_j exceeds the threshold value ($L_j > T$), the decoding decision flips the j'th information bit and associated DR bit and the checking syndromes.

III. Turbo-Like Multi-Stage Threshold Decoding

The main idea of turbo code is to encode the sequence of information bits twice, using two encoders interconnected with an interleaver. In general TLMTD encoder generate m parity bit streams by using $n,\ n \geq 2$, information bit streams. In this case $m * n$ tap connection sets are necessary. For maintaining total code rate $R = 1/2$, m will be just half of n. Figure 2 shows an turbo like encoder with $J = 4$, $K = 14$, $n = 2$ and $m = 1$ of a systematic self-orthogonal convolutional code. The turbo-like encoder generates a codeword with information bit streams once and all parity bit streams. Two tap connection sets generate a parity bit stream by using two information bit sequences $\mathbf{U}_1 \triangleq \{u_0^{(1)}, u_1^{(1)}, \ldots\}$ and $\mathbf{U}_2 \triangleq \{u_0^{(2)}, u_1^{(2)}, \ldots\}$. A tap connection set for TLMTD encode is defined as $A_{xy} \triangleq \{a_1^{(xy)}, a_2^{(xy)}, \ldots, a_{J_{xy}}^{(xy)}\}$, where $x = 1, 2, \ldots, n$ and $y = 1, 2, \ldots, m$. The i'th parity bit of component encoder one and two are defined by (4) and (5), respectively.

$$v_i^{(1)} = \bigoplus_{k=1}^{J_{11}} u_{i-a_k^{(11)}}^{(1)} \oplus \bigoplus_{k=1}^{J_{21}} u_{i-a_k^{(21)}}^{(2)} \tag{4}$$

$$v_i^{(2)} = \bigoplus_{k=1}^{J_{11}} u_{i-a_k^{(11)}}^{'(1)} \oplus \bigoplus_{k=1}^{J_{21}} u_{i-a_k^{(21)}}^{'(2)} \tag{5}$$

where $u_i^{'(x)}$ is the i'th information bit of the x'th information bit stream of interleaved version of information bit streams. The information and parity bit streams make a codeword $\mathbf{c} \triangleq \{u_i^{(1)}, v_i^{(1)}, u_i^{(2)}, v_i^{(2)}\}, i = 0, 1, \ldots$ that is transmitted through the AWGN channel as BPSK signals.

At the receiver end, hard decision received word splits into information bit streams and parity bit streams. The encoder in decoder side generates syndrome bit streams ($\mathbf{S} \triangleq \{s_0, s_1, \ldots\}$ and $\mathbf{S}' \triangleq \{s_0', s_1', \ldots\}$) for component decoder one and two, respectively. The i'th syndrome bit of component decoder 1 and component decoder 2 are given by (6) and (7), respectively.

$$s_i = \xi_i^{(1)} \oplus \bigoplus_{k=1}^{J_{11}} e_{i-a_k^{(11)}}^{(1)} \oplus \bigoplus_{k=1}^{J_{21}} e_{i-a_k^{(21)}}^{(2)} \tag{6}$$

$$s_i' = \xi_i^{(2)} \oplus \bigoplus_{k=1}^{J_{11}} e_{i-a_k^{(11)}}^{'(1)} \oplus \bigoplus_{k=1}^{J_{21}} e_{i-a_k^{(21)}}^{'(2)} \tag{7}$$

where $e_i^{(x)}$ and $\xi_i^{(x)}$ are the i'th channel errors in $\mathbf{U}_x$ and $\mathbf{V}_x$, respectively and e' and ξ' are the interleaved version of errors in the information and parity bit sequences, respectively. For decoding j'th information bit by first component decoder, the checksum value associated of x'th information bit stream is $L_j^{(x)}$ can be calculated by

$$L_j^{(1)} = \sum_{k=1}^{J_{11}} s_{j+a_k^{(11)}} + d_j^{(1)} \tag{8}$$

$$L_j^{(2)} = \sum_{k=1}^{J_{21}} s_{j+a_k^{(21)}} + d_j^{(2)} \tag{9}$$

and the second component decoder calculated them by

$$L_j^{(1)} = \sum_{k=1}^{J_{11}} s_{j+a_k^{(11)}}' + d_j^{'(1)} \tag{10}$$

$$L_j^{(2)} = \sum_{k=1}^{J_{21}} s_{j+a_k^{(21)}}' + d_j^{'(2)} \tag{11}$$

where $d_j^{(x)}$ and $d_j^{'(x)}$ are the j'th bit in the x'th DR and its interleaved version, respectively. The threshold values of TLMTD for hard decoding be $T_1 = \lfloor \frac{J_{11}+1}{2} \rfloor$ and $T_2 = \lfloor \frac{J_{21}+1}{2} \rfloor$ for decoded information bit stream one and two, respectively. When the checksum value $L_j^{(x)}$ exceeds the threshold value T_x, hard decoding decision flips the target information bit, associated DR bit and checking syndromes. After several iteration, final output is done.

IV. Soft Decoding TLMTD

A. Soft TLMTD

An iterative decoding scheme improves the decoding performance using reliability information delivered by one decoder and passing it into the another component decoder as a priori information in addition to the soft output of the channel. Figure 3 shows the turbo-like MTD in where at the first iteration a priori value set to all zero and the extrinsic value is updated bit by bit. After interleaving that soft output of first component decoder, it uses as a priori value in the second component decoder. The soft output value of second decoder de-interleaved and use as a priory value for the first component decoder and these operation is done up to predefine number of iterations.

Soft input threshold decoding is known as a posteriori probability decoding [?]. Massey's algorithm gives checksum value of j'th decoded bit by

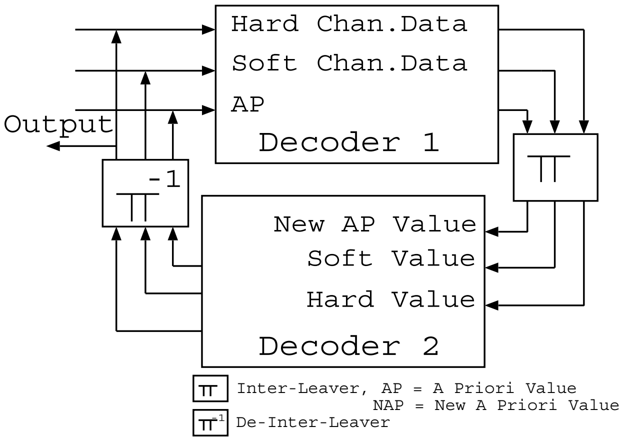

Fig. 3. A schematic diagram of turbo like threshold decoder.

$$L_j = 2\sum_{k=1}^{J} ln(q_{j+a_k}/p_{j+a_k})s_{j+a_k} \quad (12)$$

and the threshold value is calculated by

$$T_j = \sum_{k=1}^{J} ln(q_{j+a_k}/p_{j+a_k}) + ln(q_j/p_j) \quad (13)$$

where $p_{j+a_k} = 1 - q_{j+a_k}$ is the channel error probability of parity signal related to the syndrome s_{j+a_k} and $p_j = 1 - q_j$ is the channel error probability of j'th information bit. When L_j exceeds the threshold value T_j, decoding decision is flipped.

In this paper we calculate the checksum value directly from the soft received word and its hard decision syndrome values. For decoding j'th information bit by the soft TLMTD (STLMTD), the j'th checksum value L_j is calculated by

$$L_j = \sum_{k=1}^{J} w_{j,k}(1 - 2s_{j+a_k}) + w_{dj}(1 - 2d_j) + Ap_j \quad (14)$$

where $w_{j,k}$ is an absolute value of k'th received parity signal regarding to the syndrome bit s_{j+a_k}. The value w_{dj} is an absolute value of j'th received information signal. The value Ap_j is the priori value associated to the j'th decoded information bit. When $L_j < 0$, the flipping decision f_j be '1' otherwise f_j be '0'. The j'th extrinsic value ξ_j is given by

$$\xi_j = (1 - 2f_j).\sum_{k=1}^{J} w_{j,k}(1 - 2s_{j+a_k}) \quad (15)$$

A priori value for second component decoder is the interleaved extrinsic value of the first component decoder. Similarly, a priori value of the first component decoder, from second iteration, is the de-interleaved extrinsic value of the second component decoder. After flipping each information bit, the cross-correlation increases between the received word and the decoded codeword, generated from the decoded information bits. Then the Euclidian distance reduces between them. In this case, we approximate the priori value, an interleaved version of extrinsic value of the first component decoder, in component decoder 2 shares only the information bit under decoding by the component decoder 2. This approximation is valid because, the tap weight of the encoder is negligibly smaller than the information length and the pseudo random interleaver is used. The cross-correlation value regarding to the focussing information between the received word and the decoded codeword is calculated by (14). Therefore, flipping decision turns the L_j value from negative to positive and increases the cross-correlation value and then reduces the Euclidian distance between them.

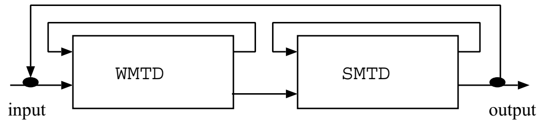

Fig. 4. Schematic diagram of combined soft decoding multi-stage threshold decoding.

B. Weighted Bit Flipping TLMTD

The weighted bit flipping (WBF) algorithm has been proposed for decoding LDPC codes [10]. This paper formulates the WBF algorithm for TLMTD called weighted bit flipping TLMTD (WBFTLMTD). The weighting value $w_{j,k}$ of WBFTLMTD is the minimum absolute value among the received signals associated of the syndrome bit s_{j+a_k}. Then the weighting value $w_{j,k}$ is put into (14) and the decoding decision is done accordingly.

C. Combined Soft Decoding TLMTD

The combined soft decoding TLMTD is formed by the serial concatenation of WBFTLMTD and STLMTD. Figure 4 shows the structure of CTLMTD with feedback (CTLMTDF). Each section of CTLMTDF terminates its decoding when no flipping decision is committed in every decoding section.

From Fig. 4, the CTLMTDF is constructed by the following switching steps:

- Step 1: Switch S_1 is closed and the decoding operation of WBFTLMTD starts. The WBFTLMTD terminates by the termination conditions or committed decoding up to setting iterations (here maximum 2 iterations be set and then switch S_1 opened.
- Step 2: Switch S_4 and S_2 are closed together and the STLMTD begins its decoding. STLMTD terminates decoding after by maximum two iterations and the switch S_4 and S_2 turn into the open state.
- Step 3: The decoded signals from STLMTD are feedback to the WBFTLMTD by closing switch S_3. Step 1 and step 2 are continued maximum 10 times and final output take place.

V. CTLMTDs with Parity Check Code

A parity check code is serially concatenated with CTLMTDF. Before TLMTD encoding, the information bit streams are segmented by predefine size. Each segment of information bit stream makes an information sub-block. The parity checking encoder generates a parity check codes against each information sub-block and then the TLMTD encoder generates a codeword. The received signals are decoded by the CTLMTDF decoder first and then the parity check is applied. When, a decoded information sub-block does not satisfy the parity check, the parity check decoder finds minimum value of checksum L_i regarding to the decoded information sub-block and then i'th information bit of that sub-block flips and final output is done. In this paper we use the information sub-block length 50 bits.

VI. Performance of TLMTDs

This section gives the error performance of TLMTD and MTD with code rate $R = 2/4$ for self-orthogonal convolutional codes. TLMTD uses pseudo random interleaver with the length equal to total information length $N = 2100$. The code length of both

TABLE I
TAP CONNECTION SETS OF SOCCS FOR: MTDS AND TLMTDS

$\mathbf{A}_{11}$	{0, 51, 198, 251, 465}	Code with
$\mathbf{A}_{12}$	{ 23, 187, 247, 370, 371}	
$\mathbf{A}_{21}$	{40, 76, 176, 200, 259}	$R = 2/4$
$\mathbf{A}_{22}$	{161, 230, 281, 328, 483}	
$\mathbf{A}_{11}$	{0, 46, 194, 464, 499}	Component Code
$\mathbf{A}_{21}$	{0, 127, 187, 470, 498}	with $R_c = 2/3$

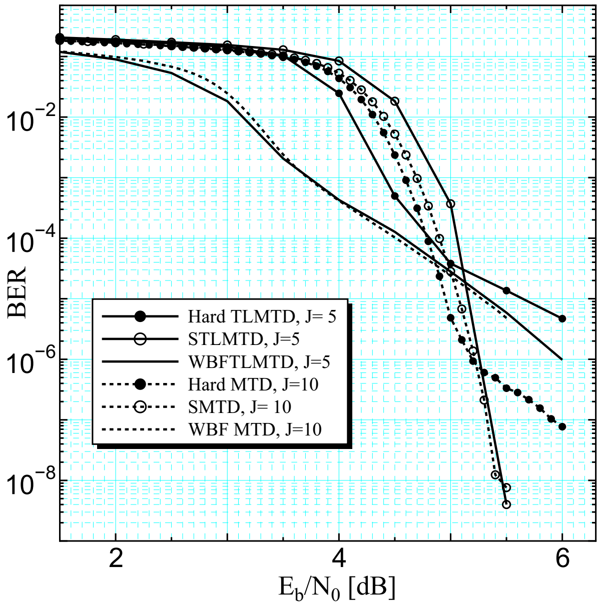

Fig. 5. BER performances of MTD and TLMTD for SOCCs with $K = 500$, $N = 2100$, $J = 10$ for MTD and $J = 5$ for TLMTD

cases are 4200. Table I gives the SOCCs for MTDs and TLMTDs. The code rate of component encoder of TLMTD is $R_c = 2/3$.

Figure 5 shows the bit error performance of hard, soft and weighted bit flipping TLMTDs and MTDs for the codes given in Table I. Hard TLMTD gives better error rate than the hard decoding MTD at lower E_b/N_0 region. On the other hand, the error flooring property of MTD is better. STLMTD achieves good error flooring property than the MTD at high E_b/N_0 region though MTD is a little better at waterfall region. Moreover, WBFTLMD is more effective than WBFMTD at lower E_b/N_0 region.

Figure 6 illustrates the bit error performance of combined soft decoding TLMTD and MTD with feedback. This figure also shows the error performance of parity check codes [6] concatenation with CTLMTDF and combined MTD with feedback (CMTDF) proposed in [6]. The CTLMTDF gives 0.35dB more coding gain than the CMTDF with parity check code at the BER 10^{-5}. Error flooring properties of both systems are the same. The error flooring property of CTLMTDF with PC approaches to the error flooring property of CMTDF with PC at high E_b/N_0 region. The CTLMTD achieves 6.8dB coding gain over the AWGN channel at the BER 10^{-5}.

VII. CONCLUSION

Previously the MTD was a good choice for low complex decoding technique. Now TLMTD may be another candidate in that field. The new soft decoding algorithm with Ap value gives

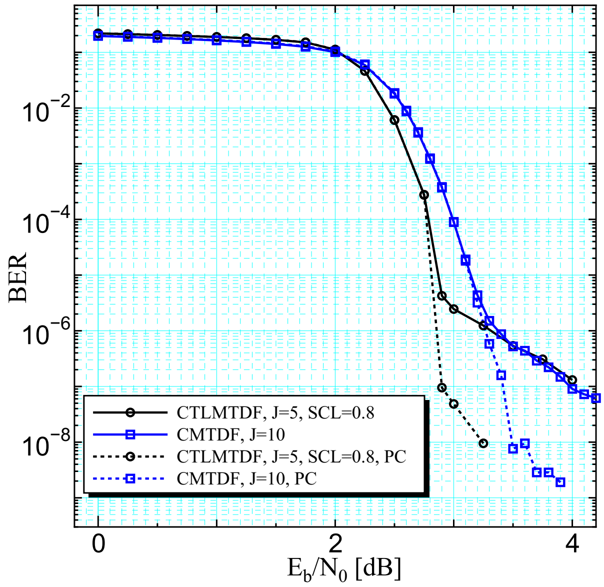

Fig. 6. BER performances of MTD and TLMTD for SOCCs with $K = 500$, $N = 2100$, $J = 10$ for MTD and $J = 5$ for TLMTD

better error performance than soft decoding MTD. We know that when the orthogonal checking number in encoder increases, the error performance of MTD as well as TLMTD improves at higher E_b/N_0 region. The TLMTD gives better error performance even the orthogonal checking number of 1/2 of the MTD. Moreover, constraint length of the SOCC for TLMTD is adjustable for shorter to longer. Therefore, we have to investigate the code with various orthogonal checking numbers and optimize on it.

REFERENCES

[1] J. Massey, Threshold Decoding, M.I.T Press, 1963.

[2] J. Massey, "Catastrophic error-propagation in convolutional codes," Circuit Theory -11th Midwest Symposium, University of Notre Dame, pp.583–587, 1968.

[3] J. P. Robinson, "Error propagation and definite decoding of convolutional codes," IEEE Trans. on Inform. Theory, Vol. IT-14, No.1, pp.121–128, Jan. 1968.

[4] S. Lin and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall,Inc. Englewood Cliffs, N.J, chap.13, 1983.

[5] X. H. Peng and P. G. Farrell, "Threshold Decoding and a New Class of Threshold Decodable Codes", IEE Proc.-Commun., Vol. 145, No.5, pp-289-296, 1998.

[6] Muhammad Ahsan Ullah, Kazuma Okada, and Haruo Ogiwara, "Multi-stage threshold decoding for self-orthogoanl convolutional codes," IEICE Trans. Fundamentals, Vol. E93–A, No. 11, pp. 1932–1941, Nov. 2010.

[7] Muhammad Ahsan Ullah and Haruo Ogiwara, "Performance improvement of multi-stage threshold decoding with difference register," *IEICE Trans. Fundamentals*, Vol. E94–A, No. 6, pp. 1449–1457, Jun. 2011.

[8] S.J. Johnson, Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes, Cambridge University Press, First Ed., 2010.

[9] Sven Riedel and Yuri V. Svirid, " Interative (Turbo) decoding of threshold decodable codes," Europian Tran. on Telecommunications, vol.6, no.5, pp.527–534, Sep.-Oct. 1995.

[10] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," IEEE Trans. on Inform. Theory, Vol.47, pp.2711–2736, Nov. 2001.