

工學碩士 學位論文

광대역 무선 통신을 위한 고속 적응형
터보 복호기 FPGA 설계

**An FPGA Design of High-Speed Adaptive Turbo Decoder
for Broadband Wireless Communications**

指導教授 鄭智元

2006年 8月

韓國海洋大學校 大學院

電波工學科

崔 德 君

本 論 文 을 崔 德 君 의 工 學 碩 士 學 位 論 文 으 로 認 准 함 .

委 員 長 : 工 學 博 士 金 基 萬 (인)

委 員 : 工 學 博 士 尹 榮 (인)

委 員 : 工 學 博 士 鄭 智 元 (인)

2006年 8月

韓 國 海 洋 大 學 校 大 學 院

電 波 工 學 科

崔 德 君

INDEX

Abstract	ii
Nomenclature	iv
Chapter I. Introduction	1
Chapter II. Adaptive Turbo Decoding Algorithm	4
2.1 Mapping of bits to signal.....	7
2.2 Coset Symbol Transformer(CST).....	8
2.3 Phase Sector Quantizer(PSQ).....	10
2.4 Simulation Results.....	13
Chapter III. High Speed Turbo Decoder Algorithm	15
3.1 Radix-4 Algorithm.....	16
3.2 Dual-Path Processing Algorithm.....	18
3.3 Parallel Decoding Algorithm.....	21
3.4 Early Stop Algorithm.....	22
3.5 Simulation Results	23
Chapter IV. Design of the Adaptive High-Speed Turbo Decoder...	24
4.1 The Adaptive High-Speed Turbo Decoder Structure.....	25
4.2 The Optimum Quantized Bits of the Adaptive Turbo Decoder ...	28
4.3 FPGA Implementation	29
Chapter V. Conclusion	33
References	35

Abstract

This thesis proposes an adaptive turbo decoding algorithm for high order modulation scheme combined with original design for a standard rate-1/2 turbo decoder for B/QPSK modulation. A transformation applied to the incoming I-channel and Q-channel symbols allows the use of an off-the-shelf B/QPSK turbo decoder without any modifications. Adaptive turbo decoder processes the received symbols recursively to improve the performance. As the number of iterations increases, the execution time and power consumption also increase as well. To reduce the latency and power consumption, this thesis employs the combination of the radix-4, dual-path processing, parallel decoding, and early-stop algorithms. This thesis implemented the proposed scheme on a field-programmable gate array (FPGA) and compared its decoding speed with that of a conventional decoder. From the result of implementation, it was found that the decoding speed of proposed adaptive decoding is faster than that of conventional scheme by 6.4 times under the following conditions : $N=212$, iteration=3, 8-states, 3 iterations, and 8PSK modulation scheme .

Nomenclature

- d_k : The information bit at time k
- c_k : The coded bit at time k
- u : The uncoded bit
- N : The interleaver size
- L_c : The reliability value of the channel
- α_k^m : The forward state metric at time k and state $m(m=0,1,2,\dots,7)$
- β_k^m : The backward state metric at time k and state $m(m=0,1,2,\dots,7)$
- δ_k^m : The branch metric at time k and state $m(m=0,1,2,\dots,7)$

Chapter I. Introduction

Iterative decoding based on symbol-by-symbol soft-in/soft-out decoding algorithm has significant attention, due to its near Shannon-limit error performance[1]. As a powerful coding technique, a turbo code offers a great promise for improving the reliability of communication systems such as those based on the Digital Video Broadcasting standard for Return Channel via Satellite(DVB-RCS)[2]. However, digital transmission via satellite can be severely affected by rain-induced signal fade. Rain-fade countermeasure has been one important design objective of any satellite communication systems, especially those offering broadband multimedia services. There are various schemes to deal with this issue, such as up-link power control, adaptive modulation and transmission, and adaptive channel coding, etc. Among them, the adaptive channel coding receives much attention recently, and it is a most powerful scheme to warrant the high reliability and high spectral efficiency over the rain-fade channel[3]. The adaptive channel coding scheme uses different channel coding techniques, depending on weather conditions. For example, under clear sky, a more spectrum-efficient modulation and coding such as an 8-PSK or 16-QAM turbo trellis-coded modulation can be

used to provide a higher data rate; while under heavy rain condition, a QPSK with a half rate turbo code can be employed to maintain an acceptable performance.

This thesis presents a single turbo decoder need to decode all modulation schemes in order to need less hardware and less power consumption, and to reduce the receiver cost. A novel decoding procedure allows the use of an off-the-shelf turbo decoder originally designed for a standard rate-1/2 turbo decoder over B/QPSK modulation, to decode some trellis coded modulation schemes over 2^m -PSK/QAM constellations, with $m \geq 3$.

this thesis presents a new method for an adaptive turbo decoding algorithm using the coset symbol transformation where received 8-PSK signals are transformed to QPSK signals after some manipulations. The novel adaptive turbo codes with two coded bits per symbol is based on a realization of rate $n/(n+1)$ trellis coded scheme using an off-the-self turbo decoder, originally design for a standard rate-1/2 turbo decoder for B/QPSK modulation.

Important issues in high-speed applications of turbo decoders are decoding delay and computational complexity. . Like maximum a posterior probability (MAP) decoding, iterative decoder processes the received symbols recursively to improve the reliability of each symbol based on constrains that

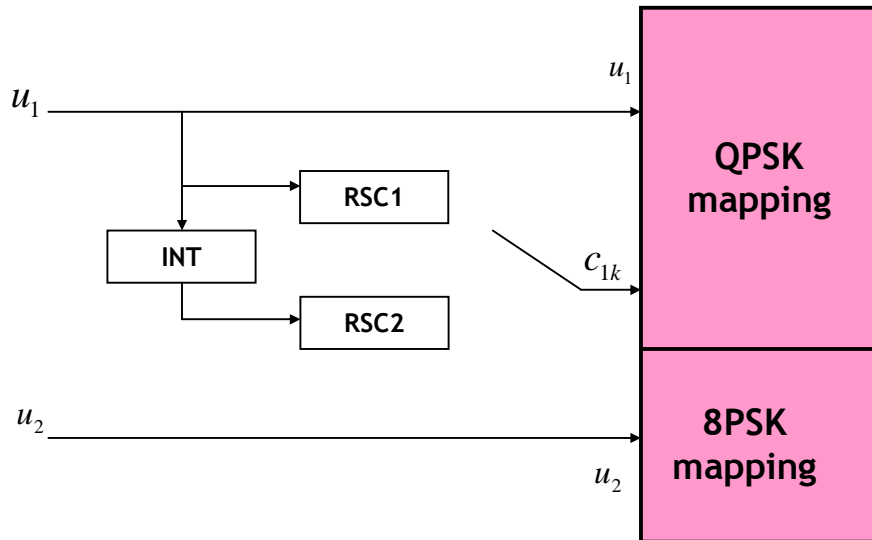
specify the code. In the first iteration, the decoder only uses the channel output, and generates soft output for each symbol. The output reliability measures of the decoded symbols at the end of each decoding iterations are used as input for next iteration. Therefore, the latency and complexity caused by several iterations and high computation order, it can be difficult to implement the decoding in hardware and to apply the high-speed wireless applications. To solve the latency problem, in this thesis four decoding algorithms are presented and combined into one decoder architecture such as the radix-4, dual-path processing, parallel decoding, and early-stop algorithms.

Chapter II. Adaptive Turbo Decoding

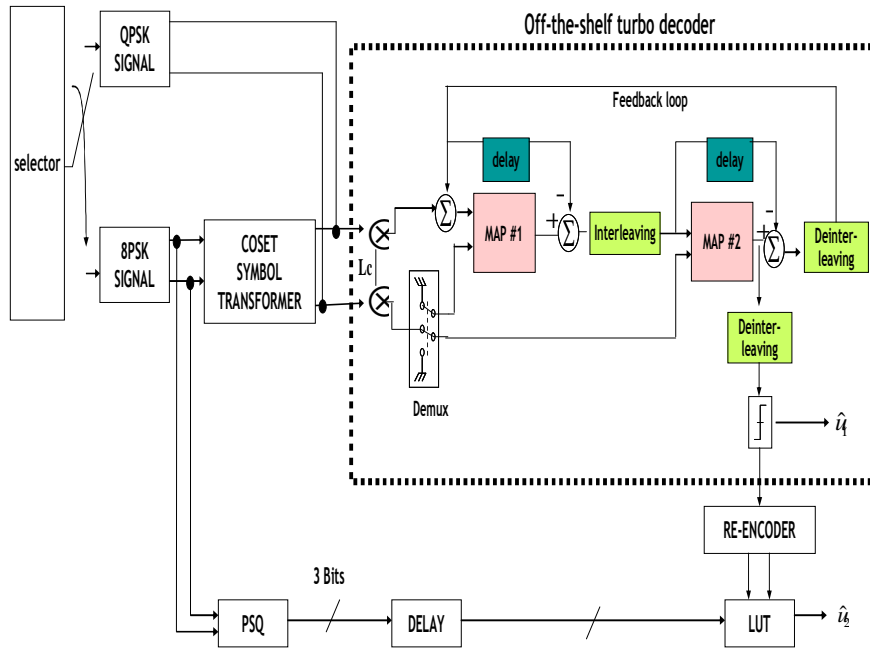
Algorithm

Trellis coded modulation(TCM) is now a well-established method in a digital communication systems capable of achieving coding gain within 3 to 6 dB range of the Shannon channel capacity for a trellis coded 8-PSK system compared to an uncoded QPSK system. The application of turbo codes to TCM has received much attention in the literature. Hauro et al. have proposed a new turbo trellis coded modulation scheme[4]. As noted above, unlike in, this thesis has chosen to combine turbo codes with a pragmatic concept. Turbo-coded pragmatic TCM(TCPTCM) is chosen as an adaptive turbo code. In this section, a TCPTCM with a rate-2/3 is presented. For the sake of clarity, only the case of 8-PSK modulation is considered, but it is easily generalized to M-PSK for M a power of 2. Fig. 2.1 shows the adaptive turbo encoder/decoder structure with a rate of 2/3 TCPTCM. Encoder consists of two RSC (Recursive Systematic Convolutional) codes, and interleaver (INT). Decoder consists of an off-the shelf turbo decoder (DEC1, DEC2) with rate 1/2, a phase sector quantizer(PSQ), coset symbol transformer(CST), and a re-encoder(RE). As an adaptive concept, this structure may be easily expanded to high-order modulation schemes. The decoding

procedures require a standard turbo decoder without any modification in calculating the log-likelihood ratio, forward/backward state metric, and branch metric.



(a) Encoder Structure



(b) Decoder Structure

Fig. 2.1 Rate-2/3 turbo-coded pragmatic TCM encoder/decoder

2.1 Mapping of bits to signal

Two information bits (u_1, u_2) are encoded to produce three coded bits (u_2, u_1, c_{1k}) , which are mapped onto 8-PSK signal points, where c_{1k} is the output of the standard turbo encoder and is punctured. The signal points are labeled by a triplet (u_2, u_1, c_{1k}) , as shown in Fig. 2.2.

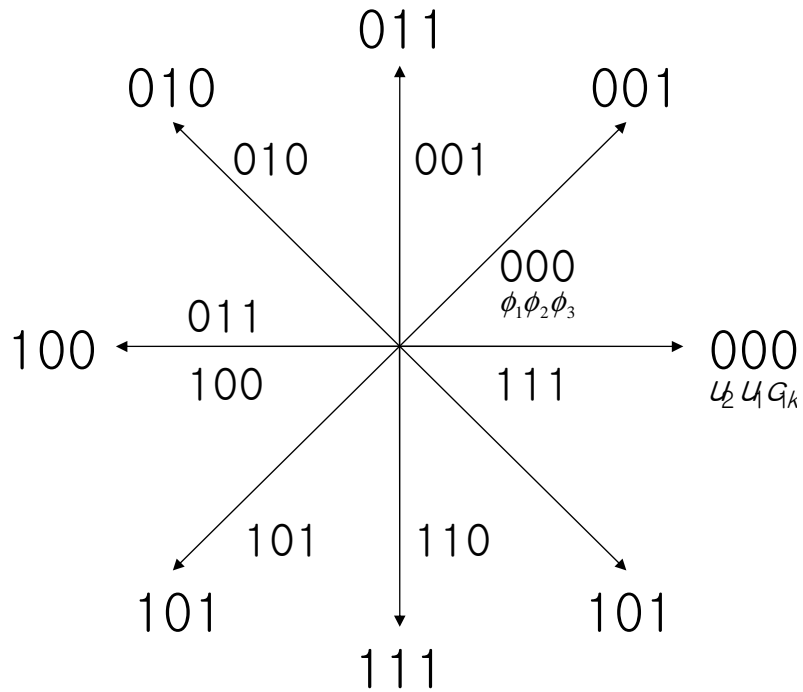


Fig. 2.2 Labels and phase information of 8-PSK constellation

2.2 Coset Symbol Transformer(CST)

Let (x, y) denote the I- and Q- channel values of the received 8-PSK symbols $r(t)$. Let φ denote the phase of the received signal.

$$r(t) = x + jy \quad (2-1)$$

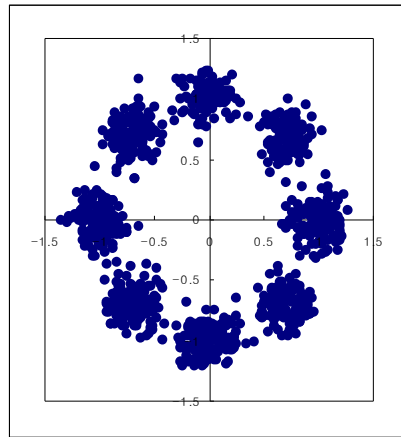
$$\varphi = \tan^{-1}(y/x) \quad (2-2)$$

In order to use the turbo decoder with rate-1/2, a transformation is applied such that the 8-PSK points are mapped into QPSK points labeled by (u_1, c_{1k}) , as shown in Fig. 2.2. The x' and y' projections in the transformed QPSK constellation are obtained from the received 8-PSK symbols by Equation (2-3).

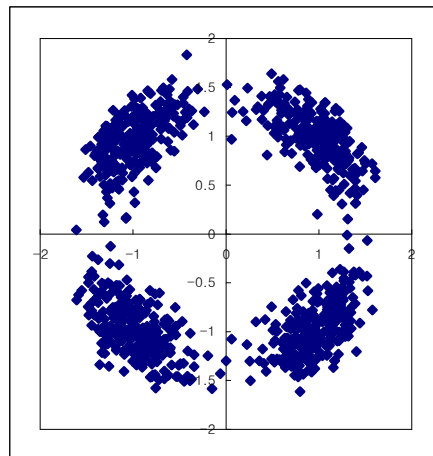
$$\begin{aligned} x' &= \sqrt{2} \cos(2(\varphi + 5\pi/8)) \\ y' &= \sqrt{2} \sin(2(\varphi + 5\pi/8)) \end{aligned} \quad (2-3)$$

$\sqrt{2}$ is the scaling factor to project onto QPSK points with $(\pm 1, \pm 1)$, $5\pi/8$ is the phase rotation constant to map into the QPSK point with (u_1, c_{1k}) : $(11) \rightarrow \pi/4$, $(01) \rightarrow 3\pi/4$, $(00) \rightarrow 5\pi/4$, and $(10) \rightarrow 7\pi/4$. Fig. 2.3

shows the received 8-PSK symbols transformed into QPSK symbols at E_b/N_0 of 12dB.



(a) Received 8-PSK symbols



(b) Transformed into QPSK symbols

Fig. 2.3 Transformed QPSK symbols (E_b/N_0 of 12dB)

2.3 Phase Sector Quantizer(PSQ)

The signal vector space is quantized to determine the locations of received symbols, and PSQ in Fig. 2.4 is used for decoding the uncoded bits, u_2 . The PSQ is designed with the assumption that the in-phase(I) and quadrature(Q) components of the received signals $r(t)$, will be converted to q quantization bits. The circuit is shown in Fig. 2.4. Three comparators and two absolute generators produce the three bits of phase information indicating one of the sectors. Each of the three phase information bits gives information about the location of the received vector: ϕ_2 and ϕ_3 indicate the quadrant and the remaining one bit indicates the location within the quadrant. When $|I| < |Q|$, ϕ_1 is one. In Fig. 2.2, the numbers in 8-PSK constellation are denoted, ϕ_1, ϕ_2, ϕ_3 .

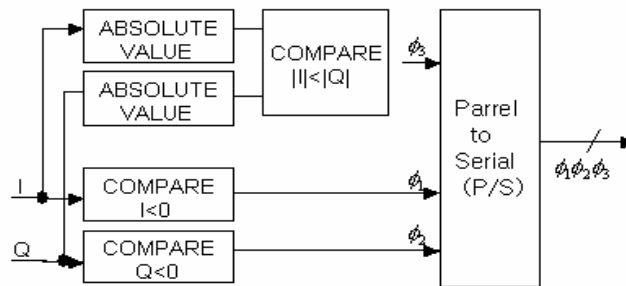


Fig. 2.4 Phase sector quantizer and Soft decision mapping block

CST outputs, (x', y') , are used by an iterative MAP turbo decoder to produce estimates, \tilde{u}_1 , of u_1 . The value \tilde{u}_1 is then encoded by the rate 1/2 turbo code encoder providing estimates, \tilde{u}_1 and \tilde{c}_{1k} of u_1 and c_{1k} . After the turbo decoder estimates the coded bits, it remains to determine the uncoded bit. This is accomplished by making a threshold decision. Using the estimated code bits, $(\tilde{u}_1, \tilde{c}_{1k})$, and phase information, (ϕ_1, ϕ_2, ϕ_3) , this thesis has to determine the uncoded bit, \tilde{u}_2 of u_2 . Due to the structure of the turbo decoding algorithm, every decoder delays the data by a fixed number of symbol periods. The phase information bits ϕ_1, ϕ_2, ϕ_3 must be delayed by the amount of turbo decoding delay to match with the reconstructed code sequence. A simple look-up table(LUT) as shown in Table 2.1 can be used to estimate \tilde{u}_2 . As an example, if re-encoded bits $(\tilde{u}_1, \tilde{c}_{1k})$ are (00) and the phase information (ϕ_1, ϕ_2, ϕ_3) is (111), then $\tilde{u}_2 = 0$.

Table 2.1 Look-up table for estimating \tilde{u}_2

estimated coded bits $\begin{pmatrix} \tilde{u}_1 & \tilde{c}_{1k} \\ \tilde{u}_1 & \tilde{c}_{1k} \end{pmatrix}$	ϕ_1, ϕ_2, ϕ_3	\tilde{u}_2
00	(000),(001),(110),(111)	0
	(010),(011),(100),(101)	1
01	(000),(001),(010),(111)	0
	(011),(100),(101),(110)	1
10	(000),(101),(110),(111)	0
	(001),(010),(011),(100)	1
11	(000),(001),(010),(011)	0
	(100),(101),(110),(111)	1

2.4 Simulation Results

The computer simulations to compare the performance of a turbo-coded pragmatic TCM(TCPTCM) is compared to pragmatic TCM(PTCM) and published simulation results of the turbo-coded TCM method in [4]. TCM method of Hauro *et al.* in [4]. The simulation results are plotted (Fig. 2.5) for comparable 2-bps/Hz systems: 1) uncoded QPSK; 2) rate-2/3 pragmatic TCM(PTCM) with a single 64-state convolutional encoder; 3) rate 2/3 turbo-coded pragmatic TCM(TCPTCM, in Fig. 2.1 with two 16-state RSC encoders, a 500-bit random interleaver, and five decoding iterations; and 4) rate-2/3 turbo TCM with two 16-state RSC encoders, 500-bit random interleaver, and five decoding iterations[4]. At a BER of 10^{-5} , TCPTCM achieves a 2 dB gain relative to PTCM. Compared to [4], the proposed decoder exhibits a small loss of less than 0.2 dB.

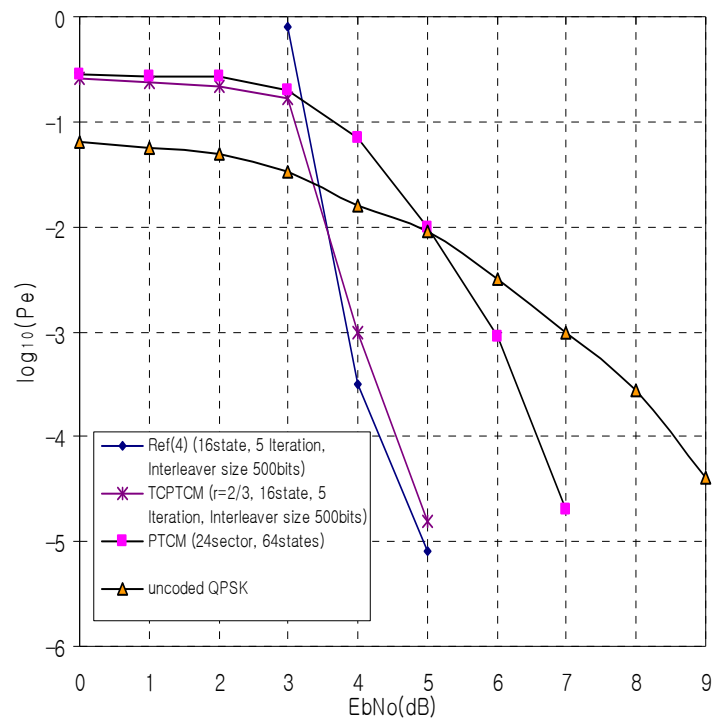


Fig. 2.5 Bit error rate performance comparison between pragmatic TCM(PTCM), turbo-coded pragmatic(TCPTCM), and turbo TCM(TTCM)

Chapter III. High Speed Turbo Decoder

Algorithm

Since convolutional turbo codes are very flexible codes, easily adaptable to a large rate of block sizes and coding rates, they have been adopted in the DVB standard for Return Channel via Satellite(DVB-RCS). The use of RCST(RCS Terminal) includes individual and collective installation(e.g. SMATV) in domestic environment. However, the applications of turbo codes are limited to specific data such as low data-rate services because of their limit of decoding speed. Therefore, it is highly required to develop the high-speed turbo decoder. To solve the problem with latency of turbo decoder, four kinds of algorithms are introduced. The first algorithm is radix-4 algorithm and the second algorithm is the dual-path processing algorithm. The third algorithm is the full parallel decoding algorithm. The fourth algorithm is the early-stop algorithm based on hard-decision-aided (HDA) scheme. The decoding iteration processes until a certain stopping condition is satisfied. Then hard decisions are made based on the reliability measures of the decoded symbol at the last decoding iteration.

3.1 Radix-4 Algorithm

The first algorithm is the radix-4 decoding algorithm, where the previous state at $t = k - 2$ goes forward to the current state at $t = k$, and the reverse state at $t = k + 2$ goes backwards from the current one such that the time interval from $t = k - 2$ to $t = k$ is merged into $t = k$. Therefore, this thesis can decode two source data bits at the same time without any performance degradation while reducing the block size buffered in memory. Using the unified approach to state metrics, a 2^{v-1} -state trellis can be iterated from time index $n-k$ to n by decomposing the trellis into 2^{v-k} sub-trellises, each consisting of k iterations of a 2^k -state trellis. Each 2^k -state's sub-trellis can be collapsed into an equivalent one-stage radix- 2^k trellis by applying k levels of look-ahead to the recursive update. Collapsing the trellis does not affect the decoder performance since there is a one-to-one mapping between the collapsed trellis and radix-2 trellis. An example of the decomposition of a 4-state radix-2 into an equivalent radix-4 trellis using one stage of look-ahead is shown in Fig. 3.1, where $v=4$, $g_1=(15)_{octal}$, $g_2=(17)_{octal}$ with v denoting the constraint length.

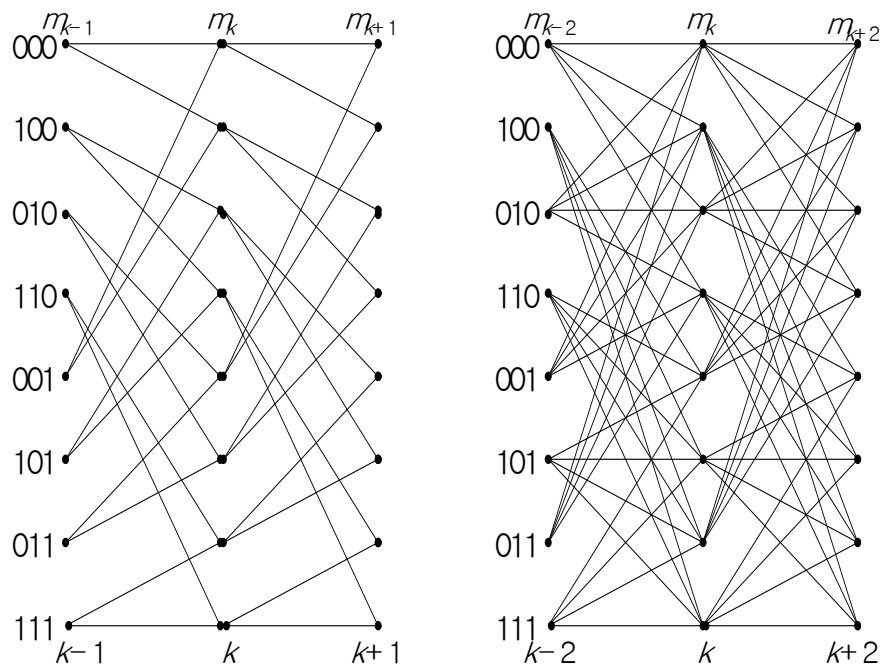


Fig. 3.1 Eight-state radix-2 to radix-4 trellis.

3.2 Dual-Path Processing Algorithm

In a conventional scheme, the decoder must wait for finishing the backward state metric (BSM) (or forward state metric (FSM)) calculations before calculating the extrinsic information. The dual-path processing method doesn't need to wait. The decoder calculates the FSM (left to right) and BSM (right to left), simultaneously. When the FSM and BSM reach the same point, then the decoder begins to calculate the extrinsic information. Fig. 3.2 shows the operation of dual-path processing.

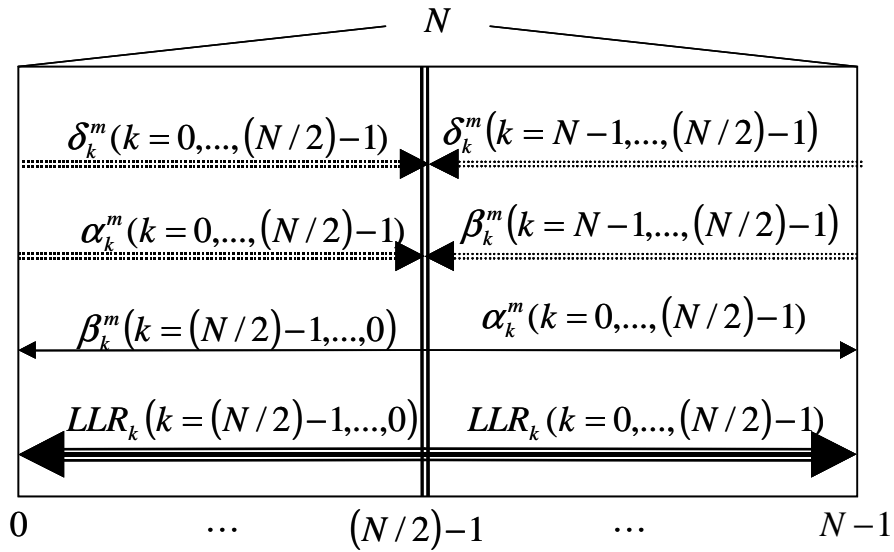


Fig. 3.2 Dual-Path Processing Algorithm.

The procedure of the dual-path processing is as follows.

Step 1: Initialize the forward state metric and backward state metric.

$$\begin{aligned}
 \alpha_0^i(s_0^i(m)) &= 1 \quad \text{for } m = 0 \\
 &= 0, \quad \text{else} \\
 \beta_N^i(s_b^i(m)) &= 1 \quad \text{for } m = 0 \\
 &= 0, \quad \text{else}
 \end{aligned} \tag{3-1}$$

$\alpha_k^i(m)$ and $\beta_k^i(m)$ are FSM and BSM at time of k , information bit of i , and state of m .

Step 2: After receiving the whole set of received symbols of N , FSMs (left to right) and BSMs (right to left) are calculated simultaneously.

$$\hat{\alpha}_k^i(m) = \exp\left(\frac{2}{\sigma^2}(x_k i + y_k Y_k(i, m))\right) \sum_{j=0}^1 \hat{\alpha}_{k-1}^j(S_b^j(m)) \quad (k = 0, \dots, (N/2) - 1) \tag{3-2}$$

$$\hat{\beta}_k^i(m) = \sum_{j=0}^1 \hat{\beta}_{k+1}^j(m) \exp\left(\frac{2}{\sigma^2}(x_{k+1} j + y_{k+1} Y_{k+1}(j, S_f^i(m)))\right) \quad (k = (N-1), \dots, (N/2)) \tag{3-3}$$

Step 3: At the middle point, begin to calculate the log likelihood ratios (LLR).

$$L(\vec{d}_k) = \log \frac{\sum_m \alpha_k^1(m) \beta_k^1(m)}{\sum_m \alpha_k^0(m) \beta_k^0(m)} \quad (k = (N/2), \dots, (N-1)) \quad (3-4)$$

$$L(\overleftarrow{d}_k) = \log \frac{\sum_m \alpha_k^1(m) \beta_k^1(m)}{\sum_m \alpha_k^0(m) \beta_k^0(m)} \quad (k = (N/2) - 1, \dots, 0) \quad (3-5)$$

$L(\vec{d}_k)$ means LLR outputs in the direction of right to left and $L(\overleftarrow{d}_k)$ means

LLR outputs in the direction of left to right.

3.3 Parallel Decoding Algorithm

Different from the original turbo decoder consisting of two decoders concatenated in a serial fashion, this thesis presents a parallel decoder structure using the parallel sum, where the two decoders operate in parallel and update each other immediately and simultaneously after each one has completed its decoding. In decoding the estimated data, this thesis use the sum of the LLR outputs of the parallel decoders to reduce the latency to half while maintaining the same performance level.

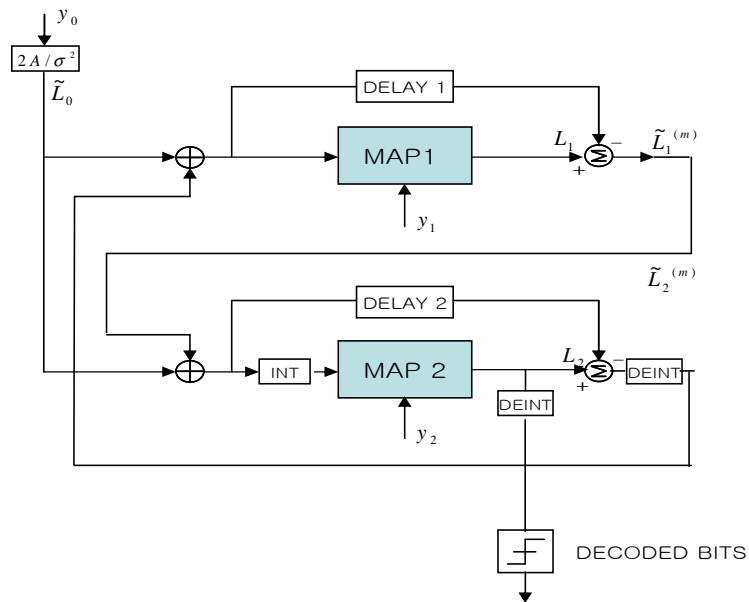


Fig. 3.3 Parallel Decoder

3.4 Early Stop Algorithm

The decoding iteration processes until a certain stopping condition is satisfied, hard decisions are made based on the reliability measures of the decoded symbol at the last decoding iteration. HAD algorithm is used as an early-stop algorithm. It compares each decision generated by the two decoders, and when the two sets of decisions match, it stops decoding on the current block and outputs the hard decision bits. Table 3.1 shows the average number of iterations in an HAD algorithm. At an E_b/N_0 of 6 dB, it requires only two iterations, this means that the decoding speed is improved or the power consumption (cost) is reduced by 74.9 %.

Table 3.1 The average number of iterations according to E_b/N_0 (the predetermined number of iterations is 8).

E_b/N_0 [dB]	serial mode		parallel mode	
	Average number of iterations	Decoding speed improvement (%)	Average number of iterations	Decoding speed improvement (%)
4	3.03	62.1%	4.75	40.6%
5	2.03	74.6%	2.67	66.6%
6	1.85	76.9%	2.01	74.9%

3.5 Simulation Results

The bit-error rate (BER) performance of the new high-speed adaptive turbo decoder architecture combining the four schemes is analyzed in this section. For a comparison purpose, Fig. 3.4 shows the performance of the new decoder and a conventional one using $v=4$ turbo codes with generator polynomials $g_1=(17)_{octal}$, $g_2=(15)_{octal}$ as a function of interleaving size of 212, rate of 2/3, eight iterations and 8 PSK modulation scheme. The performance of high speed decoder is almost the same as that of conventional decoder. Therefore, this thesis increases the decoding speed without degradation of performance.

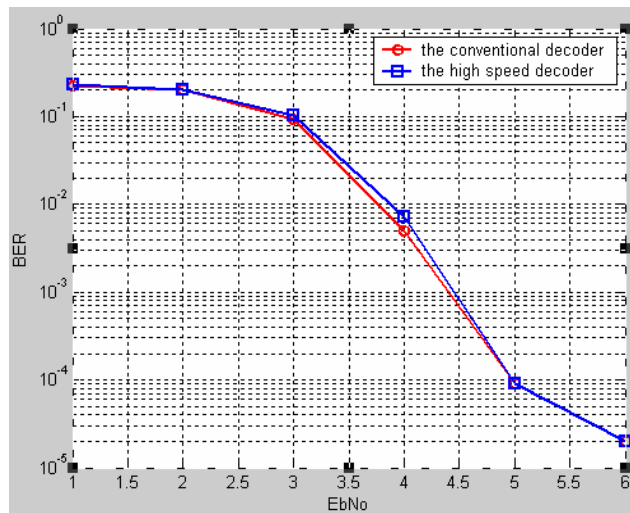


Fig. 3.4 Performance of the proposed decoder over an AWGN channel compared with that of a conventional algorithm

Chapter IV. Design of the Adaptive High-Speed Turbo Decoder

In this section, as shown in Fig. 4.1, this thesis proposes the adaptive turbo decoder architecture with a rate of $2/3$ turbo-coded pragmatic TCM decoder using the *off-the-shelf* half rate turbo decoder based on Fig. 2.1 Based on reduced latency and power consumption version of turbo decoder such as radix-4 algorithm, dual-path processing, parallel algorithm, and early top algorithm, Fig. 4.1 shows the entirely architecture of the adaptive turbo decoder. Our decoder can support both half rate turbo decoder with BPSK modulation scheme and rate of $2/3$ with 8PSK modulation scheme.

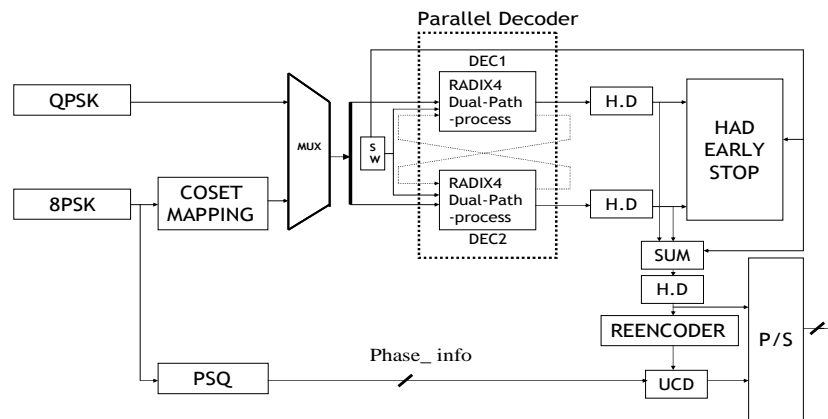
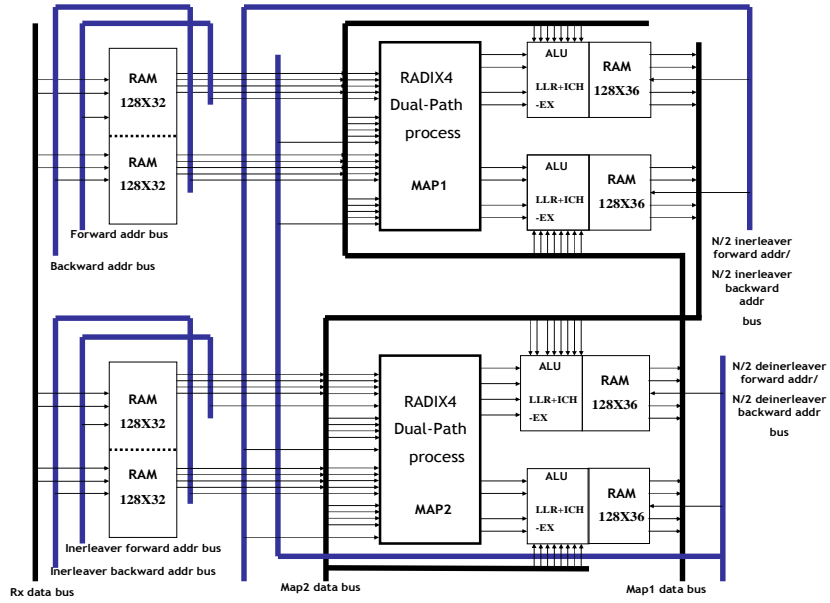


Fig. 4.1 Adaptive turbo decoder structure

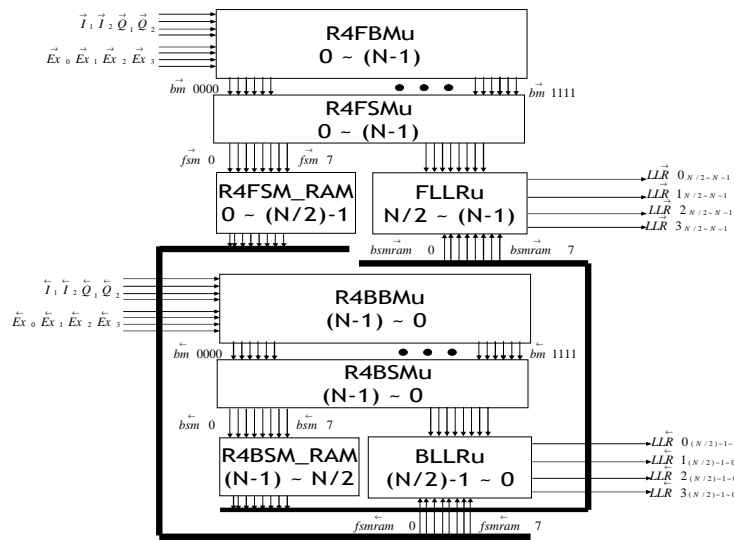
4.1 The Adaptive High-Speed Turbo Decoder Structure

A schematic diagram of the high-speed turbo decoder for implementation is shown in Fig. 4.2(a). A detailed signal flow of internal MAP based on high-speed algorithms is shown in Fig. 4.2(b). In Fig. 4.2(a), MAP 1 and MAP 2 decoders are operated in parallel, with their outputs being log-likelihood ratios of input bits as shown in Fig. 4.2(b). For instance, $\vec{LLR}0_{N/2 \sim N}$ denotes log-likelihood ratios of input bits “00”, $N/2 \sim N$ denotes From the time of $N/2$ to N , and the arrow \rightarrow denotes the direction of the LLR calculation, that is, left-to-right. The ALU (Arithmetic Logic Unit) calculates the extrinsic information using LLR outputs, the received symbol and the previous extrinsic information. To add the extrinsic information exactly in the next iteration, the decoder needs the dual port RAM(128x36) buffered ALU block. As shown in Fig. 4.2(b), the decoder consists of six major units, the Radix-4 Forward Branch Metric unit (R4FBMu), Radix-4 Backward Branch Metric unit (R4BBMu), Radix-4 Forward State Metric unit (R4FSMu), Radix-4 Backward State Metric unit (R4BSMu), Forward LLR unit (FLLRu), Backward LLR unit (BLLRu). After receiving the whole set of received symbols, the quantized I and Q samples are fed to the R4FBMu and R4BBMu at the same time. The branch

metrics between the branch codeword “0000” and the received symbols is denoted by “ $bm0000$ ”. The R4FBMu and R4BBMu calculate the branch metrics for four samples of received data in the direction of left to right and right to left simultaneously. As shown in Fig. 4.2(b), $I_n(n=1,2)$ and $Q_n(n=1,2)$ are fed to R4FBMu to calculate forward branch metric and $I_n(n=1,2)$ and $Q_n(n=1,2)$ are also fed to R4BBMu to calculate backward branch metric. From the second iteration, R4FBMu and R4BBMu need extrinsic information, Ex_n , that is LLR of input bit “ i_1i_2 ”. The index n of Ex_n means $2*i_2 + i_1$. R4FSMu calculates forward state metrics in the direction left-to-right and R4BSMu calculates backward state metrics in the direction right-to-left. The data calculated from R4FSMu and R4BSMu are buffered in dual-port RAM with size of 64x72. Therefore this thesis need the two dual-port RAM that is R4FSM_RAM and R4BSM_RAM. When the R4FSMu and R4BSMu reach the same point, then FLLRu and BLLRu begin to calculate the LLR of information n using the data read from R4FSM_RAM and R4BSM_RAM respectively. ALU block shown in Fig. 4.2(a) begins to calculate the extrinsic information and store extrinsic information in dual-port RAM with size of 128x36 in order to input to next iteration.



(a) High speed turbo decoder architecture



(b) Signal flow of internal MAP

Fig. 4.2 The block diagram of proposed high-speed turbo Decoder

4.2 The Optimum Quantized Bits of the Adaptive Turbo Decoder

In order to implement optimally the high-speed decoder, this thesis determined the optimum quantized bits of each block shown in Fig. 4.2. r_q bits of received in-phase and quadrature signals, b_q bits of R4FBMu and R4BBMu outputs, s_q bits of R4FSMu and R4BSMu outputs, and l_q bits of FLLR and BLLR. By fixed-point computer simulation, the output of the demodulator is quantized to 8 bits. The internal parameters of the turbo decoder were always saturated to 9 bits, and the optimum quantization bits of the turbo decoder derived from the fixed-point simulations are listed in Table 4.1.

Table 4.1 The optimum quantized bits of the adaptive turbo decoder.
(rate = 2/3, 8-states, N=212 bits, 3 iterations, 8PSK)

	Number of optimized quantization bits
r_q	8
b_q	9
s_q	9
l_q	9

4.3 FPGA Implementation

This thesis designed the adaptive turbo decoder using VHDL (Very high-speed Hardware Description Language) and verified its operation by RTL simulation. During the verification, some errors were added in C-description. Then this thesis confirmed that the errors were corrected while being processed in RTL simulation. VHDL and C language process communicate with each other through program language interface. The decoder designed by VHDL was synthesized using the Xilinx FPGA chip (VIRTEX2P(XC2VP30-5FG676)) as shown in Fig. 4.3, The received data file generated by C language process at E_b/N_o of 5[dB] is fed into internal memory of Xilinx FPGA chip. And the result of measurement at the Digital Oscilloscope is shown in Fig. 4.4. By using VHDL, this thesis simulates BM (Fig. 4.5), FSM/BSM (Fig. 4.6) and LLR (Fig. 4.7).

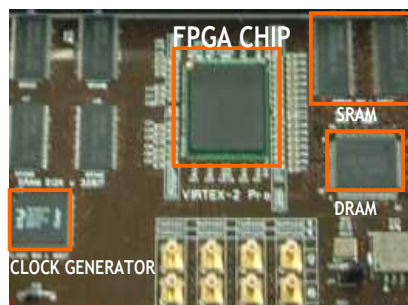


Fig. 4.3 The implement chip-set of the adapted high-speed turbo decoder

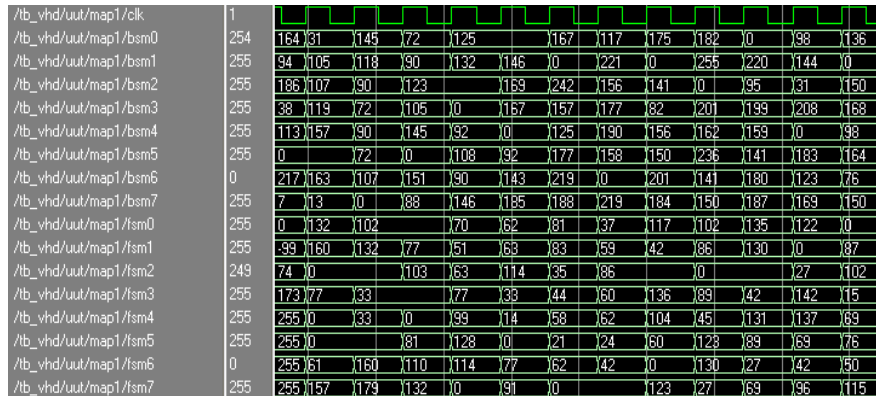


Fig. 4.6 FSM/BSM VHDL simulation result of high speed turbo decoder

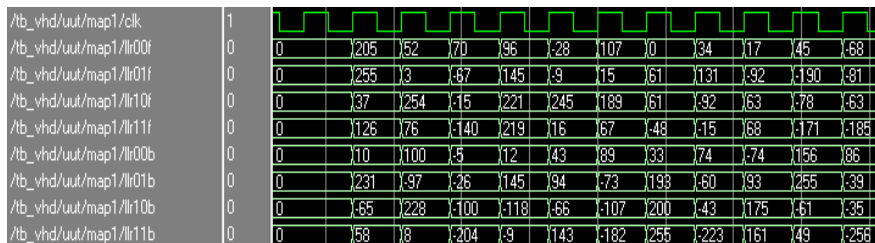


Fig. 4.7 LLR VHDL simulation result of high speed turbo decoder

This thesis designed the high-speed adaptive turbo decoder with 8-state, $N=212$, $R = 2/3$, and 8PSK modulation scheme. In order to compare the decoding speed of the conventional serial turbo decoder based on radix-2 trellis structure. To compare the decoding speed between conventional and high-speed adaptive turbo decoder, this thesis implemented the conventional decoder using the same procedures. Based on Table 3.1, since the required

iteration number is 3 in the case of $E_b/N_0=5$ [dB], this thesis fixed the iteration number to 3. The maximum operating clock speed of the conventional and proposed decoders is 18[ns] Table 4.2 shows comparison of decoding speed between conventional and high-speed decoder. In the case of combining the radix-4 and parallel and dual-path process, the proposed high-speed decoder is faster than conventional one by 6.4 times.

Table 4.2 Comparison of decoding speed between a conventional method and the proposed method (N=212, iteration = 3, 8-state, 8PSK, main clock speed = 18 ns.).

	Conventional decoder	Radix-4 + Serial mode	Radix-4 + Parallel mode	Radix-4 + Parallel mode + dual-path process
Execution time [clocks]	2861	1431	768	446
Decoding Speed	4.11M	8.23M	15.33M	26.4M

Chapter V. Conclusion

This thesis presented the adaptive turbo decoding algorithm with two coded bits per symbol, based on a realization of rate $n/(n+1)$ trellis coded scheme using an *off-the-shelf* turbo decoder. Compared with conventional turbo TCM, the proposed decoder makes the hardware simplified and miniaturized and it shows a small loss of less than 0.2 dB. Furthermore, the consumer power and receiver cost is reduced. The proposed approach may be extended to a variable coding rate(rate-5/6 and rate-8/9) , which depends on how many uncoded bits are assigned. Also, it can be used for 2^m - QAM constellations with $m \geq 4$. To extend the application area of turbo decoding to real time services, it is important to reduce the latency in the decoding process of turbo decoders. This thesis proposed a new high-speed turbo decoding algorithm and implementation architecture. Two new low latency versions of decoder are presented, radix-4 algorithm and dual-path processing combined to parallel mode and early-stop algorithm. This thesis designed the adaptive high-speed turbo decoder using the Xilinx chip(VIRTEX2P(XC2VP30-5FG676)). From the results, it was found that the decoding speed of the proposed decoder is faster than that of conventional algorithms by 6.4 times

under the following conditions : $N=212$, iteration=3, 8-states, 3 iterations,
and 8PSK modulation scheme .

References

- [1] C. Berrou, A. Glavieux, and P.Thitimajshima, "Near Shannon Limit Error-Correcting Code and Decoding: Turbo Codes", in *Proc. Of ICC'93*, 1993.
- [2] "Digital Video Broadcasting standard for Return Channel via Satellite(DVB-RCS)", ETSI TR 101 790 V1.2.1, 2003.
- [3] J.W. Jung, X.Hiang, "Performance Analysis and Optimum Design of Pragmatic Code for Rain-Attenuation Compensation in Satellite Communication", AIAA conference, Montreal, May, 2002.
- [4] Haruo Ogiwara, et al, "Improvement of Turbo Trellis-Coded Modulation System", *IEICE Trans. Fundamentals*, vol.E81-A, No.10, October 1998.
- [5] L. R. Bahl et al, "Optimal Decoding of Linear Code for Minimizing Symbol Error Rate", *IEEE Trans. On Info. Theory*, Vol. IT-20, pp.248-287, Mar. 1994.
- [6] S.S.Pietrobo, "Implementation and Performance of a Turbo/MAP Decoder," to be appear in *International Journal of Satellite Communications*.
- [7] E.choi, J.Jung, N.Kim, Y.Kim, and D.Oh, "A Simplified Decoding Algorithm Using Symbol Transformation for Turbo Pragmatic Trellis-Coded Modulation", *ETRI Journal*, Vol 27. No.2. PP.223-226, April, 2005.

감사의 글

사랑하는 사람들과 함께 했던 학부, 대학원 생활을 마무리 하면서 많이 부족했던 저를 한 단계 도약시켜 주신 고마운 분들이 생각이 납니다. 먼저 본 논문이 완성 되기까지 늘 따듯하게 배려해 주시고 용기 잃지 않게 웃으시면서 밤새 함께 해주신 정지원 교수님께 가슴 깊이 감사드립니다. 그리고 지금 사회의 첫발을 내딛는 이순간 교수님과 추억이 한편의 영화 필름처럼 지나가며 연구실, 헬스장, 외국 학회, 무수한 맛 집 등등 많은 추억의 장소에서 함께 보낸 길지 않았던 시간이지만 무엇보다 바꿀 수 없는 행복한 기억으로 인생의 한 페이지를 넘길 것 같습니다. 그리고 논문의 미비점을 보완하여 충실한 내용이 될 수 있도록 해 주셨던 김기만 교수님, 운영 교수님께 감사드리며, 항상 새로운 가르침을 주시고, 조언을 아끼지 않으셨던 김동일 교수님, 조형래 교수님, 민경식 교수님, 강인호 교수님께도 감사드립니다.

연구실에 처음 들어와서 늘 친 동생처럼 챙겨주셨던 그리고 지금도 항상 고민 들어주시는 태길이가 형에게 감사의 마음을 전하고 멀리 일본에 계신 성준이 형에게도 고마움을 전합니다. 지금은 대전에 있는 연구실에서 같이 즐거운 추억 만들었던 지금까지 힘이 되어주는 친구 인기에게 고마움을 전합니다. 그리고 위성통신연구실을 함께 하셨던 상명이 형, 상훈이 형, 재범이 형, 상우 형, 원철이 형, 상진이 형에게 고마움을 전하며 또한 연구실에 묵묵히 절 따라준 진희, 민혁이와 학부생 종태, 석순이에게도 고마움을 전하고 항상 웃으며 대학원 생활했던 동환, 경식, 세영, 제현이, 그리고 많은 조언 해주신 동한이 형, 철성이 형, 동식이 형, 충렬이 형, 영배 형, 재국이 형, 그리고 후배 형준이 찬섭이에게도 감사의 마음을 전합니다.

항상 붙어 다녔던 지금은 멀리 있지만 늘 마음속에 숨쉬고 있는 자랑스러운 친구 석봉이와 승재 그리고 98 동기 모두에게 고마움을 전하며

나를 믿고 따라준 성욱이, 수찬이, 정표에게도 고맙다고 전합니다. 그리고
고 철없던 나를 챙겨 주신 아마도 지금 웃음을 참지 못하실 것 같은
승민이 형, 세진이 형, 정광이 형, 승현이 형, 익수 형 마지막으로 상
부상조 권상조 형께도 감사의 마음 표합니다.

지금 서울에서 힘든 시간을 보내고 있는 부족한 나를 곁에서 한결같이
용기 북돋아주고 믿어 줬던 사랑하는 여자 친구 태경이에게 정말 고
맙고 지면으로나마 사랑한다는 말 전합니다.

마지막으로 건강하게 절 키워주신 지금의 이런 모습 보며 흐뭇해 하
실 것 같은 하늘 나라에 계신 할아버지 고맙습니다. 그리고 아직까지
손주 귀여워 해주시고 맛있는 된장찌개 만들어 주시는 할머니와 늘 저
를 응원해주고 걱정해주시는 아버지와 어머니 그리고 멋진 장교 생활
하는 하나뿐인 동생 덕주에게도 감사의 마음을 전합니다.

글을 마무리 하면서 지금까지 제가 있었던 건 이렇게 좋으신 분들이
곁에서 도와주신 덕분이고 그 사랑에 조금 이나마 보답하고 힘이 되
드리기 위해 감사하면서 발전하는 모습으로 살겠습니다.