

Evaluation of Soft Output Decoding for Turbo Codes

by

Fu-hua Huang

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING

Dr. F. Gail Gray, Chair
Dr. Tim Pratt
Dr. Jeffrey H. Reed
Dr. Brian D. Woerner

May 29, 1997
Blacksburg, Virginia

Keywords: Turbo, Convolutional, Codes, Viterbi, Decoding

Copyright 1997, Fu-hua Huang

Evaluation of Soft Output Decoding for Turbo Codes

by

Fu-hua Huang

Dr. F. Gail Gray, Chair

Abstract

Evaluation of soft output decoding for turbo codes is presented. Coding theory related to this research is studied, including convolutional encoding and Viterbi decoding. Recursive systematic convolutional (RSC) codes and nonuniform interleavers commonly used in turbo code encoder design are analyzed. Fundamentals such as reliability estimation, log-likelihood algebra, and soft channel outputs for soft output Viterbi algorithm (SOVA) turbo code decoding are examined. The modified Viterbi metric that incorporates a-priori information used for SOVA decoding is derived. A low memory implementation of the SOVA decoder is shown. The iterative SOVA turbo code decoding algorithm is described with illustrative examples. The performance of turbo codes are evaluated through computer simulation. It has been found that the SOVA turbo code decoding algorithm, as described in the literature, did not perform as well as the published results. Modifications to the decoding algorithm are suggested. The simulated turbo code performance results shown after these modifications more closely match with current published research work.

To My Parents

Acknowledgment

I would like to express my most sincere gratitude and appreciation to my advisor, Dr. F. Gail Gray, for his constant support and guidance throughout the course of this work. His valuable expertise, advice, and encouragement made this work possible.

I would like to thank Dr. Brian D. Woerner, for his help, valuable discussions, and comments on this research. I also would like to thank Dr. Tim Pratt and Dr. Jeffrey H. Reed for serving as members on the committee and for their suggestions regarding this work.

I am greatly indebted to my parents for their constant support and encouragement of pursuing my education. I could not have succeeded without them.

My study here at Virginia Tech as both undergraduate and graduate was valuable and enjoyable due to the excellent faculty, staff, and students. In particular, I would like to thank Mr. Tsunou Chang, Mr. Kai Dietz, Mr. Rony Husein, Mr. Tariqul Islam, Mr. Conrad Lee, Mr. Keng Jin Lian, Mr. Tung-Kuang Lin, Mr. Andre Siregar, Mr. Chihyung (Carl) Tu, Mr. Matt Valenti, and Mr. GU Yum for their insightful discussions and of course, Mr. Farooq Azam for his computer assistance.

Finally, I am grateful to my girlfriend, Mandana Massjouni, and her family for their caring support.

Table of Contents

List of Figures	vii
List of Tables	xii
1 Introduction	1
1.1 Types of Channel Codes	1
1.2 Outline of Thesis	2
2 Convolutional Codes	4
2.1 Encoder Structure	4
2.2 Encoder Representations	5
2.2.1 Generator Representation	6
2.2.2 Tree Diagram Representation	6
2.2.3 State Diagram Representation	7
2.2.4 Trellis Diagram Representation	8
2.3 Catastrophic Convolutional Code	10
2.4 Hard-Decision and Soft-Decision Decoding	10
2.5 Hard-Decision Viterbi Algorithm	11
2.6 Soft-Decision Viterbi Algorithm	17
2.6.1 Soft-Decision Viterbi Algorithm (Method 1)	17
2.6.2 Soft-Decision Viterbi Algorithm (Method 2)	18
2.7 Performance Analysis of Convolutional Code	19
2.7.1 Transfer Function of Convolutional Code	20
2.7.1.1 Distance Properties	21
2.7.1.2 Error Probabilities	21
2.7.2 Decoding Depth	22
2.7.3 Degree of Quantization	23
2.7.4 Decoding Complexity for Convolutional Codes	23
3 Turbo Code Encoder	24
3.1 Recursive Systematic Convolutional (RSC) Encoder	24
3.1.1 Trellis Termination	26
3.2 Recursive and Nonrecursive Convolutional Encoders	27
3.3 Concatenation of Codes	29
3.4 Interleaver Design	31
3.4.1 Block Interleaver	32
3.4.2 Random (Pseudo-Random) Interleaver	33
3.4.3 Circular-Shifting Interleaver	33
3.4.4 Semirandom Interleaver	34
3.4.5 Odd-Even Interleaver	35
3.4.6 Optimal (Near-Optimal) Interleaver	37

4	Iterative Turbo Code Decoder	38
4.1	Principle of the General Soft-Output Viterbi Decoder	38
4.2	Reliability of the General SOVA Decoder	39
4.3	Introduction to SOVA for Turbo Codes	43
	4.3.1 Log-Likelihood Algebra	44
	4.3.2 Soft Channel Outputs	48
4.4	SOVA Component Decoder for a Turbo Code	49
4.5	SOVA Implementation	55
4.6	SOVA Iterative Turbo Code Decoder	56
5	Performance Analysis of Turbo Codes	68
5.1	Simulation Setup	68
5.2	Simulation Results	71
5.3	Simulation Analysis	86
5.4	Comparison of Turbo Code Decoding and Viterbi Decoding	87
6	Conclusion	89
6.1	Future Work	90
	References	91
	Vita	93

List of Figures

Figure No.	Title	Page No.
Figure 2.1	Example convolutional encoder where $x^{(i)}$ is an input information bit stream and $c^{(i)}$ is an output encoded bit stream [Wic95].	4
Figure 2.2	Convolutional encoder with $k=1$, $n=2$, $r=1/2$, $m=2$, and $K=3$.	5
Figure 2.3	Tree diagram representation of the encoder in Figure 2.2 for four input bit intervals.	6
Figure 2.4	State diagram representation of the encoder in Figure 2.2.	7
Figure 2.5	The state transitions (path) for input information sequence {1011}.	8
Figure 2.6	Trellis diagram representation of the encoder in Figure 2.2 for four input bit intervals.	9
Figure 2.7	Trellis path for the state transitions in Figure 2.5.	9
Figure 2.8	Examples of catastrophic convolutional code.	10
Figure 2.9	Hard- and Soft-decision decoding [Woe94].	11
Figure 2.10	Convolutional code system.	11
Figure 2.11	The binary symmetric channel model, where p is the crossover probability.	13
Figure 2.12	The state transition diagram (trellis legend) of the example convolutional encoder.	16
Figure 2.13	HDVA decoding of the example.	17
Figure 2.14	The modified state diagram of Figure 2.4 where S_a is the start state and S_e is the end state.	20
Figure 3.1	Fundamental turbo code encoder.	24

Figure 3.2	Conventional convolutional encoder with $r=1/2$ and $K=3$.	25
Figure 3.3	The RSC encoder obtained from Figure 3.2 with $r=1/2$ and $K=3$.	26
Figure 3.4	Trellis termination strategy for RSC encoder.	27
Figure 3.5	Nonrecursive $r=1/2$ and $K=2$ convolutional encoder with input and output sequences.	27
Figure 3.6	Recursive $r=1/2$ and $K=2$ convolutional encoder of Figure 3.5 with input and output sequences.	28
Figure 3.7	State diagram of the nonrecursive encoder in Figure 3.5.	28
Figure 3.8	State diagram of recursive encoder in Figure 3.6.	29
Figure 3.9	Serial concatenated code.	30
Figure 3.10	Parallel concatenated code.	30
Figure 3.11	The interleaver increases the code weight for RSC Encoder 2 as compared to RSC Encoder 1.	31
Figure 3.12	An illustrative example of an interleaver's capability.	32
Figure 3.13	Block interleaver.	33
Figure 3.14	A random (pseudo-random) interleaver with $L=8$.	33
Figure 3.15	A circular-shifting interleaver with $L=8$, $a=3$, and $s=0$.	34
Figure 3.16	A semirandom interleaver with $L=16$ and $S=2$.	35
Figure 3.17	Odd coded bits of sequence c_2 are stored for information (systematic) sequence x .	36
Figure 3.18	3×3 block interleaver.	36
Figure 3.19	Even coded bits of sequence c_3 are stored for permuted information sequence x .	36

Figure 3.20	Information sequence x and multiplexed coded sequence.	36
Figure 4.1	A concatenated SOVA decoder where y represents the received channel values, u represents the hard decision output values, and L represents the associated reliability values.	38
Figure 4.2	Example of survivor and competing paths for reliability estimation at time t [Ber93a].	39
Figure 4.3	Example that shows the weakness of reliability assignment using metric values directly.	40
Figure 4.4	Updating process for time $t-2$ ($MEM_{low}=2$).	42
Figure 4.5	Updating process for time $t-4$ ($MEM_{low}=4$).	43
Figure 4.6	System model for SOVA derivation.	43
Figure 4.7	SOVA component decoder.	49
Figure 4.8	Source reliability for SOVA metric computation.	51
Figure 4.9	Weighting properties of the SOVA metric.	52
Figure 4.10	Example of SOVA survivor and competing paths for reliability estimation.	53
Figure 4.11	SOVA decoder implementation.	56
Figure 4.12	SOVA iterative turbo code decoder.	57
Figure 4.13	Turbo code encoder structure for the example.	59
Figure 4.14	RSC component encoder for the example.	59
Figure 4.15	State diagram of the RSC component encoder for the example.	60
Figure 4.16	Transmission state diagram of the RSC component encoder for the example.	60

Figure 4.17	Trellis legend (state transition diagram) of the RSC component encoder for the example. The first coded bit is the systematic bit and the second coded bit is the parity check bit.	61
Figure 4.18	SOVA iterative turbo code decoder for the example.	62
Figure 4.19	The first SOVA decoder's (of the SOVA1 component decoder) ML path.	63
Figure 4.20	The first SOVA decoder's (of the SOVA2 component decoder) ML path.	65
Figure 5.1	Termination of the systematic bit stream is associated with recursive encoder 1.	69
Figure 5.2	Termination of the systematic bit stream is associated with recursive encoder 2.	69
Figure 5.3	Rate 1/2 RSC code BER performance in soft-decision Viterbi decoding.	74
Figure 5.4	Comparison of turbo code BER performance for 10 decoding iterations ($R=1/2$, $K=3$, $FS=192$).	75
Figure 5.5	Comparison of turbo code BER performance for 10 decoding iterations ($R=1/2$, $K=4$, $FS=192$).	76
Figure 5.6	Turbo code BER performance for 10 decoding iterations ($R=1/2$, $K=3$).	77
Figure 5.7	Turbo code BER performance for 10 decoding iterations ($R=1/3$, $K=3$).	78
Figure 5.8	Turbo code BER performance for 10 decoding iterations ($R=1/2$, $K=4$).	79
Figure 5.9	Turbo code BER performance for 10 decoding iterations ($FS=50$).	80
Figure 5.10	Turbo code BER performance for 10 decoding iterations ($FS=192$).	81
Figure 5.11	Turbo code BER performance for 10 decoding iterations ($FS=1000$).	82

Figure 5.12	Turbo code BER performance gain ($R=1/2$, $K=3$).	83
Figure 5.13	Turbo code BER performance gain ($R=1/3$, $K=3$).	84
Figure 5.14	Turbo code BER performance gain ($R=1/2$, $K=4$).	85

List of Tables

Table No.	Title	Page No.
Table 2.1	Conventional Bit Metric Values	13
Table 2.2	Alternative Bit Metric Values	14
Table 3.1	Input and Output Sequences for Encoder in Figure 3.12	32
Table 4.1	Outcome of Adding Two Binary Random Variables u_1 and u_2	44
Table 4.2	Characteristics of the Log-likelihood Ratio $L(u)$	44
Table 4.3	Comparison of $L(u_1 \oplus u_2)$ Between Exact and Approximated Solutions	46
Table 4.4	Survivor and Competing Partial Path Metrics for the Trellis Diagram in Figure 4.18	63
Table 4.5	SOVA1's Competing Path (Bit and State Sequences) for Reliability Updates	64
Table 4.6	SOVA1's Calculated and Updated (In Bold) Reliability Values	64
Table 4.7	Survivor and Competing Partial Path Metrics for the Trellis Diagram in Figure 4.19	66
Table 4.8	SOVA2's Competing Path (Bit and State Sequences) for Reliability Updates	66
Table 4.9	SOVA2's Calculated and Updated (In Bold) Reliability Values	66
Table 5.1	Rate 1/2 RSC Component Codes Used in Simulation Results	71
Table 5.2	Rate 1/2 RSC Code BER Performance in Soft Decision Viterbi Decoding	71

Table 5.3	Comparison of Turbo Code BER Performance for 10 Decoding Iterations ($R=1/2$, $K=3$, $FS=192$)	71
Table 5.4	Comparison of Turbo Code BER Performance for 10 Decoding Iterations ($R=1/2$, $K=4$, $FS=192$)	72
Table 5.5	Turbo Code BER Performance for 10 Decoding Iterations ($R=1/2$, $K=3$)	72
Table 5.6	Turbo Code BER Performance for 10 Decoding Iterations ($R=1/3$, $K=3$)	72
Table 5.7	Turbo Code BER Performance for 10 Decoding Iterations ($R=1/2$, $K=4$)	72
Table 5.8	Turbo Code BER Performance for 10 Decoding Iterations ($FS=50$)	72
Table 5.9	Turbo Code BER Performance for 10 Decoding Iterations ($FS=192$)	73
Table 5.10	Turbo Code BER Performance for 10 Decoding Iterations ($FS=1000$)	73
Table 5.11	Turbo Code BER Performance Gain ($R=1/2$, $K=3$)	73
Table 5.12	Turbo Code BER Performance Gain ($R=1/3$, $K=3$)	73
Table 5.13	Turbo Code BER Performance Gain ($R=1/2$, $K=4$)	73