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. Chi hyung (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
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

<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1</td>
<td>Example convolutional encoder where $x^{(i)}$ is an input information bit stream and $c^{(i)}$ is an output encoded bit stream [Wic95].</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Convolutional encoder with $k=1$, $n=2$, $r=1/2$, $m=2$, and $K=3$.</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Tree diagram representation of the encoder in Figure 2.2 for four input bit intervals.</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>State diagram representation of the encoder in Figure 2.2.</td>
<td>7</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>The state transitions (path) for input information sequence {1011}.</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Trellis diagram representation of the encoder in Figure 2.2 for four input bit intervals.</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2.7</td>
<td>Trellis path for the state transitions in Figure 2.5.</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2.8</td>
<td>Examples of catastrophic convolutional code.</td>
<td>10</td>
</tr>
<tr>
<td>Figure 2.9</td>
<td>Hard- and Soft-decision decoding [Woe94].</td>
<td>11</td>
</tr>
<tr>
<td>Figure 2.10</td>
<td>Convolutional code system.</td>
<td>11</td>
</tr>
<tr>
<td>Figure 2.11</td>
<td>The binary symmetric channel model, where $p$ is the crossover probability.</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2.12</td>
<td>The state transition diagram (trellis legend) of the example convolutional encoder.</td>
<td>16</td>
</tr>
<tr>
<td>Figure 2.13</td>
<td>HDVA decoding of the example.</td>
<td>17</td>
</tr>
<tr>
<td>Figure 2.14</td>
<td>The modified state diagram of Figure 2.4 where $S_a$ is the start state and $S_e$ is the end state.</td>
<td>20</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Fundamental turbo code encoder.</td>
<td>24</td>
</tr>
</tbody>
</table>
Figure 3.2 Conventional convolutional encoder with r=1/2 and K=3.

Figure 3.3 The RSC encoder obtained from Figure 3.2 with r=1/2 and K=3.

Figure 3.4 Trellis termination strategy for RSC encoder.

Figure 3.5 Nonrecursive r=1/2 and K=2 convolutional encoder with input and output sequences.

Figure 3.6 Recursive r=1/2 and K=2 convolutional encoder of Figure 3.5 with input and output sequences.

Figure 3.7 State diagram of the nonrecursive encoder in Figure 3.5.

Figure 3.8 State diagram of recursive encoder in Figure 3.6.

Figure 3.9 Serial concatenated code.

Figure 3.10 Parallel concatenated code.

Figure 3.11 The interleaver increases the code weight for RSC Encoder 2 as compared to RSC Encoder 1.

Figure 3.12 An illustrative example of an interleaver’s capability.

Figure 3.13 Block interleaver.

Figure 3.14 A random (pseudo-random) interleaver with L=8.

Figure 3.15 A circular-shifting interleaver with L=8, a=3, and s=0.

Figure 3.16 A semirandom interleaver with L=16 and S=2.

Figure 3.17 Odd coded bits of sequence c_2 are stored for information (systematic) sequence x.

Figure 3.18 3 x 3 block interleaver.

Figure 3.19 Even coded bits of sequence c_3 are stored for permuted information sequence x.
Figure 3.20 Information sequence x and multiplexed coded sequence.

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.

Figure 4.2 Example of survivor and competing paths for reliability estimation at time t [Ber93a].

Figure 4.3 Example that shows the weakness of reliability assignment using metric values directly.

Figure 4.4 Updating process for time t-2 (MEM_{low}=2).

Figure 4.5 Updating process for time t-4 (MEM_{low}=4).

Figure 4.6 System model for SOVA derivation.

Figure 4.7 SOVA component decoder.

Figure 4.8 Source reliability for SOVA metric computation.

Figure 4.9 Weighting properties of the SOVA metric.

Figure 4.10 Example of SOVA survivor and competing paths for reliability estimation.

Figure 4.11 SOVA decoder implementation.

Figure 4.12 SOVA iterative turbo code decoder.

Figure 4.13 Turbo code encoder structure for the example.

Figure 4.14 RSC component encoder for the example.

Figure 4.15 State diagram of the RSC component encoder for the example.

Figure 4.16 Transmission state diagram of the RSC component encoder for the example.
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.

Figure 4.18 SOVA iterative turbo code decoder for the example.

Figure 4.19 The first SOVA decoder’s (of the SOVA1 component decoder) ML path.

Figure 4.20 The first SOVA decoder’s (of the SOVA2 component decoder) ML path.

Figure 5.1 Termination of the systematic bit stream is associated with recursive encoder 1.

Figure 5.2 Termination of the systematic bit stream is associated with recursive encoder 2.

Figure 5.3 Rate 1/2 RSC code BER performance in soft-decision Viterbi decoding.

Figure 5.4 Comparison of turbo code BER performance for 10 decoding iterations (R=1/2, K=3, FS=192).

Figure 5.5 Comparison of turbo code BER performance for 10 decoding iterations (R=1/2, K=4, FS=192).

Figure 5.6 Turbo code BER performance for 10 decoding iterations (R=1/2, K=3).

Figure 5.7 Turbo code BER performance for 10 decoding iterations (R=1/3, K=3).

Figure 5.8 Turbo code BER performance for 10 decoding iterations (R=1/2, K=4).

Figure 5.9 Turbo code BER performance for 10 decoding iterations (FS=50).

Figure 5.10 Turbo code BER performance for 10 decoding iterations (FS=192).

Figure 5.11 Turbo code BER performance for 10 decoding iterations (FS=1000).
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>
<thead>
<tr>
<th>Table No.</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>Conventional Bit Metric Values</td>
<td>13</td>
</tr>
<tr>
<td>Table 2.2</td>
<td>Alternative Bit Metric Values</td>
<td>14</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>Input and Output Sequences for Encoder in Figure 3.12</td>
<td>32</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Outcome of Adding Two Binary Random Variables $u_1$ and $u_2$</td>
<td>44</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Characteristics of the Log-likelihood Ratio $L(u)$</td>
<td>44</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Comparison of $L(u_1\oplus u_2)$ Between Exact and Approximated Solutions</td>
<td>46</td>
</tr>
<tr>
<td>Table 4.4</td>
<td>Survivor and Competing Partial Path Metrics for the Trellis Diagram in Figure 4.18</td>
<td>63</td>
</tr>
<tr>
<td>Table 4.5</td>
<td>SOVA1’s Competing Path (Bit and State Sequences) for Reliability Updates</td>
<td>64</td>
</tr>
<tr>
<td>Table 4.6</td>
<td>SOVA1’s Calculated and Updated (In Bold) Reliability Values</td>
<td>64</td>
</tr>
<tr>
<td>Table 4.7</td>
<td>Survivor and Competing Partial Path Metrics for the Trellis Diagram in Figure 4.19</td>
<td>66</td>
</tr>
<tr>
<td>Table 4.8</td>
<td>SOVA2’s Competing Path (Bit and State Sequences) for Reliability Updates</td>
<td>66</td>
</tr>
<tr>
<td>Table 4.9</td>
<td>SOVA2’s Calculated and Updated (In Bold) Reliability Values</td>
<td>66</td>
</tr>
<tr>
<td>Table 5.1</td>
<td>Rate 1/2 RSC Component Codes Used in Simulation Results</td>
<td>71</td>
</tr>
<tr>
<td>Table 5.2</td>
<td>Rate 1/2 RSC Code BER Performance in Soft Decision Viterbi Decoding</td>
<td>71</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.3</td>
<td>Comparison of Turbo Code BER Performance for 10 Decoding Iterations (R=1/2, K=3, FS=192)</td>
<td>71</td>
</tr>
<tr>
<td>5.4</td>
<td>Comparison of Turbo Code BER Performance for 10 Decoding Iterations (R=1/2, K=4, FS=192)</td>
<td>72</td>
</tr>
<tr>
<td>5.5</td>
<td>Turbo Code BER Performance for 10 Decoding Iterations (R=1/2, K=3)</td>
<td>72</td>
</tr>
<tr>
<td>5.6</td>
<td>Turbo Code BER Performance for 10 Decoding Iterations (R=1/3, K=3)</td>
<td>72</td>
</tr>
<tr>
<td>5.7</td>
<td>Turbo Code BER Performance for 10 Decoding Iterations (R=1/2, K=4)</td>
<td>72</td>
</tr>
<tr>
<td>5.8</td>
<td>Turbo Code BER Performance for 10 Decoding Iterations (FS=50)</td>
<td>72</td>
</tr>
<tr>
<td>5.9</td>
<td>Turbo Code BER Performance for 10 Decoding Iterations (FS=192)</td>
<td>73</td>
</tr>
<tr>
<td>5.10</td>
<td>Turbo Code BER Performance for 10 Decoding Iterations (FS=1000)</td>
<td>73</td>
</tr>
<tr>
<td>5.11</td>
<td>Turbo Code BER Performance Gain (R=1/2, K=3)</td>
<td>73</td>
</tr>
<tr>
<td>5.12</td>
<td>Turbo Code BER Performance Gain (R=1/3, K=3)</td>
<td>73</td>
</tr>
<tr>
<td>5.13</td>
<td>Turbo Code BER Performance Gain (R=1/2, K=4)</td>
<td>73</td>
</tr>
</tbody>
</table>