In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

**Effects of Clipping and Quantization on Min-Sum
Algorithm and Its Modifications for Decoding
Low-Density Parity-Check (LDPC) Codes**

by

Jianguang Zhao, M.A.Sc.

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute For Electrical and Computer Engineering
Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University
Ottawa, Ontario
September, 2003

© Copyright
Jianguang Zhao, 2003

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-88906-8

Our file Notre référence

ISBN: 0-612-88906-8

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Canada

Abstract

Belief propagation (BP) and min-sum (MS) algorithms are two well-known algorithms for decoding LDPC codes. Due to its complexity, BP algorithm is difficult to implement. On the other hand, MS is suitable in hardware implementations as it is less complex, but it sacrifices a few tenths of a dB in performance. This thesis examines different methods of improving MS algorithm with little increase in complexity. We first discuss the performance of modified MS algorithms, which includes MS with conditional correction and unconditional correction. All discussions are supported by extensive simulations. Further, we discuss the performance of quantized versions of these algorithms. Both uniform and non-uniform quantization are studied and compared. The complexity of these algorithms is also investigated. Our simulation results show that, the quantized MS with modifications can achieve performance very close to that of BP, and in some cases, can even outperform BP. This is theoretically possible for LDPC codes with cycles on their Tanner graphs, as BP algorithm is only optimal on cycle-free graphs.

Acknowledgements

I would like to thank my thesis supervisor Professor Amir H. Banihashemi for his ongoing support and encouragement in my research work. He was always available for fruitful and engaged discussions regarding my work.

I want to thank all my colleagues at our research group and laboratory. They helped me a lot by providing an intellectually stimulating and enjoyable atmosphere.

I am also grateful to Dr. Robert H. Morelos-Zaragoza [35], David J.C. Mackay [36] for providing me with some programs and LDPC codes in my simulation.

Finally, I will thank my wife who has made a great sacrifice in supporting me in my studies and daily life, and my son who is always the source of my joy. Without them, life would have been very difficult.

Table of Contents

Abstract.....	iii
Acknowledgements	iv
List of Figures and Tables.....	vi
List of Abbreviations and Symbols.....	xv
Chapter 1.....	1
1.1 Introduction.....	1
1.2 Communications scenario and simulation platform	4
Chapter 2.....	6
2.1 LDPC Codes	6
2.2 Comparison of LDPC Codes and Turbo-Codes.....	8
2.3 Iterative decoding based on Tanner graph.....	8
2.4 Belief Propagation and Min-Sum algorithms	11
2.4. 1 Belief propagation (BP) algorithm in probability domain.....	11
2.4. 2 Belief propagation (BP) algorithm in Log Likelihood Ratio (LLR) domain.....	14
2.4. 3 MS algorithm in LLR domain.....	15
Chapter 3.....	17
3.1 Overview.....	17
3.2 Modified MS Algorithm	18
3.3 Simulation results and discussion	22
3.4 Complexity analysis results	28
Chapter 4.....	29
4.1 Overview.....	29
4.2 Effect of clipping	30
4.2.1 Effects of clipping on MS algorithm	30
4.2.2 Effects of clipping on modified MS algorithms	34
4.2.3 Another clipping method	36
4.3 Uniformly quantized MS algorithm.....	38
4.3.1 Uniformly quantized MS algorithm and its performance	39
4.3.2 Uniformly quantized MS with modifications	43
4.3.2.1 Quantized MS with conditional correction.....	43
4.3.2.2 Performance for quantized MS with unconditional correction.....	46
4.4 Complexity analysis results	48
4.5 Another modification of MS algorithm	50
Chapter 5.....	52
5.1 Overview.....	52
5.2 Nonuniform quantization scheme	53
5.3 Performance of nonuniformly quantized MS	57
5.4 Performance of nonuniformly quantized MS with modifications	58
5.5 Complexity analysis results	60
Chapter 6.....	62
6.1 Conclusion	64
6.2 Future work.....	65
APPENDIX.....	66
REFERENCE.....	117

List of Figures and Tables

Fig. 1. A simple model for a communication system	1
Fig. 2. The binary-input additive white Gaussian noise channel.....	4
Fig. 3. H matrix and its Tanner graph.....	7
Fig. 4. Messages passing through check node j to bit node i	10
Fig. 5. Messages passing through bit node i to check node j	10
Fig. 6. Correction factor $f(d)$ for min-sum algorithm.	20
Fig. 7. BER (—)/WER (—) for code I, the effect of y_c and d_c on performance of MS with conditional correction at SNR=1.5dB.....	24
Fig. 8. BER (—)/WER (—) for code I, performance of MS with conditional correction for fixed y_c and d_c	24
Fig. 9. BER for code I, the effect of y_c on performance of MS with unconditional correction at SNR=1.5, 2.0dB	26
Fig. 10. BER for code I, performance of MS with conditional and unconditional correction with fixed y_c and d_c	26
Fig. 11. BER (—)/WER (—) for code II, performance of BP, MS, MS with conditional correction with $y_c=0.1$, $d_c=0.3$	28
Fig. 12. BER (—)/WER (—) for code II, performance of BP, MS, MS with unconditional correction ($y_c=0.1$).....	28
Fig. 13. BER (—)/WER (—) performance for code I for different clipping threshold at SNR=1.5dB.....	32
Fig. 14. BER (—)/WER (—) performance for code I for different clipping threshold at SNR=2.0dB.....	32
Fig. 15. BER for code I (—), II (—) and III (—), performance of MS (o), clipped MS (no character) and BP (.)	33
Fig. 16. BER/WER for code I, the effect c_{th} on performance of clipped MS at SNR=2.5, 2.75dB.....	34
Fig. 17. Optimal c_{th} values at different SNR for code I.....	34
Fig. 18. BER curves for code I (—), II (—) and III (—), decoded by MS (o), clipped MS (no character) and BP (.). The optimal c_{th} is 2.0, 2.0, 2.0, 2.5, 3.5 at SNR=1.0, 1.5, 2.0, 2.5, 2.75dB respectively for code I. The optimal c_{th} is 1.5 at SNR=2.0-4.0dB and 2.5 at SNR=4.5-5.5dB for code II. The optimal c_{th} is 1.25 for all SNR values for code III.....	35
Fig. 19. BER for code I, decoded by clipped MS with conditional and unconditional correction with fixed $c_{th}=3.0$	36
Fig. 20. BER for code I, decoded by clipped MS with unconditional correction $y_c=0.2$ with optimal $c_{th}=3.0$ at SNR=1.0-2.5dB, optimal $c_{th}=4.0$ at SNR=2.75dB	37
Fig. 21. BER for code I, uniformly quantized MS using clipping method of section 4.2 (—) and section 4.2.3 (—) ($c_{th}=2.0$).....	38
Fig. 22. Scheme of 3 bit uniform quantization	39
Fig. 23. BER for code I, the effect of c_{th} on the performance of uniformly quantized MS at SNR=1.5dB.....	41

Fig. 24. BER for code I, the effect of c_{th} on the performance of uniformly quantized MS at SNR=2.0dB	41
Fig. 25. BER for code I, performance of uniformly quantized MS for different quantization bits ($c_{th}=2.0$)	42
Fig. 26 BER for code II (--) and III (—), performance of quantized MS for different bits, with fixed $c_{th}=1.5, 1.25$ respectively	42
Fig. 27. BER (—)/WER (- -) for code I, the effect of c_{th} on performance of uniformly quantized MS at SNR=2.75dB	43
Fig. 28. BER for code I, performance of uniformly quantized MS with $c_{th}=2.0$ for all bits at SNR=1.0—2.5dB and $c_{th}=2.5, 3.0, 3.0$ for 4, 5, 6 bits at SNR=2.75dB	43
Fig.29. BER for code I, the effect of d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=2.0$)	45
Fig. 30. BER for code I, the effect of d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=2.5$)	45
Fig. 31. BER for code I, the effect of d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=3.0$)	45
Fig. 32. BER for code I, the effect of d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=3.5$)	45
Fig. 33. BER for code I, performance of uniformly quantized MS with conditional correction for fixed y_c, d_c and c_{th}	46
Fig. 34. BER (—)/WER (- -) for code I , performance of BP, MS, uniformly quantized MS with unconditional correction, where $c_{th}=2.0$ for 4 bit quantized MS , $c_{th}=3.0$ for 5, 6 bit quantized MS	48
Fig. 35. BER (—)/WER (- -) for code I, performance of BP, MS, uniformly quantized MS with unconditional correction for variable c_{th} , where $c_{th}=2.0$ for 4 bit quantized MS at SNR=1.0-2.5, $c_{th}=3.0$ for 5, 6 bit quantized MS at SNR=1.0-2.5, $c_{th}=2.5, 4.0, 4.0$ for 4, 5, 6 bit quantized MS at SNR=2.75dB respectively	49
Fig. 36. BER for code I, performance of the two modifications for 5 bit uniformly quantized MS with unconditional correction for same parameters of y_c and c_{th}	52
Fig. 37. BER for code I, the performance of MS, uniform quantization with variable c_{th} (all optimal c_{th} values are given in Fig. 28 and Fig. A40) and nonuniform quantization for different quantization bits	60
Fig 38. BER (—) for code I, performance of BP, MS, uniformly and nonuniformly quantized MS with conditional correction for 4 bit case, unconditional correction for 5 and 6 bit cases, where for 4 bit uniformly quantized MS with conditional correction $\{y_c=1, d_c=1, c_{th}=3.0\}$ at SNR=1.0-2.0dB, $\{y_c=1, d_c=0, c_{th}=3.5\}$ at SNR=2.5-2.75dB, for 5, 6 bit uniformly quantized MS with unconditional correction, $c_{th}=3.0$ for all quantization bits at SNR=1.0—2.5dB, at SNR=2.75dB, $c_{th}=4.0$ for 5 and 6 quantization bits	61

Table 3-1. The average number of iterations for different algorithms on code I.....	29
Table 3-2. The average number of iterations for different algorithms on code II	29
Table 4-1. The average iterations for quantization algorithm for (1268, 456) code with $c_{th}=2.0$	50
Table 4-2. The average number of iterations for uniformly quantized MS algorithm with conditional correction for (1268, 456) code with $c_{th}=3.0$	50
Table 4-3. The average number of iterations for uniformly quantized MS algorithm with unconditional correction for (1268, 456) code with $c_{th}=2.0$ for 4 bits and $c_{th}=3.0$ for 5, 6 bits.....	51
Table 5-1. The quantization thresholds (x_i 's) and values (y_i 's) for 3-bit quantization at SNR=2.0dB for code I	56
Table 5-2. The quantization thresholds (x_i 's) and values (y_i 's) for 4-bit quantization at SNR=2.0dB for code I	56
Table 5-3. The addition look-up table for 3-bit quantization at SNR=2.0dB for code I	56
Table 5-4. The addition look-up table for 4-bit quantization at SNR=2.0dB for code I	56
Table 5-5. The subtraction look-up table for 4-bit quantization at SNR=2.0dB for code I ($r=0.2$).....	57
Table 5-6. The quantization thresholds (x_i 's) and values (y_i 's) for 5-bit quantization at SNR=2.0dB for code I (positive value of x_i 's are given).....	57
Table 5-7. The subtraction look-up table for 5-bit quantization at SNR=2.0dB for code I ($r=0.2$)	57
Table 5-8. The average number of iterations of nonuniformly quantized MS for code I.....	63
Table 5-9. The average number of iterations for nonuniformly quantized MS with unconditional correction for code I.....	63

<u>APPENDIX</u>	66
Part 1. MS with modifications	66
Part 2. Clipped MS.....	69
Part 3. Clipped MS with modifications.....	74
Part 4. Uniformly quantized MS.....	84
Part 5. Uniformly quantized MS with conditional correction.....	88
Part 6. More results for conditional correction B	105
Part 7. Uniform quantization with unconditional correction	105
Part 8. More results on Modified MS algorithm of section of 4.5	111
Part 9. Nonuniformly quantized MS and its modifications	113
Part 10. Some tables.....	114

Fig. A1. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of MS with conditional correction at SNR=2.0dB.....	64
Fig. A2. WER for code I, the effect of y_c on performance of MS with unconditional correction at SNR=1.5dB, 2.0dB	65

Fig. A3. WER for code I, performance of MS with conditional and unconditional correction for fixed y_c and d_c	65
Fig. A4. BER (—)/WER (- -) for code II, the effect of y_c and d_c on performance of MS with conditional correction SNR=4.0dB.....	66
Fig. A5. BER (—)/WER (- -) for code II, the effect of y_c on performance of MS with unconditional correction at SNR=4.0dB.....	66
Fig. A6. WER curves for code I (—), II (- -) and III (- . -) decoded by MS (o), clipped MS (no character) and BP (.) with fixed c_{th} =2.0, 1.5 and 1.25 for code I, code II and code III respectively	67
Fig. A7. WER curves for code I (—), II (- -) and III (- . -) decoded by MS (o), clipped MS (no character) and BP (.) with optimal c_{th} =3.5 at SNR=2.5, c_{th} =4.0 at SNR=2.75dB for code I and c_{th} =2.5, 2.5, 3.0 at SNR=4.5, 5.0, 5.5dB for code II.....	67
Fig. A8. The optimal clipping threshold of code I on WER performance of MS at different SNR.....	68
Fig. A 9. BER (—)/WER (- -) for code II, the effect of clipping threshold c_{th} on performance of MS at SNR=3.5dB.....	68
Fig. A10. BER (—)/WER (- -) for code II, the effect of clipping threshold c_{th} on performance of MS at SNR=4.5dB.....	69
Fig. A 11. BER (—)/WER (- -) for code II, the effect of clipping threshold c_{th} on performance of MS at SNR=5.0dB.....	69
Fig. A12. BER (—)/WER (- -) for code II, the effect of clipping threshold c_{th} on performance of MS at SNR=5.5dB.....	70
Fig. A13. The optimal clipping threshold of code II on BER performance of MS at different SNR.....	70
Fig. A14. The optimal clipping threshold of code II on WER performance of MS at different SNR.....	71
Fig. A15. BER (—)/WER (- -) for code III, the effect of clipping threshold c_{th} on performance of MS at SNR=1.9dB.....	71
Fig. A16. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=1.5dB (c_{th} =2.0).....	72
Fig. A17. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=1.5dB (c_{th} =2.5).....	72
Fig. A18. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=1.5dB (c_{th} =3.0).....	73
Fig. A19. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=1.5dB (c_{th} =3.5).....	73
Fig. A20. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=1.5dB (c_{th} =4.0).....	74

Fig. A21. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=2.0dB ($c_{th}=2.0$).....	74
Fig. A22. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=2.0dB ($c_{th}=2.5$).....	75
Fig. A23. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=2.0dB ($c_{th}=3.0$).....	75
Fig. A24. BER (—)/WER (- -) for code I, performance of BP, MS, clipped MS with $c_{th}=2.0$, clipped MS with conditional correction ($c_{th}=3.0$) for fixed y_c , d_c values	76
Fig. A25. BER (—)/WER (- -) for code I, the effect of y_c and c_{th} on performance of clipped MS with unconditional correction at SNR=1.5dB.....	76
Fig. A26. BER (—)/WER (- -) for code I, the effect of y_c and c_{th} on performance of clipped MS with unconditional correction at SNR=2.0dB.....	77
Fig. A27. WER for code I, performance of BP, MS, clipped MS with conditional and unconditional correction for fixed y_c , d_c and c_{th}	77
Fig. A28. WER for code I, decoded by clipped MS with unconditional correction, $c_{th}=3.0$ at SNR=1.0-2.5dB, $c_{th}=4.0$ at SNR=2.75dB.....	78
Fig. A29. BER (—)/WER (- -) for code II, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=4.0dB ($c_{th}=1.5$).....	78
Fig. A30. BER (—)/WER (- -) for code II, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=4.0dB ($c_{th}=2.0$).....	79
Fig. A31. BER (—)/WER (- -) for code II, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=4.0dB ($c_{th}=2.5$).....	79
Fig. A32. BER (—)/WER (- -) for code II, the effect of y_c on performance of clipped MS with unconditional correction at SNR=4.0dB.....	80
Fig. A33. BER (—)/WER (- -) for code II, performance of BP, MS, clipped MS with conditional correction with $y_c=0.1$, $d_c=0.3$, $c_{th}=2.0$	80
Fig. A34. BER (—)/WER (- -) for code II, performance of BP, MS, clipped MS with unconditional correction ($y_c=0.1$, $c_{th}=2.0$)	81
Fig. A35. WER for code III, the effect of y_c , d_c and c_{th} on performance of clipped MS with conditional correction at SNR=1.4dB.....	81
Fig. A36. WER for code I, the effect of c_{th} on performance of quantized MS for different quantization bits at SNR=1.5dB	82
Fig. A37. WER for code I, the effect of c_{th} on performance of quantized MS for different quantization bits at SNR=2.0dB	82
Fig. A38. BER (—)/WER (- -) for code I, the effect of c_{th} on performance of quantized MS for different quantization bits at SNR=2.5dB.....	83
Fig. A39. WER for code I, performance of quantized MS for different quantization bits ($c_{th}=2.0$).....	83

Fig. A40. WER for code I, performance of quantized MS for different quantization bits, where for 4,5,6 quantization bits, at SNR=1.0-2.0dB, $c_{th}=2.0$; for 4 quantization bits at SNR=2.5,2.75dB, $c_{th}=2.5,3.5$ and for 5,6 quantization bits at SNR=2.5,2.75dB, $c_{th}=3.0,4.5$	84
Fig. A41. BER (—)/WER (- -) for, performance of quantized MS for different quantization bits at SNR=3.5dB	84
Fig. A42. BER for code III, the effect of c_{th} on performance of quantized MS for different quantization bits at SNR=1.9dB	85
Fig. A43. WER for code III, the effect of c_{th} on performance of quantized MS for different quantization bits at SNR=1.9dB	85
Fig. A44. WER performance of quantized MS for different quantization bits, where for code II (- -) $c_{th}=1.5$, for code III (—) $c_{th}=1.25$	86
Fig. A45. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=2.0$)	86
Fig. A46. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=2.5$)	87
Fig. A47. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=3.0$)	87
Fig. A48. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=3.5$)	88
Fig. A49. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=2.0$)	88
Fig. A50. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=2.5$)	89
Fig. A51. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=3.0$)	89
Fig. A52. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=3.5$)	90
Fig. A53. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=2.0$)	90
Fig. A54. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=2.5$)	91
Fig. A55. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=3.0$)	91
Fig. A56. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=1.5dB ($c_{th}=3.5$)	92

Fig. A57. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=2.0$)	92
Fig. A58. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=2.5$)	93
Fig. A59. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=3.0$)	93
Fig. A60. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=3.5$)	94
Fig. A61. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=2.0$)	94
Fig. A62. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=2.5$)	95
Fig. A63. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=3.0$)	95
Fig. A64. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=3.5$)	96
Fig. A65. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=2.0$)	96
Fig. A66. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=2.5$)	97
Fig. A67. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=2.0dB ($c_{th}=3.0$)	97
Fig. A68. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.5dB ($c_{th}=2.5$)	98
Fig. A69. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.5dB ($c_{th}=3.0$)	98
Fig. A70. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.5dB ($c_{th}=3.5$)	99
Fig. A71. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.5dB ($c_{th}=4.0$)	99
Fig. A72. BER (—)/WER (---) for code I, performance of BP, MS, quantized MS with conditional correction for different quantization bits with fixed y_c , d_c and c_{th}	100
Fig. A73. BER (—)/WER (---) for code II, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=4.0dB ($c_{th}=2.0$)	100
Fig. A74. BER (—)/WER (---) for code II, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=4.0dB ($c_{th}=2.0$)	101

Fig. A75. BER (—)/WER (- -) for code II, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=4.0dB ($c_{th}=2.0$)	101
Fig. A76. BER (—)/WER (- -) for code II, performance of BP, MS and quantized MS with conditional correction for different quantization bits with fixed y_c , d_c and c_{th}	102
Fig. A77. BER (—)/WER (- -) for code III, performance of BP, MS and 4 bit quantized MS with conditional correction for fixed y_c , d_c and c_{th}	102
Fig. A78. BER (—)/WER (- -) for code I, performance of 5 bit quantized MS with conditional correction (B) for fixed y_c and d_c at SNR=2.0dB ($c_{th}=3.0$)	103
Fig. A79. BER (—)/WER (- -) for code I, the effect of y_c and c_{th} on performance of 4 bit quantized MS with unconditional correction at SNR=1.5dB	103
Fig. A80. BER (—)/WER (- -) for code I, the effect of y_c and c_{th} on performance of 5 bit quantized MS with unconditional correction at SNR=1.5dB	104
Fig. A81. BER (—)/WER (- -) for code I, the effect of y_c and c_{th} on performance of 6 bit quantized MS with unconditional correction at SNR=1.5dB	104
Fig. A82. BER (—)/WER (- -) for code I, the effect of y_c and c_{th} on performance of 4 bit quantized MS with unconditional correction at SNR=2.0dB	105
Fig. A83. BER (—)/WER (- -) for code I, the effect of y_c and c_{th} on performance of 5 bit quantized MS with unconditional correction at SNR=2.0dB	105
Fig. A84. BER (—)/WER (- -) for code I, the effect of y_c and c_{th} on performance of 6 bit quantized MS with unconditional correction at SNR=2.0dB	106
Fig. A85. BER (—)/WER (- -) for code II, the effect of y_c and c_{th} on performance of 4 bit quantized MS with unconditional correction at SNR=4.0dB	106
Fig. A86. BER (—)/WER (- -) for code II, the effect of y_c and c_{th} on performance of 5 bit quantized MS with unconditional correction at SNR=4.0dB	107
Fig. A87. BER (—)/WER (- -) for code II, the effect of y_c and c_{th} on performance of 6 bit quantized MS with unconditional correction at SNR=4.0dB	107
Fig. A88. BER (—)/WER (- -) for code II, performance of BP, MS and quantized MS with unconditional correction for different quantization bits with fixed y_c and c_{th}	108
Fig. A89. BER (—)/WER (- -) for code III, performance of 4 bit quantized MS with conditional correction and 5,6 bit quantized MS with unconditional correction for fixed y_c and d_c	108
Fig. A90. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=2.0dB ($c_{th}=3.0$)	109
Fig. A91. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional at SNR=2.0dB ($c_{th}=3.0$)	109
Fig. A92. BER (—)/WER (- -) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional at SNR=2.0dB ($c_{th}=3.0$)	110
Fig. A93. BER (—)/WER (- -) for code I, the effect of c_{th} and y_c on performance of 5 bit quantized MS with unconditional correction at SNR=2.0dB	110

Fig. A94. WER for code I, performance of BP, MS and 5 bit quantized MS with unconditional correction for method of section 4.5 with fixed y_c and c_{th}	111
Fig. A95. WER for code I, performance of MS and nonuniformly quantized MS for different quantization bits.....	111
Fig. A 96. WER for code I, performance of BP, MS, uniformly and nonuniformly quantized MS with conditional correction for 4 bits cases, unconditional correction for 5 and 6 bits cases, where for 4 bits uniformly quantized MS with conditional correction $\{y_c=1, d_c=1, c_{th}=3.0\}$ at SNR=1.0-2.0dB, $\{y_c=1, d_c=0, c_{th}=4.0\}$ at SNR=2.5dB, $\{y_c=1, d_c=0, c_{th}=4.5\}$ at SNR=2.75dB, for 5, 6 bits uniformly quantized MS with unconditional correction, $c_{th}=3.0$ for all quantization bits at SNR=1.0—2.5dB, at SNR=2.75dB, $c_{th}=4.0$ for 5 and 6 quantization bits	112

List of Abbreviations and Symbols

Abbreviations

LDPC	Low-Density Parity-Check
BP	Belief Propagation
MS	Min-Sum
SNR	Signal-to-Noise Ratio
BER	Bit Error Rate
WER	Word Error Rate
LLR	Log-Likelihood Ratio
AWGN	Additive White Gaussian Noise
BIAWGN	Binary Input Additive White Gaussian Noise
BPSK	Biphase Shift Keying
SP	Sum-Product
APP	A Posteriori Probabilities
MSE	Mean-Squared Error
RC	Rate-Compatible
<i>MAP</i>	Maximum A Posteriori

Symbols

$\frac{N_0}{2}$	power spectral density of AWGN
σ^2	variance of additive Gaussian random variable
E_b	average energy per information bit
R	code rate
k	message length
n	codeword length
a	average energy per transmitted bit
C	codeword
c_i	a bit of a codeword C in the position of i
H	parity-check matrix
y_i	channel output at time i
x_i	channel input at time i
n_i	additive Gaussian random variable
P_i	probability
$P_r(y_s x_s)$	probability density function of y_s conditional on transmitted x_s
$P_r(x_s y_s)$	probability density function of transmitted x_s conditional on the received y_s
$\Pr(x_i y)$	a posteriori probability of x_i given that the received value is y
r_{ji}	a message emitted from a check node j to a bit node i
q_{ij}	a message emitted from a bit node i to a check node j

f_{ij}	ratio of probability
q_{ij}^0	message passed by bit node i to check node j corresponding to " $c_i=0$ ".
q_{ij}^1	message passed by bit node i to check node j corresponding to " $c_i=1$ ".
r_{ji}^0	message passed by check node j to bit node i corresponding to " $c_i=0$ ".
r_{ji}^1	message passed by check node j to bit node i corresponding to " $c_i=1$ ".
R_j	set of all the neighbors of check node j
$R_j \setminus i$	set of all neighbors of check node j excluding the one coming from bit node i
C_i	set of all the neighbors of bit node i
$C_i \setminus j$	set of all neighbors of bit node i excluding the one coming from check node j
$L()$	log function
$\phi()$	$\ln \frac{e^x + 1}{e^x - 1}$
Q_i^0	"pseudoposterior probabilities" corresponding to " $c_i=0$ ".
Q_i^1	"pseudoposterior probabilities" corresponding to " $c_i=1$ ".
k_{ij}	a parameter chosen to ensure $q_{ij}^0 + q_{ij}^1 = 1$
k_i	a parameter chosen to ensure $Q_i^0 + Q_i^1 = 1$
m_i	outgoing message
d	absolute value of difference between two outgoing message

c_{th}	clipping threshold
M_e	set of extrinsic incoming messages
e	node in Tanner graph
$m_{e,min}$	message with the smallest magnitude in M_e
$m_{e,out}$	outgoing message along e
y_c	given non-negative correction value
d_c	given conditional value
q	given quantization bit
$f(\varepsilon)$	defined function
ε	quantization error
D	distortion
$p(x)$	probability density function
Q0	quantization function

Chapter 1

Introduction and Motivation

1. 1 Introduction

Today, modern communication systems play an important role in our lives. The reliability of the systems is crucial. In 1948, Shannon [1] showed how to achieve reliable communication over a noisy transmission channel. A basic scheme for a data transmission system over an unreliable channel is shown in Fig. 1.

Fig. 1. A simple model for a communication system

Since Shannon's work, many encoding and decoding methods for error control in a noisy environment have been proposed by employing error-correcting codes. There are many error-correcting codes. Recently, the invention of turbo codes [7] set a milestone in the development of communication theory. Turbo codes have attained such large improvements in performance that, for many applications, the gap between practical systems and Shannon's theoretical limit is essentially closed. Because of the good performance of Turbo codes, researchers have realized the strengths of iterative decoding schemes and have rediscovered Low-Density Parity-Check (LDPC) codes (LDPC codes were first discovered by R. Gallager in the early 1960s [2]).

This thesis focuses on LDPC codes. These codes have what is technically known as "Near Shannon Limit" performance [3], [4], [5] as well as very good complexity-performance tradeoff. Furthermore, they can beat turbo codes at large block lengths LDPC codes, their decoding algorithms and the advantages over turbo codes are discussed in the next chapter.

Among iterative decoding algorithms for LDPC codes, the belief propagation (BP) algorithm [11], also known as sum-product (SP) [2], [4], [10], is the most famous algorithm and has the best performance. In the last few years, many researchers have worked on reducing the complexity of BP algorithm [25]-[30]. However, the decoding complexity is still a major issue in implementing iterative decoders for commercial applications. This fact led us to investigate another well-known, less complex algorithm: min-sum (MS) algorithm. In this thesis we investigate the performance of clipped MS and two modified MS algorithms (MS with conditional correction and unconditional correction [12], [13], [32], [33]). Furthermore, we discuss the quantized versions of MS and its modifications. Both uniform and nonuniform

quantizations are discussed. Our results complement the results presented in [32], [33]. In some cases, when compared to [32], [33], our simulation results provide a more precise picture of the performance and complexity of MS algorithm and its modifications. Our simulations of modified MS algorithms show that these algorithms have a performance close to BP, and can even outperform BP at high signal-to-noise ratio (SNR). Conditional correction adds some minor complexity to original MS but unconditional correction results in nearly identical complexity. Our simulations of the quantized version of MS algorithm shows that only 4 bit quantization has the same or even better performance than MS over a wide range of SNR. The quantized version of modified MS algorithms also shows that by using enough bits (five or six) for quantization, the algorithm's performance is almost the same as modified MS, and sometimes is even better. Moreover, we show by simulations that more clipping can improve the bit error rate (BER) at low SNR; however, at high SNR, it creates an early error floor. Also, we propose a minor change to modified MS algorithms that slightly improves their performance. Our results show that the performances of nonuniform quantization for MS and its modifications are better than that of uniform quantization at high SNR, but at a cost of complexity in many cases.

The thesis is organized in the following manner: Chapter 1 introduces the topic and presents the motivations for our research. Our simulation platform is also explained in this chapter. Chapter 2 describes LDPC codes, iterative decoding and two LDPC decoding algorithms: BP and MS algorithm. In Chapter 3, we discuss modifications to MS algorithm and study their performance and complexity through simulations. Chapter 4 is focused on the quantized version of MS algorithm and its modifications. The effects of clipping on the algorithms are also studied. We also discuss the effect of variable clipping threshold at different SNR values

for uniform quantization in this chapter, as well as a minor change to modified MS algorithms. Chapter 5 is our investigation for nonuniform quantization of MS algorithm and its modifications. Performance and Complexity analysis results are given through simulation results. Chapter 6 presents our conclusions. More simulation results are presented in the Appendix.

1.2 Communications scenario and simulation platform

In the following, we describe the assumptions for communications scenario and simulations. The code is binary and linear, and all codewords are equally probable. The channel is the binary-input additive white Gaussian noise (BIAWGN) channel. It is stationary and memoryless. For the channel modulation, we use BPSK where “0” is mapped to the signal “1” and “1” is mapped to the signal “-1”. Fig. 2 shows the BIAWGN channel. The input alphabet of the channel is $\{\pm 1\}$. We denote the channel input at time i by x_i , $x_i \in \{\pm 1\}$, and the channel output by y_i . For the BIAWGN channel we have $y_i = x_i + n_i$, where n_i is an independent Gaussian random variable with zero-mean and variance σ^2 .

Fig. 2. The binary-input additive white Gaussian noise channel

For signal-to-noise ratio (SNR), we use the equation $\frac{E_b}{N_0} = \frac{a}{2R\sigma^2}$ [34], where a is the average energy per transmitted bit, R is the code rate, E_b and $\frac{N_0}{2}$ are the average energy per information bit and the power spectral density of AWGN respectively, and σ^2 is the variance of additive Gaussian noise. In our simulations, we always assume that $a = 1$, thus at a given SNR, we have $\sigma = \sqrt{\frac{1}{2R(E_b/N_0)}}$. As we usually express E_b/N_0 in dB, the equation becomes $\sigma = \sqrt{\frac{1}{2R \cdot 10^{(E_b/N_0)/10}}}$. In our simulations, the maximum iteration number for decoding is 200. For each given SNR, a simulation program stops when we have 100 word errors. All the algorithms are in log-likelihood ratio (LLR) domain.

Chapter 2

LDPC codes and LDPC Decoding Algorithms

2. 1 LDPC Codes

Low-density parity-check (LDPC) codes are linear block codes with sparse parity-check matrices. Their matrices are constructed with mostly “0”s and only a small number of “1”s. An LDPC code C is denoted by (n,k) where k and n are the message length and the codeword length respectively. The code rate is defined as $R=k/n$. The parity-check matrix for the code is denoted by H and is $(n-k) \times n$ if the rows are linearly independent. Any given codeword C satisfies the equation $CH^T = 0$.

Micheal Tanner generalized the idea of codes on graphs and the graphical representation of codes [9]. In particular, LDPC codes can be represented by bipartite graphs, referred to as

Tanner graphs in his honor. A bipartite graph is a graph whose nodes are partitioned into two groups. One group is comprised of the bit nodes (or variable nodes). The other contains the check nodes. Fig. 3 shows an example of a parity-check matrix $\mathbf{H}$. The corresponding Tanner graph is also given. In a Tanner graph, each check node represents one parity-check equation (one row of $\mathbf{H}$) and each bit node represents one column of $\mathbf{H}$.

Fig. 3. $\mathbf{H}$ matrix and its Tanner graph

From the example above we can see that every bit node has degree three and every check node has degree six. Such a code is called “regular,” as all the check nodes have the same degree, and all the variable nodes have the same degree. For a given matrix $\mathbf{H}$, if we find that the degrees of all the bit nodes are not the same, or the degrees of all the check nodes are not the same, we call the corresponding LDPC code “irregular.”

2. 2 Comparison of LDPC Codes and Turbo-Codes

LDPC codes have several distinct advantages over turbo-codes. First, iterative-decoding algorithms for LDPC codes are fully parallelizable and may potentially be realized at significantly higher speeds compared to turbo-codes. Second, LDPC codes admit a large family of decoding algorithms, allowing a tradeoff between performance and complexity, while there are only a few decoding algorithms known for turbo codes. Finding a larger class of decoding algorithms for turbo codes that allow one to trade off performance with complexity in a larger range is still an open area of research [5]. Third, convergence of the LDPC decoder to a correct codeword is usually a detectable event.

LDPC codes also have their own problems. One problem is that, for example, the encoding complexity appears to be quite large for large block lengths. A lot of researches have been done for this and recently researches show that efficient encoding of LDPC codes can make encoding complexity become a small fraction of the decoding complexity [15].

To sum up, LDPC codes are potentially serious competitors for turbo codes in many applications.

2. 3 Iterative decoding based on Tanner graph

Iterative decoding algorithms are also called “message passing” algorithms. During decoding, messages pass along the edges of a Tanner graph for a number of iterations. Each iteration starts by sending messages from bit nodes to check nodes and ends by sending messages from check nodes back to bit nodes. Since the number of edges is linear in the length

of the code, it follows that the decoding complexity grows only linearly in the block length. The messages can be hard/soft information depending on the decoding algorithm. This thesis concerns with soft iterative algorithms.

Generally, an iterative algorithm has four steps:

- 1) Perform initialization for each block
- 2) Pass messages from bit nodes to check nodes
- 3) Pass messages from check nodes to bit nodes
- 4) Stop if the estimated hard-decision assignment is a codeword, or a maximum number of iterations is reached. Otherwise, go to step 2.

An iterative decoding algorithm begins with the initialization step: initial messages are created at bit nodes from observation of the channel's output and characteristics. Then these messages are passed to check nodes through the edges of the Tanner graph, and after this, iterations begin. Each iteration consists of two steps: 1) Check nodes process the information received from bit nodes and pass new messages to bit nodes (these messages reflect the estimate of check nodes from the value and reliability of bit nodes (Fig. 4)). 2) Bit nodes process the messages received from check nodes and send back updated messages (information) about their values and the associated reliability to check nodes (Fig. 5). In each iteration, a hard decision on the value of each bit is made at bit nodes and the algorithm stops if the hard decision assignment for bits satisfies all the check constraints, or a maximum number of iterations is reached. A message emitted from a bit node i to a check node j , shown as q_{ij} in Fig. 5, or a check node j to a bit node i , shown as r_{ji} in Fig. 4, along an edge e

depends on all the most recent incoming messages to this node except for the incoming message along edge e .

Fig. 4. Messages passing through check node j to bit node i

Fig. 5. Messages passing through bit node i to check node j

There are two well-known soft-decision iterative decoding algorithms for LDPC codes: one is belief propagation (BP) algorithm, and the other is min-sum [10] or BP-based algorithm [6]. BP algorithm is well known due to its very good performance, but it also has the disadvantage of being too complex to be used in many implementations. MS algorithm greatly reduces the implementation complexity but incurs a degradation of a few tenths of a dB in decoding performance.

2. 4 Belief Propagation and Min-Sum algorithms

Belief propagation (BP) algorithm (also called sum-product (SP) algorithm) is a generalization of the computation of per-symbol a posteriori probabilities (APP) in a trellis. It was first introduced in a different form by Gallager [2] and was generally outlined by Tanner [9]. Another special case is the classic turbo decoder proposed by Borrou [7], which uses APP computation in a trellis.

The min-sum (MS) algorithm [10] is a generalization of the Viterbi algorithm [8]. It was described by Tanner in his Algorithm B [9]. MS algorithm is also called BP-Based algorithm [6] or “max-sum,” [16] depending on the type of messages and the corresponding operations performed in bit and check nodes.

2. 4. 1 Belief propagation (BP) algorithm in probability domain

Considering the AWGN channel with BPSK modulation, each bit c_i of a codeword $C = (c_1, c_2, \dots, c_n)$ is mapped to x_i where $x_i \in \{\pm 1\}$. Then $X = (x_1, x_2, \dots, x_n)$ is transmitted over a binary input additive white Gaussian noise (BIAWGN) channel. At the output of the channel, we have $Y = (y_1, y_2, \dots, y_n)$ where $Y = X + n$. Then we have:

$$P_r(y_i | x_i = 1) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-(y_i-1)^2 / 2\sigma^2}$$

$$P_r(y_i | x_i = -1) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-(y_i+1)^2 / 2\sigma^2}$$

where $P_r(y_i | x_i = 1)$ and $P_r(y_i | x_i = -1)$ are the conditional probability density functions of y_i given that $x_i = 1$ and $x_i = -1$, respectively.

Assuming +1 and -1 are equally likely, the a posteriori probability of x_i given that the received value is y is:

$$\begin{aligned}
\Pr(x_i = x | y) &= \frac{P_r(y | x_i = x) \cdot P_r(x_i = x)}{P_r(y)} \\
&= \frac{\frac{1}{2} \cdot e^{-(y-x)^2/2\sigma^2}}{\frac{1}{2} \cdot e^{-(y-1)^2/2\sigma^2} + \frac{1}{2} \cdot e^{-(y+1)^2/2\sigma^2}} \\
&= \frac{e^{xy/\sigma^2}}{e^{y/\sigma^2} + e^{-y/\sigma^2}} \\
&= \frac{1}{(e^{y/\sigma^2} + e^{-y/\sigma^2}) / e^{xy/\sigma^2}} \\
&= \frac{1}{e^{y(1-x)/\sigma^2} + e^{-y(1+x)/\sigma^2}} \\
&= \frac{1}{1 + e^{-2xy/\sigma^2}}
\end{aligned}$$

BP algorithm in APP domain is then described in the following steps [18]:

- 1) Initialization: Each bit node i is assigned APP q_{ij}^0 and q_{ij}^1 , which are sent to check node j

as the initial message. q_{ij}^0 and q_{ij}^1 are initialized by

$$\begin{aligned}
q_{ij}^0 &= (1 - P_i) = \Pr(x_i = 1 | y_i) = \frac{1}{1 + e^{-2y_i/\sigma^2}} \\
q_{ij}^1 &= P_i = \Pr(x_i = -1 | y_i) = \frac{1}{1 + e^{2y_i/\sigma^2}}
\end{aligned}$$

where P_i is the probability that the transmitted x_i is a -1 conditional on the received y_i ,

q_{ij}^0 and q_{ij}^1 are the messages passed by bit node i to check node j corresponding to “ $c_i=0$ ” and “ $c_i=1$ ”, respectively.

- 2) Check node step: Compute outgoing messages for “ $c_i=0$ ” and “ $c_i=1$ ”

$$r_{ji}^0 = \frac{1}{2} + \frac{1}{2} \cdot \prod_{i' \in R_j \setminus i} (1 - 2q_{i'j}^1)$$

$$r_{ji}^1 = \frac{1}{2} - \frac{1}{2} \cdot \prod_{i' \in R_j \setminus i} (1 - 2q_{i'j}^1)$$

where r_{ji}^0 and r_{ji}^1 are the messages passed by check node j to bit node i corresponding to “ $c_i=0$ ” and “ $c_i=1$ ”, respectively. $R_j \setminus i$ is the set of all the neighbors of check node j excluding bit node i .

- 3) Bit node step: Update q_{ij}^0 and q_{ij}^1 corresponding to “ $c_i=0$ ” and “ $c_i=1$ ” respectively by

$$q_{ij}^0 = k_{ij} (1 - P_i) \prod_{j' \in C_i \setminus j} r_{ji'}^0$$

$$q_{ij}^1 = k_{ij} P_i \prod_{j' \in C_i \setminus j} r_{ji'}^1$$

where $C_i \setminus j$ is the set of all the neighbors of bit node i excluding the one coming from check node j . k_{ij} is chosen to ensure $q_{ij}^0 + q_{ij}^1 = 1$.

- 4) Calculate the information that will be used for decision:

$$Q_i^0 = k_i (1 - P_i) \prod_{j \in C_i} r_{ji}^0$$

$$Q_i^1 = k_i P_i \prod_{j \in C_i} r_{ji}^1$$

where Q_i^0 and Q_i^1 are the “pseudoposterior probability” corresponding to “ $c_i=0$ ” and “ $c_i=1$ ”, respectively. k_i is chosen to ensure $Q_i^0 + Q_i^1 = 1$.

- 5) Hard decision and stopping criteria:

$$\hat{c}_i = \begin{cases} 1 & \text{if } Q_i^1 > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Stop if the estimated hard-decision assignment, consisting of $\hat{c}_i$'s, is a codeword, or a maximum number of iterations is reached. Otherwise go back to step 2.

2. 4. 2 Belief propagation (BP) algorithm in Log Likelihood Ratio (LLR) domain

The description of BP algorithm in the previous section shows that multiplication is involved in the probability domain. When the code length and iteration increase, multiplication could become costly. Since multiplication in probability domain is converted into addition in LLR domain, algorithms become less costly. In this thesis, all algorithms are in LLR domain.

In LLR domain, the ratio of probabilities is defined as $q_{ij} = q_{ij}^0 / q_{ij}^1$. BP algorithm has the following steps [18]:

1) Initialization: initialize the bit node's outgoing message $L(q_{ij})$ by the observation of received values (y_i)

$$L(q_{ij}) = L(c_i) = \ln \frac{q_{ij}^0}{q_{ij}^1} = \ln \frac{1 + e^{2y_i / \sigma^2}}{1 + e^{-2y_i / \sigma^2}} = \ln \frac{e^{2y_i / \sigma^2} (1 + e^{-2y_i / \sigma^2})}{1 + e^{-2y_i / \sigma^2}} = 2y_i / \sigma^2$$

where $L()$ represents the log likelihood ratios function.

2) Check node step:

$$L(r_{ji}) = \prod_{i' \in R_j \setminus i} \text{sgn}(L(q_{i'j})) \cdot \Phi \left(\sum_{i' \in R_j \setminus i} \Phi |L(q_{i'j})| \right)$$

where

$$\Phi(x) = -\ln \tanh(x/2) = \ln \frac{e^x + 1}{e^x - 1}$$

$\text{sgn}(x)=1$ or -1 , depending on $x \geq 0$ or $x < 0$, respectively

3) Bit node step:

$$L(q_{ij}) = L(c_i) + \sum_{j' \in C_i \setminus j} L(r_{ji'})$$

4) Calculate the information that will be used for decision:

$$L(Q_i) = L(c_i) + \sum_{j \in C_i} L(r_{ji})$$

5) Hard decision and stopping criteria:

$$\hat{c}_i = \begin{cases} 1 & \text{if } L(Q_i) < 0 \\ 0 & \text{otherwise} \end{cases}$$

Stop if the estimated hard-decision assignment, consisting of $\hat{c}_i$'s, is a codeword, or a maximum number of iterations is reached. Otherwise go back to step 2.

2.4.3 MS algorithm in LLR domain

MS algorithm has the following steps [18]:

1) Initialization: initialize the bit node's outgoing message value $L(q_{ij})$ by the observation of received channel value y_i :

$$L(q_{ij}) = L(c_i) = 2y_i / \sigma^2$$

As we can see, $2/\sigma^2$ is a constant at a given SNR. If $L(q_{ij})$ is divided by $2/\sigma^2$, the initial equation becomes $L(q_{ij}) = y_i$. MS algorithm using this initial equation is called the normalized MS algorithm. It was proved that normalized MS has the same performance as MS [12].

The normalized MS initial equation shows that the algorithm does not depend on σ^2 . Therefore, in hardware implementation, no a priori information about the AWGN channel is required. This offers a significant advantage for hardware implementation.

In our simulations, we use the normalized MS algorithm in LLR domain.

2) Check node step:

$$L(r_{ji}) = \prod_{i' \in R_j \setminus i} \text{sgn}(L(q_{i'j})) \cdot \text{Min}_{i' \in R_j \setminus i} L(q_{i'j})$$

3) Bit node step:

$$L(q_{ij}) = L(c_i) + \sum_{j' \in C_i \setminus j} L(r_{ji'})$$

4) Calculate the information that will be used for decision:

$$L(Q_i) = L(c_i) + \sum_{j \in C_i} L(r_{ji})$$

5) Hard decision and stopping criteria:

$$\hat{c}_i = \begin{cases} 1 & \text{if } L(Q_i) < 0 \\ 0 & \text{otherwise} \end{cases}$$

Stop if the estimated hard-decision assignment, consisting of $\hat{c}_i$'s, is a codeword, or a maximum number of iterations is reached. Otherwise go back to step 2.

From MS algorithm we can see that MS algorithm is less complex compared to BP as the latter one employs log function in arithmetic operations for all the extrinsic incoming messages at each check node.

Chapter 3

Modifications of Min-Sum (MS) Algorithm

3.1 Overview

In this chapter, modifications to MS algorithm are fully discussed.

For a given code, a channel model and a prescribed error rate, MS requires up to a few tenths of a dB more transmitted power than BP. The performance of MS algorithm can be improved by introducing different modifications to this algorithm [12], [13], [32], [33]. This thesis undertakes a full investigation of the methods with extensive simulations on several LDPC codes. Our results show that, some of the simulation results presented in [32], [33] are not very precise. Moreover, in many cases, not enough simulations have been performed in [32], [33] to provide a complete picture of the performance/complexity tradeoff for different

algorithms and obtain optimal parameters for them. Furthermore, we investigate the performance of these algorithms at the error floor region at higher SNR values. This is again something that has not been properly treated in [32], [33]. The complexity of the methods is also investigated. Detailed discussions on our simulation results are also provided. We also introduce another slightly altered algorithm that results in small improvement in performance.

3. 2 Modified MS Algorithm

Inspection of MS and BP algorithms in LLR domain reveals that the two algorithms perform the exact same operation in bit nodes (addition) and that the difference is only in the operations performed in check nodes. Suppose that we have a check node p with degree 3, and let m_1 and m_2 be the input message to p with $|m_1| < |m_2|$. Also, let m_3 denote an outgoing message. From BP algorithm, we have [32], [33]:

$$\begin{aligned}
 m_3 &= \text{sgn}(m_1) \text{sgn}(m_2) \ln \frac{1 + e^{|m_1|} e^{|m_2|}}{e^{|m_1|} + e^{|m_2|}} \\
 &= \text{sgn}(m_1) \text{sgn}(m_2) \ln \left(e^{|m_1|} \cdot \frac{e^{-2|m_1|} + e^{|m_2| - |m_1|}}{1 + e^{|m_2| - |m_1|}} \right) \\
 &= \text{sgn}(m_1) \text{sgn}(m_2) \left[|m_1| + \ln \frac{e^{-2|m_1|} + e^d}{1 + e^d} \right]
 \end{aligned} \tag{1}$$

where $d = |m_2| - |m_1| > 0$

While for MS, $m_3 = \text{sgn}(m_1) \text{sgn}(m_2) |m_1|$. The two messages have the same sign but a different magnitude [28].

Now, assuming $e^{-2|m_1|} \ll e^d$ (high SNR regime), $|m_3|$ in (1) can be simplified as:

$$|m_3| \approx \max(|m_1| - \ln(1 + e^{-d}), 0) \quad (2)$$

The function $-\ln(1 + e^{-d})$ is plotted in Fig. 6. The curve shows that the function value is always negative and larger than -0.69 for $d \geq 0$. We can see that the term $f(d) = -\ln(1 + e^{-d})$, which is only a function of the magnitude of the difference between the incoming messages, is the correction factor. This factor should be added to the magnitude of the output message of the check node in MS algorithm in order to make it approximately the same as that of BP.

Fig. 6. Correction factor $f(d)$ for min-sum algorithm.

Applying this to MS is complicated. However, the function $-\ln(1 + e^{-d})$ can be implemented using some simple approaches, such as using a look-up table or a piece-wise linear approximation [27], or a stepwise function.

Our simulations are based on the following simple stepwise approximation for the correction factor that has been proposed by [32], [33]:

$$f(d) \approx \begin{cases} -y_c & \text{if } d \leq d_c, \\ 0 & \text{otherwise.} \end{cases}$$

where y_c is a given non-negative constant correction factor and d_c is a given non-negative constant condition factor.

The correction factor is applied to the magnitude of the outgoing MS messages from check nodes based on the proposed correction steps (signs remain the same as those in conventional MS).

Two correction methods have been proposed: conditional correction [32], [33] and unconditional correction [12], [13], [32], [33].

- i) **Conditional correction:** For each edge e , let M_e denote the set of extrinsic incoming messages. Also, use the notations $m_{e,\min}$ and $m_{e,\text{out}}$ to denote the message with the smallest magnitude in M_e , and the outgoing message along e , respectively. Based on MS algorithm, we have $|m_{e,\text{out}}| = |m_{e,\min}|$. The correction starts by removing $m_{e,\min}$ from M_e , and performing the following step iteratively until the set M_e is empty: Randomly select a message m from M_e , and update $|m_{e,\text{out}}|$ with $\max(|m_{e,\text{out}}| - y_c, 0)$, if $|m| \in [|m_{e,\text{out}}|, |m_{e,\text{out}}| + d_c]$, for some predetermined positive and non-negative real numbers y_c and d_c , respectively. Otherwise, leave $|m_{e,\text{out}}|$ unchanged. Remove m from M_e .

It is important to randomly select a message m from M_e . In conditional correction, for some predetermined positive and non-negative real numbers y_c and d_c , the algorithm will be associative or non-associative depending on the relationship between y_c and d_c .

When $y_c < d_c$, it can be seen that the algorithm is not associative [32]. In this case, the final corrected outgoing message depends on the order of extrinsic messages being selected. For example, assume a check node has the degree 4, for $y_c=0.1$ and $d_c=0.2$, we have three incoming messages with absolute values $|m_1|=0.4$, $|m_2|=0.45$ and $|m_3|=0.55$. If the order of message selection is $|m_1| \rightarrow |m_2| \rightarrow |m_3|$, the absolute outgoing message value is $|m_{e,out}|=0.3$. If the order of message selection is $|m_1| \rightarrow |m_3| \rightarrow |m_2|$, the absolute outgoing message value is $|m_{e,out}|=0.2$. The final absolute value depends on the order of the incoming messages. Thus the algorithm is not associative under these conditions.

It can be seen that the algorithm is associative when $y_c \geq d_c$ [32]. In this case, the final corrected outgoing message values are independent of the order of extrinsic messages being selected. Let's look at the given example above when the parameters are given as $y_c=0.2$ and $d_c=0.1$. Regardless of which order messages are selected in, the final outgoing absolute value is $|m_{e,out}|=0.2$. In fact, in this case the correction factor, $-y_c$, is applied at most once for each outgoing message.

To avoid the associative issue for a given code in our simulations, we randomly select a message m from M_e in our algorithms.

3.3 Simulation results and discussion

Simulations are done to find the optimal parameters y_c and d_c (begin at 0.1 with increasing step 0.1) for conditional correction. Fig. 7 shows the performance of MS (unclipped & unquantized Min-Sum) on an irregular LDPC code (1268, 456) [14] (we label it as code I) with different combinations of y_c and d_c at SNR=1.5dB. More simulations on code I at SNR=2.0dB and on a short length, regular LDPC code (273, 191) [36] (we label it as code II) at SNR=4.5dB are given in the Appendix.

From Fig. 7 we can see that for a given y_c , there is always one or more optimal points for d_c on the performance curves. Comparing these points with those in bit error rate (BER) curves and word error rate (WER) curves at 2.0dB, the latter curves are given in Fig. A1, we see that the optimal values of y_c are more or less the same, i.e., the optimal values of y_c are not very sensitive to SNR. Same thing happens for d_c . This is consistent with [32]. We also observe that for each value of y_c , even for $y_c=0.1$, the performance worsens as d_c is increased to very larger values. We observe that optimal performance is, on the whole, achieved by using smaller correction values with proper conditional values, such as the point $\{y_c=0.1, d_c=2.5\}$ or $\{y_c=0.2, d_c=1.0\}$ in Fig. 7. This shows that smaller corrections with proper conditions are more accurate in correcting the outgoing messages at check nodes in MS algorithm. This is different from [32]. Since the simulations on code I have only been done in a smaller range of [0.1 0.9] for y_c at 1.5dB in [32], the parameters used there are not truly optimal, and the following simulations and discussions are not precise.

Fig. 7. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of MS with conditional correction at SNR=1.5dB

Fig. 8. BER (—)/WER (---) for code I, performance of MS with conditional correction for fixed y_c and d_c

Fig. 8 shows the performance of conditional correction for MS with fixed parameter values combination $\{y_c=0.1, d_c=2.5\}$ and $\{y_c=0.2, d_c=1.0\}$. The results show that MS algorithm with conditional correction fills the gap between MS and BP algorithms over a relatively large range of SNR. These were not observed in [32], as the simulations in [32], were not performed at 2.75dB. The performance of all algorithms at high SNR region when an error floor appears is missing in [32], thus some conclusions are not precise in [32].

The conditional correction for MS can improve MS but at the expense of increasing the complexity of MS. To simplify this, we can apply the correction factor $-y_c$ only once if there exists a message m in $M_e \setminus \{m_{e,\min}\}$ such that $|m| \in [|m_{e,\min}|, |m_{e,\min}| + d_c]$. We distinguish the two conditional corrections by labels A and B, where A refers to method (i) described in page 21. Our simulations show that for both corrections, the performance, when optimized, is more or less the same. The results for method B in quantized version are given in Fig. A78 of the Appendix. In this thesis, we use method A for all modified MS algorithms in order to compare them with [32]. From method B we can see when $d_c \rightarrow \infty$ (assuming that all check nodes have degree at least 3), the correction will apply only once. In fact, $d_c \rightarrow \infty$ makes the conditional correction become unconditional. This leads us to investigate another algorithm that is proposed in [12], [13], [32], [33]:

- ii) **Unconditional correction:** For each edge e , the magnitude of the outgoing message is $|m_{e,\text{out}}| = \max(|m_{e,\min}| - y_c, 0)$, for some predetermined positive real number y_c .

Fig. 9. BER for code I, the effect of y_c on performance of MS with unconditional correction at SNR=1.5, 2.0dB

Fig. 10. BER for code I, performance of MS with conditional and unconditional correction with fixed y_c and d_c

Fig. 9 is the simulation that finds the optimal parameter values at SNR=1.5 and 2.0dB for code I. No similar simulations are presented in [32]. Fig. 9 shows that the optimal values of y_c are almost the same for SNR=1.5 and 2.0dB. They are not very sensitive to SNR. Simulations are performed to investigate the performance of MS algorithm with unconditional correction by using the best parameter values based on this observation. Fig. 10 is our simulation results for code I. The best result for conditional correction is also given as a comparison. The corresponding WER results are given in Fig. A2—Fig. A3 of the Appendix.

Fig. 10 shows that, by using the optimal parameters, the performance of MS with conditional correction is better than MS with unconditional correction at lower SNR values, but is worse at higher SNR values. Our simulations show that the conditional correction method creates the error floor at a high SNR range while the unconditional correction method does not. In general, the conditional correction method should give better results than the unconditional correction method, as the latter is a special case of the former. However one should notice here that the optimal values of y_c and d_c for conditional correction have been chosen based on the results at 1.5 and 2.0dB. They are however not optimal for larger SNR values. Interestingly, the unconditional correction method outperforms BP at a higher SNR range. This can happen, since we know that BP algorithm only results in the best performance for LDPC codes that are cycle-free. More simulation results on other codes also support this point, as we can see the results for code II in Fig. 11 and Fig. 12. The simulation results in finding the optimal parameters for code II are given in Fig. A4, Fig. A5 of the Appendix.

Fig. 11. BER (—)/WER (---) for code II, performance of BP, MS, MS with conditional correction with $\gamma_c=0.1$, $d_c=0.3$

Fig. 12. BER (—)/WER (---) for code II, performance of BP, MS, MS with unconditional correction ($\gamma_c=0.1$)

3. 4 Complexity analysis results

In this section, the average number of iterations for the converged cases, as a measure of complexity for each algorithm is studied. Table 3-1 gives the average number of iterations of the algorithms for code I. The results for code II are given in Table 3- 2. It can be seen that, by using the modified MS algorithms, the average number of iterations is reduced, particularly greatly reduced at lower SNR values, compared to the MS algorithm.

Eb/No (dB)	MS Algorithm	MS with conditional correction $\gamma_c = 0.2$, $d_c = 1.0$	MS with unconditional correction $\gamma_c = 0.1$
1.0	41.860	25.542	28.546
1.5	26.062	16.253	17.820
2.0	12.676	10.574	11.311
2.5	8.837	8.047	8.416
2.75	7.571	7.225	7.481

Table 3-1. The average number of iterations for different algorithms on code I

Eb/No (dB)	MS Algorithm	MS with conditional correction $\gamma_c = 0.2$, $d_c = 1.0$	MS with unconditional correction $\gamma_c = 0.1$
2.0	25.506	13.923	15.648
2.5	15.878	8.722	9.174
3.0	8.581	5.846	5.711
3.5	4.360	3.451	3.517
4.0	2.585	2.393	2.402
4.5	1.824	1.822	1.814
5.0	1.474	1.460	1.450

Table 3- 2. The average number of iterations for different algorithms on code II

Chapter 4

Uniformly quantized Min-Sum (MS) and its Modifications

4. 1 Overview

As MS algorithm is less complex than BP algorithm, and our analysis in the previous chapter shows that modified MS algorithms can close the gap between the performance of MS and BP algorithms. The MS algorithm seems to be more suitable for many applications. In this chapter, the effects of uniform quantization on MS algorithm and its modifications are investigated.

In recent years, researchers have addressed the quantization issues related to iterative decoding algorithms [13], [17], [32], [33]. Paper [17] investigates the quantization effect on

BP algorithm. Paper [13] discusses MS algorithm with quantization. This work uses density evolution which applies to very large block lengths. The focus of our work, as well as [32] and [33], is short block lengths.

4. 2 Effect of clipping

Before we go further into quantization, we will discuss the effect of clipping on MS algorithm and its modified versions. It is obvious that clipping is mandatory in a quantization scheme. In our simulations, we apply clipping to all the outgoing messages from bit nodes by a given clipping threshold (c_{th}).

From equation (1) in chapter 3, we see that the difference between outgoing messages in MS and BP algorithms at check nodes is:

$$|m_3| - |m_1| = \ln \frac{e^{-2|m_1|} + e^d}{1 + e^d}$$

Since $|m_1| \geq 0$, and we have $e^{-2|m_1|} \leq 1$, then

$$\ln \frac{e^{-2|m_1|} + e^d}{1 + e^d} \leq \ln \frac{1 + e^d}{1 + e^d} = 0$$

That means the outgoing message $|m_3|$ in BP is always smaller than the outgoing message $|m_1|$ in MS at check nodes. Although in MS algorithm clipping is performed only on bit nodes' outgoing messages, it somehow compensates the oversized check nodes' outgoing messages and improves the performance of MS algorithm.

4.2.1 Effects of clipping on MS algorithm

Simulations are done to investigate the clipped MS performance on three codes: code I, code II and a long regular (8000, 4000) LDPC code [36] (we label it as code III). The results for the (1268, 456) code are presented in the main body of the thesis and results for the other codes are given in Fig. A 9—Fig. A15 of the Appendix.

Fig. 13

Fig. 14

Fig. 13. BER (—)/WER (---) performance for code I for different clipping threshold at SNR=1.5dB
Fig. 14. BER (—)/WER (---) performance for code I for different clipping threshold at SNR=2.0dB

First, the optimal clipping thresholds (c_{th}), which begins at 1.0 with increasing step 0.5, on three codes are investigated. Fig. 13 and Fig. 14 are the results of the clipped MS algorithm on code I for different clipping thresholds at SNR=1.5 and 2.0dB. More results for the other two codes are given in the Appendix (see Fig. A 9—Fig. A15). Fig. 13 and Fig. 14 show that the optimal clipping thresholds are almost the same at different SNR. This result is consistent with [32]: the optimal clipping threshold is not very sensitive to SNR; for a wide range of E_b/N_0 , the optimal clipping thresholds are 2.0, 1.5 and 1.25 for code I, code II and code III respectively. For the regular (273, 191) code however, the optimal threshold is not as distinguishable as it is for other codes.

Fig. 15 shows our simulation results on these three codes by using the optimal clipping thresholds we mentioned above. For code I, the performance of clipped MS is the same as results in [32] over the range [1.0dB, 2.5dB]. However, at SNR=2.75dB, the performance of the clipped MS algorithm becomes worse than MS. Clipping creates an early error floor. Simulations on code II also show that clipping creates the error floor at a higher SNR range, 5.0–5.5dB. For code III, we have not observed the error floor for clipped MS up to BER=10⁻⁶.

Fig. 15. BER for code I (—), II (--) and III (-.-), performance of MS (○), clipped MS (no character) and BP (●)

Based on these observations, more simulations are done for code I and code II at higher SNR values with different clipping thresholds. Fig. 16 shows that at SNR=2.5dB for code I, the optimal clipping threshold for BER&WER are in fact not 2.0, but are almost 2.5 and 3.5 respectively. Fig. 16 also shows, at SNR=2.75dB, the optimal clipping threshold for BER&WER curves changes from 2.0 at lower SNR to 3.5 and 4.5 at higher SNR respectively.

Fig. 17 shows how the optimal clipping threshold for BER changes with SNR for code I. WER curve for code I is given in Fig. A8 of the Appendix. All simulation results on code II are given in Fig. A 9—Fig. A14 of the Appendix. So the optimal clipping thresholds in fact increase with SNR as Fig. A13, Fig. A14, Fig. 17 and Fig. A8 indicate.

Fig. 16. BER/WER for code I, the effect C_{th} on performance of clipped MS at SNR=2.5, 2.75dB

Fig. 17. The optimal C_{th} values at different SNR for BER performance of MS for code I

Fig. 18. BER curves for code I (—), II (---) and III (-.-), decoded by MS (o), clipped MS (no character) and BP (●). The optimal C_{th} is 2.0, 2.0, 2.0, 2.5, 3.5 at SNR=1.0, 1.5, 2.0, 2.5, 2.75dB respectively for code I. The optimal C_{th} is 1.5 at SNR=2.0-4.0dB and 2.5 at SNR=4.5-5.5dB for code II. The optimal C_{th} is 1.25 for all SNR values for code III

Fig. 18 shows that, by choosing the optimal c_{th} at different SNR, the error floor effect on clipped MS performance is degraded compared to Fig. 15 which uses a constant c_{th} independent of the SNR for each LDPC code. Clipping creates an early error floor but we can improve the error floor by using variable clipping. This is not discussed in [32].

4.2.2 Effects of clipping on modified MS algorithms

The optimal c_{th} for conditional correction and unconditional correction are investigated by extensive simulations on code I at SNR=1.5 and 2.0dB with different c_{th} , y_c and d_c . All the results are given in Fig. A16—Fig. A23 of the Appendix. The optimal values of c_{th} for conditional correction and unconditional correction are almost the same at SNR=1.5 and

2.0dB and are equal to $c_{th}=3.0$. This is different than the optimal c_{th} for clipped MS at the same SNR, which is in fact 2.0.

Fig. 19. BER for code I, decoded by clipped MS with conditional and unconditional correction with fixed $c_{th}=3.0$

Simulation results are given in Fig. 19 by using fixed optimal parameter combinations for conditional and unconditional correction for MS algorithm. The result shows that for conditional and unconditional correction versions of clipped MS, both methods perform well. They fill the performance gap separating MS and BP algorithms, and at higher SNR values, they outperform BP. This however is not the case for WER as Fig. A27 indicates. We also observe that the modified MS algorithms with clipping have better performance than unclipped versions except that at the last point, SNR=2.75dB, unconditional correction with clipping performs worse than the unclipped version. This is because clipping in general worsens the error floor. As we discussed before, using an optimal c_{th} at a given SNR can

improve the error floor. Fig. 20 shows the result when we apply optimal $c_{th}=4.0$ at SNR=2.75dB for unconditional correction. Compared with Fig. 19, we can see that unconditional correction performance is improved. WER curve is given in Fig. A28 of the Appendix. The results show that when using fixed y_c , optimal clipping thresholds are same for BER and WER.

Fig. 20. BER for code I, decoded by clipped MS with unconditional correction $y_c=0.2$ with optimal $c_{th}=3.0$ at SNR=1.0-2.5dB, optimal $c_{th}=4.0$ at SNR=2.75dB

4.2.3 Another clipping method

In this section, the effect of clipping is further studied by another proposed clipping method that effectively applies more clipping at bit nodes.

In this new method, we apply clipping after adding each two extrinsic messages at bit nodes. That means we perform the following operation at bit nodes for MS algorithm:

For example, given the bit nodes and check nodes with $i=1$ and $j'=1, 2, 3, 4, 5\dots$, in the calculation at bit node, as the equation $L(q_{ij}) = L(c_i) + \sum_{j' \in C_i \setminus j} L(r_{j'})$ states, we first add $L(r_{11})$ & $L(r_{21})$, and then clip the result. Then add the answer to $L(r_{31})$ and perform clipping again, and so forth. This method effectively introduces more clipping than the method we used before.

Fig. 21. BER for code I, uniformly quantized MS using clipping method of section 4.2 (—) and section 4.2.3 (---)
($C_{th}=2.0$)

Fig. 21 shows the results for our new clipping algorithm for quantized MS (quantization will be explained in more detail in section 4.3). From the results we observe that, at low SNR

values, this new method improves the performance compared with the previous approach. However, as SNR increases, the performance becomes worse than the previous quantized MS. This was in line with our expectations, that more clipping improves the performance at low SNR values, but worsens the error floor.

4.3 Uniformly quantized MS algorithm

In this section, uniformly quantized MS performance with different number of quantization bits is investigated.

The received values are clipped symmetrically at a threshold c_{th} , and then uniformly quantized in the range $[-c_{th}, c_{th}]$. There are $2^q - 1$ quantization intervals, each represented by q quantization bits. The central quantization interval is symmetric with respect to the origin and is represented by 0. The rest of the intervals are pair-wise symmetric with respect to the origin and are represented by integer numbers with opposite signs. Fig. 22 gives an example of 3 bit uniform quantization.

Fig. 22. Scheme of 3 bit uniform quantization

In the decoding algorithm, all the operations are integer calculations. If the summation at bit nodes results in an integer larger than $2^q - 1$ or smaller than $-2^q - 1$, the result is clipped to $2^q - 1$ or $-2^q - 1$, respectively.

4.3. 1 Uniformly quantized MS algorithm and its performance

Quantized MS algorithm comprises the following steps:

1) Initialization:

For received channel value y_i , apply clipping for the given threshold c_{th} to make y_i satisfy the condition $y_i \in [-c_{th} \ c_{th}]$, then quantize y_i as:

$$y_i \rightarrow y_i \in [-(2^{q-1} - 1), \dots, -1, 0, 1, \dots, (2^{q-1} - 1)]$$

and initiate the function $L(q_{ij}) = L(c_i) = y_i$

2) Perform bit node and check node operations the same as MS algorithm

3) After the bit node operation $L(q_{ij}) = L(c_i) + \sum_{j' \in C_i \setminus j} L(r_{ji'})$, apply clipping with threshold

$2^q - 1$:

$$L(q_{ij}) \rightarrow L(q_{ij}) \in [-(2^{q-1} - 1), \dots, -1, 0, 1, \dots, (2^{q-1} - 1)]$$

4) After the operation $L(Q_i) = L(c_i) + \sum_{j \in C_i} L(r_{ji})$, apply clipping with threshold $2^q - 1$:

$$L(Q_i) \rightarrow L(Q_i) \in [-(2^{q-1} - 1), \dots, -1, 0, 1, \dots, (2^{q-1} - 1)]$$

5) Repeat the iteration the same as that of MS algorithm

Fig. 23

Fig. 23. BER for code I, the effect of C_{th} on the performance of uniformly quantized MS at SNR=1.5dB

Fig. 24

Fig. 24. BER for code I, the effect of C_{th} on the performance of uniformly quantized MS at SNR=2.0dB

Simulations are done on code I, II and III. Fig. 23 and Fig. 24 show our simulations for different clipping thresholds on code I at SNR=1.5 and 2.0dB. More simulations on code I, code II and code III are given in Fig. A36—Fig. A39, Fig. A41—Fig. A43 of the Appendix.

From Fig. 23 and Fig. 24 we can see that the optimal c_{th} is almost the same for different number of quantization bits at a given SNR. They also show that at SNR=1.5 and 2.0dB, optimal clipping thresholds are the same, i.e., $c_{th}=2.0$. This is consistent with the results of [32], [33] which indicate that, optimal c_{th} is not very sensitive to SNR. However, we also know that, using a fixed c_{th} value, which is optimal for lower SNR values, the algorithm would create an early error floor at higher SNR. This can be seen in Fig. 25 and Fig. 26 for BER, and Fig. A39, Fig. A44 of the Appendix for WER. The results are also consistent with those of [32], [33] in SNR range [1.0dB 2.5dB] but the error floor wasn't observed in simulations of [32], [33] as the results at SNR=2.75dB for code I and at SNR=5.0-5.5dB for code II are missing.

Fig. 25. BER for code I, performance of uniformly quantized MS for different quantization bits ($C_{th}=2.0$)

Fig. 26 BER for code II (- -) and III (—), performance of quantized MS for different bits, with fixed $C_{th}=1.5, 1.25$ respectively

Fig. 27. BER (—)/WER (--) for code I, the effect of C_{th} on performance of uniformly quantized MS at SNR=2.75dB

Fig. 28. BER for code I, performance of uniformly quantized MS with $C_{th}=2.0$ for all bits at SNR=1.0—2.5dB and $C_{th}=2.5, 3.0, 3.0$ for 4, 5, 6 bits at SNR=2.75dB

The optimal clipping thresholds for 4, 5 and 6 quantization bits at SNR=2.75dB for code I are shown in Fig. 27. It can be seen that at SNR=2.75dB, for BER, the optimal value of c_{th} is about 2.5, 3.0 and 3.0 for 4, 5 and 6 bits respectively. Similar results for SNR=2.5dB, and for WER are given in Fig. A38—Fig. A39 of the Appendix.

Fig. 28 is the result of using (variable) optimal clipping thresholds and quantization. The simulation results show that by applying optimal clipping thresholds, the error floor improved considerably compared to Fig. 25. In general, as we observed before, more clipping improves the performance at low SNR values while degrading it at high SNR values. Less clipping has the opposite effect.

4.3. 2 Uniformly quantized MS with modifications

In this section, the performance of quantized MS with conditional correction and unconditional correction is investigated.

We have described modified MS and clipped MS algorithms in detail in previous section. The quantized MS algorithms with modifications are just the same as these algorithms except that all the messages, as well as the parameter y_c , are represented by integer values, and all the operations at bit nodes and check nodes are integer operations.

4.3.2. 1 Quantized MS with conditional correction

Simulations in finding the optimal c_{th} , y_c and d_c for the quantized MS algorithm with conditional correction are done on code I and code II.

Fig. 29

Fig. 30

Fig.29. BER for code I, the effect of d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($C_{th}=2.0$)

Fig. 30. BER for code I, the effect of d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($C_{th}=2.5$)

Fig. 31

Fig. 32

Fig. 31. BER for code I, the effect of d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($C_{th}=3.0$)

Fig. 32. BER for code I, the effect of d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.0dB ($C_{th}=3.5$)

Fig.29 – Fig. 32 show our simulation results with different y_c , d_c and c_{th} for 4 bit quantized MS with conditional correction at SNR=2.0dB for code I. We can see that the optimal performance parameters are: $c_{th}=3.0$, $y_c=1$ and $d_c=1$. At SNR=2.0dB, the simulation results for 5 bits (Fig. A61—Fig. A64) and 6 bits (Fig. A67—Fig. A69) also show that the

optimal c_{th} is also 3.0. However, the other optimal parameter values for the 5 bit quantized version are $y_c=1$ and $d_c=5$, and for 6 bits, there are more choices for y_c and d_c .

At SNR=1.5dB, the simulation results show that all the optimal parameters are almost unchanged for the same number of quantization bits. The simulation results for 4 bits, 5 bits and 6 bits at SNR=1.5 and 2.0dB are given in Fig. A45—Fig. A67 of the Appendix. It can be seen from Fig. A16—Fig. A23 that the optimal $c_{th}=3.0$ is also the optimal c_{th} for clipped MS algorithm (without quantization) with conditional correction at SNR=1.5 and 2.0dB.

Fig. 33 is our simulation results for code I, applying fixed $c_{th}=3.0$ for different quantized MS algorithms with conditional correction.

Fig. 33. BER for code I, performance of uniformly quantized MS with conditional correction for fixed y_c , d_c and c_{th}

From Fig. 33 we can see that quantized MS with conditional correction performs better than MS over a wide range of SNR values. For 6 bits, the performance is very close to the performance of clipped MS with conditional correction reported in Fig. 19, and also that of BP. These results are different from those of [32], [33]. As [32], [33] use sub-optimal values of y_c and d_c , there is an obvious gap between BP and 5, 6 bit quantized MS algorithms at SNR=1.0-2.5dB in [32], [33]. Again, the results with the error floor are missing in [32], [33] at SNR=2.75dB.

We have observed that clipping creates an early error floor at higher SNR. As we have discussed before, using variable clipping thresholds can solve this problem. Similar results are given in the next section in unconditional correction version of quantized MS algorithms. More simulation results on code II and code III are given in Fig. A73—Fig. A77 of the Appendix.

4.3.2. 2 Performance for quantized MS with unconditional correction

Simulations for quantized MS with unconditional correction are done on the three codes. All the simulations for finding the optimal parameters are given in Fig. A79—Fig. A87 of the Appendix. The results show that the optimal parameters c_m are the same as those in conditional correction versions except for 4 quantization bits.

Simulation results are given in Fig. 34, where we can see that unconditional correction improves MS performance. If we compare the performance for 5 and 6 bits with the corresponding performance for conditional correction, we can see that conditional correction performs slightly better than unconditional correction at low SNR region. This is more

obvious for 4 quantization bits, where the performance of unconditional correction is the best when $y_c=0$, i.e., quantized MS without correction. For 4 quantization bits, there are only 15 intervals. Thus the correction value is too large even when we use the smallest value “ $y_c=1$ ”. At larger SNR, we observe that for 5 and 6 bits, unconditional correction with optimized parameters performs better than optimized conditional correction. This would not have happened if we had used conditional correction B for our simulations.

Fig. 34. BER (—)/WER (---) for code I, performance of BP, MS, uniformly quantized MS with unconditional correction, where $C_{th}=2.0$ for 4 bit quantized MS, $C_{th}=3.0$ for 5, 6 bit quantized MS

As we observed before, using a fixed value of c_{th} which is optimal at low SNR, would create an error floor. The error floor however can be improved by using (variable) optimal c_{th} at each given SNR. Fig. 35 shows the result for variable clipping values. And we can see at SNR=2.75dB, the error floor is improved and the performance of 5 and 6 bit quantized MS

with unconditional correction is improved compared with Fig. 34 which uses fixed c_{th} in the algorithms.

Fig. 35. BER (—)/WER (---) for code I, performance of BP, MS, uniformly quantized MS with unconditional correction for variable C_{th} , where $C_{th}=2.0$ for 4 bit quantized MS at SNR=1.0-2.5, $C_{th}=3.0$ for 5, 6 bit quantized MS at SNR=1.0-2.5, $C_{th}=2.5, 4.0, 4.0$ for 4, 5, 6 bit quantized MS at SNR=2.75dB respectively

4. 4 Complexity analysis results

In this section, for each quantized MS algorithm and its modified versions, the average number of iterations for converged cases are investigated as a measure of the complexity. The average number of iterations of uniformly quantized MS algorithms for code I is given in Table 4-1. The average number of iterations of uniformly quantized MS algorithms with conditional correction and unconditional correction for code I is given in Table 4-2 and Table 4-3. The results for code II and code III are given in Table A-1—Table A-5 of the Appendix.

Results show that in general the complexity decreases as SNR increases and that the larger the number of bits, the fewer the average number of iterations. The tables also show that for 5 and 6 bits, conventional MS is more complex than MS with unconditional correction which in turn is slightly more complex than MS with conditional correction. For 4 bits, as $y_c=0$ is a non-correction case of MS with unconditional correction, the complexity is same as conventional MS and more complex than conditional correction.

Eb/No(dB)	3 bit quantization	4 bit quantization	5 bit quantization	6 bit quantization
1.0	61.500	44.602	44.545	44.388
1.5	31.489	26.489	25.668	24.339
2.0	17.164	14.076	13.694	13.597
2.5	10.268	9.206	9.050	9.020
2.75	8.619	7.892	7.789	7.773
2.75dB		7.817 ($C_{th}=2.5$)	7.650 ($C_{th}=3.0$)	7.567 ($C_{th}=3.0$)

Table 4-1. The average iterations for quantization algorithm for (1268, 456) code with $C_{th}=2.0$

Eb/No (dB)	CC with 4 bit quantization $y_c=1, d_c=1$	CC with 5 bit quantization $y_c=1, d_c=5$	CC with 6 bit quantization $y_c=1, d_c=20$
1.0	31.725	27.267	26.630
1.5	19.545	16.792	16.763
2.0	11.948	10.797	10.644
2.5	8.800	8.152	7.973
2.75	7.813	7.280	7.098

Table 4-2. The average number of iterations for uniformly quantized MS algorithm with conditional correction for (1268, 456) code with $C_{th}=3.0$

Eb/No (dB)	UC with 4 bits quantization $y_c=0$	UC with 5 bits quantization $y_c=1$	UC with 6 bits quantization $y_c=2$
1.0	44.602	29.164	28.660
1.5	26.489	18.221	17.573
2.0	14.076	11.324	11.195
2.5	9.206	8.346	8.296
2.75	7.892	7.392	7.361

Table 4-3. The average number of iterations for uniformly quantized MS algorithm with unconditional correction for (1268, 456) code with $C_{th} \approx 2.0$ for 4 bits and $C_{th} \approx 3.0$ for 5, 6 bits

Note CC: conditional correction UC: unconditional correction

4. 5 Another modification of MS algorithm

In the corrections introduced before, the output of a check node becomes zero if y_c is larger than or equal to the current values of $|m_{e,out}|$. If this happens frequently, i.e., if many zero messages are generated at the output of check nodes, the performance of iterative decoding can degrade. To prevent this, we change $|m_{e,out}| = \max(|m_{e,out}| - y_c, 0)$ into:

$$|m_{e,out}| = \begin{cases} |m_{e,out}| - y_c, & \text{if } |m_{e,out}| > y_c \\ |m_{e,out}| & \text{otherwise} \end{cases}$$

Fig. 36 shows BER results for 5 bit quantized MS with unconditional correction for the two methods. The WER performance is given in Fig. A94 of the Appendix. We can see that the new method can slightly improve the performance at low SNR region, but at high SNR, the trend changes. The study also shows that the new method has the same optimal parameters as the old one. This is shown in Fig. A90—Fig. A93 of the Appendix.

Fig. 36. BER for code I, performance of the two modifications for 5 bit uniformly quantized MS with unconditional correction for same parameters of y_c and c_{th}

Chapter 5

Nonuniformly quantized Min-Sum (MS) and its Modifications

5.1 Overview

In this chapter, the performance of nonuniformly quantized MS and its modified algorithms are investigated. In general, nonuniform quantization takes into account the input probability density and quantizes the input with nonuniform spacing of decision levels. In this way, the nonuniform quantization can yield lower quantizing noise and less sensitivity to variations in input signal statistics [20]. Steinhaus [22] and Bennet [21] first developed an effective technique for nonuniform quantization. Unaware of their work, Lloyd [23] rediscovered nonuniform quantization effects, and his contributions were essential for later extensions to vector quantizers and to the development of many quantizer optimization procedures. Later

Max [24] published a variational proof of the Lloyd's optimality properties and rediscovered Lloyd's Method II. In this section, we will investigate the algorithms' performance with nonuniform quantization by using Max's method [24].

5. 2 Nonuniform quantization scheme

For errorless transmission, the difference between the input and quantized output is the quantization error. For N levels of quantization, if we define the distortion, D , as the expected value of $f(\varepsilon)$, where $f(\varepsilon)$ is some differentiable function, and ε is the quantization error, then we have [24]:

$$D = E[f(\varepsilon)] = \sum_{i=1}^N \int_{x_i}^{x_{i+1}} f(x - y_i) p(x) dx$$

where $x_{N+1} = \infty, x_1 = -\infty$, $p(x)$ is the input probability density function and $y_i = Q(x)$, is the output of the quantizer when $x \in [x_i, x_{i+1}]$.

To obtain the minimum value of D , we apply $\frac{\partial D}{\partial x_i} = 0$ and $\frac{\partial D}{\partial y_i} = 0$. Particularly, for minimizing the mean squared error (MSE), we choose $f(x) = x^2$, and these equations result in:

$$\begin{aligned} 1) \quad & x_i = (y_i + y_{i-1})/2 \text{ or } y_i = 2x_i + y_{i-1}, \quad i=2, \dots, N, \\ 2) \quad & \int_{x_i}^{x_{i+1}} (x_i - y_i) p(x) dx = 0, \quad i=1, \dots, N, \end{aligned}$$

which means x_i is the arithmetic mean of y_i and y_{i-1} , while y_i is the centroid of $p(x)$ between x_i and x_{i+1} (conditional expectation of x in the interval $[x_i, x_{i+1}]$).

Because of the complicated functional relationships which are likely to be introduced by $p(x)$, this is not a set of simultaneous equations we can hope to solve analytically [24].

However, by starting from an initial guess for y_1 , one can use steps (1) and (2) iteratively to obtain quantization thresholds x_i 's and quantization values y_i 's.

With q quantization bits, the nonuniform quantization scheme has $2^q - 1$ intervals similar to uniform quantization case, the central quantization interval is chosen symmetric with respect to the origin and is represented by 0. The rest of the intervals are pair-wise symmetric with respect to the origin, and are represented by integer numbers with opposite signs. They are obtained numerically by iterating steps (1) and (2).

Obviously, while using nonuniform quantization, the lengths of different quantization intervals are different. The addition at bit nodes and the subtraction for modified algorithms are also more complex than just integer addition and subtraction, as was the case for uniform quantization. We use look-up tables for such operations. As we will explain later, in some cases, look-up tables can be replaced by simple integer operations.

At a bit node, for adding two integers i and j representing the quantized values y_i and y_j respectively, we obtain $y_k = Q(y_i + y_j)$. The result is then integer k corresponding to quantization value y_k . For subtracting a real number r from the output j of a check node, we first obtain $y_k = Q(y_j - r)$, where y_j is the quantization value corresponding to j . The output will then be k , the integer representing y_k . In general, the operation of nonuniform quantization and the look-up tables depend on SNR. Table 5-1 and Table 5-2 show the quantization thresholds (x_i 's) and values (y_i 's) for 3 and 4-bit at SNR=2.0dB, respectively.

SNR= 2.0dB	$x_i's$	$-\infty$	-3.5709	-1.5650	-0.5060	0.5060	1.5650	3.5709	∞
	$y_i's$	-5.0239	-2.1180	-1.0120	0.0000	1.0120	2.1180	5.0239	

Table 5-1. The quantization thresholds ($x_i's$) and values ($y_i's$) for 3-bit quantization at SNR=2.0dB for code I

SNR= 2.0dB	$x_i's$	$-\infty$	-3.994	-2.620	-1.996	-1.502	-1.059	-0.634	-0.212	0.212	0.634	1.059	1.502	1.996	2.620	3.994	∞
	$y_i's$	-5.010	-2.979	-2.260	-1.731	-1.273	-0.845	-0.423	0.000	0.423	0.845	1.273	1.731	2.260	2.979	5.010	

Table 5-2. The quantization thresholds ($x_i's$) and values ($y_i's$) for 4-bit quantization at SNR=2.0dB for code I

interval	-3	-2	-1	0	1	2	3
-3	-3	-3	-3	-3	-2	-1	0
-2	-3	-3	-3	-2	-1	0	1
-1	-3	-3	-2	-1	0	1	2
0	-3	-2	-1	0	1	2	3
1	-2	-1	0	1	2	3	3
2	-1	0	1	2	3	3	3
3	0	1	2	3	3	3	3

Table 5-3. The addition look-up table for 3-bit quantization at SNR=2.0dB for code I

Intervals	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-6	-5	-5	-4	-2	0
-6	-7	-7	-7	-7	-7	-7	-6	-6	-5	-5	-4	-3	-1	0	2
-5	-7	-7	-7	-7	-7	-6	-6	-5	-4	-3	-2	-1	0	1	4
-4	-7	-7	-7	-7	-6	-6	-5	-4	-3	-2	-1	0	1	3	5
-3	-7	-7	-7	-6	-6	-5	-4	-3	-2	-1	0	1	2	4	5
-2	-7	-7	-6	-6	-5	-4	-3	-2	-1	0	1	2	3	5	6
-1	-7	-6	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	7
0	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
1	-7	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	6	7
2	-6	-5	-3	-2	-1	0	1	2	3	4	5	6	6	7	7
3	-5	-4	-2	-1	0	1	2	3	4	5	6	6	7	7	7
4	-5	-3	-1	0	1	2	3	4	5	6	6	7	7	7	7
5	-4	-1	0	1	2	3	4	5	6	6	7	7	7	7	7
6	-2	0	1	3	4	5	5	6	6	7	7	7	7	7	7
7	0	2	4	5	5	6	7	7	7	7	7	7	7	7	7

Table 5- 4. The addition look-up table for 4-bit quantization at SNR=2.0dB for code I

Intervals r	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
0.2	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7

Table 5- 5. The subtraction look-up table for 4-bit quantization at SNR=2.0dB for code I ($r=0.2$)

SNR= 2.0dB	X_i', S		0.4894	0.6848	0.8809	1.0788	1.2804	1.4883	1.7058	1.9380	2.1921	2.4807	2.8282	3.2971	4.2895	∞
	Y_i', S		0.3917	0.5871	0.7826	0.9792	1.1784	1.3824	1.5941	1.8175	2.0584	2.3258	2.6356	3.0207	3.5736	5.0054

Table 5- 6. The quantization thresholds (X_i', S) and values (Y_i', S) for 5-bit quantization at SNR=2.0dB for code I (positive value of X_i', S are given)

intervals	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
$r=0.2$	-15	-14	-13	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Table 5- 7. The subtraction look-up table for 5-bit quantization at SNR=2.0dB for code I ($r=0.2$)

Table 5-3 and Table 5- 4 show the addition look-up tables for two cases. Table 5- 5 shows the subtraction look-up table for 4-bit case, assuming $r=0.2$ ($r=0.2$ is the optimal correction value for MS with conditional correction). Table 5- 5 also shows that for 4-bit case, the results after subtraction are always the same as before subtraction. Same thing happens for 4-bit case at other SNR values. So applying conditional correction for 4 bits at this SNR region is equivalent to no correction. But for 5 and 6-bit nonuniform quantization, the results after subtraction are not the same as before subtraction. Table 5- 6 and Table 5- 7 show, for 5-bit case of code I at SNR=2.0dB, the quantization thresholds (x_i 's) and values (y_i 's) and the subtraction look-up table for given $r=0.2$, respectively.

5.3 Performance of nonuniformly quantized MS

Fig. 37 is our simulation results for BER of nonuniform quantization of code I, WER curves are given in Fig. A95 of the Appendix. For nonuniform quantization, at bit nodes, extrinsic inputs are processed at random and two at a time. The BER curves of uniform quantization with optimal (variable) c_{th} are also given in comparison. It can be seen that the BER performance of 4, 5, 6 bit nonuniformly quantized MS is the same as (or even slightly better at high SNR than) MS and does not demonstrate an early error floor. The WER performance of nonuniform quantization with 4, 5, 6 bits is also the same as (or slightly worse at SNR=2.75 than) MS. Comparing BER and WER curves of uniform with nonuniform quantization, we can see that at high SNR, nonuniformly quantized MS has a better performance compared to uniformly quantized MS with optimal c_{th} .

Fig. 37. BER for code I, the performance of MS, uniform quantization with variable C_{th} (all optimal C_{th} values are given in Fig. 28 and Fig. A40) and nonuniform quantization for different quantization bits

5. 4 Performance of nonuniformly quantized MS

with modifications

The operation for modified nonuniformly quantized MS algorithms (with conditional and unconditional correction), are more or less the same as those for the uniformly quantized MS algorithms. For the conditional correction of section 3.1, we update $j = |m_{e,out}|$ with the integer corresponding to $\max(Q(y_j - y_c), 0)$ if $|m| \in [|m_{e,out}|, |m_{e,out}| + d_c]$, where y_j is the quantization value corresponding to j , y_c and d_c are optimized correction parameters for the MS algorithm (no clipping or quantization). For the unconditional correction, we choose

$|m_{e,out}|$ to be the integer corresponding to $\max(Q(y_j - y_c), 0)$, where y_j is the quantization value for $|m_{e,min}|$ and y_c is the optimized correction parameter for the MS algorithm.

Fig 38. BER for code I, performance of BP, MS, uniformly and nonuniformly quantized MS with conditional correction for 4 bit case, unconditional correction for 5 and 6 bit cases, where for 4 bit uniformly quantized MS with conditional correction $\{y_c=1, d_c=1, c_{th}=3.0\}$ at SNR=1.0-2.0dB, $\{y_c=1, d_c=0, c_{th}=3.5\}$ at SNR=2.5-2.75dB, for 5, 6 bit uniformly quantized MS with unconditional correction, $c_{th}=3.0$ for all quantization bits at SNR=1.0—2.5dB, at SNR=2.75dB, $c_{th}=4.0$ for 5 and 6 quantization bits

Fig 38 shows our simulation results for nonuniformly quantized MS algorithm with conditional correction and unconditional correction. Corresponding curves for uniformly quantized MS algorithm are also given for comparison. In simulations, optimal y_c parameter is chosen. We can see that nonuniformly quantized MS with unconditional correction (5 and 6 bits) results in a considerable improvement over MS. At low SNR, they perform very close to BP and at higher SNR they even outperform BP. At SNR=2.5 and 2.75dB, our simulations show that nonuniformly quantized MS with unconditional correction performs better than the uniform versions, even when the latter uses the variable optimal clipping threshold at each

given SNR. From 4-bit quantized MS with conditional correction case we can see that nonuniformly quantized version has a better performance than uniformly quantized version at SNR=2.75dB. Overall, the results show that nonuniformly quantized MS algorithm with modifications provide slight improvement over uniformly quantized versions at higher SNR values.

Using look-up tables in nonuniformly quantized MS will result in more memory in hardware implementation. When SNR is given, look-up tables will be fixed for a given number of quantization bits. In some cases, such as 3 bit quantization shown in Table 5-3, the look-up table has a regular structure that can be employed to eliminate the need for the table. In the case of Table 5-3, the look-up table can be replaced by integer addition clipped symmetrically at ± 3 .

Table 5- 4 shows an example of a look-up table that doesn't seem to have any structure and thus cannot be simplified. In such cases, nonuniform quantization has the disadvantage of higher complexity compared to uniform quantization.

5. 5 Complexity analysis results

In this section, for nonuniformly quantized MS algorithm and its modified version (unconditional correction), the average number of iterations for converged cases is given as a measure of complexity.

Table 5-8 and Table 5-9 are results for nonuniformly quantized MS and its unconditional correction version respectively. The tables show that, for quantized MS and its unconditional correction versions, using more bits can decrease the average number of iterations at a given

SNR. Also corrected versions have a lower complexity compared to standard versions, particularly at lower SNR values. Comparing these results with uniform cases, it can be seen that the average number of iterations are more or less the same for uniformly and nonuniformly quantized MS and their unconditional correction versions.

Eb/No(dB)	3 bit quantization	4 bit quantization	5 bit quantization	6 bit quantization
1.0	50.531	47.250	46.978	39.531
1.5	49.275	28.024	27.525	26.424
2.0	25.545	14.530	13.709	13.868
2.5	13.569	9.174	8.817	8.798
2.75	10.560	7.841	7.584	7.555

Table 5-8. The average number of iterations of nonuniformly quantized MS for code I

Eb/No(dB)	UC 4 bit quantization	UC 5 bit quantization	UC 6 bit quantization
1.0	47.250	29.497	27.871
1.5	28.024	18.092	17.883
2.0	14.530	11.439	11.180
2.5	9.174	8.356	8.249
2.75	7.841	7.376	7.304

Table 5-9. The average number of iterations for nonuniformly quantized MS with unconditional correction for code I

Note UC: unconditional correction

Chapter 6

Conclusion and future work

6. 1 Conclusion

In this thesis we have studied MS algorithm and its modifications. The effects of clipping and quantization (uniform and nonuniform) on these algorithms are also investigated. In particular, we have focused on the performance of algorithms at high SNR region where clipping and quantization can result in an early error floor. The error floor effect however can be improved by applying optimal clipping thresholds at a given high SNR. Moreover, we show that conditional correction and unconditional correction can improve the performance of MS algorithm considerably, in some case, to the extent that they even outperform BP algorithm. This is while they have a much lower complexity compared to BP. Our results also show that nonuniform quantization of MS algorithm and its modifications provide some

improvement over uniform quantization, particularly at high SNR values. But, in many cases, this comes at a cost in complexity.

Unconditional correction can be considered as a special case of conditional correction and is simpler to implement. For small number of quantization bits (≤ 4), the latter clearly outperform the former, while for larger number of bits (> 4), they both perform more or less the same and thus the unconditional correction is the favorable choice. Our results also indicate that BP does not always lead to the best performance if the Tanner graph contains many short cycles.

6. 2 Future work

Our results indicate that modified MS algorithms can achieve performance very close to or even better than BP. Efficient hardware implementation of the algorithms is an interesting area of research. Other topics for future research could be: 1) Other simple algorithms that perform well on graphs with cycles. 2) Performance of MS algorithm and its modifications on other channels.

Hybrid-ARQ scheme for channel coding is a hot topic in today's wireless communications. It has been shown that using rate-compatible (RC) LDPC codes can achieve near Shannon limit performance [31] by using a BP decoding algorithm. It would be interesting for future research to investigate RC-LDPC performance under these modified quantized MS algorithms. The main advantage would be the greatly reduced complexity.

APPENDIX

More simulation results

Notes:

code I represents the irregular LDPC block code (1268, 456).

code II represents the regular LDPC block code (273, 191).

code III represents the regular LDPC block code (8000, 4000).

Part 1. MS with modifications

Fig. A1. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of MS with conditional correction at SNR=2.0dB

Fig. A2. WER for code I, the effect of γ_c on performance of MS with unconditional correction at SNR=1.5dB, 2.0dB

Fig. A3. WER for code I, performance of MS with conditional and unconditional correction for fixed γ_c and d_c

Fig. A4. BER (—)/WER (---) for code II, the effect of γ_c and d_c on performance of MS with conditional correction SNR=4.0dB

Fig. A5. BER (—)/WER (---) for code II, the effect of γ_c on performance of MS with unconditional correction at SNR=4.0dB

Part 2. Clipped MS

Fig. A6. WER curves for code I (—), II (--) and III (-.-) decoded by MS (○), clipped MS (no character) and BP (●) with fixed C_{th} = 2.0, 1.5 and 1.25 for code I, code II and code III respectively

Fig. A7. WER curves for code I (—), II (--) and III (-.-) decoded by MS (○), clipped MS (no character) and BP (●) with optimal C_{th} = 3.5 at SNR=2.5, C_{th} = 4.0 at SNR=2.75dB for code I and C_{th} = 2.5, 2.5, 3.0 at SNR=4.5, 5.0, 5.5dB for code II

Fig. A8. The optimal clipping threshold of code I on WER performance of MS at different SNR

Fig. A 9. BER (—)/WER (---) for code II, the effect of clipping threshold C_{th} on performance of MS at SNR=3.5dB

Fig. A10. BER (—)/WER (---) for code II, the effect of clipping threshold C_{th} on performance of MS at SNR=4.5dB

Fig. A 11. BER (—)/WER (---) for code II, the effect of clipping threshold C_{th} on performance of MS at SNR=5.0dB

Fig. A12. BER (—)/WER (---) for code II, the effect of clipping threshold C_{th} on performance of MS at SNR=5.5dB

Fig. A13. The optimal clipping threshold of code II on BER performance of MS at different SNR

Fig. A14. The optimal clipping threshold of code II on WER performance of MS at different SNR

Fig. A15. BER (—)/WER (---) for code III, the effect of clipping threshold C_{th} on performance of MS at SNR=1.9dB

Part 3. Clipped MS with modifications

Fig. A16. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at $\text{SNR}=1.5\text{dB}$ ($C_{th}=2.0$)

Fig. A17. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at $\text{SNR}=1.5\text{dB}$ ($C_{th}=2.5$)

Fig. A18. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=1.5dB ($C_{th}=3.0$)

Fig. A19. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=1.5dB ($C_{th}=3.5$)

Fig. A20. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at
SNR=1.5dB ($C_m=4.0$)

Fig. A21. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at
SNR=2.0dB ($C_m=2.0$)

Fig. A22. BER (—)/WER (---) for code 1, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=2.0dB ($C_{th}=2.5$)

Fig. A23. BER (—)/WER (---) for code 1, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=2.0dB ($C_{th}=3.0$)

Fig. A24. BER (—)/WER (---) for code I, performance of BP, MS, clipped MS with $C_{th}=2.0$, clipped MS with conditional correction ($C_{th}=3.0$) for fixed y_c , d_c values

Fig. A25. BER (—)/WER (---) for code I, the effect of y_c and C_{th} on performance of clipped MS with unconditional correction at SNR=1.5dB

Fig. A26. BER (—)/WER (---) for code I, the effect of y_c and C_{th} on performance of clipped MS with unconditional correction at SNR=2.0dB

Fig. A27. WER for code I, performance of BP, MS, clipped MS with conditional and unconditional correction for fixed y_c, d_c and C_{th}

Fig. A28. WER for code I, decoded by clipped MS with unconditional correction, $C_{th}=3.0$ at SNR=1.0-2.5dB, $C_{th}=4.0$ at SNR=2.75dB

Fig. A29. BER (—)/WER (---) for code II, the effect of y_c and d_c on performance of clipped MS with conditional correction at SNR=4.0dB ($C_{th}=1.5$)

Fig. A30. BER (—)/WER (---) for code II, the effect of y_c and d_c on performance of clipped MS with conditional correction at $\text{SNR}=4.0\text{dB}$ ($C_{th}=2.0$)

Fig. A31. BER (—)/WER (---) for code II, the effect of y_c and d_c on performance of clipped MS with conditional correction at $\text{SNR}=4.0\text{dB}$ ($C_{th}=2.5$)

Fig. A32. BER (—)/WER (---) for code II, the effect of y_c on performance of clipped MS with unconditional correction at SNR=4.0dB

Fig. A33. BER (—)/WER (---) for code II, performance of BP, MS, clipped MS with conditional correction with $y_c=0.1$, $d_c=0.3$, $c_{th}=2.0$

Fig. A34. BER (—)/WER (---) for code II, performance of BP, MS, clipped MS with unconditional correction ($\gamma_c=0.1$, $C_{th}=2.0$)

Fig. A35. WER for code III, the effect of γ_c , d_c and C_{th} on performance of clipped MS with conditional correction at SNR=1.4dB

Part 4. Uniformly quantized MS

Fig. A36. WER for code I, the effect of C_{th} on performance of quantized MS for different quantization bits at SNR=1.5dB

Fig. A37. WER for code I, the effect of C_{th} on performance of quantized MS for different quantization bits at SNR=2.0dB

Fig. A38. BER (—)/WER (---) for code 1, the effect of C_{th} on performance of quantized MS for different quantization bits at SNR=2.5dB

Fig. A39. WER for code 1, performance of quantized MS for different quantization bits ($C_{th}=2.0$)

Fig. A40. WER for code I, performance of quantized MS for different quantization bits, where for 4,5,6 quantization bits, at SNR=1.0-2.0dB, $C_{th}=2.0$; for 4 quantization bits at SNR=2.5,2.75dB, $C_{th}=2.5,3.5$ and for 5,6 quantization bits at SNR=2.5,2.75dB, $C_{th}=3.0,4.5$

Fig. A41. BER (—)/WER (---) for, performance of quantized MS for different quantization bits at SNR=3.5dB

Fig. A42. BER for code III, the effect of C_{th} on performance of quantized MS for different quantization bits at SNR=1.9dB

Fig. A43. WER for code III, the effect of C_{th} on performance of quantized MS for different quantization bits at SNR=1.9dB

Fig. A44. WER performance of quantized MS for different quantization bits, where for code II (---) $C_{th}=1.5$, for code III (—)

$$C_{th}=1.25$$

Part 5. Uniformly quantized MS with conditional correction

Fig. A45. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at

$$SNR=1.5dB \ (C_{th}=2.0)$$

Fig. A46. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=1.5dB ($C_{th}=2.5$)

Fig. A47. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=1.5dB ($C_{th}=3.0$)

Fig. A48. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction
at SNR=1.5dB ($C_{th}=3.5$)

Fig. A49. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction
at SNR=1.5dB ($C_{th}=2.0$)

Fig. A50. BER (—)/WER (---) for code 1, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=1.5dB ($C_{th}=2.5$)

Fig. A51. BER (—)/WER (---) for code 1, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=1.5dB ($C_{th}=3.0$)

Fig. A52. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction
at SNR=1.5dB ($C_{th}=3.5$)

Fig. A53. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction
at SNR=1.5dB ($C_{th}=2.0$)

Fig. A54. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction
at SNR=1.5dB ($C_{th}=2.5$)

Fig. A55. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction
at SNR=1.5dB ($C_{th}=3.0$)

Fig. A56. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction
at SNR=1.5dB ($C_{th}=3.5$)

Fig. A57. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction
at SNR=2.0dB ($C_{th}=2.0$)

Fig. A58. BER (—)/WER (---) for code 1, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction
at SNR=2.0dB ($C_{th}=2.5$)

Fig. A59. BER (—)/WER (---) for code 1, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction
at SNR=2.0dB ($C_{th}=3.0$)

Fig. A60. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction
at SNR=2.0dB ($C_{th}=3.5$)

Fig. A61. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction
at SNR=2.0dB ($C_{th}=2.0$)

Fig. A62. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction
at SNR=2.0dB ($C_{th}=2.5$)

Fig. A63. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction
at SNR=2.0dB ($C_{th}=3.0$)

Fig. A64. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=2.0dB ($C_{th}=3.5$)

Fig. A65. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=2.0dB ($C_{th}=2.0$)

Fig. A66. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=2.0dB ($C_{th}=2.5$)

Fig. A67. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=2.0dB ($C_{th}=3.0$)

Fig. A68. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.5dB ($C_{th}=2.5$)

Fig. A69. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.5dB ($C_{th}=3.0$)

Fig. A70. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.5dB ($C_{th}=3.5$)

Fig. A71. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=2.5dB ($C_{th}=4.0$)

Fig. A72. BER (—)/WER (---) for code I, performance of BP, MS, quantized MS with conditional correction for different quantization bits with fixed y_c , d_c and c_{th}

Fig. A73. BER (—)/WER (---) for code II, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional correction at SNR=4.0dB ($c_{th}=2.0$)

Fig. A74. BER (—)/WER (---) for code II, the effect of γ_c and d_c on performance of 5 bit quantized MS with conditional correction at SNR=4.0dB ($C_{th}=2.0$)

Fig. A75. BER (—)/WER (---) for code II, the effect of γ_c and d_c on performance of 6 bit quantized MS with conditional correction at SNR=4.0dB ($C_{th}=2.0$)

Fig. A76. BER (—)/WER (---) for code II, performance of BP, MS and quantized MS with conditional correction for different quantization bits with fixed y_c , d_c and c_{th}

Fig. A77. BER (—)/WER (---) for code III, performance of BP, MS and 4 bit quantized MS with conditional correction for fixed y_c , d_c and c_{th}

Part 6. More results for conditional correction B

Fig. A78. BER (—)/WER (---) for code I, performance of 5 bit quantized MS with conditional correction (B) for fixed y_c and d_c at SNR=2.0dB ($C_{th}=3.0$)

Part 7. Uniform quantization with unconditional correction

Fig. A79. BER (—)/WER (---) for code I, the effect of y_c and C_{th} on performance of 4 bit quantized MS with unconditional correction at SNR=1.5dB

Fig. A80. BER (—)/WER (---) for code I, the effect of y_c and C_{th} on performance of 5 bit quantized MS with unconditional correction at SNR=1.5dB

Fig. A81. BER (—)/WER (---) for code I, the effect of y_c and C_{th} on performance of 6 bit quantized MS with unconditional correction at SNR=1.5dB

Fig. A82. BER (—)/WER (---) for code I, the effect of γ_c and C_{th} on performance of 4 bit quantized MS with unconditional correction at SNR=2.0dB

Fig. A83. BER (—)/WER (---) for code I, the effect of γ_c and C_{th} on performance of 5 bit quantized MS with unconditional correction at SNR=2.0dB

Fig. A84. BER (—)/WER (---) for code I, the effect of γ_c and C_{th} on performance of 6 bit quantized MS with unconditional correction at SNR=2.0dB

Fig. A85. BER (—)/WER (---) for code II, the effect of γ_c and C_{th} on performance of 4 bit quantized MS with unconditional correction at SNR=4.0dB

Fig. A86. BER (—)/WER (---) for code II, the effect of y_c and C_{th} on performance of 5 bit quantized MS with unconditional correction at SNR=4.0dB

Fig. A87. BER (—)/WER (---) for code II, the effect of y_c and C_{th} on performance of 6 bit quantized MS with unconditional correction at SNR=4.0dB

Fig. A88. BER (—)/WER (---) for code II, performance of BP, MS and quantized MS with unconditional correction for different quantization bits with fixed y_c and C_{th}

Fig. A89. BER (—)/WER (---) for code III, performance of 4 bit quantized MS with conditional correction and 5,6 bit quantized MS with unconditional correction for fixed y_c and d_c

Part 8. More results on Modified MS algorithm of section of 4.5

Fig. A90. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of clipped MS with conditional correction at

SNR=2.0dB ($C_{th}=3.0$)

Fig. A91. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 4 bit quantized MS with conditional at

SNR=2.0dB ($C_{th}=3.0$)

Fig. A92. BER (—)/WER (---) for code I, the effect of y_c and d_c on performance of 5 bit quantized MS with conditional at SNR=2.0dB ($C_{th}=3.0$)

Fig. A93. BER (—)/WER (---) for code I, the effect of C_{th} and y_c on performance of 5 bit quantized MS with unconditional correction at SNR=2.0dB

Fig. A94. WER for code I, performance of BP, MS and 5 bit quantized MS with unconditional correction for method of section 4.5
with fixed y_c and C_{th}

Part 9. Nonuniformly quantized MS and its modifications

Fig. A95. WER for code I, performance of MS and nonuniformly quantized MS for different quantization bits

Fig. A 96. WER for code I, performance of BP, MS, uniformly and nonuniformly quantized MS with conditional correction for 4 bits cases, unconditional correction for 5 and 6 bits cases, where for 4 bits uniformly quantized MS with conditional correction $\{ \gamma_c=1, d_c=1, c_{th}=3.0 \}$ at SNR=1.0-2.0dB, $\{ \gamma_c=1, d_c=0, c_{th}=4.0 \}$ at SNR=2.5dB, $\{ \gamma_c=1, d_c=0, c_{th}=4.5 \}$ at SNR=2.75dB, for 5, 6 bits uniformly quantized MS with unconditional correction, $c_{th}=3.0$ for all quantization bits at SNR=1.0—2.5dB, at SNR=2.75dB, $c_{th}=4.0$ for 5 and 6 quantization bits

Part 10. Some tables

Eb/No(dB)	MS Algorithm	Clipped MS with conditional correction $y_c=0.1, d_c=0.3, c_{th}=2.0$	Clipped MS with unconditional correction $y_c=0.1, c_{th}=2.0$
2.0	25.506	14.603	16.638
2.5	15.878	8.464	9.405
3.0	8.581	5.598	5.805
3.5	4.360	3.420	3.511
4.0	2.585	2.395	2.403
4.5	1.824	1.822	1.813
5.0	1.474	1.459	1.450

Table A-1. The average number of iterations for different algorithms for code II

Eb/No(dB)	3-bit, $c_{th}=1.5$	4-bit, $c_{th}=1.5$	5-bit, $c_{th}=1.5$	6-bit, $c_{th}=1.5$
2.0	28.733	25.892	26.056	25.645
2.5	19.137	15.532	13.970	13.876
3.0	12.745	9.147	8.726	8.720
3.5	5.635	4.300	4.105	4.102
4.0	2.911	2.495	2.499	2.533
4.5	1.758	1.707	1.750	1.780
5.0	1.318	1.333	1.370	1.394

Table A-2. The average number of iterations of uniformly quantized MS for code II

Eb/No(dB)	4-bit MS with conditional correction $y_c = 1, d_c = 0, c_{th} = 2.0$	5-bit MS with conditional correction $y_c = 1, d_c = 2, c_{th} = 2.0$	6-bit MS with conditional correction $y_c = 1, d_c = 4, c_{th} = 2.0$	4-bit MS with unconditional correction $y_c = 0, c_{th} = 1.5$	5-bit MS with unconditional correction $y_c = 1, c_{th} = 2.0$	6-bit MS with unconditional correction $y_c = 1, c_{th} = 2.0$
2.0	17.807	16.776	12.415	25.892	17.833	16.653
2.5	11.821	9.899	7.435	15.532	8.246	9.541
3.0	6.453	6.1978	5.762	9.147	5.822	5.973
3.5	3.516	3.469	3.418	4.300	3.449	3.536
4.0	2.298	2.307	2.360	2.495	2.376	2.322
4.5	1.677	1.720	1.776	1.707	1.758	1.723
5.0	1.339	1.376	1.423	1.333	1.404	1.377

Table A-3. The average number of iterations for uniformly quantized MS with modifications for code II

Eb/No(dB)	MS algorithm	Clipped MS, $c_{th} = 1.25$	3-bit $c_{th} = 1.25$	4-bit $c_{th} = 1.25$	5-bit $c_{th} = 1.25$	6-bit $c_{th} = 1.25$
1.6		52.627				
1.7	41.173	36.819	59.081	45.859	40.015	37.107
1.8	28.543	27.613	46.925	32.010	28.756	27.314
1.9	20.846	20.353	38.862	23.158	21.182	20.638
2.0	15.695	16.174	26.487	17.859	16.628	16.331
2.1	12.832	13.782	19.952	14.803	14.072	13.886
2.2			16.009	12.890		
2.3			13.660			

Table A-4. The average number of iterations for uniformly quantized MS for code III

Eb/No(dB)	4-bit MS with conditional correction $\gamma_c = 1$, $d_c = 1$, $c_{th} = 2.5$	5-bit MS with unconditional correction $\gamma_c = 1$, $c_{th} = 2.5$	6-bit MS with unconditional correction $\gamma_c = 2$, $c_{th} = 2.5$
1.2	70.125	52.525	52.448
1.3	59.592	42.084	43.708
1.4	49.198	33.149	35.280
1.5	38.018	24.306	23.253
1.6	29.001	19.356	18.567
1.7	22.637	16.269	15.793
1.8	18.807		
1.9	16.377		

Table A-5. The average number of iterations for uniformly quantized MS with modifications for code III

REFERENCES

- [1] C.E. Shannon, "A Mathematical Theory of Communication," *Bell Syst. Tech.J.*, 27, pp.379-423 (Part I), 623-656(Part II), July 1948.
- [2] R.G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21-28, Jan. 1962.
- [3] D.J.C. Mackay and R.M.Neal, "Near Shannon Limit performance of low density parity check codes," *Electronics Letters*, vol.33, No.6, pp.457-458, March 1997.
- [4] D.J.C. Mackay, "Good Error-Correcting Codes Based on Very Sparse Matrices," *IEEE Trans. Inform. Theory*, vol. 45, No.2, pp.399-431, March 1999.
- [5] T. Richardson and R. Urbanke, "The capacity of Low Density Parity Check Codes Under Message-Passing Decoding," *IEEE Trans. Inform. Theory*, vol. 47, No.2, pp.599-618, Feb.2001
- [6] M. Fossorier, et al. "Reduced Complexity Iterative Decoding of Low-Density Parity Check Codes Based on Belief Propagation," *IEEE Trans. Comm*, vol.47. No.5, pp.673-680, May 1999.
- [7] C. Berroa A. Glavieux and P.Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo Codes (1)," *Proc. ICC'93*. Geneva, Switzerland, 1993, pp.1064-1070.
- [8] G.D. Forney, Jr., "The Viterbi algorithm," *Proc, IEEE*, vol.61, No.3, pp.268-278. March 1973.
- [9] R.M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533-547, Sept. 1981.

- [10] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Dept. of Electrical Engineering, Linköping University, Linköping, Sweden, 1996.
- [11] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Interference," San Mateo, CA: Morgan Kaufmann, 1988.
- [12] J. Chen and M. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Comm. Lett.*, vol. 6, no. 5, pp. 208-210, May 2002.
- [13] J. Chen and M. Fossorier, "Density evolution for BP-based decoding algorithms of LDPC codes and their quantized versions," *Proc. IEEE Globecom*, pp. 1378-1382, Nov. 2002.
- [14] Y. Mao, A. H. Banihashemi, "A heuristic search for good low-density parity-check codes at short block lengths," *Proc. IEEE. ICC*, vol. 1, pp. 41-44, 2001.
- [15] T. Richardson and R. Urbanke, "The Renaissance of Gallager's Low-Density Parity-Check Codes," *IEEE Comm. Magazine*, vol. 41, pp.126-131, Aug. 2003.
- [16] G.D. Forney, Jr., "On iterative decoding and the two-way algorithm," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Brest, France, pp. 12-25, Sept. 1997.
- [17] L. Ping, W. K. Leung, "Decoding low-density parity check codes with finite quantization bits," *IEEE Comm. Lett.*, vol. 4, pp. 62 – 64 , No. 2, Feb. 2000.
- [18] W. Ryan "An Introduction to Low-Density Parity-Check Codes," University of Arizona, April, 2001. Unpublished.
- [19] R.M. Gray and D.L. Neuhoff, "Quantization", *IEEE Trans. Inform. Theory*, vol.44, No.6, October 1998.
- [20] A. Gersho, "Quantization", *IEEE Communications Society Magazine*, September 1977.

- [21] W.R. Bennett, "Spectra of quantized signals," *Bell Syst. Tech.J.*, vol.27, pp.446-472, July 1948.
- [22] H. Steinhaus, "Sur la division des corps materiels en parties," *Bull. Acad. Polon. Sci., CI. III*, vol. IV, 801-804, 1956.
- [23] S.P. Lloyd. "Least squares quantization in PCM," unpublished Bell Laboratories Technical Note. Portions presented at the Institute of Mathematical Statistics Meeting Atlantic City NewJersey September 1957. Published in special issue on quantization, *IEEE Trans. Inform. Theory*, vol.28, pp.129-137, Mar. 1982.
- [24] J. Max, "Quantizing for Minimum Distortion," *IEEE, Trans. Inform. Theory*, pp.7-12, March 1960.
- [25] E. Eleftheriou, T. Mittelholzer and A. Dholakia, "Reduced-complexity decoding algorithm for low-density parity-check codes," *IEEE Elec. Lett.*, vol. 37, no. 2, pp. 102-104, Jan. 2001.
- [26] J. Chen and M.P.C.Fossorier, "Decoding low-density parity-check codes with normalized APP-based algorithm," in Proc. *IEEE Globecom*, pp.1026-1030, Nov, 2001.
- [27] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold and A. Dholakia, "Efficient implementation of the sum-product algorithm for decoding LDPC codes," Proc. *IEEE Globecom*, San Antonio, TX, USA, pp. 1036-1036E, Nov. 2001.
- [28] J. Chen and M.Fossorier, "Near optimal universal belief propagation based decoding of low-density parity-check codes," in *IEEE Trans. Commun.*, vol.50, pp.406-414, Mar, 2002.
- [29] J. Chen and M. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Commun. Lett*, pp. vol.6, no.5, 208-210, May. 2002.

- [30] J. Chen, A. Dholakia, E. Eleftheriou, M. Fossorier, and X.-Y. Hu, "Near optimal reduced-complexity decoding algorithm for LDPC codes," in *Proc. IEEE Intl. Sympo. Info. Theory*, p.455, Jul. 2002.
- [31] M.R. Yazdani, "Construction of Irregular Rate-Compatible Low-Density Parity-Check Codes for Hybrid-ARQ Schemes at Short Block Lengths", Thesis, Carleton University, April, 2003.
- [32] F. Zarkeshvari, "Implementation of Iterative Decoding Algorithms on Digital VLSI Platforms", Thesis, Carleton University, October 2002.
- [33] F. Zarkeshvari and A.H. Banihashemi, "On Implementation of Min-Sum Algorithm for Decoding Low-Density Parity-Check (LDPC) Codes," *IEEE Globecom 2002*, Taipei, Taiwan, November, pp.17-21, 2002.
- [34] T.M. Cover, Joy A. Thomas, "Elements of Information Theory," John Wiley & Sons, Inc. 1991.
- [35] R.H. Morelos-Zaragoza: <http://the-art-of-ecc.com/>
- [36] <http://www.inference.phy.cam.ac.uk/mackay/codes/data.html#s14>