

# Code-Independent Error Floor Estimation Techniques for Flooding and Layered Decoders of LDPC Codes

by

**Ali Farsiabi**

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

**Doctor of Philosophy in Electrical and Computer Engineering**

Ottawa-Carleton Institute for Electrical and Computer Engineering  
Department of Systems and Computer Engineering  
Carleton University  
Ottawa, Ontario  
August, 2020

©Copyright  
Ali Farsiabi, 2020

The undersigned hereby recommends to the  
Faculty of Graduate Studies and Research  
acceptance of the dissertation

**Code-Independent Error Floor Estimation Techniques for  
Flooding and Layered Decoders of LDPC Codes**

submitted by **Ali Farsiabi**

in partial fulfillment of the requirements for the degree of

**Doctor of Philosophy in Electrical and Computer Engineering**

---

Professor Amir H. Banihashemi, Thesis Supervisor

---

Professor Bane Vasić, External Examiner

---

Professor James R. Green, Chair,  
Department of Systems and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering  
Department of Systems and Computer Engineering  
Carleton University  
August, 2020

# Abstract

Finite-length low-density parity-check (LDPC) codes under iterative message passing algorithms suffer from error floor, i.e., as the channel quality improves, at some point, the error rate does not decrease as fast as its initial rate of decrease. The literature on the error floor analysis/estimation of LDPC codes can be, in general, partitioned into two main categories. The first category requires the full knowledge of the parity-check matrix or Tanner graph of the code to estimate the error floor. The second category, on the other hand, is code-independent, in that, rather than the full knowledge of the code's Tanner graph, these techniques only require the multiplicity and topology of harmful substructures of the Tanner graph, referred to as *trapping sets* (TSs), and possibly the degree distributions of the graph, to estimate the error floor. The proposed techniques in this thesis fit within the second category.

The linear state-space model is a well-known code-independent method to estimate the contribution of a trapping set structure to the error floor of low-density parity-check codes. In this thesis we first provide an in-depth analysis of this method by incorporating a more accurate model for the incoming messages to the TS structure that takes into account the randomness and the correlation among such messages. Based on this analysis, we demonstrate that both randomness and correlation result in the over-estimation of the failure probability of the TS. We then propose an alternate code-independent technique for the error floor estimation of iterative LDPC decoders that can accurately estimate the contribution of different TS structures in the error floor. Compared to the linear state-space model, the proposed method is not only more accurate, but also more general, in that, it is applicable to any saturating iterative message-passing decoder, symmetrically quantized or unquantized, over any memoryless binary-input output-symmetric channel.

To the best of our knowledge, all the existing work on the theoretical analysis of error floor is limited to two-phase message passing algorithms, also known as *flooding* or *parallel schedule*. There are however a variety of message passing schedules which

are advantageous to flooding in terms of performance, complexity or convergence speed. An important category of schedules are *layered or serial schedules*. In this thesis, we also analyze the error floor of quasi-cyclic (QC) low-density parity-check codes decoded by the sum-product algorithm (SPA) with row layered message-passing scheduling. For this, we develop a linear state-space model of trapping sets which incorporates the layered nature of scheduling. We demonstrate that the contribution of each TS to the error floor is not only a function of the topology of the TS, but also depends on the row layers in which different check nodes of the TS are located. This information, referred to as TS layer profile (TSLP), plays an important role in the harmfulness of a TS. As a result, the harmfulness of a TS in particular, and the error floor of the code in general, can significantly change by changing the order in which the information of different layers, corresponding to different row blocks of the parity-check matrix, is updated. We also study the problem of finding a layer ordering that minimizes the error floor, and obtain row layered decoders with error floor significantly lower than that of their flooding counterparts. As part of our analysis, we make connections between the parameters of the state-space model for a row layered schedule and those of the flooding schedule.

We extend the linear state-space model of row layered schedule to the column layered decoders, and demonstrate that the model parameters for the latter are derived differently than those of the former. We also optimize the error floor performance of column layered decoders and show that their error floor can be better or worse than that of their row layered counterparts depending on TS structures and their layer profiles for each decoder.

Finally, we propose a semi-linear state-space model of TSs in which, rather than the fixed operating point of zero, used in the original linear state-space model, the operational points are estimated dynamically. Compared to the linear state-space model, the proposed method is not only more accurate, but also has the advantage of error extrapolation, i.e., estimating the error floor at different signal to noise ratios (SNRs) based on the estimated error rate at a specific SNR value.

To My Wife

# Acknowledgments

I would like to express my deep and sincere gratitude to my supervisor, Prof. Amir Banihashemi, for his dedicated support and encouragement throughout my Ph.D. program. I benefited much from his invaluable and thoughtful suggestions, editorial skills and overall insights in this field.

I am grateful to Prof. Bane Vasić, from the Department of Electrical and Computer Engineering, University of Arizona to serve as the external examiner of my Ph.D. defense. I would also like to thank other committee members of my defense: Prof. Yongyi Mao from School of Electrical Engineering and Computer Science University of Ottawa, Prof. Ian Marsland, and Prof. Calvin Plett from Systems and Computer Engineering Department of Carleton University.

Last, but not the least, I would like to thank my parents, whose love and guidance are with me in every step of my life. Most importantly, I wish to express my deepest gratitude to my wonderful wife and best friend Elnaz for her unconditional love and unbelievable support.

# Table of Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>vi</b>
<b>Table of Contents</b>	<b>vii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xi</b>
<b>Nomenclature</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Related Works . . . . .	1
1.2 Summary of Contributions . . . . .	4
1.3 Organization of the Thesis . . . . .	6
<b>2 Preliminaries</b>	<b>7</b>
2.1 Notations . . . . .	7
2.2 LDPC Codes and Decoding Algorithms . . . . .	7
2.3 Quantization . . . . .	13
2.4 Flooding and Layered Schedules . . . . .	14
2.5 Trapping Sets (TSs) . . . . .	17
2.6 Importance Sampling . . . . .	20
2.7 Density Evolution . . . . .	21
<b>3 Error Floor Estimation of LDPC Decoders - A Code Independent Approach to Measuring the Harmfulness of Trapping Sets</b>	<b>22</b>
3.1 State-Space Model of an ETS . . . . .	22

3.1.1	From non-linear to linear state-space model . . . . .	23
3.1.2	Application of Density Evolution in the State-Space Model . . . . .	28
3.1.3	Error Probability of a LETS Structure . . . . .	28
3.2	Estimation Error of the Linear State-Space Model . . . . .	33
3.3	Proposed Method . . . . .	34
3.3.1	Local importance sampling (LIS) . . . . .	35
3.3.2	Correlation model . . . . .	36
3.3.3	Complexity of LIS scheme . . . . .	37
3.4	Simulation Results of the Proposed Technique and Comparisons with the Results of Linear State-Space Model . . . . .	38
3.4.1	Simulation results of the proposed technique for quantized de- coders . . . . .	38
3.4.2	Comparison of the LIS-DE <sub>c</sub> method with linear state-space model . . . . .	46
3.5	Appendix . . . . .	48
<b>4</b>	<b>Error Floor Analysis of LDPC Row Layered Decoders</b>	<b>51</b>
4.1	Linear State-Space Model of LETSs for SPA with Row Layered Schedule . . . . .	51
4.1.1	Relationship between model matrices of flooding and row lay- ered schedules . . . . .	52
4.1.2	Linear State-Space Model of LETSs for Row Layered SPA . . . . .	58
4.1.3	The Application of DE to the Layered Decoder . . . . .	59
4.1.4	Missatisfied CN Gain Model in Layered Decoders . . . . .	62
4.1.5	Spectral Properties of LETS System Matrices in Layered De- coders . . . . .	64
4.1.6	Computing the Failure Probability of LETSs in Layered Decoders . . . . .	74
4.2	The Effect of Row Block Permutations on the Failure Probability of a LETS: Analysis and Optimization . . . . .	76
4.2.1	Effect of Different Row Block Permutations on the Dominant Eigenvalue of $\tilde{\mathbf{A}}_{J \rightarrow 1}$ . . . . .	76
4.2.2	Optimizing the Row Layered Schedule . . . . .	78
4.3	Simulation Results and Discussions . . . . .	79
4.4	Appendix . . . . .	87
<b>5</b>	<b>Error Floor Analysis of LDPC Column Layered Decoders</b>	<b>91</b>

5.1	State Space Model of the Column Layered Decoders in the Error Floor Region . . . . .	91
5.1.1	State-Space Model . . . . .	91
5.1.2	Relationship between model matrices of flooding and column layered schedules . . . . .	93
5.1.3	The Application of DE to the Column Layered Decoder . . . . .	97
5.1.4	Linear Gain Model . . . . .	99
5.1.5	Failure Probability . . . . .	102
5.2	Optimization of the Column Orders . . . . .	106
5.3	Simulation Results . . . . .	109
5.4	Appendix . . . . .	112
<b>6</b>	<b>A Semi Linear State Space Model for Error Floor Estimation of LDPC codes over the AWGN Channel</b>	<b>115</b>
6.1	Proposed Semi-linear Model with Dynamical Operating Point . . . . .	115
6.1.1	Main Ideas and Methodology . . . . .	115
6.1.2	Proposed Model . . . . .	116
6.1.3	Modifications to the Proposed Model . . . . .	126
6.1.4	Relationship between Our Proposed Method and the Semi-Analytical Method of [34] . . . . .	132
6.2	Simulation Results . . . . .	134
6.3	Appendix . . . . .	137
6.3.1	Derivation of the mean and the variance of $\xi_2$ , given in Algorithm 3 . . . . .	137
6.3.2	Calculation of the covariance matrices $\Sigma_{\mathbf{x}^{\ell(1)}}$ and $\Sigma_{\mathbf{x}^{\ell(1)\odot 2}}$ . . . . .	139
6.3.3	Derivation of the mean and the variance of the conditional error indicator function for $\ell \geq 3$ , as given in Algorithm 3 . . . . .	141
<b>7</b>	<b>Conclusion and Future Work</b>	<b>143</b>
7.1	Conclusion . . . . .	143
7.2	Future Work . . . . .	145
	<b>List of References</b>	<b>153</b>

# List of Tables

2.1	Notations and Symbols (Part A) . . . . .	8
2.2	Notations and Symbols (Part B) . . . . .	9
2.3	Notations and Symbols (Part C) . . . . .	10
2.4	Notations and Symbols (Part D) . . . . .	11
3.1	Multiplicities of $(a, b)$ LETSs of $\mathcal{C}_3$ with $a \leq 8$ and $b \leq 3$ . . . . .	40
3.2	Multiplicities of the $(a, b)$ LETSs of $\mathcal{C}_4$ within the range $a \leq 8$ and $b \leq 2$ . . . . .	42
4.1	Multiplicities of $(a, b)$ LETSs of $\mathcal{C}_2$ within the range $a \leq 8$ and $b \leq 2$ . . . . .	84

# List of Figures

2.1	A typical error rate curve of an LDPC code versus SNR that consists of two separate region known as the waterfall and the error floor, respectively. . . . .	15
2.2	Illustration of a MS decoder failure on a (5, 5) ETS belonging to a QC-LDPC (640, 192) code. Each figure represents the total likelihood of VNs at a given iteration (vertical axis) versus the VN indices (horizontal axis), respectively. . . . .	18
2.3	The subgraph of the (5, 5) ETS in the Tanner graph of the (640, 192) LDPC code. The VNs, missatisfied CNs and unsatisfied CNs are represented by black circles, white squares and gray squares, respectively. . . . .	19
3.1	A missatisfied check node connecting two variable nodes of an ETS. By combining the $d_{c_j} - 2$ external messages into $L_{s_j}$ , the equivalent check node has degree 3 regardless of the actual check node degrees of the LDPC code. The message $L_{s_j}$ is assumed to be sent by a virtual variable node $s_j$ . . . . .	23
3.2	A (5, 3) LETS. The VNs, missatisfied CNs and unsatisfied CNs are shown by black circles, white squares and gray squares, respectively. The external connections for one of the missatisfied CNs are also shown. . . . .	24
3.3	Simulation and estimation results for $\mathcal{C}_1$ decoded by quantized MSA over the AWGN channel ( $\Delta = 0.5$ , $I_{max} = 200$ , and decoder input = $2y/\sigma_{ch}^2$ ). . . . .	40
3.4	Simulation and estimation results for $\mathcal{C}_1$ decoded by 4-bit MSA over the BSC ( $I_{max} = 200$ , decoder input values 1 and $-1$ ). . . . .	41
3.5	Simulation and estimation results for $\mathcal{C}_2$ decoded by 5-bit MSA and 5-bit normalized MSA over the AWGN channel ( $C_{th} = 3$ , $I_{max} = 100$ , $\alpha = 0.75$ , decoder input = $y$ ). . . . .	41

3.6	Simulation and estimation results for $\mathcal{C}_3$ decoded by a 6-bit SPA over the AWGN channel ( $\Delta = 0.5$ , $I_{max} = 200$ , decoder input = $2y/\sigma_{ch}^2$ ). . . . .	42
3.7	Simulation and estimation results for $\mathcal{C}_4$ decoded by a 3-bit MSA ( $C_{th} = 1.8$ , $I_{max} = 100$ and decoder input = $y$ ). . . . .	43
3.8	Simulation and estimation results of (2048, 1723) RS-LDPC code ( $\mathcal{C}_5$ ) decoded by a 5-bit normalized MSA ( $\alpha = 0.75$ , $\Delta = 0.5$ , $I_{max} = 50$ and decoder input = $2y/\sigma_{ch}^2$ ). . . . .	44
3.9	Simulation and estimation results for $\mathcal{C}_6$ and $\mathcal{C}_7$ decoded by a 7-bit normalized MSA ( $C_{th} = 6$ , $\alpha = 0.6$ , $I_{max} = 100$ and decoder input = $y$ ). . . . .	44
3.10	Simulation and estimation results of Margulis (2048, 1024) code ( $\mathcal{C}_2$ ) decoded by 4-bit MSA with non-uniform quantization ( $C_{th} = 3$ , $I_{max} = 40$ , and decoder input = $y$ ). . . . .	45
3.11	Simulation and estimation results for $\mathcal{C}_2$ decoded by floating-point SPA with clipping at the check node outputs ( $I_{max} = 200$ ). . . . .	46
3.12	Simulation and estimation results for ETS classes (4, 2) and (5, 1) of $\mathcal{C}_3$ under floating-point SPA with clipping at the check node outputs ( $I_{max} = 200$ ). . . . .	47
3.13	Simulation and estimation results for $\mathcal{C}_3$ decoded by floating-point SPA with clipping at the check node outputs ( $I_{max} = 200$ ). . . . .	48
4.1	A (5, 3) LETS of the Tanner (155, 64) code whose edges are labeled in a systematic form. The edges of different layers are distinguished by different colors and line types. . . . .	54
4.2	Base graph of Example 5: The solid (red), dashed (blue) and dotted (purple) arrows, correspond to messages passed within layers $L_1$ to $L_3$ , respectively. The symbols next to the dotted arrows show the probability distributions of messages updated in the 3rd layer of the $\ell$ th iteration. . . . .	60
4.3	The computation trees of certain messages at different layers of the 1st iteration. . . . .	61
4.4	The external messages of a missatisfied CN are represented by a virtual VN. . . . .	63

4.5	The steps of constructing the digraph $D_l(\mathcal{S})$ of the (5, 3) LETS for a layered decoder from the flooding digraph $D_f(\mathcal{S})$ . (The colors (edge types) red (dotted), blue (dashed) and purple (dash-dotted) represent $L_1$ , $L_2$ and $L_3$ , respectively.) . . . . .	69
4.6	QC-LDPC codes used for simulations. The entries of the matrices, that are not equal to $-1$ , represent the right circular shift of the identity matrix to create the corresponding block of the parity-check matrix. The $-1$ entries represent zero blocks. . . . .	80
4.7	Simulation and estimation results of $\mathcal{C}_1$ for different saturation levels. The maximum number of iterations $I_{max} = 30$ . . . . .	81
4.8	The (5, 5) LETS structure of $\mathcal{C}_1$ in which the row layers for different CNs are shown. . . . .	81
4.9	The approximate estimate of the failure probability of the (5, 5) LETS of $\mathcal{C}_1$ for various row layered schedules at $E_b/N_0 = 6$ dB and saturation level 31.75. The schedules are sorted based on $\tilde{r}$ . . . . .	83
4.10	The simulation and estimation results of $\mathcal{C}_1$ for different row schedules. (The saturation level is 31.75, and the maximum number of iterations for layered schedules and the flooding schedule are set to 30 and 60, respectively.) . . . . .	83
4.11	The effect of various row layered schedules on the failure probability of different classes of LETSs in $\mathcal{C}_2$ ( $E_b/N_0 = 6$ dB, saturation level 15.75). . . . .	85
4.12	The effect of various row layered schedules on the total FER of $\mathcal{C}_2$ ( $E_b/N_0 = 6$ dB, saturation level 15.75). . . . .	85
4.13	The error estimation of the ten (7, 1) LETS groups of $\mathcal{C}_2$ for the two schedules (1, 2, 3, 4, 5, 6) and (4, 1, 5, 2, 3, 6) (saturation level 15.75). . . . .	86
4.14	Simulation and estimation results of $\mathcal{C}_2$ for the two schedules (1, 2, 3, 4, 5, 6) and (4, 1, 5, 2, 3, 6) (saturation level 15.75, $I_{max} = 30$ ). . . . .	87
5.1	A labelled (5,3) LETS of the Tanner (155, 64) code in which the flow of the state variables are shown. The VNs, mis-satisfied CNs and unsatisfied CNs are shown by black circles, white squares and gray squares, respectively. The directed edges related to the layers number 1, 2 and 3 are represented by red, blue and purple colors, respectively. . . . .	93

5.2	The VNs $v_1$ and $v_4$ related to Fig. 5.1 with the mis-satisfied CN $c_j$ connecting them. Figures (a) and (b) are related to virtual VNs, $v_{k'}$ and $v_{k''}$ , that are used during the update of VNs from layers $L_1$ and $L_3$ , respectively. . . . .	99
5.3	Base graph of Example 16: The solid (red), dashed (green), dotted (blue) and dashed-dotted (purple) arrows, respectively, illustrate the updating order which the column layered decoder perform, accordingly. . . . .	109
5.4	The error estimation related to (5, 5) LETS of $\mathcal{C}_1$ for various column layered schedules at $E_b/N_0 = 6$ dB. All the schedules result in the same dominant eigenvalue of the layered transition matrix, $\tilde{r} = 16.9536$ . The saturation level is 15.75. . . . .	111
5.5	The simulation and estimation results related to code $\mathcal{C}_1$ showing the effect of different schedules. The saturation level is 15.75 and the maximum number of iterations is 30. . . . .	111
5.6	The simulation and estimation results related to code $\mathcal{C}_2$ showing the effect of different schedules. The saturation level is 15.75. The maximum number of iterations is 30. . . . .	113
6.1	The histogram of the conditional error indicator function of a (5, 5) ETS of (640, 192) QC-LDPC code [3] at different iterations. . . . .	128
6.2	Simulation and estimation results for $\mathcal{C}_1$ decoded by SPA over the AWGN channel ( $I_{max} = 200$ ). . . . .	134
6.3	Simulation and extrapolated estimation results for $\mathcal{C}_1$ decoded by SPA over the AWGN channel ( $I_{max} = 200$ ). The extrapolated curves are obtained based on the results at $E_b/N_0 = 4.5$ dB . . . . .	136
6.4	Simulation and estimation results for $\mathcal{C}_2$ decoded by SPA over the AWGN channel ( $I_{max} = 200$ , $C_{th} = 25$ ). The extrapolation in $DSLME$ is based on the results at $E_b/N_0 = 2.75$ dB. . . . .	136
6.5	Simulation and estimation results for $\mathcal{C}_3$ decoded by SPA over the AWGN channel ( $I_{max} = 200$ ). The extrapolation in $DSLME$ is based on the results at $E_b/N_0 = 4.5$ dB. . . . .	137

# Nomenclature

Acronym	Description
AWGN	Additive White Gaussian Noise
BSC	Binary Symmetric Channel
CN	Check Node
DE	Density Evolution
ETS	Elementary Trapping Set
FNF	Frobenius Normal Form
IS	Importance Sampling
LDPC	Low Density Parity Check
LETS	Leafless Elementary Trapping Set
LIS	Local Importance Sampling
LLR	Log-Likelihood Ratio
MSA	Min-Sum Algorithm
QC	Quasi Cyclic
SNR	Signal to Noise Ratio
SPA	Sum Product Algorithm
TS	Trapping Set
TSLP	Trapping Set Layer Profile
VN	Variable Node

# Chapter 1

## Introduction

### 1.1 Motivation and Related Works

Finite-length low-density parity-check (LDPC) codes under iterative message passing algorithms suffer from error floor, i.e., as the channel quality improves, at some point, the error rate does not decrease as fast as its initial rate of decrease. The error floor problem of LDPC codes has been the topic of extensive research in recent years. In [1–14], decoders with improved error floor were devised. LDPC codes with low error floor were constructed in [15–27]. Characterization and enumeration of structures responsible for error floor were studied in [28–33], and techniques to estimate the error floor were developed in [34–49]. The error floor is attributed to some substructures of the code’s Tanner graph, referred to here collectively as *trapping sets* (TSs) [34]. Dominant trapping sets and their relative harmfulness are in general a function of the code, its Tanner graph, the channel, the decoding algorithm and the quantization scheme.

The literature on the error floor analysis/estimation of LDPC codes can be, in general, partitioned into two main categories. The first category, which contains the majority of the work in this area, is, mainly, based on importance sampling (IS) techniques, see, e.g., [3, 34, 36, 38, 40, 49]. IS techniques are basically fast Monte Carlo simulations over the Tanner graph of the code. The increase in speed is achieved by biasing the input distributions to increase the failure rate of a targeted TS. The second category contains the works that evaluate the failure rate of a TS structure independent of the code’s Tanner graph. Notable examples are the application of *critical number* to evaluate the harmfulness of TSs under hard-decision decoding algorithms over the binary symmetric channel (BSC) [37, 43], the linear state-space

model of [42, 44, 45] to evaluate the failure rate of TSs under SPA over the AWGN channel, and the notion of *absorbing set threshold* for quantized min-sum algorithms (MSAs) [50]. An important difference between the two categories is that while for the former, one needs the entire Tanner graph of the code to apply the IS, for the latter, the only required information is the TS structure (in isolation). The fact that the analysis of the harmfulness of TSs in the second category of techniques is independent of the Tanner graph of the code is interesting since it implies that the error floor can be estimated by the enumeration of dominant TSs of the code, using search algorithms such as those of [28–30, 32, 33], and then calculating the weighted sum of the contributions of such TSs to the error floor (the weights are the multiplicities of different TS structures). This approach not only provides a code-independent evaluation of TS harmfulness, but also is generally much faster than estimating the error floor using IS.

The proposed techniques in this thesis fit within the second category, in that, we evaluate the harmfulness of each TS structure independent of the Tanner graph of the code.

In [42], Sun proposed a linear state-space model to analyze the dynamics of ETSs in the error floor region over the AWGN channel. This model was based on the assumption that the decoder behavior outside an ETS can be well approximated by density evolution (DE) [51]. Schlegel and Zhang [44] proposed an improved linear state-space model in which an iteration-dependent linear gain was added to the model to account for the impact of external messages of missatisfied check nodes on the internal messages of an ETS. More recently, Butler and Siegel [45] refined and extended the linear state-space model and used it to analyze the effect of log-likelihood ratio (LLR) saturation on the error floor performance of LDPC codes with fixed variable-node degrees decoded by floating-point SPA.

The key point of the analysis of [42, 44, 45], which is the approximation of the dynamics of message-passing algorithms over a TS with a linear state-space model, provides an analytical tool to study the (relative) harmfulness of TSs. It is also shown that, for many scenarios, the linear state-space model results in accurate estimates of the error floor over the AWGN channel for SPA [44, 45]. One of the main advantages of this model is its simplicity in modeling the complex dynamics of iterative decoders in the error floor region. The linear state-space model however, has a number of

limitations. First, and probably the most important limitation of the linear state-space model is that it is not applicable to practical message-passing decoders in which the messages are quantized and represented in fixed-point (quantization is a non-linear operation). Other limitations are the assumptions on the variable-regularity of the codes, the decoding algorithm being limited to the SPA, and the channel being limited to the AWGN channel.

To the best of our knowledge, all the existing work on the theoretical analysis of error floor, including [42, 44, 45], is limited to two-phase message passing algorithms, also known as *flooding* or *parallel schedule*. In flooding schedule, each decoding iteration is divided into two parts. In the first (second) part of an iteration, all the variable (check) nodes compute their messages and pass them to their adjacent check (variable) nodes simultaneously. There are however a variety of message passing schedules which are advantageous to flooding in terms of performance, complexity or convergence speed [53–63, 71, 72]. An important category of schedules are *layered or serial schedules* [53, 54, 59, 60, 62, 63]. In such schedules, each iteration of message-passing consists of multiple sub-iterations performed serially. This allows a more frequent updating of the reliabilities in each iteration compared to the flooding schedule, which consequently results in a higher convergence speed. Moreover, due to the reuse of the same hardware for the implementation of different sub-iterations, the hardware resources required for the implementation of layered decoders are substantially lower than those of their flooding counterparts. Due to these attractive features, layered decoders are often used in practical applications along with quasi-cyclic (QC) LDPC codes. In such a combination, the row or column blocks of the parity-check matrix of the QC-LDPC code correspond to different layers. The layered decoder is then referred to as *row (horizontal) layered decoder* [53, 54, 57, 59, 61, 62] or *column (vertical) layered decoder* [60, 63–70], respectively. In [60], it has been shown that the convergence speed of both types of layered decoder can be twice that of a decoder with flooding schedule.

The vast majority of research on layered decoders is devoted to the issues concerning convergence speed and efficient implementations. In particular, the study of the error floor of such decoders has been mainly limited to empirical results [71–76]. In [71] and [72], the authors proposed a dynamic scheduling and a schedule diversity, respectively, that reduced the error floor. More recently, Raveendran and Vasic [73]

studied flooding and row layered Gallager-B algorithms applied to the (155, 64) Tanner code over the binary symmetric channel, and demonstrated that while the former decoder gets trapped in (5, 3) ETSs, the latter does not, and as a result the layered decoder has a superior error floor performance compared to the flooding decoder.

## 1.2 Summary of Contributions

In our first work, we evaluate the harmfulness of each TS structure independent of the Tanner graph of the code. Our approach here however, is quite different from and more general than the previous work in the category of code independent approaches. As it is mentioned before, the analysis of [42, 44, 45] is not applicable to practical fixed-point message-passing decoders. Also, it is limited by the assumptions that the code is variable-regular, the decoding algorithm is SPA and the channel is AWGN. We demonstrate that even when all these assumptions are met, the linear state-space model sometimes overestimates the error floor. Our results demonstrate that this over-estimation can be attributed to the linear approximation of check node operations in the model. Our technique resolves this issue by performing the check node operations accurately. In fact, to evaluate the harmfulness of a given TS structure for a given (quantized) iterative message-passing decoder, we perform the (exact) message-passing over the subgraph induced by the TS. In this process, to emulate the extrinsic messages entering the TS from the rest of the Tanner graph, we use density evolution (DE) to obtain the message distributions and then select the messages randomly from those distributions. One important component of the proposed technique, unlike the method of [42, 44, 45], is to account for the dependencies that exist among the extrinsic messages entering the TS from the rest of the Tanner graph. We first demonstrate that such dependencies in general increase the failure rate of a TS, and then propose a simple correlation model to incorporate the dependencies in the message-passing algorithm. To speed up the estimation process, we also use IS to bias the channel output distributions to increase the rate of TS failures. Our method can be applied to both regular and irregular LDPC codes, to any memoryless binary-input output-symmetric channel, and to any symmetric iterative decoding algorithm, and symmetric quantization scheme.

Motivated by the wide spread application of layered decoders and the fact that the behavior of such decoders in the error floor region is still not well understood,

in our next work, we aim at the theoretical analysis of the error floor of row layered decoders. In fact, to the best of our knowledge, this is the first work in which the dynamics of a soft layered decoder in the error floor region is theoretically analyzed. We start by developing a linear state-space model for ETSs that incorporates the layered nature of the decoding algorithm. We then use this model to study the dynamics of a saturating SPA in the vicinity of the TS over the AWGN channel. Our analysis shows that the harmfulness of a given TS, as well as the error floor of the code, can significantly change by changing the order in which the information of different layers, corresponding to different row blocks of the parity-check matrix, are updated. We then study the problem of finding the layer ordering that minimizes the error floor. As a result, we find orderings that result in error floors substantially lower than those of the decoder with flooding schedule. Connections are also made between the model parameters of the layered decoder and those of the flooding one. Finally, we compare our theoretical estimates of the error floor with Monte Carlo simulations for QC-LDPC codes, both variable-regular and irregular, and demonstrate a good match between the two.

We extend the linear state-space model of row layered decoders to the column layered decoders [60, 63–70]. We show that in column layered decoding, basically, the same ETS model of row schedule decoders can be applied. However, the model parameters must be calculated differently. We then use the model to optimize the error floor performance of column layered decoders. An efficient method will, also, be introduced to reduce the search space of the column layer permutations during the optimization process. Finally, we compare our theoretical estimates of the error floor with Monte Carlo simulations for QC-LDPC codes, both variable-regular and irregular, and demonstrate a good match between the two.

As mentioned earlier, the remarkable advantage of linear state-space model of [45] is its analytical nature in comparison with the methods involving some sort of simulations. This leads to a great simplification of modeling the complex dynamics of iterative decoders in the error floor region. However, the linearization of degree-2 CNs that is performed within this model has limited its performance [48]. In addition, the linearisation is done around the operational point of zero which in reality may not be the case and would lead to overestimation of the error floor. On the other hand, finding the true operating point is challenging as it must be determined based on most likely channel inputs that would result in the TS failure. These erroneous inputs are

a function of many other parameters such as message clipping and TS structures and are, generally, unknown. In our next work, we extend the state-space model of [45] and propose a method in which the operating points of different ETS structures at different iterations are, dynamically, approximated. In addition, we improve the model by incorporating the second order approximation (instead of linearization) at initial iterations. We, also, establish a connection between our proposed method and the code-dependant approach of Richardson [34] and demonstrate that our method can be considered as the fully-analytical counterpart of the semi-analytical method of [34]. One of the main advantages of our proposed method over the linear model of [45] is that the error rate curve, similar to Richardson's method, can be extrapolated meaning that the estimated error rate at a specific SNR can be used to extrapolate the rest of the error rate curve at other SNRs. This also implies that, within our model, the DE results as well as the model parameters at a specific SNR are obtained once and, then, are used for the rest of the SNR points. We also show that applying the extrapolation within the semi-analytical method of [34] might lead to approximation errors when the messages have higher saturation levels while our proposed method is more robust against extrapolation errors for different clipping thresholds.

### 1.3 Organization of the Thesis

The rest of the thesis is organized as follows: Chapter 2 is used for background material. In Chapter 3, we provide an in-depth analysis of the linear state-space method by incorporating a more accurate model for the incoming messages to the TS structure. We then propose an alternate code-independent technique for the error floor estimation of iterative LDPC decoders. In Chapter 4, we analyze the error floor of quasicyclic LDPC codes decoded by the sum-product algorithm with row layered message passing scheduling. We extend the row layered error floor analysis to the column layered message passing scheduling in Chapter 5. In Chapter 6, the semi-linear state-space model with dynamic operational points will be presented. Finally, the thesis is concluded in Chapter 7 and some ideas for the future works are proposed.

## Chapter 2

# Preliminaries

## 2.1 Notations

In this thesis, matrices and vectors are denoted by boldfaced upper case and lower case letters, respectively. The only exceptions are the LLR vectors where the symbol  $\mathbf{L}$  with different subscripts or superscripts is utilized for representation. All the vectors are assumed to be column vectors. A list of the notations and symbols used in this thesis is provided in Tables 2.1, 2.2, 2.3 and 2.4.

## 2.2 LDPC Codes and Decoding Algorithms

Consider a binary LDPC code  $\mathcal{C}$  with parity-check matrix  $\mathbf{H}$ . A codeword of  $\mathcal{C}$  is denoted by  $\mathbf{d}$ , and satisfies  $\mathbf{H}\mathbf{d} = \mathbf{0}$ . Let  $G = (V \cup C, E)$  be the Tanner graph representing  $\mathcal{C}$ , where  $V = \{v_1, v_2, \dots, v_n\}$  and  $C = \{c_1, c_2, \dots, c_m\}$  are the sets of variable nodes (VNs) and check nodes (CNs), respectively, and  $E = \{e_1, e_2, \dots, e_k\}$  is the set of edges. Suppose that  $\mathcal{C}$  is used for transmission over an AWGN channel using a binary phase shift keying (BPSK) modulation, where codeword bits  $d_i$  are mapped to modulated symbols  $u_i = (-1)^{d_i}$ . At the channel output, we thus have  $y_i = u_i + n_i$ , where  $\{n_i\}$  represents the noise and is a zero-mean Gaussian random process with independent and identically distributed (i.i.d.) values, each with variance  $\sigma_{ch}^2$ .

Given the noisy codeword at the receiver side, maximum likelihood (ML) sequence decoding tries to minimize the probability of frame error. In this method, a valid

**Table 2.1:** Notations and Symbols (Part A)

Notations	Descriptions
$\mathbf{H}$	$m \times n$ parity check matrix
$\mathbf{H}_b$	$m_b \times n_b$ base matrix
$\mathbf{d}$	A codeword
$\hat{\mathbf{d}}$	Estimated codeword
$I_{max}$	Maximum number of iterations
$L_i^{ch}$	Channel LLR
$L_\ell^{[i \leftarrow j]}$	$j$ th CN to $i$ th VN message at iteration $\ell$
$L_\ell^{[i \rightarrow j]}$	$i$ th VN to $j$ th CN message at iteration $\ell$
$\tilde{L}_\ell^{[i]}$	Total LLR of $i$ th VN at iteration $\ell$
$\omega_z$	Weighting factor in IS method
$N$	Number of trials in IS method
$m_s$	Number of state variables
$\mathcal{S}$	Set of VNs of a TS, $ \mathcal{S}  = a$
$\Gamma(\mathcal{S})$	Set of CNs of a TS
$\Gamma_o(\mathcal{S})$	Set of odd-degree CNs of a TS, $ \Gamma_o(\mathcal{S})  = b$
$\Gamma_e(\mathcal{S})$	Set of even-degree CNs of an ETS, $ \Gamma_e(\mathcal{S})  = \frac{m_s}{2}$
$\mathbf{x}^{(\ell)}$	$m_s \times 1$ state vector of a LETS in flooding decoder at the $\ell$ th iteration
$\Phi(\cdot)$	Element-wise missatisfied CN operator with element-wise first and second derivatives of $\Phi_x(\cdot)$ and $\Phi_{xx}(\cdot)$
$\bar{g}^{(\ell)}$	Missatisfied CN multiplicative gains in flooding decoder at the $\ell$ th iteration
$g_j^{(\ell)}$	Random gain corresponding to $j$ th virtual VN with mean and variance of $\bar{g}_j^{(\ell)}$ and $\sigma_{g_j}^2$
$g^{(\ell)}$	Random scalar gain at iteration $\ell$ with mean $\bar{g}^{(\ell)}$ and variance $\sigma_{g^{(\ell)}}^2$
$\xi_\ell$	Error indicator function of linear model with random gains at iteration $\ell$
$\mathbf{A}$	$m_s \times m_s$ transition matrix of a LETS in flooding decoder
$\mathbf{B}$	$m_s \times a$ matrix determining the channel input contributions to the state variables in flooding decoder
$\mathbf{B}_{ex}$	$m_s \times b$ matrix determining the contribution of unsatisfied CN inputs to state variables in flooding decoder
$\mathbf{C}$	$a \times m_s$ matrix determining the relation of state variables with the total LLR vector

**Table 2.2:** Notations and Symbols (Part B)

Notations	Descriptions
$\mathbf{D}_{ex}$	$a \times b$ matrix determining the contribution of unsatisfied CN inputs to the total LLR vector
$\mathbf{L}$	$a \times 1$ channel input vector in the LETS linear model
$\mathbf{L}_{ex}^{(\ell)}$	$b \times 1$ unsatisfied CN input vector in the LETS linear model at the $\ell$ th iteration
$\tilde{\mathbf{L}}^{(\ell)}$	$a \times 1$ total LLR vector in the LETS linear model
$\mathbf{P}$	A permutation matrix
$\rho(\mathbf{M})$	Spectral radius of a matrix $\mathbf{M}$
$r$	Dominant eigenvalue of the flooding transition matrix $\mathbf{A}$
$\mathbf{w}_1^T, \mathbf{u}_1$	Left and right eigenvectors corresponding to $r$
$\mathbb{C}$	Set of complex numbers
$\beta'_\ell$	Error indicator function of flooding decoder
$P_e\{\mathcal{S}\}$	Probability of failure of TS $\mathcal{S}$
$L_i$	$i$ th layer (used for both row and column layers) of a QC-LDPC code
$J$	Number of layers in a TS
$(\cdot)_k$	$k$ th element of a vector
$(m_{ch}, \sigma_{ch}^2)$	Mean and variance of the channel noise
$(m_{ex}^{(\ell)}, \sigma_{ex}^{2(\ell)})$	Mean and variance of the unsatisfied CN inputs at iteration $\ell$ of the flooding decoder
$n_{L_j}$	Number of state variables of a LETS within layer $j$
$\mathcal{A}_j$	$m_s \times m_s$ transition matrix of layer $j$
$\mathcal{B}_j$	$m_s \times a$ matrix indicating the contribution of the channel LLRs in the calculation of the state variables of the $j$ th layer
$\overset{\triangleleft}{\mathcal{B}}_{exj}, \overset{\triangleright}{\mathcal{B}}_{exj}$	$m_s \times b$ matrices indicating the contribution of $\mathbf{L}_{ex}^{(\ell-1)}$ and $\mathbf{L}_{ex}^{(\ell)}$ in updating the state variables within the $j$ th layer
$\mathbf{G}^{(\ell)}$	$m_s \times m_s$ matrix whose diagonal entries are the gains corresponding to the $m_s$ state variables
$\mathcal{G}_j^{(\ell)}$	$m_s \times m_s$ gain matrix corresponding to the $j$ th layer of iteration $\ell$
$\tilde{\mathbf{x}}^{(\ell,j)}$	Layered decoder state vector of layer $j$ at iteration $\ell$

**Table 2.3:** Notations and Symbols (Part C)

Notations	Descriptions
$\tilde{\mathbf{B}}^{(\ell)}$	$m_s \times a$ matrix indicating the contribution of the channel LLRs in calculation of the state variables at the $\ell$ th iteration of the layered decoder
$\mathbf{B}_{ex}^{\triangleleft(\ell)}, \mathbf{B}_{ex}^{\triangleright(\ell)}$	$m_s \times b$ matrices illustrating the contribution of $\mathbf{L}_{ex}^{(\ell-1)}$ and $\mathbf{L}_{ex}^{(\ell)}$ in updating the state variables at the $\ell$ th iteration of the layered decoder
$\hat{\psi}_\ell^{[k' \rightarrow j]}$	Probability distribution of the messages from virtual VN $v_{k'}$ to missatisfied CN $c_j$ at iteration $\ell$
$P_{inv,\ell}^{[k' \rightarrow j]}$	Probability of polarity inversion in the messages from virtual VN $v_{k'}$ to missatisfied CN $c_j$ at iteration $\ell$
$\bar{g}_{c_j}^{(\ell)}, \bar{g}'_{c_j}{}^{(\ell)}$	Average gains of missatisfied CN $c_j$ at the $\ell$ th iteration of layered decoder before and after adding the polarity inversion, respectively
$\tilde{\mathbf{A}}_{J \rightarrow 1}$	Transition matrix of a LETS with $J$ layers in layered decoding
$\tilde{r}$	Dominant eigenvalue of $\tilde{\mathbf{A}}_{J \rightarrow 1}$
$\tilde{\mathbf{w}}_1^T, \tilde{\mathbf{u}}_1$	Left and right eigenvectors corresponding to $\tilde{r}$
$\tilde{\mathbf{A}}$	The only irreducible diagonal block of the Frobenius normal form of $\tilde{\mathbf{A}}_{J \rightarrow 1}$ for LETSs that are not simple cycles
$\tilde{\boldsymbol{\omega}}_1^T, \tilde{\boldsymbol{\nu}}_1$	Dominant left and right eigenvectors of $\tilde{\mathbf{A}}$
$D_l$	A LETS digraph of the layered decoder with nodes $V_l$ and edges $E_l$
$D_f$	A LETS digraph of the flooding decoder with nodes $V_f$ and edges $E_f$
$n_z$	Number of zero columns of the transition matrix of the layered decoder
$(\mathbf{m}_{ex}^{(\ell)}, \boldsymbol{\Sigma}_{ex}^{(\ell)})$	$b \times 1$ mean vector and $b \times b$ covariance matrix of the inputs from unsatisfied CNs of a LETS at iteration $\ell$
$\tilde{\beta}^{(\ell)}$	Error indicator function of layered decoder
$\Upsilon_i$	Size of the $i$ th TS group
$\Pi_J$	Set of all the $J!$ permutations of the layers of a LETS
$\overleftarrow{\psi}_\ell^j$	Average distribution of CN to VN messages at layer $j$ of iteration $\ell$
$\overrightarrow{\psi}_\ell^j$	Average distribution of VN to CN messages at layer $j$ of iteration $\ell$

**Table 2.4:** Notations and Symbols (Part D)

Notations	Descriptions
$\eta_{\ell \theta'}$	State-variables conditional operational point at iteration $\ell$
$\hat{\mathbf{w}}_1$	Normalized left eigenvector of $\mathbf{A}$ , $\sum_{k=1}^{m_s} (\hat{\mathbf{w}}_1)_k = 1$
$\mathbf{n}_a$	$a \times 1$ input noise vector of a TS
$\mathbf{n}'$	$a \times 1$ residual noise vector of a TS with covariance $\Sigma_{\mathbf{n}'}$
$\dot{\mathbf{g}}^{(\ell)}$	$m_s \times 1$ random vector with elements $\dot{g}_j^{(\ell)}$ (mean: $\bar{g}_j^{(\ell)}$ , variance: $\sigma_{\dot{g}_j^{(\ell)}}^2$ ) related to constant term in Taylor expansion
$\dot{g}^{(\ell)}$	Random variable defined as $\dot{g}^{(\ell)} = \hat{\mathbf{w}}_1^T \dot{\mathbf{g}}^{(\ell)}$ (mean: $\bar{g}_j^{(\ell)}$ , variance: $\sigma_{\dot{g}_j^{(\ell)}}^2$ ) related to constant term in Taylor expansion
$\hat{\mathbf{G}}^{(\ell)}, \check{\mathbf{G}}^{(\ell)}$	$m_s \times m_s$ diagonal matrices with elements $\hat{g}_j^{(\ell)}$ (mean: $\bar{g}_j^{(\ell)}$ , variance: $\sigma_{\hat{g}_j^{(\ell)}}^2$ ) and $\check{g}_j^{(\ell)}$ (mean: $\bar{g}_j^{(\ell)}$ , variance: $\sigma_{\check{g}_j^{(\ell)}}^2$ ) related to linear and squared terms in Taylor expansion
$\hat{g}^{(\ell)}, \check{g}^{(\ell)}$	Random scalar gains with means and variances $(\bar{g}^{(\ell)}, \sigma_{\hat{g}^{(\ell)}}^2)$ , $(\bar{g}^{(\ell)}, \sigma_{\check{g}^{(\ell)}}^2)$ corresponding to the linear and squared terms in Taylor expansion of missatisfied CNs at iteration $\ell$
$\theta'$	projection of TS input noise vector on an all-one vector
$P_e\{\mathcal{S} \theta'\}$	Conditional failure probability of TS $\mathcal{S}$
$\xi'_\ell$	Conditional error indicator function at iteration $\ell$

codeword with minimum Euclidean distance to the received codeword is chosen as an approximation of the transmitted block. However, the complexity of this approach increases exponentially with the block length [95]. Due to the large length of the practical codes, ML decoding of the LDPC codes is not feasible. As a result, the decoding operation is done according to the message passing algorithms [96]. Some of the most famous decoding algorithms are explained in the following.

For decoding algorithms, we consider symmetric iterative message-passing algorithms including SPA, MSA and the modifications of MSA such as normalized MSA [83] with the following channel LLR values as the input:

$$L_i^{ch} = 2y_i/\sigma_{ch}^2. \quad (2.1)$$

The message sent from VN  $v_i$  to CN  $c_j$  at iteration  $\ell$  is given by

$$L_\ell^{[i \rightarrow j]} = L_i^{ch} + \sum_{k \in M(i) \setminus j} L_{\ell-1}^{[i \leftarrow k]}, \quad (2.2)$$

where  $M(i) \setminus j$  represents the set of CNs adjacent to VN  $v_i$  excluding  $c_j$ , and  $L_{\ell-1}^{[i \leftarrow k]}$  denotes the message sent from CN  $c_k$  to VN  $v_i$  at iteration  $\ell - 1$ . At the first iteration of the algorithm, all the messages sent from check nodes to variable nodes at iteration  $\ell - 1$  are assumed to be zero in (2.2). The CN to VN messages in SPA are computed as

$$L_\ell^{[i \leftarrow j]} = 2 \tanh^{-1} \left[ \prod_{k \in N(j) \setminus i} \tanh \frac{L_\ell^{[k \rightarrow j]}}{2} \right], \quad (2.3)$$

where  $N(j) \setminus i$  represents the set of VNs adjacent to CN  $c_j$  excluding  $v_i$ . At the end of each iteration, for each  $i \in \{1, \dots, n\}$ , first, the total LLR is calculated by

$$\tilde{L}_\ell^{[i]} = L_i^{ch} + \sum_{k \in M(i)} L_\ell^{[i \leftarrow k]}, \quad (2.4)$$

and then, a hard decision is made by

$$\hat{d}_i = [\text{sign}(\tilde{L}_\ell^{[i]}) + 1]/2. \quad (2.5)$$

If the decoded block,  $\hat{\mathbf{d}}$ , at the end of iteration  $\ell \leq I_{max}$ , is a codeword, i.e., if  $\mathbf{H}\hat{\mathbf{d}} = \mathbf{0}$ , then the decoding is terminated successfully. ( $I_{max}$  is the maximum number

of iterations.) Otherwise, if iteration  $I_{max}$  is completed and still no codeword is found, then a decoding failure is declared.

To circumvent the numerical errors in the calculation of  $\tanh(x)$  in (2.3), similar to [9, 77], we use the following equivalent operation to calculate CN to VN messages:

$$L_\ell^{[i \leftarrow j]} = \boxplus_{k \in N(j) \setminus i} L_\ell^{[k \rightarrow j]}, \quad (2.6)$$

in which the pairwise box-plus operator,  $\boxplus$ , is defined as

$$\begin{aligned} x_1 \boxplus x_2 &= \ln \left( \frac{1 + e^{x_1 + x_2}}{e^{x_1} + e^{x_2}} \right) \\ &= \text{sign}(x_1)\text{sign}(x_2) \cdot \min(|x_1|, |x_2|) + s(x_1, x_2). \end{aligned} \quad (2.7)$$

The term  $s(x_1, x_2)$  in (2.7) is given by

$$s(x_1, x_2) = \ln \left( 1 + e^{-|x_1 + x_2|} \right) - \ln \left( 1 + e^{-|x_1 - x_2|} \right). \quad (2.8)$$

The MSA is an approximation of SPA and, generally, its performance in the waterfall region is worse than SPA. To improve the performance of MSA compared to the SPA, different variants of this algorithm such as normalized MSA [83] have been proposed. The CN to VN messages of the normalized MSA are calculated as follows

$$L_\ell^{[i \leftarrow j]} = \left[ \prod_{k \in N(j) \setminus i} \text{sign}(L_\ell^{[k \rightarrow j]}) \right] \cdot \alpha \cdot \min_{k \in N(j) \setminus i} |L_\ell^{[k \rightarrow j]}|, \quad (2.9)$$

where  $0 < \alpha \leq 1$  is called the correction factor.

## 2.3 Quantization

In practice, often, the quantized versions of the decoding algorithms are utilized. For quantization, unless specified otherwise, we consider a uniform quantizer as described in [83], where the channel LLRs are first symmetrically clipped based on the clipping threshold,  $C_{th}$ , within the range  $[-C_{th}, C_{th}]$ , and then uniformly quantized into  $2^q - 1$  intervals, each with width of  $\Delta = C_{th}/[2^{(q-1)} - 0.5]$ . Within the decoder, the negative and positive messages exchanged between the variable and check nodes are clipped

to  $-(2^{q-1} - 1)$  and  $(2^{q-1} - 1)$ , respectively.

## 2.4 Flooding and Layered Schedules

It should be noted that Equations (2.2), (2.3), (2.6) and (2.9) are edgewise operations, and can thus be executed within various scheduling frameworks. In decoders with *flooding schedule*, at the first half of each iteration, Equation (2.2) is executed for all the VN to CN messages. In the second half of the iteration, all the CN to VN messages are updated based on (2.3). For decoders with *row layered schedule*, the CNs are partitioned into different subgroups (row layers). Within each CN subgroup, messages are only generated on the edges between the CNs in the subgroup and their adjacent VNs while the other edges in the graph remain inactive. After the messages within a subgroup are generated, the updated reliabilities are used in the following layers. With respect to *Column layered schedule* decoders, the VNs are divided into different subgroups (column layers). At each layer, based on the incoming messages from the adjacent CNs, the total LLRs of the corresponding VNs of the current layer are updated. However, the incoming CN to VN messages, themselves, are calculated based on the extrinsic LLRs received from the VNs of other layers that all or some of them might have been updated within previous layers. Finally, the updated reliabilities of the current layer are used within the subsequent layers. In Algorithms 1 and 2, the steps of the row and column layered SPA are presented for a QC-LDPC code. In the following chapters, we consider QC-LDPC codes whose parity-check matrices  $\mathbf{H}$  consist of an  $m_b \times n_b$  array of circulant permutation matrices (CPMs) of size  $p \times p$  and zero matrices of the same size. The Tanner graph of such codes can be considered as a cyclic  $p$ -lifting of a bipartite *base graph* with  $n_b$  VNs and  $m_b$  CNs. Each layer of the row (column) layered decoder in this case corresponds to one row (column) block of  $\mathbf{H}$ , and thus, there are  $m_b$  ( $n_b$ ) layers. It should be noted that the equation in Line 7 of Algorithm 1 is derived by combining (2.2) and (2.4). As can be seen, the total LLR of each VN may be updated several times within an iteration. So, for simplicity, the iteration index,  $\ell$ , is removed from the symbol  $\tilde{L}^{[\ell]}$ .

**Algorithm 1** Row Layered Box-Plus Decoding Algorithm

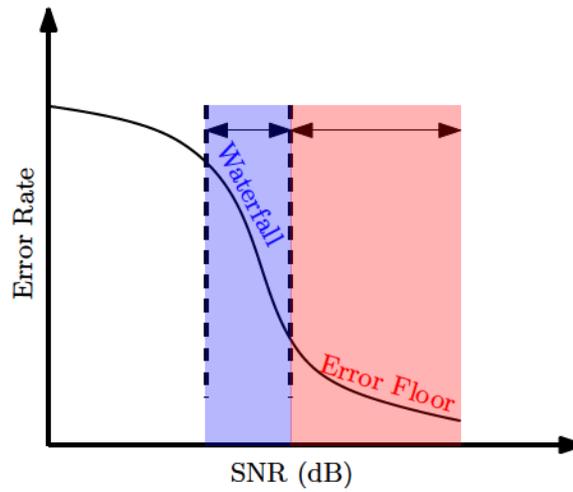
---

```

1: Input: Channel LLRs for all the VNs.
2: Initialization: All the total LLRs are initialized with the channel LLRs,  $\tilde{L}^{[i]} = L_i^{ch}$  for VNs  $i = 1 : n_b p$ .
3: for iteration  $\ell = 1, \dots, I_{max}$  do
4:   for The CN group (layer) number  $z = 1 : m_b$  do
5:     for CNs  $j = (z - 1)p + 1, \dots, zp$ , in layer  $z$  do
6:       for all VNs  $i \in N(j)$  do
7:          $L_\ell^{[i \rightarrow j]} = \tilde{L}^{[i]} - L_{\ell-1}^{[i \leftarrow j]}$ .
8:       end for
9:       for all VNs  $i \in N(j)$  do
10:         $L_\ell^{[i \leftarrow j]} = \boxplus_{k \in N(j) \setminus i} L_\ell^{[k \rightarrow j]}$ ,
11:         $\tilde{L}^{[i]} = L_\ell^{[i \rightarrow j]} + L_\ell^{[i \leftarrow j]}$ .
12:      end for
13:    end for
14:  end for
15:  Hard-decision:
16:  for all VNs  $i = 1, \dots, n_b p$  do
17:     $\hat{d}_i = [\text{sign}(\tilde{L}^{[i]}) + 1]/2$ .
18:  end for
19:  if  $\text{H}\hat{\mathbf{d}} = \mathbf{0}$  then
20:    Break.
21:  end if
22: Output:  $\hat{\mathbf{d}}$ 

```

---



**Figure 2.1:** A typical error rate curve of an LDPC code versus SNR that consists of two separate region known as the waterfall and the error floor, respectively.

---

**Algorithm 2** The Column Layered Box-Plus Decoding Algorithm
 

---

- 1: **Input:** Channel LLRs for all the VNs.
  - 2: **Initialization:** All the VN to CN messages are initialized with channel LLRs,  $L_i^{ch}$  for  $i = 1, \dots, n_b p$ .
  - 3: **for** iteration  $\ell = 1, \dots, I_{max}$  **do**
  - 4:     **for** The column group  $z = 1, \dots, n_b$  **do**
  - 5:         **for** VNs  $i = (z - 1)p + 1, \dots, zp$ , in layer  $z$  **do**
  - 6:             **for all** CNs  $j \in M(i)$  **do**
  - 7:                  $L^{[i \leftarrow j]} = \boxplus_{k' \in N(j) \setminus i} L^{[k' \rightarrow j]}$ ,
  - 8:             **end for**
  - 9:              $L^{[i \rightarrow j]} = L_i^{ch} + \sum_{k \in M(i) \setminus j} L^{[i \leftarrow k]}$ .
  - 10:              $\tilde{L}^{[i]} = L_i^{ch} + \sum_{k \in M(i)} L^{[i \leftarrow k]}$ .
  - 11:         **end for**
  - 12:     **end for**  
        Hard-decision:
  - 13:     **for all** VNs  $i = 1, \dots, n_b p$  **do**
  - 14:          $\hat{d}_i = \lceil \text{sign}(\tilde{L}^{[i]}) + 1 \rceil / 2$ .
  - 15:     **end for**
  - 16:     **if**  $\text{H}\hat{\mathbf{d}} = \mathbf{0}$  **then**
  - 17:         Break.
  - 18:     **end if**
  - 19: **end for**
  - 20: **Output:**  $\hat{\mathbf{d}}$
-

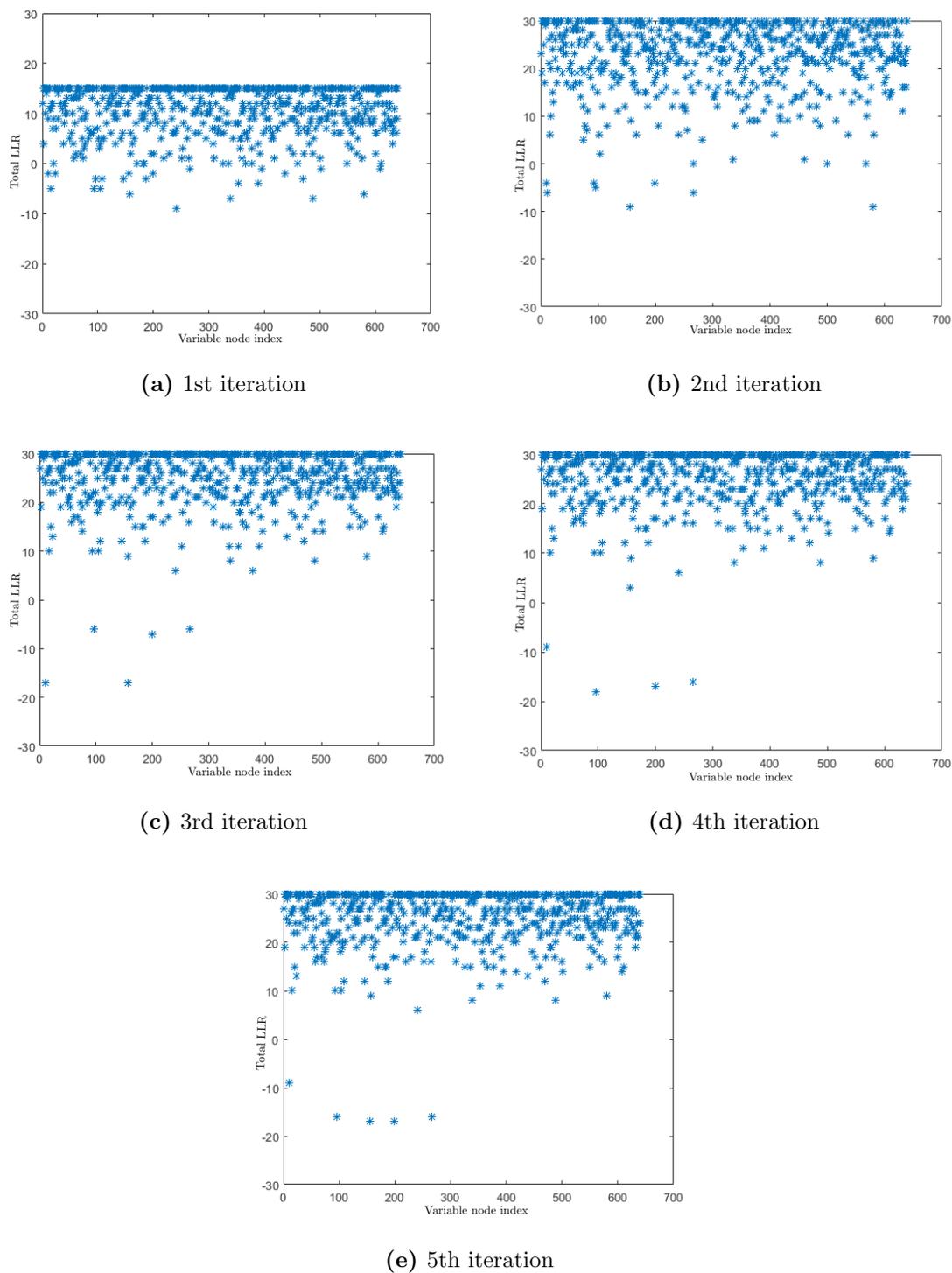
## 2.5 Trapping Sets (TSs)

The error rate performance of LDPC codes is, typically, divided into two separate region, namely waterfall and error floor, Fig. 2.1. The waterfall is close to the code threshold where the error rate curve has a sharp slope. The error floor region happens at higher signal to noise ratios (SNRs) in which the error rate does not follow the initial rapid reduction.

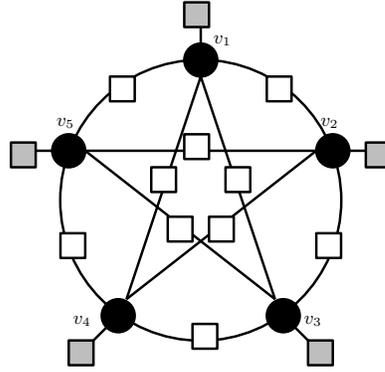
It is well-known that in the error floor region, the majority of the errors of iterative decoding algorithms occur as a result of the decoder getting trapped within a subset  $\mathcal{S}$  of VNs, i.e., although all the bits outside  $\mathcal{S}$  have correct values for sufficiently large number of iterations, all the bits inside  $\mathcal{S}$  are in error. In this case, the set  $\mathcal{S}$  is called a *trapping set (TS)*. Let  $\mathcal{S}$  be a TS, and  $\Gamma(\mathcal{S})$  be the set of neighboring CNs of  $\mathcal{S}$  in the Tanner graph  $G$ . The *induced subgraph* of  $\mathcal{S}$  in  $G$ , denoted by  $G(\mathcal{S})$ , is a graph whose nodes and edges are  $\mathcal{S} \cup \Gamma(\mathcal{S})$  and  $\{v_i c_j \in E : v_i \in \mathcal{S}, c_j \in \Gamma(\mathcal{S})\}$ , respectively. The set of CNs,  $\Gamma(\mathcal{S})$ , can be partitioned into even-degree CNs,  $\Gamma_e(\mathcal{S})$ , and odd-degree CNs,  $\Gamma_o(\mathcal{S})$ . The members of  $\Gamma_o(\mathcal{S})$  and  $\Gamma_e(\mathcal{S})$  are referred to as *unsatisfied check nodes* and *missatisfied check nodes*, respectively. If all the CNs in  $\Gamma(\mathcal{S})$  have degrees 1 or 2, the TS is called an *elementary TS (ETS)*. An ETS  $\mathcal{S}$  is referred to as a *leafless ETS (LETS)* if each variable node in  $\mathcal{S}$  is connected to at least two missatisfied CNs. Leafless ETSs are known to be the most harmful TSs over the AWGN channel [45, 52]. Similar to [45], we thus limit our discussions in this work to LETSs. In the following chapters, for simplicity, sometimes, we use the term “TS” instead of “LETS.”

Trapping sets are often identified by their size,  $|\mathcal{S}| = a$ , and the number of unsatisfied CNs in their subgraph,  $|\Gamma_o(\mathcal{S})| = b$ . In this case, the TS is said to belong to the  $(a, b)$  class, or is referred to as an  $(a, b)$  TS.

**Example 1.** *An example of a decoder failure on a TS related to a (640, 192) QC-LDPC code with variable-degree 5 and irregular check-degree is illustrated in Fig. 2.2. The vertical and horizontal axes represent the total LLR and VN indices, respectively. Assuming the all-zero codeword transmitted, as iterations go on, all the total LLRs related to different VNs grow toward the positive direction except 5 of them which remain negative causing the decoder to be trapped. These 5 VNs, causing the failure, induce a (5, 5) LETS whose structure shown in Fig. 2.3.*



**Figure 2.2:** Illustration of a MS decoder failure on a (5, 5) ETS belonging to a QC-LDPC (640, 192) code. Each figure represents the total likelihood of VNs at a given iteration (vertical axis) versus the VN indices (horizontal axis), respectively.



**Figure 2.3:** The subgraph of the  $(5, 5)$  ETS in the Tanner graph of the  $(640, 192)$  LDPC code. The VNs, missatisfied CNs and unsatisfied CNs are represented by black circles, white squares and gray squares, respectively.

The *minimum distance* of an LDPC code, like that of any other linear block code, is characterized by the minimum weight of non-zero codewords. As it was mentioned earlier, in ML decoder, a valid codeword with minimum Euclidean distance to the received noisy block is chosen as an estimation of the transmitted codeword. Therefore, the lower the *minimum distance* of a code, the higher the probability of error in ML decoder will be. Within the framework of TSs, a non-zero codeword of weight  $a$  can also be viewed as an  $(a, 0)$  TS in the code Tanner graph where the non-zero elements of the codeword correspond to the variable nodes of the TS and the number of unsatisfied CNs is zero. Suppose two codewords,  $(a_1, 0)$  and  $(a_2, 0)$  with  $a_1 < a_2$  belong to a specific code. Assuming the all zero codeword is transmitted, in ML decoder, the probability that the received noisy codeword being, falsely, decoded into  $(a_1, 0)$  is higher than false decoding into  $(a_2, 0)$ . Therefore, it is said that  $(a_1, 0)$  is more harmful than  $(a_2, 0)$  as it results in higher rate of decoding failure. Unlike the straightforward comparison of the relative harmfulness of codewords in ML decoder, determining the relative harmfulness of TSs is difficult as it is a function of channel, decoding algorithm, TSs graphical structure in the Tanner graph, precision (quantization) of message passing algorithm and so on. For example, two TSs belonging to the same  $(a, b)$  class can have significantly different harmfulness. In addition, a TS with higher number of VNs might be more harmful than the one with smaller number of VNs and there is not such a general rule as in the framework of ML decoding where the minimum-weight codewords are the most harmful error prone structures. One of our main goals in this thesis is to identify the relative harmfulness of various TS structures.

## 2.6 Importance Sampling

Importance sampling (IS) is a Monte Carlo method used in the simulation of rare events. The idea is to use a biased probability distribution to increase the occurrence frequency of the rare event, and thus to speed up the simulation. The effect of the biased distribution is then removed by reweighing the computed average. Importance sampling has been used to estimate the error floor of LDPC codes [34, 36, 38, 45]. To measure the contribution of an  $(a, b)$  TS to the error probability using IS, we have

$$P_e\{\mathcal{S}\} = \frac{1}{N} \sum_{z=1}^N I_z \omega_z, \quad (2.10)$$

where  $N$  is the number of trials,  $I_z$  is an indicator function of error, and  $\omega_z$  is the weighting factor. For the AWGN channel, if the noise distribution is biased by shifting its mean to  $-\mu$  for  $\mu > 0$ , we have

$$\omega_z = \frac{e^{\frac{-1}{2\sigma_{ch}^2} \left[ \sum_{k=1}^a (y_{k,z} - 1)^2 \right]}}{e^{\frac{-1}{2\sigma_{ch}^2} \left[ \sum_{k=1}^a (y_{k,z} - 1 + \mu)^2 \right]}}, \quad (2.11)$$

where  $y_{k,z}$  is the channel output corresponding to the  $k$ -th variable node of the TS at the  $z$ -th trial, and  $\sigma_{ch}^2$  is the channel noise power. For the quantized decoders in which the decoder inputs (from the channel) are from a finite alphabet, the weight  $\omega_z$  is calculated by

$$\omega_z = \frac{\prod_{k=1}^a p_z(k)}{\prod_{k=1}^a p'_z(k)}, \quad (2.12)$$

where  $p_z(k)$  and  $p'_z(k)$  are the probability mass functions corresponding to the decoder input of the  $k$ -th variable node of the TS at the  $z$ -th trial, for the unbiased and biased noise distributions, respectively. As a special case of the above equation, for the BSC, the weight  $\omega_z$  is given by

$$\omega_z = \frac{p^{\epsilon_z} (1-p)^{a-\epsilon_z}}{p'^{\epsilon_z} (1-p')^{a-\epsilon_z}}, \quad (2.13)$$

where  $\epsilon_z$  is the number of channel errors corresponding to the variable nodes of the TS at the  $z$ -th trial, and  $p$  and  $p'$  are the channel crossover probabilities of the unbiased and biased cases, respectively.

Without loss of generality, to measure the error rate performance of the decoder, we assume that the all-zero codeword is transmitted.

## 2.7 Density Evolution

For LDPC codes with large block lengths, various realization of a code ensemble, almost, perform identically in the waterfall region. This suggests that the performance of the LDPC codes could be evaluated theoretically rather than by simulation. Density evolution (DE) is a technique which basically tracks the probability distribution functions of the messages exchanged at different iterations of the MP algorithm. This method has been developed upon the assumptions that all zero codewords are transmitted, all the messages are independent, the code length is infinity and the Tanner graph of the code is cycle-free (Tree). Following the concentration results presented in [51], one might be able to predict the performance of a long LDPC code in the waterfall region based on its degree distributions specifications. This property known as the *asymptotic performance* of an LDPC code ensemble.

In this thesis, density evolution, unlike its conventional application, is utilized in the error floor region. In other words, to analyze TS failures within the code-independent framework, one needs to obtain the probability distribution of the messages entering the TS subgraph via unsatisfied and missatisfied CNs. Such distributions are obtained based on DE method with the underlying assumption that the all-zero codeword is transmitted, and that the surrounding neighbourhood of the TS is tree-like.

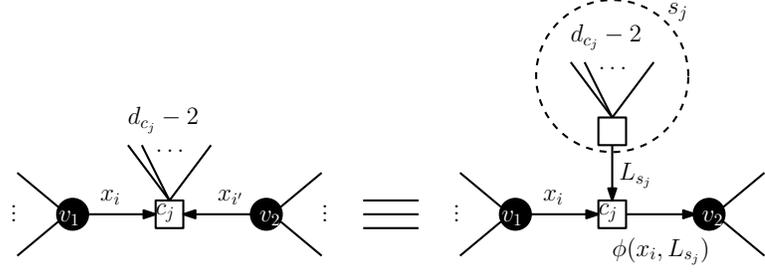
## Chapter 3

# Error Floor Estimation of LDPC Decoders - A Code Independent Approach to Measuring the Harmfulness of Trapping Sets

The organization of this chapter is as follows: In Section 3.1, we discuss the state-space model. We start by the non-linear form of the equations and then describe the process of linearization. The linear model developed here is an extension of the one in [42, 44, 45], in that, we consider the randomness of the extrinsic messages entering the TS through the incorporation of gain variances. Using this extended model, we then analyze the effect of dependencies among extrinsic messages on the failure probability of the TS. Section 3.2 is devoted to discussions on the limitations of the linear state-space model in accurately estimating the error probability. The proposed technique and its complexity analysis are then presented in Section 3.3. In Section 3.4, we provide extensive simulation results to demonstrate the accuracy of the proposed method and its comparison with the linear state-space model.

### 3.1 State-Space Model of an ETS

In this section, we first present the non-linear state-space model of an ETS. We then discuss the derivation of the linear model of [45]. Next, we extend the linear model of [45] to incorporate the randomness of the extrinsic messages entering the ETS through the missatisfied check nodes. This is achieved through the introduction of gain variances in the model, and is needed to study the effect of dependencies among



**Figure 3.1:** A missatisfied check node connecting two variable nodes of an ETS. By combining the  $d_{c_j} - 2$  external messages into  $L_{s_j}$ , the equivalent check node has degree 3 regardless of the actual check node degrees of the LDPC code. The message  $L_{s_j}$  is assumed to be sent by a virtual variable node  $s_j$ .

extrinsic messages. Finally, the extended formulation is used to explain how the dependency among the extrinsic messages increases the failure rate of an ETS based on the linear model.

### 3.1.1 From non-linear to linear state-space model

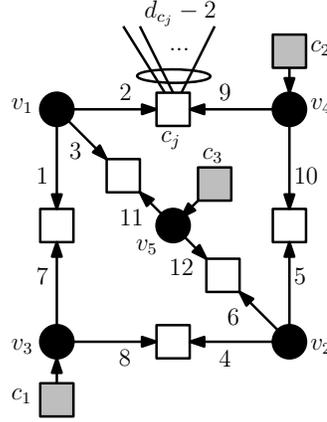
The message-passing algorithm in any ETS can be described by the following non-linear recursive equations, which we refer to as *non-linear state-space model*:

$$\mathbf{x}^{(0)} = \mathbf{B}\mathbf{L}, \quad (3.1)$$

$$\mathbf{x}^{(\ell)} = \mathbf{A}\Phi(\mathbf{x}^{(\ell-1)}, \mathbf{B}_s\mathbf{L}_s^{(\ell)}) + \mathbf{B}\mathbf{L} + \mathbf{B}_{ex}\mathbf{L}_{ex}^{(\ell)} \text{ for } \ell \geq 1, \quad (3.2)$$

$$\tilde{\mathbf{L}}^{(\ell)} = \mathbf{C}\Phi(\mathbf{x}^{(\ell-1)}, \mathbf{B}_s\mathbf{L}_s^{(\ell)}) + \mathbf{L} + \mathbf{D}_{ex}\mathbf{L}_{ex}^{(\ell)} \text{ for } \ell \geq 1. \quad (3.3)$$

In the model, all the vectors are column vectors. The elements of the *state vector*  $\mathbf{x}^{(\ell)}$ , called *state variables*, at iteration  $\ell$  are the LLR messages passed over different edges of the TS subgraph towards the missatisfied check nodes. The number of state variables for an  $(a, b)$  ETS is  $m_s = \sum_{i=1}^a d_{v_i} - b$ , where  $d_{v_i}$  is the degree of the  $i$ -th variable node in the TS. The first input vector  $\mathbf{L}_{ex}^{(\ell)}$  has  $b$  elements, where each element represents the message from one of the unsatisfied (degree-1) check nodes at iteration  $\ell$ . The second input vector  $\mathbf{L}$  contains the  $a$  channel LLRs. The last input vector to the model is  $\mathbf{L}_s^{(\ell)}$  with  $\frac{m_s}{2}$  entries that are the external messages entering the missatisfied check nodes from the virtual variable nodes  $s_j$  at iteration  $\ell$  (see, Fig. 3.1). In the model, the missatisfied check node operation is denoted by the element-wise operator  $\Phi(\cdot)$ . The initialization of the state variables is performed in



**Figure 3.2:** A (5,3) LETS. The VNs, missatisfied CNs and unsatisfied CNs are shown by black circles, white squares and gray squares, respectively. The external connections for one of the missatisfied CNs are also shown.

(5). Equation (6) is responsible for the evolution of state variables from iteration  $\ell - 1$  to iteration  $\ell$ . In each iteration  $\ell$ , Equation (7) is used to determine the total LLR values at the  $a$  variable nodes of the ETS. Binary matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ ,  $\mathbf{B}_s$ ,  $\mathbf{B}_{ex}$ , and  $\mathbf{D}_{ex}$  represent the relationships among different variables within the model. In particular, matrix  $\mathbf{A}$ , referred to as the *transition matrix*, relates the state variables at iteration  $\ell$  to those at iteration  $\ell - 1$ .

The following lemma is a simple consequence of the definition of matrices  $\mathbf{B}$  and  $\mathbf{B}_s$ , and will be used in a number of derivations in Section 6.1 of Chapter 6.

**Lemma 1.** *Let the number of missatisfied CNs connected to the  $i_{th}$  VN of an  $(a, b)$  ETS be denoted by  $\zeta_{v_i}$ , i.e.,  $\sum_{i=1}^a \zeta_{v_i} = m_s$ . Then the matrices  $\mathbf{B}$  and  $\mathbf{B}_s$  in the state-space model have the following properties:*

- a) *Each column of the  $m_s \times a$  matrix  $\mathbf{B}$  corresponds to one VN  $v_i$ ,  $i = 1, \dots, a$ , of the TS. The column corresponding to  $v_i$ ,  $i = 1, \dots, a$ , consists of  $\zeta_{v_i}$  ones and  $m_s - \zeta_{v_i}$  zeros, where the row indices of non-zero entries for different columns are different, i.e., they do not overlap.*
- b) *Each column of the  $m_s \times \frac{m_s}{2}$  matrix  $\mathbf{B}_s$  corresponds to one missatisfied CN  $c_j$ ,  $j = 1, \dots, \frac{m_s}{2}$ , of the TS. The column corresponding to  $c_j$ ,  $j = 1, \dots, \frac{m_s}{2}$ , consists of 2 ones and  $m_s - 2$  zeros, where the row indices of non-zero entries at different columns are different.*



$$\begin{aligned}
\mathbf{B} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B}_{\text{ex}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{B}_s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \\
\mathbf{C} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{D}_{\text{ex}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\end{aligned}$$

In the case of failure at high signal-to-noise ratio (SNR) values, where the internal messages of an ETS have much smaller magnitudes compared to the extrinsic messages, the missatisfied check node operations can be approximated by the linear term in the Taylor expansion (around zero) as follows:

$$\mathbf{x}^{(0)} = \mathbf{B}\mathbf{L}, \quad (3.4)$$

$$\mathbf{x}^{(\ell)} = \mathbf{A}\mathbf{G}^{(\ell)}\mathbf{x}^{(\ell-1)} + \mathbf{B}\mathbf{L} + \mathbf{B}_{\text{ex}}\mathbf{L}_{\text{ex}}^{(\ell)} \text{ for } \ell \geq 1, \quad (3.5)$$

$$\tilde{\mathbf{L}}^{(\ell)} = \mathbf{C}\mathbf{G}^{(\ell)}\mathbf{x}^{(\ell-1)} + \mathbf{L} + \mathbf{D}_{\text{ex}}\mathbf{L}_{\text{ex}}^{(\ell)} \text{ for } \ell \geq 1, \quad (3.6)$$

where

$$\mathbf{G}^{(\ell)} = \text{diag}\{\Phi_x(\mathbf{0}, \mathbf{B}_s \mathbf{L}_s^{(\ell)})\} \quad (3.7)$$

is a diagonal matrix, called the *gain matrix*, whose diagonal elements  $g_j^{(\ell)}$  are equal to the partial derivative of check node operation with respect to the state variables. Consider a missatisfied check node with the virtual variable node  $s_j$ , and with the two incoming messages represented by state variables  $x_i$  and  $x_{i'}$  (see, Fig. 3.1). For the SPA, we then have

$$g_i^{(\ell)} = g_{i'}^{(\ell)} = \tanh(L_{s_j}^{(\ell)}/2), \quad (3.8)$$

where  $L_{s_j}^{(\ell)}$  is the message entering the missatisfied check node through the virtual variable node  $s_j$  at iteration  $\ell$ . In [44, 45], the random gains  $g_i^{(\ell)}$  are replaced by their expected values,  $\bar{g}_i^{(\ell)} = \mathbb{E}\{\tanh(L_{s_i}^{(\ell)}/2)\}$ , and based on the assumption that all the external messages to missatisfied check nodes follow the same probability distribution, the average gain matrix is obtained as

$$\bar{\mathbf{G}}^{(\ell)} = \bar{g}^{(\ell)} \mathbf{I}_{m_s \times m_s}, \quad (3.9)$$

where  $\bar{g}^{(\ell)} = \bar{g}_i^{(\ell)}$ ,  $\forall i$ . As a result, in [44, 45], the effect of missatisfied check nodes is simply added to the model by a scalar gain for each iteration. Therefore, the linear state-space model of a LETS is given by [45]

$$\mathbf{x}^{(0)} = \mathbf{B}\mathbf{L} \quad (3.10)$$

$$\mathbf{x}^{(\ell)} = \bar{g}^{(\ell)} \mathbf{A}\mathbf{x}^{(\ell-1)} + \mathbf{B}\mathbf{L} + \mathbf{B}_{ex}\mathbf{L}_{ex}^{(\ell)}, \text{ for } \ell \geq 1, \quad (3.11)$$

$$\tilde{\mathbf{L}}^{(\ell)} = \bar{g}^{(\ell)} \mathbf{C}\mathbf{x}^{(\ell-1)} + \mathbf{L} + \mathbf{D}_{ex}\mathbf{L}_{ex}^{(\ell)}, \text{ for } \ell \geq 1. \quad (3.12)$$

In the above equations, the average gain value  $\bar{g}^{(j)}$  in the original linear model is replaced by  $\bar{g}^{(j)} = (1 - P_{inv,\ell})\bar{g}^{(j)}$ , where  $P_{inv,\ell}$  is the probability that the extrinsic message entering the missatisfied check node is in error (is negative) which results in polarity inversions of the LETS internal messages. This is used heuristically in [44, 45] to dampen the gain, and compensate for the over-estimation of the failure probability of an ETS as estimated through the linear model. In this thesis, we refer to the factor  $(1 - P_{inv,\ell})$  as *damping factor*.

### 3.1.2 Application of Density Evolution in the State-Space Model

In order to analyze TS failures based on the linear state-space model, one needs to obtain the probability distribution of the messages entering the TS subgraph via unsatisfied CNs at different iterations,  $\mathbf{L}_{ex}^{(\ell)}$ , as well as the distribution of the messages from the external VNs connected to the missatisfied CNs. The latter distribution is then used to obtain the linear gains  $\bar{g}'^{(\ell)}$ .

To obtain such distributions, the authors of [42, 44, 45] use density evolution (DE) [51] with the underlying assumptions that the all-zero codeword is transmitted, and that the surrounding neighborhood of the TS is tree-like. This implies that all the messages entering the TS through unsatisfied and missatisfied CNs are independent in each iteration as well as in subsequent iterations.

### 3.1.3 Error Probability of a LETS Structure

Before the calculation of error probability, we discuss some of the properties of the transition matrix  $\mathbf{A}$  relevant to the calculation.

For a given square matrix  $\mathbf{M}$  and a permutation matrix  $\mathbf{P}$ , the matrix  $\mathbf{PMP}^T$  is called the *symmetric permutation* of  $\mathbf{M}$ . If  $\mathbf{M}$  represents the adjacency matrix of a graph, symmetric permutation results in an isomorphic graph whose nodes are relabeled. The eigenvalues of  $\mathbf{M}$  and its symmetric permutation are the same. Also, their eigenvectors, up to a permutation, are equal [78].

**Definition 1.** A non-negative  $n \times n$  real matrix  $\mathbf{M}$  is *irreducible* if it cannot be symmetrically permuted by any permutation matrix  $\mathbf{P}$  into a block upper triangular matrix, i.e.,

$$\mathbf{PMP}^T \neq \left[ \begin{array}{c|c} \mathbf{M}_a & \mathbf{M}_c \\ \hline \mathbf{0} & \mathbf{M}_b \end{array} \right], \quad (3.13)$$

where  $\mathbf{M}_a$  and  $\mathbf{M}_b$  are square matrices with sizes greater than 0 (non-trivial). The matrix  $\mathbf{M}$  is *reducible* if it is not irreducible.

The *spectral radius* of a square matrix  $\mathbf{M}$  is defined as the largest absolute value of its eigenvalues, and is denoted by  $\rho(\mathbf{M})$ .

**Theorem 1** (Perron-Frobenius theorem of non-negative irreducible matrices). *Let  $\mathbf{M}$  be a non-negative irreducible matrix. Then,*

- (a)  $\mathbf{M}$  has a positive real eigenvalue equal to its spectral radius  $\rho(\mathbf{M})$ .
- (b) corresponding to  $\rho(\mathbf{M})$ , matrix  $\mathbf{M}$  has a positive eigenvector  $\mathbf{x}$ .<sup>1</sup>
- (c)  $\rho(\mathbf{M})$  is a simple (multiplicity equal to 1) eigenvalue of  $\mathbf{M}$ .
- (d)  $\mathbf{x}$  is the only non-negative eigenvector of  $\mathbf{M}$ .

Almost all the LETS structures, with the exception of simple cycles, have a non-negative irreducible transition matrix  $\mathbf{A}$  [45]. Hence, based on Theorem 1, there exists a positive dominant eigenvalue of  $\mathbf{A}$ ,  $r$ , whose corresponding left and right eigenvectors,  $\mathbf{w}_1^T$  and  $\mathbf{u}_1$ , are both positive. It is shown in [45] that for a simple cycle,  $r = 1$ , and for LETSs that are not simple cycles,  $r > 1$ .

To obtain the probability of a LETS failure, in [45], the authors first derived a non-recursive equation for the state vector:

$$\mathbf{x}^{(\ell)} = \mathbf{A}^\ell \mathbf{B}\mathbf{L} \prod_{j=1}^{\ell} \bar{g}'^{(j)} + \sum_{i=1}^{\ell} \mathbf{A}^{\ell-i} (\mathbf{B}\mathbf{L} + \mathbf{B}_{ex}\mathbf{L}_{ex}^{(i)}) \prod_{j=i+1}^{\ell} \bar{g}'^{(j)}. \quad (3.14)$$

They then showed that the projection of the state vector on the positive left eigenvector associated with the dominant eigenvalue  $r$ , given by

$$\mathbf{w}_1^T \mathbf{x}^{(\ell)} = r^\ell \mathbf{w}_1^T \mathbf{B}\mathbf{L} \prod_{j=1}^{\ell} \bar{g}'^{(j)} + \sum_{i=1}^{\ell} r^{\ell-i} \mathbf{w}_1^T (\mathbf{B}\mathbf{L} + \mathbf{B}_{ex}\mathbf{L}_{ex}^{(i)}) \prod_{j=i+1}^{\ell} \bar{g}'^{(j)}, \quad (3.15)$$

can be used as an indicator of TS failure. In fact, the following scaled version of (3.15) is used in [45] as the error indicator function:

$$\beta'_\ell = \mathbf{w}_1^T \mathbf{B}\mathbf{L} + \sum_{i=1}^{\ell} \frac{\mathbf{w}_1^T (\mathbf{B}\mathbf{L} + \mathbf{B}_{ex}\mathbf{L}_{ex}^{(i)})}{r^i \prod_{j=1}^i \bar{g}'^{(j)}}, \quad (3.16)$$

with the probability of error for the corresponding LETS  $\mathcal{S}$  given by

$$P_e(\mathcal{S}) = \lim_{\ell \rightarrow \infty} Pr\{\beta'_\ell < 0\} = \lim_{\ell \rightarrow \infty} Q\left(\frac{\mathbb{E}[\beta'_\ell]}{\sqrt{\text{VAR}[\beta'_\ell]}}\right), \quad (3.17)$$

---

<sup>1</sup>This is valid for both left and right eigenvectors.

where  $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-\frac{u^2}{2}) du$ , and  $\mathbb{E}(\beta'_\ell)$  and  $\mathbb{V}\text{AR}(\beta'_\ell)$  denote the expected value and the variance of  $\beta'_\ell$ , respectively, and calculated by

$$\begin{aligned} \mathbb{E}(\beta'_\ell) &= m_{in} \left( 1 + \sum_{i=1}^{\ell} \frac{1}{r^i \prod_{j=1}^i \bar{g}'^{(j)}} \right) \sum_{k=1}^a (\mathbf{w}_1^T \mathbf{B})_k \\ &+ \sum_{i=1}^{\ell} \frac{m_{ex}^{(i)}}{r^i \prod_{j=1}^i \bar{g}'^{(j)}} \sum_{k=1}^b (\mathbf{w}_1^T \mathbf{B}_{ex})_k, \end{aligned} \quad (3.18)$$

$$\begin{aligned} \mathbb{V}\text{AR}(\beta'_\ell) &= \sigma_{in}^2 \left( 1 + \sum_{i=1}^{\ell} \frac{1}{r^i \prod_{j=1}^i \bar{g}'^{(j)}} \right)^2 \sum_{k=1}^a (\mathbf{w}_1^T \mathbf{B})_k^2 \\ &+ \sum_{i=1}^{\ell} \frac{\sigma_{ex}^{2(i)}}{(r^i \prod_{j=1}^i \bar{g}'^{(j)})^2} \sum_{k=1}^b (\mathbf{w}_1^T \mathbf{B}_{ex})_k^2. \end{aligned} \quad (3.19)$$

In the above equations, the symbol  $(\cdot)_k$  represents the  $k$ -th element of a vector,  $m_{in} = 2/\sigma_{ch}^2$  and  $\sigma_{in}^2 = 4/\sigma_{ch}^2$  represent the mean and the variance of the channel LLR, respectively. Moreover,  $m_{ex}^{(i)}$  and  $\sigma_{ex}^{2(i)}$  are the mean and the variance of the messages passed from unsatisfied check nodes to variable nodes at iteration  $i$ .

In the following lemma, whose proof is given in the appendix (section 3.5), using perturbation theory, we consider the effect of the randomness of the gains on the error indicator function and the failure probability. This is performed by assuming that the random gain values have a small variance in the vicinity of their mean values.

**Lemma 2.** *The first order approximation of the maximum eigenvalue,  $\rho(\mathbf{A}\mathbf{G}^{(\ell)})$ , of matrix  $\mathbf{A}\mathbf{G}^{(\ell)}$  is given by*

$$\rho(\mathbf{A}\mathbf{G}^{(\ell)}) \approx r \mathbf{w}_1^T \bar{\mathbf{G}}^{(\ell)} \mathbf{u}_1 + \sum_{i=1}^{\frac{m_s}{2}} \left. \frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{\partial g_i^{(\ell)}} \right|_{\mathbf{G}^{(\ell)} = \bar{\mathbf{G}}^{(\ell)}} (g_i^{(\ell)} - \bar{g}_i^{(\ell)}), \quad (3.20)$$

in which

$$\left. \frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{\partial g_i^{(\ell)}} \right|_{\mathbf{G}^{(\ell)} = \bar{\mathbf{G}}^{(\ell)}} \approx \mathbf{w}_1^T \frac{\partial \mathbf{A}\mathbf{G}^{(\ell)}}{\partial g_i^{(\ell)}} \mathbf{u}_1, \quad (3.21)$$

and  $r$ ,  $\mathbf{w}_1$  and  $\mathbf{u}_1$  are the maximum eigenvalue and the corresponding normalized left and right eigenvectors of matrix  $\mathbf{A}$ .

**Remark 1.** The matrix  $\frac{\partial(\mathbf{A}\mathbf{G}^{(\ell)})}{\partial g_i^{(\ell)}}$  is binary with 0 and 1 elements. It is also known that the transition matrix  $\mathbf{A}$  is non-negative and irreducible for the majority of ETSs [45]. For such matrices, based on Perron-Frobenius theorem [81, p. 534], all the elements of the left and right eigenvectors are positive. We thus conclude that the derivative in (3.21) is always positive.

Based on Lemma 2, we define the following random scalar gain at iteration  $\ell$ .

**Definition 2.** The random scalar gain at iteration  $\ell$  is defined as:

$$\mathbf{g}^{(\ell)} = \frac{\rho(\mathbf{A}\mathbf{G}^{(\ell)})}{r} \approx \bar{\mathbf{g}}^{(\ell)} + \sum_{i=1}^{\frac{m_s}{2}} \frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{r \partial g_i^{(\ell)}} (g_i^{(\ell)} - \bar{g}_i^{(\ell)}), \quad (3.22)$$

where  $\bar{\mathbf{g}}^{(\ell)} = \mathbf{w}_1^T \bar{\mathbf{G}}^{(\ell)} \mathbf{u}_1$  is the mean of  $\mathbf{g}^{(\ell)}$ . Also, assuming gains  $g_i^{(\ell)}$  are independent, the variance of  $\mathbf{g}^{(\ell)}$  at a given iteration  $\ell$  can be approximated as

$$\sigma_{\mathbf{g}^{(\ell)}}^2 \approx \sum_{i=1}^{\frac{m_s}{2}} \left( \frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{r \partial g_i^{(\ell)}} \right)^2 \sigma_{g_i^{(\ell)}}^2. \quad (3.23)$$

**Remark 2.** When the distribution of the extrinsic messages of the missatisfied check nodes are identical, since  $\mathbf{w}_1^T \mathbf{u}_1 = 1$ , the mean value  $\bar{\mathbf{g}}^{(\ell)} = \mathbf{w}_1^T \bar{\mathbf{G}}^{(\ell)} \mathbf{u}_1$  is equal to the expected gain  $\bar{g}^{(\ell)}$  in Equation (3.9).

The following lemma, whose proof is given in the appendix (section 3.5), provides the error indicator function of an ETS in the presence of random gains.

**Lemma 3. (Error Indicator Function)** Consider State Evolution Equations (8)-(10) for a given ETS  $\mathcal{S}$ , with the assumption that gains  $g_i^{(\ell)}$  are random but with small differences in their values. The error indicator function  $\xi_\ell$  at iteration  $\ell$  ( $\ell \geq 1$ ) for  $\mathcal{S}$ , defined as the projection of  $\mathbf{x}^{(\ell)}$  onto the dominant left eigenvector of  $\mathbf{A}\mathbf{G}^{(\ell)}$ , can then be approximated as:

$$\xi_\ell \approx r^\ell \mathbf{w}_1^T \mathbf{B}\mathbf{L} \prod_{j=1}^{\ell} \mathbf{g}^{(j)} + \sum_{i=1}^{\ell} r^{\ell-i} \mathbf{w}_1^T (\mathbf{B}\mathbf{L} + \mathbf{B}_{ex} L_{ex}^{(i)}) \prod_{j=i+1}^{\ell} \mathbf{g}^{(j)}, \quad (3.24)$$

where  $\mathbf{w}_1^T$  and  $r$  are the dominant left eigenvector of  $\mathbf{A}$  and the corresponding eigenvalue, respectively, and  $\mathbf{g}^{(j)}$  is the random scalar gain at iteration  $j$  defined in Definition 2.

Equation (3.24) is similar to (3.15) with the exception that the expected gains are replaced with random gains defined in Definition 2. The derivation of the mean and the variance of the error indicator function from (3.24) is rather complex. In the following lemma, we calculate them recursively. The proof is straightforward but tedious and is not presented here.

**Lemma 4.** *The mean and the variance of the error indicator function, given in Lemma 3 (with  $\xi_0 \triangleq \mathbf{w}_1^T \mathbf{B}L$ ), can be calculated as follows:*

$$\begin{aligned} \underline{\ell = 0} \\ \mathbb{E}\{\xi_0\} &= m_{in} \sum_{k=1}^a (\mathbf{w}_1^T \mathbf{B})_k, \quad \mathbb{V}\text{AR}\{\xi_0\} = \sigma_{in}^2 \sum_{k=1}^a (\mathbf{w}_1^T \mathbf{B})_k^2. \\ \underline{\ell \geq 1} \\ \mathbb{E}\{\xi_\ell\} &= \bar{g}^{(\ell)} r \mathbb{E}\{\xi_{\ell-1}\} + \mathbb{E}\{\xi_0\} + m_{ex}^{(\ell)} \sum_{k=1}^b (\mathbf{w}_1^T \mathbf{B}_{ex})_k, \\ \mathbb{V}\text{AR}\{\xi_\ell\} &= r^2 \mathbb{V}\text{AR}\{g^{(\ell)} \xi_{\ell-1}\} + \mathbb{V}\text{AR}\{\xi_0\} \\ &\quad + \sigma_{ex}^{2(\ell)} \sum_{k=1}^b (\mathbf{w}_1^T \mathbf{B}_{ex})_k^2 + 2r \text{COV}\{g^{(\ell)} \xi_{\ell-1}, \xi_0\}, \end{aligned}$$

in which

$$\mathbb{V}\text{AR}\{g^{(\ell)} \xi_{\ell-1}\} = \bar{g}^{(\ell)2} \mathbb{V}\text{AR}\{\xi_{\ell-1}\} + \sigma_{g^{(\ell)}}^2 \left( \mathbb{V}\text{AR}\{\xi_{\ell-1}\} + \mathbb{E}^2\{\xi_{\ell-1}\} \right), \quad (3.25)$$

$$\text{COV}\{g^{(\ell)} \xi_{\ell-1}, \xi_0\} = \left( \sum_{j=1}^{\ell} \left( \prod_{k=j}^{\ell} \bar{g}^{(k)} \right) r^{\ell-j} \right) \mathbb{V}\text{AR}\{\xi_0\}. \quad (3.26)$$

The probability of failure of the ETS  $\mathcal{S}$  can then be found by

$$P_e\{\mathcal{S}\} = \lim_{\ell \rightarrow \infty} Q\left( \frac{\mathbb{E}\{\xi_\ell\}}{\sqrt{\mathbb{V}\text{AR}\{\xi_\ell\}}} \right). \quad (3.27)$$

**Remark 3.** *We note that the error indicator function  $\beta'_\ell$  defined in [45] is a scaled version of the error indicator function  $\xi_\ell$  defined here. Scaling back the mean of  $\beta'_\ell$ , given in (3.18), along with the assumption that  $\bar{g}'^{(j)} = \bar{g}^{(j)}, \forall j$ , one can see that the mean value is identical to that obtained through Lemma 4. One can also see that if the gain variances  $\sigma_{g^{(\ell)}}^2$ , in (3.25), are replaced with zero, then the variance of  $\xi_\ell$  obtained from Lemma 4 is identical to that in (3.19) (with the proper scaling and the assumption that  $\bar{g}'^{(j)} = \bar{g}^{(j)}, \forall j$ ). This means the failure probability (3.27) reduces to*

(3.17) if  $\sigma_{g^{(\ell)}}^2 = 0$ . On the other hand, if  $\sigma_{g^{(\ell)}}^2 \neq 0$ , then  $\text{VAR}\{\xi_\ell\}$  is increased and as a result the failure probability of the ETS in (3.27) is also increased (compared to the case of  $\sigma_{g^{(\ell)}}^2 = 0$ ).

## 3.2 Estimation Error of the Linear State-Space Model

Remark 3 shows that having a more realistic linear model of an ETS, which takes into account the randomness of the linear gains associated with the missatisfied check nodes, further deteriorates the accuracy of the error probability estimate by even more over-estimation of the probability. We note that the linear model (for saturating decoders) with average gains  $\bar{g}^{(j)}$  already over-estimates the error probability, and even with the heuristic damping of these gains to  $\bar{g}'^{(j)}$ , as proposed in [45], still in some cases the error probability is over-estimated. This suggests that the causes of the inaccuracy of the linear model are the followings: (1) the linear approximation (around zero) of the missatisfied check node operations, and (2) the dependencies that exist among the extrinsic messages entering the ETS through the check nodes. In [45], the effect of correlation among the extrinsic messages were discussed and it was asserted that such an effect in the error floor estimation is negligible in saturating decoders. Our experiments however, show that (in an accurate message-passing over the subgraph of an ETS) while the correlation among the extrinsic messages of unsatisfied check nodes can be ignored, the correlation among the extrinsic messages coming through the missatisfied check nodes needs to be taken into account for an accurate estimate of the error probability.

In the following theorem, we study the effect of correlation among the extrinsic messages entering the missatisfied check nodes on the error floor estimation by the linear state-space model, assuming that such correlations are non-negative. Our extensive simulations for a variety of ETS structures, codes and decoders verify this assumption.

**Theorem 2.** *In the linear state-space model, let  $\kappa_{ij}$  be the correlation coefficient between gains  $g_i^{(\ell)}$  and  $g_j^{(\ell)}$ . The failure rate of an ETS, given by (3.27), is always larger if  $\kappa_{ij} \geq 0$ , for every choice of  $i \neq j$ , compared to the case where the gains are independent. In addition, the failure rate increases by the increase in correlation values.*

*Proof.* In the linear state-space model, if the gains are independent, the gain variance at iteration  $\ell$  can be obtained by (3.23). If the correlation among the gains are considered, the gain variance changes to

$$\begin{aligned} \sigma_{\mathbf{g}^{(\ell)}}^2 &= \sum_{i=1}^{\frac{m_s}{2}} \left( \frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{r \partial g_i^{(\ell)}} \right)^2 \sigma_{g_i^{(\ell)}}^2 \\ &+ \sum_{i,j=1, i \neq j}^{\frac{m_s}{2}} \kappa_{ij} \left( \frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{r \partial g_i^{(\ell)}} \right) \left( \frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{r \partial g_j^{(\ell)}} \right) \sigma_{g_i^{(\ell)}} \sigma_{g_j^{(\ell)}}. \end{aligned} \quad (3.28)$$

Based on Lemma 4 (and Remark 3), the variance of the error indicator function, and as a result the error probability, increase by the increase in the gain variance  $\sigma_{\mathbf{g}^{(\ell)}}^2$ . The gain variance in (3.28) is always greater than or equal to that in (3.23), since the second term in (3.28) is always non-negative. This is due to  $\kappa_{ij} \geq 0$ , and the fact that the partial derivatives involved in the second term are positive (see Remark 1). ■

Theorem 2 suggests that the linear state-space model with correlated random gains is even less accurate in estimating the error probability and would significantly over-estimate the error floor. This leaves us to believe that the linear approximation of the missatisfied check nodes around zero is the main culprit in the estimation error. Note that there are two sources of error involving the linear approximation. One is to ignore the non-linearity, and the other is the choice of  $\mathbf{x}^{(\ell)} = \mathbf{0}$  as the operating point.

Interestingly, our simulations show that the accurate implementation of non-linearity results in the under-estimation of the error probability. It is only by including a correlation model for the extrinsic messages of missatisfied check nodes (in addition to the accurate implementation of non-linearity) that we can accurately estimate the error floor.

### 3.3 Proposed Method

In this section, we propose a code-independent technique to estimate the error probability of ETS structures. The proposed technique not only provides a more accurate estimate compared to the method of [45] for floating-point SPA (with saturation) over the AWGN channel, but also is applicable to quantized implementations of SPA

and other symmetric iterative decoding algorithms over any memoryless binary-input output-symmetric channel.

### 3.3.1 Local importance sampling (LIS)

The main idea behind the proposed technique is to accurately implement the non-linear operation of missatisfied check nodes. For this, the accurate message-passing (based on the selected iterative decoding algorithm and quantization scheme) is performed over the subgraph of the ETS induced in the code's Tanner graph. To speed up this process, we use what we call *local importance sampling (LIS)*. To apply LIS, we perform  $N$  trials of the iterative decoding (with the biased distribution) over the local subgraph induced by the ETS rather than over the entire Tanner graph. In IS, special care should be taken in the selection of the amount of biasing, see, e.g., [84, pp. 340–348]. An amount of bias which is too small can lead to rare decoding failures, slowing down the IS process. This is referred to as *underbiasing*. On the other hand, choosing the bias too large, referred to as *overbiasing*, may cause estimation errors. In fact, proper selection of the amount of bias is described as more of an art than an exact science in [84, p. 347]. Our experiments show that for the LIS method used in this work, selecting the amount of bias to result in 10% to 50% failure rate would be suitable, in the sense that it significantly speeds up the IS process without sacrificing the estimation accuracy.

To implement LIS, we also need to properly model and generate the messages that are exchanged between the rest of the Tanner graph and the TS. For this, we initially assume that the surroundings of a TS in the Tanner graph are trees, and that the messages passed from the subgraph to the rest of the Tanner graph have negligible effect on the performance of iterative decoding in the rest of the graph. (These assumptions are similar to those made in [45].) As a result, for memoryless channels, all the incoming messages to the TS from the unsatisfied and missatisfied check nodes in different iterations are independent. The distribution of these messages in different iterations can be obtained using DE [51, 85]. (We note that such distributions are only a function of the degree distributions of the Tanner graph.) For the messages from unsatisfied check nodes at each iteration  $\ell$ , we use the probability distribution function  $P_{c2v}^\ell$  of check to variable node messages. For the incoming messages to a missatisfied check node of degree  $d_{c_j}$ , to simplify the calculations, we find the distribution  $P_{L_{s_j}}^\ell$  of random variable  $L_{s_j}$  in Fig. 3.1. This random variable represents the check to

variable node messages at iteration  $\ell$  (in the DE framework) assuming that the check node degree is  $d_{c_j} - 1$ . Having the distributions  $P_{c_{2v}}^\ell$  and  $P_{L_{s_j}}^\ell$  at each iteration  $\ell$  through DE, we then sample these distributions in an independent and identically distributed (i.i.d.) fashion to obtain the actual messages for the iterative message-passing algorithm over the subgraph for different trials of the LIS simulations. To use (2.10) for estimating the contribution of a TS in the error floor, we consider an error at trial  $z$ , i.e.,  $I_z = 1$ , if all the variable nodes in the TS are in error.<sup>2</sup>

Finally, we note that the application of the proposed LIS method is limited to saturating decoders. The reason is that, for a non-saturating decoder, unlike a saturating decoder, there would be a strong correlation among the incoming messages to an ETS through unsatisfied check nodes in higher iterations [45]. This is not taken into account in our model. Moreover, in our model, we assume that the erroneous messages that are passed via unsatisfied check nodes to the rest of the Tanner graph do not have any impact on the performance of the iterative decoding algorithm in the rest of the graph. This assumption is justified by the fact that the magnitude of the erroneous messages in saturating decoders is typically small compared to the size of messages in the rest of the Tanner graph. This, however, will not be necessarily the case for non-saturating decoders. In non-saturating decoders, the erroneous outgoing messages from the ETS can affect the messages in the rest of the graph and subsequently the incoming messages to the ETS from the rest of the graph.

### 3.3.2 Correlation model

The application of our proposed method, described in Subsection 3.3.1, would result in the under-estimation of the error floor, particularly for cases where the clipping threshold of the decoder is relatively large. Extensive simulations show that the main reason for such an under-estimation is that there are positive correlations among the incoming messages to missatisfied check nodes after the iterations go beyond  $g/2$ , where  $g$  is the girth of the Tanner graph, and that such correlations are neglected in the model. The simulations also show that, on average, the correlation among these messages increases very rapidly towards one in the first few iterations following  $\ell = g/2$ . The average correlation tending to one corresponds to the incoming messages

---

<sup>2</sup>This is for the cases where TSs are fixed points of the iterative decoding algorithm. For the cases where the TSs correspond to oscillatory error patterns, the oscillatory cases are also identified as error and assigned  $I_z = 1$ .

being saturated at their maximum value (all the variables outside the TS converging to their correct value, i.e., zero). We also observe that the behavior just described, as well as the amount of correlation, is rather independent of the amount of bias imposed on the noise distribution.

The exact amount of correlation, however, depends on the clipping threshold, the code, the decoding algorithm and the structure of the graph surrounding the ETS. Such dependencies are in general too complex to investigate or to incorporate in the proposed model. In our method, we thus adopt a very simple model that would assume that the incoming messages to missatisfied check nodes are i.i.d. for iterations  $\ell < g/2$ , and then are identical for  $\ell \geq g/2$ . For the LIS process, this means that in the iterative message-passing over the TS subgraph, while  $\ell < g/2$ , we draw i.i.d. samples from  $P_{L_{s_j}}^\ell$  to generate the messages that represent the incoming messages of different missatisfied check nodes in each iteration. For  $\ell \geq g/2$ , however, in each iteration  $\ell$ , we just draw one sample from  $P_{L_{s_j}}^\ell$  and use it for all the missatisfied check nodes in the subgraph. This model assumes that the amount of correlation coefficient between any two copies of  $L_{s_j}$  variables within the subgraph is equal to one for cases that  $\sigma_{L_{s_j}} \neq 0$ . In the next section, we will demonstrate that this simple model renders an accurate estimate of the error floor for a wide variety of codes, decoders and quantization schemes.

### 3.3.3 Complexity of LIS scheme

Consider a case where the LIS scheme is applied for a given channel parameter  $\eta$  in the error floor region to estimate the error probability of an LDPC code. Suppose that for each dominant ETS structure  $\mathcal{S}$ ,  $N$  trials of the iterative decoding (with the biased distribution) over the local subgraph induced by  $\mathcal{S}$  are performed to estimate the contribution of  $\mathcal{S}$  to the error floor. Our experiments show that for an accurate estimate,  $N$  should be selected such that in the LIS process at least 500 failures occur. In Section 3.4, we obtain about 1000 failures for all estimates. This means that if the amount of distribution bias is selected to result in 10% failure rate, then  $N \approx 10,000$ .

To apply the LIS algorithm at a given channel parameter  $\eta$ , one needs to find the distributions  $P_{c_{2v}}^\ell$  and  $P_{L_{s_j}}^\ell$  for each iteration  $\ell \leq I_{\max}$  through DE (based on  $\xi$ ), where  $I_{\max}$  is the maximum number of iterations. For a given  $\eta$ , this needs to be done only once. On the other hand, the set of  $N$  trials of the iterative decoding over the ETS subgraph needs to be repeated for each non-isomorphic ETS structure.

In general, the complexity of the LIS process is dominated by that of the iterative message-passing over the ETS subgraphs.

Now, let  $\mathcal{S}$  be an  $(a, b)$  ETS in a variable-regular LDPC code with variable node degree  $d_v$ . In each iteration, the local message-passing algorithm needs to generate and pass two sets of messages on the edges of the subgraph, one set from variable nodes to missatisfied check nodes and the other set in the opposite direction. The number of messages in each direction is thus equal to the number of edges in the subgraph, i.e.,  $a \times d_v - b$ . In comparison, each iteration of message-passing over the entire Tanner graph requires  $n \times d_v$  messages to be generated and passed in each direction between variable nodes and check nodes, where  $n$  is the code's block length. Thus, the complexity per iteration of message-passing over the subgraph is less than that of the entire Tanner graph by a factor of about  $n/a$ , which is often a large number. The same complexity reduction is achieved by the LIS technique compared to the conventional IS technique if one assumes the same number of trials  $N$  for both cases. It should be noted that in addition to having a lower complexity compared to the conventional IS, LIS has the advantage of being code-independent.

## 3.4 Simulation Results of the Proposed Technique and Comparisons with the Results of Linear State-Space Model

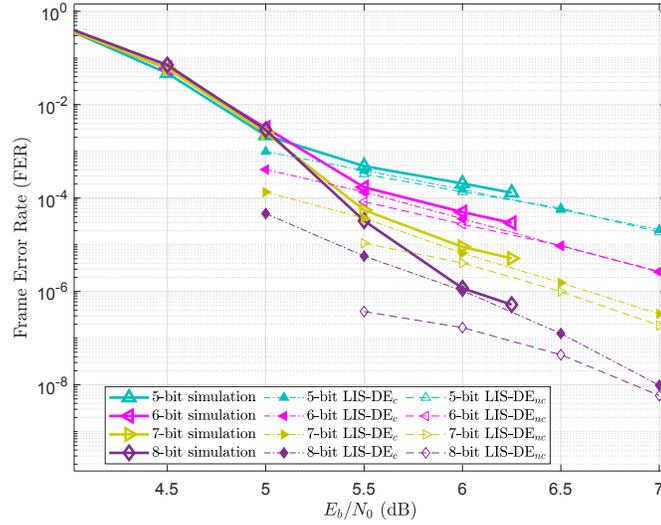
### 3.4.1 Simulation results of the proposed technique for quantized decoders

To examine the accuracy of our proposed method, we have applied it to a wide range of LDPC codes, both random and structured, decoded by different decoding algorithms and for a variety of quantization parameters over both AWGN and BSCs. Here, we report some of our results for the following seven LDPC codes: (640, 192) QC code ( $\mathcal{C}_1$ ) [3], Margulis (2640, 1320) code ( $\mathcal{C}_2$ ) [86], random variable-regular (4095, 3358) code ( $\mathcal{C}_3$ ) [87], Wimax irregular (576, 432) LDPC code ( $\mathcal{C}_4$ ) [82], the Reed-Solomon-based binary (2048, 1723) LDPC code ( $\mathcal{C}_5$ ) [88] included in the IEEE 802.3an standard, and two LDPC codes constructed based on finite geometries: a (63, 37) EG-LDPC code ( $\mathcal{C}_6$ ) [89, 90], and a (273, 191) PG-LDPC code ( $\mathcal{C}_7$ ) [89]. The biasing of

the noise distribution is selected such that the failure rate of each ETS is about 10%. For each LIS simulation point, around 1000 TS failures are collected. In the following, we use notations LIS-DE<sub>c</sub> for the proposed algorithm with the correlation model, and LIS-DE<sub>nc</sub> for the proposed algorithm when the extrinsic messages to missatisfied check nodes are selected independently in every iteration (no correlation).

Fig. 3.3 shows the Monte Carlo simulation results along with LIS-DE<sub>c</sub> and LIS-DE<sub>nc</sub> estimation results for  $\mathcal{C}_1$  decoded by quantized MSA with  $q = 5, 6, 7, 8$ , and  $\Delta = 0.5$ , over the AWGN channel, where the decoder input is the channel LLR,  $2y/\sigma_{ch}^2$ . This code has  $d_v = 5$  and irregular check node degrees. In our method, the irregularity of the check node degrees is taken into account by applying a weighted average, according to the degree distribution of the code, in DE. It is known that the most harmful substructure of this code is a (5, 5) ETS with multiplicity 64. We thus apply the proposed algorithm to this ETS structure and then multiply the result obtained through (2.10) by 64. The results of Fig. 3.3 demonstrate a good match between the LIS-DE<sub>c</sub> estimation and simulation results. The LIS-DE<sub>nc</sub> estimation results however, are lower than the simulation curves, in particular, for higher saturation levels. For  $\mathcal{C}_1$ , in Fig. 3.4, we have also reported simulation and LIS-DE<sub>c</sub> estimation results of a 4-bit MS decoder with two resolutions of  $\Delta = 0.5$  and  $\Delta = 1$ , over the BSC, where the decoder input values are set to 1 and -1 for the positive and negative received symbols, respectively. Again, the examination of these results shows a good match between simulations and estimations.

The results for  $\mathcal{C}_2$  over the AWGN channel are presented in Fig. 3.5, where the decoder input is selected to be the channel output  $y$ , and the decoding algorithm is chosen to be a 5-bit normalized MSA with normalization factor  $\alpha = 0.75$ . It is known that the most harmful substructures of  $\mathcal{C}_2$  are (12, 4) and (14, 4) ETSs, each with multiplicity 1320 [34, 86]. For the case considered in this example, and corresponding to both of these ETS structures, there are both fixed-point and oscillatory failures. The contribution of each failure type for each ETS structure is calculated using the LIS-DE<sub>c</sub> method, and is presented separately in Fig. 3.5, along with the total error floor estimation, obtained by adding up the four different contributions. As can be seen, the total estimate matches well with Monte Carlo simulation results. For comparison, the estimation results of LIS-DE<sub>nc</sub> are also given in Fig. 3.5. Again, the LIS-DE<sub>nc</sub> results under-estimate the error floor. Moreover, in Fig. 3.5, we have provided the LIS-DE<sub>c</sub> and LIS-DE<sub>nc</sub> estimates along with simulations of FER for a



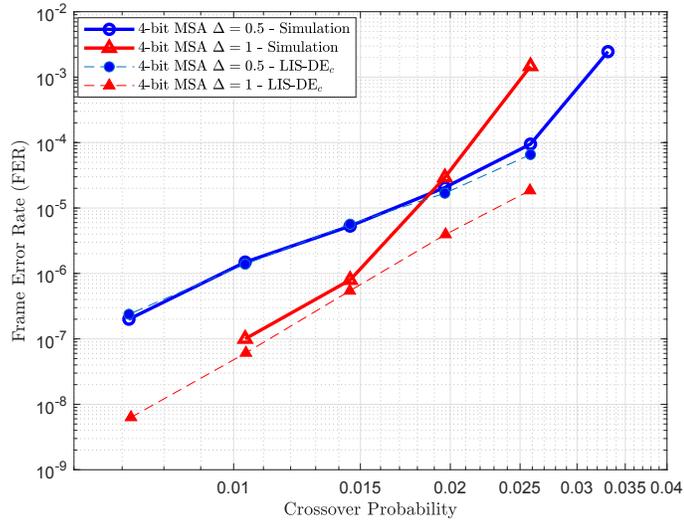
**Figure 3.3:** Simulation and estimation results for  $\mathcal{C}_1$  decoded by quantized MSA over the AWGN channel ( $\Delta = 0.5$ ,  $I_{max} = 200$ , and decoder input =  $2y/\sigma_{ch}^2$ ).

**Table 3.1:** Multiplicities of  $(a, b)$  LETSs of  $\mathcal{C}_3$  with  $a \leq 8$  and  $b \leq 3$

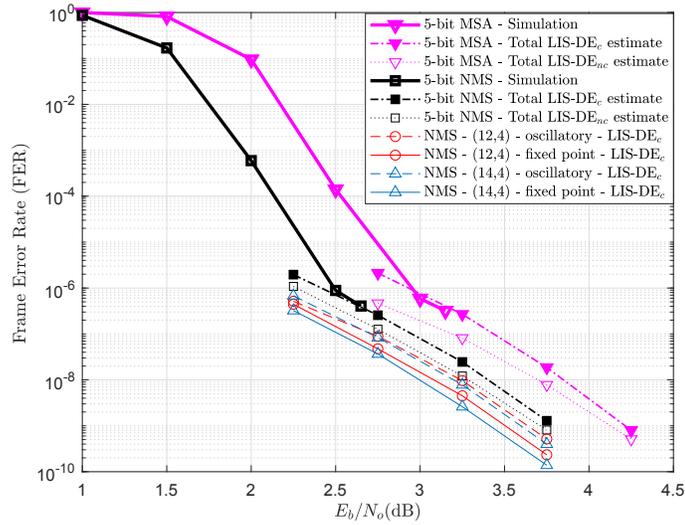
$(a, b)$	(3, 3)	(4, 2)	(5, 1)	(5, 3)	(6, 2)	(7, 1)	(7, 3)	(8, 2)
Multiplicity	5183	308	15	22876	1722	82	184868	13550

5-bit MSA in Fig. 3.5. Similar trends are also observed in this case with even a larger under-estimation of the error floor by LIS-DE<sub>nc</sub>.

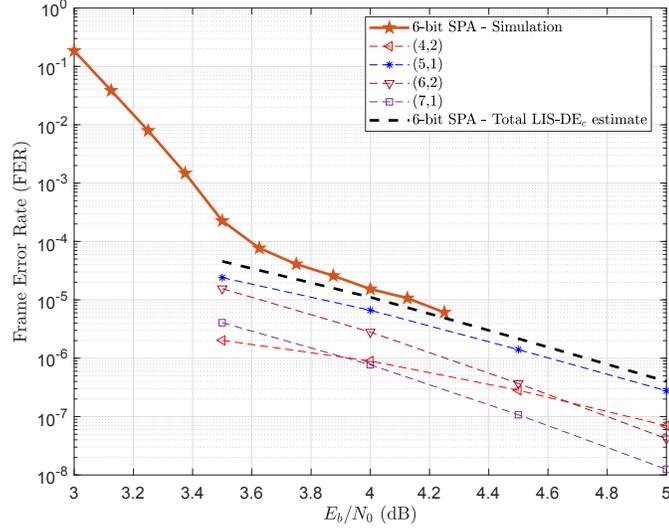
In Fig. 3.6, we have presented the results for  $\mathcal{C}_3$  decoded by a 6-bit SPA over the AWGN channel. An exhaustive list of leafless ETSs (LETs) [32] of  $\mathcal{C}_3$  in the range  $a \leq 8, b \leq 3$ , is given in Table 3.1. We use this list to estimate the error floor. For each class, we identify TSs with different non-isomorphic structures. For each such structure, we use LIS-DE<sub>c</sub> to estimate their contribution using (2.10), and then multiply the result by the multiplicity of that structure. The final estimate is obtained by adding up all contributions. Our results show that for this example, the most dominant TSs are those in classes (5, 1), (7, 1), (4, 2) and (6, 2), with (5, 1) having the most contribution to the error floor. Fig. 3.6 shows a good match between the estimation and simulation results. It also reveals that the relative harmfulness of different classes changes as a function of channel SNR. In fact, if we decrease the



**Figure 3.4:** Simulation and estimation results for  $\mathcal{C}_1$  decoded by 4-bit MSA over the BSC ( $I_{max} = 200$ , decoder input values 1 and  $-1$ ).



**Figure 3.5:** Simulation and estimation results for  $\mathcal{C}_2$  decoded by 5-bit MSA and 5-bit normalized MSA over the AWGN channel ( $C_{th} = 3$ ,  $I_{max} = 100$ ,  $\alpha = 0.75$ , decoder input =  $y$ ).



**Figure 3.6:** Simulation and estimation results for  $\mathcal{C}_3$  decoded by a 6-bit SPA over the AWGN channel ( $\Delta = 0.5$ ,  $I_{max} = 200$ , decoder input =  $2y/\sigma_{ch}^2$ ).

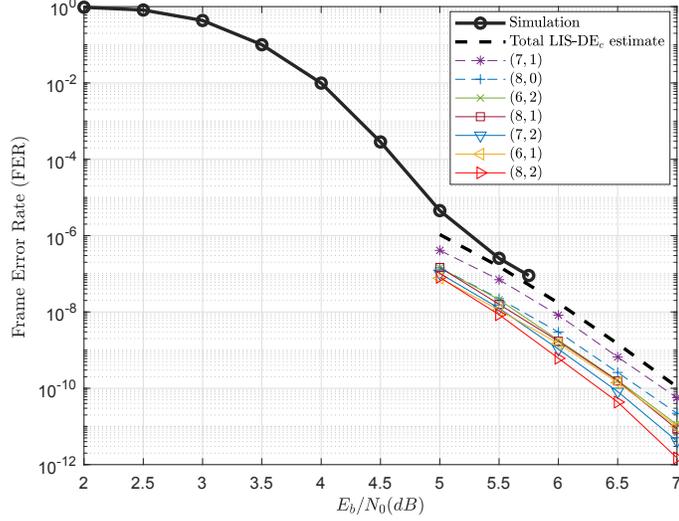
**Table 3.2:** Multiplicities of the  $(a, b)$  LETSs of  $\mathcal{C}_4$  within the range  $a \leq 8$  and  $b \leq 2$

$(a, b)$	(4, 2)	(5, 2)	(6, 1)	(6, 2)	(7, 1)	(7, 2)	(8, 0)	(8, 1)	(8, 2)
Multiplicity	144	216	48	1068	240	3600	48	720	13464

number of quantization bits to 5, the most dominant class also changes from (5, 1) to (4, 2).

As another example, we consider  $\mathcal{C}_4$  decoded by a 3-bit MS decoder with  $C_{th} = 1.8$  and input  $y$ . An exhaustive list of the LETSs of  $\mathcal{C}_4$  in the range of  $a \leq 8$  and  $b \leq 2$  is given in Table 4.1. In Fig. 3.7, we have presented the LIS- $DE_c$  estimation results for the dominant classes of ETSs along with the total error floor estimation, and Monte Carlo simulation results. As can be seen, the estimation and simulation results are in good agreement. This demonstrates the accuracy of the proposed estimation technique even for irregular codes.

For  $\mathcal{C}_5$ , we consider a 5-bit normalized MSA with  $\alpha = 0.75$  and maximum number of iterations  $I_{max} = 50$ . The quantization scheme and the decoding algorithm are selected to match those used in [12] for the same code. This code is bi-regular with  $d_v = 6$  and  $d_c = 32$ , and has a rate 0.84. The dominant TSs of  $\mathcal{C}_5$  are known to be



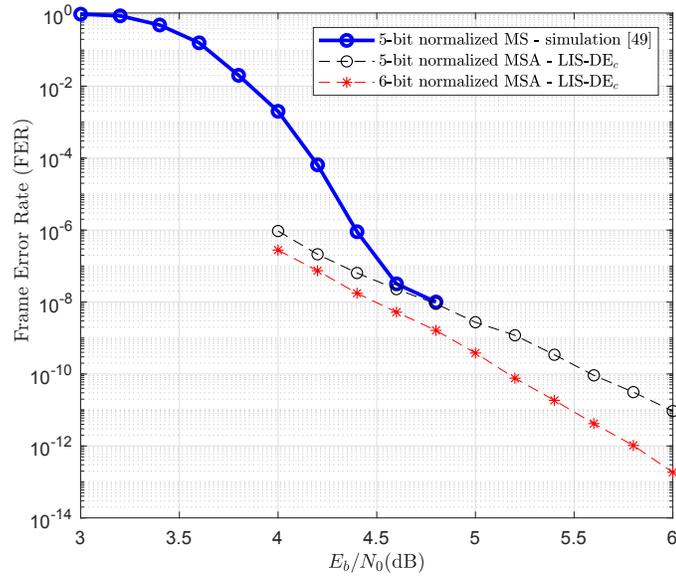
**Figure 3.7:** Simulation and estimation results for  $\mathcal{C}_4$  decoded by a 3-bit MSA ( $C_{th} = 1.8$ ,  $I_{max} = 100$  and decoder input =  $y$ ).

(8, 8) ETSs with multiplicity 14272 [44]. In Fig. 3.8, we have presented the LIS-DE<sub>c</sub> estimation results based on (8, 8) ETSs along with Monte Carlo simulation results extracted from [12]. As can be seen, there is a good match between the two sets of results. For comparison, in the same figure, we have also included the estimation results of a 6-bit normalized MSA.

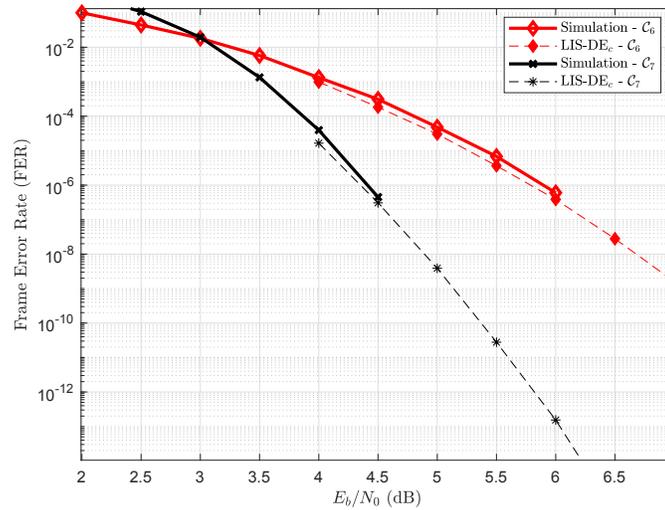
The finite-geometry codes  $\mathcal{C}_6$  and  $\mathcal{C}_7$  have rates 0.587 and 0.700, respectively. Both variable nodes and check nodes of  $\mathcal{C}_6$  ( $\mathcal{C}_7$ ) have the same degree 8 (17). The dominant trapping sets of both codes are the minimum-weight codewords [91]. Such codewords for  $\mathcal{C}_6$  and  $\mathcal{C}_7$  have weights 9 and 18 with multiplicities 1960 and 119, 857, 920, respectively [92]. In Fig. 3.9, we have provided the LIS-DE<sub>c</sub> estimate for both codes based on the minimum-weight codewords, where the decoding algorithm is selected to be a 7-bit normalized MS with  $\alpha = 0.6$  and maximum number of iterations  $I_{max} = 100$ . As can be seen, for both codes, the estimates match well with Monte Carlo simulations.

Finally, as the last example, we demonstrate that the proposed technique can also be used to estimate the error floor for a non-uniform quantization. We consider the application of  $\mathcal{C}_2$  over the binary-input AWGN channel, where the channel output is uniformly quantized by 6 bits ( $C_{th} = 3$ ). We then use the algorithm of [93],<sup>3</sup> which

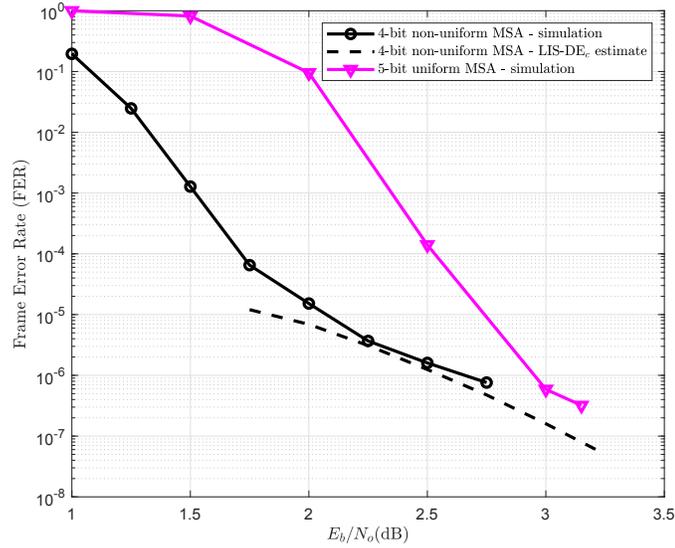
<sup>3</sup>We use the algorithm available online at URL: <http://www.jaist.ac.jp/is/labs/bits/source>.



**Figure 3.8:** Simulation and estimation results of (2048, 1723) RS-LDPC code ( $\mathcal{C}_5$ ) decoded by a 5-bit normalized MSA ( $\alpha = 0.75$ ,  $\Delta = 0.5$ ,  $I_{max} = 50$  and decoder input =  $2y/\sigma_{ch}^2$ ).

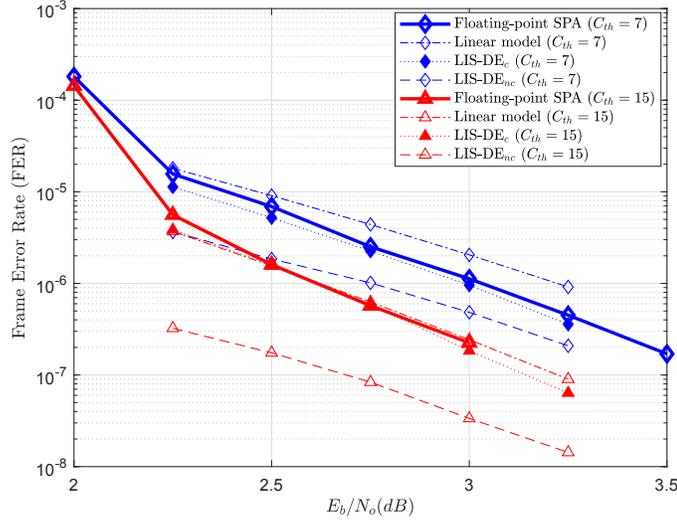


**Figure 3.9:** Simulation and estimation results for  $\mathcal{C}_6$  and  $\mathcal{C}_7$  decoded by a 7-bit normalized MSA ( $C_{th} = 6$ ,  $\alpha = 0.6$ ,  $I_{max} = 100$  and decoder input =  $y$ ).



**Figure 3.10:** Simulation and estimation results of Margulis (2048, 1024) code ( $\mathcal{C}_2$ ) decoded by 4-bit MSA with non-uniform quantization ( $C_{th} = 3$ ,  $I_{max} = 40$ , and decoder input =  $y$ ).

is based on maximizing the mutual information between the channel input and the quantizer output, to reduce the number of quantization bits to 4 based on a non-uniform quantization. We use MSA for decoding, and apply the algorithm of [93] in the process, to reduce the number of bits for the output messages of variable nodes to 4 bits. The quantization boundaries can thus be changed from one iteration to the next depending on the message distributions, obtained using DE. The non-uniform quantization is optimized for  $E_b/N_0 = 2.25$  dB and the resulted quantization scheme is used for all SNR values. (We note that the non-uniform quantization of [93] can be, in general, asymmetric. In our example, however, all the quantization boundaries are approximately symmetric.) Fig. 3.10 shows the Monte Carlo simulation results along with the estimation results of LIS-DE<sub>c</sub>, obtained based on (12, 4) ETSs. As can be seen, there is a good match between the two sets of results. For comparison, we have also included simulation results for a 5-bit uniformly quantized MSA. One can see the large improvement obtained by the non-uniform quantization in the waterfall region, although the uniform quantization is using one extra bit.

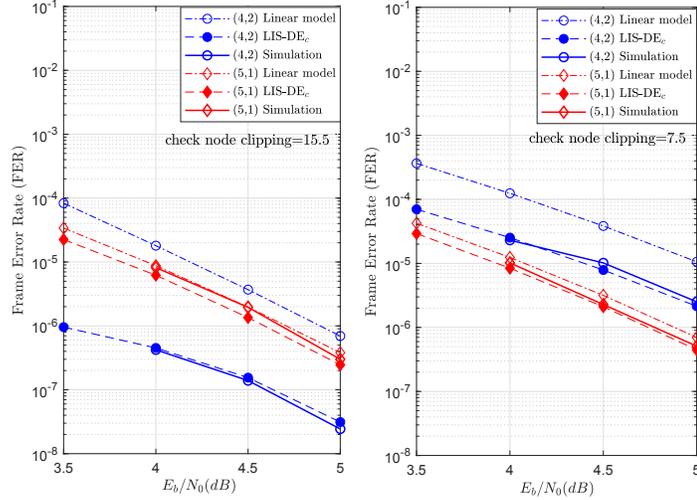


**Figure 3.11:** Simulation and estimation results for  $\mathcal{C}_2$  decoded by floating-point SPA with clipping at the check node outputs ( $I_{max} = 200$ ).

### 3.4.2 Comparison of the LIS-DE<sub>c</sub> method with linear state-space model

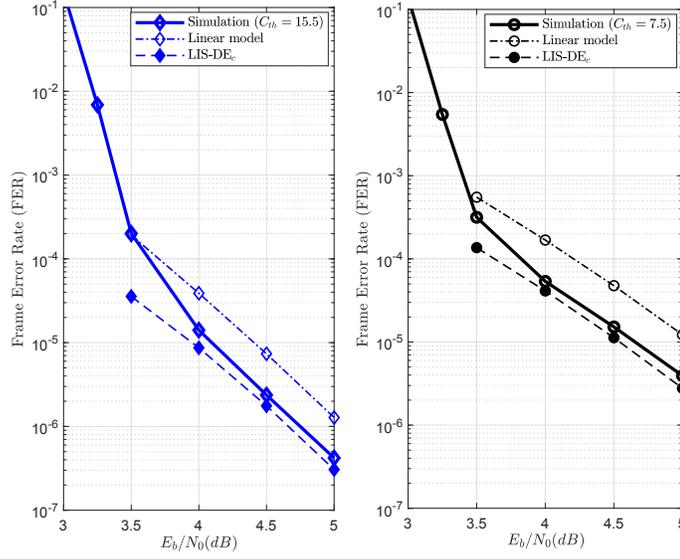
To compare the proposed LIS-DE<sub>c</sub> method with the linear state-space model of [45], we consider the scenario discussed in [45], i.e., floating-point SPA over the AWGN channel with the clipping only applied at the output of check nodes. We consider Margulis (2640, 1320) LDPC code ( $\mathcal{C}_2$ ) and two different clipping thresholds, 7 and 15. For these scenarios, the (12, 4) ETS structure is the most harmful one. The estimation results of both LIS-DE<sub>c</sub> and linear state-space methods, based on the (12, 4) ETSs, are presented in Fig. 3.11. Comparison with Monte Carlo simulation results shows that while the LIS-DE<sub>c</sub> estimates are accurate for both clipping thresholds, the linear state-space estimate is only accurate for the higher threshold and over-estimates the error floor for the lower one. The less accuracy of the linear state-space estimate for the latter case in this example may be attributed to the more severe non-linearity resulted from the lower clipping threshold. In Fig. 3.11, we have also presented LIS-DE<sub>nc</sub> estimates. As can be seen, for both threshold values, they under-estimate the error floor by up to an order of magnitude.

In the context of error floor analysis and design of codes with low error floors, it is important to know the relative harmfulness of different TS structures. A good



**Figure 3.12:** Simulation and estimation results for ETS classes (4, 2) and (5, 1) of  $\mathcal{C}_3$  under floating-point SPA with clipping at the check node outputs ( $I_{max} = 200$ ).

estimation technique should be able to accurately provide such information. In this experiment, we consider  $\mathcal{C}_3$  decoded by the floating-point SPA with clipping at the output of check nodes with two threshold values of 7.5 and 15.5. Monte Carlo simulation results indicate that the most dominant ETS classes for the threshold values of 7.5 and 15.5 are (4, 2) and (5, 1), respectively. In Fig. 3.12, we have shown the contribution of these two classes to the error floor by simulations as well as by estimation through both LIS-DE<sub>c</sub> and linear state-space methods. These results show that while the LIS-DE<sub>c</sub> estimate is accurate for all cases, the linear state-space estimate consistently over-estimates the error sometimes by up to an order of magnitude. In fact, it only provides a rather accurate estimate of the error for (5, 1) ETSs. For the (4, 2) class, it significantly over-estimates the error. As a result, while the LIS-DE<sub>c</sub> method determines the most dominant ETS class for both thresholds accurately, the linear state-space model erroneously asserts that for both thresholds the (4, 2) class is the most dominant. The total error floor estimates by the two methods are also shown in Fig. 3.13.



**Figure 3.13:** Simulation and estimation results for  $\mathcal{C}_3$  decoded by floating-point SPA with clipping at the check node outputs ( $I_{max} = 200$ ).

### 3.5 Appendix

**Proof of Lemma 2.** Based on the first order approximation of the dominant eigenvalue as a function of gain values in the vicinity of gain means, we have

$$\rho(\mathbf{A}\mathbf{G}^{(\ell)}) \approx \rho(\mathbf{A}\bar{\mathbf{G}}^{(\ell)}) + \sum_{i=1}^{\frac{m_s}{2}} \left. \frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{\partial g_i^{(\ell)}} \right|_{\mathbf{G}^{(\ell)} = \bar{\mathbf{G}}^{(\ell)}} (g_i^{(\ell)} - \bar{g}_i^{(\ell)}). \quad (3.29)$$

Suppose that  $\mathbf{w}$  and  $\mathbf{u}$  are the normalized left and right eigenvectors of  $\mathbf{A}\bar{\mathbf{G}}^{(\ell)}$  corresponding to  $\rho(\mathbf{A}\bar{\mathbf{G}}^{(\ell)})$ . Then, we have

$$\mathbf{w}^T \mathbf{A}\bar{\mathbf{G}}^{(\ell)} \mathbf{u} = \rho(\mathbf{A}\bar{\mathbf{G}}^{(\ell)}) \mathbf{w}^T \mathbf{u} = \rho(\mathbf{A}\bar{\mathbf{G}}^{(\ell)}). \quad (3.30)$$

On the other hand, the dominant eigenvectors of  $\mathbf{A}\bar{\mathbf{G}}^{(\ell)}$  can be approximated by those of  $\mathbf{A}$ . This approximation is accurate based on (3.9) if the expected gains of different missatisfied check nodes are equal. Otherwise, assuming that the differences in the average gains  $\bar{g}_i^{(\ell)}$  are rather small, then the approximation follows from the result that the eigenvectors of a matrix do not change much by small perturbations [94] (note that all zero entries of  $\mathbf{A}\bar{\mathbf{G}}^{(\ell)}$  are the same as those of  $\mathbf{A}$ , and that the non-zero

entries of  $\mathbf{A}\bar{\mathbf{G}}^{(\ell)}$  are different average gain values  $\bar{g}_i^{(\ell)}$  instead of 1s in  $\mathbf{A}$ ). From (3.30), we thus have  $\rho(\mathbf{A}\bar{\mathbf{G}}^{(\ell)}) \approx \mathbf{w}_1^T \mathbf{A}\bar{\mathbf{G}}^{(\ell)} \mathbf{u}_1 = r \mathbf{w}_1^T \bar{\mathbf{G}}^{(\ell)} \mathbf{u}_1$ . This results in the derivation of the first term on the right hand side of (3.20).

Suppose that  $\mathbf{w}'$  and  $\mathbf{u}'$  are the normalized left and right eigenvectors of  $\mathbf{A}\mathbf{G}^{(\ell)}$  corresponding to  $\rho(\mathbf{A}\mathbf{G}^{(\ell)})$ . Consider the equation

$$\mathbf{w}'^T \mathbf{A}\mathbf{G}^{(\ell)} \mathbf{u}' = \rho(\mathbf{A}\mathbf{G}^{(\ell)}) \mathbf{w}'^T \mathbf{u}' = \rho(\mathbf{A}\mathbf{G}^{(\ell)}) . \quad (3.31)$$

Taking the partial derivative of the right and the left hand sides of (3.31) with respect to  $g_i^{(\ell)}$ , we have

$$\begin{aligned} \frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{\partial g_i^{(\ell)}} &= \frac{\partial (\mathbf{w}'^T \mathbf{A}\mathbf{G}^{(\ell)} \mathbf{u}')}{\partial g_i^{(\ell)}} = \\ \frac{\partial \mathbf{w}'^T}{\partial g_i^{(\ell)}} \mathbf{A}\mathbf{G}^{(\ell)} \mathbf{u}' + \mathbf{w}'^T \frac{\partial \mathbf{A}\mathbf{G}^{(\ell)}}{\partial g_i^{(\ell)}} \mathbf{u}' + \mathbf{w}'^T \mathbf{A}\mathbf{G}^{(\ell)} \frac{\partial \mathbf{u}'}{\partial g_i^{(\ell)}} &= \\ \rho(\mathbf{A}\mathbf{G}^{(\ell)}) \left( \frac{\partial \mathbf{w}'^T}{\partial g_i^{(\ell)}} \mathbf{u}' + \mathbf{w}'^T \frac{\partial \mathbf{u}'}{\partial g_i^{(\ell)}} \right) + \mathbf{w}'^T \frac{\partial \mathbf{A}\mathbf{G}^{(\ell)}}{\partial g_i^{(\ell)}} \mathbf{u}' &= \\ \rho(\mathbf{A}\mathbf{G}^{(\ell)}) \frac{\partial (\mathbf{w}'^T \mathbf{u}')}{\partial g_i^{(\ell)}} + \mathbf{w}'^T \frac{\partial \mathbf{A}\mathbf{G}^{(\ell)}}{\partial g_i^{(\ell)}} \mathbf{u}' &= \\ 0 + \mathbf{w}'^T \frac{\partial \mathbf{A}\mathbf{G}^{(\ell)}}{\partial g_i^{(\ell)}} \mathbf{u}' . \end{aligned}$$

Since the differences in gain values are small, we assume that the dominant eigenvectors of  $\mathbf{A}\mathbf{G}^{(\ell)}$  and  $\mathbf{A}$  are approximately the same [94]. We thus have

$$\frac{\partial \rho(\mathbf{A}\mathbf{G}^{(\ell)})}{\partial g_i^{(\ell)}} \approx \mathbf{w}_1^T \frac{\partial \mathbf{A}\mathbf{G}^{(\ell)}}{\partial g_i^{(\ell)}} \mathbf{u}_1 , \quad (3.32)$$

which results in (3.21). (Note that the matrix  $\frac{\partial (\mathbf{A}\mathbf{G}^{(\ell)})}{\partial g_i^{(\ell)}}$  is binary with 0 and 1 elements. The right hand side of (3.32) is thus a constant with respect to gain values  $g_i^{(\ell)}$ .)

**Proof of Lemma 3.** To find the error indicator function, one needs to find the projection of  $\mathbf{x}^{(\ell)}$  onto the dominant left eigenvector of  $\mathbf{A}\mathbf{G}^{(\ell)}$ . Since the differences in gain values are small, we assume that the dominant eigenvectors of  $\mathbf{A}\mathbf{G}^{(\ell)}$  and  $\mathbf{A}$  are approximately the same [94]. The indicator function can then be approximated by  $\mathbf{w}_1^T \mathbf{x}^{(\ell)}$ . To prove the result, we use

$$\mathbf{w}_1^T (\mathbf{A}\mathbf{G}^{(\ell)}) = \rho(\mathbf{A}\mathbf{G}^{(\ell)}) \mathbf{w}_1^T = r g^{(\ell)} \mathbf{w}_1^T . \quad (3.33)$$

Using (3.33), it is easy to prove the claim for  $\ell = 1$ . For  $\ell = 2$ , the non-recursive form of the state vector of Equation (3.5) can be written as

$$\begin{aligned} \mathbf{x}^{(2)} &= \mathbf{A}\mathbf{G}^{(2)}\mathbf{A}\mathbf{G}^{(1)}\mathbf{B}\mathbf{L} + \mathbf{A}\mathbf{G}^{(2)}\mathbf{B}\mathbf{L} + \mathbf{A}\mathbf{G}^{(2)}\mathbf{B}_{ex}\mathbf{L}_{ex}^{(1)} \\ &\quad + \mathbf{B}\mathbf{L} + \mathbf{B}_{ex}\mathbf{L}_{ex}^{(2)}. \end{aligned}$$

Thus, we have

$$\begin{aligned} \xi_2 &= \mathbf{w}_1^T \mathbf{x}^{(2)} = r\mathbf{g}^{(2)}r\mathbf{g}^{(1)}\mathbf{w}_1^T \mathbf{B}\mathbf{L} + r\mathbf{g}^{(2)}\mathbf{w}_1^T \mathbf{B}\mathbf{L} \\ &\quad + r\mathbf{g}^{(2)}\mathbf{w}_1^T \mathbf{B}_{ex}\mathbf{L}_{ex}^{(1)} + \mathbf{w}_1^T \mathbf{B}\mathbf{L} + \mathbf{w}_1^T \mathbf{B}_{ex}\mathbf{L}_{ex}^{(2)}, \end{aligned} \quad (3.34)$$

in which we have used (3.33). It is easy to see that Equation (3.34) is the same as (3.24) for  $\ell = 2$ . It is straightforward to generalize Equation (3.34) to an arbitrary iteration  $\ell \geq 1$  to obtain (3.24).

## Chapter 4

# Error Floor Analysis of LDPC Row Layered Decoders

The chapter is organized as follows: We develop the linear state-space model of an ETS for a row layered decoder in Section 4.1. In the same section, we also establish connections between the model parameters of the row layered and flooding schedules. In Section 4.2, we analyze the effect of different row block permutations on some important parameters affecting the error floor performance of the layered decoder, and discuss the optimization of the layer ordering to minimize the error floor. In Section 4.3, we present the simulation results to evaluate the accuracy of the theoretical results in estimating the error floor.

## 4.1 Linear State-Space Model of LETSs for SPA with Row Layered Schedule

In this section, we develop a linear state-space model of LETSs for SPA with row layered schedule, and use the model to calculate the error probability of LETSs. In Subsection 4.1.1, by a proper labeling of state variables, we establish a relationship between the matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{B}_{\text{ex}}$  of the model for flooding schedule in (3.10)-(3.12) and the corresponding matrices needed in the model for the layered schedule. The recursive and non-recursive equations for the state vector of the layered schedule are then derived in Subsection 4.1.2. In Subsection 4.1.3, we present the application of DE within the linear state-space model of the layered decoder. Within this subsection, we also introduce the concept of TS layer profile which plays an important role in the proper application of DE in the model, and in identifying TSs that have the same

topology but may have different harmfulness. We then derive the gain values of the missatisfied CNs for a layered schedule in Subsection 4.1.4. The spectral properties of the LETS system matrices in layered decoders are analyzed in Subsection 4.1.5. Finally, in Subsection 4.1.6, we calculate the error probability of a LETS using the linear state-space model.

### 4.1.1 Relationship between model matrices of flooding and row layered schedules

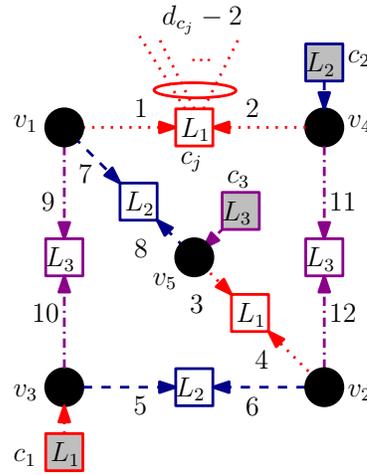
To establish a relationship between the matrices that appear in the two models, it is helpful to label the state variables in a certain order. Consider an  $(a, b)$  LETS  $\mathcal{S}$  with  $m_s$  state variables  $x_1, \dots, x_{m_s}$ . Suppose that the missatisfied CNs of  $\mathcal{S}$  are from  $J$  different layers of the parity-check matrix  $\mathbf{H}$ , where  $J \leq m_b$ . We denote these layers by  $L_1, \dots, L_J$ , where an smaller index for a layer implies that the CNs in that layer are updated earlier in an iteration. In the following, we say that such a LETS has  $J$  layers. We use the notation  $n_{L_j}$  to denote the number of state variables that are updated in layer  $L_j$ , for  $1 \leq j \leq J$ . We thus have  $n_{L_1} + \dots + n_{L_J} = m_s$ . To assign the state variables to different internal messages of  $\mathcal{S}$ , we start with the messages that are updated in  $L_1$ , and assign to them variables  $x_1, \dots, x_{n_{L_1}}$ . We then move on to the messages that are updated in  $L_2$ , and assign to them variables  $x_{n_{L_1}+1}, \dots, x_{n_{L_1}+n_{L_2}}$ . We will continue this process all the way to  $L_J$  until all the  $m_s$  messages have their state variables assigned to them. We call this labeling of state variables *systematic labeling*.

Based on the systematic labeling, the matrix  $\mathbf{A}$  of flooding schedule will be a  $J \times J$  array of matrices  $\mathbf{A}_{ij}$ ,  $1 \leq i \leq J, 1 \leq j \leq J$ , where the size of the matrix  $\mathbf{A}_{ij}$  is  $n_{L_i} \times n_{L_j}$ , and the  $J$  diagonal matrices are all-zero, i.e.,  $\mathbf{A}_{ii} = \mathbf{0}$ ,  $1 \leq i \leq J$ . Based on the partitioning of the state variables according to their layer, the rows of matrices  $\mathbf{B}$  and  $\mathbf{B}_{ex}$  can also be partitioned into  $J$  blocks, with the  $i$ th row block containing  $n_{L_i}$  rows. We refer to this representation of matrices as *systematic form*.

**Example 3.** Consider the  $(5, 3)$  LETSs of the Tanner  $(155, 64)$  code discussed in Example 2. The Tanner code has three row layers, i.e.,  $m_b = 3$ . Each of the  $(5, 3)$  LETSs has 3 unsatisfied and 6 missatisfied check nodes. Each of the three unsatisfied check nodes belongs to a different row layer. Out of 6 missatisfied check nodes, each set of two belongs to a different row layer ( $J = 3$ ). This is shown in Fig. 4.1. To

distinguish the layers, different colors and line types are used in Fig. 4.1. As can be seen, the selection of edge labels, which reflects the indices of corresponding state variables, are in systematic form. The corresponding matrices are given by:

$$\mathbf{A} = \left[ \begin{array}{c|c|c} \mathbf{0} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \hline \mathbf{A}_{21} & \mathbf{0} & \mathbf{A}_{23} \\ \hline \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{0} \end{array} \right] = \left[ \begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right],$$



**Figure 4.1:** A  $(5, 3)$  LETS of the Tanner  $(155, 64)$  code whose edges are labeled in a systematic form. The edges of different layers are distinguished by different colors and line types.

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \mathbf{B}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_{ex} = \begin{bmatrix} \mathbf{B}_{ex1} \\ \mathbf{B}_{ex2} \\ \mathbf{B}_{ex3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In a layered schedule, the state variables of a LETS are updated in  $J$  rounds, each round corresponding to one row layer, within one iteration. Corresponding to the

updating of each layer, there thus exists a set of matrices used in the model. The  $m_s \times m_s$  matrix that represents the relationship among the state variables corresponding to the updating of layer  $j$  is denoted by  $\mathbf{A}_j, 1 \leq j \leq J$ , and is referred to as the “transition matrix of layer  $j$ .” Matrix  $\mathbf{A}_j$  has the same block structure as matrix  $\mathbf{A}$ . In fact, the  $j$ th row block of the two matrices are identical. Matrix  $\mathbf{A}_j$ , however, has zero blocks everywhere else other than the diagonal blocks that are each an identity matrix. This is to indicate that as the messages in the  $j$ th layer are updated, all the other messages in the LETS remain unchanged. Similarly, notation  $\mathbf{B}_j$  is used to denote the  $m_s \times a$  matrix that is responsible for the contribution of channel LLRs in the state variables that are updated in layer  $j$ . This matrix has the same row block structure as in  $\mathbf{B}$ , with the difference that except for row block  $j$  (consisting of  $n_{L_j}$  rows) that is identical to that of  $\mathbf{B}$ , all the other row blocks are zero. Similarly, for the contribution of unsatisfied CN messages to the state variables of layer  $j$ , the  $m_s \times b$  matrix  $\mathbf{B}_{exj}$  can be defined as a matrix with  $J$  row blocks whose  $j$ th row block is equal to that of  $B_{ex}$  while the other row blocks are all zero. This matrix, however, is not directly utilized in the model. The reason is that, two groups of unsatisfied CN messages contribute to the state variables in layer  $j$  at iteration  $\ell$ . The first group are those that are updated at the end of  $(\ell - 1)$ th iteration,  $L_{ex}^{(\ell-1)}$ . The second group are the ones that are updated earlier in iteration  $\ell$ ,  $L_{ex}^{(\ell)}$ . Correspondingly, the matrices  $\overset{\triangleleft}{\mathbf{B}}_{exj}$  and  $\overset{\triangleright}{\mathbf{B}}_{exj}$  are defined to account for the two contributions, respectively, and we have

$$\overset{\triangleleft}{\mathbf{B}}_{exj} + \overset{\triangleright}{\mathbf{B}}_{exj} = \mathbf{B}_{exj}. \quad (4.1)$$

**Example 4.** For the same LETS structure discussed in Example 3, the model matrices corresponding to the second layer of row layered schedule are the followings:

$$\mathcal{A}_2 = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{0} & \mathbf{A}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \mathcal{B}_2 = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_2 \\ \mathbf{0} \end{bmatrix}, \mathcal{B}_{ex2}^{\triangleleft} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathcal{B}_{ex2}^{\triangleright} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In the second layer, the unsatisfied CNs  $c_1$  and  $c_3$  contribute to the values of state variables  $x_5$  and  $x_8$ , respectively. While the message from  $c_1$  is updated at the  $L_1$  layer (before  $L_2$ ), the message from  $c_3$  is updated only in  $L_3$  (after  $L_2$ ). So, the contribution from  $c_1$  to  $x_5$  is reflected through  $\mathcal{B}_{ex2}^{\triangleright}$  which uses a newly updated (at the current iteration) version of CN message, while the contribution from  $c_3$  to  $x_8$  is via  $\mathcal{B}_{ex2}^{\triangleleft}$  and uses the CN message updated at the end of the previous iteration.

The last set of matrices that we need in the linear state-space model of a row layered decoder are the gain matrices. In the flooding schedule, the impact of external connections of the missatisfied CNs on the internal messages of a LETS is modelled by an iteration dependent scalar gain,  $\bar{g}'_\ell$  [45]. This, in fact, corresponds to a diagonal gain matrix with equal diagonal elements, i.e.,  $\bar{g}'_\ell \mathbf{I}_{m_s \times m_s}$ , and assumes that all the missatisfied CNs have equal impact on the internal messages of the LETS. For a layered decoder, however, there can be a significant difference among the distribution of external messages of different missatisfied CNs, depending on their layer and their external connections. To model the missatisfied CN gains within a given layer  $j$  at

iteration  $\ell$ , we thus use the gain matrix  $\mathbf{g}_j^{(\ell)}$ . To define  $\mathbf{g}_j^{(\ell)}$ , we first consider the following  $m_s \times m_s$  diagonal matrix:

$$\mathbf{G}^{(\ell)} = \begin{bmatrix} \bar{g}_1^{(\ell)} & 0 & \cdots & 0 \\ 0 & \bar{g}_2^{(\ell)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \bar{g}_{m_s}^{(\ell)} \end{bmatrix},$$

whose diagonal entries are the linear gains corresponding to  $m_s$  state variables. (Note that the state variables corresponding to the same missatisfied CN have equal gains.) We then partition  $\mathbf{G}^{(\ell)}$  into  $J \times J$  block matrices. This partitioning corresponds to different layers of the LETS and is similar to the partitioning of the transition matrix  $\mathbf{A}$ . As a result, we have

$$\mathbf{G}^{(\ell)} = \begin{bmatrix} \mathbf{G}_1^{(\ell)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2^{(\ell)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{G}_J^{(\ell)} \end{bmatrix}.$$

Now, the gain matrix  $\mathbf{g}_j^{(\ell)}$  corresponding to layer  $j$  is defined as  $\mathbf{G}^{(\ell)}$  in which all the diagonal blocks, except the  $j$ th one, are replaced with the identity matrix. For example, for  $j = 2$ ,

$$\mathbf{g}_2^{(\ell)} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2^{(\ell)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

### 4.1.2 Linear State-Space Model of LETSs for Row Layered SPA

Using the matrices presented in the previous subsection, we have the following linear state-space model of a LETS for row layered SPA:

$$\tilde{\mathbf{x}}^{(0;j)} = \mathbf{0} \quad \text{for } \ell = 0, 1 \leq j \leq J \quad (4.2)$$

$$\begin{aligned} \tilde{\mathbf{x}}^{(\ell;j)} = & \quad \text{for } \ell \geq 1, 1 \leq j \leq J \\ & \mathcal{G}_j^{(\ell)} (\mathcal{A}_j \tilde{\mathbf{x}}^{(\ell-\delta_{j1}, j-1+J\delta_{j1})} + \mathcal{B}_j \mathbf{L} + \overset{\triangleleft}{\mathbf{B}}_{exj} \mathbf{L}_{ex}^{(\ell-1)} + \overset{\triangleright}{\mathbf{B}}_{exj} \mathbf{L}_{ex}^{(\ell)}). \end{aligned} \quad (4.3)$$

In the above model, the state vector at layer  $j$  of iteration  $\ell$  is denoted by  $\tilde{\mathbf{x}}^{(\ell;j)}$ . The vectors  $\mathbf{L}$ ,  $\mathbf{L}_{ex}^{(\ell-1)}$  and  $\mathbf{L}_{ex}^{(\ell)}$  are the inputs to the model and represent channel LLRs and messages from unsatisfied CNs at iterations  $\ell - 1$  and  $\ell$ , respectively, where  $\mathbf{L}_{ex}^{(0)} = \mathbf{0}$ . Also,  $\delta_{j1}$ , is the Kronecker delta function which is equal to 1, when  $j = 1$ , and is zero, otherwise. Equation (5.2) implies that at the first layer of every iteration, the state vector is updated based on the state vector from the last layer of the previous iteration, while in the other layers, the updated states are a function of the state vector of the previous layer within the same iteration.

Next, we use induction to derive a non-recursive equation for the state vector at the end of iteration  $\ell$ , i.e.,  $\tilde{\mathbf{x}}^{(\ell;J)}$ . For this, we first define some new matrices. The first matrix is defined as

$$\tilde{\mathbf{A}}_{J \rightarrow k}^{(\ell)} = (\mathcal{G}_J^{(\ell)} \mathcal{A}_J) (\mathcal{G}_{J-1}^{(\ell)} \mathcal{A}_{J-1}) \dots (\mathcal{G}_k^{(\ell)} \mathcal{A}_k), \quad (4.4)$$

which is, an ordered multiplication of the scaled version of transition matrices of different layers,  $\mathcal{G}_j^{(\ell)} \mathcal{A}_j$ , from the layer with maximum index  $J$  down to the  $k$ th layer,  $k \geq 1$ . By using  $\tilde{\mathbf{A}}_{J \rightarrow k}^{(\ell)}$ , we define three other matrices as follows:

$$\tilde{\mathbf{B}}^{(\ell)} = \tilde{\mathbf{A}}_{J \rightarrow 2}^{(\ell)} \mathcal{G}_1^{(\ell)} \mathcal{B}_1 + \tilde{\mathbf{A}}_{J \rightarrow 3}^{(\ell)} \mathcal{G}_2^{(\ell)} \mathcal{B}_2 + \dots + \mathcal{G}_J^{(\ell)} \mathcal{B}_J, \quad (4.5)$$

$$\overset{\triangleright}{\mathbf{B}}_{ex}^{(\ell)} = \tilde{\mathbf{A}}_{J \rightarrow 2}^{(\ell)} \mathcal{G}_1^{(\ell)} \overset{\triangleright}{\mathbf{B}}_{ex1} + \tilde{\mathbf{A}}_{J \rightarrow 3}^{(\ell)} \mathcal{G}_2^{(\ell)} \overset{\triangleright}{\mathbf{B}}_{ex2} + \dots + \mathcal{G}_J^{(\ell)} \overset{\triangleright}{\mathbf{B}}_{exJ}, \quad (4.6)$$

$$\overset{\triangleleft}{\mathbf{B}}_{ex}^{(\ell)} = \tilde{\mathbf{A}}_{J \rightarrow 2}^{(\ell)} \mathcal{G}_1^{(\ell)} \overset{\triangleleft}{\mathbf{B}}_{ex1} + \tilde{\mathbf{A}}_{J \rightarrow 3}^{(\ell)} \mathcal{G}_2^{(\ell)} \overset{\triangleleft}{\mathbf{B}}_{ex2} + \dots + \mathcal{G}_J^{(\ell)} \overset{\triangleleft}{\mathbf{B}}_{exJ}. \quad (4.7)$$

Finally, the non-recursive formula of the state vector at the end of iteration  $\ell$  is

derived as

$$\begin{aligned} \tilde{\mathbf{x}}^{(\ell, J)} &= \sum_{i=1}^{\ell} \left( \overset{\rightarrow}{\prod}_{j=i+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j)} \right) \tilde{\mathbf{B}}^{(i)} \mathbf{L} \\ &+ \sum_{i'=1}^{\ell} \left( \overset{\rightarrow}{\prod}_{j'=i'+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j')} \right) \left( \mathbf{B}_{ex}^{\triangleleft(i')} \mathbf{L}_{ex}^{(i'-1)} + \mathbf{B}_{ex}^{\triangleright(i')} \mathbf{L}_{ex}^{(i')} \right), \end{aligned} \quad (4.8)$$

where the right arrow on top of the product sign denotes the matrix product applied from the left. For example,

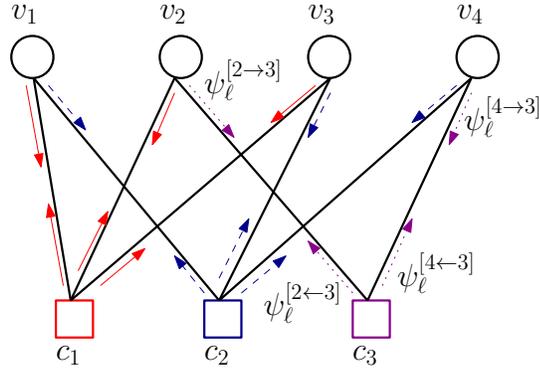
$$\overset{\rightarrow}{\prod}_{j=i+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j)} = \tilde{\mathbf{A}}_{J \rightarrow 1}^{(\ell)} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(\ell-1)} \cdots \tilde{\mathbf{A}}_{J \rightarrow 1}^{(i+1)}.$$

In the rest of the chapter, since we only consider the state vector  $\tilde{\mathbf{x}}^{(\ell, J)}$  at the end of each iteration, for simplicity, we may drop the index  $J$ , and represent the state vector by  $\tilde{\mathbf{x}}^{(\ell)}$ .

In Subsection 4.1.6, Equation (4.8) will be used to estimate the probability of a LETS failure in the layered decoder.

### 4.1.3 The Application of DE to the Layered Decoder

In the state-space analysis of layered decoders, we use DE to calculate the distribution of messages entering the LETS from the rest of the Tanner graph. To apply DE to a layered decoder of a QC-LDPC code, for each iteration, one needs to derive two distributions corresponding to each edge  $e$  of the base graph. The two distributions correspond to the messages passed from CNs (VNs) to VNs (CNs) of the Tanner graph that are connected by the cluster of edges associated with  $e$ . If edge  $e$  connects VN  $i$  to CN  $j$  in the base graph, we say that the cluster of edges corresponding to  $e$  connect Type- $i$  VNs to Type- $j$  CNs in the Tanner graph. The computation tree for the calculation of such distributions is not only a function of the base graph but also depends on the order in which the messages of different row layers are updated. To identify these distributions, we use the notations  $\psi_{\ell}^{[i \rightarrow j]}$  and  $\psi_{\ell}^{[i \leftarrow j]}$  to denote the probability distribution of the messages from Type- $i$  VNs to Type- $j$  CNs and vice versa in iteration  $\ell$ , respectively.



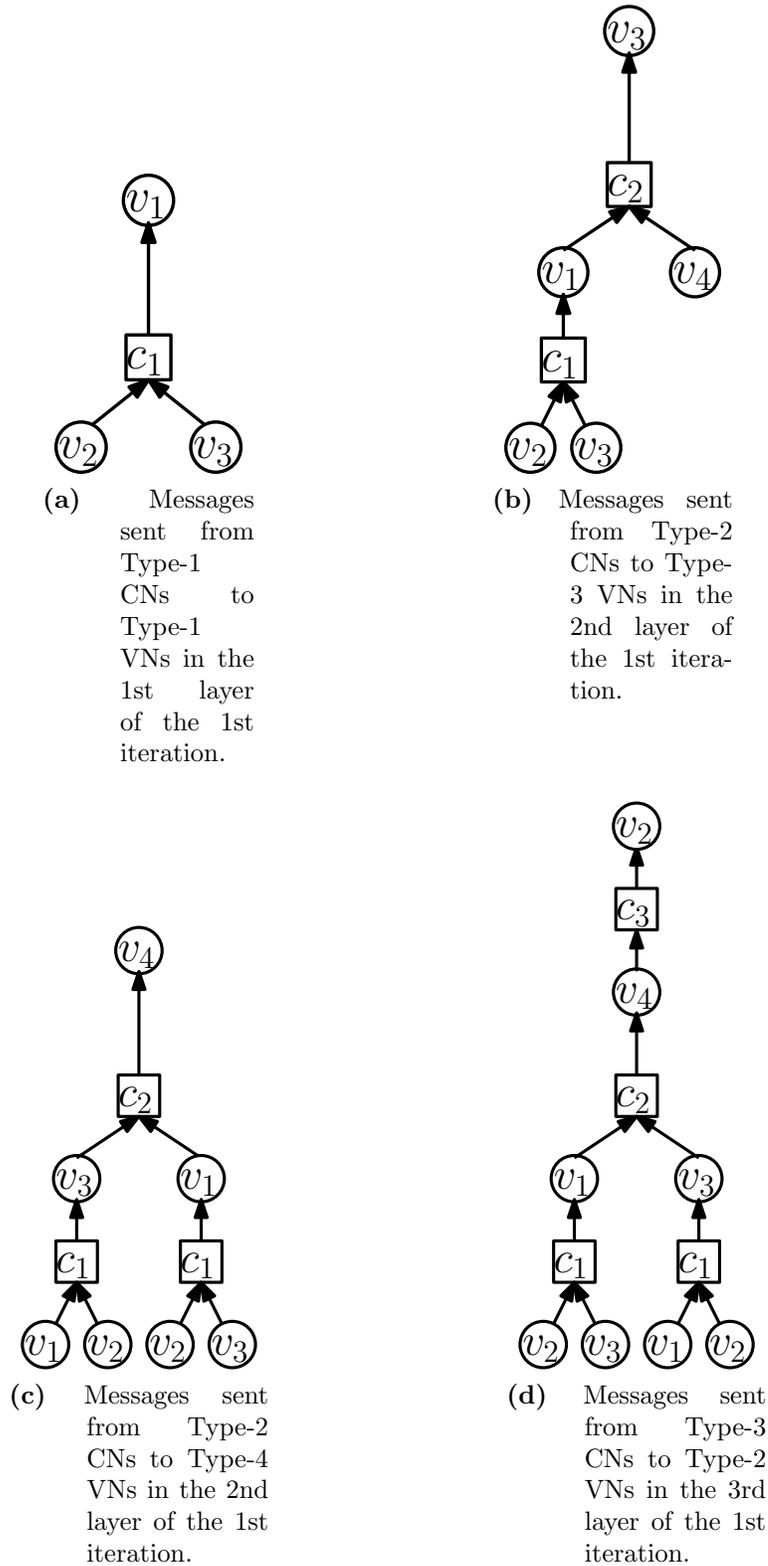
**Figure 4.2:** Base graph of Example 5: The solid (red), dashed (blue) and dotted (purple) arrows, correspond to messages passed within layers  $L_1$  to  $L_3$ , respectively. The symbols next to the dotted arrows show the probability distributions of messages updated in the 3rd layer of the  $\ell$ th iteration.

**Example 5.** Consider the base graph of Fig. 4.2 corresponding to the following base matrix:

$$\mathbf{H}_b = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}. \quad (4.9)$$

The messages updated in different layers are identified on the base graph with different colors and line types. Suppose that the three layers are updated in accordance to the increasing row index. The computation trees of some of the CN to VN messages for the first iteration are demonstrated in Figs. 4.3a to 4.3d. As can be seen, the computation trees are different depending on the type of the VN at the root, and the type of the connecting CN. In particular, the trees have different depths within the same iteration. The trees will also change by changing the order in which the layers are updated. This is unlike the flooding schedule, for which the depth of all computation trees at iteration  $\ell$  is  $2\ell$  regardless of the type of VNs or CNs, or any permutation of row layers. This implies that, for a layered decoder, the message distributions change with changing the order in which the row layers are updated.

Consider a LETS whose missatisfied CNs belong to  $J$  layers,  $L_1, \dots, L_J$ . Consider a permutation  $\pi$  over the set of integer numbers  $\{1, \dots, J\}$ . Assume that the messages within the row layers of the parity-check matrix (within one iteration) are updated



**Figure 4.3:** The computation trees of certain messages at different layers of the 1st iteration.

such that CNs with type  $\pi(1)$  within the LETS are updated first, followed by CNs with type  $\pi(2)$  and so on. This means that missatisfied CNs of the LETS that belong to  $L_i$  are of Type- $\pi(i)$ , for  $i = 1, \dots, J$ . To obtain the distribution of incoming messages to the LETS from the rest of the Tanner graph using DE, one needs to know not only the topology of the TS, but also the following information:

1. types of all missatisfied and unsatisfied CNs of the TS as well as the layer to which each CN of the TS belongs (the latter corresponds to knowing the permutation  $\pi$ ),
2. the type and layer of all the variable nodes that are externally connected to all the mis-satisfied CNs, and the type and layer of all internal VNs connected to unsatisfied CNs.

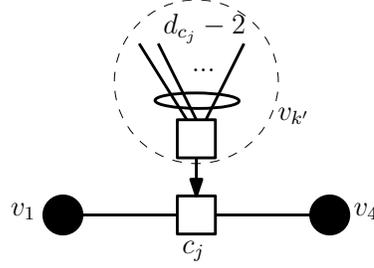
We refer to the above information as *TS layer profile* (TSLP). We note that the TSLP can be different for isomorphic TSs, thus, resulting in different harmfulness for such TSs.

**Example 6.** Consider the  $(5, 3)$  LETS of Fig. 4.1, and assume that the layers of the code are updated in the increasing order of row indices. Variable nodes  $v_1$  and  $v_5$  are of Type-3 and Type-5, respectively, while VNs  $v_2$ ,  $v_3$  and  $v_4$ , are of Type-1. As an example of a missatisfied CN, consider  $c_j$ . Check node  $c_j$  is of Type-1 and is in  $L_1$ . Also, the  $d_{c_j} - 2 = 3$  external VNs of  $c_j$  are of Type-2, -4 and -5, respectively. As a result, to derive the gain value corresponding to  $c_j$  in the linear state-space model, one needs the probability distributions of messages from Type-2, -4 and -5 VNs sent to Type-1 CNs from the DE results. As an example of an unsatisfied CN, consider  $c_2$ . Check node  $c_2$  is of Type-2 and is in  $L_2$ , and is connected to the internal VN  $v_4$ , which is of Type-1. The probability distribution of the messages sent from  $c_2$  to  $v_4$ , thus follows the DE results for messages from Type-2 CNs to Type-1 VNs.

**Lemma 5.** Let  $p$  be the lifting degree of a QC-LDPC code  $\mathcal{C}$ , and let  $\mathcal{S}$  be a LETS of  $\mathcal{C}$ . Then, the TSLP of  $\mathcal{S}$  is the same as the TSLP of any of the isomorphic LETSs whose VNs are obtained by circularly shifting (modulo  $p$ ) the VNs of  $\mathcal{S}$ .

#### 4.1.4 Missatisfied CN Gain Model in Layered Decoders

In Subsection 4.1.1, we introduced the matrix  $\mathbf{G}^{(\ell)}$  whose diagonal elements are the gains associated with missatisfied CNs at different layers of the  $\ell$ th iteration. Due to



**Figure 4.4:** The external messages of a missatisfied CN are represented by a virtual VN.

the difference in the distributions of external messages entering different missatisfied CNs, the gain for each missatisfied CN needs to be calculated separately. For simplicity, the external connections of each missatisfied CN are represented by a virtual VN, as shown in Fig. 4.4. Without loss of generality, we consider VNs  $v_1$  and  $v_4$  of the (5, 3) LETS of Fig. 4.1 with the missatisfied CN  $c_j$  connecting them (as shown in Fig. 4.4). The virtual VN is denoted by  $v_{k'}$  in Fig. 4.4. The message  $L_\ell^{[4\leftarrow j]}$  can then be calculated as

$$L_\ell^{[4\leftarrow j]} = f(L_\ell^{[1\rightarrow j]}, L_\ell^{[k'\rightarrow j]}), \quad (4.10)$$

where  $f(\cdot, \cdot)$  is the box-plus operation given in (2.7). For small values of  $L_\ell^{[1\rightarrow j]}$ , i.e.,  $L_\ell^{[1\rightarrow j]} \approx 0$ , the CN linear estimation based on Taylor expansion can be obtained as

$$L_\ell^{[4\leftarrow j]} \approx f_{x_1}(0, L_\ell^{[k'\rightarrow j]})L_\ell^{[1\rightarrow j]}, \quad (4.11)$$

where  $f_{x_1}$  represents the partial derivative of  $f(x_1, x_2)$  with respect to  $x_1$ . We thus have

$$\begin{aligned} f_{x_1}(0, L_\ell^{[k'\rightarrow j]}) &= \ln \left( \frac{e^{L_\ell^{[k'\rightarrow j]}} - 1}{e^{L_\ell^{[k'\rightarrow j]}} + 1} \right) \\ &= \tanh \left( \frac{L_\ell^{[k'\rightarrow j]}}{2} \right). \end{aligned} \quad (4.12)$$

The average gain corresponding to CN  $c_j$ ,  $\bar{g}_{c_j}^{(\ell)}$ , is then obtained as

$$\bar{g}_{c_j}^{(\ell)} = \int_{-\infty}^{\infty} \tanh\left(\frac{\lambda}{2}\right) \hat{\psi}_\ell^{[k'\rightarrow j]}(\lambda) d\lambda, \quad (4.13)$$

in which  $\hat{\psi}_\ell^{[k' \rightarrow j]}$  is the probability distribution of the message from virtual VN,  $v_{k'}$ , to the missatisfied CN  $c_j$  at iteration  $\ell$ . Given the TSLP of a LETS, this distribution can be calculated for different missatisfied CNs using DE. It is noted that there are  $\frac{m_s}{2}$  mis-satisfied CNs in a LETS subgraph, and that each of the gains calculated by (4.13) should be used for the two state variables that have the corresponding CN in common.

We further modify Equation (4.13) to take into account the effect of polarity inversion, i.e., whenever an error occurs in the messages from the virtual VN,  $v_{k'}$ , for  $k' = \{1, \dots, \frac{m_s}{2}\}$ , the polarity of the state variables passing through the corresponding missatisfied CN is altered. The probability of polarity inversion is thus calculated by

$$P_{inv,\ell}^{[k' \rightarrow j]} = \int_{-\infty}^0 \hat{\psi}_\ell^{[k' \rightarrow j]}(\lambda) d\lambda. \quad (4.14)$$

To incorporate the polarity inversion in the model, we modify the average gains as follows:

$$\bar{g}_{c_j}^{(\ell)} = (1 - P_{inv,\ell}^{[k' \rightarrow j]}) \bar{g}_{c_j}^{(\ell)}, \quad (4.15)$$

where  $\bar{g}_{c_j}^{(\ell)}$  is given by (4.13).

### 4.1.5 Spectral Properties of LETS System Matrices in Layered Decoders

A careful study of Equation (4.8) reveals that the iteration dependent matrix

$$\tilde{\mathbf{A}}_{J \rightarrow 1}^{(\ell)} = (\mathcal{G}_J^{(\ell)} \mathcal{A}_J) (\mathcal{G}_{J-1}^{(\ell)} \mathcal{A}_{J-1}) \dots (\mathcal{G}_1^{(\ell)} \mathcal{A}_1), \quad (4.16)$$

plays a crucial role in the evolution of the state vector, and consequently in the failure rate, of a LETS. In the event of a failure, as the iterations progress, the average gains of missatisfied CNs tend to one. We thus have

$$\lim_{\ell \rightarrow \infty} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(\ell)} = \mathcal{A}_J \mathcal{A}_{J-1} \dots \mathcal{A}_1 \triangleq \tilde{\mathbf{A}}_{J \rightarrow 1}. \quad (4.17)$$

The matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ , obtained by the multiplication of transition matrices of different layers of the LETS, is iteration independent. In the state-space analysis of the layered decoder,  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  plays a similar role as the transition matrix  $\mathbf{A}$  does in the model for a flooding decoder. In the following, we refer to  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  as the transition matrix of the

layered decoder, and investigate its spectral properties. In particular, we demonstrate that, unlike the case for flooding where for majority of LETSs, the transition matrix  $\mathbf{A}$  is irreducible, the transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  for layered decoding is reducible. We also show that the spectral radius of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  is always larger than that of  $\mathbf{A}$ .

Using the systematic form of system model matrices as described in Subsection 4.1.1, the first column block of the transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ , consisting of  $n_{L_1}$  columns, is zero. We thus have the following result.

**Lemma 6.** *The transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  of a LETS in a layered decoder is reducible.*

**Example 7.** *The transition matrix,  $\tilde{\mathbf{A}}_{3 \rightarrow 1}$ , of the (5, 3) LETS in Fig. 4.1 (also considered in Examples 2 and 3) is equal to*

$$\begin{aligned}
 \tilde{\mathbf{A}}_{3 \rightarrow 1} = \mathcal{A}_3 \mathcal{A}_2 \mathcal{A}_1 &= \left[ \begin{array}{c|cc} \mathbf{0} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \hline \mathbf{0} & \mathbf{A}_{21} \mathbf{A}_{12} & \mathbf{A}_{21} \mathbf{A}_{13} + \mathbf{A}_{23} \\ \hline \mathbf{0} & \mathbf{A}_{31} \mathbf{A}_{12} + \mathbf{A}_{32} \mathbf{A}_{21} \mathbf{A}_{12} & \mathbf{A}_{31} \mathbf{A}_{13} + \mathbf{A}_{32} \mathbf{A}_{21} \mathbf{A}_{13} + \\ & & \mathbf{A}_{32} \mathbf{A}_{23} \end{array} \right] \\
 &= \left[ \begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{array} \right]. \tag{4.18}
 \end{aligned}$$

**Theorem 3.** [79, p. 50] For every real  $n \times n$  matrix  $\mathbf{M}$ , there exists an  $n \times n$  permutation matrix  $\mathbf{P}$  such that

$$\mathbf{PMP}^T = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \cdots & \mathbf{M}_{1t} \\ 0 & \mathbf{M}_{22} & \cdots & \mathbf{M}_{2t} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{M}_{tt} \end{bmatrix}, \quad (4.19)$$

is a  $t \times t$  array of matrices, where each of the diagonal matrices  $\mathbf{M}_{ii}$ ,  $1 \leq i \leq t$ , is either irreducible or a  $1 \times 1$  null matrix (zero).

The form of a matrix given by (4.19) is called *Frobenius normal form* or FNF, in brief. In the above theorem, if  $\mathbf{M}$  is irreducible, then  $t = 1$ .

**Theorem 4.** [79, p. 51] Let  $\mathbf{M}$  be an  $n \times n$  non-negative matrix. Then,

- (a)  $\mathbf{M}$  has a real non-negative eigenvalue equal to its spectral radius  $\rho(\mathbf{M})$ . In addition, the corresponding real eigenvalue is positive unless  $\mathbf{M}$  is reducible and its FNF is strictly upper triangular.<sup>1</sup>
- (b) The associated eigenvector of  $\rho(\mathbf{M})$  is non-negative.

The proof of Theorem 4 follows from Theorem 1 if  $\mathbf{M}$  is irreducible. If  $\mathbf{M}$  is reducible, then by considering the FNF  $\mathbf{PMP}^T$  of  $\mathbf{M}$ , and the fact that the eigenvalues of  $\mathbf{M}$  are the union of the eigenvalues of the diagonal matrices of FNF, one can see that  $\mathbf{M}$  has either a real positive eigenvalue equal to its spectral radius (if there is at least one irreducible diagonal matrix in FNF), or the spectral radius is zero (if all the diagonal matrices of FNF are  $1 \times 1$  zero matrices).

**Theorem 5.** [80, Theorem 22] Let  $\mathbf{M}$  be an  $n \times n$  real non-negative matrix. Suppose that for a non-negative vector  $\mathbf{x} \geq \mathbf{0}$  and  $\mathbf{x} \neq \mathbf{0}$ , we have  $\mathbf{M}\mathbf{x} - \beta\mathbf{x} \geq \mathbf{0}$ , where  $\beta > 0$  is a constant. Then,

$$\rho(\mathbf{M}) \geq \beta,$$

with the inequality being strict if  $\mathbf{M}\mathbf{x} - \beta\mathbf{x} > \mathbf{0}$ .

The following result is simple to prove.

---

<sup>1</sup>A matrix  $\mathbf{M}$  is called strictly upper triangular if  $\mathbf{M}(i, j) = 0$  for  $i \geq j$ .

**Lemma 7.** Let  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  and  $\mathbf{A}$  be the transition matrices of a LETS in layered and flooding decoders, respectively. We then have

$$\tilde{\mathbf{A}}_{J \rightarrow 1} = \mathbf{A} + \mathbf{A}'(\mathbf{A} - \mathbf{I}), \quad (4.20)$$

where

$$\mathbf{A}' = \mathbf{A}_l^{J-1} + \mathbf{A}_l^{J-2} + \cdots + \mathbf{A}_l, \quad (4.21)$$

and  $\mathbf{A}_l$  is a lower triangular matrix whose non-zero elements (under main diagonal) are equal to those of matrix  $\mathbf{A}$ .

**Proposition 1.** Let  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  and  $\mathbf{A}$  be the transition matrices of a LETS in layered and flooding decoders, respectively. If  $\mathbf{A}$  is irreducible, then

$$\rho(\tilde{\mathbf{A}}_{J \rightarrow 1}) \geq \rho(\mathbf{A}).$$

*Proof.* Since  $\mathbf{A}$  is irreducible, it has a simple positive eigenvalue,  $r$ , which is equal to  $\rho(\mathbf{A})$ . To  $r$ , there corresponds a positive right eigenvector,  $\mathbf{u}_1$ . By right multiplication of (4.20) with  $\mathbf{u}_1$ , we have

$$\tilde{\mathbf{A}}_{J \rightarrow 1} \mathbf{u}_1 = r \mathbf{u}_1 + \mathbf{A}'(r - 1) \mathbf{u}_1. \quad (4.22)$$

Since  $r > 1$ ,  $\mathbf{u}_1 > \mathbf{0}$  and  $\mathbf{A}' \geq \mathbf{0}$ , we have  $\mathbf{A}'(r - 1) \mathbf{u}_1 \geq \mathbf{0}$  and  $r \mathbf{u}_1 > \mathbf{0}$ , and therefore

$$\tilde{\mathbf{A}}_{J \rightarrow 1} \mathbf{u}_1 - r \mathbf{u}_1 \geq \mathbf{0}. \quad (4.23)$$

Using this in Theorem 5 completes the proof. ■

Based on Lemma 6, we know that  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  is a non-negative reducible matrix. As a result, the FNF of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  has either irreducible or  $1 \times 1$  null matrices as its diagonal blocks. In the following, we demonstrate that the FNF of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  has only one irreducible diagonal block with all the other blocks being equal to  $1 \times 1$  zero matrices. To show this, we use the directed edge graph (digraph) representation of the transition matrices. This representation, which was also used in [45], represents each state variable with a node in the graph and the dependencies among them by directed edges, i.e., a directed edge  $x_i x_j$  in the graph means that state variable  $x_j$ , corresponding to the *head* node of the edge, at iteration  $\ell$  is a function of state variable  $x_i$ , corresponding to the *tail* node of the edge, at iteration  $\ell - 1$ . In the digraph, the tail (head)

nodes of all the incoming (outgoing) edges of a node  $x$ , i.e., all the edges with head (tail)  $x$ , are called the *parents* (*children*) of  $x$ .

It appears that the adjacency matrix of the digraph is equal to the transpose of the transition matrix. In the following, we use notations  $D_f(\mathcal{S}) = (V_f, E_f)$  and  $D_l(\mathcal{S}) = (V_l, E_l)$  to denote the digraphs corresponding to a LETS  $\mathcal{S}$  in flooding and layered decoding, respectively, where the set of nodes and edges in each graph are denoted by  $V$  and  $E$ , respectively, with indices  $f$  and  $l$  indicating “flooding” and “layered,” respectively.

A digraph is *strongly connected* if there exists a directed walk between any pair of its nodes. Consider a LETS  $\mathcal{S}$  with an irreducible transition matrix  $\mathbf{A}$  in a flooding decoder. The irreducibility of  $\mathbf{A}$  implies that the digraph  $D_f(\mathcal{S}) = (V_f, E_f)$  is strongly connected [81].

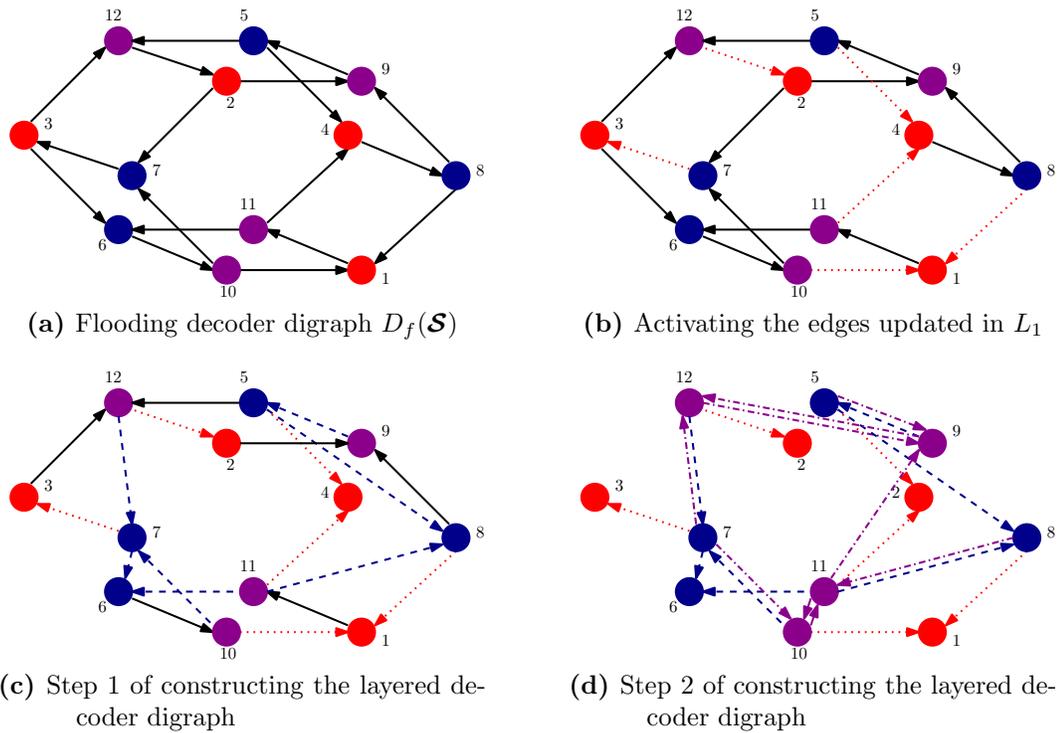
**Example 8.** *The digraph  $D_f(\mathcal{S})$  of the flooding decoder for the (5, 3) LETS of Fig. 4.1 is shown in Fig. 4.5a. As can be seen this digraph is strongly connected.*

The digraph  $D_l(\mathcal{S})$  of a LETS  $\mathcal{S}$  for a layered decoder can be obtained from its adjacency matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}^T$ . The digraph  $D_l(\mathcal{S})$  can also be constructed from the flooding digraph  $D_f(\mathcal{S})$  through  $J - 1$  steps, each corresponding to one layer of decoding. In Step  $j, j = 1, \dots, J - 1$ , one considers all the nodes corresponding to state variables that are updated in layer  $L_{j+1}$  and their incoming edges. If the tail of any of such incoming edges has been updated in the previous layers within the same iteration, then the incoming edge is removed and new incoming edges are created from the parents of the updated tail to the head of the removed edge.

**Example 9.** *The steps for the construction of  $D_l(\mathcal{S})$  from  $D_f(\mathcal{S})$  for the (5, 3) LETS of Fig. 4.1 are shown in Fig. 4.5, where different colors and line types are used to identify the state variables and updated edges in different layers. It can be verified that the adjacency matrix of the resulted digraph of Fig. 4.5d is equal to the transpose of the matrix  $\tilde{\mathbf{A}}_{3 \rightarrow 1}$ , given in Equation (4.18). In particular, the zero columns of matrix  $\tilde{\mathbf{A}}_{3 \rightarrow 1}$  correspond to nodes of Fig. 4.5d with no outgoing edges.*

The following result follows from the construction of  $D_l(\mathcal{S})$  from  $D_f(\mathcal{S})$ .

**Lemma 8.** *Consider a state variable  $v \in V_f$  of  $D_f(\mathcal{S})$ , for a LETS  $\mathcal{S}$ , updated in layer  $L_k, k = \{1, 2, \dots, J\}$ . If all the outgoing edges of  $v$  are connected to nodes*



**Figure 4.5:** The steps of constructing the digraph  $D_l(\mathcal{S})$  of the  $(5, 3)$  LETS for a layered decoder from the flooding digraph  $D_f(\mathcal{S})$ . (The colors (edge types) red (dotted), blue (dashed) and purple (dash-dotted) represent  $L_1$ ,  $L_2$  and  $L_3$ , respectively.)

from subsequent layers  $L_j$ ,  $k < j \leq J$ , the corresponding node  $v \in V_l$  of  $D_l(\mathcal{S})$  will not have any outgoing edges. The total number of nodes  $n_z$  in  $D_l(\mathcal{S})$  that have this property are greater than or equal to the number of nodes updated in the first layer, i.e.,  $n_z \geq n_{L_1}$ .

The following lemma applies to LETSs with irreducible transition matrices  $\mathbf{A}$  for flooding schedule, and follows from the construction of  $D_l(\mathcal{S})$  from  $D_f(\mathcal{S})$  and the fact that for such LETSs the digraph  $D_f(\mathcal{S})$  is strongly connected.

**Lemma 9.** *Consider a LETS  $\mathcal{S}$  with an irreducible transition matrix  $\mathbf{A}$  for flooding schedule. If after the construction of  $D_l(\mathcal{S})$  from  $D_f(\mathcal{S})$ , all the nodes with no outgoing edges, and their incoming edges are removed from  $D_l(\mathcal{S})$ , then the remaining digraph (if any) is strongly connected.*

Lemma 9 implies that the FNF of the transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  of a layered decoder for a LETS with irreducible matrix  $\mathbf{A}$  has at most one irreducible diagonal block. The following Lemma proves that the number of irreducible blocks is exactly one.

**Lemma 10.** *The FNF of a layered decoder transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  corresponding to a LETS,  $\mathcal{S}$ , with irreducible flooding transition matrix,  $\mathbf{A}$ , has one and only one irreducible diagonal block.*

*Proof.* From Proposition 1, we have  $\rho(\tilde{\mathbf{A}}_{J \rightarrow 1}) \geq \rho(\mathbf{A})$ . Moreover, since  $\mathbf{A}$  is irreducible,  $\rho(\mathbf{A}) > 1$  [45]. This means  $\rho(\tilde{\mathbf{A}}_{J \rightarrow 1}) > 1$ , and thus the FNF of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  must have at least one irreducible diagonal block. This together with Lemma 9 completes the proof. ■

Based on Lemma 10, there exists a permutation matrix  $\mathbf{P}$  such that the FNF of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  can be written as follows

$$\mathbf{P}\tilde{\mathbf{A}}_{J \rightarrow 1}\mathbf{P}^T = \left[ \begin{array}{c|c} \mathbf{0}_{n_z \times n_z} & \tilde{\mathbf{A}}'_{n_z \times (m_s - n_z)} \\ \hline \mathbf{0}_{(m_s - n_z) \times n_z} & \tilde{\mathbf{A}}_{(m_s - n_z) \times (m_s - n_z)} \end{array} \right], \quad (4.24)$$

where  $\tilde{\mathbf{A}}$  is a non-negative irreducible matrix and  $n_z$  is defined in Lemma 8.

**Theorem 6.** *Let  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  be the non-negative transition matrix of a LETS, which is not a simple cycle, in a layered decoder. Also, let the matrix  $\tilde{\mathbf{A}}$  be the only irreducible diagonal block of the FNF of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ . Then,*

- (a)  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  has a simple positive eigenvalue,  $\tilde{r} > 1$ , equal to its spectral radius  $\rho(\tilde{\mathbf{A}}_{J \rightarrow 1})$ .
- (b) To  $\tilde{r}$ , there corresponds a non-negative left eigenvector  $\tilde{\mathbf{w}}_1^T$  that is equal to the positive left eigenvector of  $\tilde{\mathbf{A}}$ , denoted by  $\tilde{\omega}_1^T$ , appended by  $n_z$  zeros.
- (c) To  $\tilde{r}$ , there corresponds a non-negative right eigenvector  $\tilde{\mathbf{u}}_1$  that is equal to the positive right eigenvector of  $\tilde{\mathbf{A}}$ , denoted by  $\tilde{\nu}_1$ , appended by the vector  $\frac{\tilde{\mathbf{A}}'}{\tilde{r}}\tilde{\nu}_1$ , in which the matrix  $\tilde{\mathbf{A}}'$  is defined in (4.24).

*Proof.* (a) The eigenvalues of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  are the roots of  $\det(\tilde{\mathbf{A}}_{J \rightarrow 1} - \mu\mathbf{I})$ , which based on (4.24) simplifies to  $\det(\tilde{\mathbf{A}} - \mu\mathbf{I})\mu^{n_z}$ . The eigenvalues of irreducible matrix  $\tilde{\mathbf{A}}$  together with  $n_z$  zeros are thus the eigenvalues of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ . Since, for a LETS that is not a simple cycle,  $\tilde{\mathbf{A}}$  is a non-negative irreducible matrix, its dominant eigenvalue,  $\tilde{r}$ , which is positive and simple, is the dominant eigenvalue of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  as well. Based on the proof of Lemma 10, this eigenvalue is larger than one, i.e.,  $\tilde{r} > 1$ .

(b) Based on (4.24), to find the left eigenvector,  $\tilde{\mathbf{w}}_1^T$ , corresponding to  $\tilde{r}$ , we need to solve

$$\tilde{\mathbf{w}}_1^T \left[ \begin{array}{c|c} \mathbf{0}_{n_z \times n_z} & \tilde{\mathbf{A}}'_{n_z \times (m_s - n_z)} \\ \hline \mathbf{0}_{(m_s - n_z) \times n_z} & \tilde{\mathbf{A}}_{(m_s - n_z) \times (m_s - n_z)} \end{array} \right] = \tilde{r} \tilde{\mathbf{w}}_1^T. \quad (4.25)$$

The solution to the above equation is a vector of the form

$$\tilde{\mathbf{w}}_1^T = \left[ \begin{array}{c|c} \mathbf{0}_{1 \times n_z} & \tilde{\omega}_1^T \end{array} \right], \quad (4.26)$$

where  $\tilde{\omega}_1^T$  is the positive left eigenvector of  $\tilde{\mathbf{A}}$  corresponding to  $\tilde{r}$ .

(c) Similar to the proof of part (b), it can be easily shown that the right eigenvector of the FNF of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  is

$$\tilde{\mathbf{u}}_1 = \left[ \begin{array}{c} \frac{\tilde{\mathbf{A}}'}{\tilde{r}}\tilde{\nu}_1 \\ \tilde{\nu}_1 \end{array} \right]. \quad (4.27)$$

■

**Example 10.** The matrices  $\tilde{\mathbf{A}}$  and  $\tilde{\mathbf{A}}'$  in the FNF of the transition matrix of the

(5, 3) LETS of the Tanner code for the layered decoder are

$$\tilde{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \tilde{\mathbf{A}}' = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The dominant eigenvalue of  $\tilde{\mathbf{A}}$  is  $\tilde{r} = 2.0136$ , which is also equal to  $\rho(\tilde{\mathbf{A}}_{3 \rightarrow 1})$ . Moreover, the corresponding dominant left and right eigenvectors of the FNF of  $\tilde{\mathbf{A}}_{3 \rightarrow 1}$  are

$$\tilde{\mathbf{w}}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.2838 \\ 0.4027 \\ 0.2525 \\ 0.3189 \\ 0.4525 \\ 0.5085 \\ 0.3584 \end{bmatrix}, \tilde{\mathbf{u}}_1 = \begin{bmatrix} 0.3342 \\ 0.2355 \\ 0.2096 \\ 0.2974 \\ 0.3756 \\ 0.2647 \\ 0.4220 \\ 0.2974 \\ 0.5329 \\ 0.3756 \\ 0.3342 \\ 0.4743 \end{bmatrix}.$$

The lower partition of the left and right eigenvectors are equal to  $\tilde{\boldsymbol{\omega}}_1$  and  $\tilde{\boldsymbol{\nu}}_1$ , the left and right eigenvectors of  $\tilde{\mathbf{A}}$ , respectively. Moreover, the upper partition of  $\tilde{\mathbf{w}}_1$  is extended by  $n_z = 5$  zeros while the upper part of  $\tilde{\mathbf{u}}_1$  is equal to  $\frac{\tilde{\mathbf{A}}'}{2.0136} \tilde{\boldsymbol{\nu}}_1$ .

In the following, to complement the result of Theorem 6, we discuss the case of LETSs that are simple cycles.

**Theorem 7.** *Consider a simple cycle  $\mathcal{S}$  of length  $2a$ , and let  $\mathbf{A}$  and  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  denote the non-negative transition matrices of  $\mathcal{S}$  for flooding and layered decoders, respectively. Then*

- (a) *The FNF of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  has two irreducible diagonal blocks.*
- (b)  *$\tilde{r} = 1$  is an eigenvalue of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ , whose right eigenvector  $\tilde{\mathbf{u}}_1$  is the same as the right eigenvector of  $\mathbf{A}$ , and whose left eigenvector,  $\tilde{\mathbf{w}}_1^T$ , is an all-one vector of length  $2a - n_z$  appended by  $n_z$  zeros (up to a permutation), where  $n_z$  is the number of zero columns of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ .*

*Proof.* (a) The flooding digraph  $D_f(\mathcal{S})$  of the simple cycle  $\mathcal{S}$  consists of two disconnected directed cycles in opposite directions [45]. By the construction of  $D_l(\mathcal{S})$  from  $D_f(\mathcal{S})$ , each of the directed cycles will be transformed into a new directed cycle, that is strongly connected, and thus corresponds to an irreducible diagonal block in the FNF of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ . In fact, the FNF of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  has the following structure

$$\left[ \begin{array}{c|cc} \mathbf{0} & & \tilde{\mathbf{A}}' \\ \hline \mathbf{0} & \tilde{\mathbf{A}}_{1(a-n_{z_1}) \times (a-n_{z_1})} & \mathbf{0} \\ \hline & \mathbf{0} & \tilde{\mathbf{A}}_{2(a-n_{z_2}) \times (a-n_{z_2})} \end{array} \right]_{2a \times 2a},$$

in which  $\tilde{\mathbf{A}}_1$  and  $\tilde{\mathbf{A}}_2$  are the irreducible blocks corresponding to the two directed cycles in opposite directions.

(b) It is known that for a simple cycle  $\mathcal{S}$ , the dominant eigenvalue of  $\mathbf{A}$  is  $r = 1$  with multiplicity 2 [45]. It can be seen that Equation (4.22) is also applicable to simple cycles. Replacing  $r = 1$  in (4.22) indicates that a right eigenvector  $\mathbf{u}_1$  of  $\mathbf{A}$  corresponding to  $r = 1$  is also a right eigenvector of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  corresponding to  $\tilde{r} = 1$ . In fact, the eigenvalue  $\tilde{r} = 1$  is the dominant eigenvalue of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  with multiplicity 2. (Each of the diagonal blocks  $\tilde{\mathbf{A}}_1$  and  $\tilde{\mathbf{A}}_2$  has one dominant eigenvalue  $\tilde{r} = 1$ .) The all-one vectors with sizes  $a - n_{z_1}$  and  $a - n_{z_2}$  are the left eigenvectors of  $\tilde{\mathbf{A}}_1$  and  $\tilde{\mathbf{A}}_2$ , respectively [45]. It is then easy to see that an all-one vector with size  $2a - n_{z_1} - n_{z_2}$  appended by  $n_z = n_{z_1} + n_{z_2}$  zeros is a left eigenvector of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  (up to a permutation). ■

### 4.1.6 Computing the Failure Probability of LETSs in Layered Decoders

As the error indicator function, we consider the projection of the state vector, given in (4.8), onto the non-negative left eigenvector  $\tilde{\mathbf{w}}_1^T$  of  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  corresponding to the dominant eigenvalue  $\tilde{r}$ :

$$\tilde{\beta}^{(\ell)} \triangleq \tilde{\mathbf{w}}_1^T \tilde{\mathbf{x}}^{(\ell)} = \gamma_{ch}^T \mathbf{L} + \sum_{i'=1}^{\ell} \left( \overset{\triangleleft}{\gamma}_{ex}^{(i')T} \mathbf{L}_{ex}^{(i'-1)} + \overset{\triangleright}{\gamma}_{ex}^{(i')T} \mathbf{L}_{ex}^{(i')} \right), \quad (4.28)$$

in which the vectors  $\gamma_{ch}^T$ ,  $\overset{\triangleleft}{\gamma}_{ex}^{(i')T}$  and  $\overset{\triangleright}{\gamma}_{ex}^{(i')T}$  with lengths equal to  $a$ ,  $b$  and  $b$ , respectively, are defined by

$$\gamma_{ch}^T = \tilde{\mathbf{w}}_1^T \sum_{i=1}^{\ell} \left( \prod_{j=i+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j)} \right) \tilde{\mathbf{B}}^{(i)}, \quad (4.29)$$

$$\overset{\triangleleft}{\gamma}_{ex}^{(i')T} = \tilde{\mathbf{w}}_1^T \left( \prod_{j'=i'+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j')} \right) \tilde{\mathbf{B}}_{ex}^{(i')} \quad i' = 1, \dots, \ell, \quad (4.30)$$

$$\overset{\triangleright}{\gamma}_{ex}^{(i')T} = \tilde{\mathbf{w}}_1^T \left( \prod_{j'=i'+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j')} \right) \tilde{\mathbf{B}}_{ex}^{(i')} \quad i' = 1, \dots, \ell, \quad (4.31)$$

respectively. The mean and variance of  $\tilde{\beta}^{(\ell)}$  are calculated as

$$\mathbb{E}[\tilde{\beta}^{(\ell)}] = (2/\sigma_{ch}^2) \sum_{k=1}^a (\gamma_{ch}^T)_k + \sum_{i'=1}^{\ell-1} \left( \overset{\triangleleft}{\gamma}_{ex}^{(i'+1)T} + \overset{\triangleright}{\gamma}_{ex}^{(i')T} \right) \mathbf{m}_{ex}^{(i')} + \overset{\triangleright}{\gamma}_{ex}^{(\ell)T} \mathbf{m}_{ex}^{(\ell)}, \quad (4.32)$$

and

$$\begin{aligned} \text{VAR}[\tilde{\beta}^{(\ell)}] &= (4/\sigma_{ch}^2) \sum_{k=1}^a (\gamma_{ch}^T)_k^2 \\ &+ \sum_{i'=1}^{\ell-1} \left( \overset{\triangleleft}{\gamma}_{ex}^{(i'+1)T} + \overset{\triangleright}{\gamma}_{ex}^{(i')T} \right) \Sigma_{ex}^{(i')} \left( \overset{\triangleleft}{\gamma}_{ex}^{(i'+1)} + \overset{\triangleright}{\gamma}_{ex}^{(i')} \right) + \overset{\triangleright}{\gamma}_{ex}^{(\ell)T} \Sigma_{ex}^{(\ell)} \overset{\triangleright}{\gamma}_{ex}^{(\ell)}, \end{aligned} \quad (4.33)$$

respectively. The symbol  $(\cdot)_k$  is used to represent the  $k$ th element of the vector inside the parentheses. The  $b \times 1$  vector  $\mathbf{m}_{ex}^{(i')}$  and the  $b \times b$  matrix  $\Sigma_{ex}^{(i')}$  are the mean and the covariance matrix of the inputs from the unsatisfied CNs at iteration  $i'$ , respectively. Since we assume that external inputs to the LETS are independent, the matrix  $\Sigma_{ex}^{(i')}$

is diagonal. The mean and the variance of the inputs from unsatisfied CNs are calculated using DE. Also, the missatisfied CN gains are calculated based on DE, as explained in Subsection 4.1.4. These gains are utilized in iteration dependent gain matrices involved in Equations (4.4) to (4.7). Finally, the vector  $\boldsymbol{\gamma}_{ch}^T$  in (4.29) as well as the set of vectors  $\boldsymbol{\gamma}_{ex}^{\triangleleft(i')T}$  and  $\boldsymbol{\gamma}_{ex}^{\triangleright(i')T}$  for  $i' = 1, \dots, \ell$  in (4.30) and (4.31), respectively, are used in Equations (4.32) and (4.33) to find the mean and the variance of the error indicator function, respectively. Finally, assuming that  $\tilde{\beta}^{(\ell)}$  is Gaussian, the failure probability of a LETS,  $\mathcal{S}$ , is obtained by

$$P_e(\mathcal{S}) = \lim_{\ell \rightarrow \infty} Pr\{\tilde{\beta}^{(\ell)} < 0\} = \lim_{\ell \rightarrow \infty} Q\left(\frac{\mathbb{E}[\tilde{\beta}^{(\ell)}]}{\sqrt{\text{VAR}[\tilde{\beta}^{(\ell)}]}}\right). \quad (4.34)$$

In practice, the error probability, given above, converges rather fast within a few iterations.

To estimate the error floor of an LDPC code, we partition the dominant LETSs of the code in accordance with their TSLP. This implies that within each class of LETSs, we first identify different non-isomorphic structures and then among those LETSs with the same structure, we identify different TSLPs. We then accordingly partition the class into different *groups*, where all the LETSs within a group have the same TSLP. Using the union bound, we then approximate the error floor of the code by

$$P_f \approx \sum_i \Upsilon_i P_e(\mathcal{S}_i), \quad (4.35)$$

where  $\mathcal{S}_i$  is the representative of the  $i$ th TS group with the size  $\Upsilon_i$ .

We recall that for a LETS  $\mathcal{S}$  with an irreducible flooding transition matrix  $\mathbf{A}$ , all the elements of the left eigenvector of  $\mathbf{A}$  corresponding to the dominant eigenvalue  $r$  are positive. For the layered transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ , however, there are some zeros in the left eigenvector  $\tilde{\mathbf{w}}_1^T$  corresponding to the dominant eigenvalue  $\tilde{r}$ . This implies that the corresponding state variables have no effect on the value of the error indicator function  $\tilde{\beta}^{(\ell)}$ . In fact, only the state variables corresponding to the irreducible part of the transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  are the ones that determine the value of  $\tilde{\beta}^{(\ell)}$ , and are the main contributors to the growth of erroneous messages inside  $\mathcal{S}$ .

## 4.2 The Effect of Row Block Permutations on the Failure Probability of a LETS: Analysis and Optimization

### 4.2.1 Effect of Different Row Block Permutations on the Dominant Eigenvalue of $\tilde{\mathbf{A}}_{J \rightarrow 1}$

In the asymptotic regime of  $\ell \rightarrow \infty$ , the dominant eigenvalue  $\tilde{r}$  of the transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  plays an important role in the failure probability of a LETS in a layered decoder. The value of  $\tilde{r}$  can however change depending on the order in which different layers within the LETS are updated. To simplify the discussions, we assume that in the layered decoding, the row blocks of the parity check matrix are updated based on their increased indices, i.e., starting from the first row block all the way down to the row block number  $m_b$ . We thus associate the different orderings of row updates with different permutations of the row blocks of the parity-check matrix. In this part, we consider the effect of different row block permutations on  $\tilde{r}$ .

In general, for a LETS with  $J$  layers, there are  $J!$  different layer permutations. We denote the set of all possible permutations of  $J$  layers by  $\Pi_J$ . A given permutation  $\pi \in \Pi_J$  is identified by the sequence  $\pi(1), \dots, \pi(J)$ , or more briefly by  $\pi_1, \dots, \pi_J$ .

**Proposition 2.** *Consider the application of a permutation  $\pi$  to the  $J$  layers of a LETS  $\mathcal{S}$ , and denote the corresponding transition matrix of  $\mathcal{S}$  in the layered decoder by  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$ . The dominant eigenvalue of  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$  is then invariant to any cyclic shift of the permutation  $\pi$ .*

*Proof.* Suppose that the right eigenvector of  $\tilde{r}$  for  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$  is  $\tilde{\mathbf{u}}_1$ . We then have

$$\underbrace{\mathcal{A}_{\pi_J} \mathcal{A}_{\pi_{J-1}} \dots \mathcal{A}_{\pi_1}}_{\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}} \tilde{\mathbf{u}}_1 = \tilde{r} \tilde{\mathbf{u}}_1.$$

Multiplying both sides of this equation sequentially with  $\mathcal{A}_{\pi_1}, \mathcal{A}_{\pi_2}, \dots, \mathcal{A}_{\pi_{J-1}}$ , results

in

$$\begin{aligned}
 \underbrace{\mathcal{A}_{\pi_1} \mathcal{A}_{\pi_J} \cdots \mathcal{A}_{\pi_2}}_{1 \text{ cyclic shift}} (\mathcal{A}_{\pi_1} \tilde{\mathbf{u}}_1) &= \tilde{r}(\mathcal{A}_{\pi_1} \tilde{\mathbf{u}}_1), \\
 \underbrace{\mathcal{A}_{\pi_2} \mathcal{A}_{\pi_1} \mathcal{A}_{\pi_J} \cdots \mathcal{A}_{\pi_3}}_{2 \text{ cyclic shifts}} (\mathcal{A}_{\pi_2} \mathcal{A}_{\pi_1} \tilde{\mathbf{u}}_1) &= \tilde{r}(\mathcal{A}_{\pi_2} \mathcal{A}_{\pi_1} \tilde{\mathbf{u}}_1), \\
 &\vdots \\
 \underbrace{\mathcal{A}_{\pi_{J-1}} \cdots \mathcal{A}_{\pi_1} \mathcal{A}_{\pi_J}}_{J-1 \text{ cyclic shifts}} (\mathcal{A}_{\pi_{J-1}} \cdots \mathcal{A}_{\pi_1} \tilde{\mathbf{u}}_1) &= \tilde{r}(\mathcal{A}_{\pi_{J-1}} \cdots \mathcal{A}_{\pi_1} \tilde{\mathbf{u}}_1),
 \end{aligned}$$

respectively. As can be seen,  $\tilde{r}$  is the eigenvalue of all the transition matrices resulted from different cyclic shifts of the permutation  $\pi$ . The above equations can be written for all the eigenvalues of  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$ , and thus,  $\tilde{r}$  remains the dominant eigenvalue of all such transition matrices.  $\blacksquare$

In the following proposition, whose proof is given in the appendix 4.4, we prove that reversing the layer permutation does not change  $\tilde{r}$ .

**Proposition 3.** *Consider the transition matrix  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$  of a LETS with  $J$  layers corresponding to a layer permutation  $\pi$ . Then, the eigenvalues of  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$  are invariant under the reversing of  $\pi$ , i.e., the transition matrix  $\tilde{\mathbf{A}}_{\pi_1 \rightarrow \pi_J}$  corresponding to the reverse permutation  $\pi_J, \dots, \pi_1$  has the same eigenvalues as  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$ .*

Based on Propositions 2 and 3, we have the following result.

**Corollary 1.** *For a LETS with  $J$  layers ( $J \geq 3$ ), the number of distinct dominant eigenvalues of the layered transition matrix corresponding to different permutations of the layers is upper bounded by  $\frac{(J-1)!}{2}$ .*

Corollary 1 implies that for a LETS with three layers ( $J = 3$ ), all the layered transition matrices corresponding to different layer permutations have the same dominant eigenvalue.

**Example 11.** *The Tanner (155, 64) LDPC code has 3 row layers, and based on Corollary 1, the dominant eigenvalue of the layered transition matrices of all of its LETSs are invariant under all the possible (6) layer permutations.*

### 4.2.2 Optimizing the Row Layered Schedule

The contribution of each LETS  $\mathcal{S}$  to the error floor of an LDPC code, decoded by a row layered iterative algorithm, depends on the distribution of messages entering  $\mathcal{S}$  through its missatisfied and unsatisfied CNs as well as the internal structure of  $\mathcal{S}$ . In the linear state-space model used in this work, the internal structure is reflected in the system matrices in general, and the transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ , in particular. More specifically, our results show that the dominant eigenvalue  $\tilde{r}$  of the transition matrix plays an important role in the growth rate of erroneous messages inside the subgraph of  $\mathcal{S}$ , with larger values of  $\tilde{r}$  generally corresponding to larger growth rate. As we discussed in Subsection 4.2.1, however, the transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  and  $\tilde{r}$  can change with layer permutations. Moreover, different layer permutations can result in notable change in the TSLP and thus the distributions of the external messages entering the TS. As a result, different row layered schedules can potentially produce error floors that are considerably different. This motivates the search for layer permutations which result in low error floors. In the following, we show that our proposed model can be used not only for the error floor estimation of layered decoders, but also as an efficient tool to find row layered schedules with low error floors.

Generally, for a given QC-LDPC code whose base matrix  $\mathbf{H}_b$  has  $m_b$  rows, there are  $m_b!$  different row layered schedules corresponding to different row permutations of  $\mathbf{H}_b$ . The complexity of an exhaustive search among all such schedules based on the exact estimation of the error floor can be prohibitive for relatively large values of  $m_b$ . The main source of complexity in our model is to obtain the distribution of external messages by DE. (Recall that, for each iteration,  $2|E_b|$  distributions are needed to be calculated, where  $|E_b|$  is the number of edges in the base graph, which is equal to the number of nonzero elements of  $\mathbf{H}_b$ .) To simplify the search among the  $m_b!$  different row layered schedules, rather than the derivation of such distributions for each schedule, we select one schedule, say the one corresponding to the original order of the rows, and then at each iteration  $\ell$  and for each layer  $j$ , we derive the average distributions of CN to VN messages and VN to CN messages, denoted by  $\overleftarrow{\psi}_\ell^j$  and  $\overrightarrow{\psi}_\ell^j$ , respectively, where the average is taken over all the corresponding distributions within layer  $L_j$ . These average distributions are then used to represent all the CN to VN and VN to CN distributions in the  $j$ th layer of decoding regardless of the schedule. As an example, for the base graph of Fig. 4.2, the distribution  $\overleftarrow{\psi}_\ell^3$  is the

average of  $\psi_\ell^{[2\leftarrow 3]}$  and  $\psi_\ell^{[4\leftarrow 3]}$ , and  $\overrightarrow{\psi}_\ell^3$  is the average of  $\psi_\ell^{[2\rightarrow 3]}$  and  $\psi_\ell^{[4\rightarrow 3]}$ .

The averaging process just explained will result in a less accurate estimate of the failure rate of a LETS. In fact, by ignoring the effect of scheduling in the distribution of external messages and by the averaging, we only observe the effect of the schedule on the internal messages of the LETS. To find the schedule with the lowest error floor, we then perform the search in two steps. In the first step, we use the above approximation/simplification and search among all the  $m_b!$  schedules to find a few candidates that have lower error floors. In the second step, we examine the candidate schedules by calculating the error floor estimates accurately (by considering the effect of scheduling in the distribution of external messages), and find the one with the lowest error floor.

### 4.3 Simulation Results and Discussions

In this section, we investigate the accuracy of the proposed linear state-space model in estimating the error floor of row layered SPA through simulations. We consider two QC-LDPC codes  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , whose exponent matrices are shown in Figs. 4.6a and 4.6b, respectively.  $\mathcal{C}_1$  is a (640, 192) variable-regular code with  $d_v = 5$  and irregular CN degrees [3] and  $\mathcal{C}_2$  is a (576, 432) irregular code used in Wimax standard [82]. The lifting degrees for the two codes are 64 and 24, respectively.

The Monte Carlo simulation results as well as estimation results for the layered decoding of  $\mathcal{C}_1$  where the order of row layers are the same as that of Fig. 4.6a, for different saturation levels are presented in Fig. 4.7. The most harmful structure of this code in the error floor region is the (5, 5) LETS, shown in Fig. 4.8, with multiplicity 64. All the (5, 5) LETSs have the same TSLP. As expected, by increasing the saturation level, the error floor is reduced. The figure also shows a good match between the linear model estimation results and simulation results. The slight over-estimation of error floor is attributed to the linear approximation of missatisfied CN operations (see [48], for more information).

We now investigate the effect of row block permutations on the error floor of  $\mathcal{C}_1$ .  $\mathcal{C}_1$  has 7 row layers, each corresponding to one of the row blocks of the parity-check matrix. We label these row blocks with numbers 1 to 7 based on their indices in Fig. 4.6a, i.e., the first row block in Fig. 4.6a is labeled by 1, the second by 2, and so on.

8	-1	26	62	-1	-1	59	19	-1	60
51	9	50	-1	39	-1	4	-1	25	26
-1	32	10	7	56	52	41	55	61	41
46	11	-1	43	-1	63	8	51	37	-1
47	7	7	50	49	53	-1	12	-1	-1
31	-1	-1	-1	31	38	-1	-1	23	48
-1	25	21	56	59	30	27	23	27	18

(a) Exponent matrix of  $\mathcal{C}_1$ : (640, 192) QC-LDPC code with lifting degree 64.

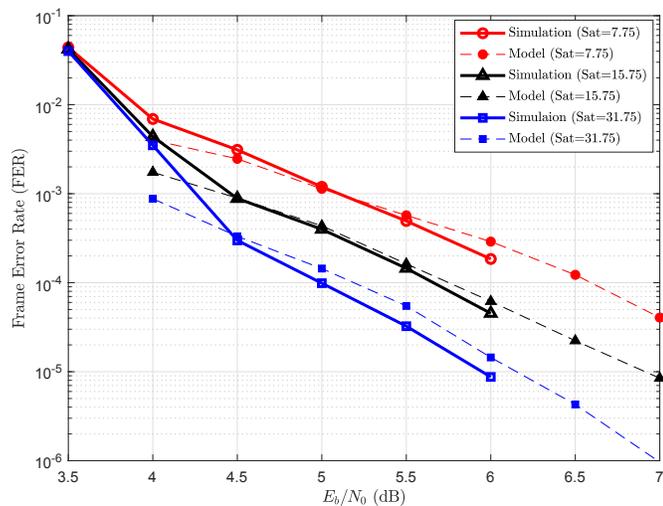
-1	20	-1	7	-1	-1	3	6	4	-1	-1	21	7	13	19	23	5	23	0	0	-1	-1	-1	-1
10	-1	3	17	8	-1	-1	-1	-1	17	10	2	9	10	8	14	9	6	-1	0	0	-1	-1	-1
-1	-1	5	-1	-1	15	9	-1	17	16	-1	9	1	18	11	7	15	1	20	-1	0	0	-1	-1
16	0	-1	-1	15	-1	-1	0	12	-1	20	3	23	2	21	9	3	4	-1	-1	-1	0	0	-1
-1	13	15	20	-1	6	18	-1	-1	-1	-1	21	19	0	0	18	15	6	-1	-1	-1	-1	0	0
19	-1	-1	-1	3	7	-1	8	-1	18	7	17	21	21	6	16	2	22	0	-1	-1	-1	-1	0

(b) Exponent matrix of  $\mathcal{C}_2$ : (576, 432) Wimax QC-LDPC code with lifting degree 24.

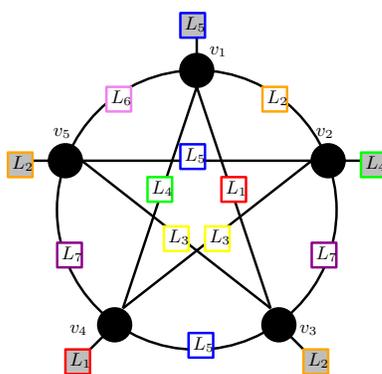
**Figure 4.6:** QC-LDPC codes used for simulations. The entries of the matrices, that are not equal to  $-1$ , represent the right circular shift of the identity matrix to create the corresponding block of the parity-check matrix. The  $-1$  entries represent zero blocks.

We then represent different schedules with different permutations of numbers from 1 to 7. For example, permutation (1, 2, 3, 4, 5, 6, 7) corresponds to a schedule which updates the row layers in the same order as they appear in Fig. 4.6a.

In order to investigate the effect of different row schedules on the error floor performance of  $\mathcal{C}_1$ , the technique of Section 4.2.2 is used. The total number of possible row permutations for  $\mathcal{C}_1$  is  $7! = 5040$ . These permutations in general correspond to different system matrices for the (5, 5) LETS. By using a single application of DE, as discussed in Section 4.2.2, the failure rate of the (5, 5) LETS for different schedules is approximated. The results for SNR of 6 dB and saturation level of 31.75 are provided in Fig. 4.9 for all the schedules. As can be seen, the estimation results are partitioned into 8 groups, separated by vertical dotted lines. The transition matrices of the schedules within each group have the same dominant eigenvalue  $\tilde{r}$ . We have also sorted the groups according to the increasing value of  $\tilde{r}$ . The eight different values of  $\tilde{r}$  are shown in Fig. 4.9, and range from 12.402 to 16.125. Note that based on Corollary 1, the upper bound on the number of different  $\tilde{r}$  values for different



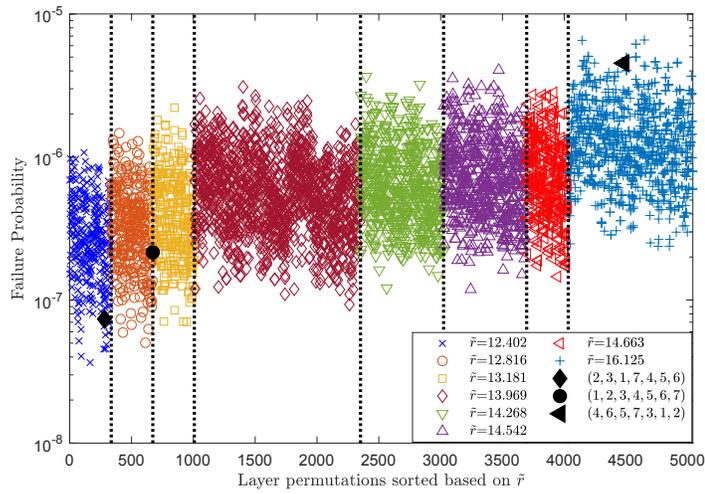
**Figure 4.7:** Simulation and estimation results of  $\mathcal{C}_1$  for different saturation levels. The maximum number of iterations  $I_{max} = 30$ .



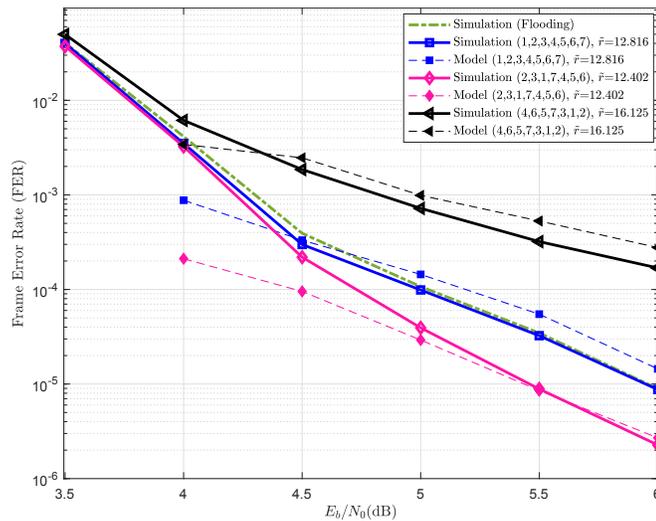
**Figure 4.8:** The (5,5) LETS structure of  $\mathcal{C}_1$  in which the row layers for different CNs are shown.

schedules for the case where the code has 7 layers is 360, in general. Fig. 4.9 shows the trend that increasing  $\tilde{r}$ , on average, increases the failure probability of the LETS. Within each group, however, the variance of the error probabilities is still rather large. This implies that while  $\tilde{r}$  plays an important role in the failure probability of a LETS, there are also other factors, including the layering structure of the TS reflected through the system matrices, that affect the harmfulness. (Note that, in this analysis, although the distributions of the external messages of the LETS for different layers, obtained through DE, remains constant for different schedules, but the assignment of different CNs and VNs of the LETS to different layers will change due to the change of schedule. As a result, the distributions of messages associated with these nodes will also change in different schedules.)

To find a schedule with low error floor, in the next step, we select a few schedules whose transition matrices have the minimum dominant eigenvalue,  $\tilde{r} = 12.402$ , and result in the lowest error rates in the first step. We then apply our estimation technique accurately to find the error floor of these candidate schedules, and select the one that has the lowest error floor. As a result, we obtain the schedule corresponding to the permutation  $(2, 3, 1, 7, 4, 5, 6)$ , shown in Fig. 4.9 with a full diamond. For comparison, we have also selected one of the schedules with the worst error floor,  $(4, 6, 5, 7, 3, 1, 2)$ , as well as the original one,  $(1, 2, 3, 4, 5, 6, 7)$ . These schedules are specified in Fig. 4.9 by a full triangle and a full circle, respectively. The simulation and estimation results of these three schedules are presented in Fig. 4.10. The maximum number of iterations and the saturation level are  $I_{max} = 30$  and  $31.75$ , respectively. For comparison, we have also included the FER of the flooding schedule with maximum number of iterations  $I_{max} = 60$  in Fig. 4.10. As can be seen, all the estimation results match closely with the corresponding simulations. Remarkably, there is a substantial difference between the FER of the best and worst layered schedules in the error floor region, with the performance of flooding schedule in the middle. This demonstrates the gain that one can obtain in performance by properly choosing the updating order of layers in a layered decoder, basically at no cost. It also shows that a layered decoder can, in general, have a better or a worse performance compared to its flooding counterpart. By proper permutation of row layers, the layered decoder not only has a faster convergence speed compared to a flooding decoder but also can have a better performance.



**Figure 4.9:** The approximate estimate of the failure probability of the (5, 5) LETS of  $\mathcal{C}_1$  for various row layered schedules at  $E_b/N_0 = 6$  dB and saturation level 31.75. The schedules are sorted based on  $\tilde{r}$ .



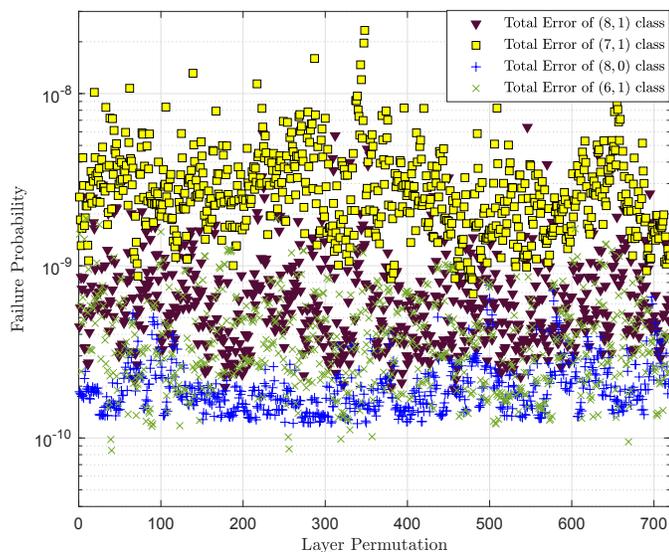
**Figure 4.10:** The simulation and estimation results of  $\mathcal{C}_1$  for different row schedules. (The saturation level is 31.75, and the maximum number of iterations for layered schedules and the flooding schedule are set to 30 and 60, respectively.)

**Table 4.1:** Multiplicities of  $(a, b)$  LETSs of  $\mathcal{C}_2$  within the range  $a \leq 8$  and  $b \leq 2$ 

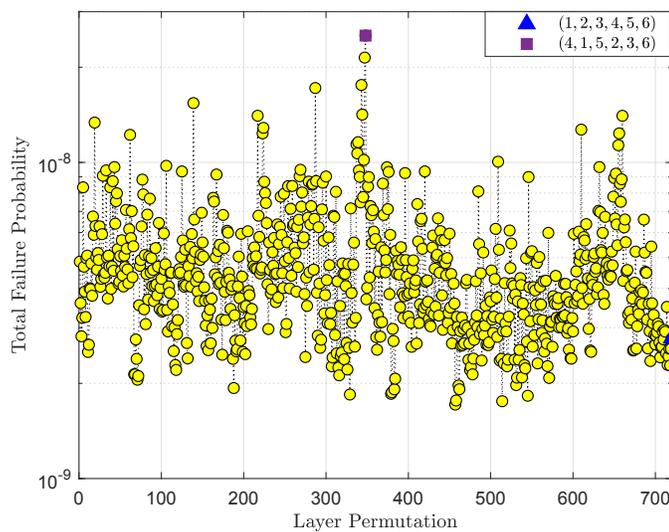
$(a, b)$	(4, 2)	(5, 2)	(6, 1)	(6, 2)	(7, 1)	(7, 2)	(8, 0)	(8, 1)	(8, 2)
Multiplicity	144	216	48	1068	240	3600	48	720	13464

As the next example, we consider  $\mathcal{C}_2$ . The multiplicity of different  $(a, b)$  LETSs of this code within the range  $a \leq 8$  and  $b \leq 2$  are listed in Table 4.1. These LETSs have been found using the exhaustive search algorithm of [33]. As can be seen, this code has a variety of LETSs that can potentially contribute to the error floor performance. Moreover, unlike the previous example, there are different non-isomorphic structures within each class of TSs. For example, there are 8 different non-isomorphic  $(7, 1)$  LETS structures in this code. For 2 out of 8 structures, the code also contains TSs with two different TSLPs. This means that there are ten  $(7, 1)$  LETSs, each with multiplicity of 24, that can have different failure probabilities under a layered decoder.

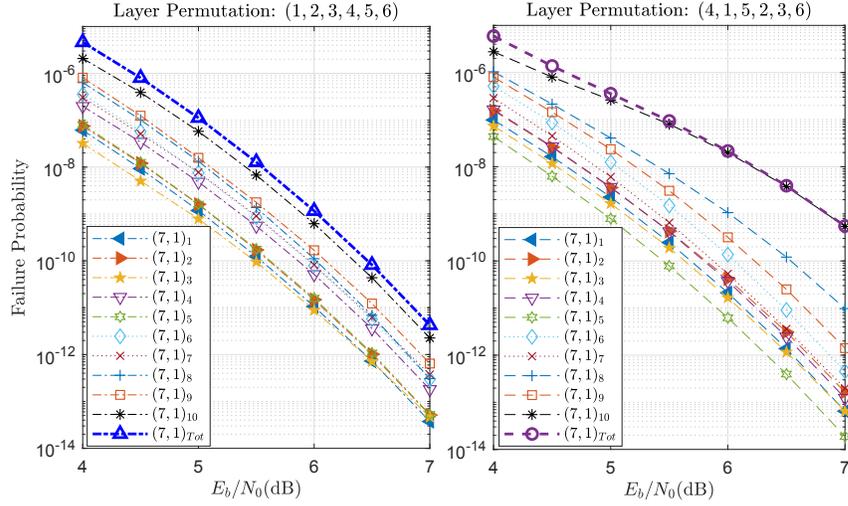
To evaluate the effect of row scheduling on the error floor of  $\mathcal{C}_2$ , the same general technique as the one employed in the previous example is used. In this regard, for each of the LETS groups (those with the same structure and TSLP) within  $(6, 1)$ ,  $(7, 1)$ ,  $(8, 0)$  and  $(8, 1)$  classes, we estimate the failure probability of various schedules. This is performed based on the approximate average DE method of Subsection 4.2.2 at  $E_b/N_0 = 6$  dB and for a saturation level of 15.75. The number of LETS groups within each of the aforementioned classes are 2, 10, 2 and 30, respectively. Each group has the same size of 24. To estimate the contribution of each LETS class to the error floor, we first estimate the failure rate of a member of each LETS group within the class, then multiply the result by 24, and finally add up the results for different groups within the class. These results for different classes are presented in Fig. 4.11, for all the  $6! = 720$  possible row schedules. As can be observed, on average, the most harmful LETS class of this code, for saturation level 15.75, is the  $(7, 1)$  class. In order to observe the overall error probability for different row schedules, the contribution of different classes from Fig. 4.11 are added and presented in Fig. 4.12. Interestingly, for  $\mathcal{C}_2$ , it appears that the original schedule  $(1, 2, 3, 4, 5, 6)$  has one of the lowest error rates. This schedule is shown in Fig. 4.12 by a full triangle. The worst schedule in Fig. 4.12 is  $(4, 1, 5, 2, 3, 6)$ , and is identified by a full square.



**Figure 4.11:** The effect of various row layered schedules on the failure probability of different classes of LETSs in  $\mathcal{C}_2$  ( $E_b/N_0 = 6$  dB, saturation level 15.75).



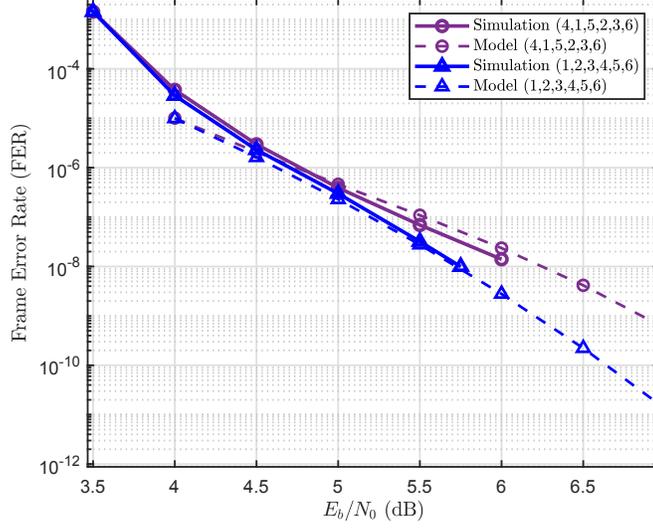
**Figure 4.12:** The effect of various row layered schedules on the total FER of  $\mathcal{C}_2$  ( $E_b/N_0 = 6$  dB, saturation level 15.75).



**Figure 4.13:** The error estimation of the ten  $(7, 1)$  LETS groups of  $\mathcal{C}_2$  for the two schedules  $(1, 2, 3, 4, 5, 6)$  and  $(4, 1, 5, 2, 3, 6)$  (saturation level 15.75).

For the two schedules  $(1, 2, 3, 4, 5, 6)$  and  $(4, 1, 5, 2, 3, 6)$ , we have estimated the failure probability of each of the 10 LETS groups within the class  $(7, 1)$ , using the exact DE. These results along with the total failure probability of the  $(7, 1)$  class are provided in Fig. 4.13. We note that the LETSs within first and third groups, i.e.,  $(7, 1)_1$  and  $(7, 1)_3$ , are isomorphic (they only differ by their TSLPs). So are the LETSs within  $(7, 1)_6$  and  $(7, 1)_7$ . The examination of Fig. 4.13 for schedule  $(4, 1, 5, 2, 3, 6)$  shows that the two groups  $(7, 1)_6$  and  $(7, 1)_7$ , despite having the same structure, have different error probabilities, due to different TSLPs. Fig. 4.13 also demonstrates that, for both schedules, the LETS group  $(7, 1)_{10}$  is the most harmful one. Another observation is that the relative harmfulness of the TSs can change depending on the schedule. For example, while the  $(7, 1)_8$  is the second most harmful group of LETSs for schedule  $(4, 1, 5, 2, 3, 6)$ , for  $(1, 2, 3, 4, 5, 6)$ , the second most harmful group is  $(7, 1)_9$ .

In order to examine the accuracy of our estimations, the simulation results for the two schedules together with the estimation results based on the contributions of  $(6, 1)$ ,  $(7, 1)$ ,  $(8, 0)$  and  $(8, 1)$  LETS classes are shown in Fig. 4.14. As can be seen, for both schedules, there is a good match between simulations and estimations.



**Figure 4.14:** Simulation and estimation results of  $\mathcal{C}_2$  for the two schedules (1, 2, 3, 4, 5, 6) and (4, 1, 5, 2, 3, 6) (saturation level 15.75,  $I_{max} = 30$ ).

## 4.4 Appendix

To prove Proposition 3, we first need the following lemmas.

**Lemma 11.** *Consider the linear state-space model of a LETS in which the two state variables corresponding to each missatisfied CN are labeled by consecutive numbers, and let  $\mathbf{A}$  be the corresponding flooding transition matrix. We then have*

- (a) *If  $\mathbf{A}(i, j) = 1$ , then  $\mathbf{A}(j + 2 \times \text{mod}(j, 2) - 1, i + 2 \times \text{mod}(i, 2) - 1) = 1$ , where  $\text{mod}(i, 2)$  is used to denote the value of  $i$  modulo 2.*
- (b) *If the nonzero entries of an  $m_s \times m_s$  symmetric and unitary permutation matrix  $\mathbf{P}$  are defined by  $\mathbf{P}(i, i + 2 \times \text{mod}(i, 2) - 1) = 1$ , for  $i = 1, \dots, m_s$ , then*

$$\mathbf{A}^T = \mathbf{PAP}. \quad (4.36)$$

*Proof.* (a) To each missatisfied CN, there correspond two state variables with even and odd labels, respectively. Denoting the state variables of a given missatisfied CN by  $x_{2k}$  and  $x_{2k-1}$ , one can see that each of them is a function of at least one other state variable corresponding to another missatisfied CN. Suppose that  $x_{2k}$  is a function of state variable  $x_{2k'}$  (or  $x_{2k'-1}$ ) from another missatisfied CN. Then,  $x_{2k'-1}$  (or  $x_{2k'}$ ) must be a function of  $x_{2k-1}$ . This corresponds to the relationship between

the entries of  $\mathbf{A}$  as described in Part (a) of the lemma. (b) The symmetric application of the permutation matrix  $\mathbf{P}$  to  $\mathbf{A}$ , i.e.,  $\mathbf{PAP}^T$ , permutes the even and odd rows and columns that correspond to each missatisfied CN. Based on the result of Part (a), this permutation results in the transpose matrix  $\mathbf{A}^T$ . Also, Equation (4.36) is derived based on the fact that  $\mathbf{P}^T = \mathbf{P}$ . ■

**Lemma 12.** *Let  $\mu_k$  and  $\mathbf{u}_k$  be an eigenvalue and its corresponding right eigenvector of a layered transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ , which is in systematic form. Then,*

$$(\mu_k \mathbf{A}_l + \mathbf{A}_u) \mathbf{u}_k = \mu_k \mathbf{u}_k, \quad (4.37)$$

where  $\mathbf{A}_l$  and  $\mathbf{A}_u$  are the lower and upper triangular parts of the corresponding flooding transition matrix  $\mathbf{A} = \mathbf{A}_l + \mathbf{A}_u$ .

*Proof.* For simplicity, we prove the result for a LETS with  $J = 3$  layers. The proof for larger values of  $J$  is similar. By the definition of an eigenvalue and the corresponding eigenvector, we have

$$\tilde{\mathbf{A}}_{3 \rightarrow 1} \begin{bmatrix} \mathbf{u}_{k1} \\ \mathbf{u}_{k2} \\ \mathbf{u}_{k3} \end{bmatrix} = \mu_k \begin{bmatrix} \mathbf{u}_{k1} \\ \mathbf{u}_{k2} \\ \mathbf{u}_{k3} \end{bmatrix},$$

in which the eigenvector is partitioned according to the three layers. By replacing  $\tilde{\mathbf{A}}_{3 \rightarrow 1}$  in the above equation with the sub-matrices from (4.18), we can write

$$\begin{aligned} \mathbf{A}_{12} \mathbf{u}_{k2} + \mathbf{A}_{13} \mathbf{u}_{k3} &= \mu_k \mathbf{u}_{k1}, \\ \mathbf{A}_{21} \underbrace{(\mathbf{A}_{12} \mathbf{u}_{k2} + \mathbf{A}_{13} \mathbf{u}_{k3})}_{\mu_k \mathbf{u}_{k1}} + \mathbf{A}_{23} \mathbf{u}_{k3} &= \mu_k \mathbf{u}_{k2}, \\ \mathbf{A}_{31} \underbrace{(\mathbf{A}_{12} \mathbf{u}_{k2} + \mathbf{A}_{13} \mathbf{u}_{k3})}_{\mu_k \mathbf{u}_{k1}} + \\ \mathbf{A}_{32} \underbrace{(\mathbf{A}_{21} (\mathbf{A}_{12} \mathbf{u}_{k2} + \mathbf{A}_{13} \mathbf{u}_{k3}) + \mathbf{A}_{23} \mathbf{u}_{k3})}_{\mu_k \mathbf{u}_{k2}} &= \mu_k \mathbf{u}_{k3}, \end{aligned}$$

or equivalently,

$$\underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mu_k \mathbf{A}_{21} & \mathbf{0} & \mathbf{A}_{23} \\ \mu_k \mathbf{A}_{31} & \mu_k \mathbf{A}_{32} & \mathbf{0} \end{bmatrix}}_{(\mu_k \mathbf{A}_l + \mathbf{A}_u)} \begin{bmatrix} \mathbf{u}_{k1} \\ \mathbf{u}_{k2} \\ \mathbf{u}_{k3} \end{bmatrix} = \mu_k \begin{bmatrix} \mathbf{u}_{k1} \\ \mathbf{u}_{k2} \\ \mathbf{u}_{k3} \end{bmatrix},$$

which is the same as (4.37) for  $J = 3$ . ■

The proof of the following lemma is similar to that of Lemma 12.

**Lemma 13.** *Let  $\tilde{\mathbf{A}}_{1 \rightarrow J} = \mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_J$  be the systematic layered transition matrix of a LETS for the row layered decoder in which the order of layers is reversed. Also, let  $\mu'_k$  and  $\mathbf{u}'_k$  be an eigenvalue and its corresponding right eigenvector of  $\tilde{\mathbf{A}}_{1 \rightarrow J}$ . Then*

$$(\mathbf{A}_l + \mu'_k \mathbf{A}_u) \mathbf{u}'_k = \mu'_k \mathbf{u}'_k, \quad (4.38)$$

where  $\mathbf{A}_l$  and  $\mathbf{A}_u$  are the lower and upper triangular part of the corresponding flooding transition matrix  $\mathbf{A} = \mathbf{A}_l + \mathbf{A}_u$ .

To prove Proposition 3, without loss of generality, we assume that  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$  is in the systematic form. Then, according to Lemma 12, for  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$ , we have

$$(\tilde{r}_i \mathbf{A}_l + \mathbf{A}_u) \tilde{\mathbf{u}}_i = \tilde{r}_i \tilde{\mathbf{u}}_i,$$

where  $\tilde{r}_i$  and  $\tilde{\mathbf{u}}_i$  are an eigenvalue and its corresponding eigenvector of  $\tilde{\mathbf{A}}_{\pi_J \rightarrow \pi_1}$ , respectively. Suppose that  $\mathbf{P}$  is the permutation matrix defined in Lemma 11. Since  $\mathbf{P}\mathbf{P} = \mathbf{I}$ , we can write

$$(\tilde{r}_i \mathbf{A}_l \mathbf{P} + \mathbf{A}_u \mathbf{P}) \mathbf{P} \tilde{\mathbf{u}}_i = \tilde{r}_i \tilde{\mathbf{u}}_i. \quad (4.39)$$

Moreover, for a systematic flooding transition matrix  $\mathbf{A}$ , one can show that  $\mathbf{A}_l \mathbf{P} = \mathbf{P} \mathbf{A}_u^T$  and  $\mathbf{A}_u \mathbf{P} = \mathbf{P} \mathbf{A}_l^T$ . Using these in (4.39), we obtain

$$(\mathbf{P} \mathbf{A}_l^T + \tilde{r}_i \mathbf{P} \mathbf{A}_u^T) \mathbf{P} \tilde{\mathbf{u}}_i = \tilde{r}_i \tilde{\mathbf{u}}_i. \quad (4.40)$$

Multiplying both sides by  $\mathbf{P}$  results in

$$(\mathbf{A}_l + \tilde{r}_i \mathbf{A}_u)^T \mathbf{P} \tilde{\mathbf{u}}_i = \tilde{r}_i \mathbf{P} \tilde{\mathbf{u}}_i. \quad (4.41)$$

From (4.41), the eigenvalues  $\tilde{r}_i$  are the roots of the following equation:

$$\det[(\mathbf{A}_l + \tilde{r}_i \mathbf{A}_u)^T - \tilde{r}_i \mathbf{I}] = \det[(\mathbf{A}_l + \tilde{r}_i \mathbf{A}_u) - \tilde{r}_i \mathbf{I}] = 0. \quad (4.42)$$

On the other hand, based on Lemma 13, the roots of the determinant on the right hand side are the eigenvalues of  $\tilde{\mathbf{A}}_{\pi_1 \rightarrow \pi_J}$ . This completes the proof.

## Chapter 5

# Error Floor Analysis of LDPC Column Layered Decoders

The rest of the chapter is organized as follows: In Section 5.1, we briefly review the linear state-space model of an ETS for a decoder with layered schedule. In the same section, we also establish connections between the model parameters of the column layered and flooding schedules. In Section 5.2, we discuss the optimization of error floor in column layered decoders. In Section 5.3, we present the simulation results to evaluate the accuracy of the theoretical results in estimating the error floor.

## 5.1 State Space Model of the Column Layered Decoders in the Error Floor Region

In this section, we develop a linear state-space model of LETSs for SPA with column layered schedule, and use the model to calculate the error probability of LETSs. It is interesting to note that while the state space model of Chapter 4 has been, originally, proposed for row layered decoders, that model can also be utilized for analysis of the error floor in column layered decoders. In other words, the general equations of the models in column and row layered schedules are the same but the parameters of the models must be calculated differently.

### 5.1.1 State-Space Model

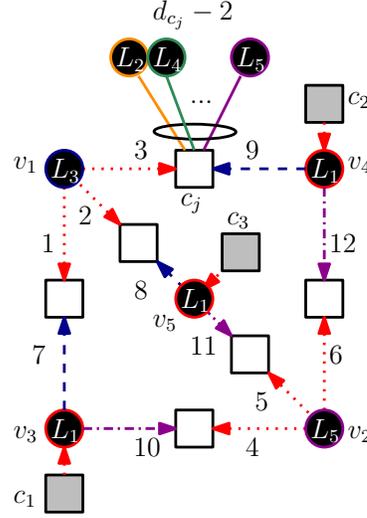
In this subsection, the state-space model of Chapter 4 which is also applicable to modelling an LETS in column layered decoding is briefly reviewed.

The state-space modelling equations of the LETSs in layered decoder are given as

$$\tilde{\mathbf{x}}^{(0,j)} = \mathbf{0} \quad \text{for } \ell = 0, 1 \leq j \leq J \quad (5.1)$$

$$\begin{aligned} \tilde{\mathbf{x}}^{(\ell,j)} = & \quad \text{for } \ell \geq 1, 1 \leq j \leq J \\ & \mathcal{G}_j^{(\ell)} (\mathcal{A}_j \tilde{\mathbf{x}}^{(\ell-\delta_{j1},j-1+J\delta_{j1})} + \mathcal{B}_j \mathbf{L} + \overset{\triangleleft}{\mathcal{B}}_{exj} \mathbf{L}_{ex}^{(\ell-1)} + \overset{\triangleright}{\mathcal{B}}_{exj} \mathbf{L}_{ex}^{(\ell)}). \end{aligned} \quad (5.2)$$

In these equations,  $\mathbf{L}$ ,  $\mathbf{L}_{ex}^{(\ell-1)}$  and  $\mathbf{L}_{ex}^{(\ell)}$  are the inputs of the model from the channel as well as the unsatisfied CNs at iterations  $\ell - 1$  and  $\ell$ , respectively. It is noted that  $\mathbf{L}_{ex}^{(0)} = \mathbf{0}$ . The symbol  $J$  is used to denote the total number of layers involved in an LETS subgraph. The vector  $\tilde{\mathbf{x}}^{(\ell,j)}$  contains the state variables (messages flowing at different edges of the LETS subgraph) at the layer  $j$  of the  $\ell$ th iteration. The transition matrix at the  $j$ th layer of the layered decoder,  $\mathcal{A}_j$ , is equal to an  $m_s \times m_s$  identity matrix whose corresponding rows related to the  $j$ th layer state-variables are replaced with the message passing mechanism of that layer. In other words, each state-variable is a function of some other state-variables and this relation at layer  $j$  is specified by matrix  $\mathcal{A}_j$ . The matrix  $\mathcal{B}_j$  is an  $m_s \times a$  matrix that indicates the contribution of the channel LLRs in calculation of the state variables of the  $j$ th layer. Similarly, the matrices  $\overset{\triangleleft}{\mathcal{B}}_{exj}$  and  $\overset{\triangleright}{\mathcal{B}}_{exj}$  are defined to illustrate the contribution of  $\mathbf{L}_{ex}^{(\ell-1)}$  and  $\mathbf{L}_{ex}^{(\ell)}$  in updating the state variables within the  $j$ th layer, respectively. Also,  $\delta_{j1}$ , is the Kronecker delta function which is equal to 1 when  $j = 1$  and is zero, otherwise. Equation (5.2) implies that at the first layer of every iteration, the new state vector is updated based on the last layer of the previous iteration while in the other layers, the updated states are a function of previous layer state vector within the same iteration. The effect of mis-satisfied CNs within a given layer  $j$  at iteration  $\ell$  is modelled by the iteration dependent gain matrix  $\mathcal{G}_j^{(\ell)}$ . It is a diagonal matrix whose diagonal elements that correspond to layer  $j$  is equal to the average gain obtained from linear estimation of the mis-satisfied CNs impacting the  $j$ th layer state variables. All the other diagonal elements are equal to one.



**Figure 5.1:** A labelled (5,3) LETS of the Tanner (155, 64) code in which the flow of the state variables are shown. The VNs, mis-satisfied CNs and unsatisfied CNs are shown by black circles, white squares and gray squares, respectively. The directed edges related to the layers number 1, 2 and 3 are represented by red, blue and purple colors, respectively.

### 5.1.2 Relationship between model matrices of flooding and column layered schedules

Regarding the TSs in column layered decoders, the VNs belong to different layers of the code. The number of VN layers that exist in an LETS is, generally, less than or equal to the total number of columns of the base graph,  $J \leq n_b$ . In this chapter, we use the notation  $L_i$  ( $i = 1, \dots, n_b$ ) to denote the  $i$ th column layer of the code. Also, the corresponding layers of a TS are denoted by  $L'_j$  ( $j = 1, \dots, J$ ). Obviously, the TS layers  $L'_j$  are a subset of the code layers  $L_i$ .

At each column layer, certain internal messages of a TS toward the missatisfied CNs become activated. These messages would propagate from the VNs of other layers that are adjacent to the missatisfied CNs connected to VNs of current layer. This is because, according to the steps of column layered decoding in Algorithm 2, when a column layer is updated, the VNs of that layer do not contribute in message passing and, only, receive information from their adjacent CNs whose information are computed based on the receiving LLRs from the VNs of other layers.

It is noted that, in column layered decoders, the corresponding messages toward a specific missatisfied CN are activated at two different layers while, as it was explained

in Chapter 4, the incoming messages of missatisfied CNs in row layered decoders are activated at the same time. This is one of the main differences between column layered and row layered decoders state-space model.

**Example 12.** *The quasi-cyclic Tanner (155, 64) code consists of 5 column layers. The VNs involved in the (5, 3) LETS of this code, shown in Fig. 5.1, are from 3 different layers,  $L'_1 = L_1$ ,  $L'_2 = L_3$  and  $L'_3 = L_5$ , hence  $J = 3$ . In Fig. 5.1, the VNs and the edges activated at the corresponding layers are illustrated by different colors and line types. For example, all the dotted red arrows are the messages that are sent from VNs to mis-satisfied CNs in the first layer of the column layered decoder. Also, the external connections of missatisfied CNs from the model point of view are activated twice (i.e., once for each of the internal messages going toward missatisfied CNs).*

To establish a relationship between the matrices that appear in the flooding and column layered decoders, it is helpful to label the state variables in a certain order. Consider an  $(a, b)$  LETS  $\mathcal{S}$  with  $m_s$  state-variables  $x_1, \dots, x_{m_s}$ . Suppose the VNs of  $\mathcal{S}$  are from  $J$  different column layers of parity check matrix  $\mathbf{H}$  where  $J \leq n_b$ . We denote the number of messages (state-variables) belonging to different layers  $L'_j$  ( $j = 1, \dots, J$ ) by  $n_{L'_j}$  ( $j = 1, \dots, J$ ). As the total number of state-variables are  $m_s$ , we have  $n_{L'_1} + n_{L'_2} + \dots + n_{L'_J} = m_s$ . Assume the state-variables of different layers are indexed with consecutive numbers meaning that the state-variables of the first layer are labelled as  $x_1, \dots, x_{n_{L'_1}}$ , the state-variables of the second layer are labelled as  $x_{n_{L'_1}+1}, \dots, x_{n_{L'_1}+n_{L'_2}}$ , and so on. This form of labelling the state-variables known as *systematic labelling* allows us to establish a relationship between the transition matrix of the flooding schedule decoder denoted by  $\mathbf{A}$  and the transition matrix at layer  $j$ ,  $\mathcal{A}_j$ , of the layered schedule decoding. In other words, based on the systematic labelling, the flooding decoder transition matrix  $\mathbf{A}$  would be a  $J \times J$  arrays of matrices  $\mathbf{A}_{ij}$ ,  $1 \leq i, j \leq J$  with sizes  $n_{L'_i} \times n_{L'_j}$  and diagonal blocks of zero matrices, i.e.,  $\mathbf{A}_{ii} = 0$  for  $i = 1, \dots, J$ . The matrix  $\mathcal{A}_j$ , basically, has the same block structure as matrix  $\mathbf{A}$ . In fact, the  $j$ th row block of the two matrices are equal while the other row blocks of  $\mathcal{A}_j$  are zero matrices except the diagonal blocks that are identity matrices. The matrices  $\mathcal{B}_j$ ,  $\overset{\triangleleft}{\mathcal{B}}_{exj}$  and  $\overset{\triangleright}{\mathcal{B}}_{exj}$ , following the systematic labelling, can also be partitioned into  $J$  row blocks. The  $j$ th row block of these matrices relate the channel inputs, the unsatisfied inputs updated at the previous iteration and the unsatisfied inputs updated at the current iteration to the state-variables being updated at layer  $j$ ,

respectively. Moreover, all the other row blocks of these matrices are equal to zero matrices.

**Example 13.** *in Fig. 5.1, the numbers 1, 2, 3, 4, 5 and 6 are assigned to the edges in the first layer,  $L'_1 = L_1$ , and, similarly, the  $L'_2 = L_3$  and  $L'_3 = L_5$  messages are labelled as 7 to 9 and 10 to 12, respectively. This results in systematic form of matrix  $\mathbf{A}$  as*

$$\mathbf{A} = \left[ \begin{array}{c|c|c} \mathbf{0} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \hline \mathbf{A}_{21} & \mathbf{0} & \mathbf{A}_{23} \\ \hline \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{0} \end{array} \right]$$

$$= \left[ \begin{array}{cccccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right],$$

which is a  $J \times J$  block matrix whose diagonal blocks are equal to zero. The reason for zero diagonal blocks is that the non-zero circulants of the single-edge quasi-cyclic codes are made of shifted version of identity matrices. As a result, a state variable belonging to a specific layer cannot be a function of other state variables within the

same layer. Otherwise, a VN must be connected to more than one CN with the same type which is a direct contradiction to the structure of the quasi-cyclic codes. Based on the systematic form of the flooding decoder matrix  $\mathbf{A}$  above, the transition matrix at the 2nd layer of the column layered decoding in Fig. 5.1 is obtained as

$$\mathcal{A}_2 = \left[ \begin{array}{c|c|c} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{A}_{21} & \mathbf{0} & \mathbf{A}_{23} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array} \right].$$

The symbol  $\mathbf{I}$  is used to represent the identity matrix. Presence of the identity matrices implies that the corresponding state variables are not updated within the current layer and remain unchanged. Also, the matrices  $\overset{\triangleleft}{\mathcal{B}}_{ex2}$  and  $\overset{\triangleright}{\mathcal{B}}_{ex2}$  that determine the contribution of unsatisfied inputs at iterations  $\ell - 1$  and  $\ell$  in the second layer,  $L'_2$ , of  $\ell_{th}$  iteration, can be obtained as

$$\overset{\triangleleft}{\mathcal{B}}_{ex2} = \left[ \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right], \quad \overset{\triangleright}{\mathcal{B}}_{ex2} = \left[ \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right].$$

As can be seen, since all the unsatisfied inputs are updated at layer  $L'_1 = L_1$ , all the elements of  $\overset{\triangleleft}{\mathcal{B}}_{ex2}$ , corresponding to layer  $L'_2 = L_3$ , are equal to zero and all the state

variables of the second layer use the updated external inputs of unsatisfied CNs,  $L_{ex}^{(\ell)}$ . Similarly, for  $\mathcal{B}_2$  we have

$$\mathcal{B}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

meaning that the channel LLRs of VNs 3, 5 and 4 are utilized in computation of state variables 7, 8 and 9, respectively.

**Remark 4.** *The systematic form of transition matrix obtained based on column layered and row layered schedules have the same block structures. In other words, in both cases, the resulted systematic transition matrix consists of arrays of matrices where all the diagonal blocks are zero matrices. Naturally, the total number of layers within a LETS as well as the subblocks of systematic transition matrix of flooding decoder in column and row schedules can be different.*

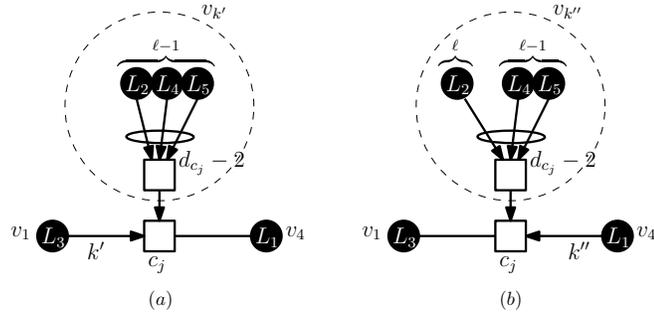
### 5.1.3 The Application of DE to the Column Layered Decoder

In the linear state-space model of a LETS, the DE method is used to obtain the distribution of external messages entering the LETS subgraph from the rest of the Tanner graph via missatisfied and unsatisfied CNs. With respect to column layered

decoding of QC-LDPC codes, the probability distribution of messages can be obtained with respect to the base graph of the code. In other words, the edges of the base graph are representatives of the cluster of edges between different types of VNs and CNs. In the base graph of a QC-LDPC code, within column layer  $j$  of iteration  $\ell$ , the type- $i$  CN to type- $j$  VN distribution denoted by  $\psi_\ell^{[j \leftarrow i]}$  is obtained based on the distribution of messages being sent toward type- $i$  CN from other VNs  $k \in N(i) \setminus j$  whose types are different than  $j$ . Naturally, depending on the type of these VNs as well as the updating order of the code layers, the VNs  $k \in N(i) \setminus j$  are partitioned into two groups: 1) those that have already been updated in the previous layers of the current iteration  $\ell$  and propagate the updated messages with distribution  $\psi_\ell^{[k \rightarrow i]}$  toward type- $i$  CN. 2) those belonging to the subsequent layers and have not been updated yet. This group would propagate messages calculated at iteration  $\ell - 1$  with distribution  $\psi_{\ell-1}^{[k \rightarrow i]}$ . Following the mechanism just discussed, the probability distribution of messages between different types of VNs and CNs at various iterations and layers can be numerically obtained.

**Example 14.** *In the base graph of Fig. 5.3, the solid (red), dashed (green), dotted (blue) and dashed-dotted (purple) arrows, respectively, illustrate the updating order which the column layered decoder perform, accordingly. The symbols next to the dashed arrows (boldfaced edges) stand for the probability distribution of the CN to VN messages activated at the 2nd layer of the  $\ell$ th iteration. Also,  $\psi_\ell^{[1 \rightarrow 1]}$  and  $\psi_{\ell-1}^{[3 \rightarrow 1]}$ , updated in current and previous iterations respectively, are VN to CN distributions that contribute in calculation of  $\psi_\ell^{[1 \leftarrow 2]}$ . Similarly, as the 4th layer has not been updated yet,  $\psi_{\ell-1}^{[4 \rightarrow 3]}$  is used to calculate  $\psi_{\ell-1}^{[2 \leftarrow 3]}$ .*

Regarding the LETSs, as it is mentioned earlier, the VNs belong to various column layers of the code. Also, the external VNs connected to different dissatisfied CNs as well as the unsatisfied CNs are of various types. This implies that the location of the TSs within the layers of the code Tanner graph is important as this will impact the internal order of layers within the TS as well as the distribution of external messages entering the TS subgraph. These two factors would potentially cause different harmfulness, even, among isomorphic TSs whose structures are similar. As a result, in the state-space model, it is important to identify the type of variable nodes connected to different check nodes of the TS and the check node types connected to different variable nodes of the TS. These information, similar to Chapter 4, are referred to as *TS layer profile* (TSLP).



**Figure 5.2:** The VNs  $v_1$  and  $v_4$  related to Fig. 5.1 with the mis-satisfied CN  $c_j$  connecting them. Figures (a) and (b) are related to virtual VNs,  $v_{k'}$  and  $v_{k''}$ , that are used during the update of VNs from layers  $L_1$  and  $L_3$ , respectively.

**Remark 5.** For a LETS in (a,b) class within the Tanner graph of an LDPC code, the total number of distinct TSLP groups in column and row schedule decoders are the same. In fact, different schedules might change the TSLP of a LETS, however, they do not alter the total number of possible TSLP groups related to a specific LETSs in a code.

### 5.1.4 Linear Gain Model

In the layered decoders state-space model, the impact of external connections of each of the missatisfied CNs on the corresponding state-variables are modelled by an average linear coefficient known as linear gain. In fact, these gains correspond to the diagonal elements of the gain matrices discussed earlier.

Suppose the external connections of each missatisfied CN at every layer, for simplicity, is represented by a virtual VN, as Fig. 5.2. Then, the average gain,  $\bar{g}_{k'}^{(\ell)}$  is obtained as

$$\bar{g}_{k'}^{(\ell)} = \int_{-\infty}^{\infty} \tanh\left(\frac{\lambda}{2}\right) \hat{\psi}_{\ell}^{[k' \rightarrow j]}(\lambda) d\lambda, \quad (5.3)$$

in which  $\hat{\psi}_{\ell}^{[k' \rightarrow j]}$  is the probability distribution of the message from virtual VN,  $v_{k'}$ , toward the mis-satisfied CN at iteration  $\ell$ . Given the TSLP of an LETS, this probability can be calculated from the DE results for different mis-satisfied CNs at a specific layer. With respect to Fig. 5.2 which is a part of Fig. 5.1,  $\hat{\psi}_{\ell}^{[k' \rightarrow j]}$  is, technically, the CN to VN message distribution of a CN with degree 4 (i.e.  $d_{c_j} - 1$ ) where the inputs are the distribution of messages from type-2, 4 and 5 VNs toward type-1 CNs ( $c_j$  is a type-1 CN). It is noted that unlike the row layered decoder, the external VN

to CN message distributions toward each missatisfied CN, depending on the column layer of the corresponding internal VNs, can be different. For example, in Fig. 5.2a, when the message  $k'$  is activated at the  $L'_1 = L_1$  layer, none of the external VNs have been updated in the current iteration yet. However, in Fig. 5.2b where the message  $k''$  is activated at  $L'_2 = L_3$  layer, one of the external VNs that belongs to layer  $L_2$  sends an updated message toward  $c_j$  while the others propagate their information from previous iteration,  $\ell - 1$ . Therefore, for each of the state variables going toward missatisfied CNs, a separate virtual VN and accordingly a separate gain must be calculated.

Equation (5.3) is further modified by adding the polarity inversion effect into the model. Whenever an error occurs in the messages from the virtual VNs,  $v_{k'}$  for  $k' = \{1, \dots, m_s\}$ , the polarity of the state variables passing through the mis-satisfied CNs is altered. Denoting the probability of inversion by  $P_{inv,\ell}^{[k' \rightarrow j]}$ , we have

$$P_{inv,\ell}^{[k' \rightarrow j]} = \int_{-\infty}^0 \hat{\psi}_\ell^{[k' \rightarrow j]}(\lambda) d\lambda. \quad (5.4)$$

In layered decoder, like the standard decoder, the effect of polarity inversion is modelled by randomly setting the gain equal to zero. As it has been explained in [45], appending the polarity inversion effect into the model is a heuristic approach and can be considered as just adding more weight to the importance of inversions. By using (5.4), the average mis-satisfied CNs gain can be written as

$$\bar{g}_{k'}^{(\ell)} = (1 - P_{inv,\ell}^{[k' \rightarrow j]}) \bar{g}_{k'}^{(\ell)}. \quad (5.5)$$

As iterations go on, the multiplicative gains of different mis-satisfied CNs approach to 1. In the following, we will discuss an alternative approach of finding the missatisfied CN gains. The new approach will be used later in the process of layer optimization.

Suppose when a state variable  $x_i$  is activated and goes through a missatisfied CN, the external VNs whose information have been updated at iterations  $\ell - 1$  and  $\ell$  are denoted by two different sets,  $\overset{\triangleleft}{\tau}_i$  and  $\overset{\triangleright}{\tau}_i$ , respectively. Using Equation (2.3) to compute the message from a missatisfied CN  $c_j$  toward a trapping set VN,  $v_{i'}$ , we have

$$L_\ell^{[i' \leftarrow j]} = 2 \tanh^{-1} \left[ \tanh \frac{x_i}{2} \prod_{k_1 \in \overset{\triangleleft}{\tau}_i} \tanh \frac{L_{\ell-1}^{[k_1 \rightarrow j]}}{2} \tanh \prod_{k_2 \in \overset{\triangleright}{\tau}_i} \tanh \frac{L_\ell^{[k_2 \rightarrow j]}}{2} \right]. \quad (5.6)$$

Denoting the right hand side of the above equation with  $f(x_i)$ , the first derivative of  $f(x_i)$  with respect to state variable when  $x_i = 0$  is

$$f_x(0) = \prod_{k_1 \in \overleftarrow{\tau}_i} \tanh \frac{L_{\ell-1}^{[k_1 \rightarrow j]}}{2} \tanh \prod_{k_2 \in \overrightarrow{\tau}_i} \tanh \frac{L_{\ell}^{[k_2 \rightarrow j]}}{2}. \quad (5.7)$$

This is similar to the approach taken in [45]. Taking the average of (5.7) would result in the average gain being used in Taylor expansion of missatisfied CNs. In fact, this is an alternative to virtual VN technique in (5.3). As it is assumed the external messages entering the missatisfied CNs are independent, we have

$$\bar{g}_i^{(\ell)} = \prod_{k_1 \in \overleftarrow{\tau}_i} \theta_{k_1}^{\overleftarrow{(\ell)}} \prod_{k_2 \in \overrightarrow{\tau}_i} \theta_{k_2}^{\overrightarrow{(\ell)}}, \quad (5.8)$$

where the partial VN to CN gains,  $\theta_{k_1}^{\overleftarrow{(\ell)}}$  and  $\theta_{k_2}^{\overrightarrow{(\ell)}}$ , are defined as

$$\theta_{k_1}^{\overleftarrow{(\ell)}} = \int_{-\infty}^{\infty} \tanh\left(\frac{\lambda}{2}\right) \psi_{\ell-1}^{[k_1 \rightarrow j]}(\lambda) d\lambda, \quad k_1 \in \overleftarrow{\tau}_i \quad (5.9)$$

$$\theta_{k_2}^{\overrightarrow{(\ell)}} = \int_{-\infty}^{\infty} \tanh\left(\frac{\lambda}{2}\right) \psi_{\ell}^{[k_2 \rightarrow j]}(\lambda) d\lambda, \quad k_2 \in \overrightarrow{\tau}_i. \quad (5.10)$$

The distributions  $\psi_{\ell-1}^{[k_1 \rightarrow j]}$  and  $\psi_{\ell}^{[k_2 \rightarrow j]}$  are the VN to CN probability distributions corresponding to different external connections of a given missatisfied CN at iterations  $\ell - 1$  and  $\ell$  (They are different from virtual VNs distribution,  $\hat{\psi}_{\ell}^{[k' \rightarrow j]}$ ). Equation (5.8) will be used in section 5.2 to simplify the layers optimization procedure. In that section, an approximate DE method is used to reduce the complexity of optimization. However, changing the order of layers would impact the parameters  $\overleftarrow{\tau}_i$  and  $\overrightarrow{\tau}_i$  of each of the missatisfied CNs implying that within virtual VN technique a new distribution must be calculated. To tackle this problem, based on (5.9) and (5.10), the partial gains can be computed in advance and only their contribution, according to  $\overleftarrow{\tau}_i$  and  $\overrightarrow{\tau}_i$  in (5.8), will be changed. It is noted that, technically, for a specific VN to CN message, we have

$$\theta_k^{\overleftarrow{(\ell)}} = \theta_k^{\overrightarrow{(\ell-1)}}. \quad (5.11)$$

**Example 15.** Suppose the external VNs of CN  $c_j$  in Fig. 5.2 are labelled with the

same subscripts of their corresponding layers as  $v'_2, v'_4, v'_5$ . Then the VN to CN distributions  $\psi_{\ell-1}^{[2 \rightarrow j]}$ ,  $\psi_{\ell-1}^{[4 \rightarrow j]}$  and  $\psi_{\ell-1}^{[5 \rightarrow j]}$  are used to calculate the virtual VN message distribution  $\hat{\psi}_\ell^{[k' \rightarrow j]}$ . Also, the VN to CN distributions  $\psi_\ell^{[2 \rightarrow j]}$ ,  $\psi_{\ell-1}^{[4 \rightarrow j]}$  and  $\psi_{\ell-1}^{[5 \rightarrow j]}$  are used to calculate the virtual VN message distribution  $\hat{\psi}_\ell^{[k'' \rightarrow j]}$ . Accordingly, the partial VN to CN gains for  $x_{k'}$  and  $x_{k''}$  are obtained as

$$\bar{g}_{k'}^{(\ell)} = \theta_2^{\triangleleft(\ell)} \theta_4^{\triangleleft(\ell)} \theta_5^{\triangleleft(\ell)}, \quad (5.12)$$

$$\bar{g}_{k''}^{(\ell)} = \theta_2^{\triangleright(\ell)} \theta_4^{\triangleleft(\ell)} \theta_5^{\triangleleft(\ell)}, \quad (5.13)$$

where  $\triangleleft_{k'} = 2, 4, 5$  and  $\triangleright_{k'}$  is empty while  $\triangleleft_{k''} = 4, 6$  and  $\triangleright_{k''} = 2$ .

### 5.1.5 Failure Probability

The probability of TS failures in column layered decoder can be, basically, obtained by the same approach as the row layered decoder. In this regard, first the non-recursive form of the state-vector equation versus the model inputs is calculated. The resulted equation, then, is used to obtain an error indicator function. Finally, by Gaussian approximation, the failure probability of a TS is estimated.

The ordered multiplication of the scaled version of transition matrices of different layers,  $\mathcal{G}_j^{(\ell)} \mathcal{A}_j$ , from the maximum layer  $J$  down to the  $k$ th layer,  $k \geq 1$  is defined as

$$\tilde{\mathbf{A}}_{J \rightarrow k}^{(\ell)} = (\mathcal{G}_J^{(\ell)} \mathcal{A}_J)(\mathcal{G}_{J-1}^{(\ell)} \mathcal{A}_{J-1}) \dots (\mathcal{G}_k^{(\ell)} \mathcal{A}_k). \quad (5.14)$$

The index  $(\ell)$  shows the iteration dependency of this matrix due to time variant gain matrices. By using  $\tilde{\mathbf{A}}_{J \rightarrow k}^{(\ell)}$ , three other matrices are defined as

$$\tilde{\mathbf{B}}^{(\ell)} = \tilde{\mathbf{A}}_{J \rightarrow 2}^{(\ell)} \mathcal{G}_1^{(\ell)} \mathcal{B}_1 + \tilde{\mathbf{A}}_{J \rightarrow 3}^{(\ell)} \mathcal{G}_2^{(\ell)} \mathcal{B}_2 + \dots + \mathcal{G}_J^{(\ell)} \mathcal{B}_J, \quad (5.15)$$

$$\mathbf{B}_{ex}^{\triangleright(\ell)} = \tilde{\mathbf{A}}_{J \rightarrow 2}^{(\ell)} \mathcal{G}_1^{(\ell)} \mathcal{B}_{ex1}^{\triangleright} + \tilde{\mathbf{A}}_{J \rightarrow 3}^{(\ell)} \mathcal{G}_2^{(\ell)} \mathcal{B}_{ex2}^{\triangleright} + \dots + \mathcal{G}_J^{(\ell)} \mathcal{B}_{exJ}^{\triangleright}, \quad (5.16)$$

$$\mathbf{B}_{ex}^{\triangleleft(\ell)} = \tilde{\mathbf{A}}_{J \rightarrow 2}^{(\ell)} \mathcal{G}_1^{(\ell)} \mathcal{B}_{ex1}^{\triangleleft} + \tilde{\mathbf{A}}_{J \rightarrow 3}^{(\ell)} \mathcal{G}_2^{(\ell)} \mathcal{B}_{ex2}^{\triangleleft} + \dots + \mathcal{G}_J^{(\ell)} \mathcal{B}_{exJ}^{\triangleleft}. \quad (5.17)$$

Using the above matrices, the non-recursive form of the state vector at the end of

iteration  $\ell$  can be expressed as

$$\begin{aligned} \tilde{\mathbf{x}}^{(\ell, J)} &= \sum_{i=1}^{\ell} \left( \prod_{j=i+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j)} \right) \tilde{\mathbf{B}}^{(i)} \mathbf{L} \\ &+ \sum_{i'=1}^{\ell} \left( \prod_{j'=i'+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j')} \right) \left( \mathbf{B}_{ex}^{(i')} \mathbf{L}_{ex}^{(i'-1)} + \mathbf{B}_{ex}^{(i')} \mathbf{L}_{ex}^{(i')} \right) \end{aligned} \quad (5.18)$$

which is a function of model inputs from unsatisfied CNs as well as the channel. The right arrow on top of the product sign denotes the matrix product applied from the left. For example

$$\prod_{j=i+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j)} = \tilde{\mathbf{A}}_{J \rightarrow 1}^{(\ell)} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(\ell-1)} \cdots \tilde{\mathbf{A}}_{J \rightarrow 1}^{(i+1)}.$$

As it is mentioned before, the mis-satisfied CNs average gains tends to 1 as the iterations go on. So we have

$$\lim_{\ell \rightarrow \infty} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(\ell)} = \mathcal{A}_J \mathcal{A}_{J-1} \cdots \mathcal{A}_1 = \tilde{\mathbf{A}}_{J \rightarrow 1}. \quad (5.19)$$

The matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  is not iteration dependant anymore. At the heart of state-space model of layered decoders, the above matrix plays a crucial role in determining the growth rate of erroneous messages in LETS subgraphs.

Based on Remark 4, the systematic form of flooding schedule transition matrices obtained according to column or row schedule systematic labelling have similar block structures. This would lead to a similar structure of layered transition matrix,  $\tilde{\mathbf{A}}_{J \rightarrow 1}$ , corresponding to the row and column layered decoders. As a result, all the spectral properties of row layered schedule transition matrix that have been proofed in Chapter 4 are, also, valid for the transition matrix of an LETS in column layered decoder. The followings are some of the important spectral properties of matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  in column layered decoders:

- (a) The transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  of a LETS in a column layered decoder is reducible.
- (b)  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  has a positive dominant eigenvalue,  $\tilde{r}$ , equal to its spectral radius whose corresponding left and right eigenvectors,  $\tilde{\mathbf{w}}_1$  and  $\tilde{\mathbf{u}}_1$ , are non-negative and, also,  $\tilde{\mathbf{w}}_1 \neq \mathbf{0}$ ,  $\tilde{\mathbf{u}}_1 \neq \mathbf{0}$ .

- (c) With respect to a LETS, the dominant eigenvalue of layered schedule transition matrix,  $\tilde{r}$ , is always greater than or equal to the corresponding dominant eigenvalue of flooding schedule matrix,  $r$ , i.e.,  $\tilde{r} \geq r$ .
- (d) Given a LETS with  $J$  column layers, the corresponding dominant eigenvalue of the transition matrix in column layered schedule is invariant to any cyclic shift of the layers order.
- (e) Given a LETS with  $J$  column layers, the corresponding dominant eigenvalue of the transition matrix in column layered schedule is invariant to the reversing of the layers order<sup>1</sup>.
- (f) For a LETS with  $J$  column layers ( $J \geq 3$ ), the number of separate dominant eigenvalues of the column layered transition matrix corresponding to different permutations of the layers is upper bounded by  $\frac{(J-1)!}{2}$ .

In Chapter 4, the dominant left eigenvector of matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  has been used to compute an error indicator function from (5.18). In the following, we take the same approach. So, the error indicator function denoted by  $\tilde{\beta}^{(\ell)}$  is obtained as

$$\begin{aligned} \tilde{\beta}^{(\ell)} &\triangleq \tilde{\mathbf{w}}_1^T \tilde{\mathbf{x}}^{(\ell, J)} \\ &= \gamma_{ch}^T \mathbf{L} + \sum_{i'=1}^{\ell} (\gamma_{ex}^{\triangleleft(i')T} \mathbf{L}_{ex}^{(i'-1)} + \gamma_{ex}^{\triangleright(i')T} \mathbf{L}_{ex}^{(i')}), \end{aligned} \quad (5.20)$$

in which the vectors  $\gamma_{ch}^T$ ,  $\gamma_{ex}^{\triangleleft(i')T}$  and  $\gamma_{ex}^{\triangleright(i')T}$  with lengths equal to  $a$ ,  $b$  and  $b$  are defined as

$$\gamma_{ch}^T = \tilde{\mathbf{w}}_1^T \sum_{i=1}^{\ell} \left( \prod_{j=i+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j)} \right) \tilde{\mathbf{B}}^{(i)} \quad (5.21)$$

$$\gamma_{ex}^{\triangleleft(i')T} = \tilde{\mathbf{w}}_1^T \left( \prod_{j'=i'+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j')} \right) \mathbf{B}_{ex}^{\triangleleft(i')} \quad i' = 1, \dots, \ell \quad (5.22)$$

---

<sup>1</sup>The proof of this property which is slightly different than the approach taken in proof of Proposition 3 in Chapter 4, is explained in Section 5.4 (Appendix).

$$\overset{\triangleright}{\boldsymbol{\gamma}}_{ex}^{(i')T} = \tilde{\mathbf{w}}_1^T \left( \prod_{j'=i'+1}^{\ell} \tilde{\mathbf{A}}_{J \rightarrow 1}^{(j')} \right) \overset{\triangleright}{\mathbf{B}}_{ex}^{(i')} \quad i' = 1, \dots, \ell. \quad (5.23)$$

Assuming that the indicator function is a Gaussian random variable, the mean and variance of  $\tilde{\beta}^{(\ell)}$  are calculated as

$$\begin{aligned} \mathbb{E}[\tilde{\beta}^{(\ell)}] &= \frac{2}{\sigma_{ch}^2} \sum_{k=1}^a (\boldsymbol{\gamma}_{ch}^T)_k \\ &+ \sum_{i'=1}^{\ell-1} \left( \overset{\triangleleft}{\boldsymbol{\gamma}}_{ex}^{(i'+1)T} + \overset{\triangleright}{\boldsymbol{\gamma}}_{ex}^{(i')T} \right) \mathbf{m}_{ex}^{(i')} + \overset{\triangleright}{\boldsymbol{\gamma}}_{ex}^{(\ell)T} \mathbf{m}_{ex}^{(\ell)} \end{aligned} \quad (5.24)$$

$$\begin{aligned} \text{VAR}[\tilde{\beta}^{(\ell)}] &= \frac{4}{\sigma_{ch}^2} \sum_{k=1}^a (\boldsymbol{\gamma}_{ch}^T)_k^2 \\ &+ \sum_{i'=1}^{\ell-1} \left( \overset{\triangleleft}{\boldsymbol{\gamma}}_{ex}^{(i'+1)T} + \overset{\triangleright}{\boldsymbol{\gamma}}_{ex}^{(i')T} \right) \boldsymbol{\Sigma}_{ex}^{(i')} \left( \overset{\triangleleft}{\boldsymbol{\gamma}}_{ex}^{(i'+1)} + \overset{\triangleright}{\boldsymbol{\gamma}}_{ex}^{(i')} \right) + \overset{\triangleright}{\boldsymbol{\gamma}}_{ex}^{(\ell)T} \boldsymbol{\Sigma}_{ex}^{(\ell)} \overset{\triangleright}{\boldsymbol{\gamma}}_{ex}^{(\ell)}. \end{aligned} \quad (5.25)$$

The symbol  $(\cdot)_k$  is used to represent the  $k$ th element of the vector inside the parentheses. In the above equations,  $m_{ch}$  and  $\sigma_{ch}^2$  are scalars representing the mean and variance of the channel LLRs. The  $b \times 1$  vector  $\mathbf{m}_{ex}^{(i')}$  includes the mean of the external inputs from the unsatisfied CNs at iteration  $i'$ . The  $b \times b$  matrix  $\boldsymbol{\Sigma}_{ex}^{(i')}$ , also, is the covariance matrix of the unsatisfied inputs at iteration  $i'$ . Since it is assumed different external inputs are independent,  $\boldsymbol{\Sigma}_{ex}^{(i')}$  is a diagonal matrix. The mean and variance of the inputs from unsatisfied CNs are calculated based on DE method. Also, the mis-satisfied CNs gains are calculated based on DE and the method of Section 5.1.4 and utilized in iteration dependent gain matrices involved in Equations (5.14) to (5.17). After that, the vector  $\boldsymbol{\gamma}_{ch}^T$  in (5.21) as well as the set of vectors  $\overset{\triangleleft}{\boldsymbol{\gamma}}_{ex}^{(i')T}$  and  $\overset{\triangleright}{\boldsymbol{\gamma}}_{ex}^{(i')T}$  for  $i' = 1, \dots, \ell$  in (5.22) and (5.23) are plugged into Equations (5.24) and (5.25) to find the mean and variance of the indicator function. Finally, the failure probability of an LETS,  $\mathcal{S}$ , is obtained by

$$P_e(\mathcal{S}) = \lim_{\ell \rightarrow \infty} \text{Pr}\{\tilde{\beta}^{(\ell)} < 0\} = \lim_{\ell \rightarrow \infty} Q\left(\frac{\mathbb{E}[\tilde{\beta}^{(\ell)}]}{\sqrt{\text{VAR}[\tilde{\beta}^{(\ell)}]}}\right). \quad (5.26)$$

In practice, it is required to calculate (5.26), only, for a few iterations as the ratio of the mean to square root of variance would converge within a few iterations.

In order to approximate the error floor of an LDPC code, the dominant LETSs

of the code are first partitioned into different non-isomorphic groups and, then, the LETSs within each group are once again classified based on their TSLPs into different subgroups. Finally, using the union bound, the error floor of an LDPC code can be estimated as

$$P_f \approx \sum_i \Upsilon_i P_e(\mathbf{s}_i), \quad (5.27)$$

where  $\mathbf{s}_i$  is the representative of the  $i$ th TS subgroup with the multiplicity of  $\Upsilon_i$ .

## 5.2 Optimization of the Column Orders

Generally, changing the layer permutation of column layered decoder, potentially, results in

- 1) Changing of the model matrices, in particular, transition matrix  $\tilde{\mathbf{A}}_{J \rightarrow 1}$  and its dominant eigenvalue  $\tilde{r}$ .
- 2) Notable change in the TSLP and thus the distributions of the external messages entering the TS.

As a result, different column layered schedules can potentially produce error floors that are considerably different. The layered state-space model of the column layered decoders not only can be used to analyse the effect of different column schedules, but it can be utilized to optimize the error floor performance.

One of the main source of complexity in state-space model is the density evolution part based on which the inputs of the model are determined. For an LDPC code with  $m_b \times n_b$  base matrix  $\mathbf{H}_b$ , there exist  $n_b! = n_b \times (n_b - 1) \times \dots \times 1$  different column permutations. So, this is not practical to run the DE for every permutation of layers. In an accurate DE technique of the column layered decoders of quasi cyclic codes, the probability distributions of CNs to VNs and VNs to CNs messages flowing on different edges of the base-graph are computed while the order in which different messages (distributions) are updated following a certain updating order of layers is also taken into account. To simplify the search among the  $n_b!$  different column layered schedules, rather than the derivation of such distributions for each schedule, we select one schedule, say the one corresponding to the original order of the columns, and then at each iteration  $\ell$  and for each layer  $j$ , derive the average distributions of CN to VN messages and VN to CN messages, denoted by  $\overleftarrow{\psi}_\ell^j$  and  $\overrightarrow{\psi}_\ell^j$ , respectively, where

the average is taken over all the corresponding distributions within column layer  $L_j$ . These average distributions are then used to represent all the CN to VN and VN to CN distributions in the  $j$ th layer of decoding regardless of the schedule.

**Example 16.** *As an example, for the base graph of Fig. 5.3, the distribution  $\overleftarrow{\psi}_\ell^2$  is the average of  $\psi_\ell^{[2\leftarrow 1]}$  and  $\psi_\ell^{[2\leftarrow 3]}$ , and  $\overrightarrow{\psi}_\ell^2$  is the average of  $\psi_\ell^{[2\rightarrow 1]}$  and  $\psi_\ell^{[2\rightarrow 3]}$  (The last two distributions have not been shown in the figure).*

Naturally, by changing the order of columns, the corresponding distributions of different edges are changed. While applying another average over a couple of different column permutations would be a more accurate approach, we found it sufficient to, only, consider the DE result of the regular order of the layers of a code, for our purpose.

In the next step, by using the average distribution of VNs to CNs per each layer,  $\overrightarrow{\psi}_\ell^j$ , a partial gain similar to Equations (5.9) and (5.10) can be obtained for each layer. Then, based on the TSLP of an ETS, for a specific permutation of column layers, Equation (5.8) can be utilized to obtain the multiplicative gains of various missatisfied CNs. It is noted that, to avoid unnecessary computations and reducing the complexity, the polarity inversion coefficient is not considered in the gain calculations during the optimization process. The advantage of using Equation (5.8) is that the partial gains related to different layers can be computed before the optimization process and different schedules would only affect the parameters  $\overleftarrow{\tau}_i$  and  $\overrightarrow{\tau}_i$  related to different state-variables passing through various missatisfied CNs. With respect to unsatisfied CNs, the average CNs to VNs distributions  $\overleftarrow{\psi}_\ell^j$  or  $\overleftarrow{\psi}_{\ell-1}^j$  can be used to compute the unsatisfied CNs input parameters. This is clarified in the next example.

**Example 17.** *Regarding the quasi-cyclic Tanner (155, 64) code, there are 5 column layers. So, 5 average VN to CN distributions denoted by  $\overrightarrow{\psi}_\ell^1$ ,  $\overrightarrow{\psi}_\ell^2$ ,  $\overrightarrow{\psi}_\ell^3$ ,  $\overrightarrow{\psi}_\ell^4$  and  $\overrightarrow{\psi}_\ell^5$  are obtained. Based on these distributions, for each column layer, a VN to CN partial gain is computed. In Fig. 5.2, the obtained partial gains are plugged into Equations (5.12) and (5.13) to find the corresponding missatisfied CN gains at layers  $L'_1 = L_1$  and  $L'_2 = L_3$ , respectively.*

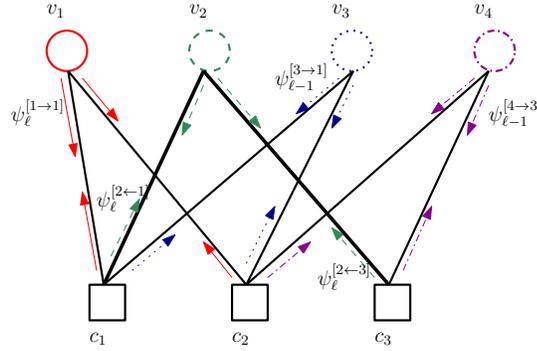
*Regarding the unsatisfied CN  $c_2$  in Fig. 5.1, the CN to VN message from  $c_2$  toward  $v_4$ , within the model, is used when the state-variables number 9 and 12 are updated. These state-variables belong to the column layers  $L'_2 = L_3$  and  $L'_3 = L_5$ , respectively.*

*As both of these layers are updated after  $L'_1 = L_1$  which is the corresponding column layer of  $v_4$ , the average distribution  $\overleftarrow{\psi}_\ell^1$  (not  $\overleftarrow{\psi}_{\ell-1}^1$ ) is used within the model to obtain the unsatisfied input parameters (mean and variance) corresponding to  $c_2$ .*

In column layered decoders, unlike the row layered decoders, the number of layers,  $n_b$ , is typically large. For example, the Wimax code, denoted by  $\mathcal{C}_2$  in the following section, has a  $6 \times 24$  base matrix. So, the total number of column layer permutations is  $24!$  which is a huge number. This means an exhaustive search among different column layer permutations is not feasible. In Chapter 4, it has been shown that the failure probability of ETS structures, on average, decreases with the reduction of dominant eigenvalue of the transition matrices corresponding to different schedules. We also have observed the same behaviour in column layered decoders. So, we will use this result to shrink the search space.

Generally, different ETSs can contribute to the error floor of an LDPC code. Among different ETSs of a code, the ETS structure that contributes the most in the error floor is called dominant ETS. Changing the column decoding schedule can potentially impact the harmfulness of different TSs including the most harmful ones. Moreover, the relative harmfulness of ETSs based on different scheduling can be changed. In order to reduce the search space of optimization process, we focus on the most harmful structure of a specific schedule, say the one corresponding to the original order of the columns. After identifying the dominant structure, the permutations of layers inside the dominant structure that result in the lowest dominant eigenvalues of the transition matrix are determined. Finally, a number of column schedules are generated randomly such that the corresponding layers order in the most dominant structure lead to the lowest possible dominant eigenvalue of the transition matrix found in the previous step.

In order to find the schedule with the lowest error floor, we perform the search in two steps. In the first step, we use the the approximate DE as well as the reduced search space, discussed above, to find a few candidate schedules that have lower error floors. In the second step, we consider the exact DE method within the linear state-space model and once again examine the corresponding error floor of the candidate schedules obtained in the previous step. Finally, the column schedule with the lowest error floor is found.



**Figure 5.3:** Base graph of Example 16: The solid (red), dashed (green), dotted (blue) and dashed-dotted (purple) arrows, respectively, illustrate the updating order which the column layered decoder perform, accordingly.

### 5.3 Simulation Results

In this section, the performance of our proposed model is investigated via simulation results. For this purpose, two QC-LDPC codes are used, namely,  $\mathcal{C}_1$  and  $\mathcal{C}_2$ . The LDPC code  $\mathcal{C}_1$  is a rate-0.3 variable-regular (640, 192) LDPC with  $d_v = 5$  and irregular CN degrees [3] and  $\mathcal{C}_2$  is a rate-0.75 irregular (576, 432) code used in Wimax standard [82].

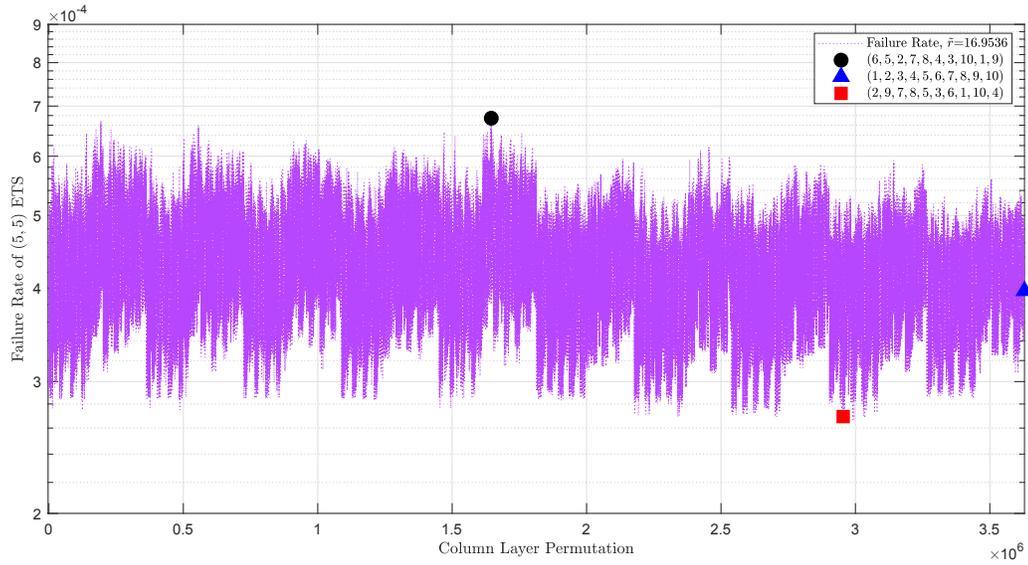
We now investigate the effect of column block permutations on the error floor of  $\mathcal{C}_1$ .  $\mathcal{C}_1$  has 10 column layers, each corresponding to one of the column blocks of the parity-check matrix in Fig. 4.6a. We label these column blocks with numbers 1 to 10 based on their original order in Fig. 4.6a, i.e., the first column block is labeled by 1, the second by 2, and so on. Different schedules can, then, be represented with different permutations of numbers from 1 to 10. For example, permutation (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) refers to the updating schedule of column layers in the same order as they appear in Fig. 4.6a.

The dominant LETS of  $c_1$  is a (5, 5) structure with multiplicity of 64. Also, all the (5, 5) LETSs have the same TSLP. In order to investigate the effect of different column schedules on the error floor performance of  $\mathcal{C}_1$ , the technique of Section 5.2 is utilized. The total number of possible column permutations for  $\mathcal{C}_1$  is  $10! = 3628800$ . These permutations result in different transition matrices for the (5, 5) LETS within the state-space model. However, unlike the row layered decoding schedules of this code presented in Section 4.3 where different row schedules produced 8 different dominant eigenvalues, all the column schedules would result in transition matrices whose

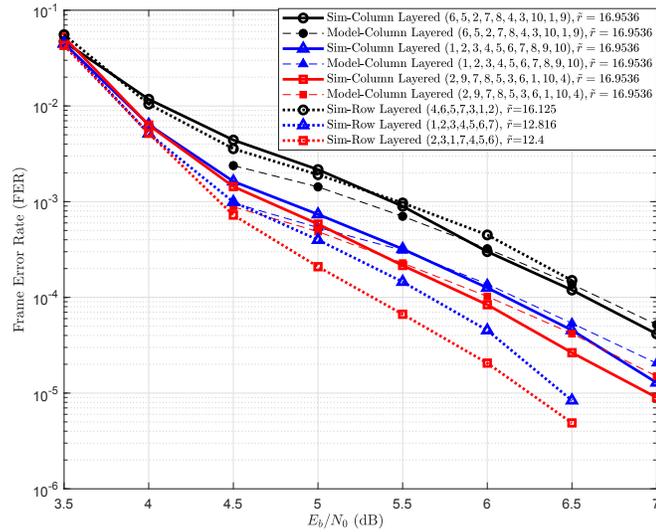
dominant eigenvalues are equal to  $\tilde{r} = 16.9536$ . By using a single application of DE, as discussed in Section 5.2, the failure rate of the (5, 5) LETS for different schedules is approximated. The results for SNR of 6 dB and saturation level of 15.75 are provided in Fig. 5.4 for all the schedules.

To find a schedule with low error floor, in the next step, we choose a few schedules leading to the lowest error rates in the first step. We then apply our estimation technique accurately to find the one that has the lowest error floor. As a result, we obtain the schedule corresponding to the permutation (2, 9, 7, 8, 5, 3, 6, 1, 10, 4), shown in Fig. 5.4 with a full square. For comparison, we have also selected one of the schedules with the worst error floor, (6, 5, 2, 7, 8, 4, 3, 10, 1, 9), as well as the original one, (1, 2, 3, 4, 5, 6, 7, 8, 9, 10). These schedules are specified in Fig. 5.4 by a full circle and a full triangle, respectively. The simulation and estimation results of these three schedules are presented in Fig. 5.5. The maximum number of iterations and the saturation level are  $I_{max} = 30$  and 15.75, respectively. For comparison, we have also presented the FER of the different row layered schedules, including regular, best and worst schedules, proposed in Section 4.3, in Fig. 5.5. As can be seen, all the estimation results match closely with the corresponding simulations. However, unlike the row layered schedules where there is a substantial difference between the FER of the best and worst layered schedules in the error floor region, the difference of the FER of the best and worst layered schedules of column layered decoder is less than an order of magnitude. This can be attributed to the fact that the dominant eigenvalues of various permutations of column layered decoder is fixed and the error floor performance is mainly determined by external messages entering the ETS subgraph. It is of importance to note that the dominant eigenvalue of different permutations in column layered decoder is  $\tilde{r} = 16.9536$  which is more than the maximum eigenvalue among row layered schedules, i.e., 16.125. Therefore, based on the results of Section 4.3 in Chapter 4, it is expected that the column layered decoding of  $\mathcal{C}_1$ , on average, to have worse performance than the row layered decoding.

As the next example, we consider  $\mathcal{C}_2$ . The multiplicity of different  $(a, b)$  LETSs of this code within the range  $a \leq 8$  and  $b \leq 2$  can be found in Table 4.1. The dominant structures of this code are from (7, 1) class. Taking the TSLPs of ETSs into account, there are ten (7, 1) LETSs, each with multiplicity of 24, that can have



**Figure 5.4:** The error estimation related to  $(5, 5)$  LETS of  $\mathcal{C}_1$  for various column layered schedules at  $E_b/N_0 = 6$  dB. All the schedules result in the same dominant eigenvalue of the layered transition matrix,  $\tilde{r} = 16.9536$ . The saturation level is 15.75.

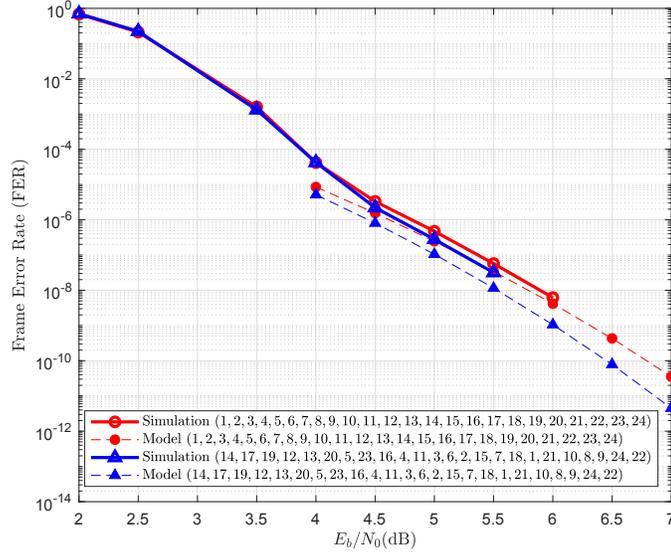


**Figure 5.5:** The simulation and estimation results related to code  $\mathcal{C}_1$  showing the effect of different schedules. The saturation level is 15.75 and the maximum number of iterations is 30.

different failure probabilities under column layered decoding. As it has been discussed in Section 5.2, due to the large number of column blocks ( $n_b = 24$ ) of this code, shown in Fig. 4.6b, we need to reduce the search space to be able to perform optimization. We found that for the original order of column layers, one of the  $(7, 1)$  LETSs, denoted by  $(7, 1)_{10}$  in Section 4.3, is the most harmful structure of this code. The dominant eigenvalue of this structure related to the regular schedule of Fig. 4.6b is  $\tilde{r} = 7.3547$ . The number of column layers involved in  $(7, 1)_{10}$  is  $J = 7$ . Therefore, there exist  $7! = 5040$  different permutations of layers that result in transition matrices with 10 different dominant eigenvalues ranging from 6.757 to 13.877. Also, there are 854 different permutations (out of 5040) whose corresponding transition matrices have the minimum dominant eigenvalue of  $\tilde{r} = 6.757$ . For each of the 854 layer permutations of  $(7, 1)_{10}$ , we generate 100 random column layer permutations of parity check matrix such that the corresponding order of layers in  $(7, 1)_{10}$ , with  $\tilde{r} = 6.757$ , are preserved. In this way, the search space will be reduced by  $85400/24!$ . By using a single application of DE, as discussed in Section 5.2, the total failure rate due to  $(7, 1)$  LETSs for different schedules is approximated. Then a handful of schedules are chosen for the second phase of optimization, as discussed before. As a result, we obtain the schedule corresponding to the permutation  $(14, 17, 19, 12, 13, 20, 5, 23, 16, 4, 11, 3, 6, 2, 15, 7, 18, 1, 21, 10, 8, 9, 24, 22)$  as one of the best schedules. The simulation and estimation results of the optimized schedule as well as the regular one are presented in Fig. 5.6. The maximum number of iterations and the saturation level are  $I_{max} = 30$  and 15.75, respectively. It can be seen that the error floor performance of the optimized schedule is slightly better than the original column permutations of Fig. 4.6b. In fact, the original order of column layers, itself, is among the good schedules of  $\mathcal{C}_2$  in terms of error floor performance. Regarding  $\mathcal{C}_2$ , we, also, have observed that the optimized FER curve of the column layered decoder and the optimized FER curve of row layered decoder in Section 4.3, interestingly, have almost the same performance in the error floor region and their curves are completely overlapped (thus not shown in Fig. 5.6).

## 5.4 Appendix

**Modification to the Proof of Proposition 3 for Column Layered Decoders.** Suppose the systematic form of the flooding schedule transition matrix obtained based



**Figure 5.6:** The simulation and estimation results related to code  $\mathcal{C}_2$  showing the effect of different schedules. The saturation level is 15.75. The maximum number of iterations is 30.

on row and column layered schedules be denoted by  $\mathbf{A}_r$  and  $\mathbf{A}_c$ , respectively. Then, there exist a permutation matrix  $\mathbf{P}_1$  such that

$$\mathbf{A}_r = \mathbf{P}_1 \mathbf{A}_c \mathbf{P}_1^T \quad (5.28)$$

where  $\mathbf{P}_1^T = \mathbf{P}_1^{-1}$ . In Lemma 11, a symmetric unitary permutation matrix,  $\mathbf{P}$ , was introduced such that

$$\mathbf{A}_r = \mathbf{P} \mathbf{A}_r^T \mathbf{P}. \quad (5.29)$$

By substitution of (5.28) into (5.29) we have

$$\mathbf{P}_1 \mathbf{A}_c \mathbf{P}_1^T = \mathbf{P} \mathbf{P}_1 \mathbf{A}_c^T \mathbf{P}_1^T \mathbf{P} \quad (5.30)$$

or equivalently

$$\mathbf{A}_c = \mathbf{P}_1^T \mathbf{P} \mathbf{P}_1 \mathbf{A}_c^T \mathbf{P}_1^T \mathbf{P} \mathbf{P}_1. \quad (5.31)$$

By defining the new unitary permutation matrix  $\mathbf{P}' = \mathbf{P}_1^T \mathbf{P} \mathbf{P}_1$  where  $\mathbf{P}' \mathbf{P}' = \mathbf{I}$  we can write

$$\mathbf{A}_c = \mathbf{P}' \mathbf{A}_c^T \mathbf{P}'. \quad (5.32)$$

In the proof of Proposition 3 in Section 4.4, it is sufficient to replace the matrix  $\mathbf{P}$  with  $\mathbf{P}' = \mathbf{P}_1^T \mathbf{P} \mathbf{P}_1$  and use the same steps to prove that the corresponding dominant eigenvalue of the transition matrix in column layered schedule is invariant to the reversing of the layers order.

## Chapter 6

# A Semi Linear State Space Model for Error Floor Estimation of LDPC codes over the AWGN Channel

The chapter is organized as follows: We develop the semi-linear state-space model of an ETS in Section 6.1. In the same section, we also establish connections between our proposed model and the semi-analytical method of [34]. In Section 6.2, we provide extensive simulation results to demonstrate the accuracy of the proposed method in comparison with the linear state-space model as well as semi-analytical method of [34].

## 6.1 Proposed Semi-linear Model with Dynamical Operating Point

### 6.1.1 Main Ideas and Methodology

One of the main factors that affects the accuracy of the Taylor expansion of a non-linear function is the choice of operating point around which the function is approximated. A proper choice can significantly improve the accuracy of the approximation. For the case of the approximation of the missatisfied CN operation, which is of interest in this chapter, the function inputs are random, and a good choice is not only dependent on the iteration number but also depends on individual missatisfied CNs within the TS. To simplify the method, in this chapter, however, we ignore the latter dependency, and dynamically find the operating point for all missatisfied CNs of the

TS at iteration  $\ell$  as the weighted average of internal messages of the TS at iteration  $\ell - 1$ . In this process of averaging, it is essential to capture the most likely inputs that cause TS failures with a reasonable complexity. For this, we develop our state-space model with a condition on the noise vector, hence the term “conditional state-space model.” The conditioning is to bias the noise vector to move in the direction that encourages more TS failures.

The first order Taylor expansion of missatisfied CN functions results in a linear state-space model. In order to increase the accuracy of the approximations, we will use the second order Taylor expansion in the first few iterations and first order expansion for the remaining iterations. Thus, our model is called “semi-linear.” The reason for a higher order Taylor expansion at early iterations is that for those iterations, the extrinsic inputs to missatisfied CNs have not sufficiently grown yet, and as a result, their size is comparable to that of internal messages. This implies that the relative degree of non-linearity of missatisfied CNs at early iterations is higher.

In the linear state-space model of [45], it is assumed that the error indicator function is a Gaussian random variable. However, we will show that in our dynamical model which is a closer model to the actual non-linear model, if the same indicator function is used to calculate the failure probability, the Gaussian approximation is only applicable in initial iterations. In higher iterations, the indicator function has a bimodal distribution which can be approximated as a mixture of two Gaussian distributions. We incorporate this distribution into the derivations of our model parameters.

### 6.1.2 Proposed Model

The operating point of the Taylor expansion of a function with random input is typically chosen as the expected value of the random input. In TSs, and in the context of the Taylor expansion of missatisfied CN functions with respect to state variables, one needs to calculate the expected values of the state variables. Now by making the simplifying assumption that all the missatisfied CNs within a TS have the same operating point at a given iteration  $\ell$ , we use a weighted average of state variables at iteration  $\ell - 1$ , as defined below, as the operating point for iteration  $\ell$ .

**Definition 3.** *The operating point of the Taylor expansions of missatisfied CNs of*

an ETS at iteration  $\ell$  is defined by

$$\eta_{\ell-1} \triangleq \mathbb{E}\{\hat{\mathbf{w}}_1^T \mathbf{x}^{(\ell-1)}\}, \quad (6.1)$$

in which the weighting vector  $\hat{\mathbf{w}}_1$  is a scaled version of the positive dominant left eigenvector of the transition matrix  $\mathbf{A}$  given by

$$\hat{\mathbf{w}}_1 = \frac{\mathbf{w}_1}{\sum_{i=1}^{m_s} (\mathbf{w}_1)_i}, \quad (6.2)$$

where the symbol  $(\mathbf{w}_1)_i$  is used to denote the  $i$ th element of  $\mathbf{w}_1$ .

We note that the weighting vector  $\hat{\mathbf{w}}_1$  is also a dominant left eigenvector of  $\mathbf{A}$ . This implies that the weighted average in (6.1) is in fact the error indicator function. This means that the calculation of the operating point comes at no extra complexity since the expected value of (6.1) is required in the calculation of the TS failure probability anyway. As we will show later, the calculation of (6.1) is performed recursively in consecutive iterations. We further note that for the ETSs whose internal VN degrees  $\zeta_{v_i}$  are equal, since  $\mathbf{w}_1$  is an all-one vector [45], then (6.1) is equivalent to a regular average of the state variables.

While Definition 3 can be used to find the operating points of missatisfied CNs of an ETS under the normal operation of the decoder, in this work, we are interested in calculating the operating points for inputs that are highly likely to cause a decoder failure over the ETS. For this, we impose a condition on the noise vector that biases the decoder towards having more failures on the ETS of interest, as explained in the following.

The input vector of a TS from the channel can be written as

$$\mathbf{L} = \frac{2}{\sigma_{ch}^2} (\mathbf{1}_{a \times 1} + \mathbf{n}_a), \quad (6.3)$$

where  $\mathbf{1}_{a \times 1}$  is an all-one vector of size  $a$ . The  $a$ -dimensional Gaussian noise vector,  $\mathbf{n}_a$ , can be expanded on orthonormal vectors  $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_a$  as

$$\mathbf{L} = \frac{2}{\sigma_{ch}^2} (\mathbf{1}_{a \times 1} + \theta_1 \boldsymbol{\gamma}_1 + \theta_2 \boldsymbol{\gamma}_2 + \dots + \theta_a \boldsymbol{\gamma}_a), \quad (6.4)$$

where i.i.d random variables  $\theta_1, \dots, \theta_a$ , each with distribution  $\mathcal{N}(0, \sigma_{ch}^2)$ ,<sup>1</sup> are the

<sup>1</sup>Notation  $\mathcal{N}(m, \sigma^2)$  is used to denote a Gaussian distribution with expected value  $m$  and variance

projections of  $\mathbf{n}_a$  on different orthonormal vectors. Now suppose that the first vector is chosen as  $\boldsymbol{\gamma}_1 = (1/\sqrt{a})\mathbf{1}_{a \times 1}$ , which is known to be a good approximation for the error direction of the TS [34]. In this case, we have

$$\mathbf{L} = \frac{2}{\sigma_{ch}^2}(\mathbf{1}_{a \times 1} + \underbrace{\frac{\theta_1}{\sqrt{a}} \mathbf{1}_{a \times 1}}_{\theta'}) + \underbrace{\frac{2}{\sigma_{ch}^2}(\theta_2 \boldsymbol{\gamma}_2 + \cdots + \theta_a \boldsymbol{\gamma}_a)}_{\mathbf{n}'}, \quad (6.5)$$

where  $\theta' \sim \mathcal{N}(0, \sigma_{ch}^2/a)$ . We refer to  $\mathbf{n}'$  as *residual noise vector*. The conditional mean of the vector  $\mathbf{L}$  given  $\theta'$  can then be obtained as

$$\mathbf{m}_{\mathbf{L}|\theta'} = \frac{2}{\sigma_{ch}^2}(1 + \theta')\mathbf{1}_{a \times 1}. \quad (6.6)$$

Also, the corresponding conditional covariance matrix  $\boldsymbol{\Sigma}_{\mathbf{L}|\theta'}$  of the channel vector, which is equal to the covariance matrix  $\boldsymbol{\Sigma}_{\mathbf{n}'}$  of the residual noise, can be obtained as

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{L}|\theta'} &= \boldsymbol{\Sigma}_{\mathbf{n}'} \\ &= \frac{4}{\sigma_{ch}^2} \underbrace{(\boldsymbol{\gamma}_2 \boldsymbol{\gamma}_2^T + \boldsymbol{\gamma}_3 \boldsymbol{\gamma}_3^T + \cdots + \boldsymbol{\gamma}_a \boldsymbol{\gamma}_a^T)}_{\mathbf{I}_a - \boldsymbol{\gamma}_1 \boldsymbol{\gamma}_1^T} \\ &= \frac{4}{\sigma_{ch}^2}(\mathbf{I}_a - \frac{1}{a}\mathbf{1}_{a \times a}), \end{aligned} \quad (6.7)$$

where  $\mathbf{I}_a$  and  $\mathbf{1}_{a \times a}$  are identity and all-one matrices with size  $a \times a$ , respectively. Therefore, the variance of the residual noise elements (diagonal elements of  $\boldsymbol{\Sigma}_{\mathbf{n}'}$ ) are equal to

$$\sigma_{n'}^2 = \frac{4}{\sigma_{ch}^2}(1 - \frac{1}{a}). \quad (6.8)$$

Following (6.1), we now define the conditional operating point of Taylor expansions for state variables at a given iteration  $\ell$  as

$$\eta_{\ell-1|\theta'} \triangleq \mathbb{E}\{\dot{\mathbf{w}}_1^T \mathbf{x}^{(\ell-1)} | \theta'\}. \quad (6.9)$$

Based on (6.9) and (6.6), the initial conditional operating point can be obtained as

$$\eta_{0|\theta'} = \frac{2}{\sigma_{ch}^2}(1 + \theta'). \quad (6.10)$$

By changing  $\theta'$  from zero to  $-2$ ,  $\eta_{0|\theta'}$  changes from  $2/\sigma_{ch}^2$  to  $-2/\sigma_{ch}^2$ . Accordingly, based on (6.6), the conditional mean of the channel input vector moves from  $2/\sigma_{ch}^2 \mathbf{1}_{a \times 1}$  towards  $-2/\sigma_{ch}^2 \mathbf{1}_{a \times 1}$ . This would result in an increase in the failure probability of the TS.

Finally, the failure probability of a TS  $\mathcal{S}$  can be calculated using total probability law as

$$P_e\{\mathcal{S}\} = \frac{\sqrt{a}}{\sqrt{2\pi\sigma_{ch}^2}} \int_{-\infty}^{+\infty} P_e\{\mathcal{S}|\theta'\} e^{-\frac{a\theta'^2}{2\sigma_{ch}^2}} d\theta'. \quad (6.11)$$

We note that the integrand in (6.11) has a maximum and decays quickly with  $\theta'$  in the vicinity of that maximum. The integral is thus largely determined by the value of the function around the region of  $\theta'$  that includes the maximum value of the function. Typically, to have an accurate approximation of  $P_e\{\mathcal{S}\}$ , it is sufficient to compute the integral for  $-2 \leq \theta' \leq -1$  [34].

In the following, we focus on calculating the conditional failure probability,  $P_e\{\mathcal{S}|\theta'\}$ , using a *conditional semi-linear state-space model*. The initial states of the model that we develop are the conditional channel inputs (6.5) with different but given values of  $\theta'$ , hence the term “conditional.” In this model, the missatisfied CN operations are approximated by second order Taylor expansions at the first two iterations, and by first order Taylor expansions for the rest of the iterations, hence the term “semi-linear.”

Suppose the operator  $\mathbf{x}^{\odot 2}$  denotes an element-wise second power of a vector  $\mathbf{x}$ . The second order Taylor expansion of the missatisfied CN function in (3.2), around the conditional operating point  $\eta_{\ell-1|\theta'}$ , can be written as

$$\Phi(\mathbf{x}^{(\ell-1)}, \mathbf{B}_s \mathbf{L}_s^{(\ell)}) \approx \dot{\mathbf{g}}^{(\ell)} + \hat{\mathbf{G}}^{(\ell)} (\mathbf{x}^{(\ell-1)} - \eta_{\ell-1|\theta'} \mathbf{1}_{m_s \times 1}) + \check{\mathbf{G}}^{(\ell)} (\mathbf{x}^{(\ell-1)} - \eta_{\ell-1|\theta'} \mathbf{1}_{m_s \times 1})^{\odot 2}. \quad (6.12)$$

The vector  $\dot{\mathbf{g}}^{(\ell)}$  and diagonal matrices,  $\hat{\mathbf{G}}^{(\ell)}$  and  $\check{\mathbf{G}}^{(\ell)}$  contain the coefficients corresponding to the constant, linear and squared terms of the second order Taylor expansion. The  $m_s \times 1$  random vector  $\dot{\mathbf{g}}^{(\ell)}$  with random elements  $\dot{g}_j^{(\ell)}$  (mean:  $\bar{g}_j^{(\ell)}$ , variance:  $\sigma_{\dot{g}_j^{(\ell)}}^2$ ) is computed as

$$\dot{\mathbf{g}}^{(\ell)} = \Phi(\eta_{\ell-1|\theta'} \mathbf{1}_{m_s \times 1}, \mathbf{B}_s \mathbf{L}_s^{(\ell)}), \quad (6.13)$$

and the  $m_s \times m_s$  diagonal random gain matrices  $\hat{\mathbf{G}}^{(\ell)}$  and  $\check{\mathbf{G}}^{(\ell)}$  with elements  $\hat{g}_j^{(\ell)}$

(mean:  $\bar{g}_j^{(\ell)}$ , variance:  $\sigma_{\hat{g}_j}^2$ ) and  $\check{g}_j^{(\ell)}$  (mean:  $\bar{g}_j^{(\ell)}$ , variance:  $\sigma_{\check{g}_j}^2$ ) are calculated as

$$\hat{\mathbf{G}}^{(\ell)} = \mathbf{diag}\{\Phi_x(\eta_{\ell-1|\theta'} \mathbf{1}_{m_s \times 1}, \mathbf{B}_s \mathbf{L}_s^{(\ell)})\}, \quad (6.14)$$

$$\check{\mathbf{G}}^{(\ell)} = \frac{1}{2} \mathbf{diag}\{\Phi_{xx}(\eta_{\ell-1|\theta'} \mathbf{1}_{m_s \times 1}, \mathbf{B}_s \mathbf{L}_s^{(\ell)})\}. \quad (6.15)$$

We use the notations  $\bar{\hat{\mathbf{G}}}^{(\ell)}$  and  $\bar{\check{\mathbf{G}}}^{(\ell)}$  to denote the expected value of the above random matrices with respect to missatisfied CN external inputs. The element-wise operators  $\Phi_x(\cdot)$  and  $\Phi_{xx}(\cdot)$  are the first and second partial derivatives of the CN function with respect to the state variables.

Consider a missatisfied check node with the virtual variable node  $s_j$ , and with the two incoming messages represented by state variables  $x_i$  and  $x_{i'}$  (see, Fig. 3.1). We then have

$$\dot{g}_i^{(\ell)} = \dot{g}_{i'}^{(\ell)} = \phi(\eta_{\ell-1|\theta'}, L_{s_j}^{(\ell)}), \quad (6.16)$$

$$\hat{g}_i^{(\ell)} = \hat{g}_{i'}^{(\ell)} = \phi_x(\eta_{\ell-1|\theta'}, L_{s_j}^{(\ell)}), \quad (6.17)$$

$$\check{g}_i^{(\ell)} = \check{g}_{i'}^{(\ell)} = \frac{1}{2} \phi_{xx}(\eta_{\ell-1|\theta'}, L_{s_j}^{(\ell)}), \quad (6.18)$$

where  $L_{s_j}^{(\ell)}$  is the message entering the missatisfied check node through the virtual variable node  $s_j$  at iteration  $\ell$ . Also, we have

$$\phi(x, y) = \ln \left( \frac{1 + e^{x+y}}{e^x + e^y} \right) \quad (6.19)$$

$$\phi_x(x, y) = \frac{e^x(e^{2y} - 1)}{(e^x + e^y)(1 + e^{x+y})}, \quad (6.20)$$

$$\phi_{xx}(x, y) = \phi_x(x, y) \left[ 1 - \frac{e^x}{(e^x + e^y)} - \frac{e^{x+y}}{(1 + e^{x+y})} \right]. \quad (6.21)$$

By plugging (6.12) into (3.2), and considering the conditional channel vector of (6.5) as input, the conditional state vectors can be written as

$$\mathbf{x}^{(0)} = \mathbf{B}(\eta_{0|\theta'} \mathbf{1}_{a \times 1} + \mathbf{n}'), \quad (6.22)$$

$$\mathbf{x}^{(1)} = \mathbf{A}(\dot{\mathbf{g}}^{(1)} + \hat{\mathbf{G}}^{(1)} \mathbf{B} \mathbf{n}' + \check{\mathbf{G}}^{(1)} \mathbf{B} \mathbf{n}'^{\odot 2}) + \mathbf{x}^{(0)} + \mathbf{B}_{ex} \mathbf{L}_{ex}^{(1)}, \quad (6.23)$$

$$\mathbf{x}^{(2)} = \mathbf{A}(\dot{\mathbf{g}}^{(2)} + \hat{\mathbf{G}}^{(2)} \mathbf{x}'^{(1)} + \check{\mathbf{G}}^{(2)} \mathbf{x}'^{(1)\odot 2}) + \mathbf{x}^{(0)} + \mathbf{B}_{ex} \mathbf{L}_{ex}^{(2)}, \quad (6.24)$$

$$\mathbf{x}^{(\ell)} = \mathbf{A}(\dot{\mathbf{g}}^{(\ell)} + \hat{\mathbf{G}}^{(\ell)} \mathbf{x}'^{(\ell-1)}) + \mathbf{x}^{(0)} + \mathbf{B}_{ex} \mathbf{L}_{ex}^{(\ell)}, \quad \text{for } \ell \geq 3, \quad (6.25)$$

where

$$\mathbf{x}'^{(\ell-1)} = \mathbf{x}^{(\ell-1)} - \eta_{\ell-1|\theta'} \mathbf{1}_{m_s \times 1}. \quad (6.26)$$

In the above equations, to simplify the notation, we have used the same notations for state vectors as those in the unconditional model. To derive (6.23), we have replaced  $\mathbf{x}'^{(0)}$  and  $\mathbf{x}'^{(0)\odot 2}$  with  $\mathbf{Bn}'$  and  $\mathbf{Bn}'^{\odot 2}$ , respectively. The first replacement follows from plugging (6.22) into (6.26) for  $\ell = 1$ , i.e.,  $\mathbf{x}'^{(0)} = \eta_{0|\theta'} \mathbf{B} \mathbf{1}_{a \times 1} + \mathbf{Bn}' - \eta_{0|\theta'} \mathbf{1}_{m_s \times 1} = \mathbf{Bn}'$ . The second replacement follows from the structure of matrix  $\mathbf{B}$  as discussed in Lemma 1, i.e.,  $\mathbf{x}'^{(0)\odot 2} = (\mathbf{Bn}')^{\odot 2} = \mathbf{Bn}'^{\odot 2}$ .

In the following, we simplify the conditional semi-linear state-space equations by replacing the gain matrices  $\hat{\mathbf{G}}^{(\ell)}$  and  $\check{\mathbf{G}}^{(\ell)}$  with scalars. For this, we adopt the same approach as the one used in Chapter 3 to replace the random gain matrix  $\mathbf{G}^{(\ell)}$  with the scalar gain  $g^{(\ell)}$ . (see Lemma 2 and Definition 2.)

The following lemma can be proved using the same approach as in the proof of Lemma 2 of Chapter 3, which is based on perturbation theory with the assumption that the nonzero elements of diagonal matrices  $\hat{\mathbf{G}}^{(\ell)}$  and  $\check{\mathbf{G}}^{(\ell)}$  have small variances.

**Lemma 14.** *The first order approximation of the maximum eigenvalues,  $\rho(\mathbf{A}\hat{\mathbf{G}}^{(\ell)})$  and  $\rho(\mathbf{A}\check{\mathbf{G}}^{(\ell)})$ , of matrices  $\mathbf{A}\hat{\mathbf{G}}^{(\ell)}$  and  $\mathbf{A}\check{\mathbf{G}}^{(\ell)}$ , respectively, are given by*

$$\begin{aligned} \rho(\mathbf{A}\hat{\mathbf{G}}^{(\ell)}) &\approx r \mathbf{w}_1^T \bar{\bar{\mathbf{G}}}^{(\ell)} \mathbf{u}_1 \\ &+ \sum_{i=1}^{\frac{m_s}{2}} \left. \frac{\partial \rho(\mathbf{A}\hat{\mathbf{G}}^{(\ell)})}{\partial \hat{g}_i^{(\ell)}} \right|_{\hat{\mathbf{G}}^{(\ell)} = \bar{\bar{\mathbf{G}}}^{(\ell)}} (\hat{g}_i^{(\ell)} - \bar{\bar{g}}_i^{(\ell)}), \end{aligned} \quad (6.27)$$

$$\begin{aligned} \rho(\mathbf{A}\check{\mathbf{G}}^{(\ell)}) &\approx r \mathbf{w}_1^T \bar{\check{\mathbf{G}}}^{(\ell)} \mathbf{u}_1 \\ &+ \sum_{i=1}^{\frac{m_s}{2}} \left. \frac{\partial \rho(\mathbf{A}\check{\mathbf{G}}^{(\ell)})}{\partial \check{g}_i^{(\ell)}} \right|_{\check{\mathbf{G}}^{(\ell)} = \bar{\check{\mathbf{G}}}^{(\ell)}} (\check{g}_i^{(\ell)} - \bar{\check{g}}_i^{(\ell)}), \end{aligned} \quad (6.28)$$

in which

$$\left. \frac{\partial \rho(\mathbf{A}\hat{\mathbf{G}}^{(\ell)})}{\partial \hat{g}_i^{(\ell)}} \right|_{\hat{\mathbf{G}}^{(\ell)} = \bar{\bar{\mathbf{G}}}^{(\ell)}} \approx \mathbf{w}_1^T \frac{\partial \mathbf{A}\hat{\mathbf{G}}^{(\ell)}}{\partial \hat{g}_i^{(\ell)}} \mathbf{u}_1, \quad (6.29)$$

$$\left. \frac{\partial \rho(\mathbf{A}\check{\mathbf{G}}^{(\ell)})}{\partial \check{g}_i^{(\ell)}} \right|_{\check{\mathbf{G}}^{(\ell)} = \bar{\check{\mathbf{G}}}^{(\ell)}} \approx \mathbf{w}_1^T \frac{\partial \mathbf{A}\check{\mathbf{G}}^{(\ell)}}{\partial \check{g}_i^{(\ell)}} \mathbf{u}_1, \quad (6.30)$$

and  $r$ ,  $\mathbf{w}_1$  and  $\mathbf{u}_1$  are the maximum eigenvalue and the corresponding normalized left and right eigenvectors of matrix  $\mathbf{A}$ .

Based on Lemma 14, for the conditional gains related to linear and squared terms of Taylor expansions, we define the following random scalar gains.

**Definition 4.** *The random scalar gains, corresponding to the conditional linear and squared terms in Taylor expansions of missatisfied CNs, at iteration  $\ell$  are defined as:*

$$\hat{g}^{(\ell)} = \frac{\rho(\mathbf{A}\hat{\mathbf{G}}^{(\ell)})}{r} \approx \bar{g}^{(\ell)} + \sum_{i=1}^{\frac{m_s}{2}} \frac{\partial \rho(\mathbf{A}\hat{\mathbf{G}}^{(\ell)})}{r \partial \hat{g}_i^{(\ell)}} (\hat{g}_i^{(\ell)} - \bar{g}_i^{(\ell)}), \quad (6.31)$$

$$\check{g}^{(\ell)} = \frac{\rho(\mathbf{A}\check{\mathbf{G}}^{(\ell)})}{r} \approx \bar{\check{g}}^{(\ell)} + \sum_{i=1}^{\frac{m_s}{2}} \frac{\partial \rho(\mathbf{A}\check{\mathbf{G}}^{(\ell)})}{r \partial \check{g}_i^{(\ell)}} (\check{g}_i^{(\ell)} - \bar{\check{g}}_i^{(\ell)}), \quad (6.32)$$

where  $\bar{g}^{(\ell)} = \mathbf{w}_1^T \bar{\mathbf{G}}^{(\ell)} \mathbf{u}_1$  and  $\bar{\check{g}}^{(\ell)} = \mathbf{w}_1^T \bar{\check{\mathbf{G}}}^{(\ell)} \mathbf{u}_1$  are the means of  $\hat{g}^{(\ell)}$  and  $\check{g}^{(\ell)}$ , respectively. Also, assuming gains  $\hat{g}_i^{(\ell)}$  as well as  $\check{g}_i^{(\ell)}$  are independent, the variances of  $\hat{g}^{(\ell)}$  and  $\check{g}^{(\ell)}$  at a given iteration  $\ell$  can be approximated as

$$\sigma_{\hat{g}^{(\ell)}}^2 \approx \sum_{i=1}^{\frac{m_s}{2}} \left( \frac{\partial \rho(\mathbf{A}\hat{\mathbf{G}}^{(\ell)})}{r \partial \hat{g}_i^{(\ell)}} \right)^2 \sigma_{\hat{g}_i^{(\ell)}}^2, \quad (6.33)$$

$$\sigma_{\check{g}^{(\ell)}}^2 \approx \sum_{i=1}^{\frac{m_s}{2}} \left( \frac{\partial \rho(\mathbf{A}\check{\mathbf{G}}^{(\ell)})}{r \partial \check{g}_i^{(\ell)}} \right)^2 \sigma_{\check{g}_i^{(\ell)}}^2. \quad (6.34)$$

**Remark 6.** *For the case where the distributions of the extrinsic messages of the missatisfied check nodes are identical, since  $\mathbf{w}_1^T \mathbf{u}_1 = 1$ , the mean values  $\bar{g}^{(\ell)}$  and  $\bar{\check{g}}^{(\ell)}$  are equal to the expected value of the gains related to each of the missatisfied CNs, i.e.,  $\bar{g}_i^{(\ell)}$  and  $\bar{\check{g}}_i^{(\ell)}$ , respectively.*

Using similar steps as those used in the proof of Lemma 3, and based on the assumption that the dominant left eigenvectors of random matrices  $\mathbf{A}\hat{\mathbf{G}}^{(\ell)}$  and  $\mathbf{A}\check{\mathbf{G}}^{(\ell)}$  are the same as the dominant left eigenvector of matrix  $\mathbf{A}$ , we can replace the diagonal gain matrices  $\mathbf{G}^{(\ell)}$  and  $\check{\mathbf{G}}^{(\ell)}$  within the conditional semi-linear state-space model with the random scalar gains defined in Equations (6.31) and (6.32), respectively. As a result, we have

$$\mathbf{x}^{(0)} = \mathbf{B}(\eta_{0|\theta'} \mathbf{1}_{a \times 1} + \mathbf{n}'), \quad (6.35)$$

$$\mathbf{x}^{(1)} = \mathbf{A}(\dot{\mathbf{g}}^{(1)} + \hat{\mathbf{g}}^{(1)} \mathbf{B} \mathbf{n}' + \check{\mathbf{g}}^{(1)} \mathbf{B} \mathbf{n}'^{\odot 2}) + \mathbf{x}^{(0)} + \mathbf{B}_{ex} \mathbf{L}_{ex}^{(1)}, \quad (6.36)$$

$$\mathbf{x}^{(2)} = \mathbf{A}(\dot{\mathbf{g}}^{(2)} + \hat{\mathbf{g}}^{(2)} \mathbf{x}'^{(1)} + \check{\mathbf{g}}^{(2)} \mathbf{x}'^{(1)\odot 2}) + \mathbf{x}^{(0)} + \mathbf{B}_{ex} \mathbf{L}_{ex}^{(2)}, \quad (6.37)$$

$$\mathbf{x}^{(\ell)} = \mathbf{A}(\dot{\mathbf{g}}^{(\ell)} + \hat{\mathbf{g}}^{(\ell)} \mathbf{x}'^{(\ell-1)}) + \mathbf{x}^{(0)} + \mathbf{B}_{ex} \mathbf{L}_{ex}^{(\ell)}, \quad \text{for } \ell \geq 3. \quad (6.38)$$

Finally, in the following, we derive the error indicator function  $\acute{\xi}_\ell$  of our model at iteration  $\ell$ ,<sup>2</sup> as the projection of the conditional state vector at iteration  $\ell$  on the normalized left eigenvector  $\acute{\mathbf{w}}_1^T$  of matrix  $\mathbf{A}$ :

$$\acute{\xi}_0 = \eta_{0|\theta'} + \acute{\mathbf{w}}_1^T \mathbf{B} \mathbf{n}', \quad (6.39)$$

$$\acute{\xi}_1 = r(\dot{\mathbf{g}}^{(1)} + \hat{\mathbf{g}}^{(1)} \acute{\mathbf{w}}_1^T \mathbf{B} \mathbf{n}' + \check{\mathbf{g}}^{(1)} \acute{\mathbf{w}}_1^T \mathbf{B} \mathbf{n}'^{\odot 2}) + \acute{\xi}_0 + \acute{\xi}_1^{ex}, \quad (6.40)$$

$$\acute{\xi}_2 = r(\dot{\mathbf{g}}^{(2)} + \hat{\mathbf{g}}^{(2)} \acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)} + \check{\mathbf{g}}^{(2)} \acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)\odot 2}) + \acute{\xi}_0 + \acute{\xi}_2^{ex}, \quad (6.41)$$

$$\acute{\xi}_\ell = r(\dot{\mathbf{g}}^{(\ell)} + \hat{\mathbf{g}}^{(\ell)} \acute{\mathbf{w}}_1^T \mathbf{x}'^{(\ell-1)}) + \acute{\xi}_0 + \acute{\xi}_\ell^{ex}, \quad \text{for } \ell \geq 3, \quad (6.42)$$

where

$$\acute{\xi}_\ell^{ex} = \acute{\mathbf{w}}_1^T \mathbf{B}_{ex} \mathbf{L}_{ex}^{(\ell)}, \quad (6.43)$$

with mean and variance equal to

$$\mathbb{E}\{\acute{\xi}_\ell^{ex}\} = m_{ex}^{(\ell)} \sum_{k=1}^b (\acute{\mathbf{w}}_1^T \mathbf{B}_{ex})_k, \quad (6.44)$$

$$\mathbb{V}\text{AR}\{\acute{\xi}_\ell^{ex}\} = \sigma_{ex}^{2(\ell)} \sum_{k=1}^b (\acute{\mathbf{w}}_1^T \mathbf{B}_{ex})_k^2. \quad (6.45)$$

To derive (6.39), we have replaced  $\eta_{0|\theta'} \acute{\mathbf{w}}_1^T \mathbf{B} \mathbf{1}_{a \times 1}$  with  $\eta_{0|\theta'}$ , since  $\mathbf{B} \mathbf{1}_{a \times 1} = \mathbf{1}_{m_s \times 1}$ . Also, the random variable  $\dot{\mathbf{g}}^{(\ell)}$  is given by

$$\dot{\mathbf{g}}^{(\ell)} = \acute{\mathbf{w}}_1^T \dot{\mathbf{g}}^{(\ell)}. \quad (6.46)$$

---

<sup>2</sup>In Chapter 3, the notation  $\xi_\ell$  (without accent at the top) is used to denote the unconditional error indicator function.

It is noted that since the elements of the  $m_s \times 1$  column vector  $\dot{\mathbf{g}}^{(\ell)}$  corresponding to the same missatisfied CN are the same, the vector  $\dot{\mathbf{g}}^{(\ell)}$  can, equivalently, be written as the multiplication of matrix  $\mathbf{B}_s$  and an  $\frac{m_s}{2} \times 1$  vector containing the  $\frac{m_s}{2}$  distinct elements of  $\dot{\mathbf{g}}^{(\ell)}$ . If the inputs to missatisfied CNs (from virtual VNs) at a given iteration  $\ell$  are assumed to be i.i.d, the mean,  $\bar{g}_j^{(\ell)}$ , and variance,  $\sigma_{g_j^{(\ell)}}^2$ , of the elements of  $\dot{\mathbf{g}}^{(\ell)}$  are equal. So, after dropping the index  $j$ , the mean and variance of  $\dot{\mathbf{g}}^{(\ell)}$  in (6.46), denoted by  $\bar{\mathbf{g}}^{(\ell)}$  and  $\sigma_{\mathbf{g}^{(\ell)}}^2$ , are calculated as follows:

$$\bar{\mathbf{g}}^{(\ell)} = \bar{g}^{(\ell)} \sum_{k=1}^{\overbrace{m_s/2}^{=1}} (\dot{\mathbf{w}}_1^T \mathbf{B}_s)_k, \quad (6.47)$$

$$\sigma_{\mathbf{g}^{(\ell)}}^2 = \sigma_{g^{(\ell)}}^2 \sum_{k=1}^{m_s/2} (\dot{\mathbf{w}}_1^T \mathbf{B}_s)_k^2. \quad (6.48)$$

The reason for the summation in the right hand side of (6.47) being equal to 1 is that according to Lemma 1, each column of  $\mathbf{B}_s$  has two non-zero elements and that the non-zero elements of different columns do not overlap.

**Lemma 15.** *The mean and variance of the initial conditional error indicator function,  $\xi_0 \triangleq \dot{\mathbf{w}}_1^T \mathbf{x}^{(0)}$ , are given by*

$$\mathbb{E}\{\xi_0\} = \eta_{0|\theta'}, \text{VAR}\{\xi_0\} = \frac{4}{\sigma_{ch}^2} \left[ \sum_{k=1}^a (\dot{\mathbf{w}}_1^T \mathbf{B})_k^2 - \frac{1}{a} \right]. \quad (6.49)$$

*Proof.* From (6.39), we have

$$\mathbb{E}\{\xi_0\} = \eta_{0|\theta'} + \dot{\mathbf{w}}_1^T \mathbf{B} \overbrace{\mathbb{E}\{\mathbf{n}'\}}^{=0} = \eta_{0|\theta'}. \quad (6.50)$$

For the variance,

$$\text{VAR}\{\xi_0\} = \dot{\mathbf{w}}_1^T \mathbf{B} \Sigma_{\mathbf{n}'} \mathbf{B}^T \dot{\mathbf{w}}_1. \quad (6.51)$$

Substituting  $\Sigma_{\mathbf{n}'}$  from (6.7) into the above equation results in

$$\mathbb{V}\text{AR}\{\xi_0\} = \frac{4}{\sigma_{ch}^2} \hat{\mathbf{w}}_1^T \mathbf{B} (\mathbf{I}_a - \frac{1}{a} \mathbf{1}_{a \times a}) \mathbf{B}^T \hat{\mathbf{w}}_1, \quad (6.52)$$

$$= \frac{4}{\sigma_{ch}^2} \left[ \hat{\mathbf{w}}_1^T \mathbf{B} \mathbf{I}_a \mathbf{B}^T \hat{\mathbf{w}}_1 - \frac{1}{a} \overbrace{\hat{\mathbf{w}}_1^T \mathbf{B} \mathbf{1}_{a \times 1}}{=1} \overbrace{\mathbf{1}_{1 \times a} \mathbf{B}^T \hat{\mathbf{w}}_1}{=1} \right], \quad (6.53)$$

$$= \frac{4}{\sigma_{ch}^2} \left[ \sum_{k=1}^a (\hat{\mathbf{w}}_1^T \mathbf{B})_k^2 - \frac{1}{a} \right], \quad (6.54)$$

in which  $\hat{\mathbf{w}}_1^T \mathbf{B} \mathbf{1}_{a \times 1} = 1$  is a consequence of  $\mathbf{B} \mathbf{1}_{a \times 1} = \mathbf{1}_{m_s \times 1}$ .  $\blacksquare$

**Lemma 16.** *If all the VNs of an ETS structure are connected to equal number of missatisfied CNs within the structure (have equal internal degrees), then the variance of initial conditional indicator function is zero, i.e.,  $\mathbb{V}\text{AR}\{\xi_0\} = 0$ .*

*Proof.* The positive dominant left eigenvector of ETSs whose VNs have equal internal degrees is proportional to the all-one vector [45]. Moreover, denoting the internal VN degrees of such a TS with  $\zeta_v$ , the number of state-variables (which is equal to the size of  $\hat{\mathbf{w}}_1^T$ ) is  $a\zeta_v$ . So, based on (6.2), all the elements of  $\hat{\mathbf{w}}_1^T$  are equal to  $1/a\zeta_v$ . On the other hand, for an ETS with regular internal VN degrees, all the columns of  $\mathbf{B}$  have the same number of ones which is equal to  $\zeta_v$ . As a result, the elements of the row vector  $(\hat{\mathbf{w}}_1^T \mathbf{B})$  are all equal to  $1/a$ . Replacing  $(\hat{\mathbf{w}}_1^T \mathbf{B})_k$  by  $1/a$  in (6.49) completes the proof.  $\blacksquare$

**Remark 7.** *We have observed that, in addition to the ETS structures discussed in Lemma 16, the variance of the initial conditional error indicator function of other types of ETS structures also, is relatively small. For example, for the (12,4) ETS of Margulis code and the (8,2) ETS of Tanner code, the variances of initial error indicator functions are  $3.1438 \times 10^{-4}(4/\sigma_{ch}^2)$  and  $1.1 \times 10^{-3}(4/\sigma_{ch}^2)$ , respectively.*

The non-recursive form of Equations (6.35)-(6.38), for  $\ell \geq 3$ , can be obtained by induction as follows:

$$\begin{aligned}
 \mathbf{x}^{(\ell)} &= \sum_{i=0}^{\ell} \mathbf{A}^{\ell-i} \mathbf{B} \mathbf{n}' \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)} + \check{\mathbf{g}}^{(1)} \mathbf{A}^{\ell} \mathbf{B} \mathbf{n}' \odot^2 \prod_{j=2}^{\ell} \hat{\mathbf{g}}^{(j)} \\
 &+ \check{\mathbf{g}}^{(2)} \mathbf{A}^{\ell-1} \mathbf{x}'^{(1)} \odot^2 \prod_{j=3}^{\ell} \hat{\mathbf{g}}^{(j)} + \sum_{i=1}^{\ell} \mathbf{A}^{\ell-i} \mathbf{B}_{ex} \mathbf{L}_{ex}^{(i)} \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)} \\
 &+ \sum_{i=1}^{\ell} \mathbf{A}^{\ell+1-i} \dot{\mathbf{g}}^{(i)} \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)} + \sum_{i=1}^{\ell} \mathbf{A}^{\ell-i} \eta_{0|\theta'} \mathbf{1}_{m_s \times 1} \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)} \\
 &- \sum_{i=1}^{\ell-1} \mathbf{A}^{\ell-i} \eta_{i|\theta'} \mathbf{1}_{m_s \times 1} \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)}. \tag{6.55}
 \end{aligned}$$

Based on (6.55), the non-recursive form of the conditional error indicator function at iteration  $\ell$  is obtained as

$$\begin{aligned}
 \hat{\xi}_{\ell} &\triangleq \hat{\mathbf{w}}_1^T \mathbf{x}^{(\ell)} = \\
 &\sum_{i=0}^{\ell} r^{\ell-i} \hat{\mathbf{w}}_1^T \mathbf{B} \mathbf{n}' \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)} + \check{\mathbf{g}}^{(1)} r^{\ell} \hat{\mathbf{w}}_1^T \mathbf{B} \mathbf{n}' \odot^2 \prod_{j=2}^{\ell} \hat{\mathbf{g}}^{(j)} \\
 &+ \check{\mathbf{g}}^{(2)} r^{\ell-1} \hat{\mathbf{w}}_1^T \mathbf{x}'^{(1)} \odot^2 \prod_{j=3}^{\ell} \hat{\mathbf{g}}^{(j)} + \sum_{i=1}^{\ell} r^{\ell-i} \hat{\xi}_i^{ex} \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)} \\
 &+ \sum_{i=1}^{\ell} r^{\ell+1-i} \dot{\mathbf{g}}^{(i)} \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)} + \sum_{i=1}^{\ell} r^{\ell-i} \eta_{0|\theta'} \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)} \\
 &- \sum_{i=1}^{\ell-1} r^{\ell-i} \eta_{i|\theta'} \prod_{j=i+1}^{\ell} \hat{\mathbf{g}}^{(j)}. \tag{6.56}
 \end{aligned}$$

**Remark 8.** *If the conditional operating point of zero is utilized in the calculation of different gains at all iterations and for all the values of  $\theta'$ , Equation (6.56) will be reduced to Equation (3.24) in Lemma 3.*

### 6.1.3 Modifications to the Proposed Model

In this part, we study the asymptotic behavior of the model proposed in the previous subsection and explain its shortcomings in properly predicting the conditional failure probability of a TS. We then propose modifications to the model to address the

shortcomings.

In general, the size of the internal messages of the TS grow with iterations. On the other hand, for a saturating decoder, the external messages entering the missatisfied CNs (through virtual VNs) also grow with iterations until they saturate at some value  $y_0$ . Under such conditions, all the gains in the conditional semi-linear state-space model tend to zero, since we have

$$\lim_{x \rightarrow -\infty} \phi_x(x, y_0) = 0, \quad (6.57)$$

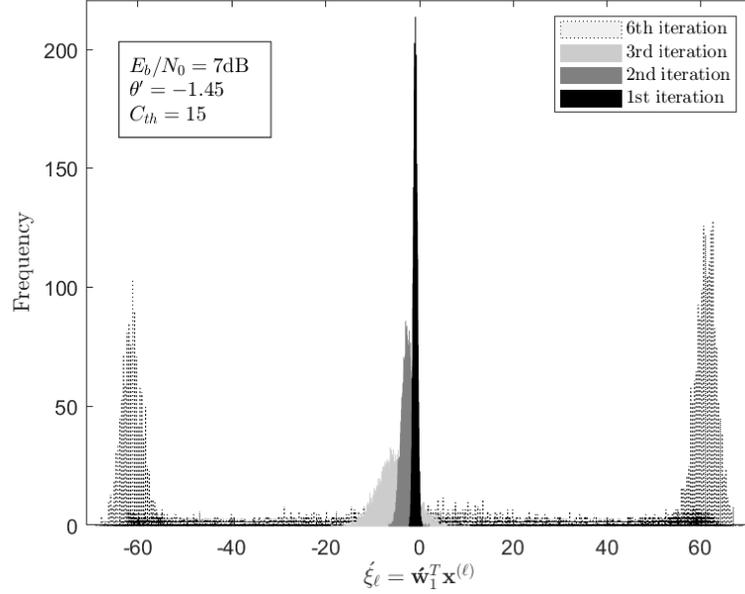
$$\lim_{x \rightarrow -\infty} \phi_{xx}(x, y_0) = 0. \quad (6.58)$$

By replacing the gains of higher iterations with zero, the conditional error indicator function of (6.56) can be approximated by

$$\xi_\ell \approx \overbrace{\mathbf{w}_1^T \mathbf{B} \mathbf{n}'}^{\xi_0} + \eta_{0|\theta'} + \xi_\ell^{ex} + r \dot{g}^{(\ell)}, \quad \text{for } \ell \gg 1, \quad (6.59)$$

in which, we have used (6.39) to replace the addition of the first two terms by  $\xi_0$ . According to Lemma 16 and Remark 7,  $\text{VAR}\{\xi_0\} \approx 0$ . Also, due to clipping, the inputs from unsatisfied CNs at  $\ell \gg 1$  would reach their maximum value, implying that  $\text{VAR}\{\xi_\ell^{ex}\} \approx 0$ . Moreover, based on (6.16) and (6.46) and the fact that the operating point (which is equal for all missatisfied CNs) would surpass the maximum value  $y_0$  of missatisfied CN external inputs (from virtual VNs), the magnitude of  $\dot{g}^{(\ell)}$  would be almost equal to  $y_0$  (because the magnitude of the CN output is approximately equal to the magnitude of the input with minimum absolute value), and thus would have a small variance. It can therefore be concluded that the variance of conditional error indicator function,  $\text{VAR}\{\xi_\ell\}$ , is small. This implies that for a given initial operating point,  $\eta_{0|\theta'}$ , the conditional error indicator function with a probability close to 1 would be equal to a constant value. In other words, as the inputs from unsatisfied and missatisfied CNs would reach their maximum values for  $\ell \gg 1$ , the conditional failure probability, almost deterministically, depends only on the initial operating point. This however, in addition to being counter-intuitive, can be shown to be incorrect by the example given below.

We consider a (640, 192) QC-LDPC code [3] whose dominant TSs are known to be (5, 5) ETSS. The histograms of conditional error indicator function for an instance of



**Figure 6.1:** The histogram of the conditional error indicator function of a (5, 5) ETS of (640, 192) QC-LDPC code [3] at different iterations.

(5, 5) ETS in this code at different iterations are shown in Fig. 6.1. To generate these results, we have simulated about 10000 instances of iterative decoding at  $E_b/N_0 = 7$  dB with the clipping threshold of 15. In the decoding process, the parity check criteria for stopping the decoder were deactivated. Moreover, the value of  $\theta'$ , in (6.5), was set to  $-1.45$ . Fig. 6.1 shows that as iterations go on, first the variance of the indicator function that has a Gaussian shape increases. Then, the distribution splits into two Gaussian-shaped distributions, moving in opposite directions.

Based on the above example, it can be observed that, as iterations go on, the distribution of conditional error indicator function, for a given initial operating point,  $\eta_{0|\theta'}$ , would turn into a bimodal distribution. This however is not consistent with the behavior predicted by (6.59). In the rest of the chapter, we use the notation  $\ell_s$  to denote the first iteration in which the distribution starts to split. Although this definition of  $\ell_s$  is not rigorous, it serves our purpose for the derivations that follow. In the rest of this subsection, we will address the inconsistency between the model and our observations by modifying the proposed model.

We start by a derivation of the conditional probability of an ETS failure that takes into account the bimodal nature of the conditional error indicator function. We

recall that

$$P_e\{\mathcal{S}|\theta'\} = \lim_{\ell \rightarrow \infty} P\{\xi_\ell^{<0}\}, \quad (6.60)$$

where  $P\{\xi_\ell^{<0}\}$  denotes  $P\{\xi_\ell < 0|\theta'\}$ . This probability can be expanded as

$$P\{\xi_\ell^{<0}\} = P\{\xi_\ell^{<0}|\xi_{\ell-1}^{<0}\}P\{\xi_{\ell-1}^{<0}\} + P\{\xi_\ell^{<0}|\xi_{\ell-1}^{\geq 0}\}P\{\xi_{\ell-1}^{\geq 0}\},$$

where  $P\{\xi_\ell^{\geq 0}\}$  denotes  $P\{\xi_\ell \geq 0|\theta'\}$ . For large values of  $\ell$ , we have  $P\{\xi_\ell^{<0}|\xi_{\ell-1}^{\geq 0}\} \approx 0$ , since it is highly unlikely that a positive error indicator function at iteration  $\ell - 1$  changes to a negative value at iteration  $\ell$ . By ignoring the second term in the above equation for  $P\{\xi_\ell^{<0}\}$ , and by applying the same argument to  $P\{\xi_{\ell-1}^{<0}\}$  all the way down to  $P\{\xi_{\ell_s+1}^{<0}\}$ , i.e.,

$$\begin{aligned} P\{\xi_{\ell-1}^{<0}\} &= P\{\xi_{\ell-1}^{<0}|\xi_{\ell-2}^{<0}\}P\{\xi_{\ell-2}^{<0}\} + \overbrace{P\{\xi_{\ell-1}^{<0}|\xi_{\ell-2}^{\geq 0}\}}^{\approx 0}P\{\xi_{\ell-2}^{\geq 0}\} \\ &\vdots \\ P\{\xi_{\ell_s+1}^{<0}\} &= P\{\xi_{\ell_s+1}^{<0}|\xi_{\ell_s}^{<0}\}P\{\xi_{\ell_s}^{<0}\} + \underbrace{P\{\xi_{\ell_s+1}^{<0}|\xi_{\ell_s}^{\geq 0}\}}_{\approx 0}P\{\xi_{\ell_s}^{\geq 0}\}, \end{aligned}$$

we can approximate (6.60) as

$$P_e\{\mathcal{S}|\theta'\} \approx \lim_{\ell \rightarrow \infty} P\{\xi_{\ell_s}^{<0}\} \prod_{i=\ell_s+1}^{\ell} P\{\xi_i^{<0}|\xi_{i-1}^{<0}\}. \quad (6.61)$$

To calculate the probabilities  $P\{\xi_i^{<0}|\xi_{i-1}^{<0}\}$  in (6.61), we assume that  $\xi_i|\xi_{i-1}^{<0}$  has a Gaussian distribution. To derive the new equations, we need the following definition of a truncated distribution.

**Definition 5.** *Suppose that the conditional error indicator function at iteration  $\ell$  is a random variable with probability density function (pdf)  $\chi(x)$ . Then, the truncated distribution  $\chi^-(x)$  is defined as*

$$\chi^-(x) = \frac{1}{P\{\xi_\ell^{<0}\}} \chi(x) u(-x) \quad (6.62)$$

where  $u(\cdot)$  is the unit step function.

For a truncated Gaussian distribution, the mean and variance are given in the

following lemma.

**Lemma 17.** *Suppose that  $\acute{\xi}_\ell$  has a Gaussian distribution with mean and variance  $m$  and  $\sigma^2$ , respectively. Then the mean  $m^-$  and the variance  $\sigma^{2-}$  of the truncated random variable  $\acute{\xi}_\ell^-$  are given by*

$$m^- = m - \sigma \frac{\phi(\varphi)}{\Phi(\varphi)},$$

$$\sigma^{2-} = \sigma^2 \left[ 1 - \varphi \frac{\phi(\varphi)}{\Phi(\varphi)} - \left( \frac{\phi(\varphi)}{\Phi(\varphi)} \right)^2 \right],$$

where  $\varphi = -m/\sigma$ ,  $\phi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$ , and  $\Phi(x) = 1 - Q(x)$ .

In the following, we use the function **Truncate**( $\cdot, \cdot$ ) to relate  $m^-$  and  $\sigma^{2-}$  to  $m$  and  $\sigma^2$ , i.e.,

$$[m^-, \sigma^{2-}] = \mathbf{Truncate}(m, \sigma^2). \quad (6.63)$$

In accordance with (6.61), we also modify the definition of the operating points of the Taylor expansions of missatisfied CN functions for  $\ell > \ell_s$  as follows.

**Definition 6.** *Consider the truncated conditional error indicator function  $\acute{\xi}_\ell^-$  with pdf  $\chi^-(x)$ . Then, for iterations  $\ell > \ell_s$ , the operating points of Taylor expansions of missatisfied CN functions are calculated as  $\eta_{\ell-1|\theta'} = \mathbb{E}\{\acute{\xi}_{\ell-1}^-\}$  (instead of  $\eta_{\ell-1|\theta'} = \mathbb{E}\{\acute{\xi}_{\ell-1}\}$  given in (6.1)).*

We have summarized in Algorithm 3 the steps required to calculate all the probabilities involved in (6.61) for the computation of the conditional failure probability of an ETS. In Algorithm 3, the mean and the variance of the conditional error indicator function, at different iterations, are calculated recursively. At each iteration  $\ell$ , the mean of the conditional error indicator function,  $\eta_{\ell|\theta'}$ , is used as the operating point for calculating the coefficients of the Taylor expansion (gain values) in the next iteration. In addition, the means and variances of the conditional error indicator function  $\acute{\xi}_\ell|\acute{\xi}_{\ell-1}^-$ , for  $\ell \geq \ell_s$ , are utilized to calculate the conditional error probabilities,  $P\{\acute{\xi}_i^{<0}|\acute{\xi}_{i-1}^{<0}\}$ . We note that the equations for the cases of  $3 \leq \ell \leq \ell_s$  and  $\ell > \ell_s$  are essentially the same except that the random variable  $\acute{\xi}_\ell$  in the former is replaced with  $\acute{\xi}_\ell|\acute{\xi}_{\ell-1}^-$  in the latter.

In Algorithm 3, the coefficients of the Taylor expansions of the error indicator functions for different iterations are calculated based on Definition 4 and Equation (6.46).

---

**Algorithm 3** The calculation of  $P\{\xi_{\ell_s}^{<0}\}$  and  $P\{\xi_i^{<0}|\xi_{i-1}^{<0}\}$  for  $i = \ell_s + 1, \dots, \ell$ .

---

- 1:  $\blacktriangleright \ell = 0$
  - 2:  $\eta_{0|\theta'} = \mathbb{E}\{\xi_0\}, \text{VAR}\{\xi_0\} = \frac{4}{\sigma_{ch}^2} \left[ \sum_{k=1}^a (\hat{\mathbf{w}}_1^T \mathbf{B})_k^2 - \frac{1}{a} \right]$ .
  - 3:  $\blacktriangleright \ell = 1$
  - 4:  $\eta_{1|\theta'} = \mathbb{E}\{\xi_1\} = r(\bar{g}^{(1)} + \bar{g}^{(1)}\sigma_{n'}^2) + \eta_{0|\theta'} + \mathbb{E}\{\xi_1^{ex}\}$ ,
  - 5:  $\blacktriangleright \ell = 2$
  - 6:  $\eta_{2|\theta'} = \mathbb{E}\{\xi_2\} = r(\bar{g}^{(2)} + \bar{g}^{(2)}\hat{\mathbf{w}}_1^T \mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\}) + \eta_{0|\theta'} + \mathbb{E}\{\xi_2^{ex}\}$ ,
  - 7:  $\text{VAR}\{\xi_2\} = r^2 \left[ \sigma_{\hat{g}^{(2)}}^2 + (\bar{g}^{(2)})^2 + \sigma_{\hat{g}^{(2)}}^2 \hat{\mathbf{w}}_1^T \Sigma_{\mathbf{x}'(1)} \hat{\mathbf{w}}_1 \right] + r^2 \left[ (\bar{g}^{(2)})^2 + \sigma_{\hat{g}^{(2)}}^2 \right] \hat{\mathbf{w}}_1^T \Sigma_{\mathbf{x}'(1) \odot 2} \hat{\mathbf{w}}_1 + \sigma_{\hat{g}^{(2)}}^2 (\hat{\mathbf{w}}_1^T \mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\})^2 + 2r^2 \hat{\mathbf{w}}_1^T \mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\} \text{COV}\{\dot{g}^{(2)}, \check{g}^{(2)}\} + \text{VAR}\{\xi_0\} + \text{VAR}\{\xi_2^{ex}\}$ ,
  - 8:  $\blacktriangleright \ell \geq 3$
  - 9: **if** ( $3 \leq \ell \leq \ell_s$ ) **then**
  - 10:  $\eta_{\ell|\theta'} = \mathbb{E}\{\xi_\ell\} = r\bar{g}^{(\ell)} + \eta_{0|\theta'} + \mathbb{E}\{\xi_\ell^{ex}\}$ ,
  - 11:  $\text{VAR}\{\xi_\ell\} = r^2(\bar{g}^{(\ell)2} + \sigma_{\hat{g}^{(\ell)}}^2) \text{VAR}\{\xi_{\ell-1}\} + r^2\sigma_{\hat{g}^{(\ell)}}^2 + \text{VAR}\{\xi_0\} + \text{VAR}\{\xi_\ell^{ex}\}$ ,
  - 12: **if** ( $\ell = \ell_s$ ) **then**
  - 13:  $P\{\xi_{\ell_s}^{<0}\} = Q\left(\mathbb{E}\{\xi_{\ell_s}\} / \sqrt{\text{VAR}\{\xi_{\ell_s}\}}\right)$ ,
  - 14:  $[\mathbb{E}\{\xi_{\ell_s}^-}, \text{VAR}\{\xi_{\ell_s}^-}] = \text{Truncate}(\mathbb{E}\{\xi_{\ell_s}\}, \text{VAR}\{\xi_{\ell_s}\})$ ,
  - 15:  $\eta_{\ell_s|\theta'} = \mathbb{E}\{\xi_{\ell_s}^-}$
  - 16: **end if**
  - 17: **else**
  - 18:  $\mathbb{E}\{\xi_\ell|\xi_{\ell-1}^-\} = r\bar{g}^{(\ell)} + \eta_{0|\theta'} + \mathbb{E}\{\xi_\ell^{ex}\}$ ,
  - 19:  $\text{VAR}\{\xi_\ell|\xi_{\ell-1}^-\} = r^2(\bar{g}^{(\ell)2} + \sigma_{\hat{g}^{(\ell)}}^2) \text{VAR}\{\xi_{\ell-1}^-\} + r^2\sigma_{\hat{g}^{(\ell)}}^2 + \text{VAR}\{\xi_0\} + \text{VAR}\{\xi_\ell^{ex}\}$ ,
  - 20:  $P\{\xi_\ell^{<0}|\xi_{\ell-1}^{<0}\} = Q\left(\mathbb{E}\{\xi_\ell|\xi_{\ell-1}^-\} / \sqrt{\text{VAR}\{\xi_\ell|\xi_{\ell-1}^-\}}\right)$ ,
  - 21:  $[\mathbb{E}\{\xi_\ell^-}, \text{VAR}\{\xi_\ell^-}] = \text{Truncate}(\mathbb{E}\{\xi_\ell|\xi_{\ell-1}^-\}, \text{VAR}\{\xi_\ell|\xi_{\ell-1}^-\})$ ,
  - 22:  $\eta_{\ell|\theta'} = \mathbb{E}\{\xi_\ell^-}$ ,
  - 23: **end if**
-

We obtain the distribution and then calculate the mean and the variance of random gains at different iterations using DE.

For  $\ell = 1, 2$ , the model is non-linear and the missatisfied CNs are approximated based on the second order Taylor expansion. For the first iteration ( $\ell = 1$ ), the variance term,  $\sigma_{n'}^2$ , is obtained using (6.8). The derivation of the mean  $\mathbb{E}\{\xi_2'\}$  and the variance  $\text{VAR}\{\xi_2'\}$  of the conditional error indicator function for the second iteration is provided in Appendix 6.3.1. To calculate  $\mathbb{E}\{\xi_2'\}$  and  $\text{VAR}\{\xi_2'\}$ , we need to find the covariance matrix of the state vector  $\mathbf{x}^{(1)}$  and its element-wise squared  $\mathbf{x}^{(1)\odot 2}$ . These covariance matrices whose formulas are derived in Appendix 6.3.2 are denoted by  $\Sigma_{\mathbf{x}^{(1)}}$  and  $\Sigma_{\mathbf{x}^{(1)\odot 2}}$ , respectively. In the equations for  $\mathbb{E}\{\xi_2'\}$  and  $\text{VAR}\{\xi_2'\}$ , the operator  $\mathbb{D}\{\cdot\}$  places the diagonal elements of a matrix into a column vector.

It is noted that for iterations larger than 2, due to the linearity of the model, the variance of the error indicator function at each iteration can be calculated based on the variance of the error indicator function in the previous iteration, and there is no need for the calculation of the covariance matrix of the state vector in the previous iteration.

The derivation of the mean and the variance of the conditional error indicator function for  $\ell \geq 3$  is provided in Appendix 6.3.3.

#### 6.1.4 Relationship between Our Proposed Method and the Semi-Analytical Method of [34]

In a pioneering work [34], Richardson proposed a semi-analytical code-dependent technique based on importance sampling for the error floor estimation of LDPC codes. In this method, the noise is biased in the direction of the TS by a certain (varying) amount,  $\theta'$ . Then, by using an orthogonal transform on the i.i.d Gaussian noise samples within the TS, the noise vector is represented with a new set of basis vectors, where one vector is in the direction of the TS. This allows for the bias value  $\theta'$  to be treated as a Gaussian random variable. Fixing  $\theta'$ , Monte Carlo (MC) simulation is then performed with respect to the other dimensions of the noise vector. The final estimate of the error floor is obtained by averaging the conditional failure rates over the Gaussian bias values,  $\theta'$ .

We note that the method of [34] for calculating the total failure probability by averaging over the bias value is equivalent to using Equation (6.11). The major difference, however, is that unlike our analytical approach in calculating  $P_e\{\mathcal{S}|\theta'\}$ ,

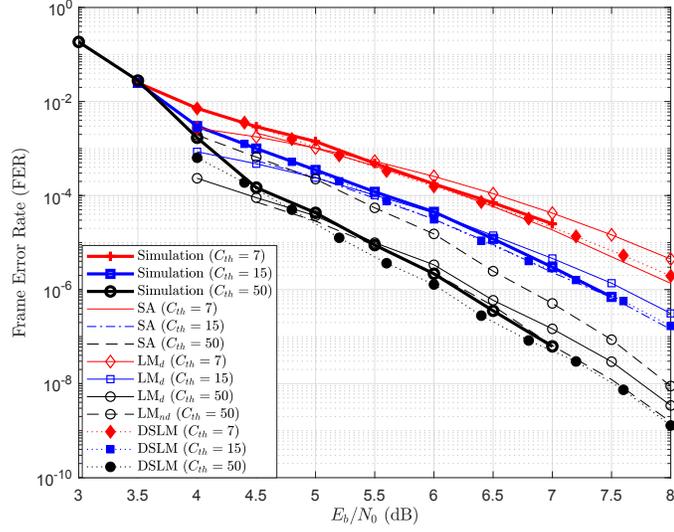
the method of [34] estimates  $P_e\{\mathcal{S}|\theta'\}$  for different values of  $\theta'$  using MC simulations.

In [34], it has been shown by some examples that the conditional failure probabilities  $P_e\{\mathcal{S}|\theta'\}$  change only slightly with SNR, and thus the error floor can be estimated by extrapolation. In other words, in [34], the failure probability  $P_e\{\mathcal{S}|\theta'\}$ , in (6.11), is estimated only at a certain SNR, and then the same estimate is used for other SNR values in the error floor region. In the simulation results of Section 6.2, we will demonstrate that this extrapolation can result in rather significant estimation error in saturating decoders with rather large clipping thresholds.

The error floor in our proposed model can also be extrapolated in the same manner as in [34], i.e., for different values of  $\theta'$ , we calculate  $P_e\{\mathcal{S}|\theta'\}$  for a specific SNR and then use it in (6.11) for all the other SNR values of interest. We show that, unlike the case for the method of [34], extrapolation in our model accurately estimates the failure probabilities for different clipping thresholds. Moreover, the application of extrapolation significantly reduces the complexity of our approach for the error floor estimation since DE calculations and the computations involved in finding the different parameters of the model need to be performed only once (for one SNR). Although, the linear state-space model of [45] does not use (6.11), and thus does not require the calculation of  $P_e\{\mathcal{S}|\theta'\}$ , the methodology of extrapolation can also be applied to the technique of [45] for the error floor estimation. For this, we use the DE results of a certain SNR to obtain the failure probabilities for all the other SNR values. Our simulation results, however, show that such extrapolations result in severe errors in the estimation of error floor.

**Remark 9.** *In order to estimate  $P_e\{\mathcal{S}|\theta'\}$ , in [34], the regular MC method is used to generate random inputs of the decoder for all VNs except for the  $a$  VNs of the TS. With respect to the VNs of the TS, Equation (6.5) is deployed. To this end, while the first dimension of the noise vector within the TS is set to a normalized all-one vector, the other  $a - 1$  orthogonal vectors are obtained using Gram-Schmidt process [97, p. 656]. Then, by fixing the first dimension (fixed  $\theta'$ ), the coefficients of the other orthonormal vectors (i.e.,  $\theta_2, \dots, \theta_a$ ) are generated randomly according to  $\mathcal{N}(0, \sigma_{ch}^2)$ .*

*In this chapter, we use a simpler approach for the implementation of the method of [34] that does not require Gram-Schmidt process. In this method, the noise vector of the TS, like the rest of the VNs of the code, is generated randomly according to  $\mathcal{N}(0, \sigma_{ch}^2)$ . Then, the projection of the TS noise vector on a normalized all-one vector*



**Figure 6.2:** Simulation and estimation results for  $\mathcal{C}_1$  decoded by SPA over the AWGN channel ( $I_{max} = 200$ ).

(first dimension) is calculated and is removed from the noise vector. Finally, the fixed bias value on the first dimension is added to the remaining part of the noise. Therefore, by using the same notations of (6.3), the TS input vector is generated as

$$L = \frac{2}{\sigma_{ch}^2} (\mathbf{1}_{a \times 1} + \mathbf{n}_a - (\boldsymbol{\gamma}_1^T \mathbf{n}_a) \boldsymbol{\gamma}_1 + \sqrt{a} \theta' \boldsymbol{\gamma}_1). \quad (6.64)$$

## 6.2 Simulation Results

To examine the accuracy of our proposed method, we have applied it to different LDPC codes, decoded by SPA over the AWGN channel. We report our results for the following LDPC codes: (640, 192) QC code ( $\mathcal{C}_1$ ) [3], Margulis (2640, 1320) code ( $\mathcal{C}_2$ ), and Tanner (155, 64) code ( $\mathcal{C}_3$ ). Regarding the MC simulations, we collect 100 frame errors at each SNR point reported in the figures. In the following, we use the notation DSLM for the proposed Semi-Linear Model with Dynamical operating point. In the proposed model, we use  $\ell_s = 4$  for all the reported results. Notations  $\text{LM}_d$  and  $\text{LM}_{nd}$  are used for the linear model of [45] with and without damping factor, respectively, and SA is used for the semi-analytical method of Richardson [34]. Moreover, we use notations  $\text{DSLME}$ ,  $\text{LM}_{dE}$  and  $\text{SA}_E$  for the cases where extrapolation is used in the

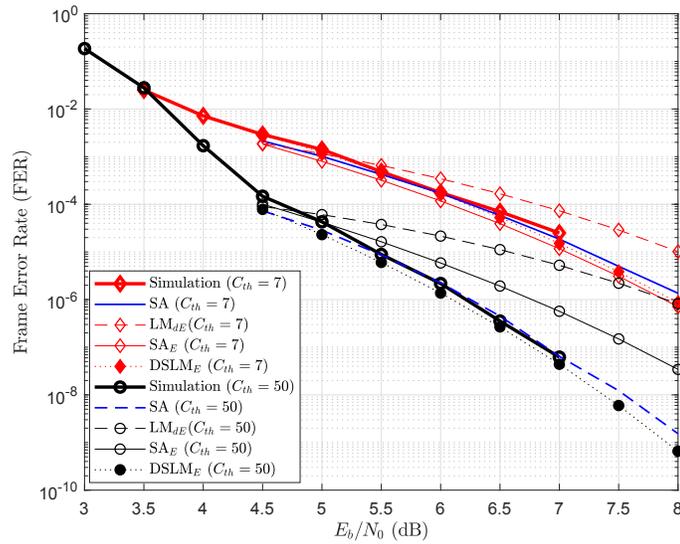
estimations.

Fig. 6.2 shows the MC simulation results along with  $\text{LM}_d$ , DSLM and SA estimation results for  $\mathcal{C}_1$  for different clipping thresholds. This code has VN degree 5 and an irregular CN degree distribution. The irregularity of the degree distribution is taken into account in our model, similar to the model of [45], by applying a weighted average based on the degree distribution of the code within the DE framework. It is known that the most harmful TS of this code is a (5, 5) ETS with multiplicity 64. We thus apply our method to this ETS structure and then multiply the result by 64. The results demonstrate a good match between DSLM and SA estimations as well as the MC simulation results. The  $\text{LM}_d$  estimation results, however, are diverging from the simulation curves, in particular, for higher saturation levels. We have also reported the result of  $\text{LM}_{nd}$  for  $C_{th} = 50$ . This represents the true performance of the linear state-space model without the addition of the heuristic damping factor. As can be seen, while our semi-linear model accurately estimates the error floor in this scenario, the results of  $\text{LM}_{nd}$  over-estimate the error floor significantly. The results of Fig. 6.2 also reveal that by increasing the SNR, since  $(1 - P_\ell) \rightarrow 1$ , the performance of  $\text{LM}_d$  converges to that of  $\text{LM}_{nd}$ .

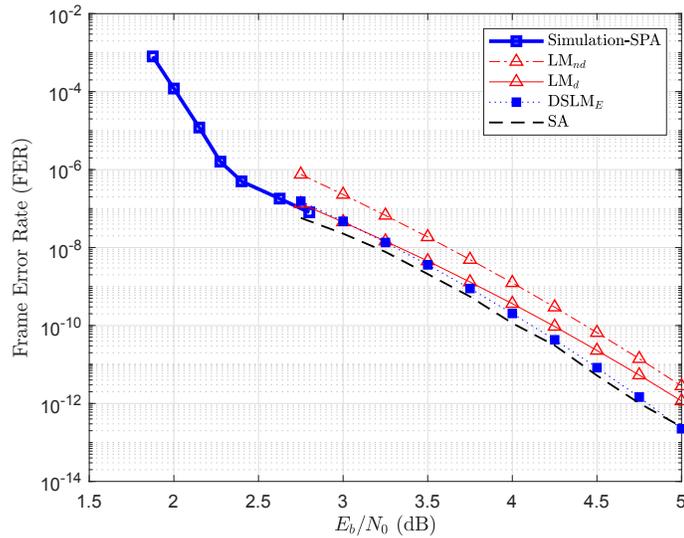
For  $\mathcal{C}_1$ , in Fig. 6.3, we have reported the estimation results with extrapolation. We have also added the result of SA (without extrapolation) to the figure for comparison. As can be seen,  $\text{LM}_{dE}$  has the worst performance for different saturation levels. Also, while  $\text{SA}_E$  and  $\text{DSLM}_E$  are well-matched with MC simulations at the lower saturation level of  $C_{th} = 7$ , for  $C_{th} = 50$ , only  $\text{DSLM}_E$  curve matches well with the MC simulation curve. The  $\text{SA}_E$  method over-estimates the error floor and even results in the wrong slope for the FER.

For the rest of simulations, we compare the performance of  $\text{DSLM}_E$  with the performance of other methods without extrapolation as their performance with extrapolation would degrade.

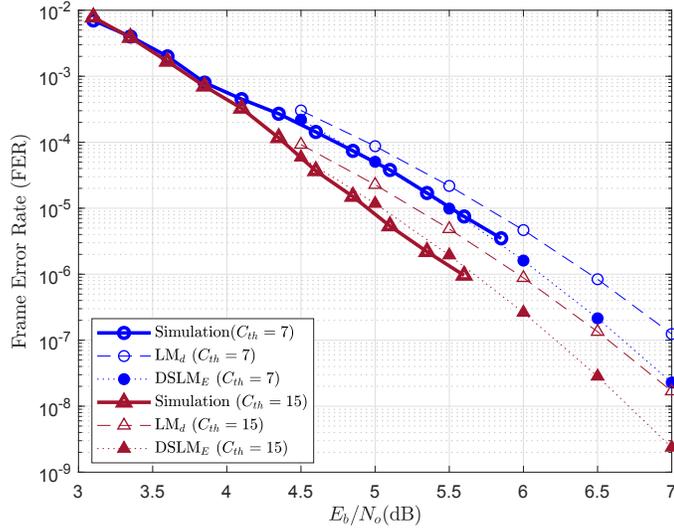
The results for  $\mathcal{C}_2$  are presented in Fig. 6.4, where the clipping threshold is  $C_{th} = 25$ . The estimation results of  $\text{DSLM}_E$ ,  $\text{LM}_d$  and  $\text{LM}_{nd}$  are obtained based on the dominant TS structure of this code, i.e., the (12, 4) ETS with multiplicity 1320 [34, 86]. The trends for this code are similar to those of  $\mathcal{C}_2$ . The estimation based on our model with extrapolation is well matched with MC simulations and SA. By increasing SNR, the estimation results of  $\text{LM}_d$  converge to those of  $\text{LM}_{nd}$ , both



**Figure 6.3:** Simulation and extrapolated estimation results for  $\mathcal{C}_1$  decoded by SPA over the AWGN channel ( $I_{max} = 200$ ). The extrapolated curves are obtained based on the results at  $E_b/N_0 = 4.5$  dB



**Figure 6.4:** Simulation and estimation results for  $\mathcal{C}_2$  decoded by SPA over the AWGN channel ( $I_{max} = 200$ ,  $C_{th} = 25$ ). The extrapolation in  $DSLME$  is based on the results at  $E_b/N_0 = 2.75$  dB.



**Figure 6.5:** Simulation and estimation results for  $\mathcal{C}_3$  decoded by SPA over the AWGN channel ( $I_{max} = 200$ ). The extrapolation in  $DSLME$  is based on the results at  $E_b/N_0 = 4.5$  dB.

considerably away from the estimation results of SA.

The results for  $\mathcal{C}_3$  are presented in Fig. 6.5. The  $(8, 2)$  ETS with multiplicity 465 is known to be the dominant error prone structure of this code [44], and is thus used to estimate the error floor based on  $LM_d$  and  $DSLME$ . As can be observed, for both saturation levels of 7 and 15,  $LM_d$  overestimates the error floor. However, the estimation results of  $DSLME$  in both cases match the MC simulation results well.

the operating point of the expansion in the proposed model

## 6.3 Appendix

### 6.3.1 Derivation of the mean and the variance of $\xi_2$ , given in Algorithm 3

In the following, we use operator  $\mathbf{diag}\{\cdot\}$  to turn a vector into a diagonal matrix whose diagonal elements are the elements of the vector in the same order. Operator  $\mathbb{D}\{\cdot\}$  performs the opposite operation, and transforms the diagonal elements of a matrix into a column vector. Notation  $\mathbb{COV}\{\cdot\}$  is used to denote the covariance of a random variable or the covariance matrix of a random vector.

From (6.41), the conditional error indicator function in the second iteration is given by

$$\acute{\xi}_2 = r(\dot{\mathbf{g}}^{(2)} + \hat{\mathbf{g}}^{(2)} \acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)} + \check{\mathbf{g}}^{(2)} \acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)\odot 2}) + \acute{\xi}_0 + \acute{\xi}_2^{ex}. \quad (6.65)$$

We have  $\mathbf{x}'^{(1)} = (\mathbf{x}^{(1)} - \eta_{1|\theta'} \mathbf{1}_{m_s \times 1}) \approx (\mathbf{x}^{(1)} - \mathbb{E}\{\mathbf{x}^{(1)}\})$ , and thus  $\mathbb{E}\{\mathbf{x}'^{(1)}\} \approx \mathbf{0}$ . By taking the expected value of both sides of (6.65), we then have

$$\begin{aligned} \mathbb{E}\{\acute{\xi}_2\} &= r(\bar{\mathbf{g}}^{(2)} + \bar{\mathbf{g}}^{(2)} \mathbb{E}\{\acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)}\} + \bar{\mathbf{g}}^{(2)} \acute{\mathbf{w}}_1^T \mathbb{D}\{\boldsymbol{\Sigma}_{\mathbf{x}'^{(1)}}\}) \\ &\quad + \eta_{0|\theta'} + \mathbb{E}\{\acute{\xi}_2^{ex}\}. \end{aligned} \quad (6.66)$$

Moreover, from (6.26), we have

$$\acute{\mathbf{w}}_1^T \mathbf{x}'^{(\ell-1)} = \acute{\mathbf{w}}_1^T (\mathbf{x}^{(\ell-1)} - \eta_{\ell-1} \mathbf{1}_{m_s \times 1}) = \acute{\xi}_{\ell-1} - \eta_{\ell-1}. \quad (6.67)$$

Taking the expected value of both sides of (6.67) for  $\ell = 2$ , we obtain  $\mathbb{E}\{\acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)}\} = 0$ . Replacing this in (6.66) gives the equation for  $\mathbb{E}\{\acute{\xi}_2\}$  in Algorithm 3.

The variance of (6.65) is calculated as

$$\begin{aligned} \text{VAR}\{\acute{\xi}_2\} &\triangleq \mathbb{E}\{(\acute{\xi}_2 - \mathbb{E}\{\acute{\xi}_2\})^2\} \\ &= r^2 \left[ \sigma_{\dot{\mathbf{g}}^{(2)}}^2 + (\bar{\mathbf{g}}^{(2)})^2 + \sigma_{\check{\mathbf{g}}^{(2)}}^2 \acute{\mathbf{w}}_1^T \boldsymbol{\Sigma}_{\mathbf{x}'^{(1)}} \acute{\mathbf{w}}_1 \right] + \\ &\quad r^2 \left[ (\bar{\mathbf{g}}^{(2)})^2 + \sigma_{\check{\mathbf{g}}^{(2)}}^2 \acute{\mathbf{w}}_1^T \boldsymbol{\Sigma}_{\mathbf{x}'^{(1)\odot 2}} \acute{\mathbf{w}}_1 + \sigma_{\check{\mathbf{g}}^{(2)}}^2 \underbrace{\mathbb{E}^2\{\acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)\odot 2}\}}_{(\acute{\mathbf{w}}_1^T \mathbb{D}\{\boldsymbol{\Sigma}_{\mathbf{x}'^{(1)}}\})^2} \right] \\ &\quad + 2r^2 \underbrace{\text{COV}\{\dot{\mathbf{g}}^{(2)}, \check{\mathbf{g}}^{(2)} \acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)\odot 2}\}}_{\acute{\mathbf{w}}_1^T \mathbb{D}\{\boldsymbol{\Sigma}_{\mathbf{x}'^{(1)}}\} \text{COV}\{\dot{\mathbf{g}}^{(2)}, \check{\mathbf{g}}^{(2)}\} (*)} + \text{VAR}\{\acute{\xi}_0\} + \text{VAR}\{\acute{\xi}_2^{ex}\} \\ &\quad + 2r^2 \underbrace{\text{COV}\{\dot{\mathbf{g}}^{(2)}, \hat{\mathbf{g}}^{(2)} \acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)}\}}_{\text{COV}\{\dot{\mathbf{g}}^{(2)}, \hat{\mathbf{g}}^{(2)}\} \mathbb{E}\{\acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)}\} (**)} \\ &\quad + 2r^2 \underbrace{\text{COV}\{\hat{\mathbf{g}}^{(2)} \acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)}, \check{\mathbf{g}}^{(2)} \acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)\odot 2}\}}_{\text{COV}\{\hat{\mathbf{g}}^{(2)}, \check{\mathbf{g}}^{(2)}\} \mathbb{E}\{\acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)} \mathbf{x}'^{(1)\odot 2 T} \acute{\mathbf{w}}_1\}}. \end{aligned} \quad (6.68)$$

To derive (6.68), we have removed the terms that contain  $\text{COV}\{\acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)}, \acute{\xi}_0\}$  or  $\text{COV}\{\acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)\odot 2}, \acute{\xi}_0\}$ . The reason is that based on Lemma 16 and Remark 7,  $\text{VAR}\{\acute{\xi}_0\}$  typically has a small value. This implies that  $\text{COV}\{\acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)}, \acute{\xi}_0\} \approx 0$  and  $\text{COV}\{\acute{\mathbf{w}}_1^T \mathbf{x}'^{(1)\odot 2}, \acute{\xi}_0\} \approx 0$ . In (\*) and (\*\*), we have used the fact that the state

variables and missatisfied CN external inputs and thus the state variables and gain values are independent. We note that the term with  $(**)$  is equal to zero since  $\mathbb{E}\{\hat{\mathbf{w}}_1^T \mathbf{x}^{(1)}\} = 0$ . The central  $k$ -order moments of a Gaussian random variable are obtained in [98, Theorem 3.1]. Using this Theorem, and since  $\mathbb{E}\{\mathbf{x}^{(1)}\} = \mathbf{0}$ , we have  $\mathbb{E}\{\hat{\mathbf{w}}_1^T \mathbf{x}^{(1)} \mathbf{x}^{(1)\odot 2T} \hat{\mathbf{w}}_1\} = 0$ . This means that the last term in (6.68) is also equal to zero. This completes the derivation of  $\text{VAR}\{\xi_2^{\check{}}\}$  as given in Algorithm 3.

### 6.3.2 Calculation of the covariance matrices $\Sigma_{\mathbf{x}^{(1)}}$ and $\Sigma_{\mathbf{x}^{(1)\odot 2}}$

From (6.36),

$$\mathbf{x}^{(1)} = \mathbf{A}(\dot{\mathbf{g}}^{(1)} + \hat{\mathbf{g}}^{(1)} \mathbf{Bn}' + \check{\mathbf{g}}^{(1)} \mathbf{Bn}'^{\odot 2}) + \mathbf{x}^{(0)} + \mathbf{B}_{ex} \mathbf{L}_{ex}^{(1)}. \quad (6.69)$$

By using (6.69) and  $\mathbb{E}\{\mathbf{x}^{(1)}\} \approx 0$ , we have

$$\begin{aligned} \Sigma_{\mathbf{x}^{(1)}} &\approx \mathbb{E}\{\mathbf{x}^{(1)} \mathbf{x}^{(1)T}\} \approx \mathbb{E}\{(\mathbf{x}^{(1)} - \mathbb{E}\{\mathbf{x}^{(1)}\})(\mathbf{x}^{(1)} - \mathbb{E}\{\mathbf{x}^{(1)}\})^T\} \\ &= \Sigma_{\mathbf{x}^{(1)}} = \underbrace{\mathbf{A} \Sigma_{\dot{\mathbf{g}}^{(1)}} \mathbf{A}^T}_{\sigma_{\dot{\mathbf{g}}^{(1)}}^2 \mathbf{A} \mathbf{B}_s \mathbf{B}_s^T \mathbf{A}^T} + \underbrace{\Sigma_{\mathbf{x}^{(0)}}}_{\mathbf{B} \Sigma_{\mathbf{n}'} \mathbf{B}^T} + \\ &(\bar{\mathbf{g}}^{(1)2} + \sigma_{\hat{\mathbf{g}}^{(1)}}^2) \mathbf{A} \mathbf{B} \Sigma_{\mathbf{n}'} \mathbf{B}^T \mathbf{A}^T + \underbrace{\text{COV}\{\hat{\mathbf{g}}^{(1)} \mathbf{A} \mathbf{Bn}', \mathbf{Bn}'\}}_{\bar{\mathbf{g}}^{(1)} \mathbf{A} \mathbf{B} \Sigma_{\mathbf{n}'} \mathbf{B}^T = \spadesuit} \\ &+ \underbrace{\text{COV}\{\hat{\mathbf{g}}^{(1)} \mathbf{A} \mathbf{Bn}', \mathbf{Bn}'\}^T}_{\spadesuit^T} + (\bar{\mathbf{g}}^{(1)2} + \sigma_{\check{\mathbf{g}}^{(1)}}^2) \mathbf{A} \mathbf{B} \Sigma_{\mathbf{n}'^{\odot 2}} \mathbf{B}^T \mathbf{A}^T \\ &+ \sigma_{\check{\mathbf{g}}^{(1)}}^2 \mathbf{A} \mathbf{B} \underbrace{\mathbb{E}\{\mathbf{n}'^{\odot 2}\}}_{\sigma_{\mathbf{n}'}^2 \mathbf{1}_{a \times 1}} + \underbrace{\text{COV}\{\check{\mathbf{g}}^{(1)} \mathbf{A} \mathbf{Bn}'^{\odot 2}, \mathbf{A} \dot{\mathbf{g}}^{(1)}\}}_{\sigma_{\mathbf{n}'}^2 \mathbf{A} \mathbf{B} \mathbf{1}_{a \times 1} \text{COV}\{\check{\mathbf{g}}^{(1)}, \dot{\mathbf{g}}^{(1)T}\} \mathbf{A}^T = \clubsuit} \\ &+ \underbrace{\text{COV}\{\check{\mathbf{g}}^{(1)} \mathbf{A} \mathbf{Bn}'^{\odot 2}, \mathbf{A} \dot{\mathbf{g}}^{(1)}\}^T}_{\clubsuit^T} + \sigma_{ex}^{2(1)} \mathbf{B}_{ex} \mathbf{I}_b \mathbf{B}_{ex}^T. \end{aligned} \quad (6.70)$$

where  $\Sigma_{\mathbf{n}'}$  is given in (6.7). Also,  $\text{COV}\{\check{\mathbf{g}}^{(1)}, \dot{\mathbf{g}}^{(1)T}\}$  is the covariance between the scalar  $\check{\mathbf{g}}^{(1)}$  and each element of the vector  $\dot{\mathbf{g}}^{(1)T}$ , which results in a  $1 \times m_s$  vector. To

obtain  $\Sigma_{\mathbf{n}'\odot 2}$  in (6.70), we use Theorem 3.1 of [98] and get

$$\Sigma_{\mathbf{n}'\odot 2} = \begin{bmatrix} E[n_1'^4] & E[n_1'^2 n_2'^2] & \cdots & E[n_1'^2 n_a'^2] \\ E[n_2'^2 n_1'^2] & E[n_2'^4] & \ddots & \vdots \\ \vdots & \ddots & \ddots & E[n_{a-1}'^2 n_a'^2] \\ E[n_a'^2 n_1'^2] & \cdots & E[n_a'^2 n_{a-1}'^2] & E[n_a'^4] \end{bmatrix} - \sigma_{n'}^4 \mathbf{1}_{a \times a},$$

where  $a$  is the number of VNs in the TS, and

$$\begin{aligned} E[n_k'^4] &= 3\sigma_{n'}^4 = 3\left(\frac{4}{\sigma_{ch}^2} \frac{a-1}{a}\right)^2, \quad \text{for } k = 1, \dots, a, \\ E[n_i'^2 n_j'^2] &= \sigma_{n'}^4 + 2\sigma_{n'_i n'_j}^2, \quad \text{for } i, j = 1, \dots, a, i \neq j. \\ &= \left(\frac{4}{\sigma_{ch}^2} \frac{a-1}{a}\right)^2 + 2\left(\frac{4}{\sigma_{ch}^2} \frac{-1}{a}\right)^2. \end{aligned}$$

Therefore,

$$\Sigma_{\mathbf{n}'\odot 2} = 2\left(\frac{4}{\sigma_{ch}^2}\right)^2 \left( \left(1 - \frac{2}{a}\right) \mathbf{I}_a + \frac{1}{a^2} \mathbf{1}_{a \times a} \right). \quad (6.71)$$

This completes the derivation of  $\Sigma_{\mathbf{x}'(1)}$ .

For  $\Sigma_{\mathbf{x}'(1)\odot 2}$ , we have

$$\begin{aligned} \Sigma_{\mathbf{x}'(1)\odot 2} &= \begin{bmatrix} E[x_1'^4] & E[x_1'^2 x_2'^2] & \cdots & E[x_1'^2 x_{m_s}'^2] \\ E[x_2'^2 x_1'^2] & E[x_2'^4] & \ddots & \vdots \\ \vdots & \ddots & \ddots & E[x_{m_s-1}'^2 x_{m_s}'^2] \\ E[x_{m_s}'^2 x_1'^2] & \cdots & E[x_{m_s}'^2 x_{m_s-1}'^2] & E[x_{m_s}'^4] \end{bmatrix} \\ &\quad - \mathbb{E}\{\mathbf{x}'(1)\odot 2\} \mathbb{E}\{\mathbf{x}'(1)\odot 2\}^T. \end{aligned} \quad (6.72)$$

Assuming  $\mathbb{E}\{\mathbf{x}'(1)\} = 0$ , the terms in (6.72) are equal to [98, Theorem 3.1]:

$$\begin{aligned} E[x_k'^4] &= 3\sigma_{x'_k}^4, \quad \text{for } k = 1, \dots, m_s, \\ E[x_i'^2 x_j'^2] &= \sigma_{x'_i}^2 \sigma_{x'_j}^2 + 2\sigma_{x'_i x'_j}^2, \quad \text{for } i, j = 1, \dots, m_s, i \neq j, \\ \mathbb{E}\{\mathbf{x}'(1)\odot 2\} &= \mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\}, \end{aligned}$$

where  $\sigma_{x'_i x'_j}$  is used to denote  $\text{COV}\{x'_i, x'_j\}$ . Thus, Equation (6.72) can be rewritten

as

$$\begin{aligned}
 \Sigma_{\mathbf{x}'(1)}^{\odot 2} = & \begin{bmatrix} \sigma_{x'_1}^2 \sigma_{x'_1}^2 & \sigma_{x'_1}^2 \sigma_{x'_2}^2 & \cdots & \sigma_{x'_1}^2 \sigma_{x'_{m_s}}^2 \\ \sigma_{x'_2}^2 \sigma_{x'_1}^2 & \sigma_{x'_2}^2 \sigma_{x'_2}^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \sigma_{x'_{m_s-1}}^2 \sigma_{x'_{m_s}}^2 \\ \sigma_{x'_{m_s}}^2 \sigma_{x'_1}^2 & \cdots & \sigma_{x'_{m_s}}^2 \sigma_{x'_{m_s-1}}^2 & \sigma_{x'_{m_s}}^2 \sigma_{x'_{m_s}}^2 \end{bmatrix} + \\
 2 & \begin{bmatrix} \sigma_{x'_1}^2 \sigma_{x'_1}^2 & \sigma_{x'_1}^2 \sigma_{x'_2}^2 & \cdots & \sigma_{x'_1}^2 \sigma_{x'_{m_s}}^2 \\ \sigma_{x'_2}^2 \sigma_{x'_1}^2 & \sigma_{x'_2}^2 \sigma_{x'_2}^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \sigma_{x'_{m_s-1}}^2 \sigma_{x'_{m_s}}^2 \\ \sigma_{x'_{m_s}}^2 \sigma_{x'_1}^2 & \cdots & \sigma_{x'_{m_s}}^2 \sigma_{x'_{m_s-1}}^2 & \sigma_{x'_{m_s}}^2 \sigma_{x'_{m_s}}^2 \end{bmatrix} - \mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\} \mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\}^T, \quad (6.73)
 \end{aligned}$$

and simplified as

$$\Sigma_{\mathbf{x}'(1)}^{\odot 2} = \mathbf{diag}\{\mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\}\} \mathbf{1}_{m_s \times 1} \mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\}^T + 2\Sigma_{\mathbf{x}'(1)}^{\odot 2} - \mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\} \mathbb{D}\{\Sigma_{\mathbf{x}'(1)}\}^T, \quad (6.74)$$

where  $\Sigma_{\mathbf{x}'(1)}^{\odot 2}$  denotes the element-wise second power of matrix  $\Sigma_{\mathbf{x}'(1)}$ .

### 6.3.3 Derivation of the mean and the variance of the conditional error indicator function for $\ell \geq 3$ , as given in Algorithm 3

From Equation (6.42), for  $\ell \geq 3$ , we have

$$\hat{\xi}_\ell = r(\dot{g}^{(\ell)} + \hat{g}^{(\ell)} \hat{\mathbf{w}}_1^T \mathbf{x}'^{(\ell-1)}) + \hat{\xi}_0 + \hat{\xi}_\ell^{ex}. \quad (6.75)$$

Taking the expected value of both sides, considering the independence of the gain  $\hat{g}^{(\ell)}$  and state variables, and the fact that  $\mathbb{E}\{\hat{\mathbf{w}}_1^T \mathbf{x}'^{(\ell-1)}\} = 0$ , we then obtain

$$\mathbb{E}\{\hat{\xi}_\ell\} = r\bar{g}^{(\ell)} + \eta_{0|\theta'} + \mathbb{E}\{\hat{\xi}_\ell^{ex}\}. \quad (6.76)$$

For the variance, since the covariance between  $\xi_\ell^{ex}$  and the other three terms on the right hand side of (6.75) is zero (due to independence), we have

$$\begin{aligned} \text{VAR}\{\xi_\ell\} &= r^2\sigma_{\hat{g}^{(\ell)}}^2 + r^2\text{VAR}\{\hat{g}^{(\ell)}\hat{\mathbf{w}}_1^T\mathbf{x}'^{(\ell-1)}\} + \text{VAR}\{\xi_0\} + \text{VAR}\{\xi_\ell^{ex}\} + \\ &\quad r^2\text{COV}\{\dot{g}^{(\ell)}, \hat{g}^{(\ell)}\hat{\mathbf{w}}_1^T\mathbf{x}'^{(\ell-1)}\} + r\text{COV}\{\dot{g}^{(\ell)}, \xi_0\} + \\ &\quad r\text{COV}\{\hat{g}^{(\ell)}\hat{\mathbf{w}}_1^T\mathbf{x}'^{(\ell-1)}, \xi_0\}. \end{aligned} \quad (6.77)$$

Due to the small variance of  $\xi_0$ , the last two terms in (6.77) are approximately equal to zero. Also, due to the independence of gains and state variables, and since  $\mathbb{E}\{\hat{\mathbf{w}}_1^T\mathbf{x}'^{(\ell-1)}\} = 0$ , the fifth term on the right hand side of (6.77) is zero. Moreover, the second term in (6.77) can be written as

$$r^2\text{VAR}\{\hat{g}^{(\ell)}\hat{\mathbf{w}}_1^T\mathbf{x}'^{(\ell-1)}\} = r^2(\bar{\hat{g}}^{(\ell)2} + \sigma_{\hat{g}^{(\ell)}}^2)\text{VAR}\{\xi_{\ell-1}\}. \quad (6.78)$$

This is because  $\mathbb{E}\{\hat{\mathbf{w}}_1^T\mathbf{x}'^{(\ell-1)}\} = 0$ , and that based on (6.67),  $\text{VAR}\{\hat{\mathbf{w}}_1^T\mathbf{x}'^{(\ell-1)}\} = \text{VAR}\{\xi_{\ell-1}\}$ .

## Chapter 7

# Conclusion and Future Work

### 7.1 Conclusion

In this thesis, we studied the harmfulness of problematic graphical structures of LDPC codes, collectively referred to as trapping sets, in the error floor region. Our proposed methods fit in the code-independent category of techniques in estimation and analysis of TSs.

The linear state-space model is a well-known code-independent approach to evaluate the harmfulness of ETS structures under SPA over the AWGN channel. In Chapter 3, we investigated this model carefully and pointed out its limitations in accurately estimating the error floor. For this, we extended the model to include random linear gains at missatisfied check nodes. The extended model was also used to study the effect of dependencies among extrinsic messages entering the TS through missatisfied check nodes. We proved that such dependencies will increase the error probability of the TS. We then proposed an alternate code-independent approach to estimate the error probability of different ETS structures. The proposed technique applies a local importance sampling (LIS) to the message-passing algorithm over the subgraph of the TS of interest. To model the extrinsic messages, the proposed method uses density evolution (DE) with a simple correlation model, hence the name “LIS-DE<sub>c</sub>.” We applied the proposed LIS-DE<sub>c</sub> method to a number of random and structured LDPC codes decoded by a variety of iterative algorithms, quantized and otherwise, over both AWGN and BSCs, and demonstrated that the error floor can be accurately estimated in all cases.

Motivated by the wide spread application of layered decoders and the fact that the behavior of such decoders in the error floor region is still not well understood, in

Chapter 4, we studied the error floor of QC-LDPC codes under row layered saturating SPA. For this, we developed a linear state-space model for LETSs of the code which incorporates the layered nature of scheduling. We then studied the system matrices of the model and made connections between these matrices and those corresponding to the linear state-space model of the flooding decoder. In particular, we demonstrated that the spectral radius of the transition matrix of the layered decoder is always larger than that of its flooding counterpart. We showed that the proposed model can estimate the failure probability of LETSs, as well as the error floor of the code rather accurately. In particular, we demonstrated that the failure rate of a LETS under layered decoding is not only a function of its topology, but also depends on the location of its constituent CNs in different layers. We called this information, TS layer profile, or TSLP, in brief. As a result, we established that the error floor of the same code under the same saturating SPA can significantly change by changing the order in which the row layers are updated. We also studied the problem of finding the schedule with the lowest error floor and devised an efficient algorithm to find it. In particular, we demonstrated that the layered decoder, with a well designed schedule, can outperform its flooding counterpart. This adds yet another advantage to the application of layered decoding in practice. The well-known advantages, prior to this result, were the faster convergence and lower hardware complexity.

We extended the linear state-space model of row layered decoders to the column layered decoders in Chapter 5 and we demonstrated that the model parameters for the latter are derived differently than those of the former. We then used the model to optimize the error floor performance of column layered decoder. Finally, we compared our theoretical estimates of the error floor with Monte Carlo simulations for QC-LDPC codes, both variable-regular and irregular, and demonstrated a good match between the two. The simulation results imply that the error floor of a given code decoded by a column layered decoder can be better or worse than that of a row layered decoder and these are the TSs structure and their layer profiles that determine the superiority of one algorithm over the other.

Finally, in Chapter 6, we proposed a novel state-space model for the behavior of SPA over the AWGN channel in the vicinity of an ETS. We then used this model to evaluate the failure probability of ETSs of an LDPC code and to estimate the code's error floor. Unlike the well-known linear state-space model of an ETS, the proposed model takes into account the inherent non-linearity of CN operations in the

early iterations of SPA. Moreover, rather than using a Taylor expansion around zero to approximate the CN operations, the operating point of the expansion in the proposed model is nonzero and is chosen dynamically in each iteration. As a result, the proposed model is more accurate in estimating the error floor, particularly for cases where the clipping threshold of the decoder is rather large. We demonstrated that the proposed semi-linear state-space model with dynamical operating point can be considered as a code-independent fully-analytical counterpart to the code-dependent semi-analytical method of Richardson [34]. We also showed that, in addition to having the advantage of being code-independent and fully-analytical, our method with extrapolation is both less complex and more accurate than the method of [34], particularly for cases where decoder messages have a large dynamic range.

## 7.2 Future Work

Insights gained from this work can enable further advancement of analysis and design of different types of LDPC codes and their corresponding decoders. As an example, suppose one is interested in construction of an LDPC code with a given degree distribution. Based on the degree distribution, possible structures of  $(a, b)$  trapping sets up to a certain range of  $a$  and  $b$  values can be identified, [32, 33]. Then, the proposed LIS-DE<sub>c</sub> method can be deployed to evaluate the harmfulness of various possible non-isomorphic TSs with  $a \leq a_{max}$  and  $b \leq b_{max}$  and the provided information may be used during the construction process to remove, only, the harmful TSs. Other future works may be the extension of LIS-DE<sub>c</sub> as well as semi-linear state-space methods to layered decoders. Moreover, an interesting line of inquiry would be to use the proposed method for error floor analysis of layered decoders, presented in Chapters 4 and 5, in the design of QC-LDPC codes with low error floor under layered decoding. Also, an exciting future direction would be to extend the approaches provided in this thesis to the analysis of TS harmfulness in the *non-binary* and *spatially coupled* LDPC codes.

## List of References

- [1] E. Cavus and B. Daneshrad, “A performance improvement and error floor avoidance technique for belief propagation decoding of LDPC codes,” in *Proc. 16th IEEE Int. Symp. Personal, Indoor Mobile Radio Commun.*, Los Angeles, CA, USA, Sep. 2005, pp. 2386–2390.
- [2] Y. Han and W. E. Ryan, “LDPC decoder strategies for achieving low error floors,” in *Proc. Inform. Theory Appl. Workshop*, San Diego, CA, USA, Jan. 2008, pp. 277–286.
- [3] Y. Zhang and W. E. Ryan, “Toward low LDPC-code floors: a case study,” *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1566–1573, Jun. 2009.
- [4] G. B. Kyung and C.-C. Wang, “Finding the exhaustive list of small fully absorbing sets and designing the corresponding low error-floor decoder,” *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1487–1498, Jun. 2012.
- [5] S. Zhang and C. Schlegel, “Controlling the error floor in LDPC decoding,” *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3566–3575, Sep. 2013.
- [6] S. K. Planjery, D. Declercq, L. Danjean and B. Vasić, “Finite alphabet iterative decoders—part I: decoding beyond belief propagation on the binary symmetric channel,” *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4033–4045, Oct. 2013.
- [7] D. Declercq, B. Vasić, S. K. Planjery and E. Li, “Finite alphabet iterative decoders—part II: towards guaranteed error correction of LDPC codes via iterative decoder diversity,” *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4046–4057, Oct. 2013.
- [8] S. Tolouei and A. H. Banihashemi, “Lowering the error floor of LDPC codes using multi-step quantization,” *IEEE Commun. Lett.*, vol. 18, no. 1, pp. 86–89, Jan. 2014.
- [9] X. Zhang and P. H. Siegel, “Quantized iterative message passing decoders with low error floor for LDPC codes,” *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 1–14, January 2014.
- [10] F. Cai, X. Zhang, D. Declercq, S. K. Planjery and B. Vasić, “Finite alphabet iterative decoders for LDPC codes: optimization, architecture and analysis,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 5, pp. 1366–1375, May 2014.

- [11] E. Dupraz, D. Declercq, B. Vasić and V. Savin, “Analysis and design of finite alphabet iterative decoders robust to faulty hardware,” *IEEE Trans. Commun.*, vol. 63, no. 8, pp. 2797–2809, Aug. 2015.
- [12] S. Kang, J. Moon, J. Ha and J. Shin, “Breaking the trapping sets in LDPC codes: Check node removal and collaborative decoding,” *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 15–26, Jan. 2016.
- [13] H. Hatami, D. G. M. Mitchell, D. J. Costello and T. E. Fuja, “A threshold-based min-sum algorithm to lower the error floors of quantized LDPC decoders,” *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2005–2015, Apr. 2020.
- [14] X. Xiao, B. Vasić, R. Tandon and S. Lin, “Designing finite alphabet iterative decoders of LDPC codes via recurrent quantized neural networks,” *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 3963–3974, Jul. 2020.
- [15] Y. Mao and A. H. Banihashemi, “A heuristic search for good low-density parity-check codes at short block lengths,” in *Proc. IEEE Int. Conf. Comm., Helsinki, Finland*, pp. 41–44, Jun. 2001.
- [16] T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel, “Selective avoidance of cycles in irregular LDPC code construction,” *IEEE Trans. Commun.*, vol. 52, pp. 1242–1247, Aug. 2004.
- [17] H. Xiao and A. H. Banihashemi, “Improved progressive-edge-growth (PEG) construction of irregular LDPC codes,” *IEEE Commun. Lett.*, vol. 8, no. 12, pp. 715–717, Dec. 2004.
- [18] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, “Regular and irregular progressive edge-growth tanner graphs,” *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.
- [19] M. Ivkovic, S. K. Chilappagari, and B. Vasić, “Eliminating trapping sets in low-density parity-check codes by using Tanner graph covers,” *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3763–3768, Aug. 2008.
- [20] X. Zheng, F. C.-M. Lau, and C. K. Tse, “Constructing short-length irregular LDPC codes with low error floor,” *IEEE Trans. Commun.*, vol. 58, no. 10, pp. 2823–2834, Oct. 2010.
- [21] R. Asvadi, A. H. Banihashemi, and M. Ahmadian-Attari, “Lowering the error floor of LDPC codes using cyclic liftings,” *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2213–2224, Apr. 2011.
- [22] S. Khazraie, R. Asvadi and A. H. Banihashemi, “A PEG construction of finite-length LDPC codes with low error floor,” *IEEE Commun. Lett.*, vol. 16, pp. 1288–1291, Aug. 2012.
- [23] D. V. Nguyen, S. K. Chilappagari, M. W. Marcellin, and B. Vasić, “On the construction of structured LDPC codes free of small trapping sets,” *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2280–2302, Apr. 2012.

- [24] X. Tao, Y. Li, Y. Liu, and Z. Hu, "On the construction of LDPC codes free of small trapping sets by controlling cycles," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 9–12, Jan. 2018.
- [25] S. Naseri and A. H. Banihashemi, "Construction of girth-8 QC-LDPC codes free of small trapping sets," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1904–1908, Nov. 2019.
- [26] S. Naseri and A. H. Banihashemi, "Spatially coupled LDPC codes with small constraint length and low error floor," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 254–258, Feb. 2020.
- [27] B. Karimi and A. H. Banihashemi, "Construction of QC LDPC codes with low error floor by efficient systematic search and elimination of trapping sets," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 697–712, Feb. 2020.
- [28] C. C. Wang, S. R. Kulkarni, and H. V. Poor, "Finding all small error-prone substructures in LDPC codes," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 1976–1999, May 2009.
- [29] M. Karimi and A. H. Banihashemi, "Efficient algorithm for finding dominant trapping sets of LDPC codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6942–6958, Nov. 2012.
- [30] M. Karimi and A. H. Banihashemi, "On characterization of elementary trapping sets of variable-regular LDPC codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5188–5203, Sep 2014.
- [31] Y. Hashemi and A. Banihashemi, "On characterization and efficient exhaustive search of elementary trapping sets of variable-regular LDPC codes," *IEEE Commun. Lett.*, vol. 19, pp. 323–326, Mar. 2015.
- [32] Y. Hashemi and A. H. Banihashemi, "New characterization and efficient exhaustive search algorithm for leafless elementary trapping sets of variable-regular LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6713–6736, Dec. 2016.
- [33] Y. Hashemi and A. H. Banihashemi, "Characterization of elementary trapping sets in irregular LDPC codes and the corresponding efficient exhaustive search algorithms," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3411–3430, May 2018.
- [34] T. Richardson, "Error floors of LDPC codes," in *Proc. 41th annual Allerton conf. on commun. control and computing*, Monticello, IL, USA, Oct. 2003, pp. 1426–1435.
- [35] C. A. Cole, S. G. Wilson, E. K. Hall, and T. R. Giallorenzi, "A general method for finding low error rates of LDPC codes," *submitted to IEEE Trans. Inf. Theory*, May 2006.
- [36] L. Dolecek, Z. Zhang, M. Wainwright, V. Anatharam, and B. Nikolic. "Evaluation of the low frame error rate performance of LDPC codes using importance sampling," in *Proc. IEEE Inf. Theory Workshop*, Lake Tahoe, CA, Sep. 2–6, 2007, pp. 202–207.

- [37] H. Xiao and A. H. Banihashemi, "Estimation of bit and frame error rates of finite-length low-density parity-check codes on binary symmetric channels," *IEEE Trans. Commun.*, vol. 55, no. 12, pp. 2234–2239, Dec. 2007.
- [38] E. Cavus, C. L. Haymes and B. Daneshrad, "Low BER performance estimation of LDPC codes via application of importance sampling to trapping sets," *IEEE Trans. Commun.*, vol. 57, no. 7, pp. 1886–1888, Jul. 2009.
- [39] L. Dolecek, P. Lee, Z. Zhang, V. Anatharam, B. Nikolic, and M. J. Wainwright, "Predicting error floors of structured LDPC codes: deterministic bounds and estimates," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 908–917, Aug. 2009.
- [40] X. Hu, Z. Li, B. Kumar, and R. Barndt, "Error floor estimation of long LDPC codes on magnetic recording channels," *IEEE Trans. Magn.*, vol. 46, no. 6, pp. 1836–1839, Jun. 2010.
- [41] H. Xiao, A. H. Banihashemi, and M. Karimi, "Error rate estimation of low-density parity-check codes decoded by quantized soft-decision iterative algorithms," *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 474–484, Feb. 2013.
- [42] J. Sun, "Studies on graph-based coding systems," Ph.D. dissertation, Dept. Elect. Eng., Ohio State Univ., Columbus, OH, USA, 2004.
- [43] B. Vasić, S. K. Chilappagari, D. V. Nguyen and S. K. Planjery, "Trapping set ontology," in *Proc. 47th Allerton Conf.*, Monticello, IL, 2009, pp. 1–7.
- [44] C. Schlegel and S. Zhang, "On the dynamics of the error floor behavior in (regular) LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3248–3264, Jul. 2010.
- [45] B. K. Butler and P. H. Siegel, "error floor approximation for LDPC codes in the AWGN channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7416–7441, Dec. 2014.
- [46] S. Tolouei and A. H. Banihashemi, "Fast and accurate error floor estimation of quantized iterative decoders for variable-regular LDPC codes," *IEEE Comm. Lett.*, vol. 18, no. 8, pp. 1283–1286, Aug. 2014.
- [47] H. Hatami, D. G. M. Mitchell, D. J. Costello and T. E. Fuja, "Performance bounds and estimates for quantized LDPC decoders," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 683–696, Feb. 2020.
- [48] A. Farsiabi and A. H. Banihashemi, "Error floor estimation of LDPC decoders - A code independent approach to measuring the harmfulness of trapping sets," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2667–2679, May. 2020.
- [49] N. Raveendran, D. Declercq and B. Vasić, "A Sub-graph expansion-contraction method for error floor computation," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 3984–3995, July 2020.

- [50] A. Tomasoni, S. Bellini and M. Ferrari, “Thresholds of absorbing sets in low-density parity-check codes,” *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3238–3249, Aug. 2017.
- [51] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb 2001.
- [52] O. Milenkovic, E. Soljanin, and P. Whiting, “Asymptotic spectra of trapping sets in regular and irregular LDPC code ensembles,” *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 39–55, Jan. 2007.
- [53] M. Mansour, N. Shanbhag, “High-throughput LDPC decoders,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 6, pp. 976–996, Dec. 2003.
- [54] D. Hocevar, “A reduced complexity decoder architecture via layered decoding of LDPC codes,” in *Proc. IEEE Workshop Signal Processing and Systems (SIPS.04)*, Austin, TX, Oct. 2004, pp. 107–112.
- [55] H. Xiao and A. H. Banihashemi, “Graph-based message-passing schedules for decoding LDPC codes,” *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2098–2105, Dec. 2004.
- [56] A. Nough and A. H. Banihashemi, “Reliability-based schedule for bit-flipping decoding of low-density parity-check codes,” *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2038–2040, Dec. 2004.
- [57] T. Brack, M. Alles, F. Kienle, N. Wehn, “A synthesizable IP core for WiMAX 802.16e LDPC code decoding,” in *Proc. IEEE 17th Int. Symp. Personal Indoor and Mobile Radio Communications*, Sept. 2006, pp. 1–5.
- [58] Z. Wang, Z. Cui, “Low-complexity high-speed decoder design for quasi-cyclic LDPC codes,” *IEEE Trans. VLSI Syst.*, vol. 15, no. 1, pp. 104–114, Jan. 2007.
- [59] K. Gunnam, G. Choi, M. Yeary, M. Atiquzzaman, “VLSI architectures for layered decoding for irregular LDPC codes of WiMax,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, June 2007, pp. 4542–4547.
- [60] E. Sharon, S. Litsyn, J. Goldberger, “Efficient serial message-passing schedules for LDPC decoding,” *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4076–4091, Nov. 2007.
- [61] C.-H. Liu, S.-W. Yen, C.-L. Chen, H.-C. Chang, C.-Y. Lee, Y.-S. Hsu, S.-J. Jou, “An LDPC decoder chip based on self-routing network for IEEE 802.16e applications,” *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 684–694, March 2008.
- [62] K. Zhang, X. Huang and Z. Wang, “High-throughput layered decoder implementation for quasi-cyclic LDPC codes,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 985–994, August 2009.

- [63] Z. Cui, Z. Wang, X. Zhang, “Reduced-complexity column-layered decoding and implementation for LDPC codes,” *IET Commun.*, vol. 5, no. 15, pp. 2177–2186, 2011.
- [64] Z. Wang, Z. Cui, and J. Sha, “VLSI design for low-density parity-check code decoding,” *IEEE Circuits Syst. Mag.*, vol. 11, no. 1, pp. 52–69, Feb. 2011.
- [65] S. Kumawat et al., “High-throughput LDPC-decoder architecture using efficient comparison techniques and dynamic multi-frame processing schedule,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 5, pp. 1421–1430, May 2015.
- [66] J. Zhang and M.-P.-C. Fossorier, “Shuffled iterative decoding,” *IEEE Trans. Commun.*, vol. 53, no. 2, pp. 209–213, Feb. 2005.
- [67] F. Guilloud, E. Boutillon, J. Touse, and J. Danger, “Generic description and synthesis of LDPC decoders,” *IEEE Trans. Commun.*, vol. 55, no. 11, pp. 2084–2091, Nov. 2007.
- [68] Z. Cui, Z. Wang, X. Zhang, and Q. Jia, “Efficient decoder design for high-throughput LDPC decoding,” in *Proc. IEEE Asia Pacific Conf. on Circuits and Syst.*, Dec. 2008, pp. 1640–1643.
- [69] Y. L. Ueng, C. J. Yang, and C. J. Chen, “A shuffled message-passing decoding method for memory-based LDPC decoders,” in *Proc. IEEE ISCAS 2009*, May 2009, pp. 892–895.
- [70] J. Lin, J. Sha, Z. Wang, and L. Li, “An improved min-sum based column-layered decoding algorithm for LDPC codes,” in *Proc. IEEE Workshop on Signal Processing Systems (SIPS)*, 2009
- [71] A. I. V. Casado, M. Griot and R. D. Wesel, “LDPC decoders with informed dynamic scheduling,” *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3470–3479, December 2010.
- [72] H. Lee and Y. Ueng, “LDPC decoding scheduling for faster convergence and lower error floor,” *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3104–3113, Sept. 2014.
- [73] N. Raveendran and B. Vasić, “Trapping set analysis of horizontal layered decoder,” in *Proc. Int. Conf. Commun. (ICC)*, Kansas City, MO, 2018, pp. 1–6.
- [74] F. Angarita, J. Valls, V. Almenar and V. Torres, “Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor,” *IEEE Trans. Circuits Syst. I*, vol. 61, no. 7, pp. 2150–2158, July 2014.
- [75] X. Chen, J. Kang, S. Lin and V. Akella, “Hardware implementation of a backtracking-based reconfigurable decoder for lowering the error floor of quasi-cyclic LDPC codes,” *IEEE Trans. Circuits Syst. I*, vol. 58, no. 12, pp. 2931–2943, Dec. 2011.
- [76] S. Kim, “Trapping set error correction through adaptive informed dynamic scheduling decoding of LDPC codes,” *IEEE Commun. Lett.*, vol. 16, no. 7, pp. 1103–1105, July 2012.

- [77] B. K. Butler and P. H. Siegel, “Numerical issues affecting LDPC error floors,” in *Proc. IEEE Global Telecommun. Conf.*, Anaheim, CA, 2012, pp. 3201–3207.
- [78] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA, USA: SIAM, 2000.
- [79] R. S. Varga, *Matrix Iterative Analysis*, 2nd ed, Berlin: Springer, 2000.
- [80] D. Noutsos, “Perron-Frobenius theory and some extensions”, Como, Italy, May 2008, [Presentation Slides]. Available: <http://www.math.uoi.gr/dnoutsos/Papers-pdf-files/slide-perron.pdf>.
- [81] R. A. Horn, C. R. Johnson, *Matrix Analysis*, Cambridge, U.K.: Cambridge Univ. Press, 1985.
- [82] *IEEE Standard for Local and Metropolitan Area Networks—Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1*, IEEE Standard 802.16e-2005 and 802.16-2004/Cor 1-2005, Feb. 2006.
- [83] J. Zhao, F. Zarkeshvari and A. H. Banhashemi, “On implementation of min-sum algorithm and its modifications for decoding LDPC codes,” *IEEE Trans. Commun.*, vol. 53, no. 4, pp. 549–554, April 2005.
- [84] C. B. Schlegel and L. C. Perez, *Trellis and Turbo Coding: Iterative and Graph-Based Error Control Coding*, 2nd ed, Hoboken, New Jersey, USA: Wiley, 2015.
- [85] J. Chen and M. P. C. 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.
- [86] D. J. C. MacKay and M. S. Postol, “Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes,” *Electron. Notes Theoretical Comput. Sci.*, vol. 74, pp. 97–104, Oct. 2003.
- [87] D. Mackay, “Encyclopedia of sparse graph codes,” URL: <http://www.inference.phy.cam.ac.uk/mackay/codes/data.html>.
- [88] I. Djurdjevic, J. Xu, K. Abdel-Ghaffar, and S. Lin, “A class of low density parity-check codes constructed based on Reed–Solomon codes with two information symbols,” *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 317–319, Jul. 2003.
- [89] Y. Kou, S. Lin, and M. Fossorier, “Low-density parity-check codes based on finite geometries: A rediscovery and new results,” *IEEE Trans. Inf. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.
- [90] Q. Huang, Q. Diao, S. Lin, and K. Abdel-Ghaffar, “Cyclic and quasi-cyclic LDPC codes on constrained parity-check matrices and their trapping sets,” *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 2648–2671, May 2012.

- [91] H. Liu, Y. Li, L. Ma, and J. Chen, "On the smallest absorbing sets of LDPC codes from finite planes," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 4014-4020, June 2012.
- [92] J. Justesen, T. Høholdt, and J. Hjaltason, "Iterative list decoding of some LDPC codes," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4276-4284, Nov. 2007.
- [93] B. M. Kurkoski and H. Yagi, "Quantization of binary-input discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4544-4552, Aug. 2014.
- [94] A. Milanese, J. Sun, and T. Nishikawa, "Approximating spectral impact of structural perturbations in large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 81, no. 4, Apr. 2010, Art. no. 046112.
- [95] J. Proakis, *Digital Communications*, 4th edition, Mc Graw-Hill, 2000
- [96] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599-618, Feb 2001.
- [97] W. E. Ryan and S. Lin, *Channel Codes: Classical and Modern*, New York, USA: Cambridge University Press, 2009.
- [98] K. Triantafyllopoulos, "Moments and cumulants of the multivariate real and complex Gaussian distributions," School Math., Univ. Bristol, Bristol, U.K., Tech. Rep. 02:11, 2002.