LOW DENSITY GRAPH CODES AND NOVEL OPTIMIZATION STRATEGIES FOR INFORMATION TRANSFER OVER IMPAIRED MEDIUM

by

Cheng-Chun Chang

B.S. National Tsing Hua University, 2001

M.S. National Taiwan University, 2003

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2008

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Cheng-Chun Chang

It was defended on

October 10, 2008

and approved by

Allen C. Cheng, Assistant Professor, Department of Electrical and Computer Engineering

Ching-Chung Li, Professor, Department of Electrical and Computer Engineering

Heung-No Lee, Assistant Professor, Department of Electrical and Computer Engineering

Xinfu Chen, Professor, Department of Mathematics

Zhi-Hong Mao, Assistant Professor, Department of Electrical and Computer Engineering

Dissertation Director: Heung-No Lee, Assistant Professor, Department of Electrical and Computer Engineering

LOW DENSITY GRAPH CODES AND NOVEL OPTIMIZATION STRATEGIES FOR INFORMATION TRANSFER OVER IMPAIRED MEDIUM

Cheng-Chun Chang, PhD

University of Pittsburgh, 2008

Effective methods for information transfer over an imperfect medium are of great interest. This thesis addresses the following four topics involving low density graph codes and novel optimization strategies.

Firstly, we study the performance of a promising coding technique: low density generator matrix (LDGM) codes. LDGM codes provide satisfying performance while maintaining low encoding and decoding complexities. In the thesis, the performance of LDGM codes is extracted for both majority-rule-based and sum-product iterative decoding algorithms. The ultimate performance of the coding scheme is revealed through distance spectrum analysis. We derive the distance spectral for both LDGM codes and concatenated LDGM codes. The results show that serial-concatenated LDGM codes deliver extremely low error-floors. This work provides valued information for selecting the parameters of LDGM codes.

Secondly, we investigate *network-coding* on relay-assisted wireless multiple access (WMA) networks. Network-coding is an effective way to increase robustness and traffic capacity of networks. Following the framework of network-coding, we introduce new network codes for the WMA networks. The codes are constructed based on sparse graphs, and can explore the diversities available from both the time and space domains. The data integrity from relays could be compromised when the relays are deployed in open areas. For this, we propose a simple but robust security mechanism to verify the data integrity.

Thirdly, we study the problem of bandwidth allocation for the transmission of multiple sources of data over a single communication medium. We aim to maximize the overall user satisfaction, and formulate an optimization problem. Using either the logarithmic or exponential form of satisfaction function, we derive closed-form optimal solutions, and show that the optimal bandwidth allocation for each type of data is piecewise linear with respect to the total available bandwidth.

Fourthly, we consider the optimization strategy on recovery of target spectrum for filter-array-based spectrometers. We model the spectrophotometric system as a communication system, in which the information content of the target spectrum is passed through distortive filters. By exploiting non-negative nature of spectral content, a non-negative least-square optimal criterion is found particularly effective. The concept is verified in a hardware implementation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to thank my advisor Professor Heung-No Lee for his invaluable guidance, helps and support in so many ways. I would also like to thank Professor Zhi-Hong Mao for his constant encouragement and support. I express my sincere gratitude to them for being wonderful mentors throughout the years. I am deeply indebted to Prof. Allen C. Cheng, Prof. Ching-Chung Li, and Prof. Xinfu Chen for their kind helps. It was my great honor to have them as my committee members.

I am grateful to Mr. Bill Choi and Mr. David Askey for supervising me through the cooperative projects. If I have made any progress, the fruits of success belong to them. I enjoyed and learned a lot from their sharp industrial viewpoints.

I wish thank my collaborators, Jianming Wu, Ning Yao, Xiaofei Song, and Mir Hamza Mahmood, for the helpful discussions on the researches; with them it has been a wonderful working experience. I would like to take this opportunity to thank the staff members in the Department of Electrical and Computer Engineering, Sandy Weisberg, Theresa Costanzo, and Julie DePascale, for helping me with complex administrative matters.

Finally, I would like to devote especial thanks to my parents, Mr. Tung-Fu Chang and Mrs. Yueh-Mei Shih, and my dearest wife, Hsin-Ling Cheng, for their constant and faithful support; without them this dissertation would be very far away from completion.

1.0 INTRODUCTION

Today, communication systems prevail in our daily lives in so many different ways. The mobile phones at our hands and the computer terminals in our offices or homes are some classical examples that sustain information transfer for the human needs in modern life. It is almost endless to list the applications involving the use of communication systems nowadays [1]. In the most fundamental sense, communication systems involve the transmission of information from one point (or region) to another. We want to transmit information across impaired channels (e.g., distortion, noisy or bandwidth limited channels) so that the receiver can determine this information with high fidelity, despite the imperfection of the channels. Effective methods to achieve this are of great interest. In this thesis, we develop and address the following four areas regarding high performance coding techniques and novel optimization strategies: (i) low density graph codes for channel coding, (ii) low density graph codes for network coding on multiple access networks, (iii) (perceptual-based) optimization on bandwidth allocation, and (iv) (non-negative constraint) optimization on content recovery in spectrophotometric systems.

1.1 CODING TECHNIQUES FOR EFFECTIVE INFORMATION TRANSFER

Channel coding theorem states that as long as the rate of information transmission is below the channel capacity, it is possible to have error-free transmission over the channel [2]. This is achieved by using a channel encoder in the transmitter and a channel decoder in the receiver. Nevertheless, the channel coding theorem is in fact a nonconstructive existence proof that error-free communication is possible over a noisy channel, but the theorem does not tell us how to design the best channel encoder and decoder. Recently to achieve the channel capacity, a considerable amount of work has been put in by using modern coding techniques — code on graphs and iterative decoding. In the following, we introduce modern coding techniques of both channel coding and network coding to achieve effective information transfer.

1.1.1 Low density graph codes for channel coding: introduction and contribution of this thesis

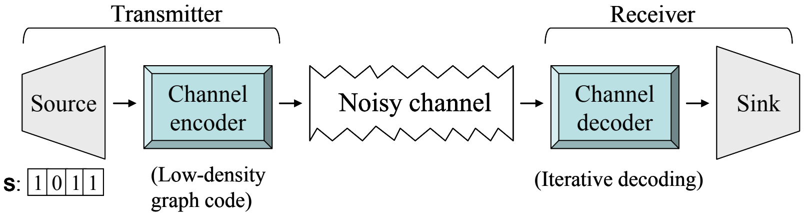


Figure 1. Block diagram of channel coding in communication systems.

Modern coding techniques--codes on graph and iterative decoding--have had a strong impact on achieving reliable communications [18]. Codes on sparse graph and iterative decoding were originally devised by Gallager in 1960, and then long forgotten [26]. It was rediscovered by Berrou, Glavieux, and Thitimajshima in 1993 in the form of turbo codes [3], and independently rediscovered by Mackay and McNeal in the mid 90's in the form close to Gallager's original construction [27]. Nowadays, the modern coding techniques have played an important role in reliable transmission of most current wireless communication systems such as DVB-S2, WiMAX, and 10GBase-T Ethernet. A block diagram of the channel coding system is depicted in Figure 1, in which the binary information from a source is encoded via a low density graph code, passed through a noisy channel, and then decoded iteratively on the receiver side.

In chapter 2 and chapter 3, we address a new high-performance channel coding technique with low system complexity, namely low density generator matrix (LDGM) codes. LDGM codes are systematic linear block code with low density generator matrix. LDGM codes provide good performance while maintaining low encoding and decoding complexities. The performance of LDGM codes under majority-rule based (MB) and sum-product (SP) decoding algorithms are extracted via density evolution techniques. The ultimate performance of the coding scheme can be revealed through distance spectrum analysis, which indicates how close a codeword in a code is to the others. We show how the distance spectrum for both LDGM codes and concatenated LDGM codes can be found. The results show that serially concatenated LDGM codes deliver extremely low error-floors. This work reveals the performance of LDGM codes and provides valued information for selecting the parameters for LDGM codes.

1.1.2 Low density graph codes for network coding on wireless multiple access networks: introduction and contribution of this thesis

Wireless multiple access networks are the main basis for current and future wireless communication networks. A wireless multiple access network consists of multiple users and a single access point. Examples of wireless multiple access networks include the uplinks in satellite communication systems, the uplinks in cellular networks, the downlinks in peer-to-peer file-distribution networks, a basic service set (BSS) in wireless local area networks (e.g., IEEE 802.11), and a BSS in wireless metropolitan area networks (e.g., IEEE 802.16). However, the performance of wireless multiple access networks can be severely limited due to the mutual interference and fading characteristics of wireless channels. Techniques that efficiently use the scarce resource of the wireless channels are demanded.

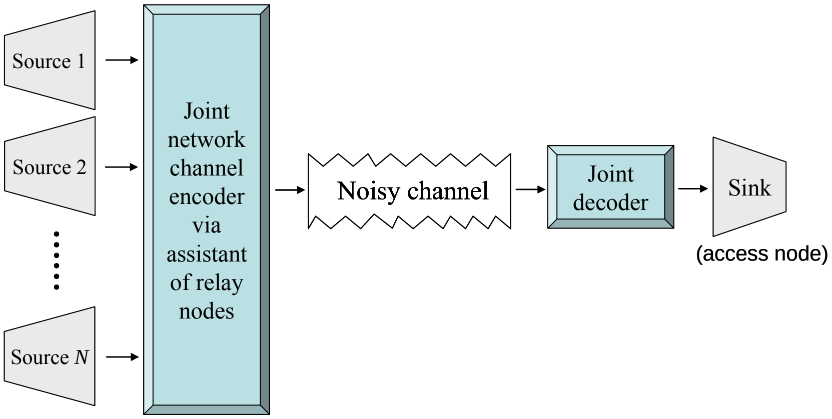


Figure 2. Block diagram of network coding on relay assisted multiple access networks.

We devote ourselves to wireless multiple access networks from *network-coding* perspective. Network-coding is a novel mechanism [25][52]. In December 2007, the *Network World NEWS* stated that network coding could re-engineer routing, content distribution, and wireless vendors. Many high tech's biggest names such as Microsoft, HP, and Intel are starting to embrace the network coding techniques in an effort to boost throughput, scalability and efficiency of wired or wireless networks [5]. The core principle behind network coding is to allow intermediate nodes to encode packets. That is, when an intermediate node (i.e., relay node) initiates to send a packet to another node, the intermediate node generates and sends a *combination* version of its incoming messages, instead of relaying the incoming messages individually. On the other side, when destination sink nodes receive enough independent combination versions of packets, they can reconstruct the original information. For linear combination, as an example, the reconstruction process is similar to solving a system of linear equations [6].

In chapter 4 and chapter 5, we address the new issues -- performance and security -- of network-coding on relay assisted wireless multiple access (WMA) networks. The block diagram of the network-coding WMA networks is depicted in Figure 2, in which messages from different source nodes are jointly encoded by means of the helps of relay nodes. On the receiver side, all the received messages are recovered jointly through iterative decoding. Under the framework of network-coding, space-time mesh codes are proposed. The codes can flexibly exploit the diversities from time-domain and from space-domain. We note that relay nodes could be deployed in open area, and might be vulnerable to various kinds of physical tampering. For this problem, we present a simple, but robust, security method that can effectively verify the data integrity from relay nodes, within the framework of the proposed network-coding scheme.

1.2 EFFECTIVE OPTIMIZATION STRATEGIES FOR INFORMATION TRANSFER

Optimization problems prevail in most disciplines such as engineering, physics, mathematics, economics, administration, commerce, social sciences, and even politics. The process of optimization is the process of obtaining the “best”, if it is possible to measure and change what is “good” or “bad”. Optimization theory is the branch of mathematic encompassing the quantitative study of optima and methods for finding them [7]. In the thesis, we provide novel approaches for the quantization of optima in two promising applications: bandwidth allocation in perceptual-based communication systems and recovery of target spectrum in filter-based spectrophotometric systems.

1.2.1 Optimization on bandwidth allocation: introduction and contribution of this thesis

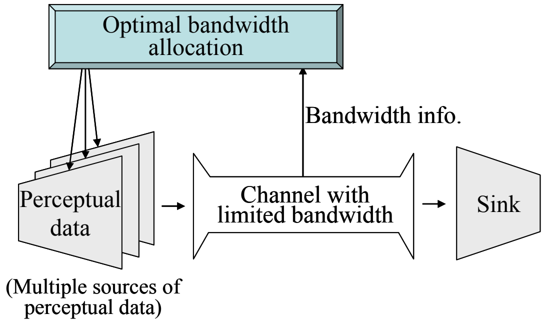

Optimal bandwidth allocation
Bandwidth info.
Perceptual data
Channel with limited bandwidth
Sink
(Multiple sources of perceptual data)

Figure 3. Block diagram of transmission of multiple sources of data.

To effectively and fairly allocate the transmission bandwidth for multiple sources of perceptual data over a common communication channel is demanded. Promising applications of this scenario include MSN, Skype, and/or tele-surgery systems. Figure 3 depicts the system block diagram, in which an optimal bandwidth allocation mechanism is critical to adjust the source rates optimally. Despite the fact that significant improvement in communication infrastructure has been attained in recent years and a point to point communication channel with large bandwidth would be available to a customer, a channel with smaller bandwidth usually costs less to the customer. Hence, how to optimally transmit multiple sources of data over a channel with limited bandwidth is of great interest. The issues concerning the optimal bandwidth allocation in reasonable time and fair manner remain challenging, and need to be solved.

In chapter 6, we study this optimization problem from a human perceptual perspective. We aim to maximize the overall user satisfaction, and formulate the optimization problem. Using either the logarithmic or exponential form of satisfaction function, we are able to derive closed-form solutions for the optimization problem. The results are quite interesting. The optimal bandwidth allocation for each type of data is shown to be piecewise linear with respect to the total available bandwidth.

1.2.2 Optimization on content recovery in spectrophotometric systems: introduction and contribution of this thesis

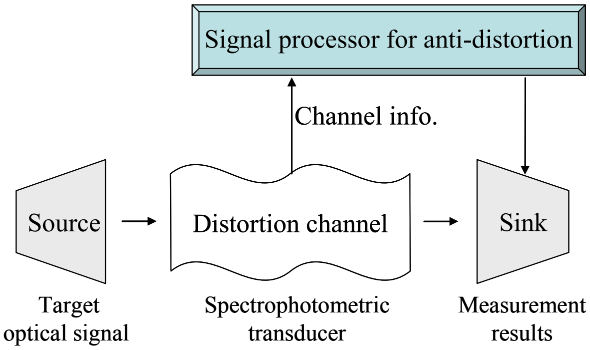


Figure 4. Block diagram of a spectrophotometric system.

The spectrophotometric system can be seen as a communication system.

Spectrophotometry is more and more often the method of choice not only in analysis of (bio)chemical substances, but also in the identification of physical properties of various objects and their classification. The applications of spectrophotometry include such diversified tasks as monitoring of optical telecommunication links, assessment of quality of food, forensic classification of papers, detecting of insect infestation of seeds, and classification of textiles [8]. In all those applications, large numbers of data, generated by spectrophotometers, can be processed by various digital means in order to extract measurement information.

In the chapter 7, we study the filter-array based spectrometers. We aim to design a fine spectrometers based on low-cost (and hence low transduction quality) filters. As depicted in Figure 4, we consider a filter-based spectrometer as a communication system, in which the information content of the light spectrum is passed through distortive channels – the imperfect filters. Effective methods for anti-distortion at the receiver side (or measuring side) are critical. In this thesis, we work on the estimations of target spectrum. A series of optimal estimators for effectively recovering target spectrum is introduced. By exploiting the novel optimization strategy – the non-negative least square (NNLS) criterion, we can effectively estimate the target spectra with high fidelity. The concept is verified in a hardware implementation. We show the feasibility of a fine spectrometer on-a-chip based on a low-cost filter-array.

1.3 THESIS OUTLINE

From chapter 2 to chapter 5, we consider effective methods for information transfer from coding perspective. In chapter 2, we present the performance analysis of regular LDGM codes under majority-rule based iterative decoding algorithms and sum-product iterative decoding algorithms. The distance spectrum analysis of LDGM codes are provided in chapter 3, in which serially-concatenated and parallel-concatenated LDGM codes are investigated as well. In chapter 4, we show the extension of the application of graph codes to relay assisted wireless multiple access networks, in which the notion of network-coding is involved. Security issues and attack resilient methods are considered in chapter 5. From chapter 6 to chapter 7, we consider the effective methods for information transfer from optimization perspective. In chapter 6, based on user satisfaction, we present the strategy for bandwidth allocation for transmission of multiple sources of perceptual data over a shared channel. By considering a spectrophotometric system as a communication system, we present effective optimization methods for content recovery with high fidelity. We present some future directions in chapter 8.

2.0 PERFORMANCE ANALYSIS OF REGULAR LDGM CODES UNDER MB AND SP ITERATIVE DECODING ALGORITHMS

We investigate the performance of regular LDGM codes under both the hard-decision majority-rule based (MB) iterative decoding algorithm and the soft-decision sum-product (SP) iterative decoding algorithm. The two eminent error-performances of LDGM codes, *threshold* and *error-floor*, are extracted. For the MB algorithm, we derive a recursive expression for performance analysis. Furthermore, a non-recursive lower-bound expression for the error floors is obtained.. On the other side, we provide a fast simulation method useful to investigate the performance of LDGM codes under the SP algorithm. Supported by the *confidence interval analysis*, the presented method is, for example, 10^8 times quicker than the Monte-Carlo computer simulation for bit-error-rate (BER) in 10^{-10} region. With these tools, one can efficiently assess the performance of LDGM codes of a given degree, and select a best LDGM code under the trade-off between performance and encoding/decoding complexities.

2.1 INTRODUCTION

Low-density generator-matrix (LDGM) codes with moderate code length are of interest not only because they can provide satisfying performance at the moderate block length while maintaining low encoding/decoding complexities [19][21][22][23], but also because the inherent systematic form of LDGM codes make the codes useful to new applications such as cooperative wireless multiple access relay network [35] and joint source-channel encoding system [24].

In this chapter, we investigate the performance of LDGM codes under both hard-decision majority-rule based (MB) iterative decoding algorithm and soft-decision sum-product (SP) iterative decoding algorithm. MB algorithm has drawn significant amount of interests in the past thanks to its simplicity and low computation complexity, which result in fast decoding ability and less hardware requirement [23][29][30][31]. On the other side, SP algorithm is well-known for its ability to approach the Shannon capacity in variouis channels [26][27][28].

For the MB algorithm, by assuming infinite block length and borrowing Gallager's framework [26], we derive a recursive expression which predicts both the threshold and error-floor behaviors of the codec precisely. Moreover, based on the recursive expression, we further derive a non-recursive lower-bound expression as the function of the *degree of variable nodes*. The bound is tight and hence allow us to efficiently assess the best performance of the codec for given degrees of LDGM codes.

For the SP algorithm, we propose a fast simulation method for performance analysis. The method is constructed using the general idea of *density evolution [28]*[54]. The method tracks the mean of *log-likelihood ratio* (LLR) samples, and thus enables faster evaluation and removes the need for time-consuming Monte-Carlo simulation. To systematize this approach, we include a *confidence interval analysis* which allows us to determine the number of samples required for a targeted accuracy.

The rest of the chapter is organized as follows. In section 2.2, we briefly introduce LDGM codes. In section 2.3, we briefly review the MB iterative decoding algorithm and then derive the recursive expression and the lower bound expression. The fast simulation method is discussed in section 2.4. Results and discussions are providd in section 2.5. Finally, we make a conclusion in section 2.6.

2.2 LDGM CODES

LDGM codes are linear block codes with parity check matrix **H** = [**P** ; **I**], where **P** is an (n - k) by k sparse matrix and **I** is the (n-k) by (n-k) identity matrix. k denotes the number of input bits and n denotes the number of output bits of an LDGM encoder. The **P** matrix can be obtained from random generation. An LDGM code is called *regular* if both the number of 1's in a column in the **P** matrix and that in a row stay fixed for all columns and rows. Though irregularity can provide performance improvement, regularity could lead to simplified modular hardware implementation. We denote the *degree* of a variable node as d_v, which is the number of ones in a column in the **P** matrix**.** Similarly, the *degree* of a check node, d_c, represents the number of ones in a row in the **H** matrix. The code can be completely specified by a bipartite graph [32] consisting of check nodes and variable nodes. Since a systematic codeword is composed of message bits and parity-check bits, the variable nodes can be further separately into *message-bit variable* (MV) *nodes* and *parity-check-bit variable* (PV) *nodes*. Based on the structure of the **H** matrix, the code rate R of (d_v, d_c)-*regular* LDGM code is given by $R = 1/\left(d_v/\left(d_c - 1\right) + 1\right)$.

2.3 ERROR PERFORMANCE ANALYSIS UNDER MB ITERATIVE DECODING ALGORITHMS

A. MB iterative decoding algorithm: There are two steps in each iteration for the MB iterative decoding algorithm. The first step is done in a check node. The output binary message from the i^{th} check node, toward the j^{th} of its *d*c variable nodes, is the result of the *XOR* operation on the rest of d_c-1 incoming binary messages. That is, $c_{i,j} = \oplus\sum_{k=1,(k\neq j)}^{d_c-1}(v_{k,j})$, where the summation is done in modular-2 addition and $v_{k,j}$ is the binary message from the k^{th} variable node to the i^{th} check node. The second step is done in a variable node at which the *majority rule* is applied. Let f_j, where $f_j \in \{0,1\}$, denote the hard-decision binary value of the j^{th} received signal from channel for the j^{th} bit transmission. The output binary message from the j^{th} variable node, toward the i^{th} of its d_v check nodes, is obtained from the rest of $d_v - 1$ incoming messages, and is given by

$$v_{j,i} = \begin{cases} \tilde{f}_j, \; if \left(\sum\limits_{k=1,(k\neq i)}^{d_v-1} XOR(f_j, c_{k,j}) \right) \geq m \\ f_j, \; \text{o.w.} \end{cases} . \tag{2.1}$$

That is, if *m* or more than *m* incoming messages are violated, then the output binary message $v_{j,i}$ is the complement of f_j; otherwise, the output message holds the value of f_j. At the last iteration, the j^{th} bit is decoded to be $\tilde{f}_j$ if $\left(\sum_{k=1}^{d_v} XOR(f_j, c_{k,j})\right) \geq m$; otherwise, the j^{th} bit is decoded to be f_j. In the algorithm, weight *m* is an integer between 0 and d_v. The weight *m* needs to be carefully chosen in each iteration as it affects the performance of the MB iterative decoding algorithm. From the above description, we note that implementation of the MB iterative decoding algorithm is extreme simple.

B. Error performance analysis: We derive the recursive expression (2.2) and the tight lower bound expression (2.11). These expressions serve as efficient tools to extract the performance of the codec for given degrees of LDGM codes.

Due to the hard-decision characteristic of the MB decoding algorithm, assume all the coded bits are transmitted through a binary symmetric channel with error-probability P_0. Consider the error-performance on a MV node. Assume infinite code length and unfold the MB iterative decoding onto a cycle-free decoding tree. The error-probability for the message on the MV node after the i^{th} iteration can be expressed by the recursive form

$$P_{i+1} = P_0\left(1 - f(m, P_i)\right) + (1 - P_0)\left(g(m, P_i)\right) \tag{2.2}$$

where

$$\begin{aligned} f(m,x) = \sum_{l=m}^{d_v-1} \binom{d_v-1}{l} \left(\frac{1+(1-2P_0)(1-2x)^{d_c-2}}{2} \right)^{l} \\ \times \left(\frac{1-(1-2P_0)(1-2x)^{d_c-2}}{2} \right)^{d_v-1-l} \end{aligned} \tag{2.3}$$

and

$$\begin{aligned} g(m,x) = \sum_{l=m}^{d_v-1} \binom{d_v-1}{l} \left(\frac{1-(1-2P_0)(1-2x)^{d_c-2}}{2} \right)^{l} \\ \times \left(\frac{1+(1-2P_0)(1-2x)^{d_c-2}}{2} \right)^{d_v-1-l} . \end{aligned} \tag{2.4}$$

The first term in (2.2) represents the probability of an event that the MV node was in the error state originally and the error correction mechanism of MB algorithm is not triggered because less than *m* extrinsic messages, out of d_v- 1 total, are in violation. Thus the error in the variable node remains unchanged. The second term represents an event that the MV node was in the correct state and the error correction mechanism of MB algorithm is falsely triggered--because of *m* or more extrinsic messages in violation--and forces an error. It is interesting to compare this recursion result (2.2) to the Gallager's analysis result on regular LDPC codes [26]. The difference is that we have $(1-2P_0)(1-2x)^{d_c-2}$ in the recursion equation, instead of $(1-2x)^{d_c-1}$. This is a characteristic result of LDGM codes, which is caused by the one and only one connection from each PV node to the corresponding check node in the bipartite graph. This will cause the error floor effect in LDGM codes.

For a given channel error probability P_0, the weight *m* and the degrees d_v and d_c determine the recursive process (2.2) and hence determine the error-performance. The optimal weight *m* which minimizes P_{i+1} in (2.2) for the i^{th} iteration can be found by exhaustively searching for the integer between 0 and d_v, or by solving the smallest integer *m* which satisfies the following inequality [26]

$$\frac{1-P_0}{P_0} \le \left(\frac{1+(1-2P_0)(1-2P_i)^{d_c-2}}{1-(1-2P_0)(1-2P_i)^{d_c-2}} \right)^{2m-d_v+1} . \tag{2.5}$$

In the following, we derive the lower bound expression based on the recursive expression (2.2). Take the partial derivative of (2.2) respective to P_i, we obtain

$$\frac{\partial P_{i+1}}{\partial P_i} = -P_0 \frac{\partial f}{\partial P_i} + (1-P_0) \frac{\partial g}{\partial P_i} , \tag{2.6}$$

where

$$\frac{\partial f}{\partial P_i} = \binom{d_v-1}{m} \left(m \xi^{+^{m-1}} \xi^{-^{d_v-1-m}} \eta^{-} \right), \tag{2.7}$$

and

$$\frac{\partial g}{\partial P_i} = \binom{d_v - 1}{m} \left(m \xi^{-m-1} \xi^{+d_v - 1 - m} \eta^{+} \right). \tag{2.8}$$

In (2.7) and (2.8), the notations $\xi^+, \xi^-, \eta^+, \text{and } \eta^-$ are defined as $\xi^+ = (1 + (1-2P_0)(1-2P_i)^{d_c-2})/2$, $\xi^- = (1 - (1-2P_0)(1-2P_i)^{d_c-2})/2$, $\eta^+ = (d_c - 2)(1-2P_o)(1-2P_i)^{d_c-3}$, and $\eta^- = -(d_c - 2)(1-2P_o)(1-2P_i)^{d_c-3}$.

Without loss of generality, assuming P_0 and P_i are restricted between [0, 0.5], we observe that (2.6) is always non-negative, i.e., $\partial P_{i+1} / \partial P_i \geq 0$. This shows that (2.2) is a monotone increasing function of P_i. Therefore, by substituting $P_i = 0$ into (2.2), we obtain a lower bound expression of P_{i+1}.

To determine the optimal weight m such that we obtain the lowest bound, we resort to (2.5). By substituting $P_i = 0$ into (2.5), we have

$$\frac{1 - P_0}{P_0} \leq \left(\frac{1 + (1 - 2P_0)}{1 - (1 - 2P_0)} \right)^{2m - d_v + 1}. \tag{2.9}$$

If P_0 is restricted within the interval [0, 0.5], $(1-P_0)/P_0$ is not less than 1. The inequality is satisfied if and only if the exponent of the right hand side is greater than 1, i.e., $2m - d_v + 1 \geq 1$. The smallest integer that satisfies this inequality is $m = \lceil d_v / 2 \rceil$, where $\lceil \bullet \rceil$ is the ceiling operation. Notice that, at the last iteration, the number of available extrinsic messages for an MV node is d_v, instead of d_v-1. Then, the optimal weight m^* to achieve the lowest error floor P_EF for a given (d_v, d_c)-regular LDGM code is expressed as

$$m^* = \left\lceil \frac{d_v + 1}{2} \right\rceil. \tag{2.10}$$

Therefore, the lowest bound expression, which is only a function of the degree of variable nodes d_v, is given by

$$
\begin{aligned}
P_{LB} = P_0 & \left(1 - \sum_{l=m^*}^{d_v} \binom{d_v}{l}^{l} (1-P_0)^{l} (P_0)^{d_v - l} \right) \\
& + (1-P_0) \left(\sum_{l=m^*}^{d_v} \binom{d_v}{l}^{l} (P_0)^{l} (1-P_0)^{d_v - l} \right).
\end{aligned} \tag{2.11}
$$

We note that the results (2.2) and (2.11) are based on the cycle-free assumption, i.e., infinite code length. Thus, this renders the best performance of the codec for given degrees of LDGM codes.

2.4 FAST SIMULATION METHOD FOR LDGM CODES UNDER SPA ITERATIVE DECODING ALGORITHM

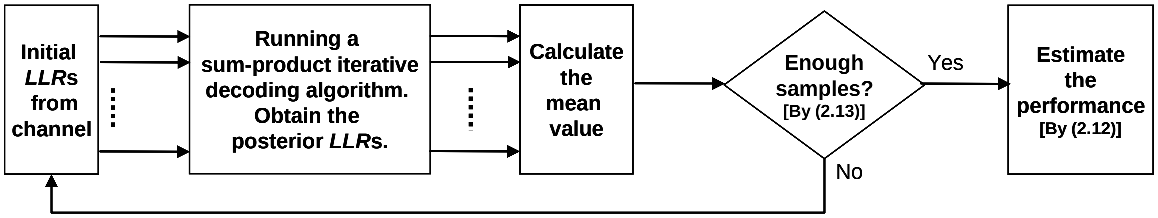

Initial *LLR*s from channel
Running a sum-product iterative decoding algorithm. Obtain the posterior *LLR*s.
Calculate the mean value
Enough samples? [By (2.13)]
Yes
No
Estimate the performance [By (2.12)]

Figure 5. The flow diagram of the fast simulation method

The simulation flow diagram of the fast simulation method is depicted in Figure 5. We feed the channel *LLRs* of the all-zero codeword to the standard SP iterative decoder. Then we obtain the posterior *LLRs*, $LLR^P s$, after running the SP iterative decoding algorithm. After that, instead of determining the decoded bits from these $LLR^P s$, we calculate the mean of the set of $LLR^P s$.

We note that the mean of $LLR^P s$ evolves only up to a certain limit for LDGM codes after which it stays at the same level no matter how many further iterations are carried out. This is one of the distinct features of LDGM codes: The information provided by the parity-check variable nodes always remains the same during the iterations. The reason for this is that all those variable nodes which participate in a single parity-check cannot pass along any extrinsic message to the check node. Extrinsic messages are generated by excluding the old message. But to those variable nodes that only have a *single* connection to the check nodes, this is not possible. This causes obstruction to continued evolution of density and hence results in error-floors. We show that this behavior can be well evaluated by our fast evaluation system.

In the output-symmetric AWGN channels, the distribution for channel $LLRs$ is symmetric and the symmetry is preserved throughout the message-passing decoding algorithm [28]. Thus, we may treat LLR^Ps as Gaussian distributed samples with mean -μ' and variance 2μ', see [54]. Then, the error-probability which is the Gaussian tail probability can be obtained by

$$P_e = \int_0^\infty \frac{1}{\sqrt{2\pi \cdot 2\mu'}} e^{-\frac{(\lambda+\mu')^2}{2\cdot 2\mu'}} d\lambda \,, \tag{2.12}$$

or by $0.5erfc(\sqrt{\mu'/4})$ in the form of complementary error function.

It is useful to determine the number of required LLR^P samples such that the estimated mean is accurate. For this, we resort to the confidence interval analysis, and obtain that

$$N = 8664 \times [erfc^{-1}(2P_e)]^2 \,, \tag{2.13}$$

where N is the required number of samples, $erfc^{-1}(\bullet)$ is the inverse of the complementary error function, and P_e is the target error probability in simulation. For example, having the number of samples around N = 10^5 is sufficient to provide a BER result of P_e = 10^{-10} region. We note that for the conventional Monte-Carlo computer simulation, if one wants to collect one thousand bit errors to have a smooth BER curve in simulation, the required number of LLR^P samples is $N^{'} \simeq 10^3 / P_e$, which requires $N' = 10^{13}$ for P_e = 10^{-10}. Therefore, the presented method is 10^8 times quicker in this example.

The derivation of (2.13) is elaborated as follows. The confidence interval for the mean α of a Gaussian distribution with known variance β^2 is given by [33] $CONF\{\bar{x}-\varepsilon \le \alpha \le \bar{x}+\varepsilon\}$, where $\bar{x}$ is the experimental mean of samples. The positive constant ε is a tolerant error and is associated by $\varepsilon = c\beta/\sqrt{N}$, where N is the number of samples, and c is a pre-calculated value depending on the confidence level γ . For example, $c = 3.291$ for the 99.9% confidence level $\gamma = 0.999$. We observe that $\varepsilon = 0.1$ is a reasonable value since the estimated mean within the error tolerance results in a negligible difference in (2.12). For a given target error probability P_e, we note that it is equivalent to investigate the Gaussian distribution with mean $\alpha = 4 \times erfc^{-1}(2P_e)$, from (2.12), and variance $\beta^2 = 8 \times erfc^{-1}(2P_e)$. Since $\varepsilon = c\beta/\sqrt{N}$, for $\varepsilon = 0.1$, we obtain $N = (c\beta/\varepsilon)^2 = 8664 \times [erfc^{-1}(2P_e)]^2$.

2.5 SIMULATION RESULTS AND DISCUSSION

Assuming BPSK modulation over AWGN channels, the error probability of the equivalent binary symmetric channel (BSC) for the MB algorithm is obtained by $P_0 = 0.5erfc\left(\sqrt{RE_b/N_0}\right)$, where R is the code rate, E_b is the energy per bit, and N_0 is the one-sided power spectral density of the noise. We evaluate the best possible performance of the codec by using the recursion form (2.2) in the following manner. While numerically evaluating the recursion expression (2.2), we test out all the possible choices of *m* in each iteration and then select the best value of *m* that results in the lowest error probability at the end of each iteration. We call it *dynamically optimized weight*. In addition, we let a large number of iterations to ensure the convergence of the recursive expression (2.2)

The dashed curves in Figure 6 are obtained from the recursive method for a set of rate 1/2 codes with degrees (8,9), (9,10), (10,11), (11,12), (12,13), (13,14), and for a set of rate around 1/3 codes with degrees (9,6), (10,6), (11,6), and (11,7). The solid curves in Figure 6 are obtained from the non-recursive low bound (2.11). As expected, the lower bound is asymptotically tight with respective to channel signal to noise ration (SNR). This is because, at high SNR, P_i in (2.2) can be evolved to a value very close to zero, and hence the assumption $P_i = 0$ used to derive the lower bound is validated. We note that the lower-bound expression predicts the performance well in the entire error floor region. Besides, we note that a code with small degrees exhibits a high error floor but a small threshold; whereas a code with large degrees shows a low error floor but a large threshold. Considering this trade-offs between error-floors and thresholds, the *best* degree d_v for LDGM codes under MB iterative decoding algorithm shall be around 10. LDGM codes with other degrees are bad, in terms of either high error floors or large thresholds. In addition, we also observe that, for rate half LDGM codes, the curves of (d_v, d_v +1) and (d_v +1, d_v +2) converge asymptotically for even d_v. This behavior of LDGM codes was also reported by J. Garcia-Frias [21] in which factor graph decoding scheme is applied.

Figure 7 shows the BER curves obtained from the Monte Carlo computer simulation. Ten iteration in the MB algorithm is used. Two randomly constructed (8, 9) and (9, 10) LDGM codes with length 6000 are simulated. To render a good threshold property while maintaining a low error-floor for the MB decoding algorithm, we use the following strategy for selecting the weight m. Initially, we choose the weight m which has the smallest threshold. This initial weight is used all the way through the last iteration, and at the last iteration the weight in use is calculated from (2.10) to deliver the lowest error floor. For the (8, 9) LDGM code, the weight of the smallest threshold is m =5 and the weight calculated from (2.10) is also m =5. Hence, we select the weight to be 5 throughout the iterations. For the (9, 10) LDGM codes, the weight of the smallest threshold is $m = 6$ whereas the weight calculated from (2.10) is $m=5$. We choose $m = 6$ for the iterations all the way until the last one, and then choose $m =5$ for the last iteration. The simulation results show that both the (8, 9) LDGM code and (9, 10) LDGM code not only can achieve the lower-bound, but also can achieve the thresholds. In other words, the derived recursive expression and non-recursive lower bound are tight and can successfully serve as efficient tools to access the error-performance of the codec.

Figure 8 shows the Monte-Carlo computer simulation results and the numerical results for rate half (7,8), (8,9), (9,10), (10,11), and (11,12) LDGM codes of length 4080 under the SP algorithm. Ten iterations are used in the SP decoding algorithm for the Monte-Carlo computer simulation. The numerical results are obtained from the fast evaluation method. The simulation and the numerical result match almost perfectly at the error-floor region, while for the waterfall region the simulation results are off a fraction of dB to the numerical results. We note that this gap can be closed by increasing the number of iterations in Monte-Carlo simulation. In Figure 8, it is also noted that a higher density code has a later waterfall region but a lower error-floor level.

Figure 9 shows the numerical results for rate 0.500 (10,11), rate 0.5238 (10,12), and rate 0.5455 (10,13) LDGM codes. We note that the higher the rate is, the larger E_b/N_0 gets for the waterfall region and the lower the error-floor gets. In fact, the error-floors remain at the same level in terms of SNR (E_s/N_0), and are determined by the minimum distance. From chapter 3, the minimum distance of this code is determined by d_v. Since d_v is 10 for all these codes, they all have the same minimum distance. As the rate changes slightly with the variation of the parameter d_c, the waterfall region is affected.

2.6 CONCLUSION

The presented performance evaluation framework provides the capability to quickly assess the performance of LDGM codes under MB or SP iterative decoding algorithms. Based on the framework, we have shown that, for both MB and SP algorithms, the number of 1's in the columns of generator matrix affects the error-floor level, whereas the number of 1's in the rows affects the waterfall region for rate ½ codes. For either MB or SP decoding algorithm, LDGM code exhibits two eminent error-performance behaviors: thresholds and error-floors.

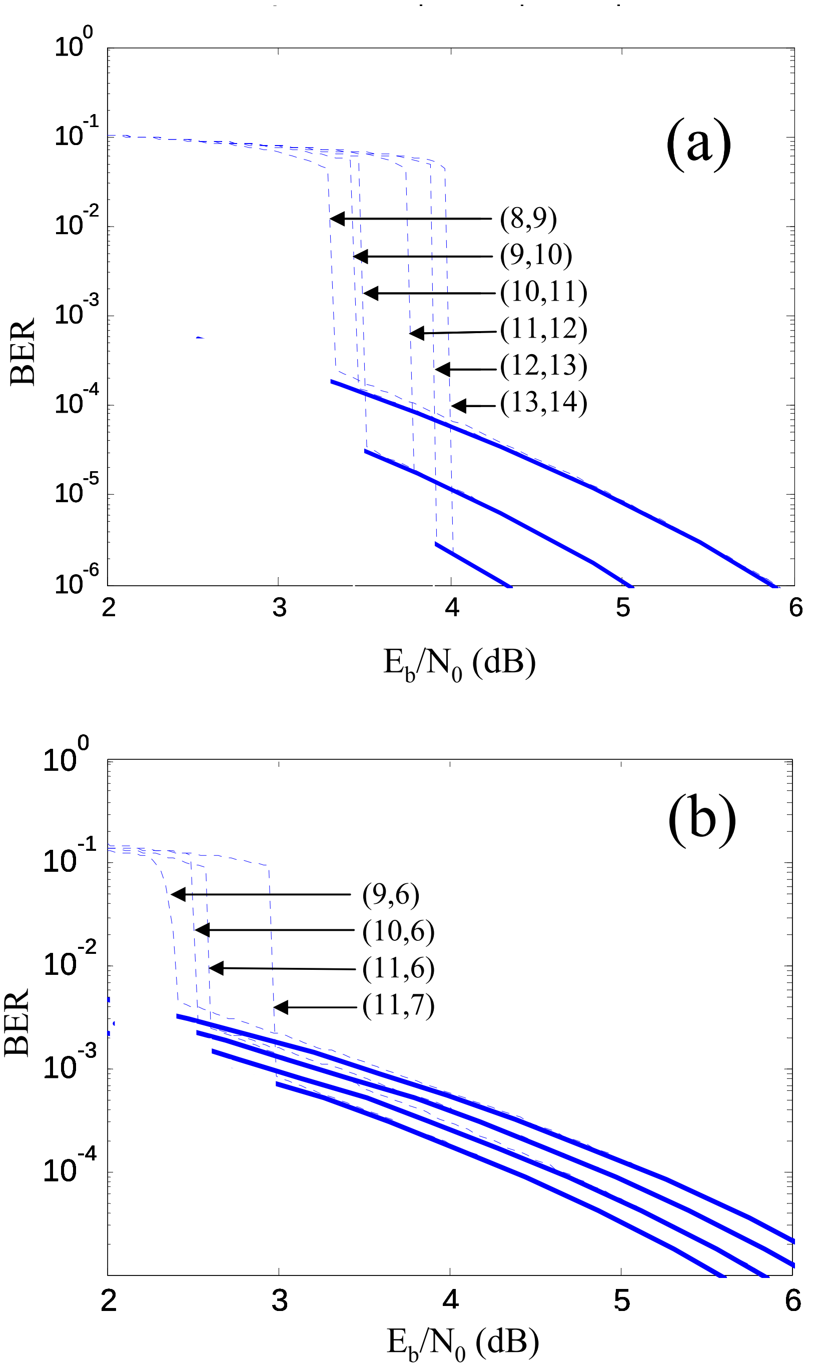


Figure 6. Analytic performance of LDGM codes under MB iterative decoding algorithm.

BER performance of (a) rate half LDGM codes, and (b) rate around 1/3 LDGM codes. In the figure, the dashed curves represent the BER obtained from the recursive expression (2.2) with dynamic optimized weight *m*; the solid curves represent the BER obtained by the non-recursive lower-bound expression (2.11).

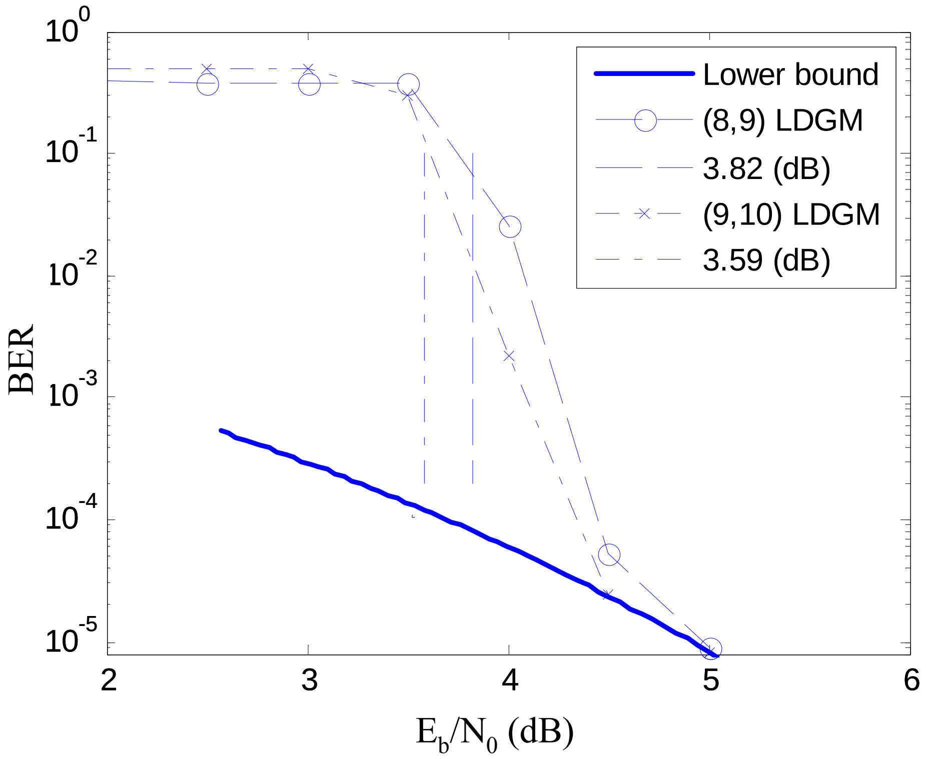


Figure 7. Simulated performance of LDGM codes under MB iterative decoding algorithm.

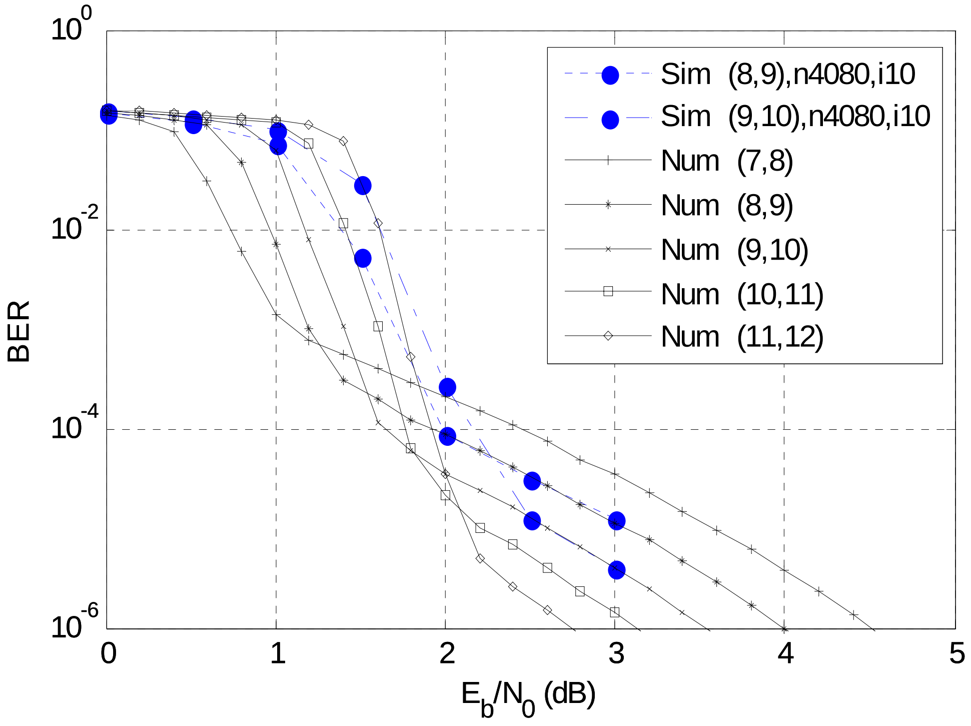


Figure 8. Performance of LDGM codes under the SP algorithm.

Performance of rate ½ LDGM codes with code length 4080 under AWGN channels. The Sim curves are obtained from Monte-Carlo computer simulation with 10 iterations. The Num curves are obtained from the fast simulation method. A higher density code has a later waterfall region but a lower error-floor level.

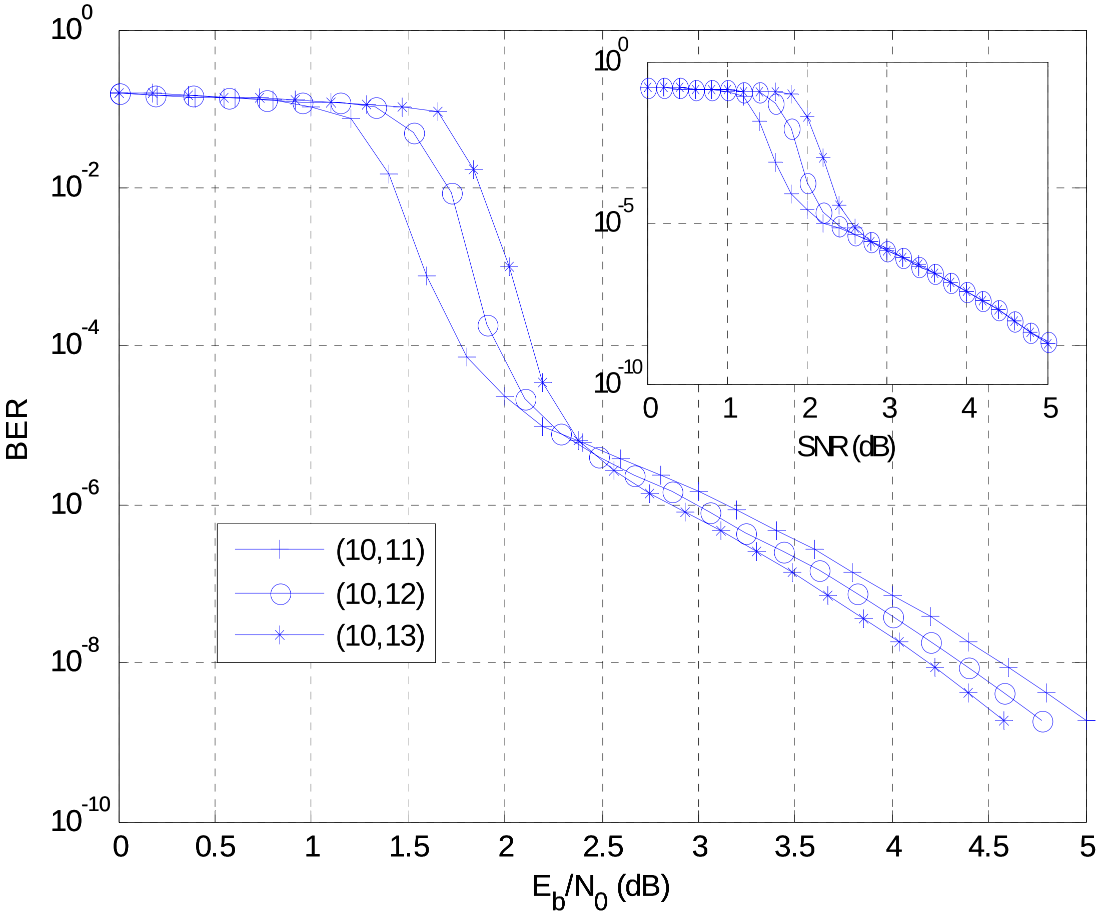


Figure 9. Performance of LDGM codes for different rates under the SP algorithm.

Performance of rate 0.500 (10,11), rate 0.5238 (10,12), and rate 0.5455 (10,13) LDGM codes obtained from the fast simulation method. The curves in the sub-figure are drawn with respective to SNR (E_s/N_0), while the curves in the main figure are calibrated to E_b/N_0. A higher rate code shows a later waterfall region but a slightly lower error-floor level

3.0 DISTANCE SPECTRUM ANALYSIS FOR LDGM CODES AND CONCATENATED LDGM CODES

Concatenated error-correction codes have been of interest in the past. By utilizing two relative weaker constituent codes, a powerful capacity-achieving code can be obtained. In this chapter, we investigate the concatenation of low-density generator matrix (LDGM) codes. By utilizing enumeration methods, we derive the average distance distributions not only for LDGM codes, but also for serially-concatenated and parallel-concatenated LDGM codes. The results serve as a fundamental step to investigate the bounds of the ensemble performance under the optimal maximum-likelihood (ML) decoder. We show that serially-concatenated LDGM codes provide extremely low error-floors.

3.1 INTRODUCTION

Concatenated error-correction codes have been of interest in the past. By utilizing two relative weaker constituent codes, it is possible to result in a new code with low-complexity but high error-correction performance [9][10]. In this chapter, we study the performance of serially-concatenated (SC-) and parallel-concatenated (PC-) systematic low-density generator-matrix (LDGM) codes, as well as LDGM codes.

LDGM codes are a simple variation of Gallager's low-density parity-check (LDPC) codes. Gallager's LDPC codes [26] are one of the known capacity achieving codes with linear decoding complexity. However, the encoding complexity is quadratic due to the dense generator matrix obtained from Gaussian elimination procedure. To have a linear encoding complexity as these in turbo codes, the notion of LDGM has been raised [21]. However, LDGM codes are relative weak codes due to the fact that the code includes the low weight codewords composing from the rows of the sparse generator matrix, and hence LDGM codes express noticeable error-floor levels.

To reduce the error-floor levels of LDGM codes, concatenation of two LDGM codes are introduced in the literatures [11][21]. The authors studied SC-LDGM codes through *EXIT functions* and PC-LDGM codes through *density evolution*. They stated that PC-LDGM codes and SC-LDGM codes provide a performance similar to turbo codes or irregular LDPC codes, while the encoding and decoding complexities are maintained to be low. In addition, the performance of SC-LDGM matches or out-performs the performance of irregular repeat accumulate (IRA) codes[1].

However, the analysis done by the EXIT functions or density evolution can only provide *threshold values* as the performance indices. It is still not clear what the capability of this class of concatenated codes is. To prove rigorously that indeed these concatenated codes can provide satisfying performance in various channel types, one starting point is to investigate the intrinsic performance through the distance distribution analysis [16][17]. The results of distance distribution analysis give the information that what the Hamming distances of a codeword are to

[1] IRA codes are another class of low-encoding complexity LDPC codes, and are promising in recent standards such as IEEE 802.11n or DVB-S2

other codewords, and hence the results serve as a fundamental step to investigate performance bounds under the optimal maximum-likelihood (ML) decoder.

In this chapter, we utilize the enumeration method [12][13] to derive the average distance distributions for ensembles of regular and irregular SC-LDGM and PC-LDGM codes. The distance distributions are expressed as a function of the design parameters such as code lengths and code degrees of the two constituent codes. In addition, we apply the union bounding technique to extract the bit-error-rate (BER) performance. The relationship between the design parameters and their effects on the error-performance of the codes can be identified. We show that SC-LDGM codes are able to provide extremely low error-floors. These bounds can be used as a guideline for the design of sub-optimal message-passing decoders in the future [14][15].

It is worth to mention that the study of SC-LDGM or PC-LDGM codes are interesting not only because these codes provide a exceeding error-performance while the encoding and decoding complexity is low, but also because there exist some new applications such as joint source channel encoding system [24] and network-coding on the cooperative wireless multiple access relay network [35] in which these codes are found very useful. In addition, the systematic form of LDGM codes make the codes as a good candidate to achieve rateless encoding or uneven protection due to the flexibility of adding or removing parity check bits [18].

The rest of the chapter is organized as following. In section 3.2, the definitions and notations for LDGM codes, PC-LDGM codes, and SC-LDGM codes are given. We derive the distance distributions in ensemble of LDGM codes, SC-LDGM codes and PC-LDGM codes in section 3.3. Section 3.4 gives the numerical evaluation methods and results. We draw a conclusion in section 3.5.

3.2 LDGM CODES, PC-LDGM CODES AND SC-LDGM CODES

LDGM codes: LDGM codes are systematic linear block codes with parity check matrix **H** = [**P** ; **I**], where **P** is an $(n - k)$ by k sparse matrix and **I** is the $(n-k)$ by $(n-k)$ identity matrix. k denotes the number of input bits and n denotes the number of output bits of an LDGM encoder. An LDGM code is called *regular* if both the number of 1's in a column in the **P** matrix and that in a row stay fixed for all columns and rows. Though irregularity can provide performance improvement, regularity could lead to simplified modular hardware implementation. For regular LDGM codes, the *degree* of a variable node, denoted as d_v, which is the number of ones in a column in the **P** matrix**.** Similarly, the *degree* of a check node, d_c, represents the number of ones in a row in the **H** matrix. Based on the structure of **H** matrix, the code rate R of (d_v, d_c)*-regular* LDGM code is given by $R = 1/\left(1 + d_v/\left(d_c - 1\right)\right)$.

PC-LDGM codes: As illustrated in Figure 11(a), for a given binary information vector **u** = $[u_1,\ldots,u_k]$, the upper LDGM encoder generate the binary parity check vector **p** = $[p_1,\ldots,p_i]$ whereas the lower LDGM encoder generate the binary parity check vector **p**' = $[p'_1,\ldots,p'_j]$. The codeword of a PC-LDGM encoder is obtained from the concatenation {**u**, **p, p**'}. If the code lengths for the upper and lower LDGM encoder are n_u and n_l, respectively, we have $k + i = n_u$ and $k + j = n_l$. For ($d_v^{(u)}$, $d_c^{(u)}$) and ($d_v^{(l)}$, $d_c^{(l)}$) regular LDGM upper code and lower code, the individual code rates are $R_u = k/n_u = 1/\left(1 + d_v^{(u)}/(d_c^{(u)} - 1)\right)$ and $R_l = k/n_l = 1/\left(1 + d_v^{(l)}/(d_c^{(l)} - 1)\right)$, respectively. The overall code rate of the PC-LDGM encoder can be associated by $R_P = k/(k + i + j) = 1/\left(1 + d_v^{(u)}/(d_c^{(u)} - 1) + d_v^{(l)}/(d_c^{(l)} - 1)\right)$. The factor graph of the PC-LDGM is depicted as shown in Figure 12(a).

SC-LDGM codes: As illustrated in Figure 11(b), for a given binary information vector u = [$u_1,\ldots,u_k$], the outer LDGM encoder generate the binary parity check vector p = [$p_1,\ldots,p_i$]. The concatenation of {u, p} is then feed to the inner LDGM encoder, and the parity check vector p' = [$p'_1,\ldots,p'_j$] is obtained. The codeword of a SC-LDGM encoder is the concatenation {u, p, p'}. If the code lengths for the outer and inner LDGM encoder are n_o and n_i , respectively, we have $k+i=n_o$ and $k+i+j=n_i$. For ($d_v^{(o)}, d_c^{(o)}$) and ($d_v^{(i)}, d_c^{(i)}$) regular LDGM outer code and inner code, the individual code rates are $R_o = k/n_o = 1/\left(1+d_v^{(o)}/(d_c^{(o)}-1)\right)$ and $R_i = n_o/n_i = 1/\left(1+d_v^{(i)}/(d_c^{(i)}-1)\right)$, respectively. The overall code rate of the SC-LDGM encoder is associated by $R_S = k/(k+i+j)$, which can be shown to be $R_S = 1/\left(\left(1+d_v^{(o)}/(d_c^{(o)}-1)\right)\left(1+d_v^{(i)}/(d_c^{(i)}-1)\right)\right)$. The factor graph of the SC-LDGM is depicted as shown in Figure 12(b).

3.3 DISTANCE SPECTRUM ANALYSIS

The weight of a codeword, which is the number of 1's in the codeword, represents the Hamming distance seen from the all-zero codeword. For a linear block code, the Hamming distance distribution seen by any codeword is identical. Therefore, finding the weight distributions, i.e., distance distributions, of a code is equivalent to finding the number of codewords having the same weight. In the following, we utilize enumeration methods for the distance distribution analysis. For the ease of reading, we summarize and list the definition and notation in the Table 1 used for the following derivation.

Averaged distance distribution for LDGM codes: Figure 10 depicts the structure of LDGM codes for distance distributions analysis from enumeration methods [13]. Let W be a random variable denoting the weight of input information vector, and H be a random variable denoting the weight of output parity-check vector. Let $\overline{Z}_{w,h}$ denote the average number of codewords (over an ensemble of LDGM codes) with input weight $W = w$ and output weight $H = h$. $\overline{Z}_{w,h}$ can be associated by

$$\overline{Z}_{w,h} = \binom{k}{w} P(H = h \mid W = w)\,, \tag{3.1}$$

where $\binom{k}{w}$ represents the total possibilities for the input information vector of weight w. We note that, in the expression, only *weight* matters. The positions of 1's do not matter. That is, codewords of the same condition of weights have the same conditional probability $P(H = h \mid W = w)$. To obtain the conditional probability $P(H = h \mid W = w)$, we decompose it as

$$P(H = h \mid W = w) = \sum_{e=0}^{e=t} P(E = e \mid W = w) P(H = h \mid E = e, W = w)\,, \tag{3.2}$$

where E is a random variable denoting the total number of edges emanating from the information variable nodes of the binary message 1. The conditional probability $P(E = e \mid W = w)$ can be derived as follows. The total possibilities for $W = w$, regardless of the number of emanating edges E, is $\binom{k}{w}$. The number of ways of having exactly e edges emanating from w information variable nodes can be computed by the enumeration function $\left\lfloor \prod_{i=1}^{\infty} (1 + x^i y)^{k\tilde{\lambda}_i} \right\rfloor_{x^e y^w}$, where $\tilde{\lambda}_i$ denotes the fraction of variable nodes which are with i edges, and $\lfloor f(x, y) \rfloor_{x^a y^b}$ denotes the coefficient of the term $x^a y^b$ in $f(x, y)$. Therefore,

$$P(E=e\,|\,W=w)=\frac{\left\lfloor \prod_{\forall i}(1+x^i y)^{k\tilde{\lambda}_i}\right\rfloor_{x^e y^w}}{\binom{k}{w}}. \tag{3.3}$$

Similarly, the conditional probability $P(H=h\,|\,E=e,W=w)$ can be derived as follows. Let t denote the total number of edges between information variable nodes and parity nodes, i.e., $t=k\sum_{i=1}^{\infty} i\tilde{\lambda}_i$. The total possibilities for the t check-node "sockets" connecting to the e edges emanating from w information variable nodes is $\binom{t}{e}$. Denote the number of check nodes as L , i.e., $L=n-k$. We note that to count the number of ways of having weight h of output parity-check vector is equivalent to count the number of ways that h check nodes are of odd number of connection and the remaining $L-h$ check nodes are of even number of connection. Counting the number of ways can be associated by the enumeration function $\left\lfloor \prod_{j=1}^{\infty}[\,f_-(x,j)y+f_+(x,j)]^{L\tilde{\rho}_j}\right\rfloor_{x^e y^h}$, where $f_-(x,j)=\sum_{l\in[1,3,5,\ldots]}^{j}\binom{j}{l}x^l$ is a polynomial representing the number of ways of odd connection on a check node, $f_+(x,j)=\sum_{l\in[0,2,4,\ldots]}^{j}\binom{j}{l}x^l$ representing the number of ways of even connection on a check node, and $\tilde{\rho}_j$ denotes the fraction of check nodes that are with j edges. In the expression, the power of the xy terms keeps tracking the number of such edges. Therefore, we see

$$P(H=h\,|\,E=e,W=w)=\frac{\left\lfloor \prod_{\forall j}[\,f_-(x,j)y+f_+(x,j)]^{L\tilde{\rho}_j}\right\rfloor_{x^e y^h}}{\binom{t}{e}}. \tag{3.4}$$

We note that $f_-(x,j)$ is equal to the expression $f_-(x,j)=\frac{1}{2}\left((1+x)^j-(1-x)^j\right)$. Likewise, $f_+(x,j)=\frac{1}{2}\left((1+x)^j+(1-x)^j\right)$. In addition, the conditional probability $P(H=h\,|\,E=e,W=w)$ is equal to

$P(H=h\,|\,E=e)$, which does not relate to the weight of the input information vector. Combining (3.1),(3.2), (3.3) and (3.4), we obtain

$$\bar{Z}_{w,h}=\sum_{e=0}^{e=t}\frac{\left\lfloor \prod_{\forall i}(1+x^{i}y)^{k\tilde{\lambda}_i}\right\rfloor_{x^{e}y^{w}}\left\lfloor \prod_{\forall j}\left[\,f_{-}(x,j)y+f_{+}(x,j)\right]^{L\tilde{\rho}_j}\right\rfloor_{x^{e}y^{h}}}{\binom{t}{e}}\;. \tag{3.5}$$

For an ensemble of (d_v,d_c) regular LDGM codes. we have followings: (i) $\tilde{\lambda}_i=1$ only when $i=d_v$; $\tilde{\lambda}_i=0$ elsewhere, (ii) $\tilde{\rho}_j=1$ only when $j=d_c$; $\tilde{\rho}_j=0$ elsewhere, and (iii) e is necessarily equal to wd_v. We have

$$P(H=h\,|\,W=w)=P(E=wd_v\,|\,W=w)P(H=h\,|\,E=wd_v,W=w)\,, \tag{3.6}$$

where

$$P(E=wd_v\,|\,W=w)=\frac{\left\lfloor(1+x^{d_v}y)^{k}\right\rfloor_{x^{wd_v}y^{w}}}{\binom{k}{w}}=\frac{\binom{k}{w}}{\binom{k}{w}}=1 \tag{3.7}$$

and

$$P(H=h\,|\,E=wd_v,W=w)=\frac{\left\lfloor\left[\,f_{-}(x,d_c)y+f_{+}(x,d_c)\right]^{L}\right\rfloor_{x^{wd_v}y^{h}}}{\binom{t}{wd_v}}=\frac{\binom{L}{h}\left\lfloor f_{-}(x,d_c)^{h}f_{+}(x,d_c)^{L-h}\right\rfloor_{x^{wd_v}}}{\binom{kd_v}{wd_v}} \tag{3.8}$$

Therefore, (3.5) can be expressed as

$$\bar{Z}^{(reg)}_{w,h}=\frac{\binom{k}{w}\binom{L}{h}\left\lfloor f_{-}(x,d_c)^{h}f_{+}(x,d_c)^{L-h}\right\rfloor_{x^{wd_v}}}{\binom{kd_v}{wd_v}} \tag{3.9}$$

Denote $\bar{A}_l$ to be the averaged (over an ensemble) number of codewords of weight l. Then the averaged distance distributions are obtained by

$$\bar{A}_l = \sum_{w=\max(0,l-L)}^{\min(k,l)} \bar{Z}_{w,l-w} \; or \; \bar{A}_l^{(reg)} = \sum_{w=\max(0,l-L)}^{\min(k,l)} \bar{Z}_{w,l-w}^{(reg)} \tag{3.10}$$

<u>Averaged distance distribution for PC-LDGM codes:</u> Consider the PC-LDGM codes as the parallel concatenation of {**u**, **p**, **p'**}, where **p** and **p'** are obtained from two independent LDGM constitute codes. The averaged number of codewords over an ensemble of PC-LDGM codes with input weight $W = w$ and output weight $H^{(upper)} = h_1$, $H^{(lower)} = h_2$ is associated by

$$\begin{aligned}
\bar{Z}_{w,h^{(u)},h^{(l)}}^{(PC-SLDGM)} &= \binom{k}{w} P(H^{(upper)} = h^{(u)} \mid W = w) P(H^{(lower)} = h^{(l)} \mid W = w) \\
&= \frac{\binom{k}{w} P(H^{(upper)} = h^{(u)} \mid W = w)}{\binom{k}{w}} \binom{k}{w} P(H^{(lower)} = h^{(l)} \mid W = w), \\
&= \frac{\bar{Z}_{w,h^{(u)}}^{(u)}}{\binom{k}{w}} \bar{Z}_{w,h^{(l)}}^{(l)}
\end{aligned} \tag{3.11}$$

where

$$P(H^{(upper)} = h^{(u)} \mid W = w) = \sum_{e=0}^{e=t^{(u)}} \frac{\left\lfloor \prod_{\forall i} (1 + x^i y)^{k^{(u)} \tilde{\lambda}_i^{(u)}} \right\rfloor_{x^e y^w} \left\lfloor \prod_{\forall j} \left[f_-(x,j) y + f_+(x,j) \right]^{L^{(u)} \tilde{\rho}_j^{(u)}} \right\rfloor_{x^e y^{h^{(u)}}}}{\binom{k^{(u)}}{w} \binom{t^{(u)}}{e}} \tag{3.12}$$

and

$$P(H^{(lower)} = h^{(l)} \mid W = w) = \sum_{e=0}^{e=t^{(l)}} \frac{\left\lfloor \prod_{\forall i} (1 + x^i y)^{k^{(l)} \tilde{\lambda}_i^{(l)}} \right\rfloor_{x^e y^w} \left\lfloor \prod_{\forall j} \left[f_-(x,j) y + f_+(x,j) \right]^{L^{(l)} \tilde{\rho}_j^{(l)}} \right\rfloor_{x^e y^{h^{(l)}}}}{\binom{k^{(l)}}{w} \binom{t^{(l)}}{e}} . \tag{3.13}$$

The superscript "(u)" indicates the parameters of upper code, whereas the superscript "(l)" indicates the parameters of lower code. The above conditional probabilities (3.12), (3.13) are obtained by following the similar derivation as (3.2). We note that $k^{(u)} = k^{(l)} = k$. Without loss of generosity, we assume $L^{(l)} \le L^{(u)}$. The averaged number of codewords of weight l for the ensemble of PC-LDGM codes is therefore given by

$$\begin{aligned}\overline{A}_l^{(PC-SLDGM)} &= \sum_{w=\max(0,l-L^{(u)})}^{\min(k,l)} \sum_{\theta=0}^{\min(l-w,L^{(l)})} \overline{Z}_{w,l-w-\theta,\theta}^{(PC-SLDGM)} \\ &= \sum_{w=\max(0,l-L^{(u)})}^{\min(k,l)} \sum_{\theta=0}^{\min(l-w,L^{(l)})} \binom{k}{w} P(H^{(upper)} = l-w-\theta \,|\, W=w) P(H^{(lower)} = \theta \,|\, W=w)\end{aligned} \tag{3.14}$$

We note that

$$\sum_{h^{(u)}} \overline{Z}_{w,h^{(u)},h^{(l)}}^{(PC-SLDGM)} = \overline{Z}_{w,h^{(l)}}^{(lower-SLDGM)} \text{ and } \sum_{h^{(l)}} \overline{Z}_{w,h^{(u)},h^{(l)}}^{(PC-SLDGM)} = \overline{Z}_{w,h^{(u)}}^{(upper-SLDGM)}, \tag{3.15}$$

where $h^{(u)} \in \{1, 2, ...H, \text{and the emptry } \phi\}$.

For regular PC-LDGM codes with $(d_v^{(u)}, d_c^{(u)})$ and $(d_v^{(l)}, d_c^{(l)})$, the distance spectrum $\overline{A}_l^{(PC-SLDGM)}$ can be expressed as

$$\overline{A}_l^{(PC-SLDGM)} = \sum_{w=\max(0,l-L^{(u)})}^{\min(k,l)} \sum_{\theta=0}^{\min(l-w,L^{(l)})} \left(\binom{k}{w} \frac{\binom{L^{(u)}}{l-w-\theta} \left\lfloor f_-(x,d_c^{(u)})^{l-w-\theta} f_+(x,d_c^{(u)})^{L^{(u)}-l+w+\theta} \right\rfloor_{x^{wd_v^{(u)}}}}{\binom{kd_v^{(u)}}{wd_v^{(u)}}} \bullet \frac{\binom{L^{(l)}}{\theta} \left\lfloor f_-(x,d_c^{(l)})^{\theta} f_+(x,d_c^{(l)})^{L^{(l)}+\theta} \right\rfloor_{x^{wd_v^{(l)}}}}{\binom{kd_v^{(l)}}{wd_v^{(l)}}} \right). \tag{3.16}$$

<u>Averaged distance distribution for SC-LDGM code:</u> Consider the SC-LDGM codes as the serial concatenation of two independent LDGM codes. The averaged number of codewords of weight l for the ensemble of SC-LDGM codes can be obtained through the following steps.

First, the averaged number of codewords of weight w_2 obtained by the ensemble of outer codes with input weight w_1 is given by

$$\bar{A}_{w_2}^{(o)} = \sum_{w_1=\max(0,w_2-L^{(o)})}^{\min(k^{(o)},w_2)} \bar{Z}_{w_1,w_2-w_1}^{(o)} , \tag{3.17}$$

where

$$\bar{Z}_{w_1,w_2-w_1}^{(o)} = \binom{k^{(o)}}{w_1} P(H^{(outer)} = w_2 - w_1 \,|\, W^{(outter)} = w_1) , \tag{3.18}$$

and

$$P(H^{(outer)} = w_2 - w_1 \,|\, W^{(outter)} = w_1) = \sum_{e=0}^{e=t^{(o)}} \frac{\left\lfloor \prod_{\forall i} (1+x^i y)^{k^{(o)} \tilde{\lambda}_i^{(o)}} \right\rfloor_{x^e y^{w_1}} \left\lfloor \prod_{\forall j} [\, f_{-}(x,j)y + f_{+}(x,j)]^{L^{(o)} \tilde{\rho}_j^{(o)}} \right\rfloor_{x^e y^{w_2-w_1}}}{\binom{k^{(o)}}{w_1}\binom{t^{(o)}}{e}} . \tag{3.19}$$

Since an ensemble of codes is considered, a “random interleaver” can be seen as virtually existing between the inner codes and the outer codes for deriving the distance spectrum calculation. The averaged number of codewords over an ensemble of the inner codes with input weight $W^{(inner)} = w_2$ and output weight $H^{(inner)} = h$ is associated by

$$\bar{Z}_{w_2,h}^{(inner)} = \bar{A}_{w_2}^{(o)} P(H^{(inner)} = h \,|\, W^{(inner)} = w_2) , \tag{3.20}$$

where

$$P(H^{(inner)} = h \,|\, W^{(inner)} = w_2) = \sum_{e=0}^{e=t^{(i)}} \frac{\left\lfloor \prod_{\forall i} (1+x^i y)^{k^{(i)} \tilde{\lambda}_i^{(i)}} \right\rfloor_{x^e y^{w_2}} \left\lfloor \prod_{\forall j} [\, f_{-}(x,j)y + f_{+}(x,j)]^{L^{(i)} \tilde{\rho}_j^{(i)}} \right\rfloor_{x^e y^{h}}}{\binom{k^{(i)}}{w_2}\binom{t^{(i)}}{e}} . \tag{3.21}$$

The averaged number of codewords of weight l for the ensemble of SC-LDGM codes is therefore associated by

$$\begin{aligned}
\overline{A}_l^{(SC-SLDGM)} &= \sum_{w_2=\max(0,l-L^{(i)})}^{\min(k^{(i)},l)} \overline{Z}_{w_2,l-w_2}^{(SC-SLDGM)} \\
&= \sum_{w_2=\max(0,l-L^{(i)})}^{\min(k^{(i)},l)} \overline{A}_{w_2}^{(o)} P(H^{(inner)} = l-w_2 \mid W^{(inner)} = w_2) \\
&= \sum_{w_2=\max(0,l-L^{(i)})}^{\min(k^{(i)},l)} \sum_{w_1=\max(0,w_2-L^{(o)})}^{\min(k^{(o)},w_2)} \overline{Z}_{w_1,w_2-w_1}^{(o)} P(H^{(inner)} = l-w_2 \mid W^{(inner)} = w_2) \\
&= \sum_{w_2=\max(0,l-L^{(i)})}^{\min(k^{(i)},l)} \sum_{w_1=\max(0,w_2-L^{(o)})}^{\min(k^{(o)},w_2)} \frac{\overline{Z}_{w_1,w_2-w_1}^{(o)}}{\binom{k^{(i)}}{w_2}} \binom{k^{(i)}}{w_2} P(H^{(inner)} = l-w_2 \mid W^{(inner)} = w_2) \\
&= \sum_{w_2=\max(0,l-L^{(i)})}^{\min(k^{(i)},l)} \sum_{w_1=\max(0,w_2-L^{(o)})}^{\min(k^{(o)},w_2)} \frac{\overline{Z}_{w_1,w_2-w_1}^{(o)}}{\binom{k^{(i)}}{w_2}} \overline{Z}_{w_2,l-w_2}^{(i)}
\end{aligned} \tag{3.22}$$

For regular SC-LDGM codes with $(d_v^{(o)}, d_c^{(o)})$ and $(d_v^{(i)}, d_c^{(i)})$, the distance spectrum $\overline{A}_l^{(SC-SLDGM)}$ can be expressed as

$$\begin{aligned}
\overline{A}_l^{(SC-SLDGM)} = \sum_{w_2=\max(0,l-L^{(i)})}^{\min(k^{(i)},l)} \sum_{w_1=\max(0,w_2-L^{(o)})}^{\min(k^{(o)},w_2)} \Bigg(& \binom{k^{(o)}}{w_1} \frac{\binom{L^{(o)}}{w_2-w_1} \left\lfloor f_-(x,d_c^{(o)})^{w_2-w_1} f_+(x,d_c^{(o)})^{L^{(o)}-w_2+w_1} \right\rfloor_{x^{w_1 d_v^{(o)}}}}{\binom{k^{(o)}d_v^{(o)}}{w_1 d_v^{(o)}}} \\
& \bullet \frac{\binom{L^{(i)}}{l-w_2} \left\lfloor f_-(x,d_c^{(i)})^{l-w_2} f_+(x,d_c^{(i)})^{L^{(i)}-l+w_2} \right\rfloor_{x^{w_2 d_v^{(i)}}}}{\binom{k^{(i)}d_v^{(i)}}{w_2 d_v^{(i)}}} \Bigg).
\end{aligned} \tag{3.23}$$

We note that, in general, PC-LDGM codes and SC-LDGM codes can be seen as irregular LDGM codes. These concatenation interpretations simplify the design and may facilitate the implementation of iterative decoder. While the performance of SC-LDGM codes is better, the PC-LDGM codes would have the advantage over the encoding and decoding complexities and delays.

Averaged distance spectrum for the layer ensemble LDGM codes: In the following, we provide the distance spectrum results based on the layer ensemble. This construction of ensemble is inspired by Gallager's ensemble of LDPC codes. For this ensemble, we observe the property of minimum distance that $d_{\min} = d_v + 1$.

Consider the following ensemble inspired by Gallager's ensemble for LDPC codes. Consider a parity check matrix composed from two parts: LHS parts and RHS parts. The RHS part is an identity matrix of size $(n-k)\times(n-k)$, where $k = nd_c/(d_v + d_c)$. The LHS part consists of k/d_c submatrices (or strips) (each strip is of size $k/d_c \times k$). The first strip is the d_c -fold concatenation of the building blocks of size $k/d_c \times d_c$ in which the i^{th} building block contains d_c 1's in its i^{th} row and 0's elsewhere. Hence, the LHS region of the first strip contains all its 1's in a descending order. The other strips are obtained by permuting at random the columns of the first strip, denoted by a series of independent random operators π_i, $i = 1,2,\cdots,d_v - 1$. An example of $n{=}18, d_v{=}3,$and $d_c{=}3$ ensemble are shown in Figure 13(a). We note the the RHS identity matrix can be virtually divided into d_v sub-identity matrix, each of which is of size $k/d_c \times k/d_c$. In the analysis, we see that each strip corresponds to a sub-identity, and, therefore, treat the ensemble of LDGM codes as parallel concatenation of d_v (1, d_c)-regular LDGM codes. The factor graph is depiected in Figure 13(b). We note that, in general, there could exist parallel links between check nodes and variable in graph-based ensemble. However, for this specific ensemble, since d_v for each strip, there is no links existing between check nodes and variable nodes.

Seen as parallel concatenation of $d_v\,(1, d_c)$ regular LDGM codes, individually, we have

$$Z_{w,h_i} = \binom{k}{w} P(H = h_i \mid W = w), \text{ for } i = 1,2,...,d_v, \text{ and } h_i \in [1, k/d_c]. \qquad (3.24)$$

From the notion of PC-concatenation, we see

$$Z_{w,h_1,h_2,...,h_{d_v}} = \binom{k}{w} P(H = h_1 \mid W = w) P(H = h_2 \mid W = w) ... P(H = h_{d_v} \mid W = w) . \qquad (3.25)$$

For regular LDGM codes, we have

$$P(H = h_i \mid W = w) = \frac{\left\lfloor \left[f_-(x,d_c)y + f_+(x,d_c) \right]^L \right\rfloor_{x^{wd_v} y^{h_i}}}{\binom{t}{wd_v}} = \frac{\binom{L}{h} \left\lfloor f_-(x,d_c)^{h_i} \, f_+(x,d_c)^{L-h_i} \right\rfloor_{x^{wd_v}}}{\binom{kd_v}{wd_v}} . \qquad (3.26)$$

Since ${d^{(ind)}}_v = 1$, (3.26) reduces to

$$P(H = h_i \mid W = w) = \frac{\binom{L}{h_i} \left\lfloor f_-(x,d_c)^{h_i} \, f_+(x,d_c)^{L-h_i} \right\rfloor_{x^w}}{\binom{k}{w}} , \qquad (3.27)$$

for each individual code. We note that (3.27) can be represented in the form of Z_{w,h_i}, i.e.,

$$Z_{w,h_1,h_2,...,h_{d_v}} = \frac{Z_{w,h_1} Z_{w,h_2} ... Z_{w,h_{d_v}}}{\binom{k}{w}^{d_v - 1}} . \qquad (3.28)$$

Here we note that since the sub-code of the first strip is fixed and determined, it may not be clear that $P(H = h_1 \mid W = w)$ is still in the form of (3.4). The explanation of validating (3.4) is as follows. The total possibilities of input of weight w is $\binom{k}{w}$. The number of input messages of weight w resulting output weight of h_1 (through the first strip) can be obtained through the enumeration polynomial $\left\lfloor \left[f_-(x,d_c)y + f_+(x,d_c) \right]^L \right\rfloor_{x^w y^{h_1}}$. Therefore, the expression for $P(H = h_1 \mid W = w)$ of the first strip is exactly the same as (3.27).

To facilitate the numerical evaluation, if we are only interested in finding Z_{w,h_o}, where h_0 is the overall output weight of the whole code, the expression can be calculate by layering the calculation into group of 2's power. That is, we calculate

$$Z^{(c_2)}_{w,h_{c_2}} = \sum_{h_{c_1}=\max(0,h_{c_2}-L^{(c_1)})}^{\min(L^{(c_1)},h_{c_2})} \frac{Z_{w,h_{c_1}} Z_{w,h_{c_2}-h_{c_1}}}{\binom{k}{w}}, \tag{3.29}$$

where $L^{(c_1)} = k/d_c$, representing parallel concatenation of two codes. Next, we obtain

$$Z^{(c_4)}_{w,h_{c_4}} = \sum_{h_{c_2}=\max(0,h_{c_4}-L^{(c_2)})}^{\min(L^{(c_2)},h_{c_4})} \frac{Z^{(c_2)}{}_{w,h_{c_2}} Z^{(c_2)}{}_{w,h_{c_4}-h_{c_2}}}{\binom{k}{w}}, \tag{3.30}$$

where $L^{(c_2)} = 2\times k/d_c$, representing parallel concatenation of four codes. Next, we can compute

$$Z^{(c_8)}_{w,h_{c_8}} = \sum_{h_{c_4}=\max(0,h_{c_8}-L^{(c_4)})}^{\min(L^{(c_4)},h_{c_8})} \frac{Z^{(c_4)}{}_{w,h_{c_4}} Z^{(c_2)}{}_{w,h_{c_8}-h_{c_4}}}{\binom{k}{w}}, \tag{3.31}$$

where $L^{(c_4)} = 4\times k/d_c$, and so on. Finally, for example, if $d_v = 9$ for the whole code, we have

$$Z^{(c_9)}_{w,h_{c_9}} = \sum_{h_{c_8}=\max(0,h_{c_9}-L^{(c1)})}^{\min(L^{(c_8)},h_{c_9})} \frac{Z^{(c_8)}{}_{w,h_{c_8}} Z_{w,h_{c_9}-h_{c_8}}}{\binom{k}{w}}, \tag{3.32}$$

where $L^{(c_8)} = 8\times k/d_c$.

3.4 NUMERICAL EVALUATION METHODS AND RESULTS

We now first introduce the log-evaluation methods for numerical analysis, union bounding techniques for BER extraction, and then provide and discuss the obtained numerical results.

The log-evaluation method: To carry on these numerical expressions with large power (or multiplication) of polynomials of large coefficients, we provide the log-evaluation methods, developed in [20], in the following. The key idea is to express the coefficients in the exp-log form.

Suppose we have two polynomials $f(x)$ and $g(x)$. We represent the coefficients in exp-log form as

$$\begin{aligned} f(x) &= a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n = \sum_{i=0}^{n} a_i x^i = \sum_{i=0}^{n} e^{\ln a_i} x^i \\ g(x) &= b_0 + b_1 x + b_2 x^2 + \dots + b_m x^m = \sum_{j=0}^{m} b_j x^j = \sum_{j=0}^{m} e^{\ln b_j} x^j \end{aligned} . \tag{3.33}$$

The coefficients of the polynomial of $f(x)g(x)$ (from x^0 to x^{n+m}) can be expressed as $coef[f(x)g(x)] = conv([e^{\ln a_i},\ i = 1,2,...,n],[e^{\ln b_j},\ j = 1,2,...,m])$, where $[e^{\ln a_i}]$ indicates the series of the coefficients, i.e., $[e^{\ln a_i},\ i = 1,2,...n] \triangleq [e^{\ln a_0}, e^{\ln a_1},..., e^{\ln a_n}]$. Similarly, $[e^{\ln b_i},\ j = 1,2,...,m] \triangleq [e^{\ln b_0}, e^{\ln b_1},..., e^{\ln b_n}]$. We see that

$$\begin{aligned} coef[f(x)g(x)] &= conv([e^{\ln a_i}],[e^{\ln b_j}]) \\ &= [e^{\ln a_0 + \ln b_0}, e^{\ln a_0 + \ln b_1} + e^{\ln a_1 + \ln b_0}, e^{\ln a_0 + \ln b_2} + e^{\ln a_1 + \ln b_1} + e^{\ln a_2 + \ln b_0}, ...], \\ &\triangleq [e^{c_0^{(0)}}, e^{c_0^{(1)}} + e^{c_1^{(1)}}, e^{c_0^{(2)}} + e^{c_1^{(2)}} + e^{c_2^{(2)}}, ...], \end{aligned} \tag{3.34}$$

where $c_k^{(l)} = (\ln a_k + \ln b_{l-k})$. We note that l is the indices for the l^{th} coefficient, $l = 0,1,2,...,n+m$. According to the convolution rule, k is a positive integer running from $\max(0, l-m)$ to $\min(l, n)$.

Now, if we let $\mu^{(l)} = \max(c_k^{(l)}, k = \max(0, l-m) : \min(l, m))$, we see that the coefficients of the polynomial $f(x)g(x)$ in log form can be expressed as

$$\ln(coef[f(x)g(x)]) = [c_\mu^{(l)} + \ln(1 + \sum_{\substack{k=\max(0,l-m),\\ k\neq\mu}}^{k=\min(l,n)} e^{-(c_\mu^{(l)} - c_k^{(l)})}),\ l = 0,1,...,n+m]. \tag{3.35}$$

Since $0 < e^{-(c_\mu^{(l)} - c_k^{(l)})} \le 1$, the implementation in the exp-log form avoid the over flow numerical problem in programming.

Recall that $c_k^{(l)} = (\ln a_k + \ln b_{l-k})$, we note that the output coefficients in log-form is a function of the input coefficients in log-form. Since both of the input and output are in log-form, we can use this method recursively when there are more than two polynomials in multiplication.

Union Bound techniques for BER performance extraction: Union bound is known to be tight at high SNR. Let $d_{\min}$ and $d_{\max}$ designate, respectively, the minimal and the maximal Hamming weight (*d*) so that $S_d \neq 0$. The union upper bound on the ML decoding block error probability of linear block codes whose transmission takes place over an AWGN channel is give by [17]

$$P_e \le \sum_{d=d_{\min}}^{d_{\max}} S_d Q\left(\sqrt{\frac{2dE_s}{N_0}}\right), \tag{3.36}$$

To compute an upper bound on the bit error probability, the distance spectrum S_d is replaced by $\sum_{w=1}^{nR}\left\{\left(\frac{w}{nR}\right)A_{w,d}\right\}$ where $R \triangleq \frac{k}{n}$ designates the rate and $A_{w,d}$ designates the number of codewords whose Hamming weight is equal to d, and which are encoded by information bits of Hamming weight w (i.e., $A_{w,d}$ designates the input-output weight enumerator of the code $\mathcal{C}$).

For regular LDGM codes, we have $A_{w,d} = Z_{w,d-w}$. For a given d, w is ranging from $\max(0, d-L)$ to $\min(k,d)$. For the calculation of substitution of S_d, we have $w = \max(1, d-L):\min(k,d)$. For PC-LDGM codes, we have

$$A_{w,d} = \sum_{\theta=0}^{\min(d-w, L^{(l)})} \frac{Z^{(u)}_{w,d-w-\theta}}{\binom{k}{w}} Z^{(l)}_{w,\theta}, \tag{3.37}$$

where w is ranging from $\max(0, d-L^{(u)})$: $\min(k,d)$. Therefore, For the calculation of the substitution of S_d, we have $w = \max(1, d-L^{(u)}):\min(k,d)$. The notation $L^{(l)}$ or $L^{(u)}$ represent the number of parity bits in the lower or upper encoder. For SC-LDGM codes, we have

$$A_{w,d} = \sum_{w_2=\max(0, d-L^{(i)}, w)}^{\min(k^{(i)}, d, w+L^{(o)})} \frac{Z^{(o)}_{w,w_2-w}}{\binom{k^{(i)}}{w_2}} Z^{(i)}_{w_2, d-w_2}, \tag{3.38}$$

where w is ranging from $0:k^{(o)}$. For the calculation of substitution of S_d, we have $w = 1:k^{(o)}$.

<u>*Numerical results for LDGM codes and concatenated LDGM codes:*</u> Figure 14 shows the numerical results of the average distance distributions for (a) $n = 1100,\ d_v = 9,\ d_c = 9$ graph ensemble LDGM codes, (b) $n = 1100,\ d_v = 7,\ d_c = 7$ graph ensemble LDGM codes, (c) $n = 1080,\ d_v = 9,\ d_c = 9$ layered ensemble LDGM codes (i.e., number of block = 60, 9 strips, n=600,L=60 for each strip), and (d) $n = 1120,\ d_v = 7,\ d_c = 7$ layered ensemble LDGM codes (i.e., number of block = 80, 7strips, n=640,L=80 for each strip). The parameters (or the code length) of layered ensemble are chosen such that the resulting code length is most close to 1100. The upper subfigure shows the whole distance spectrum, while the lower subfigure shows the details for the number of codewords with weights up to 40. We note that codes with high degree have better distance property in the sense that the number of codeword with low weight is suppressed

more. Similarly, layered ensemble LDGM codes have better distance property than graph ensemble LDGM codes. For layered ensembles, we note that the minimum distance is $10 = d_v + 1$ and $8 = d_v + 1$, respectively. Besides, $A_{l=10} = 540.0124$ and $A_{l=8} = 560.0704$, respectively. It implies that the number of codewords with minimum distance directly comes from the 540 rows (and the 560) rows of G matrix (with the input weight 1). The minimum distance property of the layered ensemble is better than the graph-based ensemble, and thus the distance spectrum derived based on the graph ensemble could be used as a bottom line for performance evaluation.

Figure 15 shows the BER performance among the graph ensemble LDGM codes and the layered ensemble LDGM codes. We As suggested by the distance distributions, we see that the error floors are decreased when the degrees of the codes increase. Also, the layered ensemble LDGM codes have slightly lower error floors than the graph ensemble LDGM codes. .

Figure 16 shows the distance distributions of PC-LDGM codes for (a) upper code $n = 1000,\ d_v = 9,\ d_c = 9$ and lower code $n = 550,\ d_v = 5,\ d_c = 50$, (b) upper code $n = 1000,\ d_v = 9,\ d_c = 9$ and lower code $n = 550,\ d_v = 9,\ d_c = 90$, and (c) upper code $n = 1000,\ d_v = 9,\ d_c = 9$ and lower code $n = 550,\ d_v = 15,\ d_c = 150$, and (d) upper code $n = 1000,\ d_v = 7,\ d_c = 7$ and lower code $n = 550,\ d_v = 5,\ d_c = 50$, and (e) upper code $n = 1000,\ d_v = 7,\ d_c = 7$ and lower code $n = 550,\ d_v = 9,\ d_c = 90$, and (f) upper code $n = 1000,\ d_v = 7,\ d_c = 7$ and lower code $n = 550,\ d_v = 15,\ d_c = 150$. The settings are suggested by [24], as for low error floor while maintaining low threshold. In these bottom subfigures, we see that there is no codeword of weight one. Also, two layers appears in the distance distribution – one for the codewords of even weight and the other for the codewords of odd weight. Compared among (a) to (f), we see that when degree increase in either upper or lower code, the number of codewords with low weights are suppressed.

The BER performance based on union bound analysis for these PC-LDGM codes are shown in Figure 17. As suggested by the distance distributions, codes with higher degrees have lower error-floors. Increasing the degree of upper codes from (7,7) to (9,9) provides (lower code stay at (5,50)) has the similar effect as increasing the degree of lower codes form (5,50) to (9,90) (upper codes stay at (7,7)). It suggests that increasing the degree of the upper codes is more efficient to lower the error floor than increasing the degree of the lower codes.

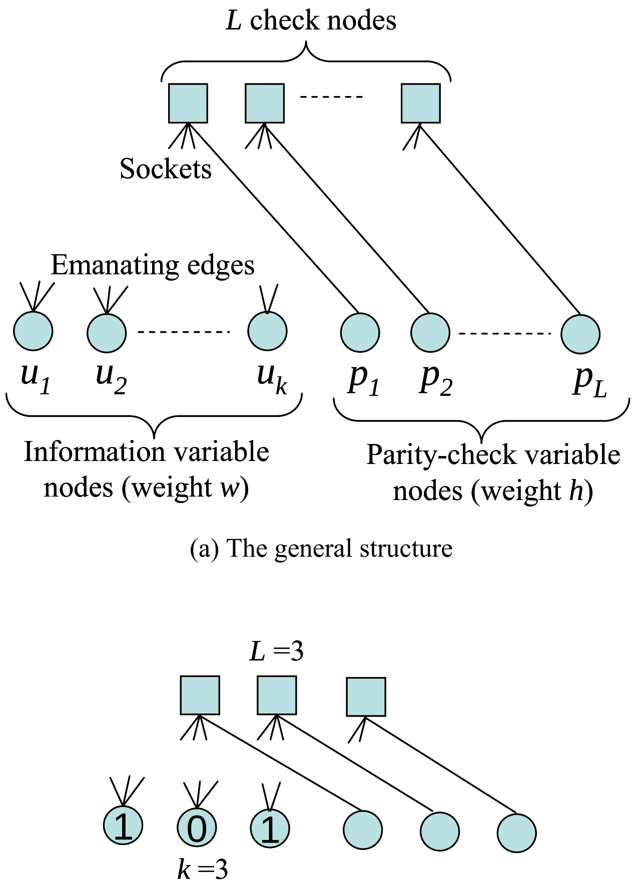
Figure 18 shows the distance distributions of SC-LDGM codes while Figure 19 shows the BER performance based on union bound for (a) outer code $n=1100,\ d_v=9,\ d_c=9$ and inner code $n=550,\ d_v=5,\ d_c=50$, (b) outer code $n=1100,\ d_v=9,\ d_c=9$ and inner code $n=550,\ d_v=9,\ d_c=90$, and (c) outer code $n=1100,\ d_v=9,\ d_c=9$ and inner code $n=550,\ d_v=15,\ d_c=150$, and (d) outer code $n=1100,\ d_v=7,\ d_c=7$ and inner code $n=550,\ d_v=5,\ d_c=50$, and (e) outer code $n=1100,\ d_v=7,\ d_c=7$ and inner code $n=550,\ d_v=9,\ d_c=90$, and (f) outer code $n=1100,\ d_v=7,\ d_c=7$ and inner code $n=550,\ d_v=15,\ d_c=150$. The settings are suggested by [11], as for low error floors while maintaining low thresholds. In these bottom subfigures, we see that there is no codeword of odd weights. In the bottom subfigures, we see tooth-like shapes appear in the distance distributions. Compared among (a) to (f), we see that when degree increase in either outer or inner code, the number of codewords with low weights are suppressed. Compared to PC-LDGM codes, SC-LDGM codes with similar settings provide significant suppression of codewords with low weights. In the distance distributions, PC-LDGM codes provide a deeper starting (refer to the number of codewords of weight two) but a big slope, whereas the SC-LDGM codes provide a shallow starting but a relatively flat slope. This property helps SC-LDGM codes to achieve extremely low error floors, as shown in Figure 19. We see that, with similarly settings, SC-LDGM outperforms PC-LDGM codes significantly.

3.5 CONCLUSION

We provide self-contained methods to numerically calculate the distance distributions of ensembles of graph LDGM codes, layered LDGM codes, parallel-concatenated LDGM codes, and serially-concatenated LDGM codes. The distance distributions reveals the internal properties of these codes and provide valuable information such as (a) the minimum distance of layered ensemble LDGM ocdes is $d_v + 1$ (b) serial concatenation results in a tooth-like distance distribution with relatively flat slope for low weight codewords, whereas parallel concatenation results in a two-layered-like distance distribution with relatively high slope for low weight codeword. Based on these distance distributions, the achievable error-floors under ML decoders for ensembles of graph LDGM codes, layered LDGM codes, PC-LDGM codes, and SC-LDGM are illustrated through union bounding analysis. While both PC-LDGM codes and SC-LDGM codes can bring down the error floors effectively, SC-LDGM codes in particular deliver extremely low error-floors.

Table 1. Definition and notation for distance spectrum calculation

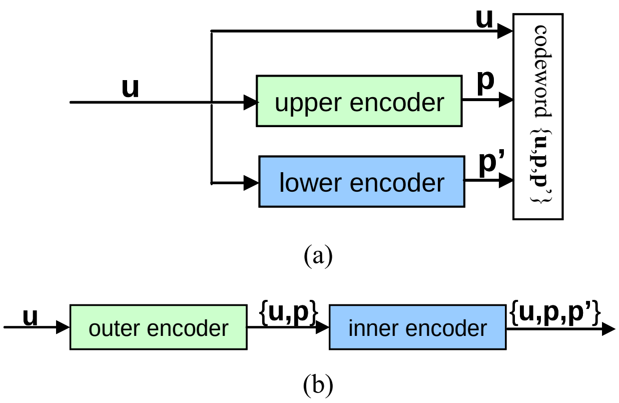
$\tilde{\lambda}_i$	The fraction of variable nodes which are with i edges. These variable nodes will be treated emanating i edges in the enumeration method.
$\tilde{\rho}_i$	The fraction of check nodes that are with j edges. These check nodes will be treated as having j sockets in the enumeration method.
H	A random variable denoting the weight of output parity-check vector.
W	A random variable denoting the weight of input information vector.
$Z_{w,h}$	The number of codewords (of a LDGM code) with input weight $W = w$ and output weight $H = h$
$\bar{Z}_{w,h}$	The average number of codewords (over an ensemble of LDGM codes) with input weight $W = w$ and output weight $H = h$. For a fixed w, $\bar{Z}_{w,h}$ gives the averaged (over the ensemble) weight-distribution of the output parity-check vector.
$\bar{A}_l$	Denote the averaged (over an ensemble) number of codewords of weight l .
E	A random variable denoting the total number of edges emanating from the information variable nodes of the binary message 1.
t	$t \triangleq k\sum_{i=1}^{\infty} i\tilde{\lambda}_i$, which is the total number of edges between information variable nodes and parity nodes.
L	$L = n - k$, denoting the number of check nodes. Recall that n is the code length and k is the length of input information vector.
$\lfloor f(x,y) \rfloor_{x^a y^b}$	For a polynomial $f(x,y)$, we denote by $\lfloor f(x,y) \rfloor_{x^a y^b}$, the coefficient of the term $x^a y^b$ in $f(x,y)$, i.e., $f(x,y) = \sum_a \sum_b \lfloor f(x,y) \rfloor_{x^a y^b} x^a y^b$.



(a) The general structure

(b) An example of definition of parameters: $w = 2,\ t = 8,\ e = 5,\ \tilde{\lambda}_2 = 1/3,\ \tilde{\lambda}_3 = 2/3,\ \tilde{\rho}_2 = 1/3,$ and $\tilde{\rho}_3 = 2/3.$

Figure 10. Structure of LDGM codes for distance distributions analysis.

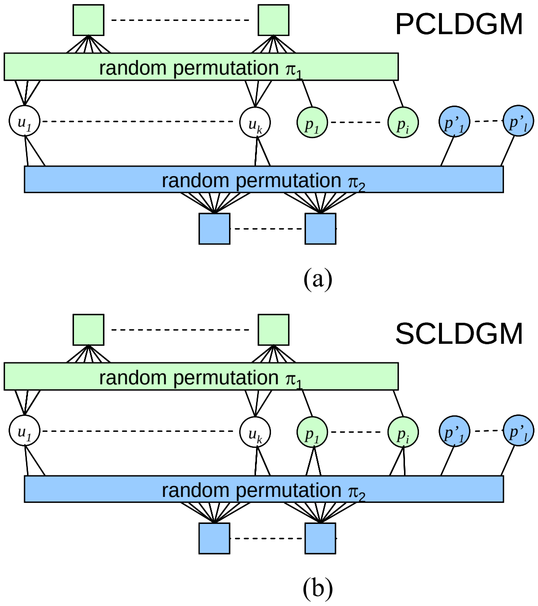


(a)

(b)

Figure 11. Block diagrams of parallel and serially concatenated LDGM codes.

We show (a) parallel-concatenated LDGM codes and (b) serially-concatenated LDGM codes. Information bits are denoted by u = [$u_1,\ldots,u_k$]. Parity bits are denoted by p = [$p_1,\ldots,p_i$] and p' = [$p'_1,\ldots,p'_j$], respectively, for the two constituent codes.

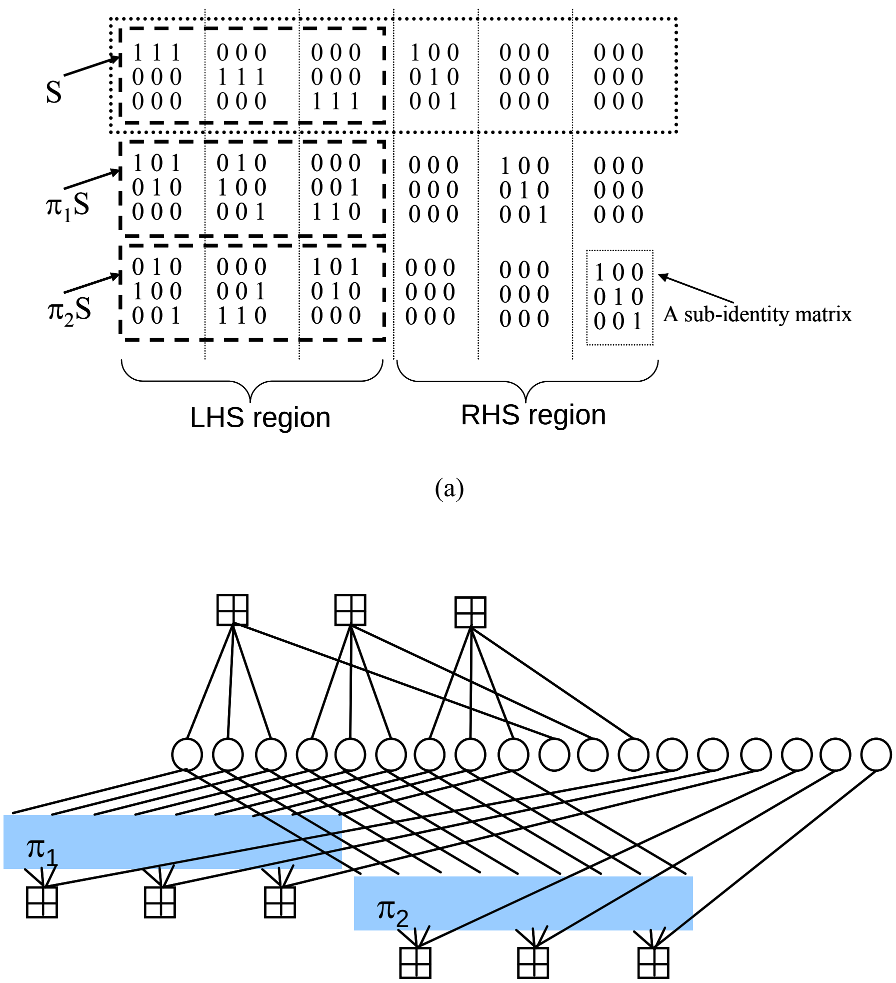


(a)

(b)

Figure 12. Factor graphs of parallel and serially concatenated LDGM codes.

(a) parallel-concatenated LDGM codes and (b) serially-concatenated LDGM codes. Squares denote the check nodes. Cycles denote the variable nodes. Information bits are denoted by white cycles, $u_1,\ldots,u_k$. Parity bits are denoted by green cycles, $p_1,\ldots,p_i$, and blue cycles, $p'_1,\ldots,p'_j$, respectively, for the two constituent codes.



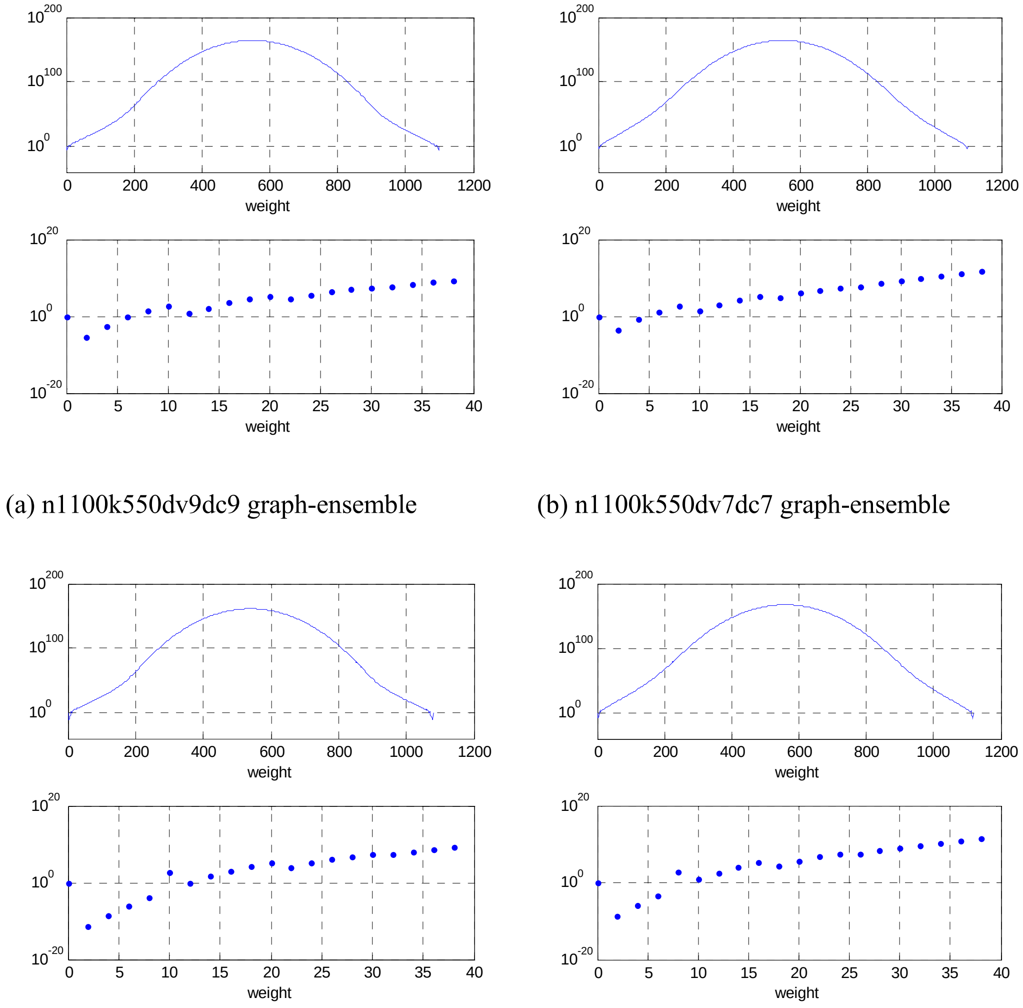
(a)

(b)

Figure 13. An example of the layered ensemble of LDGM codes.

In the figure, symbol π represents a random permutation. Subfigure (a) represents a parity-check matrix for n=18, d_c=3, d_v=3, and subfigure (b) represents the factor graph.

(*The subfigure (a) is adopted from* [20])



(a) n1100k550dv9dc9 graph-ensemble

(b) n1100k550dv7dc7 graph-ensemble

(c) n1080k540dc9dv9 layered ensemble

(d) n1120k560dc7dv7 layered ensemble

Figure 14. Distance spectrum of the graph ensemble and layered ensemble LDGM codes.

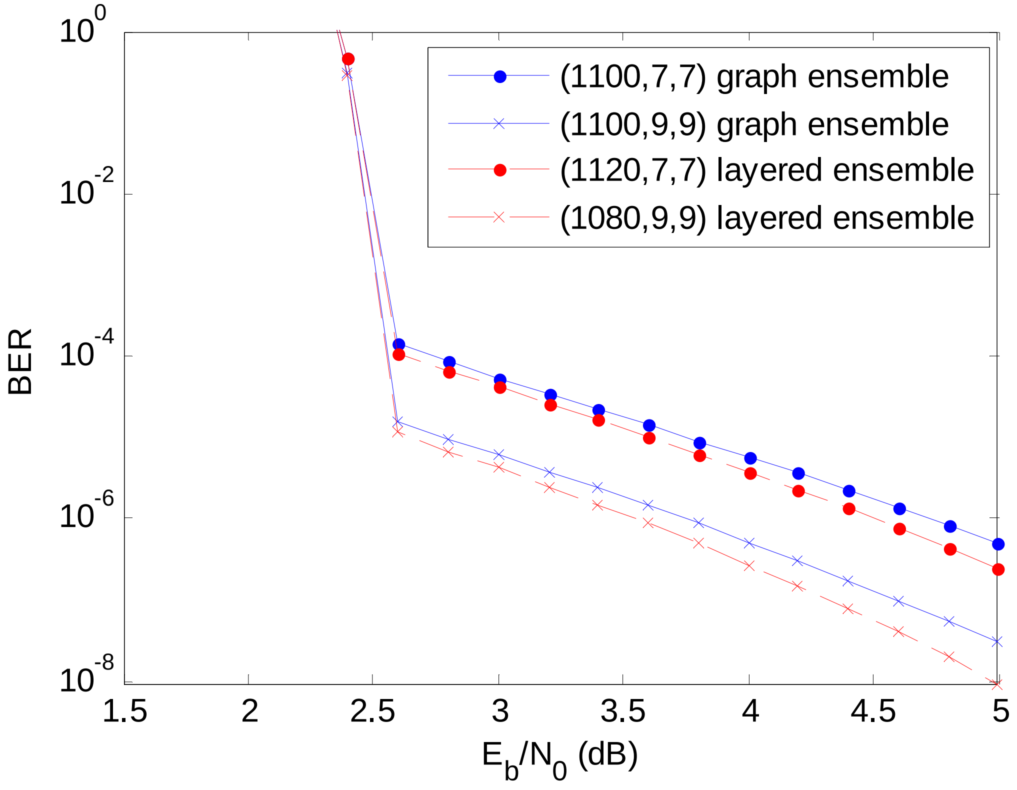
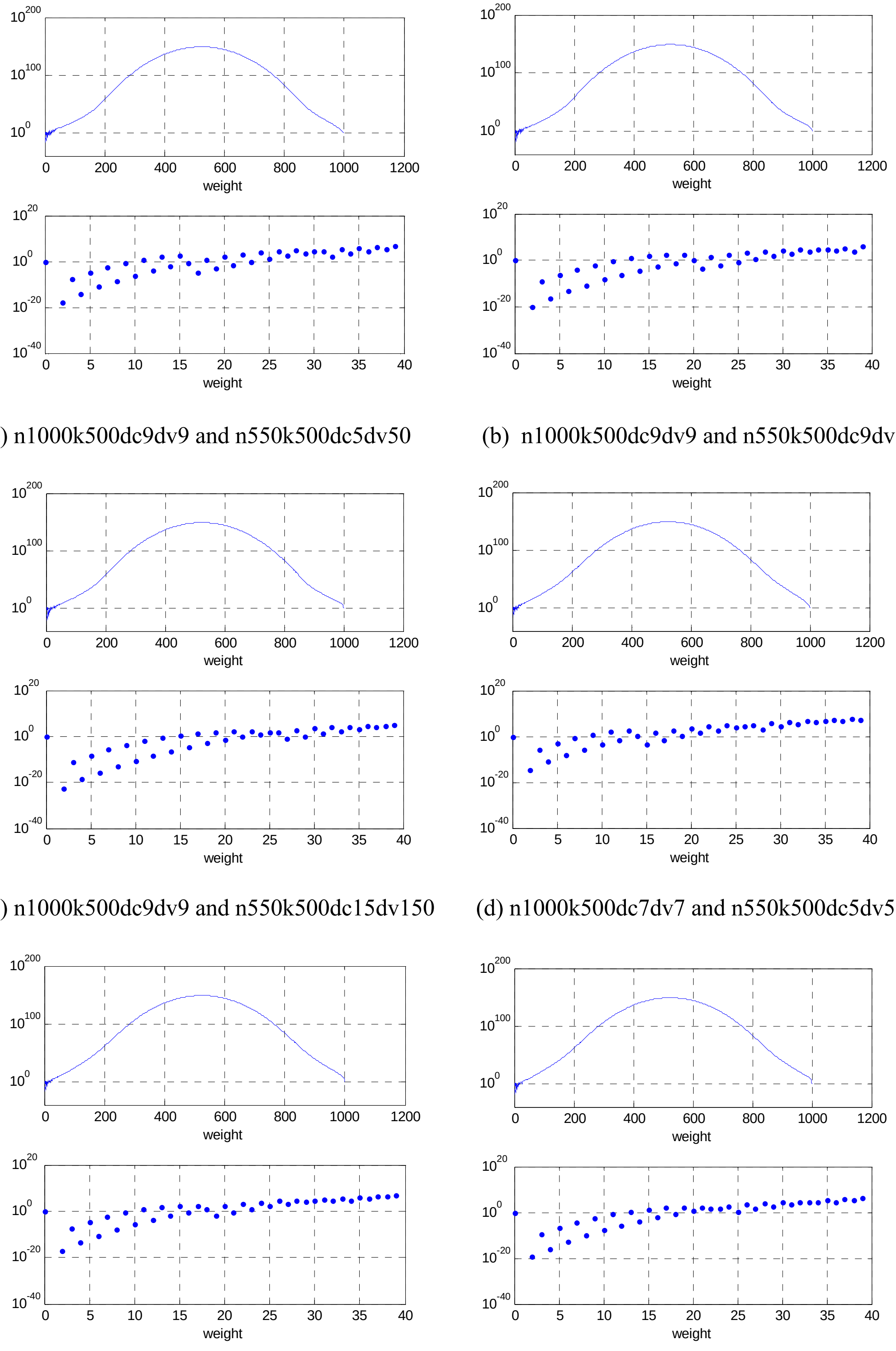


Figure 15. Union bound on BER performance of LDGM codes.

Graph-ensemble versus layered-ensemble. The layered ensemble has slightly lower error-floor than the graph-ensemble.



(a) n1000k500dc9dv9 and n550k500dc5dv50

(b) n1000k500dc9dv9 and n550k500dc9dv90

(c) n1000k500dc9dv9 and n550k500dc15dv150

(d) n1000k500dc7dv7 and n550k500dc5dv50

(e) n1000k500dc7dv7 and n550k500dc9dv90

(f) n1000k500dc7dv7 and n550k500dc15dv150

Figure 16. Distance distributions of PC-LDGM codes.

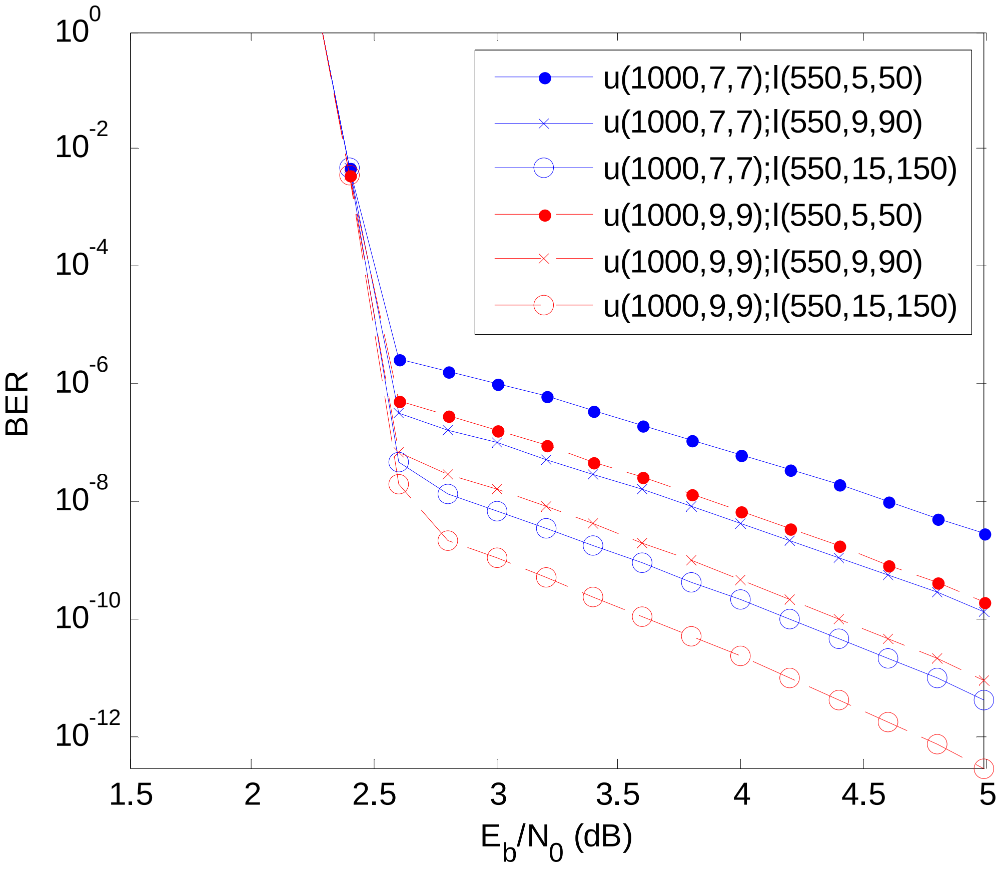
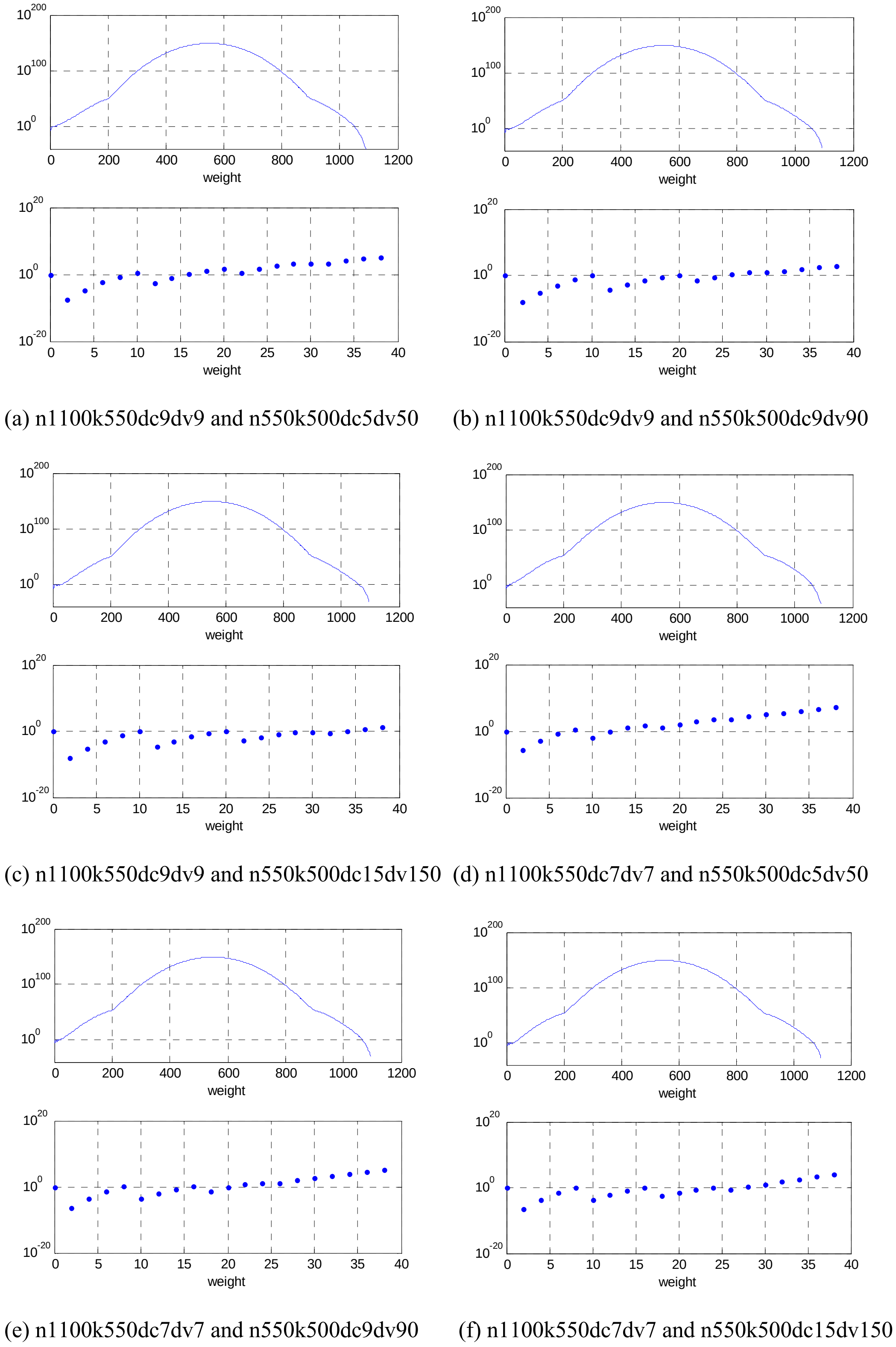


Figure 17. Union bound on BER performance of PC-LDGM codes.



(a) n1100k550dc9dv9 and n550k500dc5dv50 (b) n1100k550dc9dv9 and n550k500dc9dv90

(c) n1100k550dc9dv9 and n550k500dc15dv150 (d) n1100k550dc7dv7 and n550k500dc5dv50

(e) n1100k550dc7dv7 and n550k500dc9dv90 (f) n1100k550dc7dv7 and n550k500dc15dv150

Figure 18. Distance distributions of SC-LDGM codes.

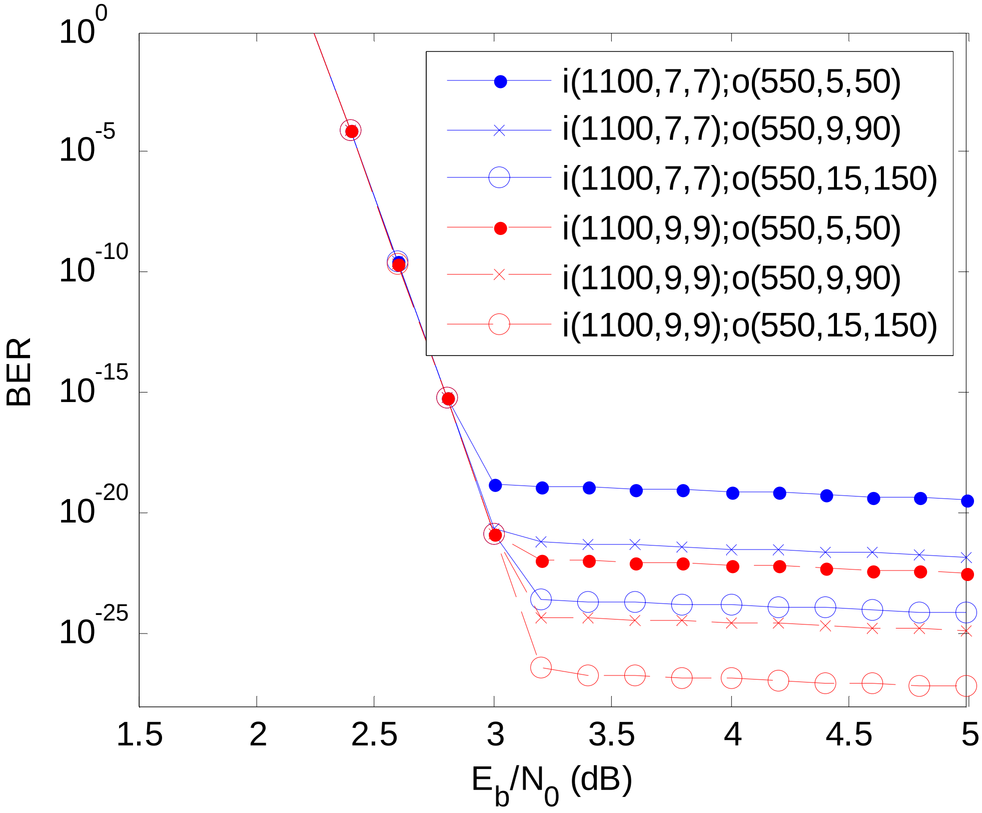


Figure 19. Union bound on BER performance of SC-LDGM codes.

4.0 SPACE-TIME MESH CODE FOR WMA-NETWORKS

We consider a wireless multiple access relay network where the sender nodes are aided by a number of relay nodes. A transmission of bit-messages is completed in two phases: in the first phase each sender node originates its message which is overheard at the relay node, and in the second phase each relay node transmits the parity bit calculated from the overheard bit-messages. Low-density parity-check (LDPC) codes are used at the sender nodes in the time-domain. At the access node, the received bits from multiple sender nodes and relay nodes are thus encoded in both the time and the spatial domain. We call this combination a space-time *mesh* code here. An iterative decoding scheme is designed for the mesh code and its BER performance in AWGN, fast Rayleigh fading, and quasi-static Rayleigh fading channels are investigated. We note that there is an important trade-off relation between the time-domain and the spatial domain coding. Namely, the time-domain coding is desired when the channel exhibits fast fading; while the spatial domain coding is preferred when the channel is in quasi-static fading state..

4.1 INTRODUCTION

Since the seminal paper by Ahlswede, Cai, Li, and Yeung [25] , the idea of *network-coding* has drawn a lot of interests from the research community. In the *network-coding* framework, an intermediate relay node can be configured to transmit the result of linear combination of its incoming messages over a finite field. It has been shown that the use of this network-coding can increase the traffic carrying capacity of certain wired networks in the multicast application [25][52]. Specifically, if senders and relays are only allowed to cope with binary messages, the linear combination operation is reduced to a simple modulo-2 addition. That is, a relay transmits the result of the binary parity-check operation of its incoming bit-messages. We call this the *parity-checking* network-coding in this chapter. The parity-checking network-coding has a further application in the wireless multiple-access relay network. Since all bits from the senders, information bits, and relays, the parity-check bits, are collected at the access node, the access node can virtually treat all the received bits as a codeword of a linear block code. Thus, the access node can utilize the *built-in* spatial-domain coding offered by the multiple-access relay network to improve the reception.

Bao and Li proposed a two-phase-transmission scenario for the multiple-access network similar to ours (refer to section 4.2 for a detailed description), and showed that the parity-checking network-coding is better than simple routing in simulations [35]. In [37], the authors further investigated the spatial-domain diversity offered by implementing the parity-checking network-coding. However, they assume that each sender processes only a single information bit to be transmitted at a time, rather than a coded bit stream. Thus, the time-domain diversity is not utilized in this scenario.

On the other hand, Hausl *et. al*. in [34] considered the time-domain coding, rather than a single-bit transmission, in a similar multiple access relay network. However, they considered a rather limited cooperation scheme in which there are only few senders, each of which employs low-density parity-check (LDPC) code. They investigated the performance of the iterative receiver at the access node in simulation. In this scenario, although the time-domain diversity is utilized, the spatial-domain diversity is not fully explored, due to lack of senders. Straightforwardly, a coding scheme that can jointly utilize time-domain and spatial-domain diversities is demanded.

In this chapter, we propose the idea of *space-time mesh code.* The space-time mesh code can utilize both the spatial and time diversity which might available in the channel. The other effect is that the block length of the code can be increased by combining signaling over the both dimensions. We provide an iterative decoder for this code, and present its bit error rate (BER) simulation results. We show that with the proposed coding framework we can investigate the trade-off relationship between the spatial and the time domain coding. Utilizing this tradeoff relation the network code can adapt to different channel condition in an optimal manner.

The organization of this chapter is as follows. An introduction is given in section 4.1. Section 4.2 provides the model of the wireless multiple access relay network. Section 4.3 discusses the space-time mesh codes and iterative decoders at access node. In section 4.4, experimental performance analysis is provided. We draw a conclusion and future work in section 4.5.

4.2 WMA-NETWORK MODEL

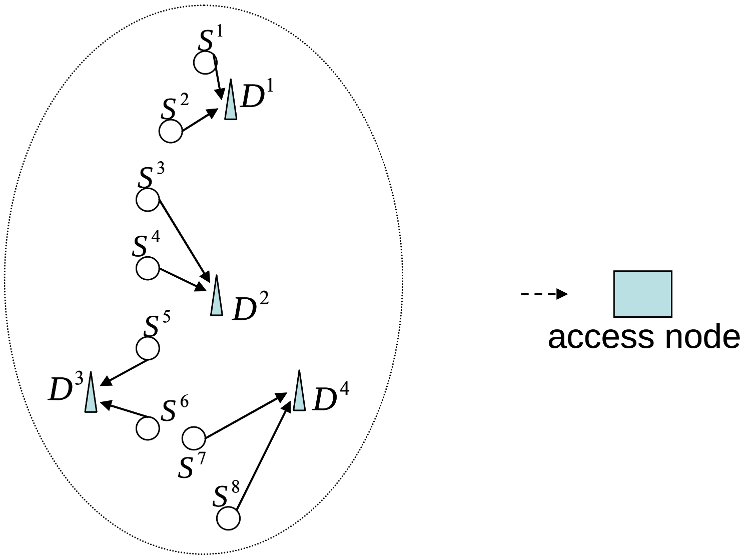


Figure 20. Example of a multiple-access relay network.

In the network $s^1 \cdots s^8$ are sender nodes and $D^1 \cdots D^4$ are relay nodes.

<u>A. System of interest</u>: The wireless multiple-access relay network is depicted in Figure 20. There is a single *access node* depicted as the square box. The circle nodes are the *traffic-originating* sender nodes and the triangle nodes are the *relay* nodes. Each sender node transmits *k* information bits independently generated from those of other sender nodes. This information bits are individually encoded with a low-density parity-check (LDPC) code. Here, we adopt the LDPC code due to its ability not only to achieve channel capacity [38], but also to cooperate with the spatial systematic low-density generator matrix (LDGM) code (details in section 4.3). An LDPC codeword of length *n* is to be transmitted to the access node through each sender's dedicated wireless channel. The dedicated channels imply that each sender has its own transmission channel which can be achieved either in a random-access manner or in a fixed-

access manner. For simplicity, here we assume that the common access node can provide the necessary synchronization and channel assignment, and consider the fixed-access manner in which the signals are multiplexed into a different time, a different frequency or a different spreading code. The objective is that no inter-user signal interference occurs at the access node. To better define the system, we assume

1. The sender and the relay nodes form a cluster such that these nodes within the cluster are randomly but closely located with each other. The access node is located (far) outside the cluster.
2. Each relay node has the capability to listen to signals from its neighboring sender nodes. A particular relay is able to pick up a number of channels on which the reception quality is good. It can decode the coded bit streams, if necessary, to ensure that the wireless links from the picked sender nodes to the relay node are error-free.

To utilize the broadcast nature of the wireless media and apply the notion of network-coding, we provide the following *two phase transmission schemes* built on [36].

In the first phase period, each sender node transmits a coded bit-stream to the access node. Meanwhile, because of the broadcast nature, a relay node can collect, and store in its buffer, a number of error-free messages from its neighboring sender nodes in this period. In the second phase period, each relay node transmits the calculated parity-checking bit-stream, by summation on its incoming bit-messages under mod-2 operation, to the access node through its dedicated channel (and hence no interference incurred).

Practical network-coding system for multicasting information in packet networks has been studied in [39]. We note that the parity-checking multiple-access relay network is highly applicable to real packet networks. Figure 21 shows a possible packet format for the system. The cluster ID indicates which cluster the packet belongs to, if more than one cluster exists. Similarly, the sender node ID in a sender node packet indicates which sender node the packet belongs to. The encoding vector in a relay node packet records the composition of this packet as a result of parity-checking the indicated source messages. That is, the payload in this packet is obtained from $\sum_{i=1}^{N_S} \mathbf{1}_{S_i} \mathbf{x}_i (\bmod 2)$, where $\mathbf{1}_{S_i} \in \{0,1\}$ is an indicator function, and $\mathbf{x}_i$ is the payload of S_i

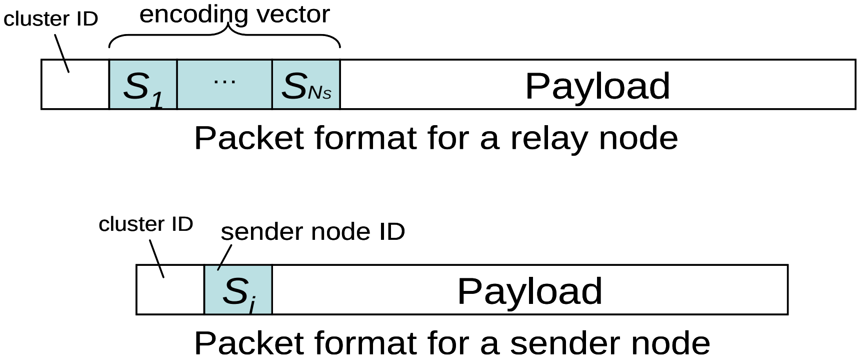


Figure 21. A possible packet format for practical system.

(*The figure is adopted from* [38])

<u>*B. Network channel model*</u>: Each sender's signal sent in different channels can be collected at the access node. The received signal $y_{s,t}$ at the access node is written as

$$y_{s,t} = \sqrt{E_S}\alpha_{s,t}x_{s,t} + w_{s,t}, \tag{4.1}$$

for $s = 1, 2, \cdots, N_s, \cdots, (N_s + N_D)$, and $t = 1, 2, \cdots, n$. The index *s* is for the spatial-channels, and *t* for the time index. E_S is the transmitted symbol energy at each sender or relay; $x_{s,t}$ is the binary phase shift keying symbol for the t^{th} time-epoch of the s^{th} transmitter. It either refers to the signal sent by the senders if the spatial index *s* is less than the number of sender nodes N_s, i.e. for $s = 1, 2, \cdots, N_s$, or the signal sent by the relays if $s = N_S + 1, \cdots, (N_S + N_D)$, where N_D is the number relay nodes. We assume perfect phase de-rotation. The fading gain is denoted as $\alpha_{s,t}$, samples of $\alpha_{s,t}$ are drawn from the Rayleigh distribution. For any fixed spatial-index *s*, the channel is called *quasi-static fading* when $\alpha_{s,t}$ is held as a constant during the whole codeword length (i.e., $\alpha_{s,t}$ is fixed once chosen for the duration of whole transmission period, $t = 1, 2, \cdots, n$). It independently varies from one period of the codeword to the other. On the other hand, the *fast fading* channel is implemented by having $\alpha_{s,t}$ independently varied at every time index t. In this chapter, we assume all the spatial channels are independent and undergo the same type of fading, i.e., all undergo either the quasi-static or the fast fading channel condition. It shall be noted that we can let $\alpha_{s,t} = 1$ for the AWGN channel.

4.3 SPACE-TIME MESH CODE AND THE ITERATIVE DECODER

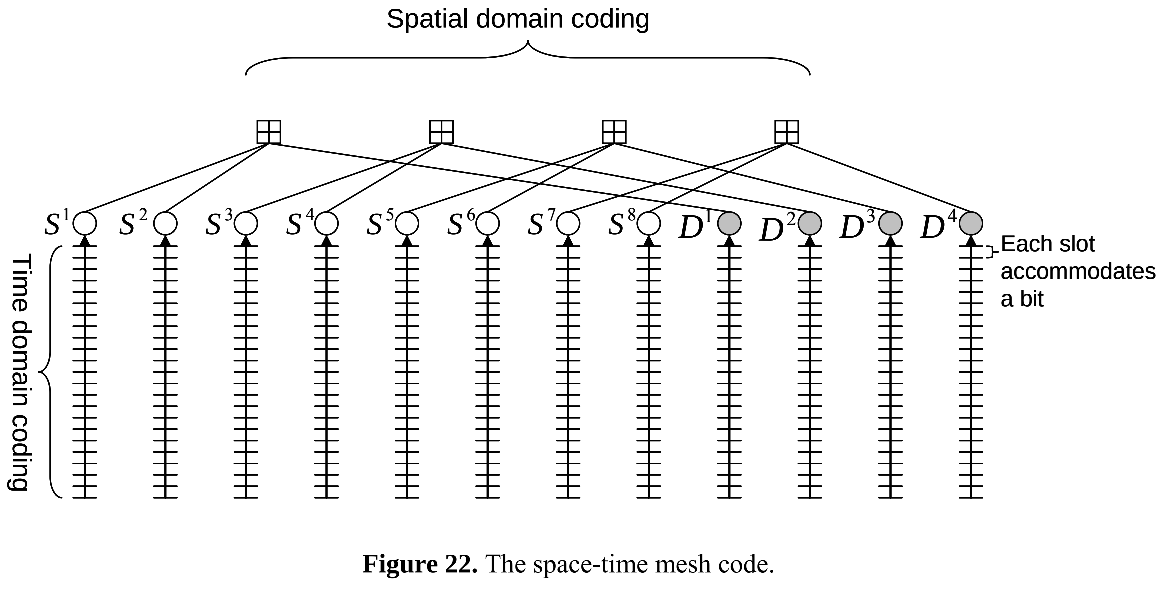


Figure 22. The space-time mesh code.

To visualize the relationship among transmitted message bits and parity bits received at the access station, we use Figure 22 which depicts an example of the *Tanner graph* for the space-time mesh code. The Tanner graph shown in the horizontal axis is for the coding done across the spatial domain. It represents the parity-check equations, generated through the two phase transmission scheme. The bits related by a parity-check equation sum up to zero under the mod-2 operation. The graph is formed across the transmitting and relay nodes for a single time-epoch. For each of the senders $S^1 \cdots S^8$, there is a vertical arrow which represents the time domain LDPC code. It has its own corresponding *Tanner graph,* although it is not shown there for simplicity. Thus, each bit transmitted either in time or space is related with some others. Due to this entanglement across time and space, we call this space-time *mesh* code in this chapter.

It shall be noted that the arrows for delays $D^1 \cdots D^4$ are the derivatives for the coded bit-streams of $S^1 \cdots S^8$, and are the consequences of the two phase transmission operation for a single epoch *t*. Moreover, we notice that the code in the spatial domain is in the form of Low-density Generator Matrix (LDGM) codes.

In general, the parity-check matrix $\mathbf{H}_{sp}$ of the spatial LDGM code for total N_S senders and N_D relays can be described by

$$\mathbf{H}_{sp} = \begin{bmatrix} | & | & & | & \\ \mathbf{h}_{sp,1} & \mathbf{h}_{sp,2} & \cdots & \mathbf{h}_{sp,N_S} & ; I_{N_D \times N_D} \\ | & | & & | & \end{bmatrix}_{N_D \times (N_S + N_D)}, \tag{4.2}$$

where the *j*-th position of 1's in the N_D by 1 vector $\mathbf{h}_{sp,k}$, $k = 1, \cdots, N_S$, represents there exists a error-free data link from the *k*-th sender to *j*-th relay. For each sender, $s = 1, 2, \ldots, N_S$, the parity-check matrices $\mathbf{H}_t^s$ of the time domain LDPC codes with code length *n* and code rate *k*/*n* is given by

$$\mathbf{H}_t^s = \begin{bmatrix} | & | & & | & & | \\ \mathbf{h}_{t,1}^s & \mathbf{h}_{t,2}^s & \cdots & \mathbf{h}_{t,k}^s & \dots & \mathbf{h}_{t,n}^s \\ | & | & & | & & | \end{bmatrix}_{(n-k)\times n} \tag{4.3}$$

for $s = 1, 2, \dots, N_S$. Given the parity-check matrices $\mathbf{H}_{sp}$ and $\mathbf{H}_t^s, s = 1, 2, \dots, N_S$, and consider a codeword of the space-time code by concatenating bits in Figure 22 row by row, it can be shown that the parity-check matrix $\mathbf{H}$ for the space-time mesh code is given by the matrix (4.4).

$$\mathbf{H} = \begin{bmatrix}
\begin{bmatrix} | & | & & | & \\ \mathbf{h}_{sp,1} & \mathbf{h}_{sp,2} & \cdots & \mathbf{h}_{sp,N_S} & ;\mathbf{I}_{N_D\times N_D} \\ | & | & & | & \end{bmatrix}_1 & \mathbf{0}_{N_D\times(N_S+N_D)} & \cdots & \mathbf{0}_{N_D\times(N_S+N_D)} \\
\mathbf{0}_{N_D\times(N_S+N_D)} & \begin{bmatrix} | & | & & | & \\ \mathbf{h}_{s,1} & \mathbf{h}_{s,2} & \cdots & \mathbf{h}_{s,N_S} & ;\mathbf{I}_{N_D\times N_D} \\ | & | & & | & \end{bmatrix}_2 & \cdots & \mathbf{0}_{N_D\times(N_S+N_D)} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{0}_{N_D\times(N_S+N_D)} & \mathbf{0}_{N_D\times(N_S+N_D)} & \cdots & \begin{bmatrix} | & | & & | & \\ \mathbf{h}_{s,1} & \mathbf{h}_{s,2} & \cdots & \mathbf{h}_{s,N_S} & ;\mathbf{I}_{N_D\times N_D} \\ | & | & & | & \end{bmatrix}_n \\
\begin{bmatrix} | & | & & | & \\ \mathbf{h}_{t,1}^1 & \mathbf{0}_{v,2} & \cdots & \mathbf{0}_{v,N_S} & ;\mathbf{0}_{(n-k)\times N_D} \\ | & | & & | & \end{bmatrix}_1 & \begin{bmatrix} | & | & & | & \\ \mathbf{h}_{t,2}^1 & \mathbf{0}_{v,2} & \cdots & \mathbf{0}_{v,N_S} & ;\mathbf{0}_{(n-k)\times N_D} \\ | & | & & | & \end{bmatrix}_2 & \cdots & \begin{bmatrix} | & | & & | & \\ \mathbf{h}_{t,n}^1 & \mathbf{0}_{v,2} & \cdots & \mathbf{0}_{v,N_S} & ;\mathbf{0}_{(n-k)\times N_D} \\ | & | & & | & \end{bmatrix}_n \\
\begin{bmatrix} | & | & & | & \\ \mathbf{0}_{v,2} & \mathbf{h}_{t,1}^2 & \cdots & \mathbf{0}_{v,N_S} & ;\mathbf{0}_{(n-k)\times N_D} \\ | & | & & | & \end{bmatrix}_1 & \begin{bmatrix} | & | & & | & \\ \mathbf{0}_{v,2} & \mathbf{h}_{t,2}^2 & \cdots & \mathbf{0}_{v,N_S} & ;\mathbf{0}_{(n-k)\times N_D} \\ | & | & & | & \end{bmatrix}_1 & \cdots & \begin{bmatrix} | & | & & | & \\ \mathbf{0}_{v,2} & \mathbf{h}_{t,n}^2 & \cdots & \mathbf{0}_{v,N_S} & ;\mathbf{0}_{(n-k)\times N_D} \\ | & | & & | & \end{bmatrix}_1 \\
\vdots & \vdots & \dots & \vdots \\
\begin{bmatrix} | & | & & | & \\ \mathbf{0}_{v,2} & \mathbf{0}_{v,3} & \cdots & \mathbf{h}_{t,1}^{N_S} & ;\mathbf{0}_{(n-k)\times N_D} \\ | & | & & | & \end{bmatrix}_1 & \begin{bmatrix} | & | & & | & \\ \mathbf{0}_{v,2} & \mathbf{0}_{v,3} & \cdots & \mathbf{h}_{t,2}^{N_S} & ;\mathbf{0}_{(n-k)\times N_D} \\ | & | & & | & \end{bmatrix}_1 & \cdots & \begin{bmatrix} | & | & & | & \\ \mathbf{0}_{v,2} & \mathbf{0}_{v,3} & \cdots & \mathbf{h}_{t,n}^{N_S} & ;\mathbf{0}_{(n-k)\times N_D} \\ | & | & & | & \end{bmatrix}_1
\end{bmatrix}_{\substack{n(N_D+N_S)-kN_S \\ \times n(N_D+N_S)}} \tag{4.4}$$

We denote that $\mathbf{0}_{v,k}$, $k = 2, \cdots, N_S$, is an n by 1 all zero vector. The code rate of the mesh code is defined as the ratio of the total number of information bits to the total transmitted coded bits such that

$$R_{Mesh} = \frac{kN_S}{n(N_S + N_D)} \tag{4.5}$$

One thing that can be noted here is that the length of the mesh code can become very large, proportionately increasing with the number of the cooperating sender nodes, and that of the relay nodes. Thus, the space-time mesh code can be used to subsume both the traditional time-domain coding and the emerging spatial network-coding.

After having the parity-check matrix of a typical space-time mesh code (4.4), the access node can apply the standard message-passing iterative decoding algorithm to decode the received message from all senders. In this chapter, we adopt the Gallager's sum-product algorithm [26][38]for decoding of the mesh code, and refer to it as the *mesh decoder*. Based on the check equations embedded in (4.4), the mesh decoder updates the extrinsic log-likelihood ratios (LLRs) between the check nodes and the bit nodes.

One alternative decoding strategy is to divide and conquer. From Figure 22, we know that the access node will get each row of the mesh code one at a time. The access node can choose to decode the LDGM-code coded bits row by row upon it receives each of them. In fact, without making hard decisions on the log-likelihood ratios (LLRs), they can be forwarded to the next step. After all rows have been processed, decoding over the time domain coding can be initiated at each column. This phase of decoding is based on the code graph of the time-domain LDPC code. The extrinsic LLRs from the first step can be used to initiate this decoding process.

This option can shorten the decoding latency and may lead to a reduced complexity implementation solution thanks to a shortened decoding block at each decoder. However, if we only allow a one way "message flow," i.e., only a single flow from the spatial-domain to the time-domain, the method will suffer from performance loss. In fact, our simulation results confirm that the performance of this method can be as much as about 3dB worse than that of the mesh code decoder. The performance of the second method would probably be improved by further iterations between the spatial and the temporal decoders. However, we note that this iterative decoder would still be inferior to the mesh decoder in terms of the performance.

In the sequel, therefore, we assume the use of the mesh decoder, and focus on the determination of the performance of the mesh decoder under different channel settings.

4.4 EXPERIMENTAL PERFORMANCE ANALYSIS

In this section, we first compare the performance of the *mesh* code in various temporal and spatial domain settings. One benefit of this study is that we will be able to adjust the parameters of the mesh code according to the variation of the channel's fading state. In this chapter, we show our results on the following four settings:

1. N_S=20, N_D=10, *ICR*=8, *n*=200 LDPC codes
2. *N*s=200, N_D=100, *ICR*=8, *n*=20 LDPC codes
3. N_S=20, N_D=0, *ICR*=0, *n*=200 LDPC codes
4. N_S=200, N_D=0, *ICR*=0, *n*=20 LDPC codes

where N_S is the number of senders, N_D is the number of relays, *ICR* is the number of Incoming Connections per Relay. For example, the *ICR* in Figure 20 is two. We use 5 iterations for the mesh code decoder. Figure 23 shows the extensive computer simulation results of these settings. Here we let each and every relay/sender has the same incoming/outgoing connections. The parameter *n* is the codeword length of the time-domain LDPC codes. We use the ensemble of the Gallager's (3, 6) LDPC codes, i.e., three 1's in each column and six 1's in each row of the parity-check matrix. In addition, each sender has its own parity-check matrix

Based on the code rate defined in (4.5), the BER curves are calibrated with respect to E_b / N_0, the ratio of the information bit energy to the power-spectral density, for fair comparison. Settings 1 and 2 represent the performance of the space-time mesh codes, whereas settings 3 and 4 represent the performance, averaged over all senders, of a single LDPC decoder ($N_{\mathrm{D}} = 0$). As expected, the space-time entangled mesh code easily out-performs the single LDPC code. It should be noted that the both entangled mesh codes in settings 1 and 2 are of code length 6000 which is much longer than 200 and 20 of the settings 3 and 4 respectively.

Let us consider the settings 1 and 2 more closely. They both have the same size parity-check matrix of the form given in (4.4), and thus have the same code length and the same code rate. An interesting observation is that the performance of the two mesh codes is however very different. Setting 2 tends to produce an error floor in BER curves while Setting 1 does not. This phenomenon is in fact somewhat expected and can be explained from the parity-check matrix of the mesh codes.

Consider the extreme case for $n = 1$ and only 1-bit information is to be transmitted at each sender such that there is no time-domain LDPC codes applied. Then, the dimension of the parity-check matrix given in (4.4) becomes N_D by $N_D + N_S$. This matrix is exactly in the form of LDGM code. That is, the parity-check matrix is in the systematic form, $\mathrm{H}_{LDGM} = [\mathbf{P}; \mathbf{I}]$, where $\mathbf{P}$ is a sparse matrix and $\mathbf{I}$ is the identity matrix. We note that the minimal distance of regular LDGM codes are equal to the *number of 1's in a column of the P matrix* plus *one*, i.e., *degree*+1, and this small minimal distance causes a significant error floor. For a fixed code length, as we observe from (4.4), the higher the ratio N_D / n, the more the code looks and behaves like an LDGM code (e.g. Setting 2). On the other hand, the lower the ratio N_D / n, the more the code looks and behaves like an LDPC code, i.e., the proportion of the identity matrix in (4.4) becomes smaller (e.g. Setting 1). It shall be noticed that the minimum distance of an LDPC code increases in proportion to the code length [26]. Thus, Setting 1 shows much better BER performance than Setting 2 does.

Now we investigate the first two settings in different channel conditions, such as AWGN, the fast Rayleigh fading, and the quasi-static Raleigh fading channels and see how the BER performance changes in these different channels. The results shown in Figure 24 indicate that in both AWGN and fast Rayleigh fading channels, the LDPC-like mesh code, setting 1, is better than the LDGM-like code, setting 2. On the other hand, for the quasi-static Rayleigh channel the results of the LDGM-like mesh code, setting 2, is better. This phenomenon can be explained in the following way. For the fast fading channel, each and every redundant bit, either in time or spatial domain, suffers an independent fading coefficient drawn from the same Rayleigh distribution. Thus, statistically, there is no difference as to placing the redundant bits either in the time or in the spatial domain. Even though the total number of redundant bits are the same in Setting 1 and Setting 2, placing more redundant bits in the time domain makes the mesh code shaped more like an LDPC code, which out-performs the LDGM-like mesh code formed by setting 2. For the quasi-static fading channel, however, the situation takes a different form. Under quasi-static fading, if a sender suffers a deep fade, then all of its bit-messages are likely to be lost. There is no time-diversity benefit at all. Thus, it is more beneficial to put coding effort more in the spatial domain.

From the results so far, it is worth to note that the LDGM-like mesh code would be beneficial in certain network situations where some links from senders/relays to the access node are completely blocked (erased) perhaps by their surrounding building, but the other links from the senders/relays to the access node are clean.

In addition, it is worthy to mention that the *ICR* determines, and is proportional to, the degree of the spatial domain LDGM code. Also, recall that the minimum distance of the LDGM code is proportional to the degree. Thus, for LDGM-like mesh codes such as Setting 2, choosing a higher ICR value will have the error floor lowered. Our simulation results show that by increasing *ICR* to 8 from 4 in Setting 2, the error floor is lowered by as much as 1dB at BER 10^{-4}.

Finally, we have considered ways to pick a better mesh code from the ensemble. We first notice that if we let all the sender nodes employ exactly the same parity-check matrix for their LDPC codes, instead of varying them one from another, the parity-check matrix of the mesh code (4.4) becomes quite regular. This regularity may cause short cycles, which limit the extrinsic information flow in the iterative decoding and hence degrade the BER performance. However, by randomly choosing a single parity-check matrix $\mathbf{H}_t$ and applying it to all sender nodes, our experimental result shows that the BER performance is only 0.5 dB worse at the BER 10^{-4} in the setting Ns=8, Nd=4, *ICR*=2, and n=30. At the expense of this much performance loss, one possible benefit is the reduction in hardware complexity of the iterative decoder thanks to the regular structure.

4.5 CONCLUSION

We propose the idea of the space-time mesh codes for the multiple-access relay network, and present detailed discussions on the BER performance of the codes, under the sum-product iterative decoding algorithm. Given a fixed length of the mesh code, the parity-check matrix can be varied from a LDGM-like code to a LDPC-like code by choosing an appropriate parametric setting of the mesh code for the multiple-access relay network. The more number of relays and the shorter LDPC code on senders, the more LDGM-like the mesh code becomes. We provided the BER simulation results of the mesh code in different types of channels such as AWGN, fast fading, and quasi-static fading channels. The LDPC-like mesh code out-performs the LDGM-like mesh code in AWGN and fast fading channels, whereas the LDGM-like mesh code out-performs the LDPC-like mesh code in quasi-static fading channels. Namely, we confirm that time-domain coding should be emphasized when the channel exhibits large time-diversity benefit while the spatial-domain coding should be emphasized when the channel exhibits large spatial diversity benefit.

Our future work focuses on finding the distance spectrum properties of an ensemble of mesh codes and providing performance prediction based on union bound techniques. We envision that this analytic tool can serve as design guidance for finding the best network code for different multiple-access network.

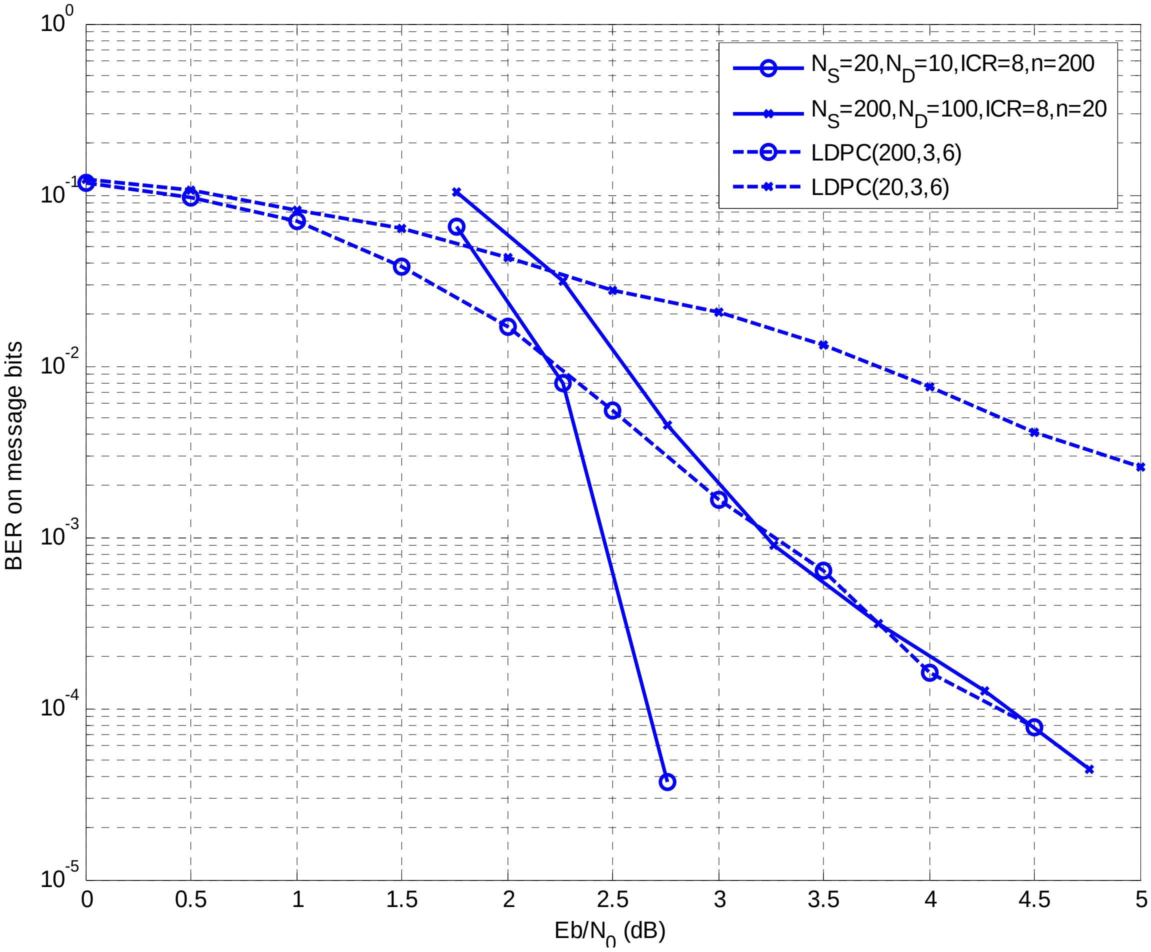


Figure 23. BER performance: the space-time mesh code versus the single LDPC code.

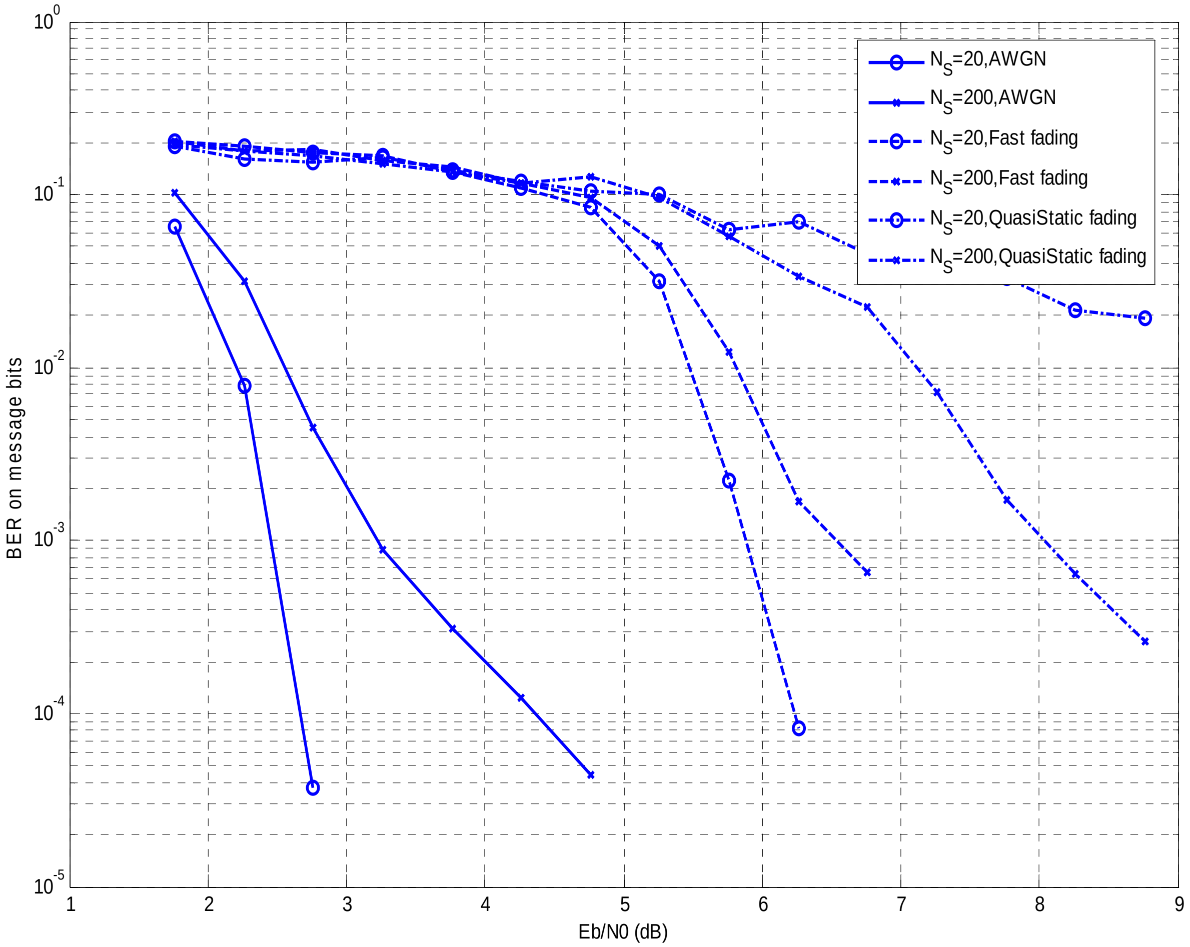


Figure 24. BER performance of the space-time mesh code in different types of channels. In the figure, AWGN channels, fast Rayleigh fading channels , and quasi-static Rayleigh fading channels are simulated.

5.0 A SIMPLE PHYSICAL LAYER SECURITY MECHANISM FOR WMA-NETWORKS

Relay assisted wireless multiple access networks equipped with network-coding via a two-phase transmission protocol have drawn significant interests in recent years owing to their robust performance. However, relays can be compromised, and malicious data from compromised relays can be lethal to reliable network communications. For this problem, we propose a simple, but robust, security mechanism operating within the realm of channel coding in the physical layer. The method can not only check the integrity of the data from relay nodes but also reverse the effect of attack

5.1 INTRODUCTION

Network-coding over wireless multiple access networks, implemented with a two-phase transmission (see section 5.2), has received significant attention recently, as seen from examples in [35][40][41][42][43]. Messages overheard in the first phase transmission are parity checked by relay nodes and then forwarded to the access node in the second phase transmission. By means of this two phase transmission, a virtual low density parity-check (LDPC) code graph can be formed across the spatial domain, in which parity bits forwarded by a relay node are produced from a selected subset of overheard source bits. We call this Wireless Multiple Access (WMA) network code, and call such networks WMA networks. The WMA networks are, thus, related to diversity-coding networks [44][45] and relay-assisted CDMA networks [46]. By allowing the relays to encode the messages from source nodes, robust communications between the source nodes and the access node can be established.

The very feature that allows relay nodes to mix (parity-check) their incoming data, however, makes the network vulnerable to *jamming attack [47].* An adversary can gain access to the relay nodes which are often left alone in an open field and compromise them to produce malicious messages afterward. In WMA networks, since the WMA network code is an expander graph, a single compromised relay can have an effect on the messages of many source nodes. With a strong attack (many relays attacked), the decoder at the access node may fail causing frequent access errors. To protect these network-coding networks from jamming attacks, *homomorphic hashing schemes* (HHS) have been proposed [47][48].

Homomorphic hashing functions have the property that the hash value of a linear combination of inputs can be constructed by a combination of the individual hash values. For example, for three input binary messages $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$, the hash value of a linear combination $\tilde{\mathbf{m}} = c_1\mathbf{m}_1 + c_2\mathbf{m}_2 + c_3\mathbf{m}_3$ can be obtained from product of individual hashes, i.e., $h(\tilde{\mathbf{m}}) = h^{c_1}(\mathbf{m}_1)\cdot h^{c_2}(\mathbf{m}_2)\cdot h^{c_3}(\mathbf{m}_3)$. Then, with *error free* observation of the messages $h(\tilde{\mathbf{m}})$, $h(\mathbf{m}_1)$, $h(\mathbf{m}_2)$ and $h(\mathbf{m}_3)$ (or $h(\tilde{\mathbf{m}})$, $\mathbf{m}_1$, $\mathbf{m}_2$ and $\mathbf{m}_3$), the integrity of the encoded message $\tilde{\mathbf{m}}$ can be checked before the transmission of the message $\tilde{\mathbf{m}}$. Since $h(\tilde{\mathbf{m}})$ is of much smaller size than $\tilde{\mathbf{m}}$ is, the HHS provides an advantage in terms of bandwidth efficiency.

However, the threat of jamming attacks remains unsolved in the WMA networks. First, it is easy for sophisticated attackers to only transmit a correct $h(\tilde{\mathbf{m}})$, but then still transmit a corrupted message $\tilde{\mathbf{m}}$. Hence, the HHS can completely fail, and in fact a real-time monitoring on the data integrity of $\tilde{\mathbf{m}}$ is needed. Second, we notice that it is trivial to determine whether or not a relay node is under attack with the availability of the messages $\tilde{\mathbf{m}}$, $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$. Decoding for each message individually can be made with the use of conventional channel coding techniques. But the individual decoding of messages will nullify the key performance enhancing feature of the WMA network code: the coding benefit obtained through the network code and its joint decoding on the *soft* signals received from both the source and the relay nodes.

We are interested in a security mechanism that can check the data integrity while allowing the joint decoding operation. The main contribution of this chapter therefore is to propose a security mechanism for the WMA network code and its decoder. It is capable of both checking data integrity and correcting errors caused by jamming attacks and channel noises; thus it enables robust and secure communications for the WMA networks.

Code based cryptosystems such as McEliece or Niederreiter systems have been studied in the past for public-key cryptography in open networks [49][50]. Physical layer treatment for secure wireless transmissions has also been considered in [51]. In this chapter, we initiate a new effort on the network code based physical layer security to remedy jamming attacks for the WMA networks. By exploiting the parity check relationships of the graph code constructed across the network, we propose an attack detection method to test the integrity of data from relay nodes. Besides, methods for reversing the effect of attacks are provided (see section 5.3). We show that the proposed method is robust and effective via simulation (see section 5.4).

5.2 NETWORK MODEL

As depicted in Figure 25, we consider the relay assisted wireless multiple access network in which K source nodes, denoted by $S^1, S^2, ..., S^K$, forward their messages to the access node with the assistance of L relay nodes, denoted by $R^1, R^2, \cdots, R^L$. A rate $K/(L+K)$ network code formed with two phase transmission protocol can be implemented as follows.

<u>*Phase I: Originating Transmission Phase*</u>: Source nodes send their messages directly to the access node through their orthogonal channels. For the k^{th} source node, let the message be a binary codeword of length n, denoted by $\mathbf{m}^{[k]} = [m_1^{[k]}, m_2^{[k]}, ..., m_n^{[k]}]$, where $m_i^{[k]} \in \{0,1\}$. The binary sequence is modulated via binary phase shift keying (BPSK), i.e., $x_i^{[k]} = 2m_i^{[k]} - 1,\ x_i^{[k]} \in \{-1,1\}$. The received signals at the access node under additive white Gaussian noise (AWGN) channels are written as $y_i^{[k]} = \sqrt{E^{[k]}} x_i^{[k]} + w_i^{[k]}$, for $i = 1,2,...,n$ and $k = 1,2,...,K$. $E^{[k]}$ is the transmission symbol energy of the k^{th} source node, and $w_i^{[k]}$ is the additive noise drawn from i.i.d. Gaussian distribution of zero mean and variance $N_0/2$. Because of the broadcast nature of the wireless transmission, relay nodes can overhear sources' transmissions, and each is able to successfully decode messages from a subset of sources.

<u>*Phase II: Relaying Transmission Phase*</u>: Let $\mathcal{D}_l$ denote the set of indices for successfully decoded source nodes for the l^{th} relay node. The l^{th} relay node successfully overhears and decodes the message from the k^{th} source node if $k \in \mathcal{D}_l$. The l^{th} relay node, for $l = 1,2,...,L$, encodes the overheard information by XORing the overheard messages $m_i^{[k]}$, $k \in \mathcal{D}_l$. We denote it as $m_i^{[l+K]} = \oplus \sum_{k \in \mathcal{D}_l} m_i^{[k]}$. Note that this linear encoding complies well with the idea of network-coding [52][53]. In phase II, each relay node transmits the processed binary sequences of length n to

the access node through its orthogonal channel. Then, the received signals at the access node can be written as $y_i^{[l+K]} = \sqrt{E^{[l+K]}}(2m_i^{[l+K]} - 1) + w_i^{[l+K]}$, for $l = 1, 2, ..., L$ and $i = 1, 2, ..., n$. Similary, $E^{[l+k]}$ is the transmission symbol energy of the l^{th} relay node, and $w_i^{[l+k]}$ is the i.i.d. Gaussian distributed noise of zero mean and variance $N_0 / 2$.

The consequence of performing the two phase transmission is that a systematic graph-code is created across the spatial network domain. As depicted in Figure 25, the circles represent source nodes, while the triangles represent relay nodes. Each relay node can overhear some source nodes. The parity-checkings of these overheard messages at relay nodes are represented by edges in the bi-partite graph. In the graph, each relay node shown as the darkened circle is paired with a *check node* shown as the square. Thus, there are equal number of relay nodes and check nodes, and there is a single edge connecting each relay to its check node. The number of edges connecting a check node to source nodes can be more than one. It is the number of messages that the relay node can overhear and successfully decode. Thus, the number of edges connecting the check node paired with the l^{th} relay node to the source nodes on the graph-code is the size of the set $\mathcal{D}_l$.

The parity-check relationships for the messages from source nodes and relay nodes can be succinctly described in the bi-partite graph shown in Figure 25, which are given by

$$\begin{cases} m_i^{[1]} \oplus m_i^{[2]} \oplus m_i^{[3]} \oplus m_i^{[1+8]} = 0 \\ m_i^{[3]} \oplus m_i^{[4]} \oplus m_i^{[2+8]} = 0 \\ m_i^{[4]} \oplus m_i^{[5]} \oplus m_i^{[6]} \oplus m_i^{[3+8]} = 0 \\ m_i^{[6]} \oplus m_i^{[7]} \oplus m_i^{[8]} \oplus m_i^{[4+8]} = 0 \end{cases} \quad \forall i \cdot$$

Utilizing these parity-check relationships, a parity-check matrix can be formed, and a standard message passing (MP) decoder can be implemented at the access node. That is, the decoder takes the received signals carrying for the messages from both source nodes and relay nodes, forms log-likelihood ratios (LLRs), and feeds them to the message passing decoder working on the bi-partite graph. After a fixed number of iterations, or after convergence, the independent messages from sources can be decoded. This spatial domain coding provides significant coding and diversity benefits [35][40].

5.3 METHODS FOR DETECTING AND RELIEVING ATTACKS

Relay nodes are often deployed in an open field, and thus are vulnerable to attacks. Many kinds of attacks are conceivable. Here we consider the kind of attacks in which an adversary directly alters the messages forwarded by relay nodes. First, the adversary can choose to block or erase the messages from relay nodes. In such a case, the channel between relay nodes and the access node can be modeled as erasure channels. This kind of attack is easy to detect due to the absence of the received signals. Once labeled as compromised nodes, the access node can simply discard the messages from these relay nodes and feed the MP decoder with zero LLRs. Second, the adversary can choose to alter the message in a way that is not easily detectable. One choice is to flip the binary value of the parity bit. This is non-trivial to detect, and will lead to the worst situation in terms of its impact on decoder's performance degradation.

The second type of attack, which is stronger, is the main consideration in the chapter. In subsequent sections, we present our methods for detecting and reversing the effect of attack.

Detecting Attacks: At the access node, the MP decoding algorithm is run on the bi-partite graph. This algorithm is set to run once it is fed with channel LLRs. The channel LLRs are generated from the signals received from the source and relay channels. The MP decoder refines the initial LLRs by enforcing the parity-check relationships and produces posterior LLRs. This refinement can continue for a specified number of iterations, say for the maximum iteration count. Then, the MP stops and completes the decoding by thresholding the most updated posterior LLRs at the last iteration. Now, we discuss how the normal MP decoding algorithm can be modified into an attack detection mechanism. Let's denote $LLRc$ as the channel LLR, and denote $LLRe$ as the extrinsic LLR obtained at the last iteration. Also, let's denote $LLRc_H$ and $LLRe_H$ as their hard decisions, i.e., $LLRc_H = 1 \text{ if } LLRc > 0 \quad LLRc_H = 0 \text{ otherwise}$, and similarly, $LLRe_H = 1 \text{ if } LLRe > 0$, $LLRe_H = 0 \text{ otherwise}$.

As an attack detection mechanism to detect whether a relay node is under attack or not, we propose the *polarity contradiction* (PC) probability between the two numbers $LLRc_H$ and $LLRe_H$.

It is inspired by the analysis given below. For this analysis, we assume the system is operating at a decent signal to noise ratio (SNR) at which 10^{-3} or smaller bit error rates are expected, without an attack. Since the graph-code is a linear code, without loss of generality, the all-zero message is assumed to be transmitted from source nodes. This is for the analysis purpose only. We use randomly generated codeword transmission in simulations given in section 5.4 to test our detection method (given in Table 2).

Analysis on Attack Detection: Let δ be the binary random variable indicating the polarity contradiction between the two hard decisions, $LLRc_H$ and $LLRe_H$. That is, $\delta = 1$ when they contradict with each other, and $\delta = 0$ otherwise. Hence, the PC probability for a relay node can be written as

$$\begin{aligned} P(\delta = 1) = & P(LLRc_H = 1 \mid LLRe_H = 0)P(LLRe_H = 0) \\ & + P(LLRc_H = 0 \mid LLRe_H = 1)P(LLRe_H = 1). \end{aligned} \tag{5.1}$$

Using the idea from density evolution [54], and the assumption of a decent SNR where the graph-code is decodable with low error rates, it implies the mean of $LLRe$ is far below zero. This suggests that $P(LLRe_H = 1) \to 0$ and $P(LLRe_H = 0) \to 1$. Thus, we have

$$\begin{aligned} P(\delta = 1) &\approx P(LLRc_H = 1 \mid LLRe_H = 0) \\ &= P(LLRc_H = 1) \end{aligned}. \tag{5.2}$$

For the second line, we use the assumption that $LLRc_H$ and $LLRe_H$ are independent after the maximum number of iterations. This assumption is valid since $LLRe_H$ is generated after all information from other nodes have been incorporated via iterations.

Eq. (5.2) implies the following: Under the assumption that there is no attack, PC probability $P(\delta = 1)$ can be accurately approximated by the probability of a positive channel LLR, i.e., $P(LLRc_H = 1)$.

Now, let us include attack in the input/output model for relay nodes. That is, we have $y = \sqrt{E}\left(2(a \oplus x) - 1\right) + w$, where a is the random variable indicating the probability of an attack on the relay node, i.e., $a = 1$ if the adversary attacks, and $a = 0$ otherwise. Then, the probability $P(LLRc_H = 1)$ can then be written by

$$\begin{aligned} P(LLRc_H = 1) = P(a = 0)P(LLRc_H = 1 \mid a = 0) \\ + P(a = 1)P(LLRc_H = 1 \mid a = 1) \end{aligned}. \tag{5.3}$$

In AWGN channels, the $LLRc$ of the all-zero codeword is Gaussian distributed with the mean $\mu = -4E_s / N_0$ and the variance $\sigma^2 = 2|\mu|$. The conditional probability can be calculated from

$$\begin{cases} P(LLRc_H = 1 \mid a = 0) = 0.5\, erfc(\sqrt{\mu / 2}) \\ P(LLRc_H = 1 \mid a = 1) = 1 - 0.5\, erfc(\sqrt{\mu / 2}) \end{cases}. \tag{5.4}$$

It shall be noticed that $0.5\, erfc(\sqrt{\mu / 2}) < 1 - 0.5\, erfc(\sqrt{\mu / 2})$ when $\mu < 0$.

Eq. (5.3) can be used to find the estimator for attack detection. If there is no attack, i.e., $P(a=0)=1$, then the probability is $P(LLRc_H=1)=P(LLRc_H=1\,|\,a=0)$. For a given SNR, this probability can be calculated from (5.4). Let us call it *baseline PC* probability. Now let's see what happens when $P(a=1)\neq 0$. Looking back at (5.3), we first notice that $P(LLRc_H=1\,|\,a=0)<P(LLRc_H=1\,|\,a=1)$ and $P(a=0)+P(a=1)=1$, we see that as the attack probability $P(a=1)$ is increased from zero, the PC probability grow larger than the baseline PC probability as well. In other word, for any relay node under attack, the PC probability grows larger than the baseline PC probability.

The analysis so far implies that by observing the level of polarity contradiction for each relay we are able to determine whether or not a relay is under attack. We develop this idea into an attack detection method in the next subsection.

Attack Detection Algorithm: In this subsection, we aim to provide an algorithm to estimate the polarity contradiction probability for the l^{th} relay node. The algorithm is outlined in Table 2. Namely, we compute the *average-suspicion-index* (*ASI*) as an estimate of the polarity contradiction probability. The PC probability is estimated as the frequency of polarity contradiction occurrences here. Each relay sends *n* bits for a burst. For each bit transmission, polarity contraction can be obtained. Averaging them for *n* trials, we have an estimate.

For each bit transmission, say the i^{th}, the channel LLR and the extrinsic LLR for the l^{th} relay node, $LLRc^{[l]}$ and $LLRe^{[l]}$, can be compared to see if their polarities contradict with each other. Here we use the superscript notation $[l]$ as the index for the l^{th} relay node. We use *instantaneous suspicion index*, denoted as $ISI_i^{[l]}$, to record the presence of contradiction for the i^{th} bit of l^{th} relay node, i.e., it is 1 if the two contradict and 0 otherwise. Averaging $ISI_i^{[l]}$ over a total of *n* bits, we obtain the $ASI^{[l]}$ for the l^{th} relay node. This calculation can be repeated for each relay, i.e., $l = 1, 2, \cdots, L$. As n increases, the estimate becomes more accurate.

One more observation we can make from (5.3) is that when the network is operating at a decent channel SNR, $P(LLRc_H = 1)$ is approximately equal to $P(a = 1)$. The probabilities $P(LLRc_H = 1 | a = 0)$ and $P(LLRc_H = 1 | a = 1)$ are close to 0 and 1 respectively. Therefore, we see that, via (5.2) and (5.3), $ASI^{[l]}$ can serve as an estimation of the attack probability for the l^{th} relay node, i.e., $P(a = 1)^{[l]}$. Summarizing observation made so far, we have:

1. $ASI^{[l]}$ is an estimate of attack probability for the *l*-th relay node.
2. When a relay node is under attack with a significant attack probability, its *ASI* deviates significantly from the baseline.

Table 2. The attack detection method for WMA-networks

I. For *i*th bit, $i = 1, 2, \cdots, n$,

1. Make hard decisions on $LLRc$ and $LLRe$ for the l^{th} relay nodes for all l, i.e.,

$$LLRc_H^{[l]} = \begin{cases} 0, \text{ if } LLRc^{[l]} < 0 \\ 1, \text{ if } LLRc^{[l]} > 0 \end{cases}, \forall l$$

$$LLRe_H^{[l]} = \begin{cases} 0, \text{ if } LLRe_H^{[l]} < 0 \\ 1, \text{ if } LLRe_H^{[l]} > 0 \end{cases}, \forall l$$

2. Compute the *instant suspicion index* (*ISI*) for the l^{th} relay nodes for all l by

$$ISI_i^{[l]} = LLRc_H^{[l]} \oplus LLRe_H^{[l]} \quad, \forall l$$

II. Obtain the *average suspicion index* (*ASI*) by averaging ISI over *n ISI*s:

$$ASI^{[l]} = \frac{\sum_{i=1}^{n} ISI_i^{[l]}}{n} \quad, \forall l$$

<u>*Method for Reversing Attacks*</u>: If the attacked relay node can be identified successfully, could we use that information to reverse the action of attack? We aim to answer this question now. Namely, if the identification of attacked relays is possible, why don't we take an active step and try to correct the compromised messages. This is an attempt to reverse the effect of the flip attacks and restore the original performance of the WMA network code. This is possible. It can be approached in two slightly different ways. The first approach is to discard the channel LLR information from attacked relays; the second is to adjust the channel LLR information from those relay nodes identified as under attack. The first response is simply setting $LLRc^{[l]} = 0$, for all

those relay nodes which are identified as under attack. The second approach is to correct their channel LLRs by the following operation

$$\begin{cases} \hat{L}^{[l]} = LLRc^{[l]} - 2(4E_b / N_0), & LLRc^{[l]} \geq 0 \\ \hat{L}^{[l]} = LLRc^{[l]} + 2(4E_b / N_0), & LLRc^{[l]} < 0 \end{cases}. \tag{5.5}$$

Since the magnitude of the mean of the channel LLRs is $4E_b / N_0$, this operation is equivalent to flip the polarity of the bit sent by the relay. It should be noticed that this treatment will be more effective when the attack probability $P(a=1)$ is high. Simulation results for both the treatments are provided in section 5.4.

5.4 SIMULATION RESULTS AND DISCUSSION

The simulation is based on 100 source and 100 relay nodes. Each relay node is able to check with 5 source nodes. We assume that all source nodes and relay nodes have equal distance to the access node such that the noise variances of the channels are the same. This corresponds to the case where source nodes and relay nodes are located in the vicinity of each other and are all sufficiently far from the destination.

First, we set the attack probability equal to 1, i.e., $P(a=1)=1$, and investigate the bit error rate (BER) performance of the WMA network code as we vary *attack density*. The attack density is defined as the percentage of relay nodes under attack. We tried 0%, 5% and 15%. The results are shown in Figure 26 (see the "raw" curves). We note that the performance degrades dramatically as attack density is increased. With 0% attack density, 10^{-4} BER is obtained at 4dB E_b/N_o. BER is increased to 10^{-2} and 10^{-1} at 4dB E_b/N_o for attack density of 5% and 15%

respectively. In fact, with 15% attack density, the BER curve forms an irreducible error floor at 10^{-1}.

Figure 27 (a)-(f) show the *ASI*'s averaged over 1000 samples for 5% and 15% attack density. We tried 0.3, 0.5, and 1.0 attack probability respectively. In simulation, the outputs from 1st to 5th relay node are flipped (see Figure 27 (a), (c), and (e)) and that from 1st to 15th relay node are flipped (see Figure 27 (b), (d), and (f)). We tested 0dB and 5dB SNR in each case. Note that in general the *ASI*'s are large for the attacked relay nodes but small for the un-attacked relay nodes. As discussed in section 5.3, we expect that, for the 5dB curves, the *ASI*'s of the attacked relay nodes in Figure 27 (a) and (b) are approximate to the attack probability 0.3. Similarly, the *ASI*'s of the attacked relay nodes (for the 5 dB curves) in Figure 27 (c) and (d) are approximate to the attack probability 0.5. The attack probability 1.0 is well captured in Figure 27 (e), but not in Figure 27 (f). We note that, in Figure 27 (f), the baseline is significantly higher. This suggests that the WMA network code is not decodable to a low error-rate when the attack density is 15% with 1.0 attack probability. In each of the cases (a) through (f), although the gaps between the *ASI*'s of attacked relay nodes and those of un-attacked relay nodes at 0dB SNR are smaller than that at 5dB SNR, the gaps are still clearly visible. This shows the robustness of the attack detection algorithm which can provide a decent identification capability even at the low SNR of 0dB.

Now, we show the restored performance after the attack detection is done. We tried two sample sizes, n = 100 and n = 1000 for the two proposed attack reversing methods: (i) discarding the LLR information (see the "w/E" curves in Figure 26), and (ii) correcting the LLR information (see the "w/C" curves in Figure 26). It is clear that both methods can restore the performance significantly for every attack density we tried. We note that both the curves "5%

w/C” overlap each other. Both the curves “5% w/E” also overlap each other. This implies that averaging over 100 samples (*n* = 100) is good enough for 5% attack density. On the other hand, averaging over 1000 samples (*n* = 1000) provides significant performance improvement when attack density is 15% (see both the curves “15% w/C” and both the curves “15% w/E”). This is expected because it is more difficult to accurately identify the under attack relay nodes when the attack density increases – the graph code becomes more vulnerable. Besides, *erasing* is more effective than *correcting* the LLR information from attacked relay nodes. We note that erasing the LLR information effectively results in a new graph code with a higher network code rate. On the other hand, the correction mechanism involves hard-decision process and hence results in a sub optimal performance.

5.5 CONCLUSION

We show that it is effective to identify attacked relay nodes by simply investigating the level of contradiction between the hard decision LLRs. After suspicious relay nodes are identified, the attack actions can be reversed by discarding the un-trustable messages from these relay nodes. Our proposed method is simple but robust. The graph coded wireless multiple access relay networks not only provide an enhancement in error-performance, but also provide an opportunity to verify the data integrity for messages from relay nodes. Further considerations such as optimal threshold for identifying (un)attacked relay nodes will be carried out in the future.

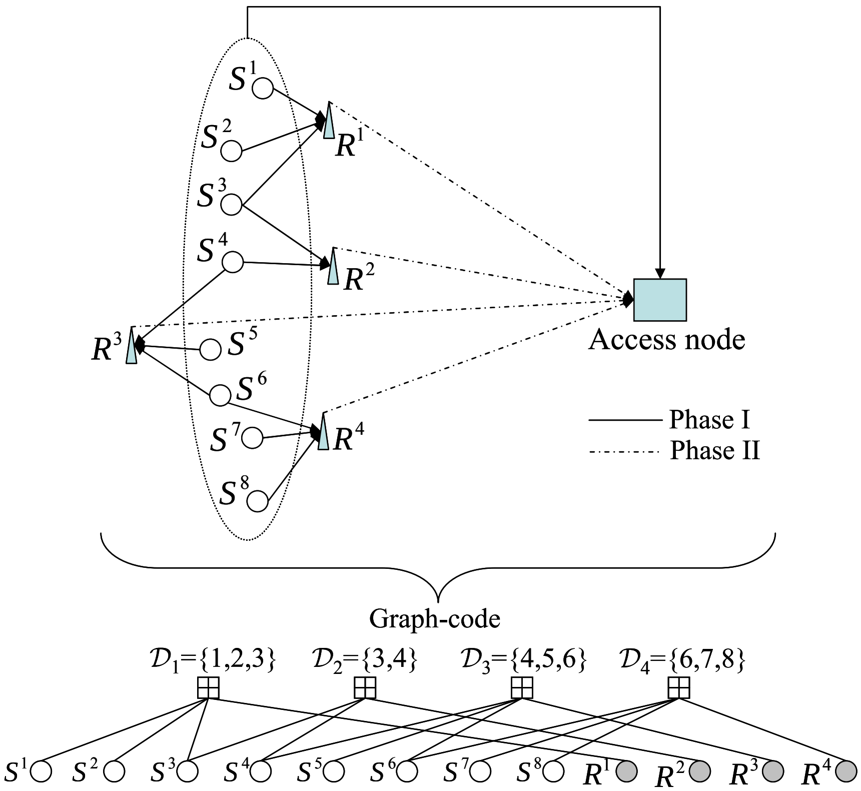


Figure 25. Example of the WMA-network and its corresponding *graph-code*.

In the figure, $s^1 \cdots s^8$ are source nodes and $R^1 \cdots R^4$ are relay nodes. The graph-code is obtained through the two phase transmission protocol.

(*The upper part of the figure is modified from* [46])

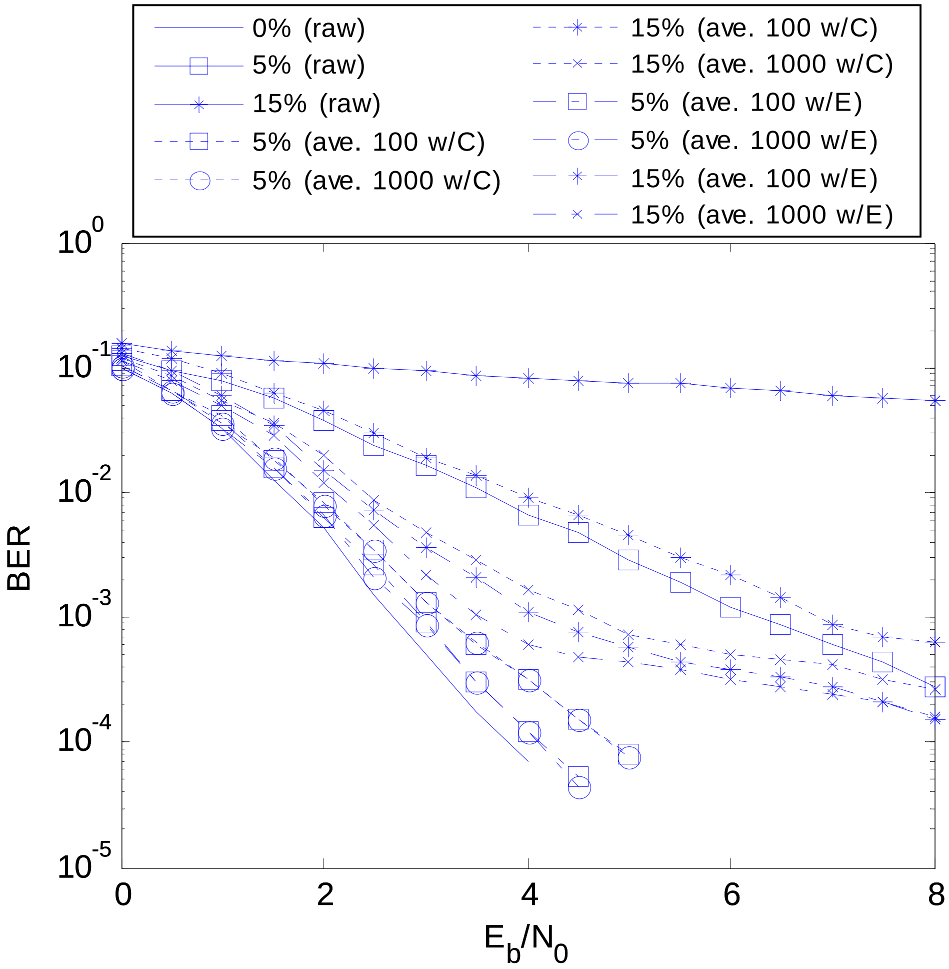


Figure 26. BER performance of the WMA network code under different attack densities.

We show the BER curves with and without the restoring methods. Averaging over 100 and 1000 samples are performed for both restoring methods. In the network, 100 source nodes and 100 relay nodes are simulated.

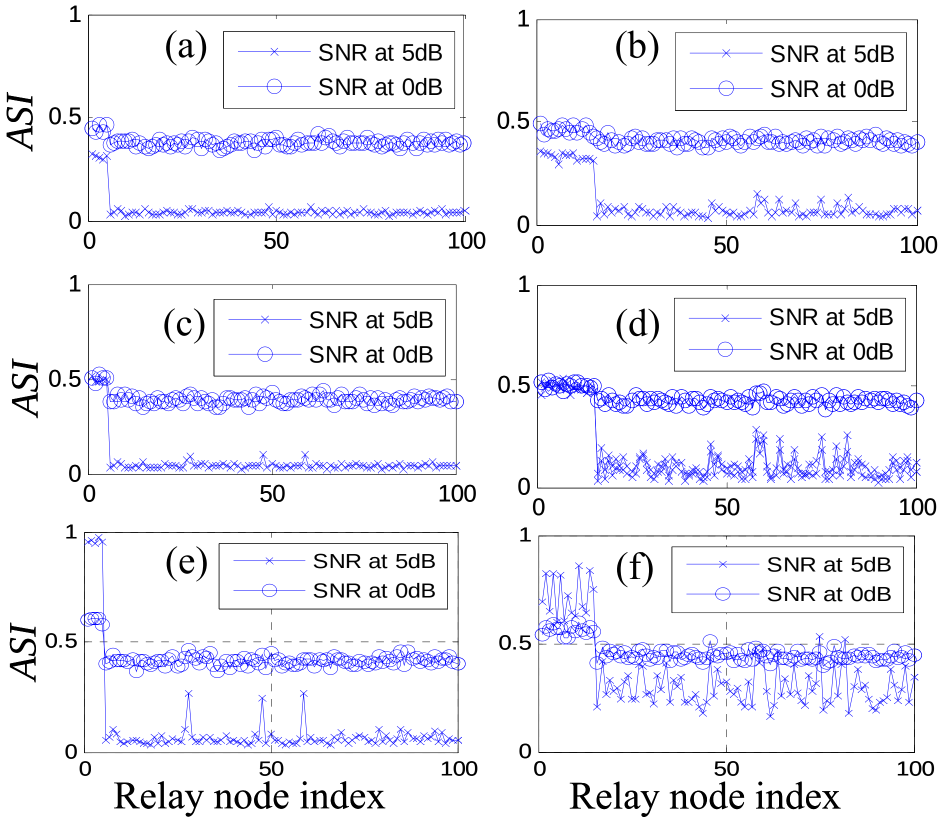


Figure 27. *Average suspicious index* under different attack densities and attack probabilities.

(a) 5% attack density with 0.3 attack probability, (b) 15% attack density with 0.3 attack probability, (c) 5% attack density with 0.5 attack probability, (d) 15% attack density with 0.5 attack probability, (e) 5% attack density with 1.0 attack probability, and (f) 15% attack density with 1.0 attack probability. All *ASI* realizations are obtained by averaging over 1000 samples for both 0dB SNR and 5dB SNR.

6.0 USER-SATISFACTION BASED BANDWIDTH ALLOCATION FOR TRANSMISSION OF MULTIPLE SOURCES OF DATA

In this chapter, we study the bandwidth allocation for multiple sources of data transmitted over a single communication medium. We aim to maximize the overall user satisfaction in data transmission, and formulate an optimization problem for the bandwidth allocation. Using either the logarithmic or exponential form of satisfaction function, we are able to derive closed-form solutions for the optimization problem. We show that the optimal bandwidth allocation for each type of data is piecewise linear with respect to the total available bandwidth.

6.1 INTRODUCTION

In recent years, more and more communication systems involve simultaneous transmission of multiple sources of data over a single communication medium. For example, in a telesurgery system, streams of video, audio, and haptic data need to be sent from a field hospital to a remote surgeon via a packet-switched network or a dedicated satellite link. Each type of data demands a certain range of transmission rate. This might create conflicts among these data when the available bandwidth is limited. So as to achieve the best overall QoS (quality of service) or user satisfaction, it is desirable to optimize bandwidth allocation for different types of data. Moreover, since the available bandwidth and quality of communication may vary significantly from time to time, it is important that the optimal solutions for bandwidth allocation are obtained and implemented in real time.

User-satisfaction based bandwidth allocation has been investigated in several studies (e.g., [55][56][57][58]). These studies have quantified the QoS based on user-satisfaction oriented models, and formulated the bandwidth allocation for multiple sources of data into optimization problems. However, solving these optimization problems is not so straightforward. Although necessary conditions have been derived for the optimal strategy of bandwidth allocation [58], an explicit form of the solution is still not available. Miao and Niu have proposed to use Rosen's gradient project method to find the optimal strategy [55][56][59], but their method requires iterative searching and thus cannot provide closed-form solutions.

In this chapter, we aim to solve the optimization problem for bandwidth allocation that maximizes the overall user satisfaction. We model the user satisfaction as a weighted sum of the satisfactory functions for individual types of data. Researchers have already used logarithmic functions (e.g., [60][61][62]) or exponential functions (e.g., [63]) to characterize human perceptual satisfaction. Utilizing these forms of satisfaction functions, we are able to derive closed-form solutions for the bandwidth allocation problem (Section 6.3). Moreover, we show that the optimal bandwidth allocation for each type of data is *piecewise linear* with respect to the total available bandwidth (Section 6.3). This result allows us to calculate the optimal bandwidth allocation immediately (in real time) without replying to the conventional iterative updates. In the situations when the parameters of the satisfaction functions or weights of priorities for the data types are unknown, we provide strategies to characterize these parameters using data from human experiments (Section 6.4). We show that the piecewise-linear property of the optimal solution enables closed-form expressions for parameter estimation (Section 6.5).

6.2 PROBLEM FORMULATION

The problem of bandwidth allocation for multiple sources of data can be formulated into an optimization problem:

$$\begin{aligned} &\text{maximize } \sum_{i=1}^{N} w_i I_i(r_i) \\ &\text{subject to } \sum_{i=1}^{N} r_i \le R \text{ and} \qquad (6.1) \\ &r_{i,\min} \le r_i \le r_{i,\max} \text{ for } i = 1,\ldots,N. \end{aligned}$$

The notations in the above problem are explained as follows. We use w_i to represent the weights of importance or priority for different types of data, where the subscript i is the index for a data type (e.g., video, audio, or haptic data) and N is the total number of data types under consideration. We use r_i to denote the bandwidth or transmission rate allocated for the i-th type of data. The rate r_i is bounded from below by $r_{i,\min}$ and from above by $r_{i,\max}$, where (i) $r_{i,\min}$ is the minimum perception-quality requirement for the i-th type of data ($r_{i,\min}$ can also be the minimum required rate for the encoder/decoder of the i-th type of data to work) and (ii) $r_{i,\max}$ is the maximum rate at which the quality of the i-th type of data can be fully satisfied. Since all the sources of data are simultaneously transmitted over a common communication medium, they are subject to the total available bandwidth of the medium, denoted R. We use $I_i(\cdot)$ to represent the satisfaction function of the i-th data type. It is a monotonic increasing and concave function of the data transmission rate. Following [60][61][62][63], we utilize two forms of satisfaction functions for $I_i(\cdot)$: the logarithmic and exponential functions (see Section 6.3).

The optimization problem (6.1) is easy to solve for two special cases of R (the total available bandwidth): (i) $R < R_{\min}$ and (ii) $R > R_{\max}$, where $R_{\min}$ and $R_{\max}$ denote $\sum_{i=1}^{N} r_{i,\min}$ and

$\sum_{i=1}^{N} r_{i,\max}$, respectively. For case (i), (6.1) has no feasible solution. For case (ii), the available bandwidth allows all types of data to be transmitted at their maximum rates, and hence the optimal solution is $r_i = r_{i,\max}$ for $i = 1,...,N$.

Therefore, in the rest of the chapter, we only need to consider the value of *R* within the range of $R_{\min} \le R \le R_{\max}$. For this range of *R*, it can be seen that the optimal solution of (6.1) must satisfy the equality in $\sum_{i=1}^{N} r_i \le R$ (making full use of the bandwidth), and thus the first constraint of (6.1) is reduced to

$$\sum_{i=1}^{N} r_i = R. \tag{6.2}$$

6.3 OPTIMAL BANDWIDTH ALLOCATION

In this section, we consider two forms of satisfaction functions, the logarithmic and exponential functions, to characterize the human perceptual satisfaction in presence of different transmission rates of a specific type of data. We aim to derive closed-form solutions for the optimal bandwidth allocation.

A. Using Logarithmic Satisfaction Function: Consider the following form of satisfaction function

$$I_i(r_i) = \alpha_i \ln \frac{r_i - \beta_i}{\gamma_i} \tag{6.3}$$

where α_i, β_i, and γ_i are the parameters characterizing the shape of the satisfaction function for the *i*-th type of data transmitted within the range of $r_{i,\min} \le r_i \le r_{i,\max}$.

To solve (6.1) with the equality constraint (6.2), we construct the Lagrangian function

$$\begin{aligned} L &= \sum_{i=1}^{N} w_i I_i(r_i) \\ &+ \lambda(\sum_{i=1}^{N} r_i - R) + \sum_{i=1}^{N} \mu_i (r_{i,\min} - r_i) + \sum_{i=1}^{N} \upsilon_i (r_i - r_{i,\max}). \end{aligned} \tag{6.4}$$

Let r_i^*, $i = 1,...,N$, be the optimal solution to (6.1) for a value of R satisfying $R_{\min} \le R \le R_{\max}$. According to the Kuhn-Tucker conditions [64], there exist unique Lagrange multipliers λ^*, μ_i^*, and υ_i^*, $i = 1,...,N$, such that

$$\left.\frac{\partial L}{\partial r_i}\right|_{r_i = r_i^*} = \frac{w_i \alpha_i}{r_i^* - \beta_i} + \lambda^* - \mu_i^* + \upsilon_i^* = 0, \;\; i = 1,...,N \tag{6.5a}$$

$$\sum_{i=1}^{N} r_i^* - R = 0 \tag{6.5b}$$

$$\mu_i^* \le 0 \;\; \text{for} \;\; i \in \mathcal{K} \equiv \{ i \,|\, r_{i,\min} - r_i^* = 0, \; i = 1,...,N \} \tag{6.5c}$$

$$\mu_i^* = 0 \;\; \text{for} \;\; i \notin \mathcal{K} \tag{6.5d}$$

$$\upsilon_i^* \le 0 \;\; \text{for} \;\; i \in \mathcal{Q} \equiv \{ i \,|\, r_i^* - r_{i,\max} = 0, \; i = 1,...,N \} \tag{6.5e}$$

$$\upsilon_i^* = 0 \;\; \text{for} \;\; i \notin \mathcal{Q} \tag{6.5f}$$

Consider the following two cases.

Case 1: Both $\mathcal{K}$ and $\mathcal{Q}$ [defined in (6.5c) and (6.5e), respectively] are empty for the optimal solution. That is, $r_i^* > r_{i,\min}$ and $r_i^* < r_{i,\max}$ for $i = 1,...,N$. From (6.5d) and (6.5f), we see $\mu_i^* = 0$ and $\upsilon_i^* = 0$ for any *i*. Therefore, (6.5a) becomes

$$\frac{w_i \alpha_i}{r_i^* - \beta_i} + \lambda^* = 0, \;\; i = 1,...,N\,. \tag{6.6}$$

Using (6.5b) and (6.6), we can find the optimal solution:

$$r_i^* = \frac{w_i\alpha_i}{\sum_{j=1}^{N}(w_j\alpha_j)} R - \frac{w_i\alpha_i \sum_{j=1}^{N}\beta_j}{\sum_{j=1}^{N}(w_j\alpha_j)} + \beta_i,\ \ i = 1,\ldots,N\ . \tag{6.7}$$

The above expression can also be written as

$$r_i^* = c_i + d_i R \tag{6.8}$$

where

$$d_i = w_i\alpha_i / \textstyle\sum_{j=1}^{N}(w_j\alpha_j) \tag{6.9a}$$

$$c_i = \beta_i - d_i\left(\textstyle\sum_{j=1}^{N}\beta_j\right). \tag{6.9b}$$

It can be seen that $0 < d_i < 1$, $\sum_{i=1}^{N} d_i = 1$, and $\sum_{i=1}^{N} c_i = 0$.

Equations (6.7) and (6.8) clearly show that r_i^*, the optimal bandwidth allocation for the *i*-th type of data, is a *linear function* of the total available bandwidth *R* as long as the condition of Case 1 is satisfied.

Case 2: $\mathcal{K} \cup \mathcal{Q}$ is nonempty for the optimal solution. Then (6.5a) becomes

$$\frac{w_i\alpha_i}{r_i^* - \beta_i} + \lambda^* = 0 \ \text{ for } \ i \notin \mathcal{K} \cup \mathcal{Q} \tag{6.10a}$$

and (6.5b) becomes

$$\sum_{i \notin \mathcal{K}\cup\mathcal{Q}} r_i^* - \left(R - \sum_{j\in\mathcal{K}} r_{j,\min} - \sum_{k\in\mathcal{Q}} r_{k,\max} \right) = 0\ . \tag{6.10b}$$

By using (6.10a) and (6.10b), the optimal bandwidth r_i^* allocated for the *i*-th data type can be calculated:

$$r_i^* = \begin{cases} r_{i,\min} & \text{for} \quad i \in \mathcal{K} \\ r_{i,\max} & \text{for} \quad i \in \mathcal{Q} \\ c_i + d_i\left(R - \sum_{j \in \mathcal{K}} r_{j,\min} - \sum_{k \in \mathcal{Q}} r_{k,\max}\right) & \text{for} \quad i \notin \mathcal{K} \cup \mathcal{Q} \end{cases} \tag{6.11}$$

where

$$d_i = w_i \alpha_i / \sum_{j \notin \mathcal{K} \cup \mathcal{Q}} (w_j \alpha_j) \tag{6.12a}$$

$$c_i = \beta_i - d_i \left(\sum_{j \notin \mathcal{K} \cup \mathcal{Q}} \beta_j \right). \qquad (\tag{6.12b}$$

It can be seen that $0 < d_i < 1$, $\sum_{i \notin \mathcal{K} \cup \mathcal{Q}} d_i = 1$, and $\sum_{i \notin \mathcal{K} \cup \mathcal{Q}} c_i = 0$.

Equation (6.1) shows that r_i^*, the optimal bandwidth allocation for the *i*-th type of data, is a *linear function* of the total available bandwidth *R* as long as $\mathcal{K}$ and $\mathcal{Q}$ remain unchanged. Equation (6.7) or (6.8) can be viewed as a special case of (6.11) when both $\mathcal{K}$ and $\mathcal{Q}$ are empty.

Summarizing the observations from the above two cases, we may conclude that the optimal bandwidth allocation for each type of data should be a *piecewise linear* function of *R*. Further discussion about this will be given in Section 6.3-C.

B. Using Exponential Satisfaction Function: Consider the following form of satisfaction function

$$I_i(r_i) = \delta_i \left[1 - e^{-\alpha_i (r_i - \beta_i)} \right] + \gamma_i \tag{6.13}$$

where α_i, β_i, γ_i, and δ_i are the parameters characterizing the shape of the satisfaction function for the *i*-th type of data transmitted within the range of $r_{i,\min} \le r_i \le r_{i,\max}$.

So as to solve the bandwidth allocation problem given (6.3), we follow the similar steps as described in Section 6.3-A. Constructing the Lagrangian function in the same form of (6.4), we may derive

$$\left.\frac{\partial L}{\partial r_i}\right|_{r_i=r_i^*} = \frac{w_i\delta_i\alpha_i}{e^{\alpha_i(r_i^*-\beta_i)}} + \lambda^* - \mu_i^* + \upsilon_i^* = 0,\ \ i=1,\ldots,N \tag{6.14}$$

together with (6.5b)-(6.5f). Following the similar derivation as provided in Section 6.3-A, we can obtain

$$r_i^* = \begin{cases} r_{i,\min} & \text{for} \quad i \in \mathcal{K} \\ r_{i,\max} & \text{for} \quad i \in \mathcal{Q} \\ c_i + d_i\left(R - \sum\limits_{j\in\mathcal{K}} r_{j,\min} - \sum\limits_{k\in\mathcal{Q}} r_{k,\max}\right) & \text{for} \quad i \notin \mathcal{K}\cup\mathcal{Q} \end{cases} \tag{6.15}$$

where $\mathcal{K}$ and $\mathcal{Q}$ are defined in (6.5c) and (6.5e), respectively, and

$$c_i = \frac{1}{\alpha_i}\ln(w_i\delta_i\alpha_i) - \frac{\sum\limits_{j\notin\mathcal{K}\cup\mathcal{Q}}\left\{\left[\ln(w_j\delta_j\alpha_j)\right]/\alpha_j + \beta_j\right\}}{\alpha_i\sum\limits_{j\notin\mathcal{K}\cup\mathcal{Q}} 1/\alpha_j} \tag{6.16a}$$

$$d_i = \frac{1}{\alpha_i\sum\limits_{j\notin\mathcal{K}\cup\mathcal{Q}} 1/\alpha_j}. \tag{6.16b}$$

It can be seen that $0 < d_i < 1$, $\sum_{i\notin\mathcal{K}\cup\mathcal{Q}} d_i = 1$, and $\sum_{i\notin\mathcal{K}\cup\mathcal{Q}} c_i = -\sum_{i\notin\mathcal{K}\cup\mathcal{Q}} \beta_i$.

Similar to the results in Section 6.3-A, the optimal bandwidth allocation derived here is also a piecewise linear function of *R*, as demonstrated in (6.15).

C. Piecewise Linear Solutions: From the results obtained in Sections 6.3-A and III-B, we have observed piecewise linear dependency of the optimal bandwidth allocation on the total available bandwidth *R*. Here we provide further illustrations.

Figure 28(a) shows the piecewise linear "trajectories" for the optimal bandwidth allocation in the case of three sources of data. Let us first consider the range of R within which the optimal solution to (6.1), r_i, satisfies $r_{i,\min} < r_i < r_{i,\max}$ for all the three data types ($i = 1, 2$, and 3). In this range of R [depicted in region (R3) of Figure 28(a)], the optimal trajectory of r_i is a strictly increasing line for each data type, according to (6.8) where $0 < d_i < 1$ for any i. Within this region, r_i decreases as R decreases. Without loss of generality, let us assume r_1 hits $r_{1,\min}$ first as R decreases. When R further decreases from there, now in region (R2) of Figure 28(a), r_1 remains at $r_{1,\min}$, and r_2 and r_3 decrease linearly with R and share a bandwidth of $R - r_{1,\min}$. This continues until r_2 hits $r_{2,\min}$ or r_3 hits $r_{3,\min}$. Without loss of generality, assume r_2 hits $r_{2,\min}$ first. Now if R further decreases [in region (R1) of Figure 28(a)], r_1 and r_2 will remain at $r_{1,\min}$ and $r_{2,\min}$, respectively, and all the rest of bandwidth $R - r_{1,\min} - r_{2,\min}$ will be allocated to r_3. When R is less than $R_{\min} = r_{1,\min} + r_{2,\min} + r_{3,\min}$, the optimization problem (6.1) has no solution. In the other direction, when we increase the value of R from region (R3) to regions (R4) and (R5) in Figure 28(a), similar characteristics of piecewise linearity may be observed in the optimal trajectories of r_i. It can be verified that the number of regions of linearity, where the slopes of the optimal trajectories of r_i stay constant, should be no more than $2N - 1$ (N is the total number of data types under consideration).

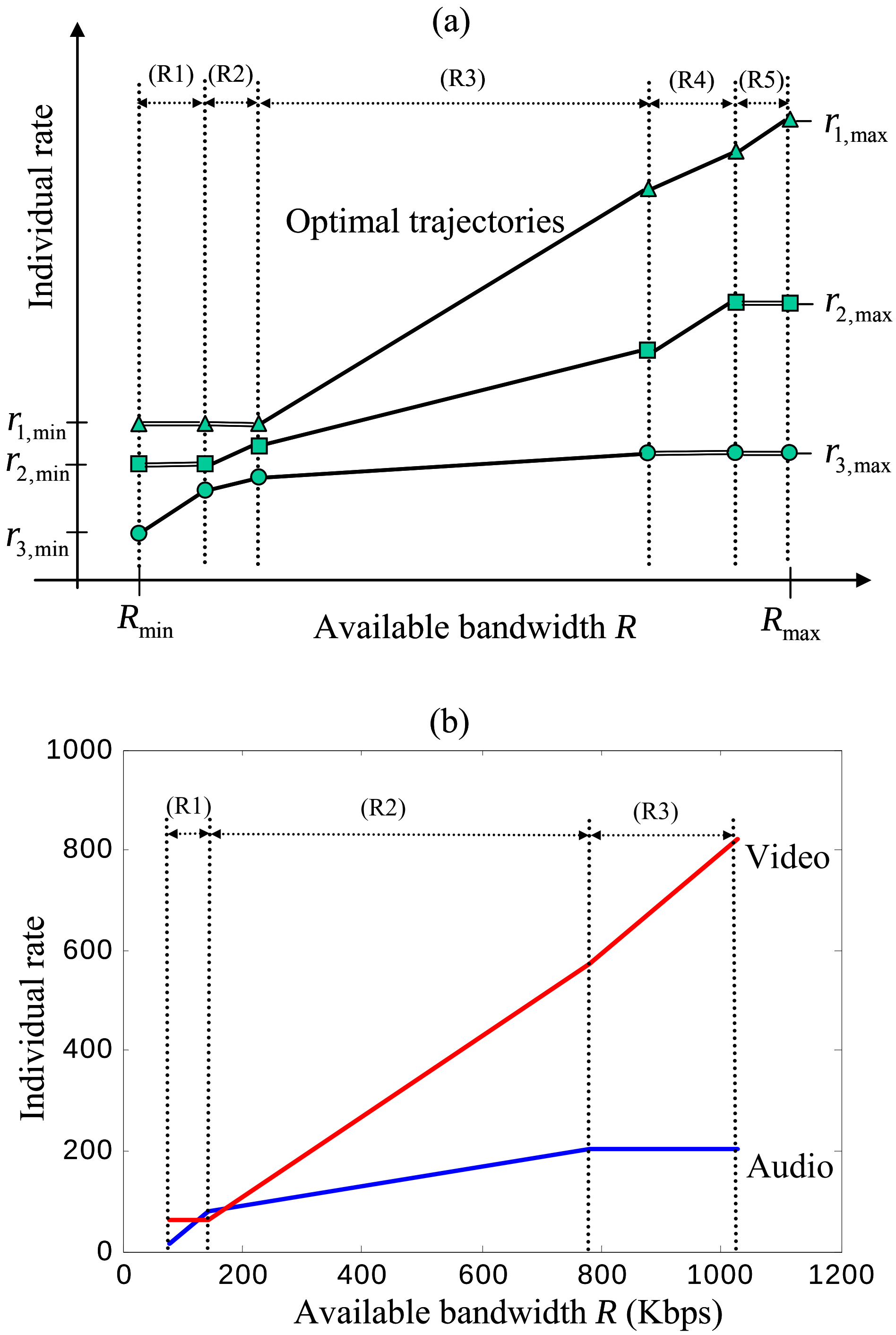


Figure 28. The optimal distribution of bandwidth allocation for a given available bandwidth.

(a) Illustration of the piecewise linear solution for the optimal bandwidth allocation. Within each indicated range of R, denoted by (R1), (R2), etc., the optimal bandwidth allocation is linear with respect to R. (b) A toy example, where only video and audio data are involved.

Figure 28(b) presents a toy example, where only video and audio data are involved. These two types of data are given the same priority ($w_1 = w_2$), and their satisfaction functions are chosen as $I_1(r_1) = 1 - e^{-0.0223 r_1}$ (r_1 ranging from 16 Kbps to 206 Kbps) for the audio data and $I_2(r_2) = 1 - e^{-0.0056 r_2}$ (r_2 ranging from 64 Kbps to 822 Kbps) for the video data, respectively. Figure 28(b) demonstrates the piecewise linear dependency of the optimal solution on the total available bandwidth *R*. There are three regions of linearity. In region (R2), the optimal trajectories of r_i with respect to *R* are strictly increasing lines for both the audio and video data. In this region, incremental bandwidth will be allocated to the two types of data in proportion. In region (R1), the optimal strategy is to allocate bandwidth to the audio data as much as possible while keeping the minimally required bandwidth for the video data. In this region, increasing bandwidth of the audio data tends to create more satisfaction than increasing bandwidth of the video data by the same amount. On the other hand, in region (R3), the optimal strategy is to “sacrifice” the audio data, because the bandwidth allocated to it, r_1, already reaches $r_{1,\max}$.

6.4 TRAJECTORY FITTING FOR THE OPTIMAL SOLUTION

In Section 6.3, we derived closed-form solutions for the optimal bandwidth allocation for multiple sources of data. However, the calculation cannot be completed if the parameters of the satisfaction functions or weights of priorities for different types of data are not available. To deal with this situation, we consider using data from human experiments. The purpose of the experiments is to obtain samples of human decisions on the best combinations of bandwidths allocated for multiple sources of data for some given values of *R*. However, in this section, our focus is not to design or run the human experiments, but to propose a strategy of using the experimental data (assumed to be available already) to fit the optimal solutions—the optimal trajectories of r_i with respect to R.

Suppose we acquired the data $R^{[k]}$ and $r_i^{[k]}$ ($k = 1,\ldots,K$; $i = 1,\ldots,N$) from some perception experiments tested on a human subject, where: (i) $R^{[k]}$, $k = 1,\ldots,K$, is a set of K values of the total available bandwidth tested in the experiments; (ii) for each $R^{[k]}$, the bandwidth combination $\{r_1^{[k]},\ldots,r_N^{[k]}\}$ is what the human subject decides to be the best among a set of candidate combinations of bandwidths allocated for the N sources of data. We further assume that these data samples belong to the same region of linearity, e.g., region (R3) in Figure 28(a). (If not, we can pick up those samples that belong to the same region. The regions of linearity should be visually identifiable due to the piecewise linearity of the optimal trajectories of r_i.) In the following, we try to find the optimal trajectories of r_i that fit the experimental data best.

We may consider two approaches. The first approach is to find the parameters w_i, α_i, β_i, γ_i, and δ_i that minimize $\sum_{k=1}^{K}\sum_{i=1}^{N}\left(r_i^{[k]} - r_i^*\right)^2$, where r_i^* is a function of these parameters and is determined by (6.1)-(6.2b) or (6.5)-(6.6b). However, it is very difficult to obtain explicit forms of the parameters via this approach.

The second approach is to find the parameters c_i and d_i [see (6.1) and (6.5)] that minimize $\sum_{k=1}^{K}\sum_{i=1}^{N}\left(r_i^{[k]} - r_i^*\right)^2$. Based on the property that r_i^* is a linear function of c_i and d_i, we can derive closed-form expressions for these parameters, which suffice to determine the optimal trajectories of r_i.

Without loss of generality, in the following we only consider the situation where $\mathcal{K} \cup \mathcal{Q}$ is empty, i.e., $r_{i,\min} < r_i < r_{i,\max}$ for any $i = 1,\ldots,N$. For the other situations involving nonempty $\mathcal{K} \cup \mathcal{Q}$, the procedure to fit c_i and d_i is similar.

If we use logarithmic satisfaction functions, the problem is to find c_i and d_i that minimize $\sum_{k=1}^{K}\sum_{i=1}^{N}\left(r_i^{[k]}-c_i-d_iR^{[k]}\right)^2$ subject to $\sum_{i=1}^{N}d_i=1$ and $\sum_{i=1}^{N}c_i=0$. The closed-form solution is

$$c_i=\frac{\left(\sum_{k=1}^{K}r_i^{[k]}-\eta\right)\left(\sum_{k=1}^{K}(R^{[k]})^2\right)-\left(\sum_{k=1}^{K}R^{[k]}\right)\left(\sum_{k=1}^{K}r_i^{[k]}R^{[k]}-\xi\right)}{K\sum_{k=1}^{K}(R^{[k]})^2-\left(\sum_{k=1}^{K}R^{[k]}\right)^2} \tag{6.17}$$

$$d_i=\frac{K\left(\sum_{k=1}^{K}r_i^{[k]}R^{[k]}-\xi\right)-\left(\sum_{k=1}^{K}r_i^{[k]}-\eta\right)\left(\sum_{k=1}^{K}R^{[k]}\right)}{K\sum_{k=1}^{K}(R^{[k]})^2-\left(\sum_{k=1}^{K}R^{[k]}\right)^2} \tag{6.18}$$

where

$$\eta=\frac{1}{N}\left(\sum_{i=1}^{N}\sum_{k=1}^{K}r_i^{[k]}-\sum_{k=1}^{K}R^{[k]}\right)$$

$$\xi=\frac{1}{N}\left(\sum_{i=1}^{N}\sum_{k=1}^{K}r_i^{[k]}R^{[k]}-\sum_{k=1}^{K}(R^{[k]})^2\right).$$

If using the exponential satisfaction function, we need to consider the constraint $\sum_{i=1}^{N}c_i=-\sum_{i=1}^{N}\beta_i$, the right-hand side of which is an unknown quantity. For curve fitting purpose, we ignore this constraint, and only consider the constraint $\sum_{i=1}^{N}d_i=1$ when minimizing $\sum_{k=1}^{K}\sum_{i=1}^{N}\left(r_i^{[k]}-c_i-d_iR^{[k]}\right)^2$. We can derive

$$c_i=\frac{\left(\sum_{k=1}^{K}r_i^{[k]}\right)\left(\sum_{k=1}^{K}(R^{[k]})^2\right)-\left(\sum_{k=1}^{K}R^{[k]}\right)\left(\sum_{k=1}^{K}r_i^{[k]}R^{[k]}-\xi\right)}{K\sum_{k=1}^{K}(R^{[k]})^2-\left(\sum_{k=1}^{K}R^{[k]}\right)^2} \tag{6.19}$$

$$d_i=\frac{K\left(\sum_{k=1}^{K}r_i^{[k]}R^{[k]}-\xi\right)-\left(\sum_{k=1}^{K}r_i^{[k]}\right)\left(\sum_{k=1}^{K}R^{[k]}\right)}{K\sum_{k=1}^{K}(R^{[k]})^2-\left(\sum_{k=1}^{K}R^{[k]}\right)^2} \tag{6.20}$$

where

$$\xi = \frac{1}{N}\left(\sum_{i=1}^{N}\sum_{k=1}^{K} r_i^{[k]} R^{[k]} - \sum_{k=1}^{K}(R^{[k]})^2 \ - \frac{1}{K}(\sum_{k=1}^{K} R^{[k]})[\sum_{i=1}^{N}\sum_{k=1}^{K} r_i^{[k]} - \sum_{k=1}^{K} R^{[k]}]\right).$$

6.5 CONCLUSION

In this chapter we studied the maximization of user satisfaction in the transmission of multiple sources of data over a single communication medium. We derived closed-form solutions for optimal bandwidth allocation for different types of data, using either logarithmic or exponential satisfaction functions. This optimal solution piecewise-linearly depends on the total available bandwidth, and can be easily calculated in real time. When the parameters of the satisfaction functions or priorities of the data types are not totally unknown, we provided strategies to characterize these parameters through a set of specifically-designed human experiments.

7.0 ON THE ESTIMATION OF TARGET SPECTRUM FOR FILTER-ARRAY BASED SPECTROMETERS

Miniature spectrometers have been drawn researchers much attention due to its wide variety of possible applications. In this chapter, we show the achievability of a fine spectrometer on-a-chip based on a low-performance, low-cost filter-array. A low quality filter-array is augmented with digital signal processing techniques. A series of estimators for recovering target spectrum is introduced. By exploiting non-negative nature of spectral content, a non-negative least-square algorithm is found particularly useful for spectrum recovery. The concept is verified in a hardware implementation.

7.1 INTRODUCTION

Recently, miniature spectrometers have been drawn researchers great attention. Miniature spectrometers provide solutions to a variety of promising applications in biological, chemical, medical, or pharmaceutical industries, in which small, light-weight, and non-fragile properties of spectrometers are demanded [65][69][70]. Currently, MEMS, CMOS, micro-optic electromechanical systems, or integrated optics technologies are the main means to build a miniature- or micro- spectrometer. Based on the underlying operation principle, spectrometers can be classified such as grating-based, Fourier-transform based, or filter-based [67][68][69].

Filter-based static spectrometers utilize different filter functions to filter spectral energy emanating from a target. Recent literatures [70][71][72] have demonstrated that filter-based spectrometers are capable of high-resolution and allowed to fabricate on-a-chip. By implementing multiple filters and detectors, these spectrometers are avoided to have moving elements, and hence are static and rigid. Furthermore, they have the ability to capture the target spectrum in a very short time. This property is demanded for certain applications especially in biological, biochemical or biomedical industries.

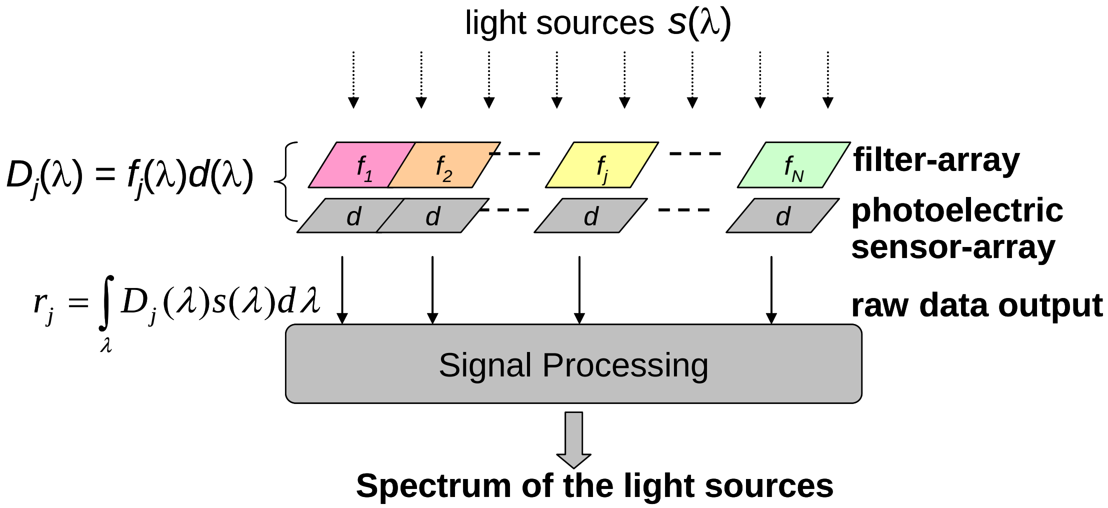
For low-cost fabrication, filters may not have delta-function-like shapes with a narrow range response. The spectrum obtained directly from these filtered results is severely distorted, and hence is unacceptable as it is. However, signal processing techniques can be applied to estimate and restore the target spectrum.

In this chapter, we discuss and provide digital-signal-processing (DSP) methods for filter-array based spectrometers. Based on a discrete linear system model, a series of estimators for spectrum recovery is introduced. By exploiting the non-negative nature of spectral content, we found the non-negative least-squares (NNLS) algorithm particularly useful to estimate and restore the target spectrum with high fidelity. A hardware implementation for the filter-array based spectrometer is demonstrated via a commercialized CCD camera and a DSP board. This prototype shows the achievability of a fine spectrometer on-a-chip based on a low-performance, low-cost filter-array.

Section 7.2 shows the system model of the filter-array based spectrometer. Section 7.3 discusses and provides the estimators for restoring a target spectrum. A hardware implementation and experimental results are shown in section 7.4. Section 7.5 draws the summary and conclusion.

7.2 SYSTEM MODEL

Figure 29 shows the basic system model of the static filter-array based spectrometer. An array of filters is directly placed on top of an array of photoelectric sensors such as CCD sensors. A filter may correspond to a CCD sensor or a group of CCD sensors. The outputs from the CCD sensors are then fed into a digital signal processor. We note that the 1-D structure shown in Figure 29 can be extended to a 2-D structure straightforwardly by placing filters and CCD sensors in a 2-D plane. In this chapter, we will restrict our discussion on the drawn 1-D structure.



light sources $s(\lambda)$
$D_j(\lambda) = f_j(\lambda)d(\lambda)$
f_1
f_2
f_j
f_N
filter-array
d
d
d
d
photoelectric sensor-array
$r_j = \int_\lambda D_j(\lambda)s(\lambda)d\lambda$
raw data output
Signal Processing
Spectrum of the light sources

Figure 29. System structure of the filter-array based spectrometers.

We assume all the CCD sensors have the same sensitivity function, denoted by $d(\lambda)$, where λ is the continuous wavelength. Denote $f_j(\lambda)$ as the filter transmission function for j^{th} filter with respect to wavelength λ (each filter may have a different transmission function). A filter and its corresponding CCD sensors compose a spectral detector. The overall sensitivity function of the j^{th} spectral detector can be expressed as $D_j(\lambda) = f_j(\lambda)d(\lambda)$. For a target with spectral content $s(\lambda)$, the output from the j^{th} detector is $r_j = \int_{\lambda} D_j(\lambda)s(\lambda)d\lambda$.

We consider the system and the target spectrum based on a *discrete* model. The transformation between the target spectrum and the CCD-sensor outputs is associated by the matrix equation

$$\mathbf{r} = \mathbf{Hs} + \mathbf{n}, \tag{7.1}$$

where

$$\mathbf{r} = \begin{bmatrix} r_1 \\ \vdots \\ r_j \\ \vdots \\ r_N \end{bmatrix}, \ \mathbf{H} = \begin{bmatrix} D_1(\lambda_1) & \cdots & D_1(\lambda_M) \\ \vdots & \vdots & \vdots \\ D_j(\lambda_1) & \cdots & D_j(\lambda_M) \\ \vdots & \vdots & \vdots \\ D_N(\lambda_1) & \cdots & D_N(\lambda_M) \end{bmatrix}, \ \mathbf{s} = \begin{bmatrix} s(\lambda_1) \\ s(\lambda_2) \\ \vdots \\ \vdots \\ s(\lambda_M) \end{bmatrix}, \text{ and } \mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ \vdots \\ n_N \end{bmatrix}$$

The dimensionalities of $\mathbf{r}$, $\mathbf{H}$, $\mathbf{s}$, and $\mathbf{n}$ are $N \times 1$, $N \times M$, $M \times 1$, and $N \times 1$, respectively. $\mathbf{r}$ is an observed signal vector whose elements are the outputs of CCD-sensors. The elements in $\mathbf{r}$ can be observed simultaneously via each individual detector. $\mathbf{n}$ is a noise vector. There are N detectors in this model. $\mathbf{H}$ is a detector sensitivity matrix. The M elements in a row of $\mathbf{H}$ matrix represent the sensitivity function of a detector, obtained by evenly sampling the sensitivity function over a certain wavelength range. $\mathbf{s}$ is a source signal vector, whose elements represent the target spectrum evenly sampled in the wavelength domain. The minimum number of samples required for a given target-spectrum can be obtained through the *sampling theorem.* That is, consider the shape of the target-spectrum as a continuous function of a unit interval. If B is the minimum value such that the Fourier transform of the function over the unit interval $S(\theta)$ satisfying $S(\theta) = 0$ for $|\theta| > B$, $2B$ is the minimum number of samples to specify the continuous function [74]. We call these samples *resolved points* in this chapter. The number of resolved points for a given application is critical since it determines the number of required detectors as we will see in section 7.3. In addition, we note that the sensitivity characteristic of the detectors needs not to be a delta-function shape with a narrow range response.

7.3 ESTIMATORS FOR RESTORING TARGET SPECTRAL

Working on the observation vector $\mathbf{r}$, an estimator provides an estimation $\hat{\mathbf{s}}$ of the input spectrum by considering all possible source signal vectors $\mathbf{s}$. One criterion we can use here as the starting point is the *maximum a posteriori* (MAP) rule [74]. The MAP estimator is obtained by maximizing the *posterior* probability, i.e.,

$$\hat{\mathbf{s}}_{MAP} = \arg\max_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{r}) . \tag{7.2}$$

From the Bayes' rule, the *posterior* probability can be written as $P(\mathbf{s} \mid \mathbf{r}) = P(\mathbf{r} \mid \mathbf{s}) P(\mathbf{s}) / P(\mathbf{r})$. When we do not have any information on the source signal such that $P(\mathbf{s})$ is uniformly distributed, the MAP estimator becomes the *maximum likelihood* (ML) estimator. The ML estimator maximizes the likelihood function, i.e.,

$$\hat{\mathbf{s}}_{ML} = \arg\max_{\mathbf{s}} P(\mathbf{r} \mid \mathbf{s}) . \tag{7.3}$$

For the filter-array spectrometer, the observed signal vector $\mathbf{r}$ and the source signal vector $\mathbf{s}$ can be associated by Eq. (7.1) as discussed. Now assume the noise vector $\mathbf{n}$ is multivariate Gaussian with zero mean and covariance matrix $\mathbf{R}_n$, i.e., $E[\mathbf{n}] = \mathbf{0}$, and $E[\mathbf{n}\mathbf{n}^T] = \mathbf{R}_n$, where the superscript T denotes the transpose operation. The ML estimator then is obtained by maximizing the likelihood function

$$P(\mathbf{r}\,|\,\mathbf{s}) = \frac{1}{(2\pi)^{N/2}\left|\mathbf{R}_n\right|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{r}-\mathbf{Hs})^T\mathbf{R}_n^{-1}(\mathbf{r}-\mathbf{Hs})\right]. \tag{7.4}$$

To solve for the estimator, it is equivalent to find the vector $\mathbf{s}$ which minimizes the scale exponent $(\mathbf{r}-\mathbf{Hs})^T\mathbf{R}_n^{-1}(\mathbf{r}-\mathbf{Hs})$. The solution can be found by solving the partial differential equation $\partial(\mathbf{r}-\mathbf{Hs})^T\mathbf{R}_n^{-1}(\mathbf{r}-\mathbf{Hs})/\partial\mathbf{s}=\mathbf{0}$. That is, $\partial(\mathbf{r}^T\mathbf{R}_n^{-1}\mathbf{r}-2\mathbf{r}^T\mathbf{R}_n^{-1}\mathbf{Hs}+\mathbf{s}^T\mathbf{H}^T\mathbf{R}_n^{-1}\mathbf{Hs})/\partial\mathbf{s}= -2\mathbf{H}^T\mathbf{R}_n^{-1}\mathbf{r}+2\mathbf{H}^T\mathbf{R}_n^{-1}\mathbf{H}=\mathbf{0}$. If the matrix $\mathbf{H}^T\mathbf{R}_n^{-1}\mathbf{H}$ is nonsingular (i.e., if inverse exists), the solution is

$$\hat{\mathbf{s}}_{ML} = (\mathbf{H}^T\mathbf{R}_n^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}_n^{-1}\mathbf{r}. \tag{7.5}$$

Furthermore, if there is no knowledge about the correlation of the Gaussian noise vector (or if the elements are mutually independent), it is reasonable to substitute the covariant matrix $\mathbf{R}_n$ by the identity matrix $\mathbf{I}$. Thus the ML estimator, Eq. (7.5), is reduced to the *least-squares* (LS) estimator, i.e.,

$$\hat{\mathbf{s}}_{LS} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{r}. \tag{7.6}$$

It requires that the inverse of the square matrix $\mathbf{H}^T\mathbf{H}$ exists. Recall that the dimensionality of $\mathbf{H}$ is $N\times M$. For the inverse to exist, M needs to be less than or equal to N and the $M\times M$ $\mathbf{H}^T\mathbf{H}$ matrix should be of full rank M. *That is, the number of filters used in the filter-array spectrometer needs to be greater than or equal to the number of resolved points in the wavelength-domain*. For a practical consideration, we take $M=N$, i.e., $\mathbf{H}$ is a square matrix. Then, the LS estimator can be reduced to

$$\hat{\mathbf{s}}_{inv} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{r} = \mathbf{H}^{-1}\mathbf{r}. \tag{7.7}$$

It is worth to mention that, for zero-mean noise, the $\hat{\mathbf{s}}_{ML}$, $\hat{\mathbf{s}}_{LS}$, and $\hat{\mathbf{s}}_{inv}$ are unbiased, e.g., $E[\hat{\mathbf{s}}_{ML}] = (\mathbf{H}^T\mathbf{R}_n^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}_n^{-1}\mathbf{Hs}=\mathbf{s}$. Therefore, for a fixed unknown source signal vector $\mathbf{s}$, we may have the received signal vector $\mathbf{r}$ measured multiple times over either the temporal or spatial domain. This unbiased property ensures the enhancement of estimation accuracy after averaging operation.

The estimation-error covariance-matrix of the ML estimator, Eq. (7.5), can be calculated and expressed as $E\left[(\hat{\mathbf{s}}_{ML}-\mathbf{s})(\mathbf{s}_{ML}-\mathbf{s})^T\right]=\left(\mathbf{H}^T\mathbf{R}_n^{-1}\mathbf{H}\right)^{-1}$. We note that it is a function of the filter matrix **H**. Thus, it can tell us how good an estimator can be for a particular filter-array. We note that, although the covariance matrix of system noise $\mathbf{R}_n$ is fixed, the variance of the estimation error can be amplified by the detector sensitivity matrix $\mathbf{H}$. In this chapter, we are interested in the case that $\mathbf{H}$ is a square matrix. Conventionally, the singular value decomposition (SVD) is considered as a powerful technique to deal with the noise amplification issue. This method computes the inverse of the $\mathbf{H}$ matrix based on the singular value decomposition where an eigenvalue less than a certain threshold can be discarded. The threshold needs to be carefully chosen. A larger threshold results in a worse approximation of the **H** matrix, but less noise amplification.

By exploiting the non-negative nature of the spectral content, we found the non-negative constrained least-squares (NNLS) algorithm work particularly well to estimate the target spectra. NNLS can be seen as a member of the family of the least squares estimator. NNLS returns the vector $\hat{\mathbf{s}}$ that minimizes the norm $\left\|\mathbf{H}\hat{\mathbf{s}}-\mathbf{r}\right\|_2$ subject to $\hat{\mathbf{s}}>\mathbf{0}$ [66]. The original design of the algorithm was by C. L. Lawson,and R. J. Hanson [73]. Although the NNLS algorithm solves the solution iteratively, the iteration always converges.

It is worth to point out the weakness of using pseudo-inverse intending for "high" resolution estimation. Consider the same model of Eg. (7.1), and assume $N<M$, i.e., the number of filters is less than the number of resolved points in the wavelength-domain. Suppose we could

find an $M \times N$ matrix $\mathbf{H}^*$ (by any methods) such that $\mathbf{H}^*\mathbf{H} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Then we could have $E[\mathbf{H}^*\mathbf{r}] = E[\mathbf{s} + \mathbf{H}^*\mathbf{n}] = \mathbf{s}$. In this case, we could recover the source signal vector with high resolution by only using a few filters. However, this supposition can not be fulfilled, and the estimation done this way is always *biased*. Since $\mathbf{H}$ is an $N \times M$ matrix with $N < M$, the $M \times M$ $\mathbf{H}^*\mathbf{H}$ matrix is not possible to be full rank and thus can not be the identity matrix, i.e., $\mathbf{H}^*\mathbf{H} = \mathbf{A}$, where $\mathbf{A}$ is an arbitrary matrix other than the identity matrix. Also the inverse of $\mathbf{A}$ (in the strict sense) does not exist. Therefore, $E[\mathbf{H}^*\mathbf{r}] = E[\mathbf{A}\mathbf{s} + \mathbf{H}^*\mathbf{n}] = \mathbf{A}\mathbf{s}$, *the estimated source signal vector is always a biased version of the original input spectrum even in a no noise environment.*

7.4 DSP IMPLEMENTATION AND EXPERIMENTAL RESULTS

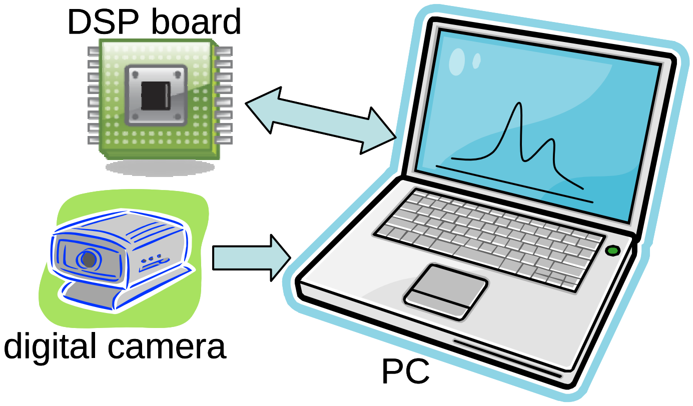


Figure 30. System set-up for DSP implementation.

Figure 30 illustrates the system set-up for our experimental filter-array based spectrometer. A digital camera (IMx-1040FT), a DSP board (TMDSDSK6713), and a personal computer (PC) are used for the preliminary demonstration. The PC serves as a *bridge* and *monitor*. The camera is connected to the PC via the FireWire interface carrying digital signals from each CCD sensor in 8-bit depth. The PC passes the digitalized signals received from the camera to the DSP board via the USB interface. Digital signal processing is performed on the DSP board. The processed data are then acquired by the PC and shown on the PC screen. Forty filters arranged in a straight line configuration are directly placed on top of CCD-sensors. Each of the filters occupies 5 CCD sensors along the horizontal direction. In this system, we only adopt the signal from one CCD sensor near the center of the filter for reducing the possible stray-light effect, which may come from the gap between filters and CCD sensors. We note that this preliminary system serves as a complete prototype for spectrometers on-a-chip consisting of a detector unit and a DSP unit. We implemented both the SVD-inverse algorithm and the NNLS algorithm on the DSP board.

Ideally, if the sensitivity functions are delta-function-like with narrow response ranges, the output directly from the detectors would compose the spectral content. However, the spectral detectors used in this system are far from delta-function-like for the consideration to low-cost fabrication. As illustrated in Figure 31, the detectors show broad ranges of response while each has its own peak response spot. The shapes of the responses are different as well. Therefore, the spectral outputs obtained directly from the detectors are very different from the input spectral shape.

Figure 32 shows the experimental results of the filter-array based spectrometer measured by shining a red LED whose center peak is at 650nm (LED model: HLMP-4100). The original data were in 10nm intervals. The curves shown in Figure 32 are obtained by a cubic interpolation processing as suggested by the *Commission Internationale de l'Eclairage* (CIE). Figure 32 (a) depicts the spectrum of the LED provided by the manufacturer. Figure 32 (b) shows the spectral content directly obtained from the detector outputs, whereas Figure 32 (c) shows the estimated spectral content by the NNLS algorithm. The NNLS algorithm shows an excellent estimation of the target spectrum. We have also tried different LEDs whose center peaks vary from 450nm to 700nm. All of the estimations show excellent agreements with the peak-wise target spectra. On the other hand, Figure 32 (d) shows the estimated spectral content by the SVD-inverse algorithm with the threshold set to zero. We have tried every possible threshold, and note that the results from the SVD-inverse algorithm do not reflect the LED spectra correctly.

We note that the required memory size of both algorithms are dominated by the pre-stored digitalized coefficients of the detector sensitivity matrix $\mathbf{H}$, and are less than 200 kilobyte. However, the required processing power is significantly different. In our implementation, the total number of *execution cycles* for the NNLS algorithm is 64,898,987 whereas that for the SVD-inverse algorithm is 1,200,737, which are equivalent to 0.1102 second and 0.0013 second, respectively, execution time on the TI 6713 DSP chip operating at 225 MHz. The NNLS is two orders of magnitude slower than the SVD-based method. Thus, a further research on high-performance but low-complexity algorithms is desired.

7.5 CONCLUSION

In this chapter, we consider a filter-array based spectrometer in which a low-quality but low-cost filter-array is used. Target spectra can be estimated and recovered accurately through DSP techniques. By exploiting the non-negative property of the spectral content, NNLS algorithm is found particularly useful for this application. Through hardware demonstration, we verified the achievability of a fine spectrometer based on a low-quality, low-cost filter-array

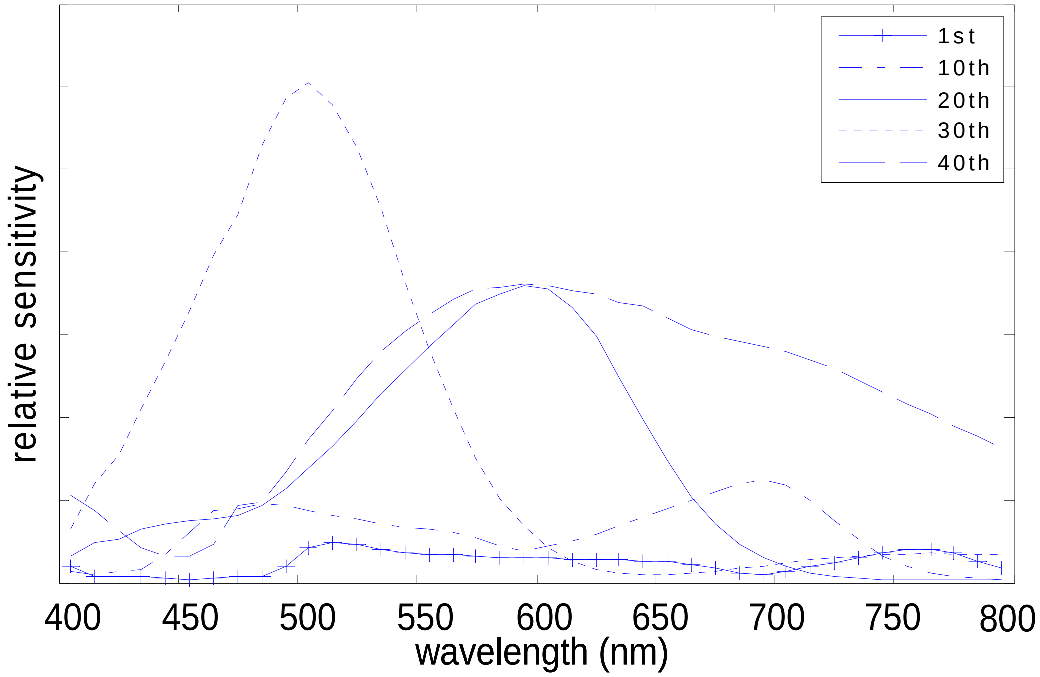


Figure 31. Sensitivity response of the 1st, 10th, 20th, 30th,and 40th spectral detectors.

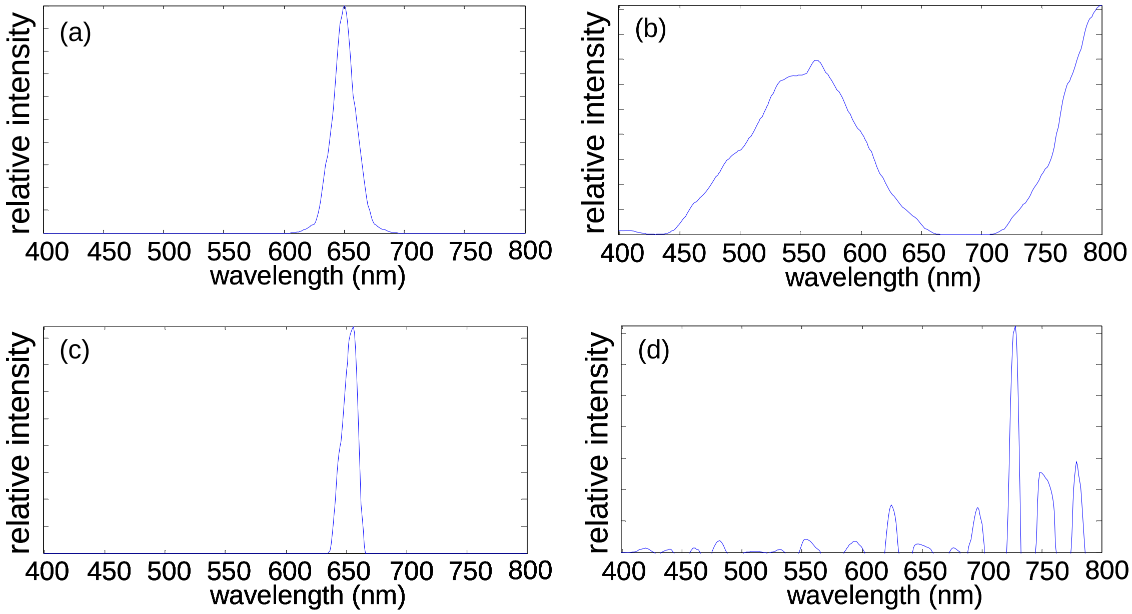


Figure 32. Experimental results for HLMP-4100 red LED, peak at 650nm.

(a) spectrum of the LED provided by the manufacturer, (b) spectrum obtained directly from the spectral detector outputs, (c) spectrum obtained after digital-signal-processing (DSP) based on the NNLS algorithm, and (d) spectrum obtained after DSP based on the SVD algorithm.

8.0 FUTURE DIRECTION

It is probably for sure that there is no boundary for any research in any field. In the following, we would like to point out some possible future directions related to the thesis.

One direction to extend the work in chapter 3 is to apply tight bounding techniques such as the tangential sphere bound (TSB). These techniques could help us to understand the genuine performance, and therefore help us to investigate the trade-off between encoding/decoding hardware complexity and error-performance. Regarding the relay assisted wireless multiple access networks, as discussed in chapter 4 and chapter 5, one interesting and critical direction is to investigate the practical or realistic benefits after imposing the management costs of the implementation of network-coding techniques. Studying the trade-off between the coding gain and coding (system) overhead shall be a significant direction. In chapter 6, we think it is interesting to study and establish accurate human perceptual models, as it would be a great help for efficiently designing sensory-based or perceptual-based (communication) systems dedicated to human users. In chapter 7, we wish to design application-oriented algorithms for spectrometers to extract valued information in bio-applications such as toxic test or glucose measurement.

BIBLIOGRAPHY

[1] S. Haykin, Communication systems, 4th edition, Wiley, 2000.

[2] C. E. Shanon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379-423, 623-656, July, October, 1948.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, ``Near Shannon limit error-correcting coding and decoding: turbo codes," *Proc. IEEE Intern. Conf. on Communication (*ICC), Geneva Switzerland, pp 1064-1070, 1993.

[4] D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” IEEE Trans. Inform. Theory, vol. 45, no. 2, pp. 399-431. 1999.

[5] J. Duffy, Network coding: networking’s next revolution, NetworkWorld, Dec. 2007.

[6] R. Koetter and M. Medard, “ An algebraic approach to network coding,” IEEE/ACM Trans. Networking, vol. 11, no. 5, pp. 782-795, 2003.

[7] P. E. Gill, W. Murray, and M. H. Wright, Practical optimization, Academic Press, 1982.

[8] R. Z. Morawski, “Spectrophotometric applications of digital signal processing,” *Measurement Science and Technology*, vol. 17, no. 9, pp. 117-144, 2006.

[9] D. Divsalar, and S. Dolinar, “Concatenation of Hamming codes and accumulator codes with high-order modulations for high-speed decoding,” *IPN progress report*, Feb. 2004.

[10] C. H. Hsu and A. Anastasopoulos, “Capacity-achieving codes with bounded graphical complexity on noisy channels, “ in *Proc. Allerton Conf. Commun., Control, Comp.*, Allerton House, IL, Sept. 2005.

[11] M. Gonzalez-Lopez, F. J. Vazquez-Araujo, L. Castedo, and J. Garcia-Frias, “ Serially-concatenated low-density generator matrix (SCLDGM) codes for transmission over AWGN and Rayleigh fading channels, “ *IEEE Trans. on Wireless Commun.*, vol. 6, pp. 2753-2758, Aug. 2007.

[12] D. Burshtein, and G. Miller, “Asymptotic enumeration methods for analyzing LDPC codes,” *IEEE Trans. on Inform. Theory*, vol. 50, No. 6, June 2004.

[13] C.-H. Hsu, and A. Anastasopoulos,”Asymptotic weight distributions of irregular repeat-accumulate codes,” *Proc. of IEEE GLOBECOM*, pp. 1147 – 1151, 2005.

[14] N. Varnica and M. Fossorier, “Belief-propagation with information correction: improved near maximum-likelihood decoding of low-density parity-check codes,” in *Proc. International Symposium on Inform. Theory*, Chicago, USA, June 2004.

[15] H. Pishro-Nik and F. Fekri, “On decoding of low-density parity-check codes over the binary erasure channel,” *IEEE Tran. Inform. Theory*, vol. 50, no. 3, pp. 439-454, Mar. 2004.

[16] D. Divsalar, “A simple tight bound on error probability of block codes with application to turbo codes,” Tech. Rep. TDA Progress Report, Jet Propulsion Labs., Pasadena, CA, Nov. 1999.

[17] I. Sason and S. Shamai, *Performance analysis of linear codes under maximum-likelihood decoding*, Now Publishers Inc, Hanover MA, 2006.

[18] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge University Press, 2008.

[19] J. F. Cheng, and R. J. McEliece, “Some high-rate near capacity codes for the Gaussian Channel,” in *Proc. of 34th Allerton Conf. Commun., Control and Computing*, Oct. 1996.

[20] Heung-No Lee, “Exact distance spectrum for low density parity-check codes,” technique note.

[21] J. Garcia-Frias, and W. Zhong, “Approaching Shannon performance by iterative decoding of linear codes with low-density generator matrix,” *IEEE Comm. Letters*, pp. 266-268, June 2003.

[22] L. Ping, S. Chan, and K. L. Yeung, “Iterative decoding of multi-dimensional concatenated single parity check codes,” *IEEE International Conf. Commun.*, vol. 1, pp. 131-135, June 1998.

[23] M. G. Luby M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, “Practical loss-resilient codes,” Proc. 29th Symp. on Theory of Computing, pp. 150-159, 1997.

[24] W. Zhong, and J. Garcia-Frias, “LDGM codes for channel coding and joint source-channel coding of correlated sources,” *EURASIP Journal on Applied Signal Processing*, pp. 942-953, June 2005.

[25] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, vol. 46, pp. 1204-1216, 2000.

[26] R. G. Gallager, *Low-Density Parity-Check codes.* Cambridge, MA: MIT Press, 1963.

[27] D. J. C. MacKay, ”Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inform. Theory*, Vol. 45, No. 2, pp. 399-431, Mar. 1999.

[28] T. Richardson, A. Shokrollahi, and R. Urbanke, “Design of Capacity Approaching Irregular Low-Density Parity-Chek codes,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 619-637, Feb. 2001.

[29] P. Zarrinkhat, and A. H. Banihashemi,”Threshold values and convergence properties of majority-based algorithms for decoding regular low-density parity-check codes,” *IEEE trans. Commun.*, vol. 52, pp. 2087-2097, Dec. 2004.

[30] L. Bazzi, T. Richardson, and R. Urbanke, “Exact thresholds and optimal codes for the binary symmetric channel and Gallager’s decoding algorithm A,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 2010-2021, Sept. 2004.

[31] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Efficient Erasure Correcting Codes,” *IEEE Trans. Inform. Theory*, vol. 47, No. 2, pp. 569-584, Feb. 2001.

[32] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. on Inform. Theory,* vol. 27, no. 5, pp. 533-547, Sep. 1981.

[33] E. Kreyszig, *Advanced Engineering Mathematics*, 9th edition, Wiley.

[34] C. Hausl, F. Schreckenbach, I. Oikonomidis, and G. Bauch, "Iterative network and channel decoding on a Tanner graph", *Proc. Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, Sep. 2005.

[35] X. Bao and J. Li, “Matching code-on-graph with network-on-graph: adaptive network coding for wireless relay networks,” in *Proc. Allerton Conf. on Commun., Control and Computing,* Urbana Champaign, IL, Sept. 2005.

[36] J. N. Laneman, and G. W. Wornell, “Distributed Space-Time-Coded protocols for exploiting cooperative diversity in wireless networks,” *IEEE Trans. Inform. Theory,* vol. 49, pp. 2415-2425, Oct. 2003.

[37] Yingda Chen, Salinee Kishore, and Jing Li, "Wireless Diversity through Network Coding," *Proc. of IEEE Wireless Communications and Networking Conf. (WCNC)*, Las Vegas, NV, March, 2006.

[38] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Hoboken, New Jersey. Wiley-Interscience, 2005, pp. 634-649.

[39] P. A. Chou, Y. Wu, and K. Jain, “Practical network coding,” *41st Allerton Conf. Comm., Control and Computing,* Oct. 2003.

[40] C.-C. Chang and H.-N. Lee, “Space-time mesh codes for the multiple-access relay network: space v.s. time diversity benefits,” in *Proc. Inform. Theory and Applications Workshop* (ITA), San Diego, CA, Jan. 2007.

[41] S. W. Kim, S. G. Kim, and B. K. Yi, “Decentralized random parity forwarding in multi-source wireless relay networks,” in *IEEE Global Telecommunications Conference*, Washington DC, Nov. 2007.

[42] C. Hausl and P. Dupraz, “Joint network-channel coding for the multiple-access relay channel,” in *proc. Intern. Workshop on Wireless Ad Hoc and Sensor Networks*, New York, USA, Jun. 2006.

[43] D. H. Woldegebreal, and H. Karl, “Multiple-access relay channel with network coding and non-ideal source-relay channels,” in *proc. IEEE International Symposium on Wireless Communication Systems* (ISWCS), Trondheim, Norway, Oct. 2007.

[44] J. L. Laneman, D. N. C. Tse, and G. W. Wornell, “Cooperative diversity in wireless networks: efficient protocols and outage behavior,” *IEEE Trans. Information Theory,* vol. 50, no. 12, pp. 3062−3080, Dec. 2004.

[45] E. Ayanoglu, C.-L. I, R. D. Gitlin, and J. E. Mazo, “Diversity coding for transparent self-healing and fault-tolerant communication networks,” *IEEE Trans. Communications*, vol. 41, no. 11, pp. 1677−1685, Nov. 1993.

[46] W.-J. Huang, Y.-W. P. Hong, and C.-C. J. Kuo, “Relay-assisted decorrelating multiuser detector (RAD-MUD) for cooperative CDMA networks, “ *IEEE J. Selected Areas in Communications*, vol. 26, no. 3, pp. 550−560, Apr. 2008.

[47] C. Gkantsidis, and P. R. Rodriguez, “Cooperative security for network coding file distribution,”, in *Proc. of the IEEE Intern. Conf. Computer Commun.* (INFOCOM 2006), April 2006.

[48] M. Krohn, M. Freedman, and D. Mazieres, “On-the-fly verification of rateless erasure codes for efficient content distribution,” in *IEEE Symp. Security and Privacy*, Berkeley, CA, 2004.

[49] R. J. McEliece, “A public-key cryptosystem based on algebraic coding theory,” *The Deep Space Network Progress Report* (DSN PR), pp. 114-116, Feb. 1978.

[50] H. Niederreiter, “Knapsack-type cryptosystems and algebraic coding theory,” *Problems of Control and Information Theory*, vol. 15, no. 2, pp. 159-166, 1986.

[51] X. Li, J. Hwu, and E. P. Ratazzi, “Using antenna array redundancy and channel diversity for secure wireless transmissions,” *Journal of Commun.*, vol. 2, no. 3, pp. 24-32, May 2007.

[52] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, “The Benefits of Coding over Routing in a Randomized Setting,” ISIT 2003.

[53] J.-S. Park, M. Gerla, D. S. Lun, Y. Yi, and M. Medard, “Codecast: a network-coding-based ad hoc multicast protocol,” *IEEE Wireless Communication*, vol. 13, no. 5, pp. 76-81, Oct. 2006.

[54] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657-670, Feb. 2001.

[55] G. Miao and Z. Niu, "A generalized satisfaction oriented bandwidth management," in *Proceedings of the 2005 Asia-Pacific Conference on Communicatios*, pp. 625–629, Perth, Western Australia, Oct. 2005.

[56] G. Miao and Z. Niu, "Bandwidth management for mixed unicast and multcast multimedia flows with perception based QoS differentiation," in *Proceedings of the IEEE International Conference on Communication*, pp. 687–692, Istanbul, Turkey, Jun. 2006.

[57] S. Pal, S. K. Das, and M. Chatterjee, "User-satisfaction based differentiated services for wireless data networks," in *Proceedings of IEEE International Conference on Communication*, pp. 1174–1178, Seoul, Korea, May 2005.

[58] H. Ji, "An economic model for bandwidth allocation in broadband communication networks," in *Proceedings of the IEEE International Conference on Communication*, pp. 658–662, New Delhi, India, Jun. 1996.

[59] J. B. Rosen, "The gradient projection method for nonlinear programming, part I: linear constraints," *Journal of the Society for Industrial and Applied Mathematics,* vol. 8, no. 1, pp. 181–217, 1960.

[60] A. Stutzer, "The role of income aspiration in individual happiness," *Journal Economic Behavior and Organization,* vol. 54, no. 1, pp. 89–109, 2004.

[61] M. Andrews, J. Cao, and J. McGown, "Measuring human satisfaction in data network," in *Proceedings of the 25th IEEE International Conference on Computer Communications,* Barcelona, Catalunya, Spain, April 2006.

[62] H. Koumaras, T. Pliakas, and A. Kourtis, "A novel method for pre-encoding video quality prediction," in *Proceedings of the 16th IST Mobile and Wireless Communications Summit,* Budapest, Hungary, Jul. 2007.

[63] H. Koumaras, A. Kourtis, D. Martakos, and J. Lauterjung, "Quantified PQoS assessment based on fast estimation of the spatial and temporal activity level," *Multimedia Tools and Applications,* vol. 34, no. 3, pp. 355–374, 2007.

[64] D. P. Bertsekas, *Nonlinear Programming.* Belmont, MA: Athena Scientific, 1995.

[65] C. P. Bacon, Y. Mattley, and R. Defrece, "Miniature spectroscopic instrumentation: applications to biology and chemistry," *Rev. Sci. Instrum*., vol. 75, pp. 1-16, 2004.

[66] D. C. Heinz, and C.-I Chang, "Fully constrained least-squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens*. vol. 39, pp. 529-546, 2001.

[67] O. Manzardo, H. P. Herzig, C. R. Marxer, and N. F. de Rooij, "Miniaturized time-scanning Fourier transform spectrometer based on silicon technology," *Opt. Lett.*, vol. 24, pp. 1705-1707, 1999.

[68] K. Chaganti, I. Salakhutdinov, I. Avrutsky, G. W. Auner, "A simple miniature optical spectrometer with a planar waveguide grating coupler in combination with a plano-convex leng," *Opt. Express*, vol. 14, pp. 4064-4072, 2006.

[69] R. F. Wolffenbuttel, "State-of-the-art in integrated optical microspectrometers," *IEEE Trans. Instrum. Meas.*, vol 53, pp. 197-202, 2004.

[70] R. Shogenji, Y. Kitamura, K. Yamada, S. Miyatake, and J. Tanida, "Multispectral imaging using compact compound optics," *Opt. Express*, vol. 12, pp. 1643-1655, 2004.

[71] S.-W. Wang, C. Xia, X. Cheng, W. Lu, L. Wang, Y. Wu, and Z. Wang, "Integrated optical filter arrays fabricated by using the combinatorial etching technique," *Opt. Lett.*, vol. 31, pp. 332-334, 2006.

[72] S.-W. Wang, C. Xia, X. Cheng, W. Lu, M. Li, H. Wang, W. Zheng, and T. Zhang, "Concept of a high-resolution miniature spectrometer using an integrated filter array," *Opt. Lett.*, vol. 32, 632-634, 2007.

[73] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, 1974.

[74] J. G. Proakis, *Digital Communications*, McGraw Hill, 2000.