Title
Systolic Design of MIMO Lattice Detection and Channel Modeling for Wireless Communication Systems

Author
Wang, Ni-Chun

Publication Date
2012

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Systolic Design of MIMO Lattice Detection and Channel Modeling for Wireless Communication Systems

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Ni-Chun Wang

2012

Abstract of the Dissertation

Systolic Design of MIMO Lattice Detection and Channel Modeling for Wireless Communication Systems

by

Ni-Chun Wang

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2012

Professor Kung Yao, Chair

Two independent but equally challenging problems in the wireless communication systems are considered in this dissertation. First, the systolic design of the multiple-input, multiple-output (MIMO) lattice reduction-aided detection is proposed. Lattice-reduction-aided detection (LRAD) has been shown to be an effective low-complexity method with near-ML performance. However, lattice reduction needs to be performed as channel state changes. As the channel change rate is high, or a large number of channel matrices need be processed such as in a MIMO-OFDM system, a fast-throughput algorithm and the corresponding implementation structure are needed for real-time applications. In this dissertation we advocate the use of systolic array architectures for MIMO receivers, and in particular we exhibit one of them based on LRAD. The "LLL lattice reduction algorithm" and the ensuing linear detections or successive spatial-interference cancellations can be located in the same array, which is hardware-efficient. Two modified LLL algorithms suitable for parallel processing are considered here for the systolic design, LLL algorithm with full-size reduction and all-swap lattice-reduction algorithm. In order to simplify the systolic array design, we replace the Lovász' condition in the definition of LLL-reduced lattice with the looser Siegel's condition and limit the

range of μ value. Simulation and FPGA emulation results show that the proposed systolic LLL is 1.6 time faster than the conventional LLL while the bit-error-rate performance of LRAD is still maintained with these relaxations.

Second, we consider the modeling of fading channels under abrupt changes. Fading channel is generally nonstationary in time, especially when there are moving objects near the field of transmission. The statistics of the channel are changing due to the temporal and spatial inhomogeneity. To characterize the temporal variation of the channel, short-term statistics need to be estimated. Instead of estimating the statistics over a fixed short period, we applied the Bayesian change point detection (CPD) for five common channel models to capture the locations of changes in time. The detected change points partition the channel into segments that are characterized by different parameters. We also derive the MAP and MMSE estimators for the model parameters of each segment based on the intermediate results of CPD. Therefore, once a change is detected, the parameters are obtained immediately. Test results on 802.11n channel simulator and channel measurement show the effectiveness of the CPD and the proposed estimators.

We also found CPD to be useful in biological applications. A bird phrase segmentation using entropy-based change point detection is proposed. Spectrograms of bird calls are usually sparse while the background noise is relatively white. Therefore, considering the entropy of a sliding time- frequency block on the spectrogram, the entropy dips when detecting a signal and rises back up when the signal ends. Rather than a hard threshold on the entropy to determine the beginning and ending of a signal, CPD is used to detect the statistical changes in the entropy sequence. With the novel spectral whitening method as the front-end processing, our proposed segmentation method generates more accurate time labels, reduces the false alarm rate and achieves higher classification rates than the conventional time-domain energy detection method.

The dissertation of Ni-Chun Wang is approved.

Babak Daneshrad

Dejan Markovic

Henry Huang

Kung Yao, Committee Chair

University of California, Los Angeles

2012

To my parents,

Chi-Min and Angel

Table of Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

It was 30 minutes after I passed my Ph.D. defense by the time I wrote down this section. I believe this is the best timing. When the committee members shook my hand and congratulated me, it was one of the most emotional moments in my life. The images of all those people who have helped my to accomplish this dissertation work was just like a movie flashing through my head.

First, I would like to thank my advisor, Prof. Kung Yao, who guided me into the world of systolic array that leads to this dissertation work. He not only advises my research work but always gives great suggestions and support for my life in the United States. Second, I would like to thank my co-advisor, Prof. Ezio Biglieri, for always giving me useful comments on MIMO communication systems. Without their guidance, I would not be able to won the best paper award.

I would also like to thank Dr. Farrokh Abrishamkar, whom I worked with during my internship in Qualcomm. By that time I was struggling with my research work and it was him that encouraged me to move on and finish the Ph.D. work. He was also the one who introduced the change point detection theory to me, and that leads to my second half of the dissertation.

I would like to thank Dr. Ralph Hudson, for always providing me insightful comments, which polished my research work. I would also like to thank my committee members, Prof. Babak Daneshrad, Prof. Dejan Markovic, and Prof. Henry Huang, who gave me great comments on my work during the qualify exam and Ph.D. defense.

I would like to thank Prof. Charles Taylor who gave me the opportunity to work on the NSF project on soundscape analysis of birds. With his lead, it was very enjoyable to work in this project toward the end of my Ph.D. study. I would also like to thank the team members of this project, Lee Ngee Tan, George Kossan, Zac Harlow, and Kantapon Kaewtip. I really learned a lot from those people who

have totally different background from me.

I would like to thank Dr. Chia-Hsiang Yang for all the help on FPGA emulation. I would also like to thank my former and current lab mates, Dr. Chiao-En Chen, Dr. Andreas Ali, Dr. Wei-Ho Chung, Dr. Juo-Yu Lee, Dr. Chih-Kai Chen, and Cheng-An Yang, for all the useful and constructive discussions.

I would like to thank to the Taiwan Merit Scholarship by Taiwan National Science Council for funding me the first four years of my Ph.D. study (TMS-094-2-A-002).

I would like to thank to my parents and my sister. Even though they are all in Taiwan and we only see each other once a year, I can always feel their warmest caring, which carried me through the whole Ph.D. years.

Last but not least, I would like to thank my beloved wife, Chi-Min, for all the efforts on taking care my life and our newborn girl so I can focus on my research. She makes my life colorful and is always my strongest support.

Vita

2001	Bachelor of Science, Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
2003	Master of Science, Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan
2003 – 2005	Served in Marching Band of Reserve Command, Taiwan
2005 – 2009	Taiwan Merits Scholarship
2007 – 2012	Teaching Fellow, Electrical Engineering Department and Online Master Program of School of Engineering and Applied Science (SEAS), UCLA. Taught EE103 : Applied Numerical Computing, EE131A : Probability and EE230A : Estimation and Detection in Communication and Radar Engineering.
2009	Interim Intern, Qualcomm CDMA Technologies, San Diego, California, USA
2009	International Scholarships, Phi Beta Kappa Alumni of Southern California
2012	Best Paper Award. Journal of Communications and Networks

CHAPTER 1

Introduction

Two challenging and equally important problems in the wireless communication systems are considered in this dissertation. First, we proposed a systolic array solution for high-throughput MIMO lattice-reduction-aided detection. Second, to characterize the temporal variation of the fading channel, we proposed a novel channel modeling method based on Bayesian change point detection and parameter estimation. Besides the work on wireless communication systems, we also found change point detection to be very useful on biological problems. With the collaboration with field biologists, we proposed an automated bird phrase segmentation by entropy-based change point detection.

1.1 MIMO Lattice-Reduction-Aided Detection

Multiple-input, multiple-output (MIMO) technology, using several transmit and receive antennas in a rich-scattering wireless channel, has been shown to provide considerable improvement in spectral efficiency and channel capacity [1]. MIMO systems yield spatial diversity gain, spatial multiplexing gain, array gain, and interference reduction over single-input single-output (SISO) systems [2]. However, these benefits come at the price of a computational complexity of the detector that may be intolerably large. In fact, optimal maximum-likelihood (ML) detection in large MIMO systems may not be feasible in real-time applications as its complexity increases exponentially with the number of antennas. Low-complexity receivers, employing linear detection or successive interference cancellation (SIC), are com-

putationally less intensive, and amenable to simple hardware implementation [3–5]. However, diversity and error-rate performance of these low-complexity detectors are not comparable to those achieved with ML detection.

Lattice-reduction-aided detection (LRAD), which combines lattice reduction (LR) techniques with linear detections or SIC, has been shown to yield some improvement on error-rate performance [6–9]. Lenstra-Lenstra-Lovász (LLL) algorithm [10] is the most widely used lattice reduction algorithm due to its polynomial time complexity. It was first invented to solve the polynomial factorization problem and later found to be useful in cryptology. Research on the LLL algorithm, is still very active [11]. In communications, complex numbers are useful to represent the baseband channel state information and the signals and it has been shown in [12] that LLL algorithm can be applied to complex-valued lattices. The performance of complex LLL-aided linear detection in MIMO systems was analyzed in [13]. LLL-based LRAD was shown to achieve full receiver diversity [14]. It was also shown that the LR-aided minimum mean-square-error decoding achieves the optimal diversity-multiplexing tradeoff [18]. When applied to MIMO detection, the average complexity of LLL algorithm is polynomial in the dimension of the channel matrix (the worst-case complexity could be unbounded [15]). A fixed-complexity LLL algorithm, which modifies the original version to allow more robust early termination, has recently been proposed in [19]. An embedded LLL-based lattice reduction MIMO decoding method was proposed in [20, 21], which showed a close-to-ML performance. The proximity factors, which are used to evaluate the performance of lattice-reduction decoding incluidng LLL, were proposed in [22].

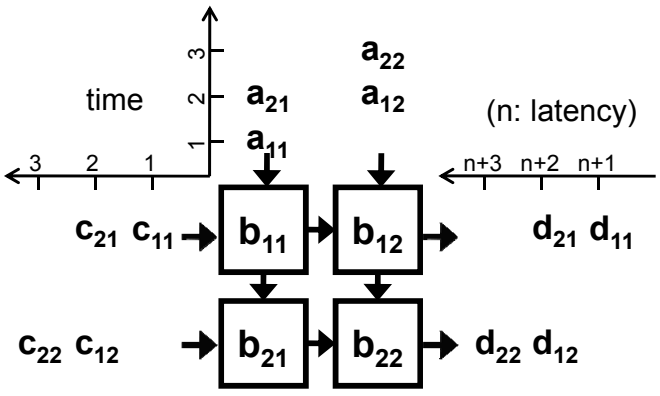
In LRAD, lattice algorithm needs be performed only when the channel state changes. If the channel change rate is high, or a large number of channel matrices need be processed such as in a MIMO-OFDM system, a fast-throughput algorithm and the corresponding implementation structure are needed for real-time applications. Seveal hardware implementations of the lattice reduction algorithms

have been proposed. In [23], the VLSI implementaion of Brun's algorithm was proposed for MIMO broadcast precoding. The field-programmable gate array (FPGA) implementation of Clarkson's algorithm was proposed in [24]. The VLSI and FPGA implementation of LLL algorithm were considered in [16,25,26]. In this dissertation, we consider LLL algorithm. To obtain the fast-throughput, we first discuss two variants of LLL algorithm, suitably modified for parallel processing. Second, we propose a novel systolic array structure implementing the two modified LLL algorithms and the ensuing detection methods. In the next section, we will give a brief introduction of systolic array.

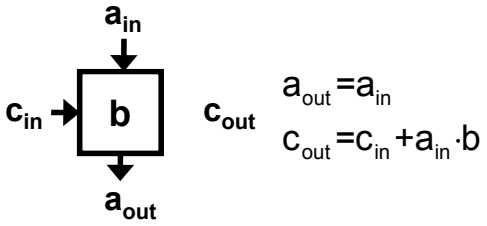
1.2 Systolic Array

A systolic array [27,28] is a network of processing elements (PE) which transfer data locally and regularly with nearby elements and work rhythmically. In Fig. 1.1(a), a simple two-dimensional systolic array is shown as an example. In this case, the matrix operation $\mathbf{D} = \mathbf{A} \cdot \mathbf{B} + \mathbf{C}$ is calculated by the systolic array, where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are 2×2 matrices. The operation of each PE is shown in Fig. 1.1(b). The inputs of the systolic array, the entries of matrices $\mathbf{A}$ and $\mathbf{C}$, are pipelined in a slanted manner for proper timing. Since all PEs can work simultaneously, the latency is shorter than with a single processor system, and the results of $\mathbf{D}$ are outputted in parallel.

Systolic algorithms and the corresponding systolic arrays have been designed for a number of linear algebra algorithms, such as matrix triangularization [29], matrix inversion [30] , adaptive nulling [31], recursive least-square [32,33], etc. An overview of systolic designs for several computationally demanding linear algebra algorithms for signal processing and communications applications was recently published in [34]. While systolic arrays allow simple parallel processing and achieve higher data rates without the demand on faster hardware capabilities, the existence



(a)



(b)

Figure 1.1: (a) Two-dimensional systolic array performing matrix calculation $\mathbf{D} = \mathbf{A} \cdot \mathbf{B} + \mathbf{C}$, where $a_{ij}, b_{ij}, c_{ij}, d_{ij}$ are the (i, j) entries of the matrix $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$. (b) The operation of each processing element.

of multiple PEs implies a higher cost of circuit area. Thus, time efficiency is traded off with circuit area in hardware design. For the application we are advocating in this dissertation (MIMO detectors), systolic arrays offer an attractive solution, as we must cope with a high computational load while requiring high throughput and real-time operation. Systolic arrays have been previously suggested for MIMO applications. In [35], the authors proposed a universal systolic array for adaptive and conventional linear MIMO detectors. In [36], a reconfigurable systolic array processor based on coordinate rotation digital computer (CORDIC) [37] is proposed to provide efficient MIMO-OFDM baseband processing. Also, matrix factorization and inversion are widely used in MIMO detection, with systolic arrays used to increase the throughput [5, 38].

1.3 Systolic Design for LLL-based LRAD

Our objective is to provide a novel systolic array design for LLL-based LRAD. The ideas are described from a system-level perspective instead of detailed discussion on the hardware-oriented issues. The system model and how LRAD works are briefly described in Chapter 2. Since the original LLL algorithm [8, 10–17] is not designed for parallel processing, and hence is not suitable for systolic design, two modified LLL algorithms are considered in Chapter 3. Note that we are not claiming the two algorithms works better than the original LLL in terms of the LRAD bit-error-rate (BER) performance. First, we improve on the format of conventional LLL algorithm by altering the flow of size-reduction process (we call it "LLL with full size-reduction," or FSR-LLL). FSR-LLL is more time-efficient in parallel processing than the conventional format, and hence suitable for systolic design. We also consider a variant of the LLL algorithm called "all-swap lattice reduction (ASLR)," which combines the FSR with the concept of swapping all columns within each iteration proposed by [39]. A crucial difference between ASLR and LLL algorithm is that with ASLR all lattice basis vectors are simultaneously processed during a single iteration. In both algorithms, in order to simplify the systolic array operations we replace the original Lovász' condition [10] of LLL algorithm with the slightly weaker Siegel's condition [40]. Surprisingly, for LR-aided linear detections the BER performance with Siegel's condition under the proper parameter setting is just as good as the one using Lovász' condition. However, for LR-aided SIC, the performance with Lovász' condition is still slightly better due to less error propagation. The mapping from algorithm to systolic array is introduced in Chapter 4. In our design, ASLR and FSR-LLL can be operated on the same systolic-array structure, but the external logic controller is also required to control the algorithm flow. Additionally, since ASLR was originally designed for parallel processing, a systolic array running ASLR is on the average more efficient than one running FSR-LLL. Simulation results also show that ASLR-based

LRAD has a BER performance very similar to that of LLL algorithm. After the channel state matrix has been lattice-reduced, linear detectors or SIC can also be implemented by the same systolic array without any extra hardware cost, which is also discussed in Chapter 4. The FPGA emulation of the proposed systolic design is discussed in Chapter 5. To reduce the hardware complexity, the μ values in FSR is bounded, while the overall performance of the LLL algorithm was still maintained. Comparison between our proposed design and the conventional LLL in FPGA implementation shows that the systolic arrays do provide faster processing speed with a moderate increase of hardware resources.

1.4 Temporal Variation of the Fading Channel Envelope

In wireless communications, the transmitted signal propagating through the channel will suffer from the power attenuation due to the combination of three major effects, path loss, shadowing and multipath propagation [41, 42]. The level of path loss is determined by the distance between transmit and receive antenna. The larger separation of antennas would cause the greater path loss. The shadowing effect comes from the obstacles in field of the transmission that absorb the radio signals. The wireless link at different locations or environments may experience significantly different shadowing even with the same antenna separation. The multipath effect is due to the reflection, refraction and scattering of the radio signals by the fixed or moving objects near the field of transmission, such that the transmit signals arrive at the receive antenna by more than one paths with different delays, amplitudes, and phases. The multipath dispersion distorts the transmit signals and could largely degrade the transmission quality of the communication systems.

The degree of the fading effects mentioned above may be different depending on the environment and locations of the antennas or the settings of the communication system. However, fading effects are inevitable and are difficult to be mitigated in

wireless communications. Hence, it is crucial to characterize the fading channel of the various environments of interest, in order to design the system that can match the channel and achieve optimal performance. The fading channel amplitude is usually characterized by probability models, such as Rayleigh, Rice, Nakagami-m, Weibull and lognormal distributions. To fit those models with the empirical data, the parameters of the probability models needs to be estimated accurately. The parameters are important for the design of the system, such as optimal reception of the signals [43], optimizing the transmitter diversity [44, 45], and adaptive modulation and coding [46], and are equally important for the evaluation of the system [47–51] . Estimating parameters based on a sequence of channel measurement using maximum likelihood estimation or moment estimators has been well-studied [45, 52–54].

Research on the channel measurement over various transmission environments shows that the channel is only stationary for a short amount of time. Generally, the channel is nonstationary in time, especially when the transmitters/receivers are in motion, or there are moving objects near the fixed wireless link which changes the phases of the arriving multipath signals. Therefore, the statistics or even the distribution model are changing due to the temporal and spatial inhomogeneity [41]. Studying the insight of temporal variations of the channel envelope becomes more important for wireless local area networks (WLANs) and modern mobile communications, as their transmission power is designed to be relatively low. Several research groups have done the modeling and the analysis on the temporal variations of indoor and mobile channels causing by various conditions. In [55], it is shown that for indoor channel people moving around receive antennas causes larger variation in the signal level than the movement around the transmit antennas. It is also shown that Weibull and Nakagami-m both fit the temporal fading well. The human shadowing effect is also studied for fixed wireless link under WLAN [56] and ultra-wide band (UWB) [57] indoor channels. It is shown

that people moving in between the transmission path significantly changes the Rician K-factor, received power, and RMS delay spreads. In [58, 59], the effects of nearby traffic on the variations of Rician K-factor of outdoor urban channels have been studied. It is shown that heavy traffic causes richer scattering and thus lower K-factor. Temporal fading was also studied for the industrial indoor environment with lots of machinery movement [60]. In [61], the first-order statistical model of the Rician K-factor was proposed, and the variations of K with time, frequency and location were also considered. In those analyses, the parameters of the probability models were usually estimated over a short period of observations, varying from a minute to 15 minutes. In other words, the channel samples of that fixed period were assumed to stationary and share the same statistics.

1.5 Change Point Detection and Parameter Estimation of the Channel Envelope

However, the changes in the environment, such as people movement or street traffic are usually unpredictable, and the channel measurement within an interval of a few minutes could contain more than one statistical states. Estimating parameters from the whole interval may lose some insights of the true temporal variations. Hence, to precisely study the temporal variation of the channel statistics, there should be a mechanism that detects locations of the changes in time. These change points naturally partition the observation into segments that are characterized by different parameters or even different probability models. We proposed using an online Bayesian change point detection (CPD), which is based on the work of [62], to detect the changes in a sequence of the channel envelope samples.

Change point detection is to detect or predict sudden change in the generative parameters of a time series. The change points may occur at random time instants of the collected data, and split the data into a set of disjoint segments.

Conventionally, each segment is modeled by a single model with certain parameters, and the models and parameters are different across the segments. For example, the signals may suffer drastic shift in mean, variance, or phase due to the source characteristic or even the sensor errors. CPD is shown to be a key aspect of many real world applications, such as detection of computer network attacks [63, 64], data selection in the sensor networks [65], heart beat variability [66] and volatility modeling for financial data [67].

In Chapter 6, the theory of CPD is briefly introduced and how we fit the theory into those common channel probability models is discussed in details. Additionally, we derived the maximum a posteriori (MAP) and minimum mean square error (MMSE) estimators of the parameters of each probability model which only utilize the intermediate results of CPD. Consequently, as the CPD detects the changes in the sequence, the parameters of each segment can be obtained immediately. In Chapter 7, we tested the proposed methods with random sequences, the channels generated by 802.11n channel simulator, and the channel measurement. The results show the effectiveness of CPD on detecting changes, and the probability models based on the proposed estimators fit the empirical data well and pass the Komolgorov-Smirnov goodness-of-fit test.

1.6 Bird Phrase Segmentation Using Change Point Detection

Besides fading channel, we also applied the Bayesian change point detection on biological applications. An automated system capable of reliably segmenting bird songs and identifying species is an indispensable tool for analyzing an audio database, used for studying behavior of vocalizing species [68]. As more attention is directed toward "soundscape ecology" [69] and specifically a more refined understanding of bird communication, an automated solution becomes more

important. Several species identification methods have been shown to be useful in many aspects; however the automated segmentation of the bird songs has received less attention. Manually segmented bird songs were used for bird species identification in [70–73]. Time-domain energy detection is used in [74–76] for segmentation. In [77], the authors used Kullback-Leibler (KL) divergence between the audio power spectrum and the uniform distribution for segmentation. A time-frequency segmentation by machine learning methods is proposed in [78].

We propose a time/frequency segmentation method by entropy-based change point detection. Entropy is calculated from the sliding time-frequency blocks in the spectrogram. Since the spectrogram of bird songs is generally sparse in the sense that high power signals occupy a small fraction of the time and frequency bins. This is because the call usually consists only a single frequency at any given instant. Harmonics may be present but even so the instantaneous spectrum is sparse. The call could be slowly modulated in amplitude or frequency [79] but still the spectrogram is sparse, assuming there is no strong background noise occupying a large number of frequency bins. In contrast, when there is no call present, the spectrogram displays a random response whose statistics do not vary with time or frequency. Therefore, the entropy drops when the sliding block is moving from an time interval without any bird songs ("quiet period" will be used hereafter) to the start of a song. The entropy stays low in the time interval of a bird song ("call period" will be used hereafter) and rises up as the block is leaving the call period. A polynomial-based spectral whitening method is also proposed to serve as the front-end of the system. The purpose is to enlarge the difference between the entropy levels of a call and a quiet period.

Bird songs are divided into four hierarchical levels of notes, syllables, phrases, and song [80]. Phrases are typically the basic units of understanding the bird communication. Our goal is to find accurate and consistent phrase labels such that the segmentation results could be passed to a phrase classifier for reliable

classification. A phrase usually consists of several syllables with short silence in between, which makes phrase segmentation non-trivial. Therefore, the Bayesian change point detection is used here to detect the statistical change in the entropy sequence.

When applying the segmentation method to the long field recordings, the entropy of bird calls is not always at the same level and is sometimes even higher than the one at certain quiet periods, depending on the interference level. Taking hard thresholding can easily miss those calls. The advantage of using CPD over using a hard threshold is that CPD only focuses on the changes in the statistics of the data. Therefore, it can distinguish call periods from quiet periods as long as there are changes in the entropy level.

1.7 Notations

The following notations are used throughout the remaining chapters. Capital bold letters denote matrices, and lower case bold letters denote column vectors or a collection of samples of a sequence. For example, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_{2,} \cdots, \mathbf{x}_m]$ is a matrix with m columns of $\mathbf{x}_1$ to $\mathbf{x}_m$. The entry of a matrix $\mathbf{X}$ at position (i, j) is denoted by $x_{i,j}$, and the k^{th} element of a vector $\mathbf{x}$ is denoted by x_k. The submatrix (subvector) formed from the a^{th} to b^{th} rows and m^{th} to n^{th} columns of $\mathbf{X}$ is denoted by $\mathbf{X}_{a:b,m:n}$. Denote the k^{th} row of a matrix $\mathbf{X}$ as $\mathbf{X}^{[k]}$. The notations $(\cdot)^+$, $(\cdot)^T$, $(\cdot)^H$ and $(\cdot)^\dagger$ are used for conjugate, transpose, Hermitian transpose, and Moore-Penrose pseudo-inverse of a matrix, respectively. $\|\mathbf{x}\|$ is the Euclidean norm of the vector $\mathbf{x}$. $\Re(\cdot)$ and $\Im(\cdot)$ are the real and imaginary parts of a complex number, respectively. $\lceil x \rfloor$ indicates the closest Gaussian integer to x; namely $\lceil x \rfloor = \lceil \Re(x) \rfloor + j \lceil \Im(x) \rfloor$ and $j = \sqrt{-1}$. $\mathbf{I}_m$ and $\mathbf{0}_m$ are $m \times m$ identity and null matrices, respectively. The all one's vector is denoted by $\mathbf{1}$. The Hadamard product between matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \circ \mathbf{B}$. The elementwise k^{th} power of the vector $\mathbf{x}$ is denoted

by $\mathbf{x}^k$. The notations $\cap$ and $\cup$ denote the intersection and the union in the set theory, respectively. Gaussian probability density function (pdf) with mean μ and variance σ^2 is denoted by $\mathcal{N}(\mu, \sigma^2)$ and inverse-Gamma pdf with parameters α and β is denoted by $\mathcal{IG}(\alpha, \beta)$. $\Gamma(\cdot)$ is the Gamma function. The determinant of a matrix $\mathbf{X}$ is denoted by $|\mathbf{X}|$. The natural logarithm is denoted by log.

CHAPTER 2

Background of MIMO Low Complexity Detection Methods

2.1 MIMO System Model

We consider a MIMO system with n_t transmit and n_r receive antennas in a rich-scattering flat-fading channel. Spatial multiplexing is employed, so that data are transmitted as n_t substreams of equal rate. We consider the uncoded transmission and these substreams are mapped onto the same finite subset of Gaussian integers $\mathcal{M} = \{a+bj | a, b \in \mathbb{Z}\}$, such as QAM constellation with proper scaling and shifting. Let $\mathbf{x} \in \mathcal{M}^{n_t}$ denote the complex-valued $n_t \times 1$ transmitted signal vector with independent and zero-mean, and $\mathbf{y}$ the complex-valued $n_r \times 1$ received signal vector. The baseband model for this MIMO system is

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{2.1}$$

where $\mathbf{H}$ is the $n_r \times n_t$ channel matrix: its entries are uncorrelated, zero-mean, unit-variance complex circularly symmetric Gaussian fading gains h_{ij}, and $\mathbf{n}$ is the $n_r \times 1$ additive white complex Gaussian noise vector with zero mean and $E[\mathbf{n}\mathbf{n}^H] = \sigma_n^2\mathbf{I}$. The substreams of $\mathbf{x}$ are assumed to be zero-mean and independent, and the average power of each transmitted signal x_i is assumed to be σ_x^2. Namely, $E[\mathbf{x}\mathbf{x}^H] = \sigma_x^2\mathbf{I}$. Additionally, we assume that the channel matrix entries are fixed during each frame interval, and the receiver has perfect knowledge of the realization of $\mathbf{H}$.

2.2 Linear Detection

In linear detection, the estimated signal $\hat{\mathbf{x}}$ is computed by first premultiplying the received signal $\mathbf{y}$ by an $n_r \times n_t$ "weight matrix" $\mathbf{W}$. The two most common design criteria for $\mathbf{W}$ are zero-forcing (ZF) and minimum mean-square error (MMSE). In zero-forcing detection, the weight matrix $\mathbf{W}_{ZF}$ is set to be the Moore-Penrose pseudo-inverse $\mathbf{H}^{\dagger}$ of the channel matrix $\mathbf{H}$, i.e.,

$$\hat{\mathbf{x}}_{ZF} = \mathbf{W}_{ZF}\mathbf{y} = \mathbf{H}^{\dagger}\mathbf{y} = \mathbf{x} + \mathbf{H}^{\dagger}\mathbf{n}. \tag{2.2}$$

It is known that zero-forcing detection suffers from the noise enhancement problem, as the channel matrix may be ill-conditioned. To overcome this problem, the noise level is taking into consideration in MMSE detection. Under MMSE criterion, the weight matrix $\mathbf{W}$ is chosen in such a way that the mean-squared-error between the transmitted signal $\mathbf{x}$ and the estimated signal $\hat{\mathbf{x}}$ is minimized. The mean-squared-error (MSE) is defined as MSE $\triangleq E\left[\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right] = E\left[(\mathbf{x} - \mathbf{W}\mathbf{y})^H(\mathbf{x} - \mathbf{W}\mathbf{y})\right]$. The weight matrix $\mathbf{W}$ that minimizes the MSE is

$$\mathbf{W}_{MMSE} = \left(\mathbf{H}^H\mathbf{H} + \frac{\sigma_n^2}{\sigma_x^2}\mathbf{I}\right)^{-1}\mathbf{H}^H, \tag{2.3}$$

It is easy to see that as $\sigma_n^2/\sigma_x^2 \to 0$, the weight matrix $\mathbf{W}_{MMSE}$ approaches $\mathbf{W}_{ZF}$. Since $\mathbf{W}_{MMSE}$ takes noise power into consideration, MMSE detection suffers less from noise enhancement than ZF detection. In [8, 81], it is shown that MMSE is equivalent to ZF in an extended system model, i.e.,

$$\hat{\mathbf{x}}_{MMSE} = \mathbf{W}_{MMSE}\underline{\mathbf{y}} = \underline{\mathbf{H}}^{\dagger}\underline{\mathbf{y}} = (\underline{\mathbf{H}}^H\underline{\mathbf{H}})^{-1}\underline{\mathbf{H}}^H\underline{\mathbf{y}}, \tag{2.4}$$

where

$$\underline{\mathbf{H}} = \begin{bmatrix} \mathbf{H} \\ \frac{\sigma_n}{\sigma_x}\mathbf{I}_{n_t} \end{bmatrix} \text{ and } \underline{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_{n_t \times 1} \end{bmatrix}. \tag{2.5}$$

Comparing (2.2) with (2.4), it follows that the two detection methods can share the same structure in systolic-array implementation, which we shall elaborate upon in Chapter 4.

2.3 Lattice-Reduction-Aided Linear Detection

First, we define the concept of a complex lattice. Let $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n$ be linearly independent vectors in $\mathbb{C}^m$. The *complex lattice* L is a set of points in $\mathbb{C}^m$ defined by

$$L = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i, \quad x_i \in \mathbb{Z} + j\mathbb{Z} \right\}. \tag{2.6}$$

The set of vectors $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n$ is the basis of the lattice. Let the matrix $\mathbf{B}$ consist of these basis vectors as the columns, the lattice L is said to be generated by $\mathbf{B}$ and is often denoted as $L(\mathbf{B})$.

The idea underlying lattice reduction is the selection of a set of basis vectors for the lattice under some goodness criterion [82]. We observe that, under the assumption of QAM transmission, the transmitted vector $\mathbf{x}$ is an integer point of a square lattice (after proper scaling and shifting of the original QAM constellation). By interpreting the columns of the channel matrix $\mathbf{H}$ as a set of lattice basis vectors, $\mathbf{Hx}$ forms the lattice $L(\mathbf{H})$. In lattice theory, if two basis sets $\mathbf{H}$ and $\tilde{\mathbf{H}}$ are related by $\tilde{\mathbf{H}} = \mathbf{H} \cdot \mathbf{T}$, $\mathbf{T}$ a *unimodular* matrix, they generate the same set of lattice points. In MIMO detection, the objective of the lattice reduction algorithm is to derive a better-conditioned channel matrix $\tilde{\mathbf{H}}$. In the following discussion, we focus on the complex-valued LLL algorithm [12, 13]. More details about the LLL algorithm will be provided in Chapter 3.

After lattice-reduction of the channel matrix, we can perform the linear detection, as described in Section 2.2, based on $\tilde{\mathbf{H}}$. Consider ZF first where the weight matrix $\mathbf{W}_{ZF}$ is now $\tilde{\mathbf{H}}^{\dagger}$. The estimated signal $\hat{\mathbf{x}}$ can be written as

$$\begin{aligned}
\hat{\mathbf{x}} &= \mathbf{W}_{ZF}\mathbf{y} \\
&= \mathbf{W}_{ZF}\left[(\mathbf{HT})(\mathbf{T}^{-1}\mathbf{x}) + \mathbf{n}\right] \\
&= \mathbf{T}^{-1}\mathbf{x} + \mathbf{W}_{ZF}\mathbf{n} \qquad (2.7) \\
&= \hat{\mathbf{x}} + \mathbf{W}_{ZF}\mathbf{n}. \qquad (2.8)
\end{aligned}$$

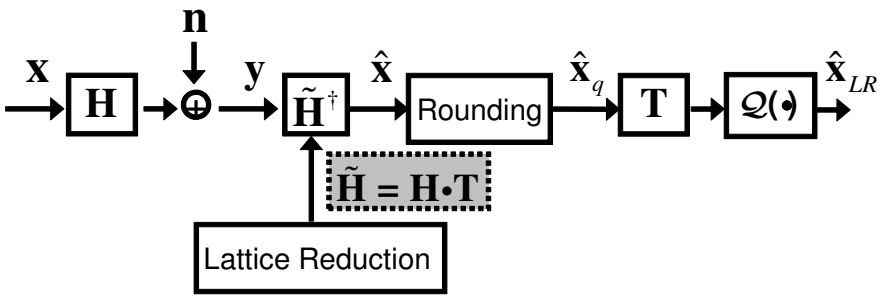


Figure 2.1: Block diagram of linear lattice-reduction-aided detection

Since $\tilde{\mathbf{H}}$ is a better conditioned matrix than $\mathbf{H}$, the noise enhancement problem should be reduced. Without considering the correlated noise $\mathbf{W}_{ZF}\mathbf{n}$, a suboptimal way to estimate $\hat{\mathbf{x}}_q$ is by elementwise slicing, i.e.,

$$(\hat{\mathbf{x}}_q)_i = \arg \min_{z_i \in \mathcal{M}} \left\| \mathbf{W}_{ZF}^{[i]} \mathbf{y} - z_i \right\|, \ i = 1, \cdots, n_t. \tag{2.9}$$

By ignoring the finite size of the Gaussian integer set $\mathcal{M}$ and the boundary effect, $\hat{\mathbf{x}}_q$ is equal to $\lceil \mathbf{W}_{ZF}\mathbf{y} \rfloor = \lceil \hat{\mathbf{x}} \rfloor$. The next step is to transform $\hat{\mathbf{x}}_q$ back into its original domain, which is done by multiplying $\hat{\mathbf{x}}_q$ by the unimodular matrix $\mathbf{T}$. Since the vector entries after the transformation could lie outside the QAM constellation boundary, we finally quantize those points outside the boundary to the closest constellation point, i.e., $\hat{\mathbf{x}}_{LR} = \mathcal{Q}(\mathbf{T}\hat{\mathbf{x}}_q)$. Fig. 2.1 shows the block diagram of LR-aided ZF detection for MIMO. It is easy to see that the same structure can also be used for MMSE detection, by simply replacing $\mathbf{H}$ and $\mathbf{y}$ with the extended matrix $\underline{\mathbf{H}}$ and the vector $\underline{\mathbf{y}}$ defined in (2.5), respectively. The remaining operations are the same as in ZF.

2.4 Lattice-Reduction-Aided Successive Interference Cancellation

Besides being suitable linear detection systolic design can be used to exploit the regularity of successive spatial-interference cancellation (SIC). In [8], it is

shown that LR-aided SIC outperforms linear detection methods, while exhibiting a complexity comparable to linear detection. The LR-aided SIC can be conveniently described in terms of the QR decomposition of the reduced channel matrix. Here we summarize briefly the procedure of LR-aided ZF-SIC only, as the LR-aided MMSE-SIC can be derived in a similar way. Let the QR decomposition of the reduced channel matrix be $\tilde{\mathbf{H}} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}}$. First, multiply $\tilde{\mathbf{Q}}^H$ to $\mathbf{y}$ in (2.1), we obtain

$$\mathbf{v} \triangleq \tilde{\mathbf{Q}}^H\mathbf{y} = \tilde{\mathbf{R}}\mathbf{z} + \tilde{\mathbf{Q}}^H\mathbf{n}, \text{ where } \mathbf{z} = \mathbf{T}^{-1}\mathbf{x}. \tag{2.10}$$

Since $\tilde{\mathbf{R}}$ is an upper triangular matrix, v_{n_t} is only associated with z_{n_t} without the interferences from other transmit substreams. That is,

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n_t} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{R}}_{1:n_t-1,1:n_t-1} & \tilde{\mathbf{R}}_{1:n_t-1,n_t} \\ \mathbf{0} & \tilde{r}_{n_t,n_t} \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n_t} \end{bmatrix} + \tilde{\mathbf{Q}}^H\mathbf{n}. \tag{2.11}$$

Therefore, z_{n_t} can be easily solved by dividing v_{n_t} by $\tilde{r}_{n_t,n_t}$ and then rounding to the closest integer. Before proceeding to the second to last substream, the interference of z_{n_t} should be eliminated from the vector $\mathbf{v}$ using the estimated result $\hat{z}_{n_t}$, so that the updated $\mathbf{v}$,

$$\mathbf{v} := \mathbf{v} - \tilde{\mathbf{R}}_{1:n_t,n_t}\hat{z}_{n_t}, \tag{2.12}$$

is no longer depending on z_{n_t} assuming the estimation $\hat{z}_{n_t}$ is correct. By repeating these steps, we can solve for $\mathbf{z}$ layer by layer starting from the bottom to the top, i.e.

$$\hat{z}_i = \left\lceil \frac{\mathbf{v}_i}{\tilde{r}_{ii}} \right\rfloor \tag{2.13}$$

$$\mathbf{v} := \mathbf{v} - \tilde{\mathbf{R}}_{1:i,i}\hat{z}_i, \tag{2.14}$$

where i starts from n_t to 1 and $\hat{z}_i$ is the estimate of each entry of $\mathbf{z}$. It is clear that if the decision of the first few substreams are wrong, the detection error

will propagate through (2.14). Hence, it is desired to reorder the columns of the channel matrix such that we could start with the layer with the highest signal-to-noise ratio (SNR), then the layer with second highest SNR, and so on. This is the V-BLAST detection [83]. The LR-based V-BLAST detection can achieve close-to-ML performance; however, the computational complexity associated with the column reordering is much higher than other low complexity methods. In [84], the sorted QR decomposition is proposed to combine with SIC detection, which shows a comparable results to V-BLAST while the computational complexity is reduced.

CHAPTER 3

Two Variants of the Lenstra-Lenstra-Lovász Algorithm

In this chapter, we introduce two variants of Lenstra-Lenstra-Lovász (LLL) algorithm which are more time-efficient than the classical LLL algorithm when using parallel processing. Since systolic arrays yield a simple form of parallel processing, our systolic array design for LRAD is based on these two algorithms.

3.1 Lenstra-Lenstra-Lovász Algorithm

LLL algorithm was first introduced in [10] for real-valued lattice. It is further generalized to complex-valued lattice in [12]. The complex LLL algorithm was shown to have lower computational complexity while maintaining the same BER performance as the real LLL algorithm [12, 13]. Here we will only consider the complex LLL algorithm. Let $\mathbf{H}$ (an $n_r \times n_t$ matrix) be a set of lattice basis vectors, with QR decomposition $\mathbf{H} = \mathbf{QR}$. The basis set $\mathbf{H}$ is *complex LLL-reduced* with parameter δ $(1/2 < \delta < 1)$, if the following two conditions are satisfied [12, 13]:

(a)

$$\left|\Re\left(\frac{r_{i,j}}{r_{i,i}}\right)\right| \leq \frac{1}{2}, \quad \left|\Im\left(\frac{r_{i,j}}{r_{i,i}}\right)\right| \leq \frac{1}{2}, \quad 1 \leq i < j \leq n_t, \tag{3.1}$$

(b)

$$\delta - \left|\frac{r_{i-1,i}}{r_{i-1,i-1}}\right|^2 \leq \frac{|r_{i,i}|^2}{|r_{i-1,i-1}|^2}, \; 2 \leq i \leq n_t. \tag{3.2}$$

Algorithm 3.1 Lenstra-Lenstra-Lovász algorithm

Input($\mathbf{Q}$, $\mathbf{R}$)

Initialization $\mathbf{T} = \mathbf{I}_{n_t}$, $k = 2$

while $k \leq n_t$ **do**

Size Reduction

  for $i = k-1, \cdots, 1$ **do**

    $\mu_{i,k} = \lceil r_{i,k}/r_{i,i} \rfloor$

    $\mathbf{R}_{1:i,k} := \mathbf{R}_{1:i,k} - \mu_{i,k}\mathbf{R}_{1:i,i}$

    $\mathbf{T}_{1:n_t,k} := \mathbf{T}_{1:n_t,k} - \mu_{i,k}\mathbf{T}_{1:n_t,i}$

  end for

  if $\delta - \left|\frac{r_{k-1,k}}{r_{k-1,k-1}}\right|^2 > \frac{\left|r_{k,k}\right|^2}{\left|r_{k-1,k-1}\right|^2}$ **then**

  Column Swap

    Swap columns $k-1$ and k in $\mathbf{R}$ and $\mathbf{T}$

  Givens Rotation

    $\eta_1 = r_{k-1,k-1}/\|r_{k-1:k,k-1}\|$

    $\eta_2 = r_{k,k-1}/\|r_{k-1:k,k-1}\|$

    $\mathbf{G} = \begin{bmatrix} \eta_1^+ & \eta_2 \\ -\eta_2 & \eta_1 \end{bmatrix}$

    $\mathbf{R}_{k-1:k,k-1:n_t} := \mathbf{G} \cdot \mathbf{R}_{k-1:k,k-1:n_t}$

    $\mathbf{Q}_{1:n_r,k-1:k} := \mathbf{Q}_{1:n_r,k-1:k} \cdot \mathbf{G}^H$

    $k := \max\{k-1, 2\}$

  else

    $k := k + 1$

  end if

end while

return $\tilde{\mathbf{Q}} = \mathbf{Q}$, $\tilde{\mathbf{R}} = \mathbf{R}$, $\mathbf{T}$

The second condition in (3.2) is called the *Lovász' condition*, and the basis set that satisfies (3.1) is called *size reduced*. To achieve these two conditions, the lattice basis vectors could be processed by LLL algorithm, which is described in Algorithm 3.1.

The input of the algorithm is the QR decomposition of the lattice basis set. The output $\tilde{\mathbf{Q}}$ and $\tilde{\mathbf{R}}$ are the QR decomposition of the reduced lattice basis set $\tilde{\mathbf{H}}$, and $\mathbf{T}$ is the unimodular matrix such that $\mathbf{H} \cdot \mathbf{T}$ is LLL-reduced. As shown in Algorithm 3.1, the algorithm consists of three parts: size reduction, column swap and Givens rotation. Size reduction is to make the basis set satisfy (3.1). Column swap is to make the basis set satisfy (3.2). However, once the columns are swapped, the matrix $\mathbf{R}$ no longer retains the upper-triangular structure. This is why we need Givens rotation after the column swap step. In the standard form of LLL algorithm considered in the literature [8, 10, 12–17], size reduction applies only to one column of $\mathbf{H}$ during a single iteration. Now, systolic arrays, allowing simple parallel processing, are capable of updating the whole matrix without introducing extra delays. Hence, our proposed systolic array is first designed based on the LLL algorithm in a different form, which we call it "LLL algorithm with full size reduction (FSR-LLL)."

3.2 LLL algorithm with Full Size Reduction (FSR-LLL)

Algorithm 3.2 shows the LLL algorithm with full size reduction. In the following discussion, we refer to the lines in Algorithm 3.2. There are three main differences between FSR-LLL and the conventional complex LLL algorithm, although the lattice reduced bases from both algorithms are still the same. First, the full size reduction (lines 3 to 9) is executed in each iteration of the *while* loop (line 2), which means that all columns of $\mathbf{R}$ and $\mathbf{T}$ are size-reduced at the beginning of each iteration. While in LLL algorithm, only one column is size reduced in each iteration of the *while* loop. The advantage here is that, once condition (3.2) is also fulfilled after full size reduction (i.e., no k' is found in line 10), then the FSR-LLL can immediately end the process (line 20). For example, suppose that k equals 3 at current iteration. Since all columns in $\mathbf{R}$ and $\mathbf{T}$ are size-reduced after full size

Algorithm 3.2 FSR-LLL algorithm

Input($\mathbf{Q}^H,\ \mathbf{R}$)

Initialization $\mathbf{T} = \mathbf{I}_{n_t},\ k = 2$

while $k \leq n_t$ **do**

Full Size Reduction

  for $j = n_t, \cdots, 2$ **do**

    for $i = j-1, \cdots, 1$ **do**

      $\mu_{i,j} = \lceil r_{i,j}/r_{i,i} \rfloor$

      $\mathbf{R}_{1:i,j} := \mathbf{R}_{1:i,j} - \mu_{i,j}\mathbf{R}_{1:i,i}$

      $\mathbf{T}_{1:n_t,j} := \mathbf{T}_{1:n_t,j} - \mu_{i,j}\mathbf{T}_{1:n_t,i}$

    end for

  end for

  Find the smallest $k'(k \leq k' \leq n_t)$ such that $\delta - \left|\frac{r_{k'-1,k'}}{r_{k'-1,k'-1}}\right|^2 > \frac{\left|r_{k',k'}\right|^2}{\left|r_{k'-1,k'-1}\right|^2}$

  if k' exists **then**

    Givens Rotation

    $\eta_1 = r_{k'-1,k'}/\|r_{k'-1:k',k'}\|$

    $\eta_2 = r_{k',k'}/\|r_{k'-1:k',k'}\|$

    $\mathbf{G} = \begin{bmatrix} \eta_1^+ & \eta_2 \\ -\eta_2 & \eta_1 \end{bmatrix}$

    $\mathbf{R}_{k'-1:k',k'-1:n_t} := \mathbf{G} \cdot \mathbf{R}_{k'-1:k',k'-1:n_t}$

    $\mathbf{Q}^H_{k'-1:k',1:n_r} := \mathbf{G} \cdot \mathbf{Q}^H_{k'-1:k',1:n_r}$

    Column Swap

    Swap columns $k'-1$ and k' in $\mathbf{R}$ and $\mathbf{T}$

    $k := \max\{k'-1, 2\}$

  else

    return $\tilde{\mathbf{Q}}^H = \mathbf{Q}^H,\ \tilde{\mathbf{R}} = \mathbf{R},\ \mathbf{T}$

  end if

end while

reduction, if no k' can be found in line 10 (a search that a systolic array can make in parallel), then no further processing is needed in FSR-LLL. However, in the conventional LLL format, the process will end until columns 3 to n_t are sequentially size-reduced. With a systolic-array implementation, FSR-LLL is faster, and its efficiency is especially apparent when n_t is large. The second difference is that the Givens rotation (lines 12 to 16) is executed before the column swap (line 17). This is because the Givens rotation process can work in parallel with full size reduction, whereas the columns swap cannot. This point will be made clear in Section 4.1. To accommodate this change, the $\mathbf{G}$ matrix in line 14 is also modified from the one in line 12 of Algorithm 3.1. Third, the QR decomposition $\mathbf{Q}^H\mathbf{H} = \mathbf{R}$ is considered as the input of the algorithm, instead of $\mathbf{H} = \mathbf{QR}$. From lines 15 and 16, the Givens rotation matrix $\mathbf{G}$ applies to the same two rows of $\mathbf{Q}^H$ and $\mathbf{R}$, which simplifies the design of the systolic array. Additionally, after FSR-LLL, $\tilde{\mathbf{Q}}^H$ is ready for calculating the pseudoinverse of $\tilde{\mathbf{H}}$ for linear-detection.

3.3 All-Swap Lattice Reduction (ASLR) Algorithm

The ASLR algorithm is a variant of the LLL algorithm, and was first proposed for real number lattices only [39]. Algorithm 3.3 describes its extension to a complex version. One significant difference between FSR-LLL and ASLR is that every pair of columns k and $k-1$ with even (or odd) index k could be swapped simultaneously. The algorithm begins with full size reduction, which is the same as FSR-LLL. Givens-rotation and column-swap operations (same as in Algorithm 3.2, lines 12 to 16) should be executed on all possible even (odd) k that violate the condition in (3.2), and then start another iteration with the indicator variable "*order*" set to odd (even). If condition (3.2) holds for all even (odd) k, Givens rotation and columns swap will not be executed. Meanwhile, we can immediately check for all odd (even) k instead. Matrix $\mathbf{R}$ is already full-size reduced, with no need to start

Algorithm 3.3 ASLR algorithm

Input($\mathbf{Q}^H,\ \mathbf{R}$)

1: Initialization $\mathbf{T} = \mathbf{I}_{n_t}$

2: *order* = EVEN

Full Size Reduction

3: Execute lines 3 to 9 in Algorithm 3.2

4: **while** Any swap is possible in lines 6 and 10 **do**

Givens Rotation and Column Swap

5: **if** *order* = EVEN and $\delta - \left|\frac{r_{k-1,k}}{r_{k-1,k-1}}\right|^2 > \frac{\left|r_{k,k}\right|^2}{\left|r_{k-1,k-1}\right|^2}$ for any even k **then**

6: Execute lines 12 to 16 in Algorithm 3.2 for those even k such that $\delta - \left|\frac{r_{k-1,k}}{r_{k-1,k-1}}\right|^2 > \frac{\left|r_{k,k}\right|^2}{\left|r_{k-1,k-1}\right|^2}$

7: Perform FSR

8: *order* = ODD

9: **else if** *order* = ODD and $\delta - \left|\frac{r_{k-1,k}}{r_{k-1,k-1}}\right|^2 > \frac{\left|r_{k,k}\right|^2}{\left|r_{k-1,k-1}\right|^2}$ for any odd k **then**

10: Execute lines 12 to 16 in Algorithm 3.2 for those odd k such that $\delta - \left|\frac{r_{k-1,k}}{r_{k-1,k-1}}\right|^2 > \frac{\left|r_{k,k}\right|^2}{\left|r_{k-1,k-1}\right|^2}$

11: Perform FSR

12: *order* = EVEN

13: **end if**

14: **end while**

15: **return** $\tilde{\mathbf{Q}}^H = \mathbf{Q}^H,\ \tilde{\mathbf{R}} = \mathbf{R},\ \mathbf{T}$

the next iteration with full size reduction. If neither an even nor odd k violates condition (3.2) after full size reduction, the ASLR process ends.

3.4 Replacing the Lovász' condition with the Siegel's condition

From the previous discussion, it is clear that all basis vectors are size reduced within one processing iteration of full size reduction. Additionally, according to line 10 in Algorithm 3.2 and lines 5 and 9 in Algorithm 3.3, the lattices processed by FSR-LLL and ASLR both satisfy the Lovász' condition in (3.2). Therefore, we can conclude that these two algorithms also generate LLL-reduced lattice. Consequently, like the conventional LLL, FSR-LLL-aided and ASLR-aided detections also achieve full receive diversity in MIMO system [13, 14].

The Lovász' condition involves two diagonal elements and one off-diagonal element in the matrix $\mathbf{R}$. In order to simplify the data communication between processing elements in the systolic array, we relax the Lovász' condition by replacing it with

$$\delta - \frac{1}{2} \leq \frac{|r_{i,i}|^2}{|r_{i-1,i-1}|^2},\ 2 \leq i \leq n_t, \tag{3.3}$$

where δ lies in the range $(1/2, 1)$, the same as for the Lovász' condition. The condition (3.3) is also called the Siegel's condition [40], and it is weaker than the Lovász' condition because

$$\delta - \frac{1}{2} \leq \delta - \left|\frac{r_{i-1,i}}{r_{i-1,i-1}}\right|^2 \leq \frac{|r_{i,i}|^2}{|r_{i-1,i-1}|^2},\ 2 \leq i \leq n_t. \tag{3.4}$$

The first inequality follows from (3.1). Similar approximation as in (3.3) can be found in [85]. The advantage of using this new condition is that only two neighboring diagonal elements of $\mathbf{R}$ are involved. We will have more discussion on the impact of designing systolic array with this new condition in Chapter 4. Another advantage comes from the fact that the new condition check can be done by taking

the square-root in (3.3). In hardware implementation, it implies that we can save precision bits by storing $|r_{i,i}|/|r_{i-1,i-1}|$ rather than $|r_{i,i}|^2/|r_{i-1,i-1}|^2$. Additionally, the condition check can be done without a division, simply by comparing the value of $|r_{i,i}|$ and $\sqrt{\delta - 1/2}\,|r_{i-1,i-1}|$, where $\sqrt{\delta - 1/2}$ is a pre-computed constant once δ is determined. In the balance of the discussion, when we refer to FSR-LLL and ASLR we mean FSR-LLL and ASLR *with the Siegel's condition*.

Since the Siegel's condition is weaker than the Lovász' condition, one might expect the performance of the lattice reduction algorithm with condition (3.3) to be worsened. Yet, by a proof similar to that in [13, 14] we can show that the LLL algorithm with the Siegel's condition also achieves maximum receive diversity in MIMO systems. In the proof of the LLL-aided detection achieving full diversity [13, 14], the key step and the only step involving the LLL-reduced conditions is that the orthogonality defect κ ($\kappa \geq 1$) of the LLL-reduced basis set $\mathbf{H}$ is upper bounded by

$$\kappa \triangleq \frac{\prod_{i=1}^{n_t} \|\mathbf{h}_i\|^2}{\det\left(\mathbf{H}^H\mathbf{H}\right)} \leq 2^{-n_t}\left(\frac{2}{2\delta - 1}\right)^{\frac{n_t(n_t+1)}{2}}, \tag{3.5}$$

where $\mathbf{h}_i$'s are the columns of $\mathbf{H}$. In particular, (3.5) also holds for the lattices reduced by the LLL algorithm with the Siegel's condition. This can be justified by the same proof as in [13, Appendix B], whose details will be omitted in this discussion. Hence, the LLL algorithm with the Lovász' condition replaced by the Siegel's condition also achieves maximum diversity in MIMO system. However, achieving maximum receive diversity does not automatically imply that the bit-error-rate (BER) performance is as good as using the conventional LLL algorithm. One can easily observe that if δ is very close to 1/2 , condition (3.3) is almost always true. Thus, the Givens rotation and column swap steps in the reduction algorithm would seldom be performed, which causes the BER performance to be much worse than with conventional LLL. On the contrary, as δ approaches 1 one can expect the performance of the FSR-LLL and the ASLR to be closer to

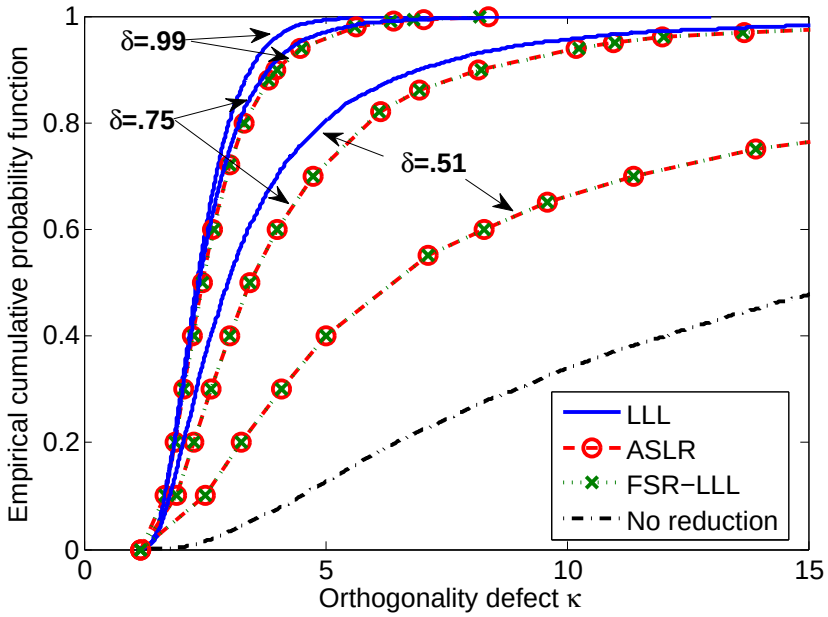


Figure 3.1: The empirical cumulative probability functions of the orthogonality defect κ for the 4×4 channel matrices under three different reduction algorithms.

the conventional LLL. In Fig. 3.1, we show the empirical cumulative probability functions of the orthogonality defect κ for 4×4 channel matrices under three different reduction algorithms. The results of the FSR-LLL and the ASLR overlap for all three values of δ, which implies that the effects of these two methods on lattice reduction are almost the same. At $\delta = 0.99$, the FSR-LLL and the ASLR give results close to the LLL with $\delta = 0.75$, which is a very common setting as documented in previous works [8, 10, 14]. For $\delta = 0.51$ and 0.75, the gap between the LLL and the FSR-LLL (ASLR) is much larger than for $\delta = 0.99$. In section 4.3, we will show that for δ equal to 0.99, the BER performance of LR-aided linear detections using the FSR-LLL and the ASLR are not worse than the one using the conventional LLL with the same δ value. Based on these results, in our systolic array design we choose $\delta = 0.99$.

CHAPTER 4

Systolic Array Design for MIMO LLL-Aided Detection

From Fig. 2.1, the whole process of the LRAD can be viewed as taking two steps: lattice reduction for the channel matrix and detection. In this section, we first exhibit our systolic array design for the FSR-LLL and the ASLR, in Sections 4.1 and 4.2, respectively. The ensuing linear detection or SIC on systolic array will be discussed in Section 4.4. The performance comparison between the FSR-LLL and the ASLR is discussed in Section 4.3.

In the following discussion, we assume that the channel matrix has been QR decomposed without considering QR decomposition in the proposed systolic array design. It is known that QRD can be implemented in systolic array based on a series of Givens rotations, since Given rotations can be executed in a parallel manner [29–31]. Since the conventional systolic array for QRD usually contains square root operations, which are computationally intensive in hardware implementation, a square-root-free systolic QRD based on Squared Givens rotations (SGR) can be used (the interested readers can refer to [38, 86]). In [8], it is also shown that the sorted QRD (SQRD) can reduce the number of column swaps in the LLL algorithm, and hence leads to less processing time. However, it also requires higher hardware complexity and latency to implement SQRD than the conventional QRD [87].

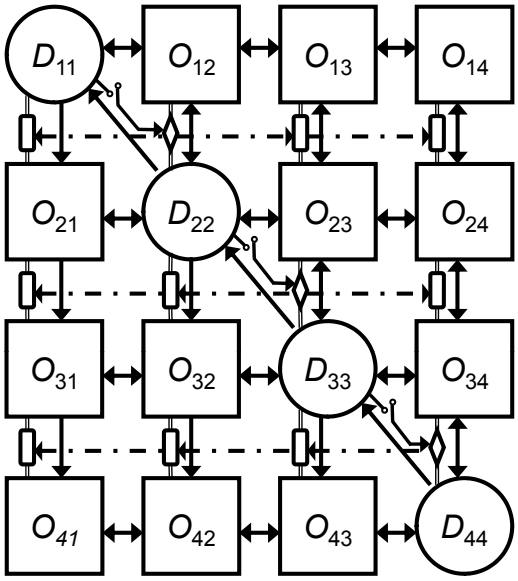
4.1 Systolic Array for the FSR-LLL algorithm

In the following, we assume a 4×4 MIMO system (i.e., $n_t = 4$, $n_r = 4$) and illustrate the proposed systolic algorithm in three parts: full size reduction, Givens rotation, and column swap.

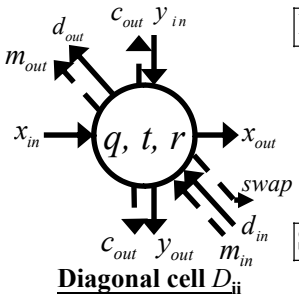
4.1.1 Full Size Reduction

The systolic array for the remaining parts of the LRAD is shown in Fig. 4.1(a) . Four different kinds of PEs are used, viz., diagonal cells, off-diagonal cells, vectoring cells, and rotation cells. For the full size reduction part, only diagonal and off-diagonal cells are needed: the operations of these two types of PEs are shown in detail in Fig. 4.1(b). The vectoring cell and rotation cell will be introduced with the Givens rotation description. There is a slight difference between the off-diagonal cells in the upper-triangle part and those in the lower-triangle part. Fig. 4.1(b) shows only the off-diagonal cell in the upper-triangle part. Those off-diagonal cells in the lower-triangle part have y_{in} and c_{in} come from the top, while c_{out} leaves from the bottom. Except for this minor difference in the data interface, the operations are the same as the off-diagonal cells in the upper-triangle part. Additionally, in Fig. 4.1(b) the dotted lines represent the logic control signals transmitted between cells, and the solid lines represent the data transmitted. To initialize the process, each element of the matrices $\mathbf{R}$ and $\mathbf{Q}^H$ (denoted as r and q, respectively, in Fig. 4.1(b)) from QR decomposition are stored in the PE at the corresponding position. For example, $q_{i,i}$ and $r_{i,i}$ are stored in the corresponding diagonal cell D_{ii}. The off-diagonal elements $q_{i,j}$ and $r_{i,j}$ are stored in the off-diagonal cell O_{ij}. Additionally, the elements of the unimodular matrix $\mathbf{T}$ (denoted as t in Fig. 4.1(b)) are also stored in the arrays, with $\mathbf{T}$ initially set to the identity matrix.

Fig. 4.2 shows the overall processes of the full size reduction in the systolic array. In this stage, two major processing modes are defined in each diagonal and off-



(a)



Data mode *(If m_{in} = "#")*

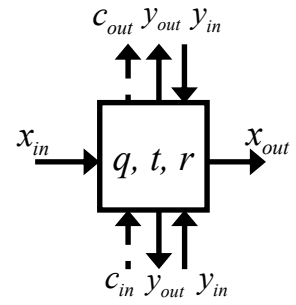
$m_{out} = c_{out} = \text{"\#"}$

$d_{out} = r;\ \ x_{out} = (r,t)^{(*)}$

$$swap = \begin{cases} \text{"true"}, \text{ if } |d_{in}|^2 / |r|^2 < \delta - \frac{1}{2} \\ \text{"false"}, \quad otherwise \end{cases}$$

Size Reduction mode *(Default)*

$t := t - y_{in} \cdot x_{in};\ \ y_{out} = y_{in},\ x_{out} = x_{in}$



Data mode *(If c_{in} = "#")*

$x_{out} = (r,t);\ \ c_{out} = c_{in}$

Size Reduction mode *(Default)*

If x_{in} carries "*":

$\mu = \lceil r / x_{in} \rfloor$

$r := r - \mu \cdot x_{in},\ t := t - \mu \cdot x_{in}$

$y_{out} = \mu;\ \ x_{out} = x_{in}$

If x_{in} doesn't carry "*":

$r := r - y_{in} \cdot x_{in},\ t := t - y_{in} \cdot x_{in}$

(b)

Figure 4.1: (a) The systolic array for the linear LRAD of an 4×4 MIMO system. (b) The operations of the diagonal and the off-diagonal cells in the systolic array.

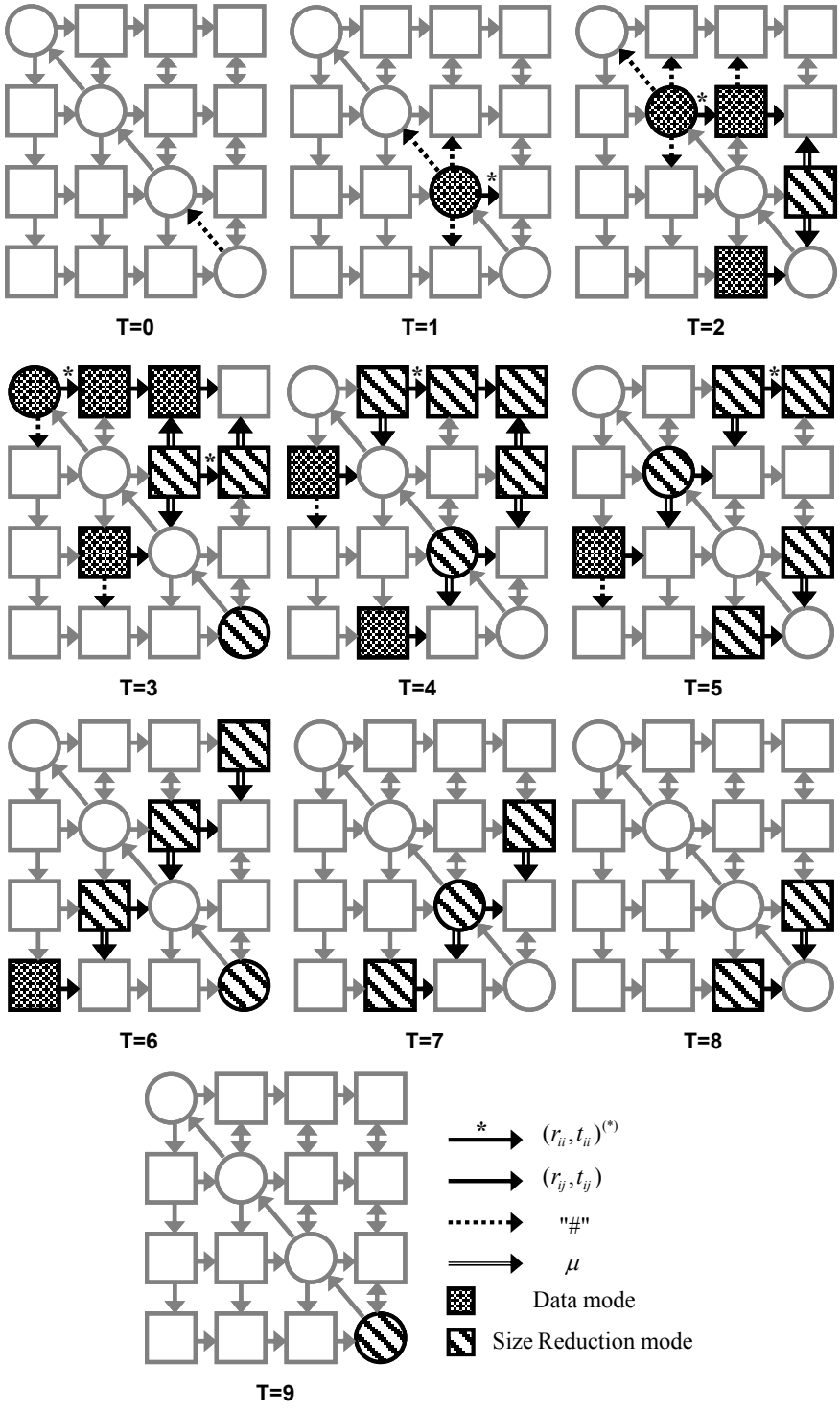


Figure 4.2: Flow chart of the full size reduction operations in the systolic array.

diagonal cell, the *size reduction mode* and the *data mode* as detailed in Fig. 4.1(b). In the *size reduction mode*, the objective of each cell is to make condition (3.1) valid. On the other hand, the cell only performs data propagation in the *data mode*. The cell decides to work in either mode depending on the occurrence of the logic control signal "#". For simplicity, we assume the cells execute all operations in the *data mode* or the *size-reduction mode* in one normalized cycle[1]. At $T = 0$, the external controller sends in the logic control signal "#" to cell D_{33} through cell D_{44}. At $T = 1$, cell D_{33} works in the *data mode* due to the control signal "#" and spreads out the "#" logic control signal to the neighboring 3 cells. Meanwhile, D_{33} sends out the data $(r_{3,3}, t_{3,3})^{(*)}$ to cell O_{34}. Note that the superscript (*) is a tag bit attached to the data, which indicates that the data are sent out by a diagonal cell. The occurrence of a tag bit (*) will drive the off-diagonal cell to compute μ, and use μ to update the data stored in that cell. As a result, at $T = 2$, cell O_{34} sends out the newly computed μ to the two neighboring cells O_{24} and D_{44}. At next time instant ($T = 3$), the μ signal generated by O_{34} meets the data coming from cell O_{23} (O_{43}) inside the cell $O_{24}(D_{44})$, and executes the size reduction update. At the same time instant, data $(r_{2,2}, t_{2,2})^{(*)}$ enter cell O_{23}. As cell O_{34} did at $T = 2$, cell O_{23} computes μ, updates $(r_{2,3}, t_{2,3})$, and sends out μ to the neighboring cells O_{13} and D_{33}. The most important fact here is that cell O_{23} also propagates the data $(r_{2,2}, t_{2,2})^{(*)}$ to cell O_{24}, and thus starts the column operations between column 2 and column 4 at $T = 4$. Similarly, the column operations between column 1 and column 4 begins at $T = 6$ as $(r_{2,2}, t_{2,2})^{(*)}$ enter cell O_{14}. Essentially, full size reduction is a series of column operations between column j and columns $j-1,\ j-2, \cdots,\ 1$, for all $2 \leq j \leq n_t$, and we can conclude the following facts for an $n_t \times n_t$ MIMO system:

Fact 1. *In this systolic flow, the column operation between column j and column i $(i < j)$ begins at $T = n_t + j - 2i$ as $(r_{i,i}, t_{i,i})^{(*)}$ enters cell O_{ij}.*

[1]The real hardware cycle counts could be multiples of the normalized cycle.

Proof. Data $(r_{i,i}, t_{i,i})^{(*)}$ leaves cell D_{ii} at $T = n_t - i$, and it takes $j - i$ cycles to have $(r_{i,i}, t_{i,i})^{(*)}$ propagates from cell D_{ii} to cell O_{ij}. □

Fact 2. *All column operations on column j end at $T = 2n_t + j - 3$ in cell $O_{n_t j}$.*

Proof. In this systolic flow, the last column operation on column j is always between column j and column 1, which starts at $T = n_t + j - 2$ in cell O_{1j} according to fact 1. It takes $n_t - 1$ more cycles to propagate μ from cell O_{1j} to cell $O_{n_t j}$ and finish the column operation. □

Fact 3. *The full size reduction ends at $T = 3n_t - 3$, when all updates on column n_t are done.*

Proof. The full size reduction ends when column n_t finish all the column operations. Therefore, it follows the result in fact 2 that the last step is at $T = 3n_t - 3$. □

Referring back to the example mentioned in Section 3.2, we can have a more concrete view about the advantage of the FSR-LLL over the conventional LLL form when a systolic array is used. If the FSR-LLL is applied, the systolic array takes a total of $3n_t - 3$ cycles to end the all processes. However, for non-systolic LLL, it takes $2n_t + j - 3$ to process column j, and all column operations cannot be done in parallel. So the total time to perform size reduction in non-systolic LLL would be $\sum_{j=3}^{n_t} (2n_t + j - 3) = 2.5n_t^2 - 6.5n_t + 3$ cycles in that example. In this case, as n_t increases beyond 3, the advantage of the FSR-LLL over the conventional format becomes significant.

4.1.2 Givens Rotation

As mentioned in Section 3.4, we use the Siegel's condition in the lattice reduction algorithm, which only relates two r elements in the neighboring diagonal cells. Hence, this condition can be checked during a full size reduction step. For example, in Fig. 4.2 at $T = 1$, cell D_{33} sends data $r_{3,3}$ to cell D_{22} along with the "#" signal.

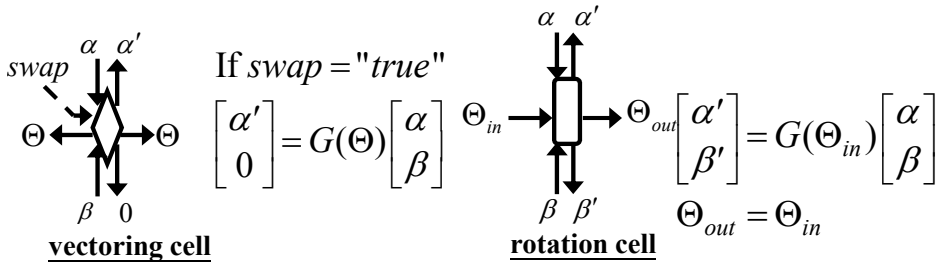


Figure 4.3: The operations of the vectoring cell and the rotation cell in the systolic array.

At the next time instant, cell D_{22} will check this condition based on $|r_{3,3}|^2/|r_{2,2}|^2$, and also generate the logic control signal "*swap*" (see Fig. 4.1(b)). If $\delta - 1/2$ is greater than $|r_{i,i}|^2/|r_{i-1,i-1}|^2$ then "*swap*" is "*true*", and drives the vectoring cell to work. The operations of vectoring and rotation cells are shown in Fig. 4.3. The vectoring cell zeros out the input data β by the Givens rotation matrix $\mathbf{G}$, which is calculated based on Algorithm 3.2 lines 12 to 14. The rotation cell simply rotates the input data with the angle Θ given by the neighboring vectoring cell. Hence, the vectoring and rotation cells also work in a systolic way, with the rotation angle Θ propagating between cells. As shown in Fig. 4.1(a), there are 3 rotation cells and 1 vectoring cell between every two consecutive rows of the systolic array. These cells perform the Givens rotation to the $\mathbf{R}$ and $\mathbf{Q}^H$ data in those two rows. The vectoring cell is located between cells D_{ii} and $O_{i-1,i}$ because the Givens rotation step is executed prior to the column-swap step in the FSR-LLL, and data $r_{i,i}$ need be zeroed so that the matrix $\mathbf{R}$ is still upper triangular after column swap.

Note that Givens rotation only applies to rows k' and $k'-1$ during one iteration of the FSR-LLL if k' exists (lines 12 to 16 in Algorithm 3.2). However, every D_{ii} $(i = 1, \cdots, n_t - 1)$ could generate the "*swap*" signal during the full size reduction step. Therefore, we need a direct access from the external controller to each diagonal cell in order to control the data path between the diagonal cell and the vectoring cell. Namely, only cell $D_{k'k'}$ can pass the signal "*swap*" to the vectoring

cell and perform the Givens rotation to rows k' and $k'-1$. In Fig. 4.1(a), we use a "*switch*" symbol between each pair of a diagonal cell and a vectoring cell to represent the control by the external controller. Only one switch is turned on during one iteration.

Additionally, a Givens rotation on rows k' and $k'-1$ can begin right after $r_{k'-1,k'}$ is updated during the full size reduction step. For example, $r_{3,4}$ is updated at $T=2$ as shown in Fig. 4.2, and Givens rotation on rows 3 and 4 could start as early as $T=3$ without any interference to the remaining operations of full size reduction. This way, the time necessary to perform Givens rotations can be partially hidden by the full size reduction and this is the reason why we want the Givens rotation to occur prior to column swap in our design.

For hardware implementation, one could consider using only one rotation cell between every two neighboring rows to reduce the hardware complexity. For example, in Fig. 4.1(a) one rotation cell can be used between the third and the fourth rows to connect to O_{31}, O_{32}, D_{33} and O_{41}, O_{42}, O_{43}. The rotation cell still works in a pipelining manner and deals with two cells one at a time. This will not lead to significant increase in time if we consider performing Givens rotation and full size reduction in parallel.

4.1.3 Column Swap

The columns k' and $k'-1$ of $\mathbf{R}$ (and $\mathbf{T}$) should be swapped, after the Givens rotation is done. However, it is possible that the column swap be partially overlapped in time with size reduction and Givens rotation. For example, the column swap could begin after $\mathbf{R}$ being rotated but prior to $\mathbf{Q}^H$ being updated since there is no need to swap columns of $\mathbf{Q}^H$.

The FSR-LLL stops when there is no possible column swap, i.e., a k' in Algorithm 3.2, line 10, does not exist. The system flow (lines 2 and 18 in Algorithm 3.2)

is controlled by the external processor. The lattice reduced matrices $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{Q}}^H$ and the unimodular matrix $\mathbf{T}$ stay in the PEs. The systolic array along with these matrices will be used for linear detection, as described in Section 4.4 below.

4.2 All-Swap Lattice Reduction (ASLR) Algorithm

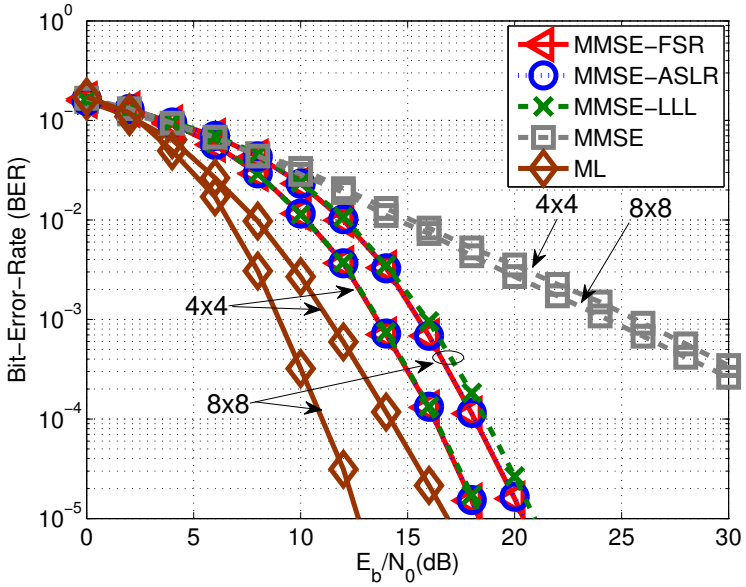
The ASLR algorithm can also be performed by the systolic array shown in Fig. 4.1(a). The process of full size reduction is the same as in Fig. 4.2. During full size reduction, the Siegel's condition is also checked in each diagonal cell $D_{11} \sim D_{n_t-1,n_t-1}$. If the current value of "*order*" is even (odd), then the "*switch*" between each cell $D_{k-1,k-1}$ with even (odd) index k and the vectoring cell is turned on by the external controller. Consequently, for every even (odd) index k, Givens rotation between rows $k-1$ and k could be executed if needed. As for the column swap step, more than one pair of columns could be swapped during one iteration, but all these pairs are swapped in parallel. Hence, the time spent on columns swap is the same as on swapping a single pair of columns. Based on this observation, we can expect the systolic ASLR to work more efficient than the systolic FSR-LLL. Comparisons between these two algorithms in terms of bit-error-rate performance and of efficiency in execution time are deferred to the next subsection.

Note that in our description we limit the applications of this systolic array only to an $n_t \times n_t$ MIMO system. For $n_t \times n_t$ MMSE-LRAD, although the matrix $\mathbf{Q}^H$ is $n_t \times 2n_t$ (the extended channel model in (2.5)), we can treat the submatrix $\mathbf{Q}^H_{1:n_t,(n_t+1):2n_t}$ as another square matrix, and store each element of $\mathbf{Q}^H_{1:n_t,(n_t+1):2n_t}$ in the PE at the corresponding position. Namely, $q_{i,j}$ and $q_{i,j+n_t}$ should be stored in the same PE, which still keeps the systolic array square.

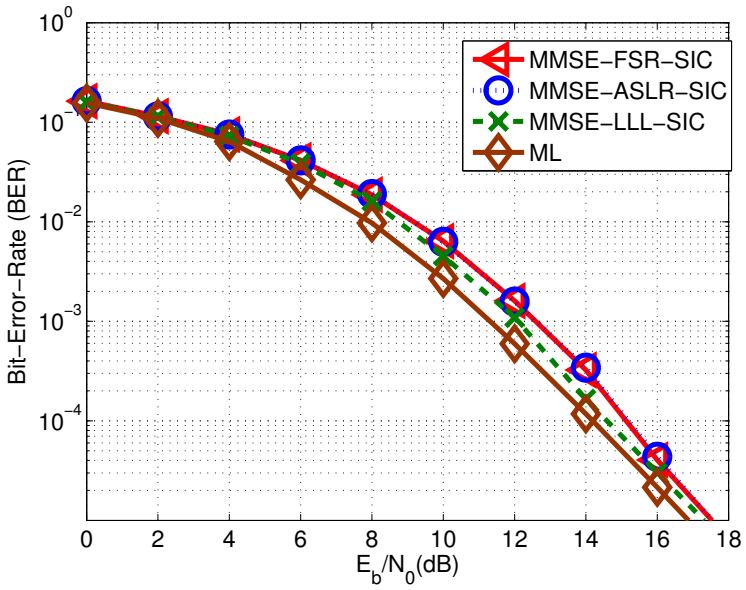
4.3 Comparison between the FSR-LLL and the ASLR algorithms

First, we compare the two algorithms in term of bit-error-rate (BER) performance, and also compare them with the conventional LLL algorithm. In our simulation, 4-QAM is assumed for the transmitted symbols. The constant δ is set to 0.99 in all algorithms for fair comparison. Let E_b be defined as the equivalent energy per bit at the receiver, and thus E_b/N_0 is $n_t\sigma_x^2/(\sigma_n^2 \log_2 M)$. Fig. 4.4(a) shows the BER results of minimum mean-square-error LRAD (in 4×4 and 8×8 MIMO systems) based on the FSR-LLL (denoted as MMSE-FSR), the ASLR algorithm (denoted as MMSE-ASLR) and the LLL algorithm (denoted as MMSE-LLL). The BER results for ML detection and MMSE without lattice reduction are also shown for comparison. As $\delta = 0.99$, the FSR-LLL and the ASLR work as well as the LLL algorithm, and even slightly better in the case of $n_t = 8$. It clearly shows that using the insignificantly weaker Siegel's condition does not deteriorate the BER performance of linear detections in an MIMO system as compared to the conventional LLL. In Fig. 4.4(b), the BER performance of an 4×4 MIMO system using the LR-aided MMSE SIC based on different lattice reduction algorithms are shown. Unlike the linear detection case, the LLL-aided SIC works better than the other two algorithms. Since the detection of the first layer in SIC dominates the overall performance, it implies that due to the Siegel's condition the FSR-LLL-reduced or the ASLR-reduced channel provides lower SNR for the first layer in SIC than the one given by the conventional LLL. Additionally, FSR-LLL and ASLR lead to almost the same results in all three MIMO systems, which is consistent with the results in Fig. 3.1. Hence, we can conclude that although the FSR-LLL and the ASLR give different lattice reduced matrices, the LRAD based on these two algorithms have very similar BER performances.

Next, we compare the efficiency of the systolic array for both algorithms. It



(a)



(b)

Figure 4.4: BER performance of the FSR-LLL and the ASLR- based MMSE LRADs. (a)Linear detection (4×4 and 8×8 MIMO systems) (b)SIC (an 4×4 MIMO system)

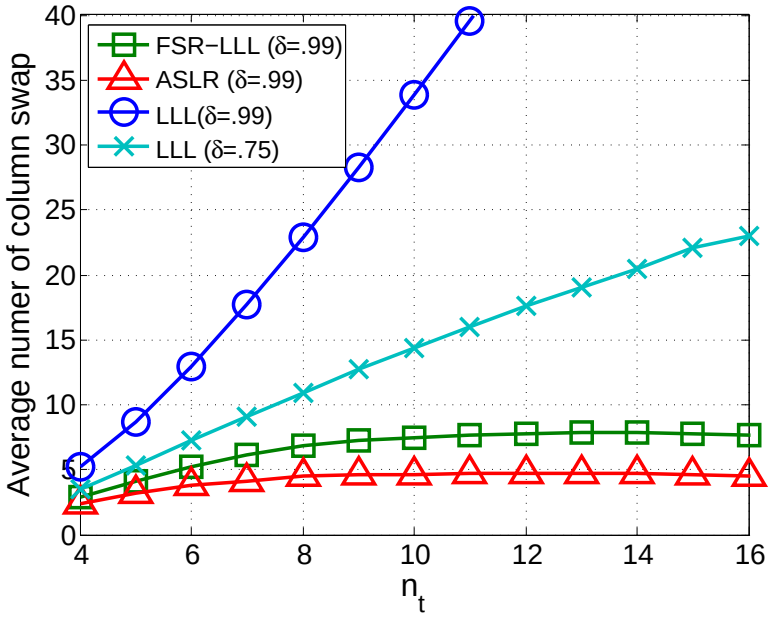


Figure 4.5: The average number of column swaps in the FSR-LLL, the ASLR and the LLL-aided MMSE detections in an $n_t \times n_t$ MIMO system with E_b/N_0 fixed at 20 dB.

is known that the number of iterations of the FSR-LLL and the ASLR depends on the condition number of the channel matrix. If $\mathbf{H}$ is well-conditioned, lattice reduction takes less iterations, and thus less cycles in the systolic array. Since both algorithms begin with full size reduction, the total execution time is fully determined by the number of column swaps in the overall process. Less column swapping implies less iterations. Fig. 4.5 shows the average number of column swaps in the FSR-LLL and the ASLR-aided MMSE detections (with E_b/N_0 fixed at 20dB) in $n_t \times n_t$ MIMO systems ($n_t = 4 \sim 16$). Note that for the ASLR we count all the even or odd columns swaps during one iteration as only one swap since they are executed in parallel. In an 4×4 MIMO, the difference between the two algorithms is almost negligible. However, as the number of antennas grows, the advantage of the ASLR becomes significant. For $n_t \geq 8$, the ASLR has less than 65% the column swaps compared to the FSR-LLL. Based on BER performance and time-efficiency comparisons, the ASLR should be a better algorithm to be

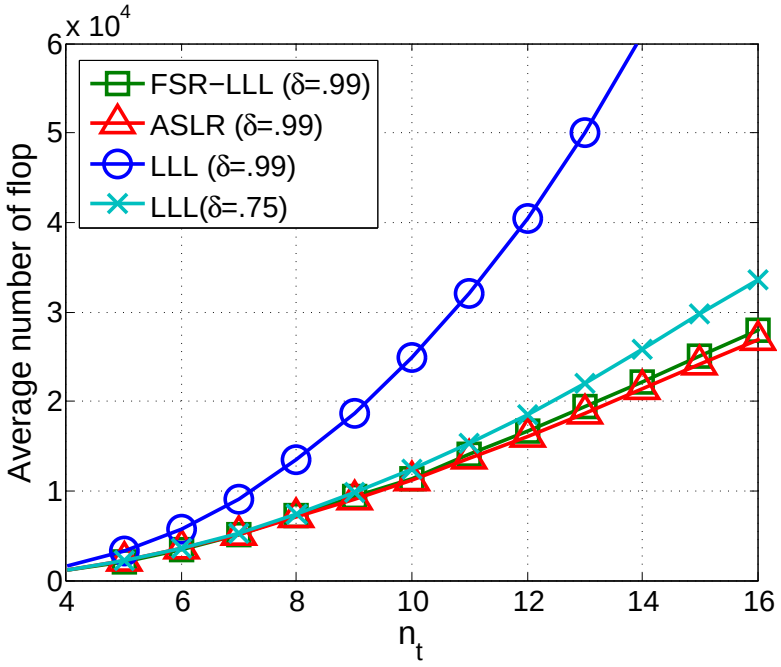


Figure 4.6: The average number of floating point operations in the FSR-LLL, the ASLR and the LLL-aided MMSE detections in an $n_t \times n_t$ MIMO system with E_b/N_0 fixed at 20 dB.

applied on our systolic array, especially with a large number of antennas.

For comparison, the results of the conventional LLL with $\delta = 0.99$ and 0.75 are also shown in Fig. 4.5. As expected, the LLL with $\delta = 0.99$ has a higher complexity than the LLL with $\delta = 0.75$. Furthermore, the conventional LLL has a much higher average number of column swaps than the FSR-LLL and the ASLR have in the higher-dimensional MIMO system ($n_t \geq 8$). However, it is not fair to conclude that the complexities of the FSR-LLL and the ASLR are much lower than the conventional LLL; in fact, full size reductions are performed in the former two algorithms, and full size reduction needs more computation efforts than the conventional size reduction in the LLL. In Fig. 4.6, we compare the number of floating point operations (flop) in the LLL, the FSR-LLL, and the ASLR using the same settings as in Fig. 4.5. The flops are counted in terms of number of real additions and real multiplications. One complex addition is counted as two flops

(two real additions) and one complex multiplication is counted as six flops (four real multiplications and two real additions). The complexity of the QR decomposition is neglected, since this is done only once at the beginning of the three algorithms. It is shown that the LLL with $\delta = 0.99$ has the highest complexity among the three. Under the same δ ($= 0.99$) setting, the FSR-LLL and the ASLR have a much lower computational complexity than the LLL. On the other hand, the complexity of the LLL with $\delta = 0.75$ is just slightly higher than the FSR-LLL and the ASLR, even though the average number of column swaps of the LLL with $\delta = 0.75$ is more than two times larger than the one of the ASLR for $n_t \geq 10$. This implies that the process of full size reduction introduces some additional complexity. However, thanks to the (insignificantly) weaker Siegel's condition, the complexities of the ASLR and the FSR-LLL for $n_t \geq 10$ are less than 50% of the complexity of the LLL with the same δ setting.

To further explore the advantage of using systolic array, we implement our proposed architecture for an 4×4 MIMO system onto a FPGA. The FPGA design and emulation results will be introduced in Chapter 5.

4.4 Systolic Array for Detection Methods

4.4.1 Linear Detection in Systolic Array

After lattice reduction, the matrices $\tilde{\mathbf{Q}}^H$ and $\tilde{\mathbf{R}}$, along with the unimodular matrix $\mathbf{T}$, are stored in the systolic array. As shown in Fig. 2.1, the first step of a linear detection consists of premultiplying the received signal vector $\mathbf{y}$ by $\tilde{\mathbf{H}}^\dagger$, which yields $\hat{\mathbf{x}} = \tilde{\mathbf{H}}^\dagger \mathbf{y} = \tilde{\mathbf{R}}^{-1}\tilde{\mathbf{Q}}^H \mathbf{y}$. Second, the result of a matrix–vector multiplication needs to be rounded element-wise. The final step is to multiply the rounded results by the unimodular matrix $\mathbf{T}$ and constrain all results within the constellation boundary. If $\hat{\mathbf{x}}_q$ denotes the element-wise-rounded $\hat{\mathbf{x}}$, the final decision of the LRAD is $\hat{\mathbf{x}}_{LR} = \mathcal{Q}(\mathbf{T} \cdot \hat{\mathbf{x}}_q)$, as described in Section 2.3.

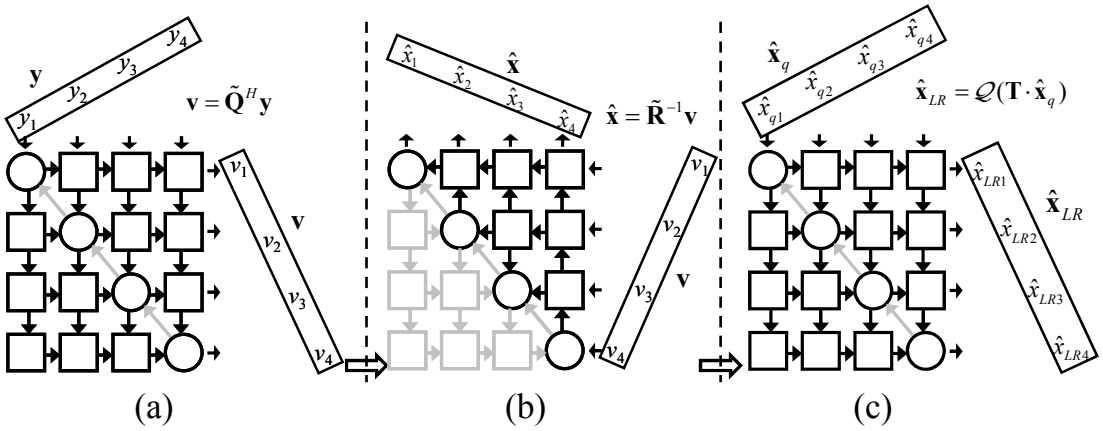
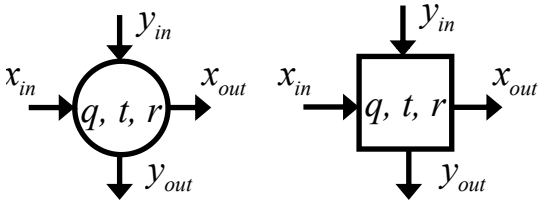
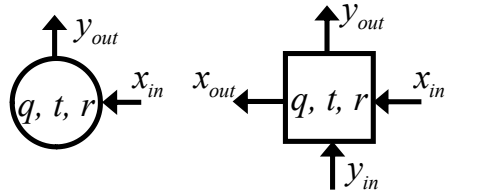


Figure 4.7: The linear detection operations in the systolic array. (a)$\mathbf{v} = \tilde{\mathbf{Q}}^H\mathbf{y}$ (b)$\hat{\mathbf{x}} = \tilde{\mathbf{R}}^{-1}\mathbf{v}$ (c)$\hat{\mathbf{x}}_{LR} = \mathcal{Q}(\mathbf{T}\cdot\hat{\mathbf{x}}_q)$.



operation	diagonal and off-diagonal cells
$\tilde{\mathbf{Q}}^H\mathbf{y}$	$x_{out} = x_{in} + q\cdot y_{in};\ y_{out} = y_{in}$
$\mathbf{T}\cdot\hat{\mathbf{x}}_q$	$x_{out} = x_{in} + t\cdot y_{in};\ y_{out} = y_{in}$

(a)



operation	diagonal cell	off-diagonal cell
$\hat{\mathbf{x}} = \tilde{\mathbf{R}}^{-1}\mathbf{v}$	$y_{out} = x_{in}/r$	$x_{out} = x_{in} - r\cdot y_{in}$ $y_{out} = y_{in}$

(b)

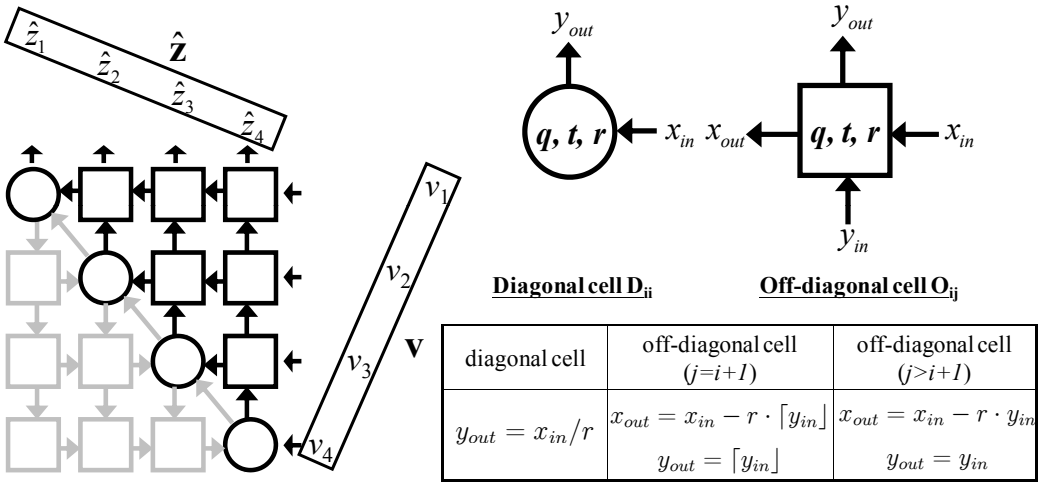
Figure 4.8: The detailed operations of the diagonal cell and the off-diagonal cell in the systolic array at different stage. (a)$\tilde{\mathbf{Q}}^H\mathbf{y}$ and $\mathbf{T}\cdot\hat{\mathbf{x}}_q$ (b)$\tilde{\mathbf{R}}^{-1}\mathbf{v}$.

In the following discussion, we again assume an 4×4 MIMO system, and consider the zero-forcing detection first. The first and last steps of a linear detection can be implemented by the same systolic array of Fig. 4.1 without using extra cells. As for the rounding and the final constellation boundary check, they should be done outside the systolic array (they are not shown in Fig. 4.7). To execute $\hat{\mathbf{x}} = \tilde{\mathbf{R}}^{-1}\tilde{\mathbf{Q}}^H\mathbf{y}$ in the systolic array, we separate it into two matrix–vector multiplications $\mathbf{v} = \tilde{\mathbf{Q}}^H\mathbf{y}$ and then $\hat{\mathbf{x}} = \tilde{\mathbf{R}}^{-1}\mathbf{v}$. Since $\tilde{\mathbf{Q}}^H$ stays in the systolic arrays after the lattice reduction ends, the received signal vector $\mathbf{y}$ can be fed to the systolic arrays from the top in a skewed manner as shown in Fig. 4.7(a). The vector $\tilde{\mathbf{Q}}^H\mathbf{y}$ is pumped out from the rightmost column of the array. Diagonal and off-diagonal cells are needed at this stage, and the operations of the cells are shown in Fig. 4.8(a). Every cell performs the multiply-and-add operation. If MMSE is chosen, the input vector should be changed to an $2n_t \times 1$ vector $\underline{\mathbf{y}}$ according to the extended model (2.5). Let $\underline{\mathbf{y}} = \left[\mathbf{y_1}^T\ \mathbf{y_2}^T\right]^T$ and $\tilde{\mathbf{Q}}^H = [\mathbf{Q_1}\ \mathbf{Q_2}]$, where $\mathbf{y}_1$, $\mathbf{y}_2$ are $n_t \times 1$ vectors and $\mathbf{Q}_1$, $\mathbf{Q}_2$ are $n_t \times n_t$ matrices. As mentioned in Section 4.2, the elements of $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are stored in the same PEs. To compute $\mathbf{v} = \tilde{\mathbf{Q}}^H\underline{\mathbf{y}}$ using the systolic array, first we let $\mathbf{y}_1$ enter the array from the top and multiply it by $\mathbf{Q}_1$, which is the same as shown in Fig. 4.7(a). Then $\mathbf{y}_2$ enters the array right after $\mathbf{y}_1$, also in a skewed manner, and is multiplied by $\mathbf{Q}_2$. Hence, for MMSE we need an extra operation at the output of the array, which is $\mathbf{v} = \mathbf{Q}_1\mathbf{y}_1 + \mathbf{Q}_2\mathbf{y}_2$. For the remaining operations in the systolic array, there is no difference between ZF and MMSE detections.

The second stage consists of computing $\hat{\mathbf{x}} = \tilde{\mathbf{R}}^{-1}\mathbf{v}$. Instead of computing $\tilde{\mathbf{R}}^{-1}$ directly, the following recursive equation [88] is considered for the systolic design

$$\hat{x}_j = \frac{1}{\tilde{r}_{j,j}}\left(v_j - \sum_{i=j+1}^{m} \tilde{r}_{j,i}\hat{x}_i\right),\ j \ \text{ starts from } \ n_t \text{ to } 1. \tag{4.1}$$

According to (4.1), it is clear that $\tilde{\mathbf{R}}^{-1}\mathbf{v}$ can be computed directly from the components of $\tilde{\mathbf{R}}$ without computing $\tilde{\mathbf{R}}^{-1}$. Additionally, it can be implemented



diagonal cell	off-diagonal cell ($j=i+1$)	off-diagonal cell ($j>i+1$)
$y_{out} = x_{in}/r$	$x_{out} = x_{in} - r \cdot \lceil y_{in} \rfloor$ $y_{out} = \lceil y_{in} \rfloor$	$x_{out} = x_{in} - r \cdot y_{in}$ $y_{out} = y_{in}$

Figure 4.9: The data flow and the detailed operations of the cells in the systolic array for the interference-cancellation step of LR-aided SIC.

by the upper triangle part of the systolic array, where matrix $\tilde{\mathbf{R}}$ has already been stored. As shown in Fig. 4.7(b), the vector $\mathbf{v} = \tilde{\mathbf{Q}}^H\mathbf{y}$ enters the array from the right, and $\hat{\mathbf{x}} = \tilde{\mathbf{R}}^{-1}\mathbf{v}$ is computed by the triangular array with cell operations shown in Fig. 4.8(b). The output vector $\hat{\mathbf{x}}$ is then rounded element-wise outside the systolic array. The final step consists of multiplying the quantized vector $\hat{\mathbf{x}}_q$ by the unimodular matrix $\mathbf{T}$, which is also stored in the array. Similar to the first step of a linear detection, it is a matrix–vector multiplication between $\mathbf{T}$ and $\hat{\mathbf{x}}_q$. Hence, the data flow in Fig. 4.7(c) is the same as Fig. 4.7(a). The cell operations for $\mathbf{T} \cdot \hat{\mathbf{x}}_q$ are shown in Fig. 4.8(a), and the array output being quantized to the closest constellation point is the final result $\hat{\mathbf{x}}_{LR}$ of the linear LRAD.

4.4.2 Spatial-Interference Cancellation in Systolic Array

The successive spatial-interference cancellation (SIC) can also be performed on this systolic array with some modifications to the PEs. Observing the first step of the LR-aided SIC showing in (2.10), it should be apparent that $\tilde{\mathbf{Q}}^H\mathbf{y}$ can be performed in the systolic array in the same fashion as in Fig. 4.7(a) and Fig. 4.8(a).

The second step (2.13) of LR-aided SIC can be done in the systolic array as shown in Fig 4.9. It is almost the same operations as the one Fig. 4.8(b), except that we have to do a rounding in the off-diagonal cells O_{ij} at the super-diagonal position ($j = i + 1$). The rounding operations are for the decision of each $\hat{z}_i$. Similar to the linear LRAD, the final step of the LR-aided SIC is to multiply $\mathbf{z}$ by the unimodular matrix $\mathbf{T}$ and bound all the output within the QAM constellation. It can be done in the same way as in Fig 4.7(c) and Fig. 4.8(a), with $\hat{\mathbf{x}}_q$ being replaced by $\hat{\mathbf{z}}$.

Notice that lattice reduction and linear detection (or SIC) are performed in the same systolic array, and it can be hardware-efficient to share the adder/multiplier/ divider designed for lattice reduction processing. For instance, there is one addition, one multiplication, and one division in each diagonal cell, and one addition and one multiplication in each off-diagonal cell for linear detection or SIC, be it ZF or MMSE. These operations are also contained in each cell at the LLL lattice reduction stage. For SIC, it seems that we need extra rounding operations in those off-diagonal cells at the superdiagonal position. Now, we need those rounding operations in the off-diagonal cells during the full size reduction processing as well. Hence, there need be no extra hardware cost (adders or multipliers) in each cell for linear detection. Only extra control logic to the array is needed in order to have each PE work correctly in different modes.

CHAPTER 5

FPGA Emulation for Systolic MIMO Lattice Reduction Aided Detection

In Chapter 4 we proposed a systolic design for two variants of the LLL algorithm, the FSR-LLL and the ASLR. The proposed systolic flow and the advantage of systolic arrays have been verified by Matlab simulations. In order to further explore the advantage of using systolic array and the trade-off between hardware resource and throughput, we implement our proposed systolic architecture onto a field-programmable gate array (FPGA). As shown in the Algorithm 3.2 and the Algorithm 3.3, the QR decomposition (QRD) should be pre-processed prior to the lattice reduction algorithms. In our emulation, we assume QRD or sorted QRD (SQRD) [87] has been processed, and we will focus on the implementation of systolic array for the ASLR and the FSR-LLL. The overall design flow and the software used are introduced in Section 5.1. The design of the diagonal cell, the off-diagonal cell and the rotation/vectoring cell are introduced in Sections 5.2, 5.3 and 5.4, respectively. The FPGA emulation results are shown in Section 5.5.

5.1 Design Flow

First, we build our design using Xilinx System Generator 11.5 (XSG) block-set in the Simulink design environment. Simulink provides a graphic interface for building up the system with a set of built-in models/blocks, which is convenient and intuitive for system designers. XSG provides a set of basic fixed-point arithmetic

models that is readily to use in Simulink for building larger systems. Also, XSG can automatically generates the VHSIC/Verilog Hardware Description Language (HDL) from the models created by the users in Simulink, which is good for system designers who are not familiar with HDL design, and for IC designers who want to have quick and preliminary results of their design. In our case, a Verilog program is generated by XSG and is then synthesized by Xilinx XST. The place and route are done by Xilinx ISE 11.5. From the emulation results given by Xilinx ISE 11.5, we will focus on the hardware working speed and the hardware resoure utilization.

5.2 Design of the Diagonal Cell

5.2.1 Siegel's Condition Check

The operations of a diagonal cell D_{ii} are shown in Fig. 4.1(b) again. To generate the "*swap*" signal, the diagonal cell has to do the Siegel's condition check on $|d_{in}|^2/|r|^2$, which requires two squares and a division. In hardware implementation, we certainly try to avoid using dividers since they require longer latency and higher hardware complexity. Since the systolic arrays require very accurate timing in the data communications between cells and the long latency of a divider could greatly delay the overall process or even ruin the systolic flow. Thanks to the relaxation by the Siegel's condition, we can simply compare the value of $|d_{in}|$ and $\sqrt{\delta - \frac{1}{2}}|r|$, where $\sqrt{\delta - \frac{1}{2}}$ is a pre-computed constant once δ is determined. In addition, there is no need to compute the square of $|r_{i,i}|$ and $|r_{i-1,i-1}|$, which saves even more bits in our design. The Simulink/XSG model for the Siegel's condition check is shown in Figure 5.1, where the constant δ is set to 0.99, and thus a constant multiplier "CMult($\times$0.7)" is in the middle path for $\sqrt{\delta - \frac{1}{2}}$.

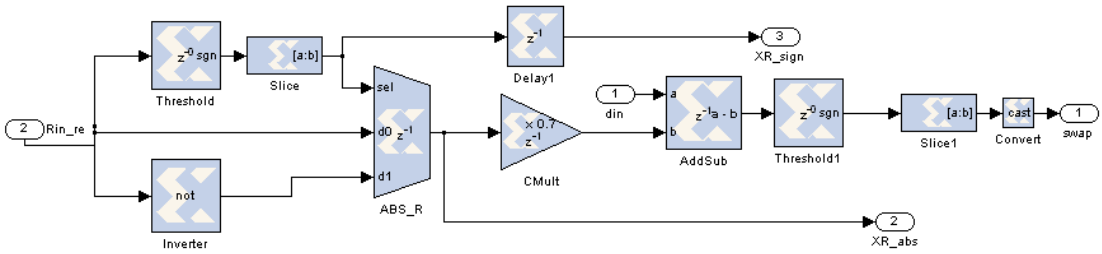


Figure 5.1: Simulink/XSG model for the Siegel's condition check

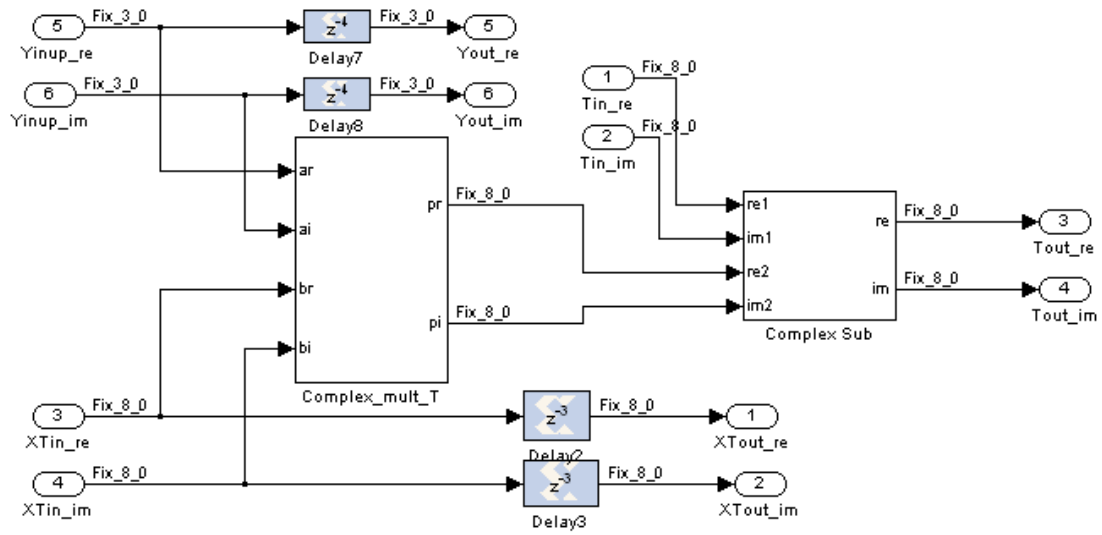


Figure 5.2: Simulink/XSG model for the Size Reduction mode in diagonal cells

5.2.2 Size Reduction with Fewer Multipliers

The size reduction mode contains one complex multiplications and one complex subtraction, which is shown in Fig. 5.2. Since every diagonal cell and off-diagonal cell contains size reduction operations, the total number of real multipliers could be quite high for large systems. To reduce the number of real multipliers, we use an efficient complex multiplication form [89]

$$(a+bj)(c+dj) = [c\cdot(a-b)+b\cdot(c-d)] + j\,[d\cdot(a+b)+b\cdot(c-d)]\,. \quad (5.1)$$

It requires three real multiplications and five real additions or subtractions. Namely, comparing to conventional complex multiplication we reduce one multiplier while adding three adders. Since adders are much cheaper than multipliers in terms of hardware complexity, and it is worthwhile to trade three adders for one multiplier especially when we have several complex multiplications in our systems. The Simulink/XSG model for the overall diagonal cell is shown in Fig. 5.3, where we use multiplexers and the m_{in} signal to control the cell to perform in the size reduction or in the data mode.

5.3 Design of the Off-Diagonal Cell

5.3.1 Computing μ without Division

The operations of an off-diagonal cell O_{ij} are shown in Fig. 4.1(b). The size reduction mode is performed by the module similar to the one in Fig. 5.2, except that μ has to be calculated when x_{in} is coming from a diagonal cell (carries a "*"). However, a division is needed to calculate μ , which is not desired in hardware implementations. If the value of μ (i.e. $\lceil r_{i,j}/r_{i,i} \rfloor$) always lies in a small range, we might have the chance to avoid performing division by using a set of comparators. For example, if $|\Re(r_{i,j})|$ is less than $|r_{i,i}|/2$, $\Re(\mu)$ should be 0[1]. On the other

[1]Note that $r_{i,j}$ is the diagonal element of $\mathbf{R}$ and is always a real number in this design.

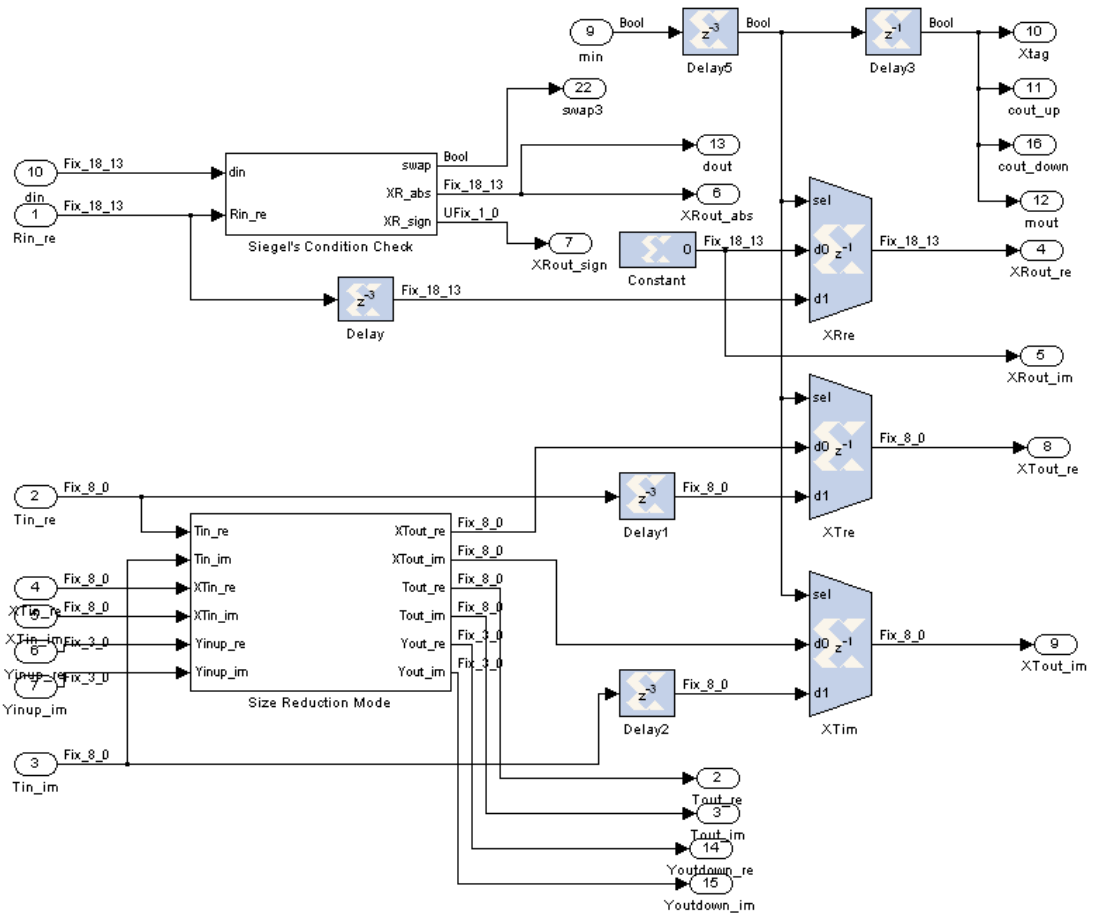
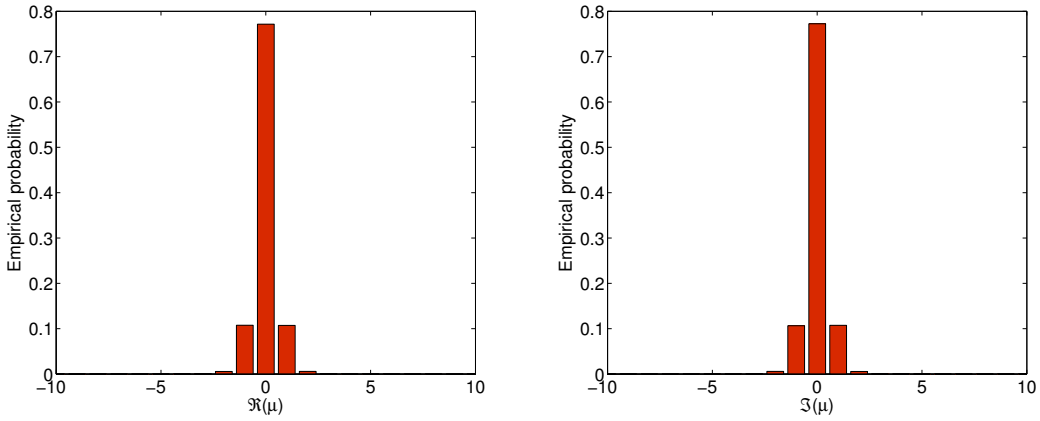
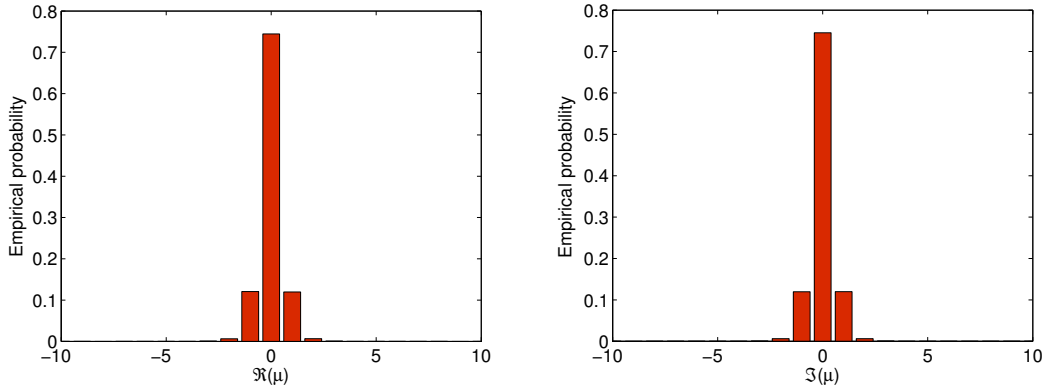


Figure 5.3: Simulink/XSG model for the diagonal cell



(a) (b)

Figure 5.4: Empirical probability mass function of (a) $\Re(\mu)$ and (b) $\Im(\mu)$ under the FSR-LLL algorithm



(a) (b)

Figure 5.5: Empirical probability mass function of (a) $\Re(\mu)$ and (b) $\Im(\mu)$ under the ASLR algorithm

X=	$P(\|X\| > 2)$	
	FSR-LLL	ASLR
$\Re(\mu)$	0.20%	0.22%
$\Im(\mu)$	0.20%	0.21%

Table 5.1: The probabilities of $\Re(\mu)$ and $\Im(\mu)$ locating outside $[-2, 2]$

hand, if $|\Re(r_{i,j})|$ is greater than $|r_{i,i}|/2$ but less than $3|r_{i,i}|/2$, $\Re(\mu)$ should be 1 or -1, depending on the sign of $\Re(r_{i,j})$. Therefore, we first check the distribution of μ . The empirical probability mass function (p.m.f.) for the 4×4 case is obtained through the Monte-Carlo simulation over 20,000 frequency-flat fading channel matrices. The results are shown in Fig. 5.4 and 5.5 for the FSR-LLL and the ASLR,respectively. It is clear that the real imaginary parts of μ are mostly concentrated on the interval $[-2, 2]$. Table 5.1 lists the probabilities that the real and imaginary parts of μ lie outside the range $[-2, 2]$. Almost 99.8% of μ resides in this interval. From the observations above, we propose an simple algorithm in Algorithm 5.1 to approximate μ without calculating $r_{i,j}/r_{i,i}$, and the accuracy of this approximation is about 99.8% empirically. In Algorithm 5.1, we only show the part for $\Re(\mu)$. Same algorithm can easily be applied to $\Im(\mu)$. Figure 5.6 shows the Simulink/XSG model for this algorithm. Only three adders, two threshold blocks, and several bit operators are needed in this model. Not only the cost of hardware is lower as compared to the one of a divider, the processing latency can be as low as one cycle. According to this algorithm, every $|\Re(\mu)|$ or $|\Im(\mu)|$ beyond 2 will be saturated to 2. However, this only happens about 0.2% of the time, which we assume that does not affect the BER performance significantly. We will verify this assumption by looking into the orthogonality defect of the channel matrix and also the BER when applying this approximation in the lattice reduction algorithms.

Algorithm 5.1 Algorithm to approach $\Re(\mu)$ without division

Input $X = \Re(r_{ij})$, $Y = |r_{i,i}|$

1: **if** $|X| < Y/2$ **then**

2: $\quad \Re(\mu) = 0$

3: **else if** $|X| \geq 3Y/2$ **then**

4: $\quad \Re(\mu) = \text{sgn}(X) \cdot 2$

5: **else**

6: $\quad \Re(\mu) = \text{sgn}(X)$

7: **end if**

8: **return** $\Re(\mu)$

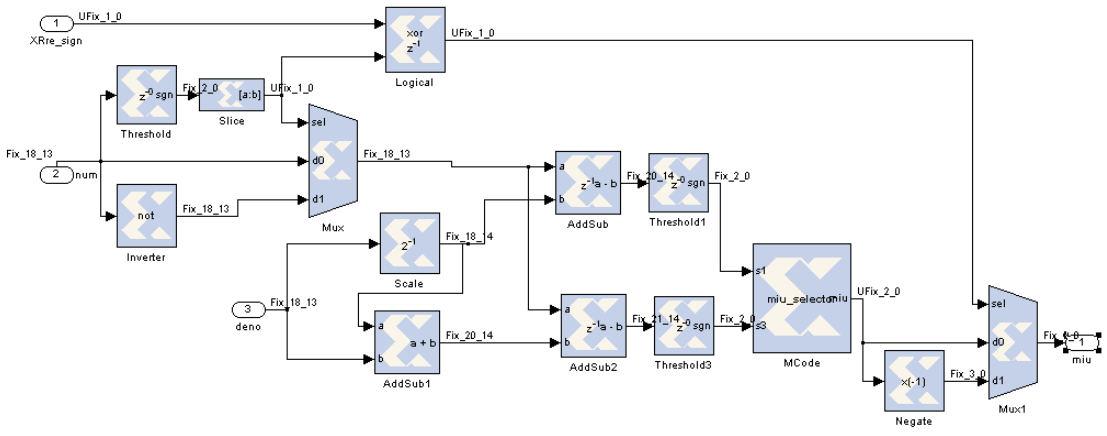


Figure 5.6: Simulink/XSG model for computing μ

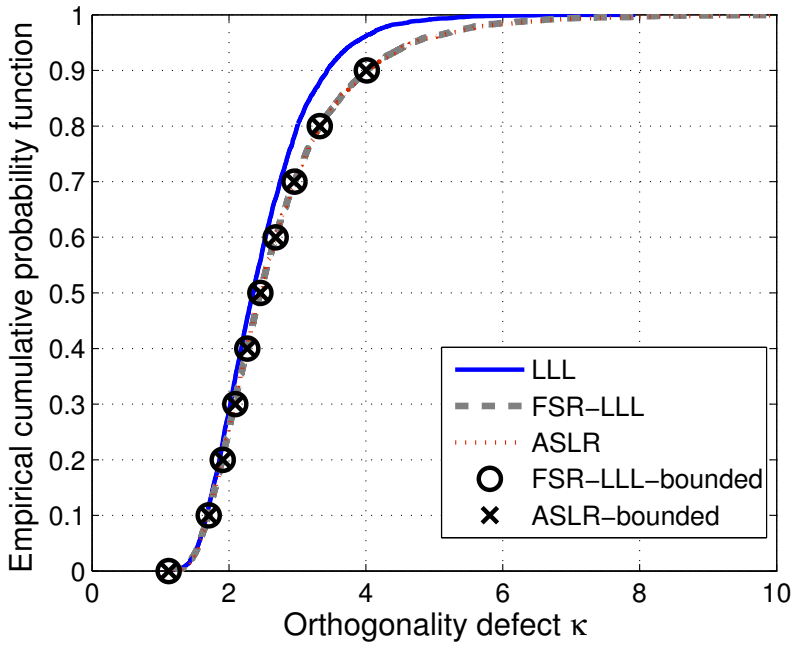


Figure 5.7: Orthogonality defect of the channel matrix using different lattice reduction algorithms

5.3.2 Orthogonality Defect with Bounded μ

The orthogonality defect κ of a matrix $\mathbf{H}$ ($\kappa \triangleq \frac{\prod_{i=1}^{m}\|\mathbf{h}_i\|^2}{\det(\mathbf{H}^H\mathbf{H})} \geq 1$) is a measure to see how close is a matrix $\mathbf{H}$ to an orthogonal matrix. If κ is 1, it means the matrix $\mathbf{H}$ is orthogonal. We compare the orthogonality defect of the channel matrices after applying the conventional LLL, the FSR-LLL, the FSR-LLL with bounded μ (denoted as FSR-LLL-bounded), the ASLR, and the ASLR with bounded μ (denoted as ASLR-bounded) when μ is set to 0.99. From Figure 5.7, there is almost no difference between the FSR-LLL, the FSR-LLL-bounded, the ASLR and the ASLR-bounded.

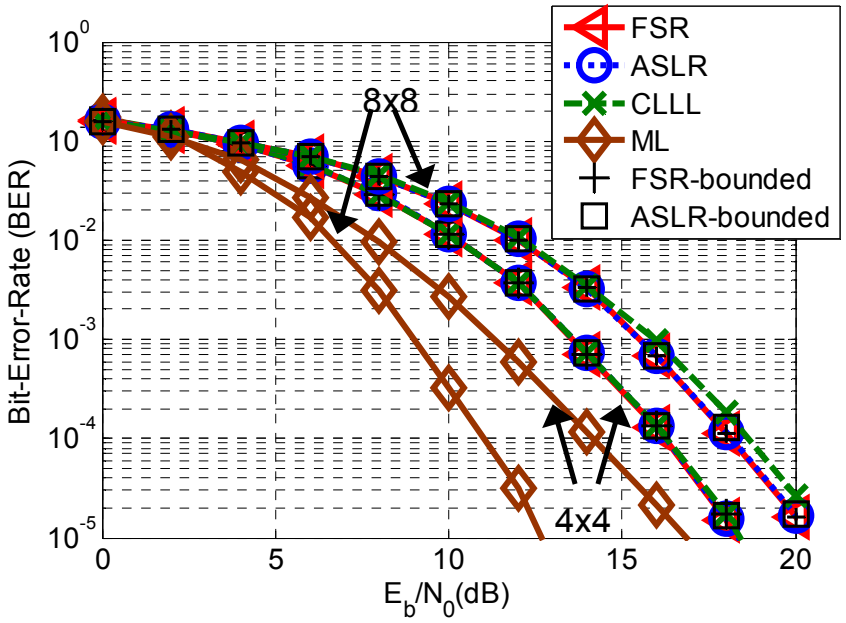


Figure 5.8: BER performance of the LR-aided MMSE detection in an 4×4 and an 8×8 systems using different lattice reduction algorithms

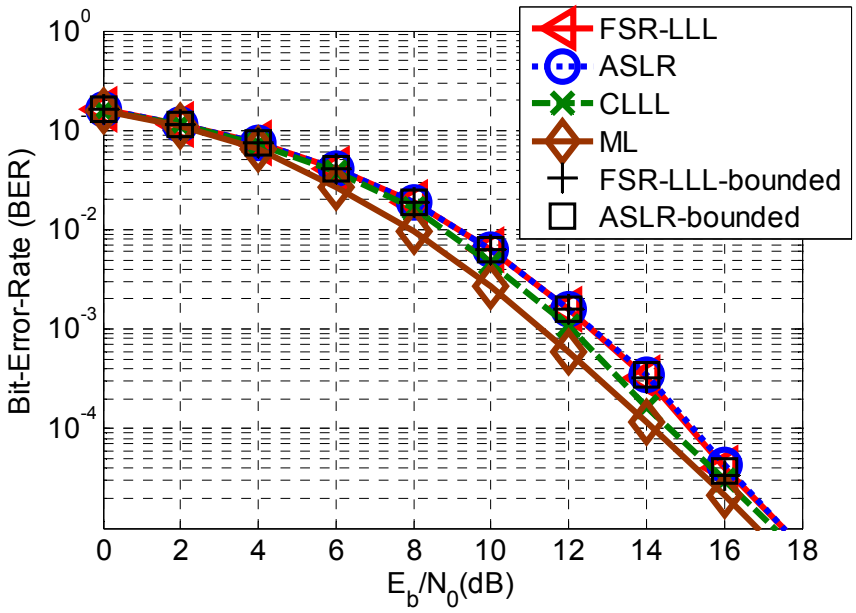


Figure 5.9: BER performance of the LR-aided MMSE-SIC detection in an 4×4 system using different lattice reduction algorithms

5.3.3 BER Performance with Bounded μ

We compare the BER performance of the MMSE and the MMSE-SIC using the five lattice reduction algorithms as in the previous section. As shown in Figures 5.8 and 5.9, the performance of the algorithms with bounded μ is almost the same as the one with true μ, which consequently verifies our assumption that the bounded μ does not affect the performance significantly.

5.3.4 Simulink/XSG Model

The overall structure for an off-diagonal cell is shown in Fig. 5.10. Two modules of computing "bounded μ" as shown in Fig. 5.6 are embedded in the size-reduction module, one for real part of μ and one for imaginary part of μ. For data-mode module, it simply sends out the r and t data stored in this cell. The multiplexers select the correct output signals from either size-reduction module or data-mode module according to the incoming signal c_{in} . Note that for off-diagonal cells in first column of the systolic array, only data-mode modules are needed since there is no size-reduction operation in the first column.

5.4 Design of the Rotation and Vectoring Cells

The givens rotation matrix $\mathbf{G}$ for rows $k-1$ and k in both the ASLR and the FSR-LLL algorithms can be written as

$$\eta_1 = \frac{r_{k-1,k}}{\|r_{k-1:k,k}\|},\ \eta_2 = \frac{r_{k,k}}{\|r_{k-1:k,k}\|}, \tag{5.2}$$

$$\mathbf{G} = \begin{bmatrix} \eta_1^+ & \eta_2 \\ -\eta_2 & \eta_1 \end{bmatrix}. \tag{5.3}$$

The vectoring cell computes the matrix $\mathbf{G}$ and applies it on $r_{k-1:k,k}$, and the rotation cell applies the same matrix $\mathbf{G}$ on the rest of the rows $k-1$ and k of $\mathbf{R}$

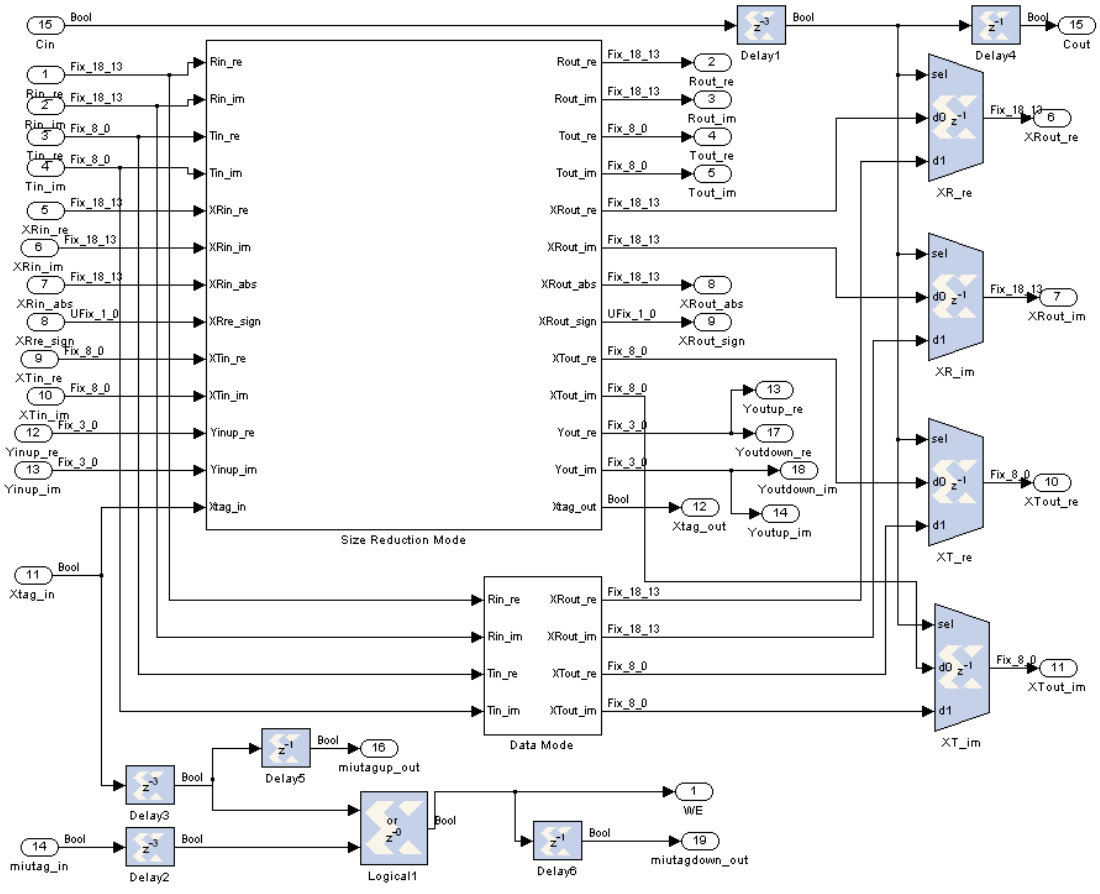


Figure 5.10: Simulink/XSG model for the off-diagonal cell

and $\mathbf{Q}^H$. As shown in (5.2), the vectoring cell needs to compute the inverse square roots, $\frac{1}{\|r_{k-1:k,k}\|}$. In our implementation, the Newton-Raphson algorithm [90] is chosen to compute the inverse square roots. The Newton-Raphson algorithm is an iterative inverse square roots algorithm as described in (5.4), which has a quadratic convergence rate when the initial value is properly selected.

$$p_{k+1} = \frac{p_k}{2} \cdot (3 - X \cdot p_k^2), \quad \lim_{k\to\infty} p_{k+1} = \frac{1}{\sqrt{X}}. \tag{5.4}$$

The Simulink/XSG model for the Newton-Raphson algorithm is shown in Fig. 5.11. It requires 10 hardware cycles for one iteration of processing. Even though the Givens rotation can be done in parallel with the size reduction and part of the latency could be hidden, we still want it to end in two iterations in order to speed up the overall process. With the initial value p_0 moderately close to the true value (25% relative error), the Newton-Raphson algorithm can converge to the result less than 5% relative error within two iterations [91]. Several look-up table based methods as in [92, 93] can be applied to generate fairly accurate initial values. However, the dynamic range of X in our case (namely $|r_{k-1,k}|^2 + |r_{k,k}|^2$) is not very large (mostly below 8) compared to general applications. We can simply set up a set of comparators and tree logic to determine the initial value.

Once the inverse square root is computed, the vectoring cell will send out this value along with $r_{k-1,k}$ and $r_{k,k}$ to the rotation cell, and the rotation cell can start the rotation between rows $k-1$ and k of $\mathbf{R}$ and $\mathbf{Q}^H$. Fig. 5.12 shows the Simulink/XSG model of the rotation cell. Note that this model is a straightforward implementation of the Givens rotation once the matrix $\mathbf{G}$ is given. It requires 16 multipliers which is not a heavy load for the modern FPGA board. However, if there is a limitation on the hardware resource, one may consider to apply the CORDIC algorithm [94, 95], which provides a hardware efficient way to perform Givens rotation. We will not consider CORDIC in our work since our implemented system only occupied about 10% of the total FPGA slices. More discussion on the

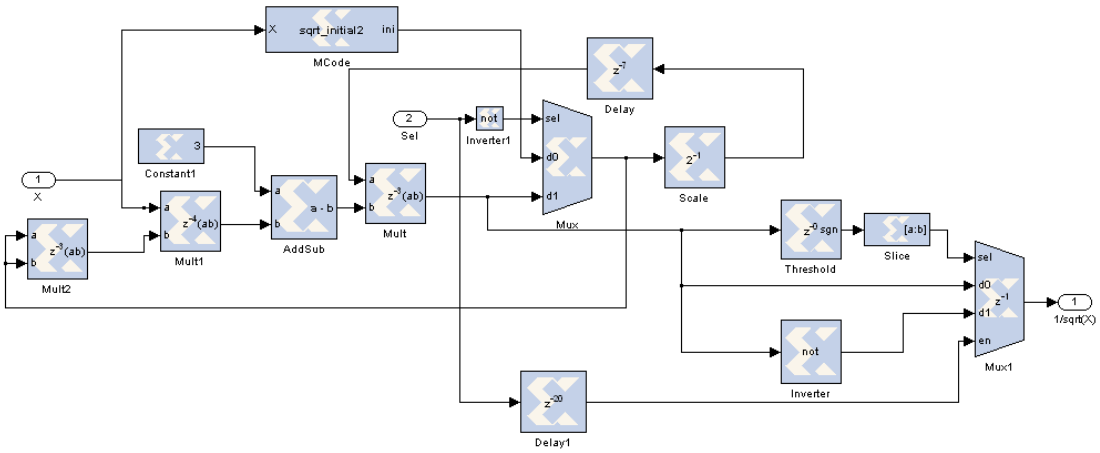


Figure 5.11: Simulink/XSG model for the Newton-Raphson algorithm

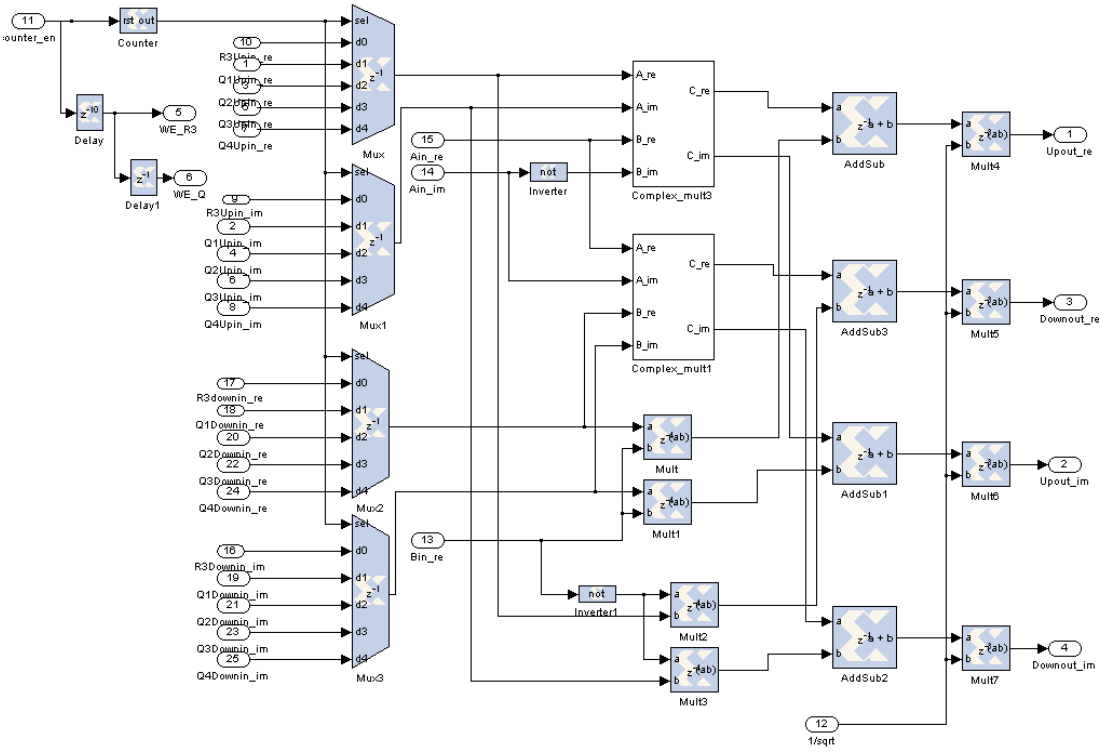


Figure 5.12: Simulink/XSG model for the rotation cell

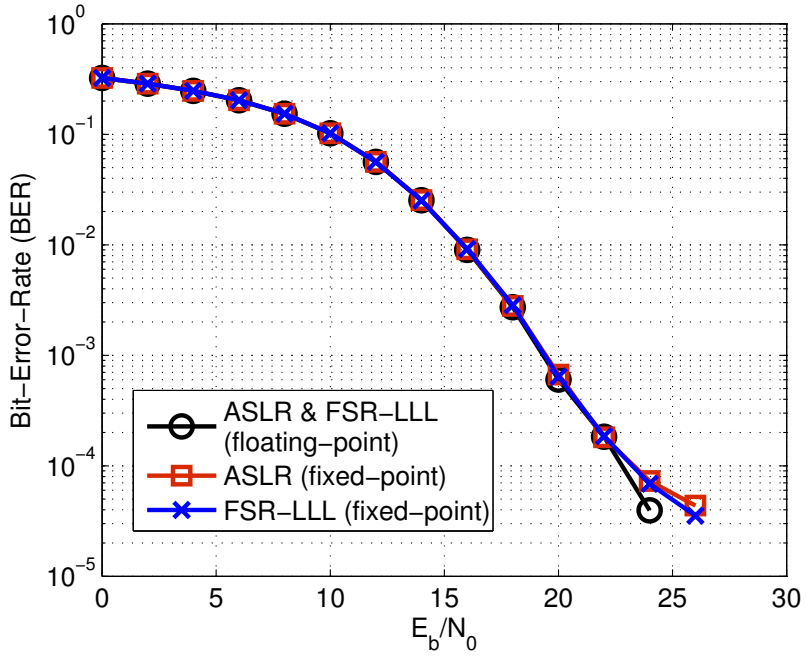


Figure 5.13: Comparison between the fixed-point and the floating-point lattice reduction algorithms using the ZF-SIC detection in an 4×4 MIMO system.

implementation results will be described in the next section.

5.5 Implementation Results

The Simulink/XSG model of the systolic array for an 4×4 MIMO system is shown in Fig. 5.14. The white rectangular boxes are the diagonal and off-diagonal cells, and the gray boxes next to them are the memories that stores the entries of $\mathbf{R}$ and $\mathbf{T}$. The light blue box next to each memory box is the memory address controller. The blue boxes between rows of the systolic array are the vectoring and rotation cells. There is no off-diagonal cell in the first column since there are only data-mode operations at those positions. The orange box is the external controller that controls the column-swap signals. In this implementation, the word-length of $\mathbf{R}$, $\mathbf{Q}^H$, $\mathbf{T}$ and μ are set to $(18, 13)$, $(14, 13)$, $(8, 0)$ and $(3, 0)$,

Target Algorithm	ASLR		FSR-LLL		Conventional LLL	
Device	Virtex 5 xc5vfx130t	Virtex 6 xc6vlx130t	Virtex 5 xc5vfx130t	Virtex 6 xc6vlx130t	Virtex 4	Virtex 5
Slices	2322/20480	1812/20000	2335/20480	1798/20000	3617/67584	1712/17280
Max. Clock Frequency	160MHz	249MHz	155MHz	247MHz	140 MHz	163 MHz
Avg. cycles/time per channel matrix	80 (SQRD)		84 (SQRD)		130 (SQRD)	
	500.0ns	321.3ns	541.9ns	340.1ns		
	146 (QRD)		164 (QRD)		928.6ns	797.5ns
	912.5ns	586.3ns	1058.1ns	664.0ns		

Table 5.2: FPGA emulation results

respectively. The BER performance of the fixed-point systolic implementation for an 4×4 MIMO system is shown in Fig. 5.13, where 16-QAM modulation and ZF-SIC detection are applied. It shows that even under fixed-point emulation, the BER performance is almost identical to the floating-point simulation down to 10^{-4}. The FPGA implementation results are shown in Table 5.2. We consider both the QRD and the SQRD as the pre-process parts of the lattice reduction algorithms. From the results, the ASLR is superior to the FSR-LLL in terms of the average processing time, and this advantage is significant when QRD is applied. The hardware complexity for the ASLR and the FSR-LLL are about the same, since they only differ from each other in the external controllers. It is also clear that the SQRD reduces the average processing time by over 40% compared to using the normal QRD, at the cost of higher computation efforts on the SQRD. In Table 5.2, a FPGA implementation result for the conventional complex LLL (CLLL) [16] is also listed for comparison. Under Virtex 5 and with the SQRD, the systolic ASLR operates at a slightly lower speed than the one of CLLL; however our designs require only 61.5% average clock cycles of theirs. As a result, the ASLR is on average faster than the CLLL by a factor of 1.6. This verifies the high-throughput advantage of the systolic arrays. On the other hand, systolic arrays implementation may have higher hardware complexity since it requires several processing elements to work in parallel. The results in Table 5.2 show that

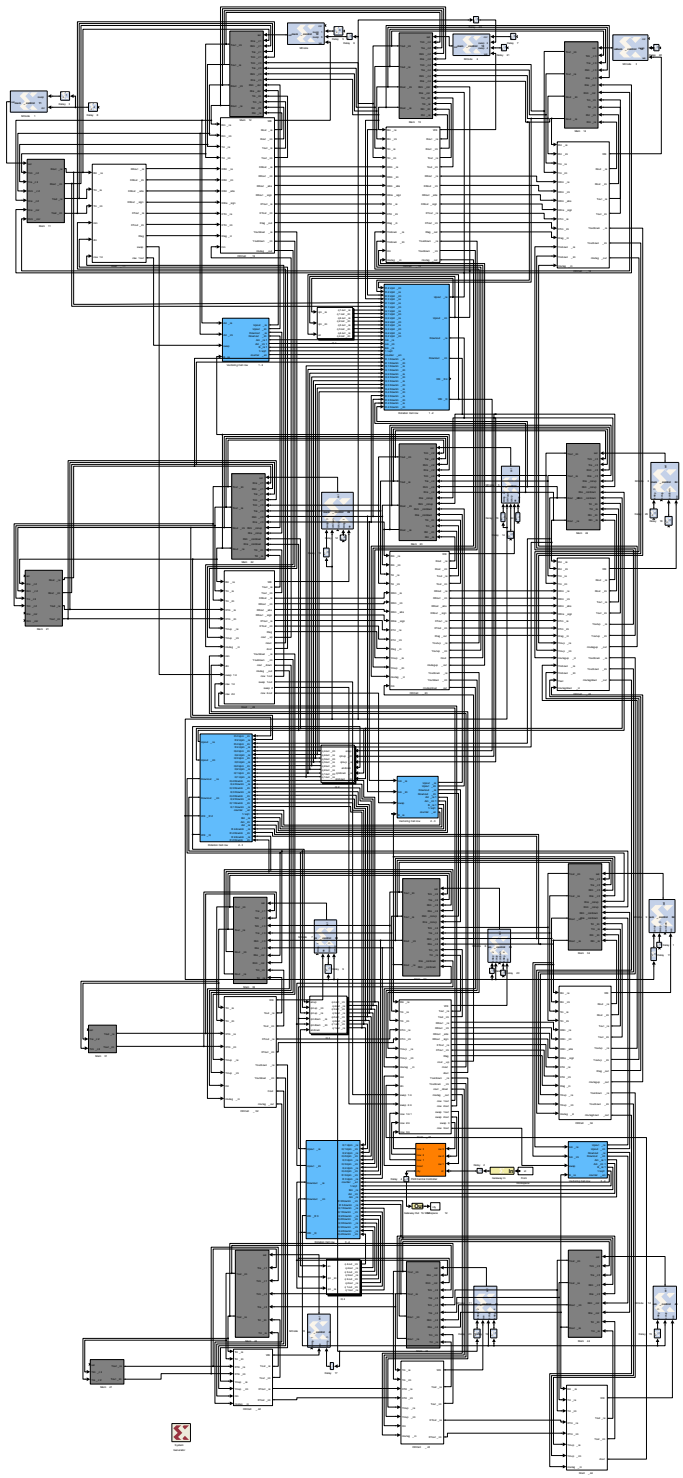


Figure 5.14: Simulink/XSG model for the systolic array for lattice reduction algorithms

our designs occupied $36 \sim 38\%$ more FPGA slices than the one in CLLL. However, as the advance of FPGA technology and the semiconductor processing, one may consider to trade some areas for a faster processing speed. As shown in Table 5.2, when using the latest Xilinx Virtex 6 FPGA device, our systolic designs could run up to 249MHz and it only requires less than 10% of the total FPGA slices.

CHAPTER 6

Fading Channel Modeling and Parameter Estimation Under Abrupt Temporal Variations

In the following two chapters, we consider the modeling and parameter estimation of the fading channel with sudden changes in the statistics. To fit the empirical data with common probability models of the fading channel envelope, such as Rayleigh, Nakagami-m, and Weibull distributions, maximum likelihood (ML) estimation or the moment estimation of the model parameters are usually applied [45, 52–54]. To use these estimators, the assumption are made that the data are independent and identically distributed. The independence can be approximated by longer sampling intervals so that the temporal correlation between samples are low. However, it is not trivial to see whether the data are identically distributed since the fading channel is nonstationary over time. In the literature, researchers did the parameter estimation using the measurement from a fixed and short period of time since it is assume that the channel is stationary or quasi-stationary within a short period. In this chapter, we advocate the usage of Bayesian change point detection on detecting abrupt changes in the fading channel statistics and perform the parameter estimation based on the change point detection results. Hence, it is a more precise way to see whether the measurement are identically distributed. If not, we are able to estimate the parameters for each segment and characterize the temporal variation more accurately.

6.1 Narrow-Band Fading Channel Formulation

We consider transmitting narrow-band signals over a wireless channel. The complex path gain g due to multipath effect can be written as

$$g = g^I + j \cdot g^Q = \sum_{i=1}^{N} a_i e^{j\theta_i}, \tag{6.1}$$

where g^I and g^Q are the real and imaginary part of g, respectively, N is the number of multipath components, a_i and θ_i are the random amplitude and phase of the i^{th} path, respectively. In this chapter, we assume the complex path gain is time varying due to the random time-varying behavior of the amplitude and phase of each path. Consequently, the subscript t is appended to each variable in (6.1) in the rest of the chapter to denote the time.

For fixed wireless link, where the positions of the transmitter and the receiver are fixed, the channel gain is static or time-invariant if there is no disturbance or moving objects in the field of view of antennas. However, in the environment such as office, shopping mall or the street with heavy traffic, the channel is usually observed as quasistatic for only a short period of time since there are usually people walking around or cars driving by that causes different signal levels [41, 56, 59, 61]. For mobile channel, where one end is usually fixed (such as the base station) and the other end is usually moving (such as the handheld device), the temporal variation in the channel gain is usually caused by the location variation. Transmission distance, shadowing effect, line-of-sight (LOS) or non-line-of-sight (NLOS) could be different at different location, which results in different channel gain. Hence, we assume the channel measurement could be divided into segments. Within each segment, the channel is stationary and all data are sampled from the same distribution with the same parameter set. The channel data among different segments are sampled from the same distribution but with different parameter sets. Since the channel is quasistatic only for a short period of time, we expect the time length of each segment to be short. The boundaries between segments are called the change

points. In this chapter, our goals are to detect those change points and at the meantime to estimate the parameter sets of each segment.

In the following discussions, we consider two different detection methods, coherent detection and non-coherent detection. With coherent detection, the instantaneous path gain g_t could be estimated, and the change point detection is performed directly on g_t. More commonly seen is the noncoherent detection, where the power level of the received signal is measured. Thus, we focus on the temporal variation of the amplitude level of the path gain, so the change point detection is applied on the envelope of the complex path gain $|g_t| = \sqrt{(g_t^I)^2 + \left(g_t^Q\right)^2}$. For simplicity, we will use h_t to denote $|g_t|$ in the rest of the chapter.

6.2 Bayesian Change Point Detection

The change point detection technique used here is based on the work in [62]. In this section, the methodology proposed in [62] will be briefly introduced. Denote $\mathbf{x}_{1:t}$ as the data sequence, $x_1, x_2, \cdots, x_t$, observed from time 1 to t. Assume all the data are independently sampled from the same type of probability distribution, but the parameter set of the distribution could be changing over time. Therefore, the sequence is divided into mutually exclusive segments, and the data within each segment are independently sampled from the distribution of the same parameter set. A change point occurs when there is a change in the parameter set, which is at the beginning of each segment. Define the run length r_t as the time since the last change point observed at time t, so $r_t = 0$ indicates a change point at time t. The principle of this method is to estimate r_t based on $P(r_t|\mathbf{x}_{1:t})$, the posterior distribution of r_t given the data observed so far $\mathbf{x}_{1:t}$. The run length posterior probability is proportional to the joint probability $P(r_t \cap \mathbf{x}_{1:t})$, which

can be recursively computed over the data and run length [62].

$$\begin{aligned} & P(r_t \cap \mathbf{x}_{1:t}) \\ & = \sum_{r_{t-1}} P(r_t|r_{t-1}) P(x_t|r_{t-1}, \mathbf{x}^{r_{t-1}}) P(r_{t-1} \cap \mathbf{x}_{1:t-1}), \end{aligned} \tag{6.2}$$

where $\mathbf{x}^{r_{t-1}}$ denotes the set of data associated with the run length r_{t-1}. The probability $P(r_t|r_{t-1})$ is the run length transition probability and $P(x_t|r_{t-1}, \mathbf{x}^{r_{t-1}})$ is the predictive probability of x_t given the past observations of the same segment. With suitable probability models for these two terms, we are able to calculated the joint probability $P(r_t \cap \mathbf{x}_{1:t})$ recursively. Furthermore, by normalizing this joint probability, we can obtain the run length posterior probability $P(r_t|\mathbf{x}_{1:t})$.

The run length transition probability $P(r_t|r_{t-1})$ could be defined by first selecting appropriate probability model for $P(r_t = 0|r_{t-1})$, which is the probability that a change point occurs at t after observing r_{t-1} consecutive data from the same segment. Once this probability model is determined, $P(r_t \neq 0|r_{t-1})$ is simply

$$P(r_t = r|r_{t-1}) = \begin{cases} 1 - P(r_t = 0|r_{t-1}), & r = r_{t-1} + 1. \\ 0, & r \neq r_{t-1} + 1 \text{ or } 0. \end{cases} \tag{6.3}$$

This is because there are only two possibilities at t, a change point ($r_t = 0$) or no change point ($r_t = r_{t-1} + 1$).

The predictive probability $P(x_t|r_{t-1}, \mathbf{x}^{r_{t-1}})$ in (6.2) is associated with the probability distribution where the data are sampled from through

$$P(x_t|r_{t-1}, \mathbf{x}^{r_{t-1}}) = \int P(x_t|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathbf{x}^{r_{t-1}}) d\boldsymbol{\theta}, \tag{6.4}$$

where $P(x_t|\boldsymbol{\theta})$ is the data distribution at t with parameter set $\boldsymbol{\theta}$. As previously mentioned, $P(x_t|\boldsymbol{\theta})$ is always the same type of distribution (Gaussian, for instance); however, the parameter set underlying changes once in a while. Even with the given data distribution, computing (6.4) is generally not trivial since evaluating the posterior probability $P(\boldsymbol{\theta}|\mathbf{x}^{r_{t-1}})$ may require a lot of computation. Under the

assumption that the data are i.i.d. within each segment, $P(\boldsymbol{\theta}|\mathbf{x}^{r_{t-1}})$ is proportional to

$$\propto \left[\prod_{i=t-r_{t-1}}^{t-1} P(x_i|\boldsymbol{\theta})\right] P(\boldsymbol{\theta}) \tag{6.5}$$

$$\propto P(x_{t-1}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\eta}^{r_{t-2}}), \tag{6.6}$$

$$\propto P(\boldsymbol{\theta}|\boldsymbol{\eta}^{r_{t-1}}), \tag{6.7}$$

where $\boldsymbol{\eta}^{r_i}$ is the hyperparameter set [96] of the prior distribution of $\boldsymbol{\theta}$ that is updated by all the data points in the current segment up to the newly observed x_i. The hyperparameter set $\boldsymbol{\eta}^{r_i}$ can also be viewed as the sufficient statistics of $\mathbf{x}^{r_i}$. From (6.6) and (6.7), it is clear that the integrand of (6.4) is proportional to $P(\boldsymbol{\theta}|\mathbf{x}^{r_t})$, which is the posterior probability of $\boldsymbol{\theta}$ to be used for calculating the predictive probability at time $t+1$. It is briefly mentioned in [62] and is thoroughly discussed in [96] that if there exists a conjugate prior $P(\boldsymbol{\theta}|\boldsymbol{\eta}^{r_{i-1}})$ of the sample distribution $P(x_i|\boldsymbol{\theta})$, the posterior probability $P(\boldsymbol{\theta}|\mathbf{x}^{r_i})$ will still be in the same class of distribution as the conjugate prior. Consequently, if the conjugate prior is applied, (6.5) to (6.7), the probability $P(\boldsymbol{\theta}|\mathbf{x}^{r_{t-1}})$, and also the integrand in (6.4) will all stay in the same class of distribution. Furthermore, if the integral of that class of distribution has an analytical form, then (6.4) is just a function of x_t and $\boldsymbol{\eta}^{r_{t-1}}$ with an analytic express and the predictive probability can be evaluated without numerical integration. The concept of conjugate prior will be made clearer in the next section.

The estimated run length at time t is determined by the maximum a posteriori probability (MAP) estimation

$$\hat{r}_t = \max_{r_t=0,1,\cdots,t} P(r_t|\mathbf{x}_{1:t}). \tag{6.8}$$

Ideally, we mark a change point at time t when $\hat{r}_t = 0$. However, there are cases when the estimated run length begins a new run with $\hat{r}_t = k \neq 0$. This usually occurs when the change between contiguous segments are not significant so that

the detector does not detect the change immediately. In these cases, the change point is marked at $t-k$ to compensate the detection delay.

6.3 Exponential Family and Conjugate Prior

As mentioned in Section 6.2, the existence of the conjugate prior of the data model is crucial to the change point detection algorithm, as it can greatly simplify the computation of the predictive probability and facilitate the online processing. Fortunately, the common models of the fading channel envelope, such as Rayleigh, Nakagami-m, and Weibull distribution, are exponential families and it is pointed out in [96] that the probability distributions that belong to exponential families have *natural* conjugate priors. The word "natural" means not only the posterior probability distributions are in the same class of distributions as the prior, but also they have the same functional form as the likelihood. In other words, the conjugate prior of the likelihood selecting from the exponential family is also in the exponential family.

The distribution of a random variable X with parameter set $\boldsymbol{\theta}$ belongs to an exponential family if the pdf can be written as

$$f_X(x|\boldsymbol{\theta}) = b(x)g(\boldsymbol{\theta})\exp\left(\boldsymbol{\phi}(\boldsymbol{\theta})^T\mathbf{u}(x)\right), \tag{6.9}$$

where $b(x)$ and $g(\boldsymbol{\theta})$ are known functions, and $\boldsymbol{\phi}(\boldsymbol{\theta})$ and $\mathbf{u}(x)$ are vectors of the same dimension as that of $\boldsymbol{\theta}$. The likelihood function of $\boldsymbol{\theta}$ when observing a sequence of i.i.d. observation $\mathbf{x}_{1:n}$ is

$$P(\mathbf{x}_{1:n}|\boldsymbol{\theta}) = \left[\prod_{i=1}^{n} b(x_i)\right] g(\boldsymbol{\theta})^n \exp\left(\boldsymbol{\phi}(\boldsymbol{\theta})^T \sum_{i=1}^{n} \mathbf{u}(x_i)\right). \tag{6.10}$$

If the prior of $\boldsymbol{\theta}$ is specified as

$$P(\boldsymbol{\theta}) \propto g(\boldsymbol{\theta})^{\gamma_0} \exp\left(\boldsymbol{\phi}(\boldsymbol{\theta})^T\boldsymbol{\nu}_0\right), \tag{6.11}$$

with hyperparameter set initiated as $\boldsymbol{\eta}_0 = \{\gamma_0, \boldsymbol{\nu}_0\}$, it is a conjugate prior of (6.10).

It is easy to see that the posterior probability associate with (6.10) and (6.11) is

$$P(\boldsymbol{\theta}|\mathbf{x}_{1:n}) \propto g(\boldsymbol{\theta})^{n+\gamma_0} \exp\left(\boldsymbol{\phi}(\boldsymbol{\theta})^T \left(\boldsymbol{\nu}_0 + \sum_{i=1}^{n} \mathbf{u}(x_i)\right)\right) \tag{6.12}$$

$$= g(\boldsymbol{\theta})^{\gamma_1} \exp\left(\boldsymbol{\phi}(\boldsymbol{\theta})^T \boldsymbol{\nu}_1\right), \tag{6.13}$$

where

$$\gamma_1 = \gamma_0 + n \quad \text{and} \quad \boldsymbol{\nu}_1 = \boldsymbol{\nu}_0 + \sum_{i=1}^{n} \mathbf{u}(x_i). \tag{6.14}$$

This shows that the posterior probability is in the same parametric form as the prior. Hence, in the following section we will use (6.11) to find the conjugate prior of the distributions of interest. Additionally, it can be seen from (6.12) and (6.13) that $\boldsymbol{\theta}$ depends on $\mathbf{x}$ only through the hyperparameter set $\boldsymbol{\eta}_1 = \{\gamma_1, \boldsymbol{\nu}_1\}$. Hence, $\boldsymbol{\eta}_1$ is the *sufficient statistic* for $\boldsymbol{\theta}$, and we can conclude that $P(\boldsymbol{\theta}|\mathbf{x}) = P(\boldsymbol{\theta}|\boldsymbol{\eta}_1)$. As observing more data, we should update the posterior probability through updating the sufficient statistic using the equations as shown in (6.14).

6.4 Change Point Detection and Parameter Estimation for Various Models of the Fading Channel Envelope

In this section, we will discuss how to apply the change point detection theory introduced in Section 6.2 on various models of fading channel envelope. As discussed in Section 6.2, the run length transition probability can be defined by selecting appropriate probability model for $P(r_t = 0|r_{t-1})$, which is generally independent to the data distribution. Since the change points of the fading channel occurs randomly, we simply take $P(r_t = 0|r_{t-1}) = 1/\lambda$ which indicates that a change point occurs at time t does not depend on the previous run length. The constant λ is expected length of each segment, which should be tuned for different transmission environments. This is also equivalent of assuming the length of each segment follows a geometric distribution with mean λ. On the other hand, the predictive probability is distribution-dependent. In the rest of this section, we will focus

on deriving the predictive probability for each distribution model. Additionally, we will derive the *maximum a posteriori* (MAP) and the *minimum mean square error* (MMSE) estimators of the model parameters for each segment based on the posterior probabilities $P(\boldsymbol{\theta}|\mathbf{x})$. The MAP estimator of the parameter $\boldsymbol{\theta}$ is selected by

$$\boldsymbol{\theta}^{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} P(\boldsymbol{\theta}|\mathbf{x}). \tag{6.15}$$

The MMSE estimator is the conditional expectation of $\boldsymbol{\theta}$ given $\mathbf{x}$ [97],

$$\boldsymbol{\theta}^{\text{MMSE}} = E[\boldsymbol{\theta}|\mathbf{x}] = \int_{\boldsymbol{\theta}} \boldsymbol{\theta} P(\boldsymbol{\theta}|\mathbf{x}) d\boldsymbol{\theta}. \tag{6.16}$$

The vector expression in (6.16) is equivalent to

$$\theta_i^{\text{MMSE}} = \int_{\boldsymbol{\theta}} \theta_i P(\boldsymbol{\theta}|\mathbf{x}) d\boldsymbol{\theta} \tag{6.17}$$

$$= \int_{\theta_i} \theta_i P(\theta_i|\mathbf{x}) d\theta_i, \ i = 1, 2, \cdots, \text{dimension}(\boldsymbol{\theta}). \tag{6.18}$$

6.4.1 One-Dimensional Gaussian Distribution

First, we consider the case of coherence detection, where the complex path gain g_t is detected. By the expression in (6.1) and the central limit theorem, as N is large enough, g_t^I and g_t^Q can be approximated as Gaussian random processes. In this section, we first discuss the case of using one-dimensional Gaussian distribution. In the next section, we will generalize the results to multivariate Gaussian, which is a better fit for our application. Hence, we denote the observed data x_t in the following discussion and x_t can be viewed as either the I-channel realization g_t^I or the Q-channel realization g_t^Q.

Gaussian pdf has a natural conjugate prior since it belongs to the exponential family. The conjugate prior of a Gaussian likelihood with mean μ and variance σ^2

is known to be a Gaussian-inverse-gamma distribution [96],

$$\begin{aligned}&\mathcal{NIG}(\mu,\sigma^2|m,\tau,\alpha,\beta)\\&=\frac{(\sigma^2)^{-\alpha-\frac{3}{2}}}{\Gamma(\alpha)}\frac{\beta^\alpha}{\sqrt{2\pi\tau}}\exp\left\{-\frac{(\mu-m)^2+2\tau\beta}{2\tau\sigma^2}\right\}.\end{aligned}\tag{6.19}$$

Or equivalently,

$$P(\mu|\sigma^2)\sim\mathcal{N}(m,\tau\sigma^2)\text{ and }P(\sigma^2)\sim\mathcal{IG}(\alpha,\beta).\tag{6.20}$$

As a result, using this conjugate prior we can obtain the posterior probability $P(\mu,\sigma^2|\mathbf{x}^{r_{t-1}})$ also in a Gaussian-inverse-gamma distribution, $\mathcal{NIG}(\mu,\sigma^2|\boldsymbol{\eta}^{r_{t-1}})$, where the parameter set $\boldsymbol{\eta}^{r_{t-1}}$ is $\{m^{r_{t-1}},\tau^{r_{t-1}},\alpha^{r_{t-1}},\beta^{r_{t-1}}\}$. Therefore, the predictive probability in (6.4) is the integral of the product of a Gaussian pdf and a Gaussian-inverse-gamma pdf, which by the derivation in Appendix A.1 can be shown as a Student's t-pdf,

$$P(x_t|r_{t-1},\mathbf{x}^{r_{t-1}})=\mathcal{T}\left(x_t\left|2\alpha^{r_{t-1}},\frac{\alpha^{r_{t-1}}}{\beta^{r_{t-1}}(\tau^{r_{t-1}}+1)},m^{r_{t-1}}\right.\right).\tag{6.21}$$

The Student's t-pdf is a function of three parameters,

$$\mathcal{T}(x|\nu,\lambda,m)=\sqrt{\frac{\lambda}{\pi\nu}}\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{\lambda(x-m)^2}{\nu}\right)^{-\frac{\nu+1}{2}}.\tag{6.22}$$

Similar to the case in (6.14), the hyperparameter set $\boldsymbol{\eta}^{r_{t-1}}$ that defines the predictive probability (6.21) needs to be updated when observing the new data x_t,

$$m^{r_t}=\frac{\tau^{r_{t-1}}x_t+m^{r_{t-1}}}{\tau^{r_{t-1}}+1},\tag{6.23}$$

$$\tau^{r_t}=\frac{\tau^{r_{t-1}}}{\tau^{r_{t-1}}+1},\tag{6.24}$$

$$\alpha^{r_t}=\alpha^{r_{t-1}}+\frac{1}{2},\tag{6.25}$$

$$\beta^{r_t}=\beta^{r_{t-1}}+\frac{(x_t-m^{r_{t-1}})^2}{2(\tau^{r_{t-1}}+1)}.\tag{6.26}$$

The derivation of (6.23) to (6.26) are shown in the Appendix A.1. Using the updated hyperparameter set $\boldsymbol{\eta}_t$, we are able to obtain $P(\mu,\sigma^2|\boldsymbol{\eta}^{r_t})$ for the next iteration.

To estimate the parameters (μ, σ^2) of the channel model at time t based on the observations $\mathbf{x}^{r_t}$ associated with the current segment, we derive the MAP and the MMSE estimators based on the posterior probability $P(\mu, \sigma^2|\mathbf{x}^{r_t}) = P(\mu, \sigma^2|\boldsymbol{\eta}^{r_t})$, which is still a Gaussian-inverse-gamma pdf.

MAP estimators: Let L_t be the logarithm of the posterior probability at time t,

$$L_t = \left(-\alpha^{r_t} - \frac{3}{2}\right) \log \sigma^2 - \frac{(\mu - m^{r_t})^2 + 2\tau^{r_t}\beta^{r_t}}{2\tau^{r_t}\sigma^2} + C, \tag{6.27}$$

where C is a constant not related to μ and σ^2. Using the MAP criterion in (6.15), the MAP estimation of μ and σ^2 at time t can be obtained by solving two equations $\partial L_t/\partial\mu = \partial L_t/\partial\sigma^2 = 0$, which yields

$$\mu_t^{\text{MAP}} = m^{r_t}, \tag{6.28}$$

$$\left(\sigma^2\right)_t^{\text{MAP}} = \frac{\beta^{r_t}}{\alpha^{r_t} + \frac{3}{2}}. \tag{6.29}$$

Hence, after the estimated run length $\hat{r}_t$ is determined by (6.8), the MAP estimation of the parameter set can be immediately calculated by (6.28) and (6.29) using the hyperparameters associated with $\hat{r}_t$.

MMSE estimators: The marginal probability $P(\mu|\mathbf{x}^{r_t})$ can be calculated by

$$\int_{\sigma^2} \mathcal{NIG}(\mu, \sigma^2|m^{r_t}, \tau^{r_t}, \alpha^{r_t}, \beta^{r_t}) d\sigma^2 \tag{6.30}$$

$$= \frac{\Gamma\left(\alpha^{r_t} + \frac{1}{2}\right)}{\Gamma\left(\alpha^{r_t}\right)} \frac{(\beta^{r_t})^{\alpha^{r_t}}}{\sqrt{2\pi\tau^{r_t}}} \left(\beta^{r_t} + \frac{(\mu - m^{r_t})^2}{2\tau^{r_t}}\right)^{-\left(\alpha^{r_t} + \frac{1}{2}\right)} \times \int_{\sigma^2} \mathcal{IG}\left(\sigma^2 \left| \alpha^{r_t} + \frac{1}{2}, \beta^{r_t} + \frac{(\mu - m^{r_t})^2}{2\tau^{r_t}} \right.\right) \tag{6.31}$$

$$= \frac{\Gamma\left(\alpha^{r_t} + \frac{1}{2}\right)}{\Gamma\left(\alpha^{r_t}\right)} \frac{1}{\sqrt{2\pi\beta^{r_t}\tau^{r_t}}} \left(1 + \frac{(\mu - m^{r_t})^2}{2\beta^{r_t}\tau^{r_t}}\right)^{-\left(\alpha^{r_t} + \frac{1}{2}\right)} \tag{6.32}$$

$$= \mathcal{T}\left(\mu \left| 2\alpha^{r_t}, \frac{\alpha^{r_t}}{\beta^{r_t}\tau^{r_t}}, m^{r_t} \right.\right). \tag{6.33}$$

Using (6.18), the MMSE estimator for μ associated with $\hat{r}_t$ is the mean of the Student-t distribution in (6.33), which is

$$\mu_t^{\text{MMSE}} = m^{\hat{r}_t}. \tag{6.34}$$

For the variance σ^2, the marginal pdf $P(\sigma^2|\mathbf{x}^{r_t})$ can be easily found to be an inverse-gamma distribution with parameters α^{r_t} and β^{r_t} using (6.20). Therefore, the MMSE estimator of σ^2 is the mean of the inverse-gamma distribution, which is

$$(\sigma^2)_t^{\text{MMSE}} = \frac{\beta^{\hat{r}_t}}{\alpha^{\hat{r}_t} - 1}. \tag{6.35}$$

It is observed that, for μ, the MMSE estimator is the same as the MAP estimator, while this is not true for σ^2.

6.4.2 Multivariate Gaussian Distribution

In this section, we generalize the results in 6.4.1 to multivariate Gaussian distribution, so we are able to jointly detect the changes from both g_t^I and g_t^Q, and also jointly estimate the parameters of bivariate Gaussian model. Assume the observation $\mathbf{x}_t$ is a N-dimensional Gaussian sequence. Hence, $P(\mathbf{x}_t|\boldsymbol{\theta})$ is now a multivariate Gaussian pdf and $\boldsymbol{\theta}$ consists of the $N \times 1$ mean vector $\boldsymbol{\mu}$ and the $N \times N$ covariance matrix $\boldsymbol{\Sigma}$ of $\mathbf{x}_t$. The conjugate prior of a multivariate Gaussian likelihood is known to be a Gaussian-inverse-Wishart distribution [98],

$$\begin{aligned} &\mathcal{NIW}(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{m}, \tau, \nu, \mathbf{W}) \\ &= \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}, \tau\boldsymbol{\Sigma})\mathcal{IW}(\boldsymbol{\Sigma}|\nu, \mathbf{W}), \end{aligned} \tag{6.36}$$

where $\mathcal{IW}(\mathbf{X}|\nu, \mathbf{W})$ is the inverse-Wishart distribution with parameters ν and $N \times N$ matrix $\mathbf{W}$,

$$\mathcal{IW}(\mathbf{X}|\nu, \mathbf{W}) = \frac{|\mathbf{W}|^{\frac{\nu}{2}} |\mathbf{X}|^{-\frac{\nu+N+1}{2}}}{2^{\frac{\nu N}{2}} \Gamma_N\left(\frac{\nu}{2}\right)} e^{-\frac{1}{2}\text{Tr}(\mathbf{W}\mathbf{X}^{-1})}, \tag{6.37}$$

and $\Gamma_N(x) = \pi^{N(N-1)/4} \prod_{j=1}^{N} \Gamma(x + \frac{1-j}{2})$ is the N-varitate gamma function. With the conjugate prior, the posterior probability $P(\boldsymbol{\mu}, \boldsymbol{\Sigma}|\mathbf{x}^{r_{t-1}})$ is also a Gaussian-inverse-Wishart distribution with hyperparameter set $\boldsymbol{\eta}^{r_{t-1}}$ consisting of $\mathbf{m}^{r_{t-1}}$, $\tau^{r_{t-1}}$, $\nu^{r_{t-1}}$, and $\boldsymbol{\Sigma}^{r_{t-1}}$. Therefore, the predictive probability can be calculated from

the integral of the product of a multivariate Gaussian pdf and a Gaussian-inverse-Wishart pdf. Through the similar derivation as in Appendix A.1, the predictive probability is

$$P(\mathbf{x}_t|r_{t-1},\mathbf{x}^{r_{t-1}}) = \frac{\Gamma\left(\frac{\nu^{r_{t-1}}+1}{2}\right)}{\pi^{\frac{N}{2}}\Gamma\left(\frac{\nu^{r_{t-1}}+1-N}{2}\right)}\left|(\tau^{r_{t-1}}+1)\mathbf{W}^{r_{t-1}}\right|^{-\frac{1}{2}} \times \left[1+\frac{1}{\tau^{r_{t-1}}+1}(\mathbf{x}_t-\mathbf{m}^{r_{t-1}})^T\left(\mathbf{W}^{r_{t-1}}\right)^{-1}(\mathbf{x}_t-\mathbf{m}^{r_{t-1}})\right]^{-\frac{\nu_{r_{t-1}}+1}{2}}, \tag{6.38}$$

$$= \mathcal{T}\left(\mathbf{x}_t\middle|\nu^{r_{t-1}}+1-N, \frac{(\tau^{r_{t-1}}+1)\mathbf{W}^{r_{t-1}}}{\nu^{r_{t-1}}+1-N}, \mathbf{m}^{r_{t-1}}\right), \tag{6.39}$$

where the $\mathcal{T}(\cdot)$ here is the N-variate Student-t pdf,

$$\mathcal{T}(\mathbf{x}|\nu,\mathbf{\Lambda},\mathbf{m}) = \frac{\Gamma\left(\frac{\nu+N}{2}\right)}{(\nu\pi)^{\frac{N}{2}}\Gamma\left(\frac{\nu}{2}\right)}|\mathbf{\Lambda}|^{-\frac{1}{2}}\left[1+\frac{1}{\nu}(\mathbf{x}-\mathbf{m})^T\mathbf{\Lambda}^{-1}(\mathbf{x}-\mathbf{m})\right]^{-\frac{\nu+N}{2}}. \tag{6.40}$$

The updating equtions for the hyperparameter set $\boldsymbol{\eta}^{r_{t-1}}$ when observing the new data vector $\mathbf{x}_t$ can also be derived in the similar way as in Appendix A.1, which yields

$$\mathbf{m}^{r_t} = \frac{\tau^{r_{t-1}}\mathbf{x}_t + \mathbf{m}^{r_{t-1}}}{\tau^{r_{t-1}}+1}, \tag{6.41}$$

$$\tau^{r_t} = \frac{\tau^{r_{t-1}}}{\tau^{r_{t-1}}+1}, \tag{6.42}$$

$$\nu^{r_t} = \nu^{r_{t-1}}+1, \tag{6.43}$$

$$\mathbf{W}^{r_t} = \mathbf{W}^{r_{t-1}} + \frac{(\mathbf{x}_t-\mathbf{m}^{r_{t-1}})(\mathbf{x}_t-\mathbf{m}^{r_{t-1}})^T}{\tau^{r_{t-1}}+1}. \tag{6.44}$$

MAP estimators: The posterior probability $P(\boldsymbol{\mu},\mathbf{\Sigma}|\mathbf{x}^{r_t})$ is in a Gaussian-inverse-Wishart form, so the logarithm of the posterior probability is

$$\begin{aligned} L_t &= \log P(\boldsymbol{\mu},\mathbf{\Sigma}|\mathbf{x}^{r_t}) \\ &= -\frac{\nu^{r_t}+N+2}{2}\log|\mathbf{\Sigma}| \\ &\quad -\frac{1}{2}(\boldsymbol{\mu}-\mathbf{m}^{r_t})^T(\tau^{r_t}\mathbf{\Sigma})^{-1}(\boldsymbol{\mu}-\mathbf{m}^{r_t}) - \frac{1}{2}\mathrm{Tr}(\mathbf{W}^{r_t}\mathbf{\Sigma}^{-1}) + C \end{aligned} \tag{6.45}$$

where C is a constant term not related to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. By setting $\partial L_t/\partial \boldsymbol{\mu} = 0$, it is easily shown that

$$-(\tau^{r_t}\boldsymbol{\Sigma})^{-1}\boldsymbol{\mu} + (\tau^{r_t}\boldsymbol{\Sigma})^{-1}\mathbf{m}^{r_t} = 0 \tag{6.46}$$

Consequently, the MAP estimation of $\boldsymbol{\mu}$ at time t that associated with the estimated run length $\hat{r}_t$ is

$$\boldsymbol{\mu}_t^{\text{MAP}} = \mathbf{m}^{\hat{r}_t}. \tag{6.47}$$

To find the MAP estimation of $\boldsymbol{\Sigma}$, we need the following identities for the symmetric nonsingular matrix $\mathbf{X}$ from the matrix calculus [99],

$$\frac{\partial |\mathbf{X}|}{\partial \mathbf{X}} = |\mathbf{X}|\,\mathbf{X}^{-1}, \tag{6.48}$$

$$\frac{\partial \log \mathbf{X}}{\partial \mathbf{X}} = \mathbf{X}^{-1}, \tag{6.49}$$

$$\frac{\partial \mathbf{a}^T\mathbf{X}^{-1}\mathbf{a}}{\partial \mathbf{X}} = -\mathbf{X}^{-1}\mathbf{a}\mathbf{a}^T\mathbf{X}^{-1}, \tag{6.50}$$

$$\frac{\partial \text{Tr}(\mathbf{A}\mathbf{X}^{-1})}{\partial \mathbf{X}} = -\mathbf{X}^{-1}\mathbf{A}\mathbf{X}^{-1}, \tag{6.51}$$

where $\mathbf{A}$ is a symmetric matrix of the same size as $\mathbf{X}$ and $\mathbf{a}$ is a column vector. Taking the derivative of L_t with respect to $\boldsymbol{\Sigma}$ yields

$$\begin{aligned}\frac{\partial L_t}{\partial \boldsymbol{\Sigma}} =& -\frac{\nu^{r_t} + N + 2}{2}\boldsymbol{\Sigma}^{-1} \\ &+ \frac{1}{2}(\tau^{r_t})^{-1}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{m}^{r_t})(\boldsymbol{\mu} - \mathbf{m}^{r_t})^T\boldsymbol{\Sigma}^{-1} \\ &+ \frac{1}{2}\boldsymbol{\Sigma}^{-1}\mathbf{W}^{r_t}\boldsymbol{\Sigma}^{-1}.\end{aligned} \tag{6.52}$$

By setting $\partial L_t/\partial \boldsymbol{\Sigma} = 0$ and plugging in the MAP estimator of $\boldsymbol{\mu}$ in (6.47), we can solve the MAP estimator of $\boldsymbol{\Sigma}$ associated with $\hat{r}_t$ as

$$\boldsymbol{\Sigma}_t^{\text{MAP}} = \frac{\mathbf{W}^{\hat{r}_t}}{\nu^{\hat{r}_t} + N + 2}. \tag{6.53}$$

The two MAP estimators derived here are indeed the generalization of those of one-dimensional case. To see this, by setting $N = 1$ the two estimators (6.47) and (6.53) are reduced to (6.28) and (6.29), respectively.

MMSE estimators: Using the similar derivations for the MMSE estimators of one-dimensional Gaussian, we can derive the marginal pdf of $\boldsymbol{\mu}$ as

$$P(\boldsymbol{\mu}|\mathbf{x}^{r_t}) = \mathcal{T}\left(\boldsymbol{\mu}\left|\nu^{r_t}+1-N, \frac{\tau^{r_t}\mathbf{W}^{r_t}}{\nu^{r_t}+1-N}, \mathbf{m}^{r_t}\right.\right). \tag{6.54}$$

And the marginal pdf of $\boldsymbol{\Sigma}$ is

$$P(\boldsymbol{\Sigma}|\mathbf{x}^{r_t}) = \mathcal{IW}\left(\boldsymbol{\Sigma}\,|\nu^{r_t}, \mathbf{W}^{r_t}\right). \tag{6.55}$$

Hence the MMSE estimators are the conditional mean of (6.54) and (6.55), respectively,

$$\boldsymbol{\mu}_t^{\text{MMSE}} = \mathbf{m}^{\hat{r}_t}, \tag{6.56}$$

$$\boldsymbol{\Sigma}_t^{\text{MMSE}} = \frac{\mathbf{W}^{\hat{r}_t}}{\nu^{\hat{r}_t}-N-1}. \tag{6.57}$$

6.4.3 Rayleigh Distribution

For noncoherent detection, we measure the amplitude or envelope of the complex path gain, $h_t = \left|g_t^I + jg_t^Q\right|$. Rayleigh distribution is widely used to model the channel envelop for narrow-band signals under non line-of-sight (NLOS) transmission. The physical interpretation of the Rayleigh fading envelope is that the received signal is the superposition of many independent reflections of the original radio wave from the random objects as expressed in (6.1). Assuming the amplitude of each reflection path is about the same level, by central limit theorem the channel path gain can be viewed as a circularly symmetric complex Gaussian random variable with zero mean [100]. The envelope of this complex Gaussian is Rayleigh distributed. On the other hand, if there is line-of-sight (LOS) between the transmitter and the receiver, the complex Gaussian is circularly symmetric around a nonzero mean. In that case, the envelop is Rice distributed.

For Rayleigh fading envelope h_t, the pdf is

$$f(h_t) = \frac{h_t}{\sigma^2} e^{-\frac{h_t^2}{2\sigma^2}},\ h_t \geq 0, \tag{6.58}$$

where σ^2 is the variance of Gaussian x_I and x_Q. In order to find the conjugate prior of (6.58), the Rayleigh pdf is decomposed into the form of (6.10),

$$b(h_t) = h_t,\ g(\sigma^2) = \frac{1}{\sigma^2},\ \phi(\sigma^2) = -\frac{1}{\sigma^2},\ u(h_t) = \frac{h_t^2}{2}. \tag{6.59}$$

Using (6.11), the conjugate prior of Rayleigh likelihood should be proportional to

$$P(\sigma^2) \propto g(\sigma^2)^\alpha \exp\left(\phi(\sigma^2)\beta\right) \tag{6.60}$$

$$\propto \left(\sigma^2\right)^{-\alpha} \exp\left(-\frac{\beta}{\sigma^2}\right). \tag{6.61}$$

for some parameter α and β. From (6.61), it is clear that the conjugate prior should be a inverse-gamma distribution

$$P(\sigma^2) = \frac{1}{\Gamma(\alpha)}\beta^\alpha(\sigma^2)^{-\alpha-1}\exp\left(-\frac{\beta}{\sigma^2}\right),\ \sigma^2 \geq 0 \tag{6.62}$$

Using this conjugate prior, we can derive the predictive probability

$$\begin{aligned} &P(h_t|r_{t-1}, \mathbf{h}^{r_{t-1}}) \\ &= \int_{\sigma^2} P(h_t|\sigma^2)P(\sigma^2|\alpha^{r_{t-1}}, \beta^{r_{t-1}})d\sigma^2 \\ &= \frac{h_t\alpha^{r_{t-1}}\left(\beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}}}{\left(\frac{h_t^2}{2} + \beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}+1}}. \end{aligned} \tag{6.63}$$

Also the updating equations of the hyperparameter set $\{\alpha, \beta\}$ are

$$\alpha^{r_t} = \alpha^{r_{t-1}} + 1, \text{ and} \tag{6.64}$$

$$\beta^{r_t} = \beta^{r_{t-1}} + \frac{h_t^2}{2}. \tag{6.65}$$

The derivation of (6.63), (6.64) and (6.65) are shown in Appendix A.2.

MAP estimator: To estimate σ^2, the only parameter of the Rayleigh pdf, we find the one that maximizes the current posterior probability $P(\sigma^2|\mathbf{h}^{r_t})$, which is also the form of inverse-gamma pdf

$$P(\sigma^2|\mathbf{h}^{r_t}) = P(\sigma^2|\alpha^{r_t}, \beta^{r_t}) = \frac{1}{\Gamma(\alpha_t)}\left(\beta^{r_t}\right)^{\alpha^{r_t}}\left(\sigma^2\right)^{-\alpha^{r_t}-1}\exp^{-\frac{\beta^{r_t}}{\sigma^2}}. \tag{6.66}$$

Taking the derivative of the logarithm of $P(\sigma^2|\mathbf{h}^{r_t})$ with respect to σ^2, we obtain

$$\frac{\partial \log P(\sigma^2|\mathbf{h}^{r_t})}{\partial \sigma^2} = -(\alpha^{r_t}+1)(\sigma^2)^{-1} + \beta^{r_t}(\sigma^2)^{-2}. \tag{6.67}$$

By setting the derivative to 0, the MAP estimator of σ^2 at time t corresponding to the run length $\hat{r}_t$ can be found as

$$\left(\sigma^2\right)_t^{\text{MAP}} = \frac{\beta^{\hat{r}_t}}{\alpha^{\hat{r}_t}+1}. \tag{6.68}$$

MMSE estimator: Since the posterior probability $P(\sigma^2|\mathbf{h}^{r_t})$ is an inverse-gamma pdf with parameters α^{r_t} and β^{r_t}, the MMSE estimator for σ^2, which is the conditional mean, can easily be found as

$$\left(\sigma^2\right)_t^{\text{MMSE}} = \frac{\beta^{\hat{r}_t}}{\alpha^{\hat{r}_t}-1}. \tag{6.69}$$

Since σ^2 is directly related to the power of the channel envelope, detecting the abrupt changes in the Rayleigh fading channel is equivalent to detecting the changes in the power of the envelope, which could be caused by the movement of the surrounding objects or the movement of the transmitter/receiver. However, Rayleigh fading model is only suitable for the NLOS scenario. If the transmission is changing from NLOS to LOS, change point detection using Rayleigh model may not perform properly. In the next section, we will discuss using Nakagami-m model to solve this problem.

6.4.4 Nakagami-m Distribution

Nakagami-m distribution [101] was developed to model the empirical channel envelope and was found to be a good fit in a variety of fading environments. The distribution is very versatile and simple in the way that it can be used to model the severe and weak fading channels under either NLOS or LOS scenario through the fading parameter m. Also, it leads to analytical expressions for the performance analysis of many communication systems [48, 50, 51].

The pdf of Nakagami-m distribution is written as

$$f(h_t) = \frac{2}{\Gamma(m)} \left(\frac{m}{\Omega}\right)^m h_t^{(2m-1)} e^{-\frac{mh_t^2}{\Omega}},\ h_t \geq 0,\ m \geq \frac{1}{2}, \tag{6.70}$$

where Ω is the expected second moment of h_t. The parameter m indicates the level of fading. The smaller m is, the severer the fading. Nakagami-m is equivalent to several known distributions. For example, it is half-Gaussian when $m = 0.5$, and is Rayleigh when $m = 1$. It can also approximate the Rice distribution

$$f(h_t) = \frac{h_t}{\sigma^2} \exp\left(-\frac{h_t^2 + \nu^2}{\sigma^2}\right) I_0\left(\frac{h_t \nu}{\sigma^2}\right) \tag{6.71}$$

$$= 2\frac{h_t(k+1)}{\nu^2 + 2\sigma^2} \exp\left\{-\frac{(1+k)h_t^2}{\nu^2 + 2\sigma^2} - k\right\} I_0\left(2h_t\sqrt{\frac{k(1+k)}{\nu^2 + 2\sigma^2}}\right), \tag{6.72}$$

where $k = \nu^2/2\sigma^2$ is the Rician K-factor which is the ratio between the power of the direct transmission path (the dominant path whose amplitude is not random) and the power of scatters (the random part of the channel gain). Rice pdf (6.72) with parameters k and σ is well-approximated by Nakagami-m through the conversions [100]

$$m = \frac{(1+k)^2}{1+2k}, \tag{6.73}$$

$$\Omega = 2\sigma^2(1+k). \tag{6.74}$$

Nakagami-m accounts for various fading environments and lends itself to an appealing model for channel envelope.

To apply Nakagami-m to the Bayesian change point detection algorithm, the conjugate prior needs to be found. First, we replace the parameter $1/\Omega$ with γ for the convenience of the future derivations and rewrite the Nakagami-m pdf into,

$$f(h_t) = \frac{2}{\Gamma(m)} (m\gamma)^m \exp\left\{-m\gamma h_t^2 + (2m-1)\log h_t\right\}. \tag{6.75}$$

Then decompose the pdf into the form of (6.10) and yields

$$b(h_t) = 2,\ g(m,\gamma) = \frac{1}{\Gamma(m)} (m\gamma)^m, \tag{6.76}$$

$$\boldsymbol{\phi}(m,\gamma) = \left[-m\gamma\ \ 2m-1\right]^T,\ \mathbf{u}(h_t) = \left[h_t^2\ \ \log h_t\right]^T. \tag{6.77}$$

The form of conjugate prior can be derived using (6.11),

$$P(m,\gamma|n,\nu,s,p) \propto \frac{1}{\Gamma^n(m)} (m\gamma)^{\nu m} \exp\left\{-(m\gamma)\, s + (2m-1)\log p\right\}, \tag{6.78}$$

with hyperparameter set $\boldsymbol{\eta} = \{n, \nu, s, p\}$. There is no known distribution in the exponential family with pdf in the form of (6.78). However, it is not difficult to find the normalization constant of (6.78) by numerical integral. Let K be the normalization constant such that $\frac{1}{K\Gamma^n(m)} (m\gamma)^{\nu m} \exp\left\{-(m\gamma)\, s + (2m-1)\log p\right\}$ is a eligible pdf. Namely,

$$K(\boldsymbol{\eta}) = \int_m \int_\gamma \frac{1}{\Gamma^n(m)} (m\gamma)^{\nu m} p^{2m-1} \exp\left\{-ms\gamma\right\} d\gamma dm. \tag{6.79}$$

From (6.11) we can observe that the conditional pdf $f(\gamma|m)$ can be written into a gamma distribution, so K can be simplified as

$$K(\boldsymbol{\eta}) = \int_m \int_\gamma \underbrace{\frac{(ms)^{\nu m+1}}{\Gamma(\nu m+1)} \gamma^{(\nu m+1)-1} e^{-ms\gamma}}_{f(\gamma|m)} d\gamma \frac{\Gamma(\nu m+1)}{ms^{\nu m+1}} \frac{p^{2m-1}}{\Gamma^n(m)} dm \tag{6.80}$$

$$= \int_{1/2}^{\infty} \frac{\Gamma(\nu m+1)}{\Gamma^n(m)} \frac{p^{2m-1}}{ms^{\nu m+1}} dm. \tag{6.81}$$

To find K, we applied the Laplace's method [102], in which the integrand is approximated by a Gaussian pdf. Suppose we would like to find $\int_x f(x)dx$ for some given function $f(x)$ and it is equivalent as finding $\int_x \exp\{\log f(x)\}\, dx$. Let $q(x) = \log f(x)$ and $x^* = \max_x q(x)$. The second order Taylor's approximation of $q(x)$ around x^* is $\hat{q}(x) = q(x^*) + \frac{q''(x^*)}{2}(x-x^*)^2$, since $q'(x^*) = 0$. The idea of the Laplace's method is to approximate $f(x)$ by $\exp\{\hat{q}(x)\}$, which gives

$$\int_x f(x)dx \approx \int_x \exp\{\hat{q}(x)\}\, dx$$

$$= \exp\{q(x^*)\} \sqrt{\frac{2\pi}{|q''(x^*)|}} \int_x \frac{1}{\sqrt{2\pi}\sqrt{\frac{1}{|q''(x^*)|}}} \exp\left\{-\frac{(x-x^*)^2}{2\frac{1}{|q''(x^*)|}}\right\} dx \tag{6.82}$$

$$= f(x^*) \sqrt{\frac{2\pi}{|q''(x^*)|}} \int_x \mathcal{N}\left(x; x^*, \frac{1}{|q''(x^*)|}\right) dx. \tag{6.83}$$

The integral of a Gaussian pdf in (6.83) can be expressed by the Q-function, which is defined as $Q(x) = \int_x^\infty 1/\sqrt{2\pi}\exp(-t^2/2)\,dt$. The Q-function is widely used in statistics and the performance analysis of communication systems and can be easily approached by numerical approximation. Hence, to apply the Laplace's method, one only needs to find $q''(x)$ and x^*.

Using Laplace's method to find K in (6.81), first we define

$$q(m) = \log \frac{\Gamma(\nu m + 1)}{\Gamma^n(m)} \frac{p^{2m-1}}{m s^{\nu m+1}}. \tag{6.84}$$

The second derivative of $q(m)$ is

$$q''(m) = \nu^2 \psi^{(1)}(\nu m + 1) - n\psi^{(1)}(m) + m^{-2}, \tag{6.85}$$

where $\psi^{(k)}(m)$ is the polygamma function defined as the $(k+1)^{\text{th}}$ derivative of the logarithm of the gamma function. From (6.83) and (6.85),

$$K(\boldsymbol{\eta}) \approx \frac{\Gamma(\nu m^* + 1)}{\Gamma^n(m^*)} \frac{p^{2m^*-1}}{m s^{\nu m^*+1}} \sqrt{\frac{2\pi}{|q''(m^*)|}} Q\left(\frac{\frac{1}{2} - m^*}{\sqrt{1/|q''(m^*)|}}\right). \tag{6.86}$$

Therefore, m^*, the m that maximizes $q(m)$, is the only thing left we need to solve. To find m^*, we set $q'(m)$ to zero; namely

$$q'(m^*) = \nu\psi^{(0)}(\nu m^* + 1) - n\psi^{(0)}(m^*) - \frac{1}{m^*} + \log\frac{p^2}{s^\nu} = 0. \tag{6.87}$$

From [103], it is shown that $\psi^{(0)}(m)$ can be approximated by

$$\psi^{(0)}(m) = \log(m) - \frac{1}{2m} - \frac{1}{12m^2} + \frac{1}{120m^4} - \frac{1}{252m^6} + O\left(\frac{1}{m^8}\right). \tag{6.88}$$

By only taking the first term in the series above and ignoring $1/m^*$ in (6.87), we can find a crude estimate of m^*

$$m^* \approx \left(\frac{s}{\nu p^{2/\nu}}\right)^{\frac{\nu}{\nu-n}}. \tag{6.89}$$

Take this estimate as a initial point and perform Newton's method, we can find a more accurate m^*. In Fig. 6.1, an example of using Laplace's approximation

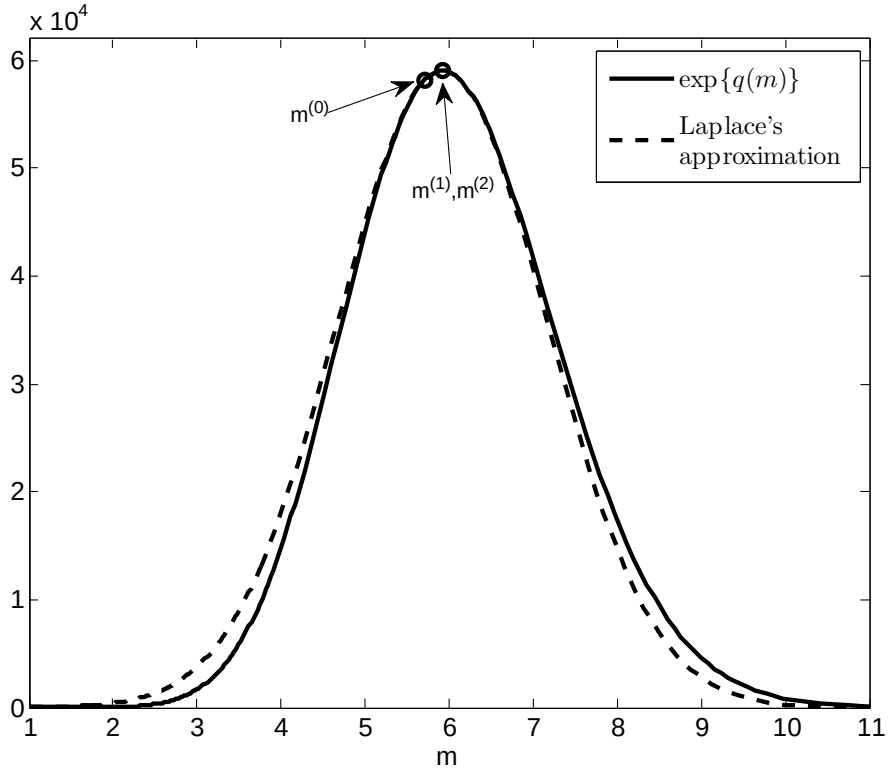


Figure 6.1: An example of using Laplace's approximation for $\exp\{q(m)\}$. In this example, $n = 10,\ \nu = 7,\ s = 5,$ and $p = 3$. The results of each iteration of the Newton's method for searching m^* are also shown, where $m^{(k)}$ is the result of k^{th} iteration and $m^{(0)}$ is the initial value.

for the original integrand $\exp\{q(m)\}$ is shown. In this example, the parameters of $q(m)$ are set to $n = 10,\ \nu = 7,\ s = 5,$ and $p = 3$. The original integrand $\exp\{q(m)\}$ is fairly close to a Gaussian bell shape, so $\exp\{\hat{q}(m)\}$ leads a good approximation. The results of the Newton's method for searching m^* are also shown in the figure,where $m^{(k)}$ is the result of k^{th} iteration and $m^{(0)}$ is the initial value. In this example, the Newton's method converges in only two iterations since the initial estimate is close to the true m^*.

With $K(\boldsymbol{\eta})$ and the conjugate prior form in (6.78), we are able to derive the predictive probability

$$P(h_t|r_{t-1}, \mathbf{h}^{r_{t-1}}) = \frac{2K(\boldsymbol{\eta}^{r_t})}{K(\boldsymbol{\eta}^{r_{t-1}})}, \tag{6.90}$$

and the updating equations for the hyperparameter set $\boldsymbol{\eta}^{r_t}$,

$$n^{r_t} = n^{r_{t-1}} + 1, \tag{6.91}$$

$$\nu^{r_t} = \nu^{r_{t-1}} + 1, \tag{6.92}$$

$$p^{r_t} = p^{r_{t-1}} \cdot h_t, \tag{6.93}$$

$$s^{r_t} = s^{r_{t-1}} + h_t^2. \tag{6.94}$$

Equations (6.90) and (6.91)–(6.94) are derived in Appendix A.3.

MAP estimators: To estimate the parameters of the Nakagami-m pdf based on the observations in the current segment $\mathbf{h}^{r_t}$ by MAP criterion, we select the ones that maximize the posterior probability $P(m, \gamma|\mathbf{h}^{r_t})$ which is also in the same form as in (6.78),

$$P(m, \gamma|\mathbf{h}^{r_t}) = P(m, \gamma|\boldsymbol{\eta}^{r_t}) = \frac{(m\gamma)^{\nu^{r_t} m} \left(p^{r_t}\right)^{2m-1}}{K(\boldsymbol{\eta}^{r_t})\Gamma^{n^{r_t}}(m)} \exp\left\{-s^{r_t} m\gamma\right\}. \tag{6.95}$$

First we take the partial derivative of the logarithm of $P(m, \gamma|\mathbf{h}^{r_t})$ with repsect to γ and set to 0,

$$\frac{\partial \log P(m, \gamma|\mathbf{h}^{r_t})}{\partial \gamma} = \frac{\nu^{r_t} m}{\gamma} - s^{r_t} m = 0. \tag{6.96}$$

The MAP estimator of γ at time t corresponding to $\hat{r}_t$ can be easily solved,

$$\gamma_t^{\text{MAP}} = \frac{\nu^{\hat{r}_t}}{s^{\hat{r}_t}}. \tag{6.97}$$

Note that the MAP estimator of Ω is not $1/\gamma_t^{\text{MAP}}$ since the MAP estimator does not commute over nonlinear transformations [97].

On the other hand, by setting $\partial \log P(m, \gamma | \mathbf{h}^{r_t})/\partial m = 0$, we obtain

$$\log m - \frac{n^{r_t}}{\nu^{r_t}} \psi^{(0)}(m) = \frac{s^{r_t}\gamma}{\nu^{r_t}} - \frac{2 \log p^{r_t}}{\nu^{r_t}} - \log \gamma - 1. \tag{6.98}$$

Plugging in the MAP estimator of γ, (6.98) becomes

$$\log m - \frac{n^{r_t}}{\nu^{r_t}} \psi^{(0)}(m) = \log \frac{s^{r_t}}{\nu^{r_t} \left(p^{r_t}\right)^{2/\nu^{r_t}}}. \tag{6.99}$$

Apply the approximation in (6.88) and preserve the first three terms of the series, the MAP estimator for m can be approximated by the solution of

$$\left(1 - \frac{n^{r_t}}{\nu^{r_t}}\right) \log m + \frac{1}{2m} + \frac{1}{12m^2} = \log \frac{s^{r_t}}{\nu^{r_t} \left(p^{r_t}\right)^{2/\nu^{r_t}}}. \tag{6.100}$$

m_t^{MAP} can be solved from (6.100) by numerical methods. A faster but less accurate way to approach m_t^{MAP} is by observing from (6.91) and (6.92) that $\lim_{r_t \to \infty} \frac{n^{r_t}}{\nu^{r_t}} = 1$, so that (6.100) can be simplified to a quadratic equation

$$12 \log \frac{s^{r_t}}{\nu^{r_t} \left(p^{r_t}\right)^{2/\nu^{r_t}}} m^2 - 6m - 1 = 0. \tag{6.101}$$

Consequently, the MAP estimator at time t corresponding to $\hat{r}_t$ has a closed-form expression

$$\hat{m}_t^{\text{MAP}} = \frac{6 + \sqrt{36 + 48 C^{\hat{r}_t}}}{24 C^{\hat{r}_t}}, \text{ where } C^{\hat{r}_t} = \log \frac{s^{\hat{r}_t}}{\nu^{\hat{r}_t} \left(p^{\hat{r}_t}\right)^{2/\nu^{\hat{r}_t}}}. \tag{6.102}$$

Note that the other root of (6.101) is not considered since it is never positive. The estimator in (6.102) is accurate as we collect more data from the same segment such that $\frac{n^{r_t}}{\nu^{r_t}}$ is close to 1. Also, it is known from the estimation theory that the MAP estimator converges to maximum likelihood (ML) estimator as the number of observations grows large [97]. Our MAP result in (6.102) indeed converges to the ML estimator derived in [52].

MMSE estimators: To derive the MMSE estimators, we make use of the integral in (6.17). For estimating m,

$$m_t^{\text{MMSE}} = \int_m \int_\gamma m \cdot P(m, \gamma | \boldsymbol{\eta}^{r_t}) d\gamma dm \tag{6.103}$$

$$= \int_m \int_\gamma \frac{m^{\nu^{r_t} m+1}}{K(\boldsymbol{\eta}^{r_t}) \Gamma^{n^{r_t}}(m)} (\gamma)^{\nu^{r_t} m} (p^{r_t})^{2m-1} \exp\{-s^{r_t} m\gamma\} d\gamma dm \tag{6.104}$$

$$= \int_m \frac{m^{\nu^{r_t} m+1}}{K(\boldsymbol{\eta}^{r_t}) \Gamma^{n^{r_t}}(m)} (p^{r_t})^{2m-1} \int_\gamma (\gamma)^{\nu^{r_t} m} \exp\{-s^{r_t} m\gamma\} d\gamma dm \tag{6.105}$$

$$= \int_m \frac{\Gamma(\nu^{r_t} m + 1)}{K(\boldsymbol{\eta}^{r_t}) \Gamma^{n^{r_t}}(m)} \frac{(p^{r_t})^{2m-1}}{(s^{r_t})^{\nu^{r_t} m+1}} dm \tag{6.106}$$

By the expression of $K(\boldsymbol{\eta})$ in (6.81), the MMSE estimator of m can also be written as

$$m_t^{\text{MMSE}} = \frac{\int_m \frac{\Gamma(\nu^{r_t} m+1)}{\Gamma^{n^{r_t}}(m)} \frac{(p^{r_t})^{2m-1}}{(s^{r_t})^{\nu^{r_t} m+1}} dm}{\int_m \frac{\Gamma(\nu^{r_t} m+1)}{\Gamma^{n^{r_t}}(m)} \frac{(p^{r_t})^{2m-1}}{m(s^{r_t})^{\nu^{r_t} m+1}} dm}. \tag{6.107}$$

As for γ, the MMSE estimator can also be derived as

$$\gamma_t^{\text{MMSE}} = \int_m \int_\gamma \gamma \cdot P(m, \gamma | \boldsymbol{\eta}^{r_t}) d\gamma dm \tag{6.108}$$

$$= \int_m \int_\gamma \frac{m^{\nu^{r_t} m}}{K(\boldsymbol{\eta}^{r_t}) \Gamma^{n^{r_t}}(m)} (\gamma)^{\nu^{r_t} m+1} (p^{r_t})^{2m-1} \exp\{-s^{r_t} m\gamma\} d\gamma dm \tag{6.109}$$

$$= \int_m \frac{m^{\nu^{r_t} m}}{K(\boldsymbol{\eta}^{r_t}) \Gamma^{n^{r_t}}(m)} (p^{r_t})^{2m-1} \int_\gamma (\gamma)^{\nu^{r_t} m+1} \exp\{-s^{r_t} m\gamma\} d\gamma dm \tag{6.110}$$

$$= \int_m \frac{\Gamma(\nu^{r_t} m + 2)}{K(\boldsymbol{\eta}^{r_t}) \Gamma^{n^{r_t}}(m)} \frac{(p^{r_t})^{2m-1}}{m^2 (s^{r_t})^{\nu^{r_t} m+2}} dm. \tag{6.111}$$

Replace $K(\boldsymbol{\eta}^{r_t})$ by its integral form, the MMSE estimator of γ can be written as

$$m_t^{\text{MMSE}} = \frac{\int_m \frac{\Gamma(\nu^{r_t} m+2)}{\Gamma^{n^{r_t}}(m)} \frac{(p^{r_t})^{2m-1}}{m^2 (s^{r_t})^{\nu^{r_t} m+2}} dm}{\int_m \frac{\Gamma(\nu^{r_t} m+1)}{\Gamma^{n^{r_t}}(m)} \frac{(p^{r_t})^{2m-1}}{m(s^{r_t})^{\nu^{r_t} m+1}} dm}, \tag{6.112}$$

Both integrals in the numerators of (6.107) and (6.112) can be computed by Laplace's method in the same way as it was done for $K(\boldsymbol{\eta}^{r_t})$ in the denominators.

6.4.5 Weibull Distribution

Weibull distribution [104] has been used to model the multipath fading channels for a wide range of propagation environments, including both indoor and outdoor transmission. [41, 105–107]. The Weibull fading channel model is also applied for performance analysis ofr various digital communication systems [47, 49, 108].

The Weibull pdf is written as

$$f(h_t) = \frac{\alpha}{\beta} h_t^{\alpha-1} e^{-\frac{h_t^{\alpha}}{\beta}},\ h_t \geq 0, \tag{6.113}$$

where $\alpha > 0$ is the shape parameter and $\beta > 0$ is the scale parameter. It is equivalent to an exponential pdf with $\alpha = 1$, and is equivalent to a Rayleigh pdf with $\alpha = 2$. While the shape parameter α is varying, Weibull pdf does not belong to exponential family since it does not have the form in (6.9). Besides, it does not have a conjugate prior for the likelihood of α, since there is no sufficient statistics of fixed dimension [109]. On the other hand, if α is fixed, Weibull is an exponential family and the conjugate prior for likelihood of the scale parameter β can easily be found as a inverse-gamma distribution [109]

$$p(\beta) = \frac{1}{\Gamma(p)} q^p \beta^{-p-1} \exp\left(-\frac{q}{\beta}\right), \tag{6.114}$$

where the hyperparameter set $\boldsymbol{\eta}$ consists of $p > 0$ and $q > 0$. In the remaining of this chapter, we will only discuss the case when α is fixed and β is varying.

With the conjugate prior in (6.114), we can derive the predictive probability

for the Weibull fading channel envelope h_t

$$\begin{aligned} & p(h_t|r_{t-1}, \mathbf{h}^{r_{t-1}}) \\ &= \int_\beta p(h_t|\beta) p(\beta|p^{r_{t-1}}, q^{r_{t-1}}) d\beta \\ &= \frac{\alpha h_t^{\alpha-1} p^{r_{t-1}} (q^{r_{t-1}})^{p^{r_{t-1}}}}{(h_t^\alpha + q^{r_{t-1}})^{p^{r_{t-1}}+1}}. \end{aligned} \tag{6.115}$$

The updating equations for p^{r_t} and q^{r_t} when observing the new data h_t are

$$p^{r_t} = p^{r_{t-1}} + 1, \tag{6.116}$$

$$q^{r_t} = q^{r_{t-1}} + h_t^\alpha. \tag{6.117}$$

The derivations of (6.115), (6.116) and (6.117) are shown in Appendix A.4. As previously mentioned, Rayleigh distribution is a special case of Weibull distribution when $\alpha = 2$. The results in (6.63)–(6.65) are also the special cases of (6.115)–(6.117) repectively as $\alpha = 2$.

Next, we derive the MAP and MMSE estimators for the unknown parameter β with fixed α.

MAP estimator: Using the fact that the posterior probability of β is still an inverse-gamma distribution, we are able to find the derivative of $\log P(\beta|\mathbf{h}^{r_t})$,

$$\frac{\partial \log P(\beta|\mathbf{h}^{r_t})}{\partial \beta} = \frac{\partial \log P(\beta|p^{r_t}, q^{r_t})}{\partial \beta} = -\frac{p^{r_t}+1}{\beta} + \frac{q^{r_t}}{\beta^2}. \tag{6.118}$$

Set the derivative to 0 and the MAP estimator of β at time t corresponding to the run length $\hat{r}_t$ can be solved as

$$\beta_t^{\text{MAP}} = \frac{q^{\hat{r}_t}}{p^{\hat{r}_t}+1}. \tag{6.119}$$

As r_t approaches to ∞, this MAP estimator converges to the ML estimator as shown in [54]. Not surprisingly, the MAP estimator for the Rayleigh parameter σ^2 in (6.68) is also the special case of (6.119) as $\alpha = 2$.

MMSE estimator: Since the posterior probability $P(\beta|\mathbf{h}^{r_t})$ is an inverse-gamma pdf with parameters p^{r_t} and q^{r_t}, the MMSE estimator of β is

$$\beta_t^{\text{MMSE}} = E[\beta|\mathbf{h}^{\hat{r}_t}] = \frac{q^{\hat{r}_t}}{p^{\hat{r}_t} - 1}. \tag{6.120}$$

6.4.6 Lognormal Distribution

Lognormal distribution has been used to characterize the path loss and shadowing effect of the signal amplitudes in a multipath fading environment [41,110]. It is also shown in [60,61] that the temporal variation of the Rician K-factor due to moving objects in the environment can be modeled as lognormal distribution. Unlike Rayleigh or Nakagami-m distribution, there are no strong physical interpretation of using lognormal to characterize the small-scale variation of the fading channel. Nevertheless, it has been shown in literatures that lognormal provides good fit for the empirical data in either wideband or narrow band transmission [57,111,112].

h_t is lognormal distributed if $\log(h_t)$ is Gaussian distributed. Consequently, the lognormal pdf can be written as

$$f(h_t) = \frac{1}{h_t\sigma\sqrt{2\pi}}\, e^{-\frac{(\log h_t - \mu)^2}{2\sigma^2}}, \quad h_t > 0 \tag{6.121}$$

where μ and σ^2 are the mean and variance of $\log h_t$, respectively. Since the logarithm of h_t is Gaussian distributed and the changing parameters are the mean and variance of this Gaussian, one can simply perform the Gaussian change point detection on $\log h_t$. Nevertheless, change point detection also works directly on lognormal h_t, since it is exponential family and the conjugate prior can easily be found. Decomposing the lognormal pdf into the form of (6.9), we obtain the same $g(\sigma^2)$ and $\phi(\mu, \sigma^2)$ as the ones of Gaussian distribution. As a result, the conjugate prior of lognormal likelihood of μ and σ^2 is also Gaussian-inverse-gamma distribution as in 6.19. The predictive probability and the updating equations of the hyperparameter set η_t can also be derived in the similar way as in Section 6.4.1.

The predictive probability is

$$
\begin{aligned}
&p(h^{r_t}|r_{t-1}, \mathbf{h}^{r_{t-1}}) \\
&= \int_{\mu}\int_{\sigma^2} p(h^{r_t}|\mu,\sigma^2)p(\mu,\sigma^2|m^{r_{t-1}},\tau^{r_{t-1}},\alpha^{r_{t-1}},\beta^{r_{t-1}})d\sigma^2 d\mu \qquad (6.122)\\
&= \frac{1}{h^{r_t}\sqrt{2\pi\beta^{r_{t-1}}(1+\tau^{r_{t-1}})}}\frac{\Gamma(\alpha^{r_{t-1}}+\frac{1}{2})}{\Gamma(\alpha^{r_{t-1}})}\left\{1+\frac{(\log h^{r_t}-m^{r_{t-1}})^2}{2\beta^{r_{t-1}}(1+\tau^{r_{t-1}})}\right\}^{-(\alpha^{r_{t-1}}+1/2)} \qquad (6.123)\\
&= \frac{1}{h_t}\mathcal{T}\left(\log h^{r_t}\,\middle|2\alpha^{r_{t-1}}, \frac{\alpha^{r_{t-1}}}{\beta^{r_{t-1}}(\tau^{r_{t-1}}+1)}, m^{r_{t-1}}\right), \qquad (6.124)
\end{aligned}
$$

which can be expressed by a Student-t distribution scaled by $1/h_t$ or a log Student-t distribution. The scaling factor is due to the variable conversion. The updating equations of the hyperparameter set are

$$
\begin{aligned}
m^{r_t} &= \frac{m^{r_{t-1}}+\tau^{r_{t-1}}\log h^{r_t}}{1+\tau^{r_{t-1}}}, &(6.125)\\
\tau^{r_t} &= \frac{\tau^{r_{t-1}}}{1+\tau^{r_{t-1}}}, &(6.126)\\
\alpha^{r_t} &= \alpha^{r_{t-1}}+\frac{1}{2}, &(6.127)\\
\beta^{r_t} &= \beta^{r_{t-1}}+\frac{(\log h^{r_t}-m^{r_{t-1}})^2}{2(1+\tau^{r_{t-1}})}. &(6.128)
\end{aligned}
$$

These equations are almost the same as the ones of one-dimensional Gaussian, except that h_t is replaced by $\log h_t$.

Since the conjugate prior of lognormal likelihood is the same as the one of one-dimensional Gaussian, that implies the posterior probabilities $P(\mu,\sigma^2|\mathbf{h}^{r_t})$ are in the same form. Consequently, the MAP and the MMSE estimators of μ and σ^2 are also in the same forms as the ones of Gaussian.

MAP estimators: The MAP estimators of μ and σ^2 at time t corresponding to the estimated run length $\hat{r}_t$ are,

$$
\begin{aligned}
\mu_t^{\mathrm{MAP}} &= m^{\hat{r}_t}, &(6.129)\\
\left(\sigma^2\right)_t^{\mathrm{MAP}} &= \frac{\beta^{\hat{r}_t}}{\alpha^{\hat{r}_t}+\frac{3}{2}}. &(6.130)
\end{aligned}
$$

MMSE estimators: The MMSE estimators of μ and σ^2 are

$$\mu_t^{\mathrm{MMSE}} = m^{\hat{r}_t},\ (\sigma^2)_t^{\mathrm{MMSE}} = \frac{\beta^{\hat{r}_t}}{\alpha^{\hat{r}_t} - 1}. \tag{6.131}$$

The derivations of (6.124)–(6.131) are omitted since they are very similar to the ones for one-dimensional Gaussian.

CHAPTER 7

Evaluation of the Fading Channel Modeling and Parameter Estimation Based on Change Point Detection

7.1 Test on Random Sequences

To validate the change point detection (CPD) methods and parameter estimators for the 5 distributions discussed in Chapter 6, we first test with the pseudo random sequences of the corresponding distribution generated by Matlab. The data are independent and identically distributed. Each sequence is manually partitioned into 4 non-overlapped segments. The parameters for each segment are also chosen manually. After the change points are detected, the sequential MAP and MMSE estimators are also shown for each segment (between detected change points).

One-dimensional Gaussian Distribution: The results of the CPD on a Gaussian random sequence are shown in Fig. 7.1. The parameters for the segments are $\mu = \{0.3, 0, 0, 0.15\}$ and $\sigma = \{0.35, 0.5, 0.3, 0.3\}$. To see the capability of the CPD algorithm under the scenario that only one of the parameters changes slightly, we purposely let the second and the third segment have the same mean but different variances, and let the third and the fourth segments have different means but the same variances. The true change point locations are $t = \{301, 451, 651\}$ and the detected change point locations are $t = \{302, 451, 636\}$ which are fairly close to the true ones. The detected change points sometime locate even earlier

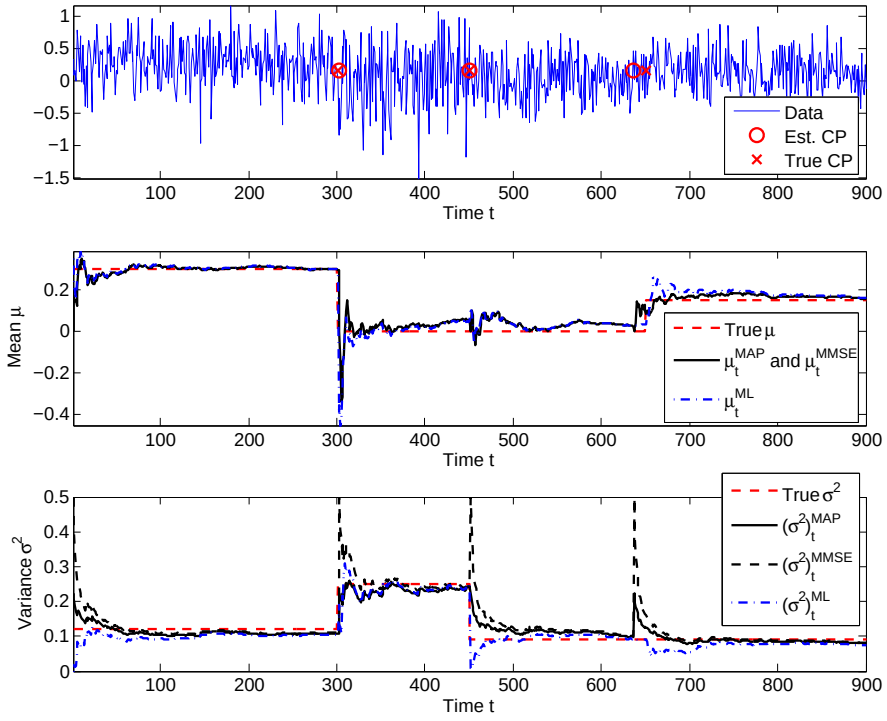


Figure 7.1: Results of the Gaussian change point detection on a Gaussian random sequence with 4 segments. The parameters of the segments are $\mu = \{0.3, 0, 0, 0.15\}$ and $\sigma = \{0.35, 0.5, 0.3, 0.3\}$. The MAP and the MMSE estimations of μ and σ^2 are shown in the second and third subplots, respectively.

than the true ones. This happens when the parameters are not substantially different between segments, and the detectors may regard the last few samples of the previous segment as part of the next segment. On the other hand, the detector marked the change point later than the true ones because the first few samples of the current segment may also have high probability to be sampled from the distribution of the previous segment.

The sequential MAP and the MMSE estimators for μ and σ^2 are also shown in Fig. 7.1. The difference between the MAP and the MMSE estimators are only noticeable at the beginning of each segment. As observing more data of the same segment, both estimators are close to the true value and the difference between them are very small. For comparison, we also show the sequential ML estimator

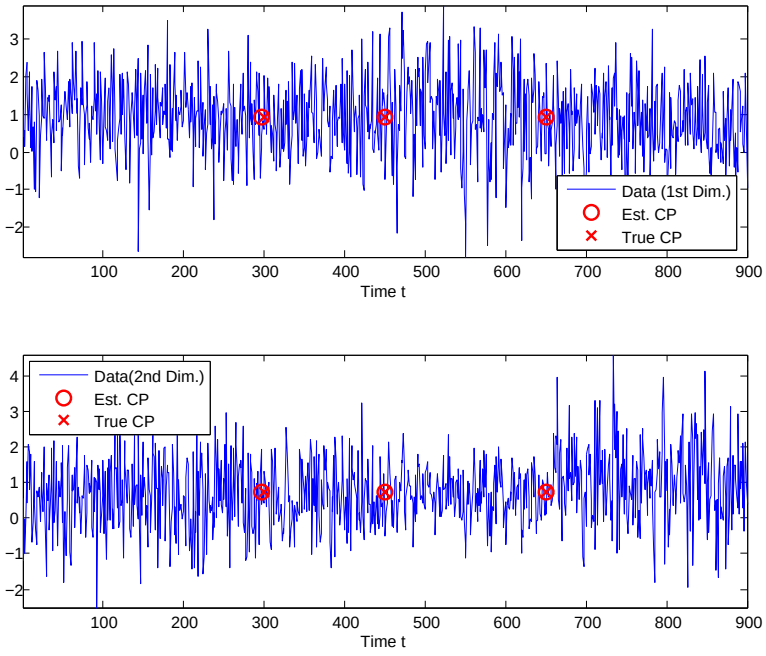


Figure 7.2: Results of the 2-D Gaussian change point detection on a bivariate Gaussian random sequence with 4 segments. The detection is performed jointly on both dimensions.

in the same figure. Note that this sequential ML estimator is computed based on the full knowledge of the change point locations and thus can be viewed as one of the best statistical estimators if the change point locations are provided. It can be seen from the figure, our MAP and MMSE estimators work almost as well as the sequential ML estimator in this example. Also, they converge to the ML estimator as the number of observations increases. For off-line processings, one does not need sequential estimators. We can simply use the MAP or the MMSE estimator from the last time instant of the segment, which in this example we showed that the results are fairly close to the true values even with only 150 to 300 samples.

Multivariate Gaussian Distribution: A bivariate Gaussian random vector sequence is generated. The sequence is partitioned into 4 segments with mean vectors

$$\boldsymbol{\mu} = \{[1 \;\; 0.5]^T, [0.8 \;\; 0.6]^T, [1.1 \;\; 0.7]^T, [0.8 \;\; 1]^T\}, \tag{7.1}$$

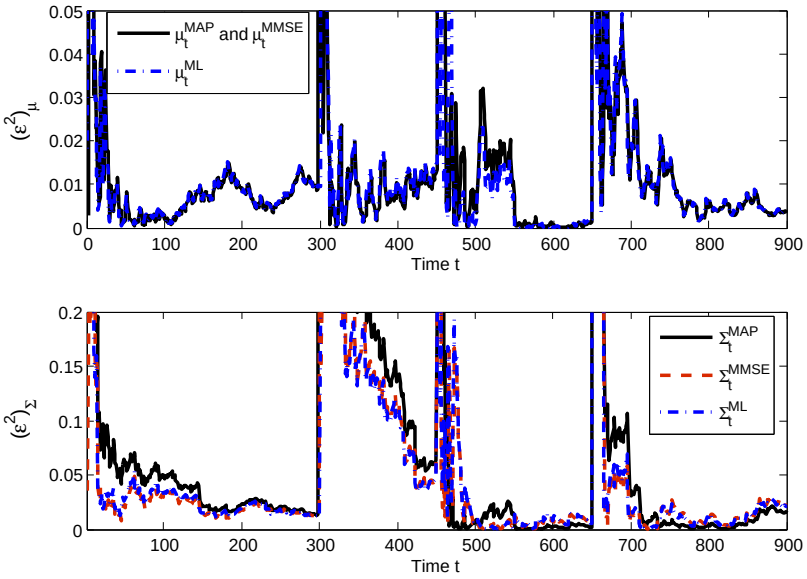


Figure 7.3: Normalized squared errors of the MAP and the MMSE estimators for μ and Σ of a bivariate Gaussian pdf. The sequential ML estimators with the full knowledge of the change point locations are also shown for reference.

and covariance matrices $\mathbf{\Sigma}$,

$$\mathbf{\Sigma} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \begin{bmatrix} 1.5 & 0.7 \\ 0.7 & 0.4 \end{bmatrix}, \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 1.1 \end{bmatrix} \right\}. \tag{7.2}$$

The covariance matrices cover the cases of independence, weak correlation, and strong correlation between two dimensions. The CPD results are shown in Fig. 7.2, where the Gaussian sequence of each dimension is shown in one subplot. Note that even though we marked the estimated change points on each dimension, they are in fact jointly estimated and share the same outcome. To picturize the sequential estimation results of $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, the normalized squared errors of the MAP and MMSE estimators for $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are shown in Fig. 7.3. The normalized squared errors are defined as

$$(\epsilon_t^2)_{\boldsymbol{\mu}} = \frac{\left\| \boldsymbol{\mu}_t - \boldsymbol{\mu}_t^{EST} \right\|_2^2}{\left\| \boldsymbol{\mu}_t \right\|_2^2}, \tag{7.3}$$

$$(\epsilon_t^2)_{\mathbf{\Sigma}} = \frac{\left\| \mathbf{\Sigma}_t - \mathbf{\Sigma}_t^{EST} \right\|_F^2}{\left\| \mathbf{\Sigma}_t \right\|_F^2}, \tag{7.4}$$

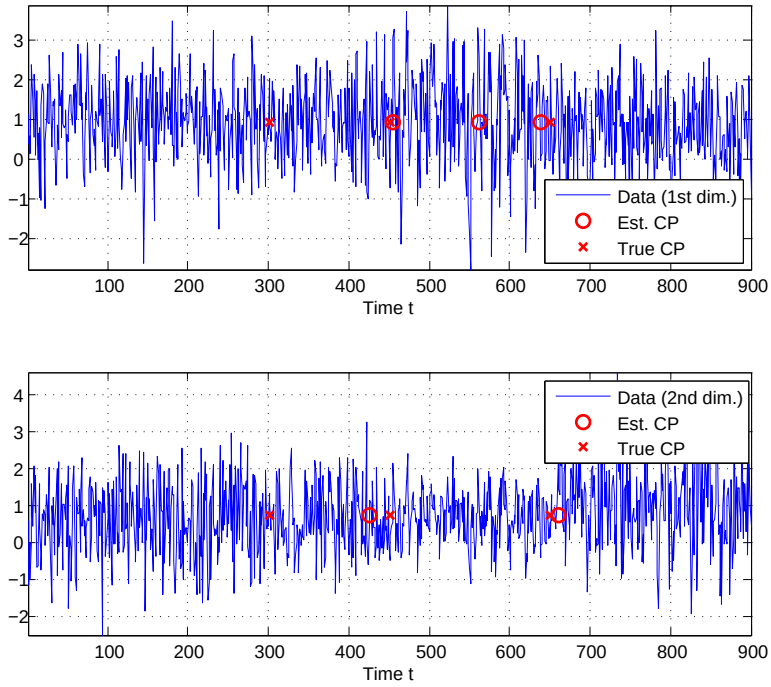


Figure 7.4: 1-D Gaussian change point detection individually on each dimension of the same bivariate Gaussian random sequence as in Fig. 7.2.

where $\|\cdot\|_F$ is the matrix Frobenius norm and $\|\cdot\|_2$ is the vector Euclidean norm. The sequential ML estimation based on the full knowledge of the true change point locations is also shown for reference. For the mean vector estimation, both the MAP and the MMSE estimation have almost the same performance as the ML estimation. For the covariance matrix, the MMSE estimator is still close to the ML, while the MAP estimator in the last two segments are slightly better than the other two. The length of the second segment is relatively short compared to the other three, so the normalized estimation error of the covariance matrix at the end of the segment is over 5%, which indicates that the estimator has not converged.

To see the difference between multivariate and 1-D Gaussian CPD, we perform the 1-D Gaussian CPD on each dimension of the same bivariate Gaussian random sequence. The results are shown in Fig 7.4. The main change between the first two segments are the cross-correlation of two dimensions, while the variance of each

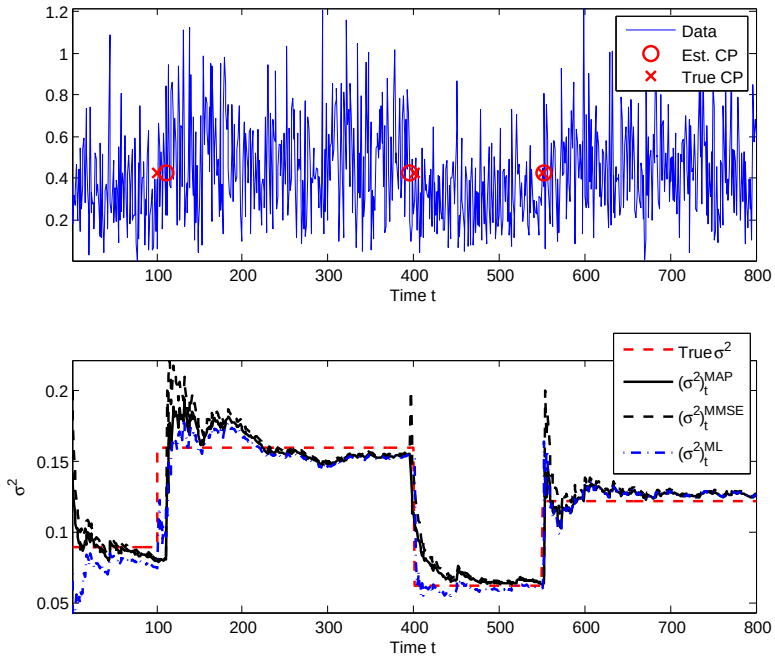


Figure 7.5: Results of the Rayleigh change point detection on a Rayleigh random sequence with 4 segments. The parameters σ^2 of the segments are $\{0.09, 0.16, 0.0625, 0.1225\}$, respectively. The MAP and the MMSE estimations of σ^2 are shown in the second subplot.

dimension is unchanged. Performing detection separately on each dimension was not able to detect the change. Besides, the accuracy of the change point locations are not as good as performing joint detection. This shows the advantage of joint detection on multiple dimensions.

Rayleigh Distribution: The four segments of the Rayleigh random sequence are generated with parameters $\sigma^2 = \{0.09, 0.16, 0.0625, 0.1225\}$, respectively. The CPD results are shown in Fig. 7.5. The true change point locations are $t = \{101, 401, 551\}$ and the detected change point locations are $t = \{111, 396, 553\}$. This random sequence simulates the scenario of an NLOS transmission with different levels of shadowing, which could be caused by people walking in between the path. Therefore, the amplitude of the channel gain varies over a short period of time.

The parameter estimation results are also shown in Fig. 7.5, where the sequential ML estimation with full knowledge of the change point locations are also shown for comparison. It is worth-noting that although all the estimators did not perform well in the first segment due to small number of observations, the MAP and the MMSE estimation yield closer-to-true-value results. This is because the proper initial hyperparameters of the conjugate prior could lead to better estimation at the early stage of the estimation. As the number of observations increases to infinity, all these estimators will converge to the same value, which are the cases of the second and the fourth segments. For fading channel measurement, the envelope level is usually within a limited range. Therefore, the initial hyperparameters of the conjugate prior could be properly set to match the range of interest for a better performance.

Nakagami-m Distribution: The power of the Nakagami envelope is known to be a gamma distribution [100]. Therefore, to generate the Nakagami-m random sequence, we make use of the gamma random number generator in Matlab Statistics Toolbox and take the square root of it. The parameters of the 4 segments are $m = \{1, 2, 4, 3\}$ and $\gamma = \{1, 1.42, 0.83, 1.33\}$. As mentioned in Chapter 6, Nakagami-m distribution covers wide range of fading scenarios, including the common Rayleigh and Rician fading. For $m = 1$, a Nakagami-m random variable is equivalent to a Rayleigh random variable, while $m > 1$ it can well approximate Rice distribution. Hence, the random sequence we generated covers these two different fading channels, which simulates the situation when the transmission is turning from NLOS to LOS.

Fig. 7.6 shows the CPD and parameter estimation results. The true change point locations are $t = \{251, 551, 751\}$ and the estimated change point locations are $t = \{262, 551, 751\}$. The ML estimator for m and γ are from [52]. For the γ estimation, all the estimators have similar behavior and approach to the true values, except for the second segment. However, for the m estimation, the MMSE

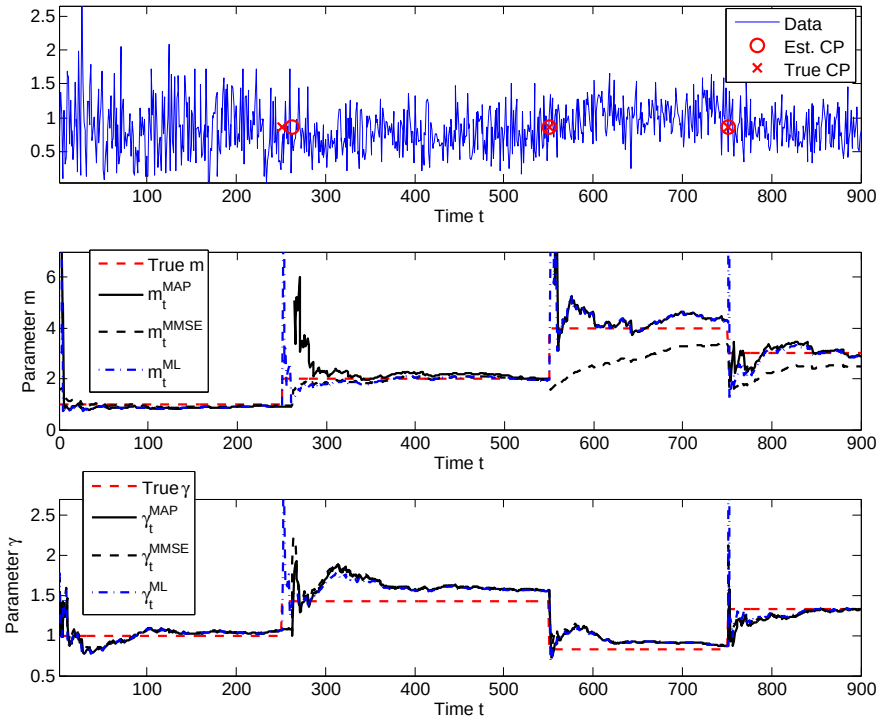


Figure 7.6: Results of the Nakagami-m change point detection on a Nakagami-m random sequence with 4 segments. The parameters of the segments are $m = \{1, 2, 4, 3\}$ and $\gamma = \{1, 1.42, 0.83, 1.33\}$. The MAP and the MMSE estimations of m and σ^2 are shown in the second and third subplots, respectively.

estimator is no longer close to MAP estimator, and does not perform well for the third and the fourth segments. This may be caused by the numerical inaccuracies of the two numerical integrals, as shown in (6.107), in the simulation. The error could be further amplified especially division is involved. Using the MAP estimator of m does not require numerical integrals or other numerical methods, which is much more computationally efficient. Therefore, we will only consider the MAP estimators for m and γ in the remaining of this chapter.

Weibull Distribution: We use two Weibull sequences with the same $\beta = \{0.45, 0.25, 0.4, 0.2\}$ for each segment but α is fixed at 1 or 3 for each sequence, respectively. The fixed α is known to detector, so we are only detecting and estimating the change of β. The CPD and the parameter estimation results for

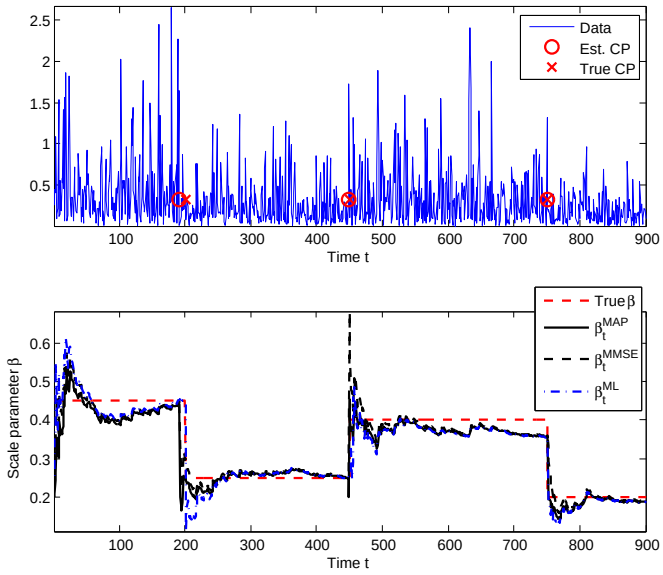


Figure 7.7: Results of the Weibull change point detection on a Weibull random sequence with 4 segments. The parameters of the segments are $\beta = \{0.45, 0.25, 0.4, 0.2\}$ and α is fixed at 1. The MAP and the MMSE estimations of β are shown in the second subplot.

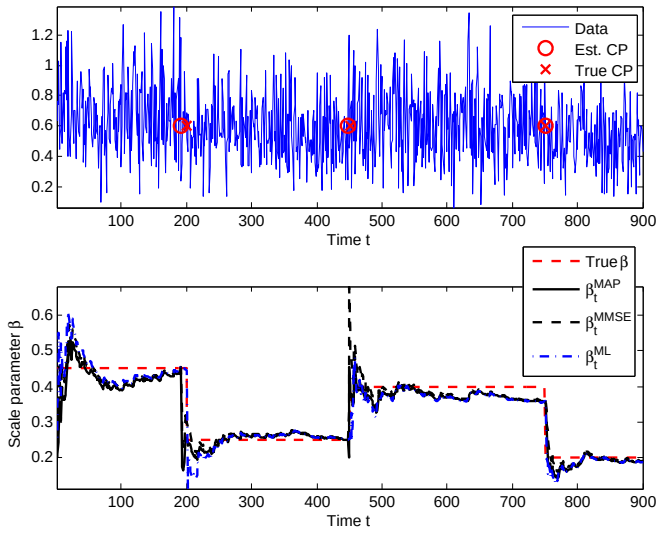


Figure 7.8: Results of the Weibull change point detection on a Weibull random sequence with 4 segments. The parameters of the segments are $\beta = \{0.45, 0.25, 0.4, 0.2\}$ and α is fixed at 3. The MAP and the MMSE estimations of β are shown in the second subplot.

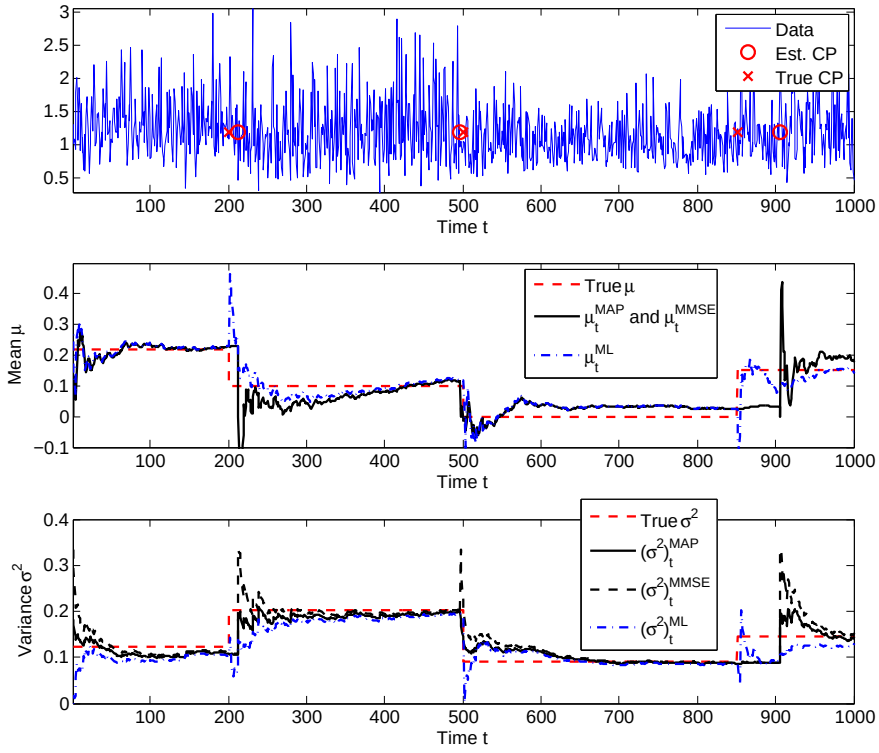


Figure 7.9: Results of the lognormal change point detection results on a lognormal random sequence with 4 segments. The parameters of the segments are $\mu = \{0.22, 0.1, 0, 0.15\}$ and $\sigma = \{0.35, 0.45, 0.3, 0.38\}$. The MAP and the MMSE estimations of μ and σ^2 are shown in the second and third subplots, respectively.

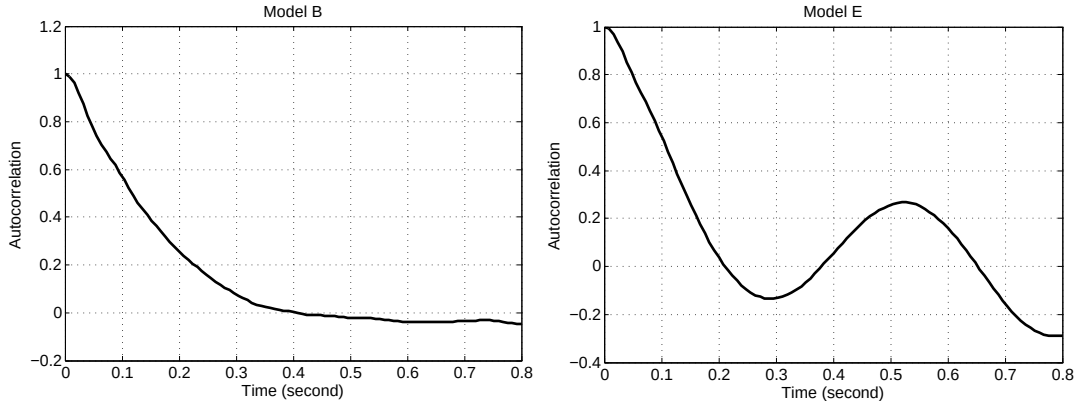
the sequence with $\alpha = 1$ are shown in Fig. 7.7. Note this random sequence is equivalent to an exponential random sequence. As we change α from 1 to 3, the CPD works equally well as the case when $\alpha = 1$. The estimators also behave similarly in these two examples. Fig. 7.8 shows the CPD and β estimation results for the Weibull sequence with $\alpha = 3$.

Lognormal Distribution: The change point locations are set to be $t = \{201, 501, 851\}$, and the parameters of the 4 segments of the lognormal random sequence are $\mu = \{0.22, 0.1, 0, 0.15\}$ and $\sigma = \{0.35, 0.45, 0.3, 0.38\}$. The CPD and parameter estimation results are shown in Fig. 7.9. The estimated change point locations are $t = \{212, 496, 905\}$. The reason that the third estimated change point

is way behind the true one can be seen from the estimation result of σ^2. It can be observed from $(\sigma^2)_t^{ML}$ between $t = 851$ to 905 that the variance of that short period is very close to the true variance value of the previous segment. Therefore, the detector classified that period to the wrong segment. After $t = 905$, the MAP and the MMSE estimators for σ^2 quickly converge toward the true value. However, the ML estimator using the full knowledge of the change point locations does not react that fast because the time average is being "pulled down" by the data in the period $t = 851 \sim 905$.

7.2 Test on 802.11n WLAN Channel Simulator

In this section, we test the CPD algorithm and the parameter estimators on the simulated channels generated using the model proposed by the IEEE 802.11 TGn channel model special committee [113, 114]. The channel simulator is part of the IST project (IST-2000-30148 I-METRA) and is available at [115]. Six different MIMO wireless local area network (WLAN) channel models (Models A–F) are considered in the simulator, which cover from the smaller environments, such as typical residential homes, to large open space, such as big offices or outdoor environments. The distance between the transmitter and the receiver, the number of antennas at each end, the carrier frequency and the channel sampling frequency can be user-defined. For our applications, we always choose single transmit and single receive antenna and fix the carrier frequency at 2.4 GHz. The transmission distance varies between different channel segment in order to obtain different signal levels. The channel generated is time correlated, which is against the independence assumption in the CPD and parameter estimation algorithm. As pointed out in [55, 58, 59], as the channel sampling interval increases, the time correlation between samples drops rapidly and can be viewed as "approximately independent". In Fig. 7.10, we show the autocorrelation function of channel model B and model



(a) (b)

Figure 7.10: Autocorrelation function of the simulated channel operating at 2.4 GHz using (a) Model B (b) Model E.

E, which are the two channel models we will consider in the following discussion. Model B simulates the typical residential environment with 15 ns rms delay spread and model E is for typical large open space and office environments with 100 ns rms delay spread. As we can see from the figure, the autocorrelation drops below 0.4 after 0.15 second separation. Consequently, we set the sampling interval to be 0.2 second in all the simulation.

First, we test on the channel generated by model B, with 4 different transmission distances, 3, 4, 5 and 6 meters. The lengths of the channels are 40, 50, 36 and 48 seconds, respectively. We concatenate the 4 segments in order into a long channel sequence, and thus 3 artificial change points. There are 9 channel taps in model B. Since we assume narrow band transmission, which implies that the received signal is a summation of all taps. It is known the channel is generated based on Rayleigh assumption, so we will only perform Rayleigh and Nakagami-m CPDs on the channel envelope. The channel sequence has been normalized to have average power equal to 1. The results of a Rayleigh CPD are shown in Fig. 7.11. Despite the third estimated change points is about 3.4 seconds early, it seems that the Rayleigh CPD is able to detect all the power changes in the channel envelope. From

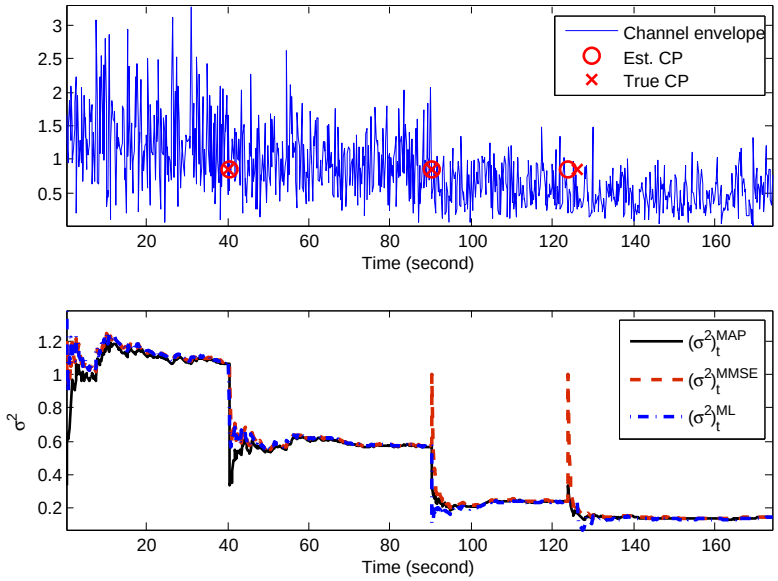


Figure 7.11: Results of the Rayleigh change point detection on model B channel and the MAP and the MMSE estimations of the Rayliegh parameter σ^2.

the parameter estimation results, the MAP and the MMSE estimators both reach to a steady value after collecting data of about 20 seconds (100 samples). They also stay very close to the ML estimator using full knowledge of the true change point locations. To further verify the simulated channel is Rayleigh distributed, we perform the Nakagami-m CPD. The Nakagami-m CPD results are shown in Fig. 7.12. The CPD results are very similar to the one of Rayleigh CPD. From the parameter estimation results, it shows that the channel envelop is indeed a Rayleigh distribution since the estimated m values of each segment are all around 1. To be precise, the estimated m at the last time instant of each estimated segment are $\{1.033, 1.0342, 0.9396, 0.9958\}$. So the changing parameter is only γ, which is directly related to the signal power level.

Next, we test the Nakagami-m CPD on the channel changing between NLOS and LOS. Hence, not only the power level changes but envelope distribution is changing between the Rayleigh and the Rice as well. In the simulator, a strong LOS component is added to the first tap of model E if the transmission distance is

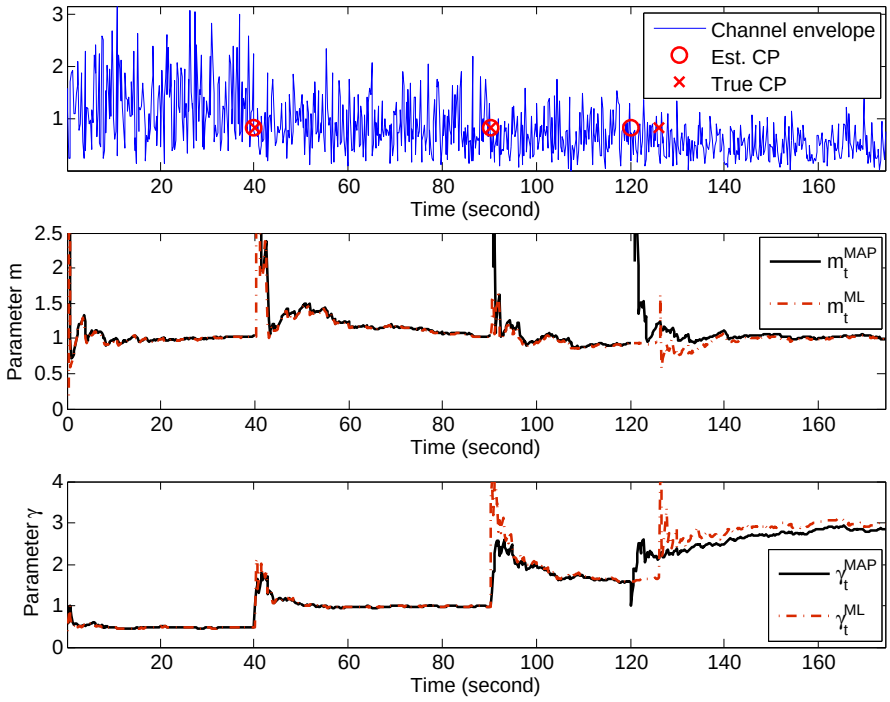


Figure 7.12: Results of the Nakagami-m change point detection on model B channel and the MAP and the MMSE estimations of the Nakagami parameter m and γ

below 20 meters. Hence, we test our Nakagami-m CPD on the first tap of model E channel, with distances set to $\{19, 20, 21, 18\}$ and the corresponding length $\{40, 50, 36, 42\}$ seconds. The results are shown in Fig. 7.13. It shows that the Nakagami-m CPD not only detect the power level changes within Rayleigh fading (at 90 seconds), but can detect the change from LOS to NLOS (at 40 seconds) and from NLOS to LOS (at 126 seconds) as well. The MAP estimated m and γ of the first segment is 2.7423 and 0.5812 repectively, which are equivalent to Rice-K factor 3.9281 and Rice parameter $\sigma^2 = 0.1746$ using the conversion in (6.73) and (6.74). To see the performance of the parameter estimators, we compare between the empirical CDF of the first segment and the Rice CDF with the estimated parameters, which is shown in Fig 7.14. The results show a very good fit using the estimated parameters.

The I- and Q- component of the complex channel path follows the Gaussian

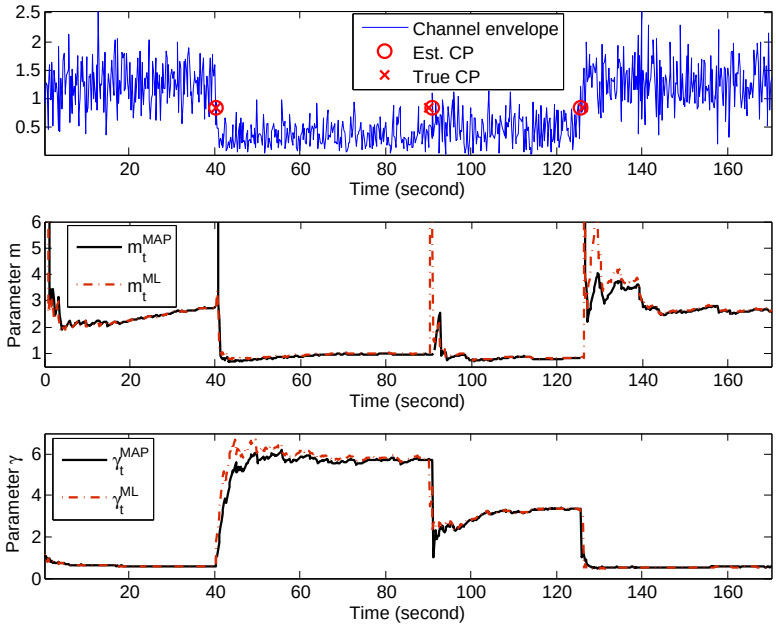


Figure 7.13: Results of the Nakagami-m change point detection on model E channel, including the LOS and the NLOS transmissions. The MAP and the MMSE estimations of the Nakagami-m parameters m and γ are shown in the second subplot.

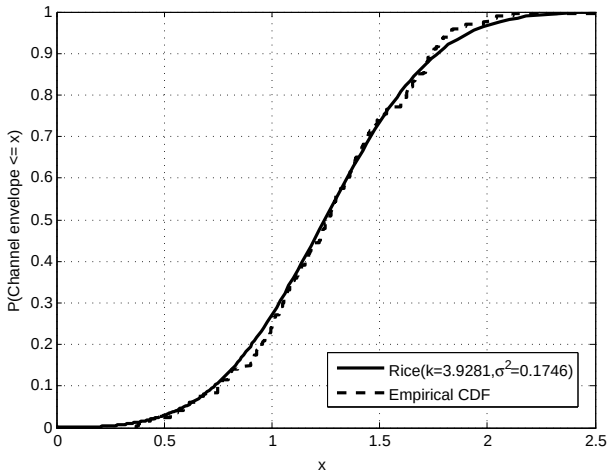


Figure 7.14: Comparison between the empirical CDF of the first segment of the model E channel with the Rice CDF with estimated parameters $k = 3.9281$ and $\sigma^2 = 0.1746$.

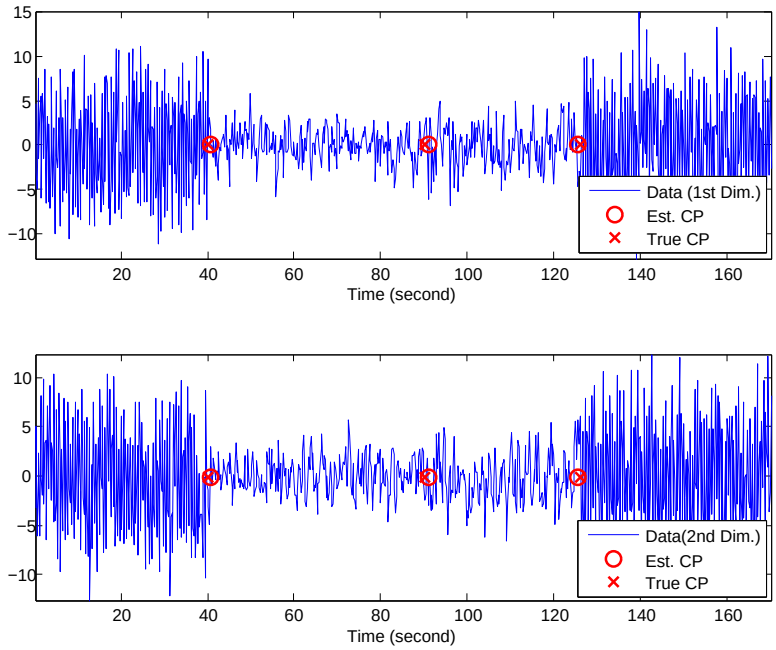


Figure 7.15: Performing the 2-D Gaussian change point detection on the I- and Q-component of the complex channel gain.

distribution in the simulator. Using the same model E channel, we test the 2-D Gaussian CPD on the complex channel gain. The CPD results are in Fig. 7.15. It shows the effectiveness of 2D Gaussian CPD, if the coherent complex channel gain is available. The covariance estimation results of each segment, which the numbers are not shown, indicate weak correlation between I- and Q- channel.

7.3 Test on Channel Measurement

The channel measurement was conducted in the Public Safety Network Systems (PSNS) lab of UCLA. The room was surrounded by wooden walls and glasses windows. Both transmitter and receiver are the radio module SC2000 by Silvus Technologies. The carrier frequency is set to 2.49 GHz. In each experiment, 4000 channel measurement were captured over an 8 minutes period. To reduce the correlation between contiguous samples, the measurement is downsampled by 4.

Namely, the channel sampling interval is roughly 0.48 second. The transmitter and the receiver were fixed at all time during the measurement, so the temporal variation of the channel was caused by the moving objects in the environment, such as people walking or cars driving by in the field of transmission. The time average power of the channel envelope in each experiment was normalized to 1.

In the first lab room measurement, the distance between the transmitter and the receiver is 3.2 meters and the there is a no object in between the transmission path. 4 prearranged people were walking nearby the receiver. As reported in [55], people moving around the receiver would cause larger variation in the Rician K-factor than people moving around the transmitter. Since there is a clear LOS in this experiment, we presume the envelope distribution is Rician-like. Therefore, we applied the Nakagami-m CPD on the channel envelope and the results are shown in Fig. 7.16. There are 4 detected segments and 3 change points at $t = \{300.96, 445.92, 461.28\}$ seconds respectively. For each segment, the MAP estimation of m and γ are obtained from all the data of that estimated segment, so we are able to check the goodness of fit of our estimated Nakagami-m model on the empirical data. The number of sample points in the last two estimated segments are too small, so we will only consider the first two. The MAP estimated m and γ of the first segment are 19.9613 and 0.8864 respectively. The MAP estimated m and γ of the second segment are 17.7114 and 1.2319 respectively. The empirical CDFs of the two segments and the two estimated Nakagami-m CDF are plotted in Fig. 7.17. It is clear that the distributions of the two segments are significantly different. By visualing from the figure, the two Nakagami-m distributions with the MAP estimated parameters seem fit well for the empirical data. Using the Komolgorov-Smirnov goodness-of-fit test (KS test) [116], both cases pass the test under the significance level $\alpha = 1\%$. Also, from the estimated m values the equivalent Rician K-factor are over 30, which implies a strong LOS component exists and matches to our assumption.

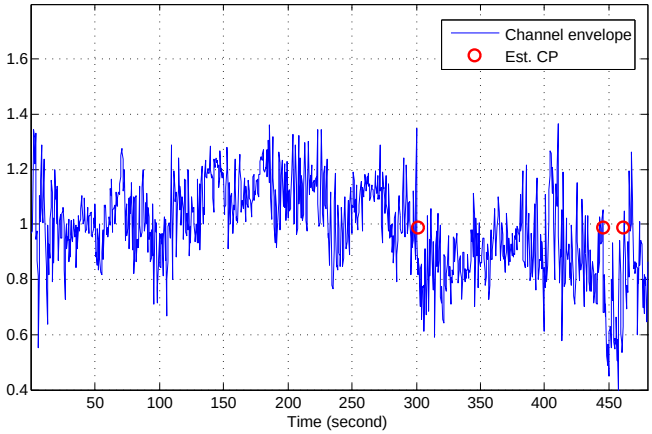


Figure 7.16: Performing the Nakagami-m change point detection on the channel envelope measured in the lab room with people walking near by the receiver.

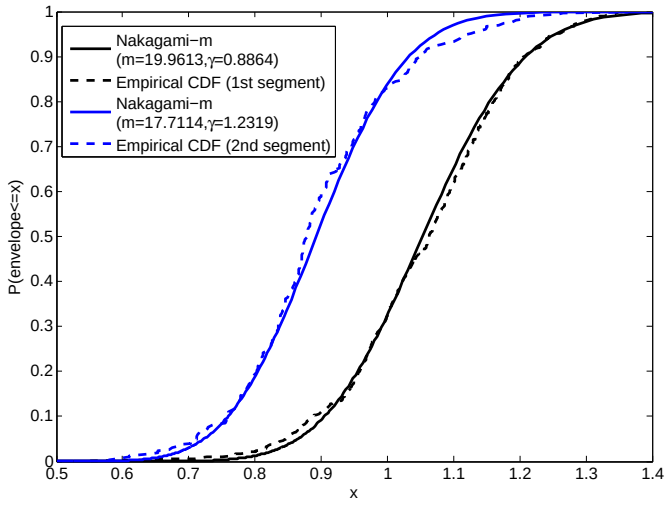


Figure 7.17: Comparison between the empirical CDFs of the detected segments from the first lab room measurement and the corresponding Nakagami-m CDFs using the MAP estimated parameters.

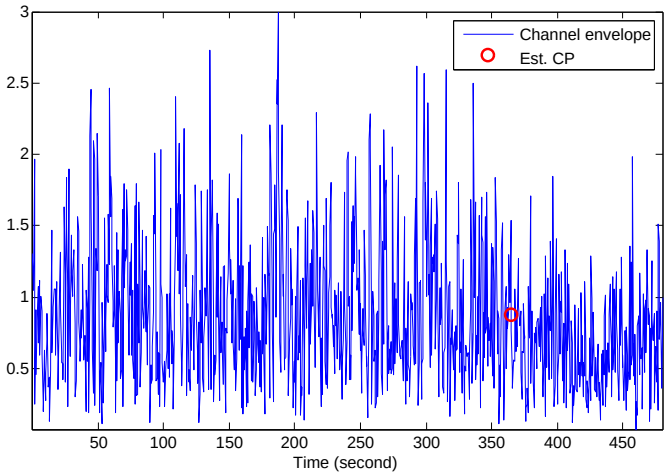


Figure 7.18: Performing the Nakagami-m change point detection on the channel envelope measured in the lab room with people walking or standing still in between the transmit and the receive antennas.

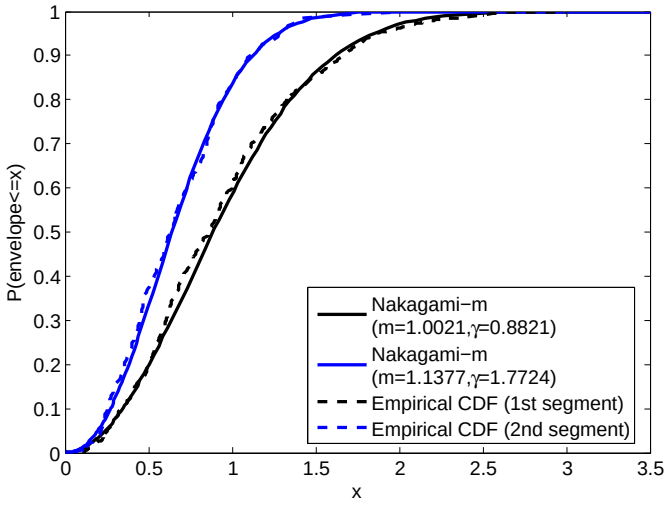


Figure 7.19: Comparison between the empirical CDFs of the detected segments from the second lab room measurement and the correponding Nakagami-m CDFs using the MAP estimated parameters.

In the second lab room measurement, the physical settings are the same as the first measurement except that people were consistently walking or standing still in between the transmit and receive antennas. Therefore, there is no direct sight between antennas at most of the time. Again, we first applied the Nakagami-m CPD since it covers a wide range of fading conditions. The results of the CPD are shown in Fig. 7.18. One change point is detected at 364.8 second. The MAP estimated m and γ of the first segment are 1.0021 and 0.8821 respectively. The MAP estimated m and γ of the second segment are 1.1377 and 1.7724 respectively. Using the estimated parameters, the comparison between the empirical CDF of the segments and its corresponding Nakagami-m CDF are shown in Fig. 7.19, and it shows a good fit in both segments. The two Nakagami-m distributions also pass the KS test with a significance level $\alpha = 1\%$. The estimated m of both segment are very close to 1 which implies there is a strong multipath fading effect and the envelope distribution is also close to a Rayleigh. Applying the Rayleigh CPD on the same channel measurement, we obtain one change point at 354.24 second, which is not far from the result of Nakagami-m CPD. The MAP estimated σ^2 for the two segments are 0.5730 and 0.3004, respectively. The MMSE estimated σ^2 are very close to the MAP results, so we will discuss only the MAP estimation. The Rayleigh with MAP estimated σ^2 for both segment both pass the KS test with a significance level $\alpha = 1\%$. The empirical CDF of the two detected segments and the corresponding Rayleigh models are shown in Fig. 7.20.

In the first experiment, Weibull CPD with various selections of α has also been tested. For α at 3 and 4, CPD was able to detect changes around 300 seconds. However, the models using the MAP and MMSE estimated parameter β do not show reasonable fit. For the second experiment, since it has been verified by Rayleigh and Nakagami-m CPD that the channel envelope is close to a Rayleigh distribution and Weibull pdf at $\alpha = 1$ is the same as a Rayleigh, so there is no need to perform Weibull CPD on the second channel measurement.

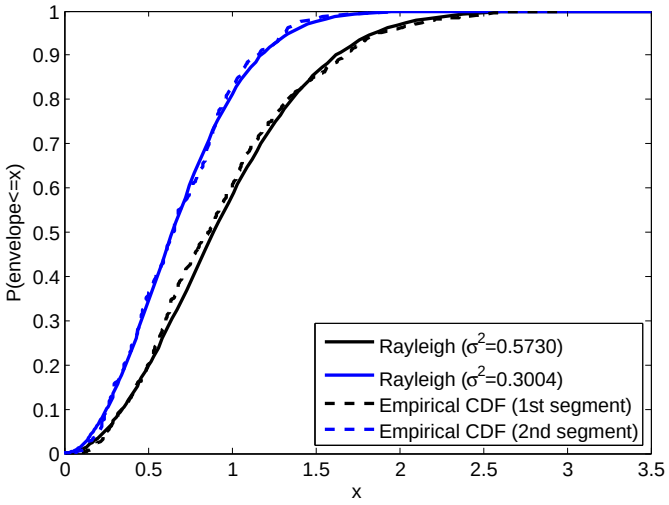


Figure 7.20: Comparison between the empirical CDFs of the detected segments from the second lab room measurement and the correponding Rayleigh CDFs using the MAP estimated parameters.

In sum, Nakagami-m CPD seems to be the most useful tool for modeling our indoor channel measurement since the pdf is very flexible to cover wide range of fading conditions. By performing Nakagami-m CPD can provide us the fading condition and the power variation of the channel envelope. The MAP or MMSE parameter estimator can be viewed as the "by-product" when performing CPD since the estimators are derived directly from the hyperparameter sets of the conjugate prior. CPD combining with parameter estimation can give us a more accurate view of modeling the empirical channel data. This is useful especially for the indoor measurement as in our experiment, since the channel envelope usually suffers from human shadowing and has larger temporal variations.

CHAPTER 8

Bird Phrase Segmentation Using Entropy-Based Change Point Detection

Similar to the temporal variation of the fading channel, bird calls cause the temporal (and also frequency domain) variation of the spectrogram. Hence, with suitable front-end processing on the spectrogram, we are able to segment the bird phrases by applying Bayesian change point detection. With the reliable automated bird phrase segmentation method, it can save a large amount of time for biologists and linguists analysing the long field recordings. In Section 8.1, the two types of front-end processing on the spectrogram, and the efficient way of calculating entropy are introduced. In Section 8.2, the way of fitting the change point detection method into the phrase segmentation is discussed. Experimental results and discussions are shown in Section 8.3.

8.1 Front-End Processing

Our proposed segmentation method begins with the calculation of the entropy sequence from the spectrogram or short term Fourier transform (STFT). The time-frequency blocks over the spectrogram are characterized by its entropy, the level of the entropy sequence should drop or rise (i.e. a change point) at the instant when a call appears or ends, respectively.

In the following, we will describe how we characterize each time-frequency block in the spectrogram by its entropy in an efficient way. Additionally, two

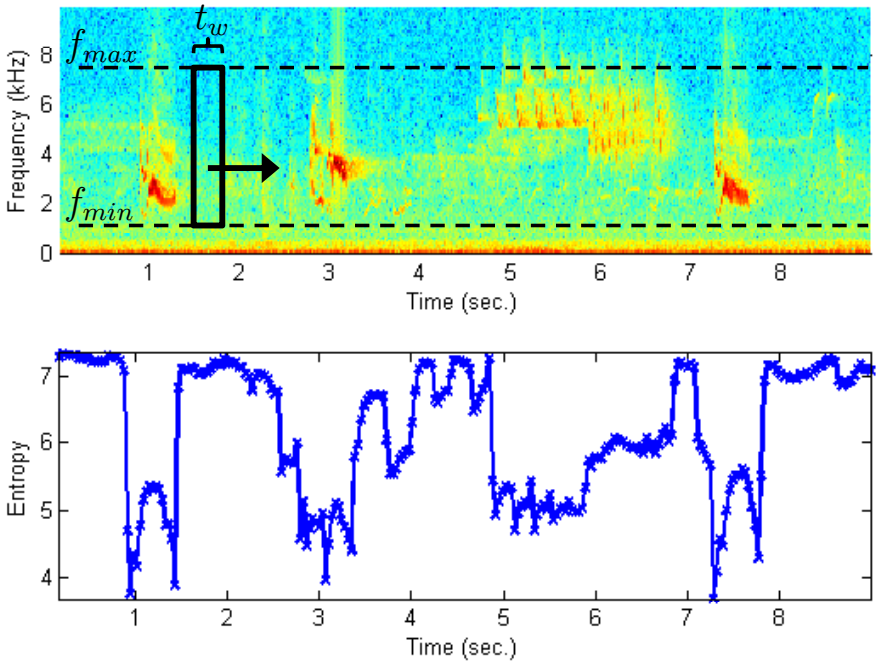


Figure 8.1: Spectrogram of a sampled bird sound recording and the corresponding entropy sequence calculated from the sliding time-frequency block.

types of front-end processing are introduced. Spectral whitening is applied to the spectrogram before calculating entropy in order to further distinguish the entropy level between a call period and a quiet period. Spectral subtraction can also be used as a front-end processing to mitigate the interference and background noise.

8.1.1 Entropy Calculation

A time-frequency block of time length t_w and containing F frequency bins from f_1 to f_F is sliding horizontally from the beginning of the spectrogram, as shown in Fig. 8.1. The selection of the block length and the frequency range depends on the targeted bird species. The frequency limits f_1 and f_F should be properly selected to cover only the frequency band of interest so that the background noise outside the band will not be recognized as bird calls. As in Fig. 8.1, the frequency range is selected to prevent the low frequency noise. The block length t_w should be no

longer than the length of the shortest quiet period between two phrases.

Entropy is calculated from each time-frequency block. Denote $p(n, f)$ as the power spectrum at time n and frequency bin f calculated by STFT. The entropy H_t at time t is obtained by

$$H_t = -\sum_{n=t-t_w+1}^{t} \sum_{f=f_1}^{f_F} z(n, f) \log z(n, f), \tag{8.1}$$

where $z(n, f)$ is the normalized power spectrum within the block, namely

$$z(n, f) = \frac{p(n, f)}{\sum_{n=t-t_w}^{t} \sum_{f=f_1}^{f_F} p(n, f)}. \tag{8.2}$$

As shown in (8.1), the entropy time index t is chosen to be the right edge of the block. So the entropy level should drop immediately as the sliding block transits from the quiet period to the beginning of a bird call. On the other hand, there would be a t_w delay in the entropy level at the end of the call. The end time labels will be adjusted, which is discussed in Section 8.2.3. There are two main reasons why we calculate the entropy from a time-frequency block instead of at every time instant: 1) The entropy sequence is smoothed to prevent the "border effect" [74,117,118], which usually causes segmentation errors at the beginning and toward the end of a call. 2) The entropy given by a block is more representative and suffers less from the bursty background noise.

However, when the block rate is high and the blocks are highly overlapped, calculating entropy purely by (8.1) and (8.2) is expensive in terms of memory consumption and computational load. To this end, an alternative expression of H_t is developed;

$$H_t = -\sum_n \sum_f \frac{p(n, f)}{\sum_n \sum_f p(n, f)} \left(\log p(n, f) - \log \sum_n \sum_f p(n, f) \right) \tag{8.3}$$

$$= -\frac{\sum_n \sum_f p(n, f) \log p(n, f)}{\sum_n \sum_f p(n, f)} + \log \sum_n \sum_f p(n, f). \tag{8.4}$$

The lower and upper limits of the summations above are the same as in (8.1) but they are not listed here for simplicity. Define $A_t = \sum_n \sum_f p(n, f)$ and

$B_t = \sum_n \sum_f p(n,f) \log p(n,f)$. By (8.4) the entropy of a block of spectrogram can be computed as

$$H_t = -\frac{B_t}{A_t} + \log A_t. \tag{8.5}$$

This calculation can be significantly reduced by saving two partial sums a_n and b_n over frequencies of interest, where $a_n \stackrel{\text{def}}{=} \sum_f p(n,f)$ and $b_n \stackrel{\text{def}}{=} \sum_f p(n,f) \log p(n,f)$, at the completion of each STFT. Then A_t and B_t can be obtained by accumulating the partial sums over the current block time to get inputs to the entropy calculation; i.e.

$$A_{t+\tau} = -\sum_{n=t-t_w+1}^{t-t_w+\tau} a_n + A_t + \sum_{n=t+1}^{t+\tau} a_n, \tag{8.6}$$

$$B_{t+\tau} = -\sum_{n=t-t_w+1}^{t-t_w+\tau} b_n + B_t + \sum_{n=t+1}^{t+\tau} b_n, \tag{8.7}$$

where τ is the block shift. For the notation simplicity, we define h_k as $H_{t_w+k\tau}$, where $k = 0, 1, 2, \cdots$. Consequently, the entropy sequence calculated from the sliding time-frequency blocks now becomes $h_0, h_1, h_2, \cdots$.

8.1.2 Spectral Cleaning and Whitening

It is known that the entropy defined in (8.1) is maximized if $z(n,f)$ in a block is uniformly distributed. Consequently, the entropy of a block with white background noise is higher than the one with color background noise. In contrast, the entropy is low as there are only few strong power components that dominate others within the block. This implies that we can get a lower entropy if we are able to mitigate the noise within a block. From the above observations, one can enlarge the difference of the entropy levels between a quiet period block and a block containing bird calls by applying either a spectral cleaning or a spectral whitening method.

8.1.2.1 Spectral Subtraction

Spectral subtraction is a well-known noise reduction technique in speech processing [119–121]. It is shown to be effective to improve the speech quality under additive noise. The basic idea is to subtract the short-term estimated noise from the noisy speech in the power spectrum domain. Since we compute the entropy also from the power spectrum, spectral subtraction can be easily integrated to our system. The processed power spectrum $\hat{p}(n, f)$ is obtained by

$$\hat{p}(n, f) = p(n, f) - \alpha \cdot \hat{p}_N(n, f), \tag{8.8}$$

where $\hat{p}_N(n, f)$ is the noise estimate from the quiet period and α is a factor depending on the SNR estimate at each frequency band [121]. To prevent negative power spectrum, $\hat{p}(n, f)$ is set to be $\epsilon \cdot p(n, f)$ if $p(n, f)$ is less than $\alpha \cdot \hat{p}_N(n, f)$, and ϵ is selected to be a small positive number.

8.1.2.2 Polynomial-based Whitening Filter

We proposed a simple polynomial-based whitening method which is computationally efficient. The basic principle is to multiply the power spectrum at time t, $\mathbf{p}_t = \begin{bmatrix} p(t, f_1) & \cdots & p(t, f_F) \end{bmatrix}^T$, by a c^{th} degree polynomial over the frequency bins of interest. The coefficients of this polynomial have to be adaptively adjusted along the time. Let the whitening filter at time t and over frequency bins f_1 to f_F be written as

$$\mathbf{Q} \cdot \mathbf{g}_t, \tag{8.9}$$

where $\mathbf{g}_t$ is a $(c + 1) \times 1$ vector representing the polynomial coefficients varying with time, and $\mathbf{Q}$ is a $F \times (c+1)$ matrix with orthonormal columns. The degree of the approximation polynomial c need not to be large because the background noise is usually not rapidly changing. A quadratic polynomial to capture the dynamic of the spectrum is used in this work. For quadratic polynomial the three basis columns of $\mathbf{Q}$ are, a constant vector, a vector linear in frequency, and a vector

quadratic in frequency. With proper shift and scale in the frequency, $\mathbf{Q}$ can easily be chosen as an orthonormal matrix. For example, the following matrix consists of three orthogonal columns.

$$\mathbf{Q} = \left[\mathbf{1} \quad \mathbf{f} \quad \mathbf{f}^2 - \frac{\mathbf{1}^T\mathbf{f}^2}{F}\right], \tag{8.10}$$

where $\mathbf{f}$ is the centralized frequency vector, $\left[f_1 \cdots f_F\right]^T - \mu_f\mathbf{1}$ and $\mu_f = \sum_{k=1}^{F} f_k/F$. With further normalization on each column, we can obtain an orthonormal matrix.

Ideally, the whitening filter gain of the power spectrum at a fixed time instant t is the reciprocal of $p(t, f)$, for all the frequency bins f of interest. With the proposed whitening filter representation (8.9), it should achieve

$$\mathbf{p}_t \circ [\mathbf{Q} \cdot \mathbf{g_t}] = \mathbf{1}. \tag{8.11}$$

However, it is not desired to whiten the bird call power spectrum along with the background noise power spectrum. To reduce the sensitivity to the sparse and high energy bird calls when present, the whitening polynomial captures the power spectrum variation in the log domain. Namely, the whitening processing can be rewritten as

$$\mathbf{p}_t \circ \exp\{\mathbf{Q} \cdot \mathbf{g}_t\} = \mathbf{1}. \tag{8.12}$$

Consequently, the polynomial coefficients $\mathbf{g}_t$ is

$$\mathbf{g}_t = -\mathbf{Q}^T \log \mathbf{p}_t. \tag{8.13}$$

Additionally, to further mitigate the effect of bird calls, our proposed whitening filter is set to be the reciprocal of the long term time-averaged power spectrum,

$$\boldsymbol{l}_t = \frac{\sum_{n=t-M+1}^{t} \log \mathbf{p}_n}{M}, \tag{8.14}$$

where M should be much larger than the number of STFT in a single bird phrase. Similar to (8.13), the polynomial coefficients now becomes

$$\mathbf{g}_t = -\mathbf{Q}^T \log \boldsymbol{l}_t. \tag{8.15}$$

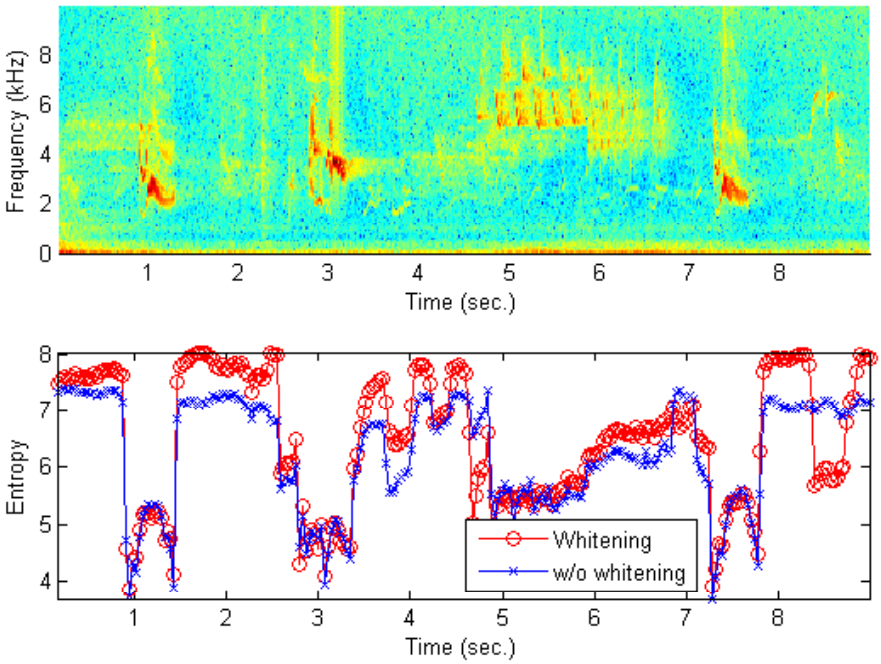


Figure 8.2: The upper figure is the spectrogram of the same recording as in Fig. 8.1 after whitening. The lower figure is the corresponding entropy sequence compared to the the one without whitening.

From (8.14) and (8.15), it can be easily shown that the polynomial coefficients $\mathbf{g}_t$ can be updated in a recursive manner as the new STFT output $\mathbf{p}_{t+1}$ is available,

$$\mathbf{g}_{t+1} = \mathbf{g}_t + \frac{-\mathbf{Q}^T \log \mathbf{p}_{t+1} - \mathbf{g}_t}{M}, \tag{8.16}$$

In Fig. 8.2, we show the spectrogram and the entropy sequence of the same recording as in Fig. 8.1 after processing by the proposed whitening method. It is clear that the background noise is much closer to a white noise compared to Fig. 8.1. Also, the entropy level of the quiet period becomes higher while the entropy of the bird call period is about the same level, which verifies the effectiveness of the proposed whitening filter.

8.2 Change Point Detection and Post-Processing

Given the entropy sequence computed by the method described in Section 8.1, we need to distinguish the bird calls from the quiet periods by its level. Instead of applying an adaptive hard threshold which is usually used in the energy-based segmentation [74–76] and KL divergence segmentation methods [77], a Bayesian change point detection method is used to judge the starting and end points of a bird phrase.

8.2.1 Bayesian Change Point Detection for Entropy

The CPD algorithm used here is the same one introduced in Section 6.2. For bird phrase segmentation, the input data to the change point detection is the entropy sequence h_t from (8.5), which we assume is a Gaussian sequence. Hence, CPD is performed by using the methodology in Section 6.4.1.

The run length transition probability $P(r_t|r_{t-1})$ in (6.2) can be modeled if there is a prior knowledge on the statistic distribution of the time between change points. Let T be a random variable defined as the time between contiguous two change points, and $F_T(t)$ be the corresponding cumulative distribution function (CDF). The probability that there is a change point at time t given the previous run length can be obtained by

$$\begin{aligned} & p(r_t = 0|r_{t-1}) \\ &= \frac{p(T \leq t \cap T > t-1)}{p(T > t-1)} \\ &= \frac{F_T(t) - F_T(t-1)}{1 - F_T(t-1)}. \end{aligned} \tag{8.17}$$

In the bird phrase segmentation, T means the length of a call period or the quiet time between calls. Therefore, $F_T(t)$ should depend on the bird species. Also, given the previous run length r_{t-1} there are only two possibilities in r_t, which are 0 and $r_{t-1} + 1$. Hence, $P(r_t = r_{t-1} + 1|r_{t-1})$ is simply $1 - p(r_t = 0|r_{t-1})$.

With run length transition probability (8.17) and the predictive probability (6.21) plugging back into (6.2), we are able find the run length posterior probability $P(r_t|\mathbf{h}_{1:t})$. For the long-term field recordings, t can be too large to save the posterior probability of all possible run lengths at time t. An upper limit of the run length, $R_{\max}$, is set so that no posterior probability of the run length beyond $R_{\max}$ would be evaluated and saved. The estimated run length at time t is determined by the maximum a posteriori probability (MAP) estimation

$$\hat{r}_t = \max_{r_t=0,1,\cdots,\min(t,R_{max})} P(r_t|\mathbf{h}_{1:t}). \tag{8.18}$$

8.2.2 Bird Phrase Segmentation

After the change points are marked, we need to determine if the segment between change points is a call period or a quiet period. The idea is to compare the time-averaged entropy of the segment with a threshold γ_h. If it is lower than γ_h, the segment is determined to be a call period; otherwise, it is a quiet period. The threshold γ_h should be adaptively adjusted over time, in the way similar to the one used in the energy detection [74]. The main difference between the proposed method with the thresholding method in energy detection is that our threshold is less sensitive to the short-term variation in the entropy due to the bursty noise, since it is updated based on the time-averaged entropy of predefined segments thanks to the change point detection.

In fact, there is no extra computation needed for the time-averaged entropy of a segment. From section 6.4.1, we know the MAP and MMSE estimator of the mean μ of a Gaussian segment is the hyperparameter m, which is the mean of the prior probability of μ. Therefore, we can simply substitute the time-averaged entropy of a segment by the m that is updated by all the data points in the segment using (6.23). They are not exactly equal due to the initial value of m, but they are very close when the number of data points in a segment is large.

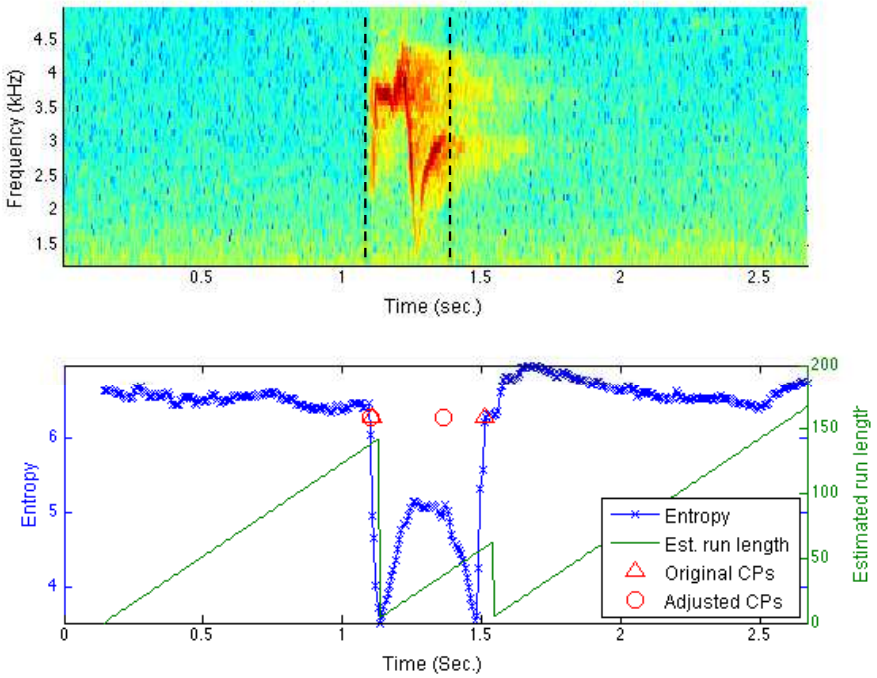


Figure 8.3: The spectrogram of a bird phrase with lower energy smear at the tail part. The black dashed lines are the boundaries of the phrase labeled by experts. The entropy, estimated run length and the change points before and after adjustment are shown in the lower figure.

8.2.3 End Label Adjustment

In the experiment, we observed that the spectrogram of a bird phrase usually contains a high energy part and a low energy smear at the tail. The spectrogram of a bird phrase is shown in Fig. 8.3, where the dark red area and the light yellow are at the tail are the high energy part and low energy part of a phrase, respectively. The black dashed line in the spectrogram represents the start and the end of the phrase labeled by experts. The low energy part is not counted as part of the phrase since the energy is too low to be heard by human. However, if the energy of that tail part is still higher than the background noise level, the entropy of that tail part is still low enough for CPD to take that as part of the call. In other words, the entropy "hears" more than we do. As shown in Fig. 8.3, the

entropy sequence is still low even after the human end label. Consequently, the marked change point by the estimated run length falls behind the human label. For phrase classification and other biology research purposes, it is desired to have the automated segmentation results to be as close to the human labels as possible. To resolve this problem, an energy thresholding method is applied to exclude the tail lower energy part from the segment.

Assume the data from time $t_1, \cdots, t_k$ have been marked as a call period by the detector, and $E(t_i)$ denotes as the energy at t_i (which is a direct result from a_n in Section 8.1.1). After adjustment, data from time t_{cut} to the end of the segment are excluded from the segment, where

$$t_{cut} = \arg \max_{t=t_1,\cdots,t_k} \left\{ E(t) \geq \epsilon \sum_{i=1}^{k} E(t_i) \right\}, \tag{8.19}$$

and ϵ should be a carefully chosen so it won't discard too much tail area. In Fig. 8.3, we also show the change point after the adjustment using $\epsilon = 0.005$. It can be seen that the new end label stays much closer to the human label compared to the original one.

8.3 Evaluation

8.3.1 Field Recordings

We evaluate the proposed segmentation method on recordings of Cassin's Vireo (*Vireo cassinii*). Thirteen separate recordings were obtained between 23 April and 8 June, 2010 near Fort Ann Mine, Amador county, California (38°29'0"N, 120°38'04"W) in a mixed conifer-oak forest at approximately 800 meters elevation. The length of each recording varies from 72 seconds to 551 seconds, and the total length is over 50 minutes. Two males on two different territories (approximately 200 meters apart) were recorded. Phrase repertoires of the two males were similar, though not identical. Songs were recorded in WAV format (16-bit, mono) at a

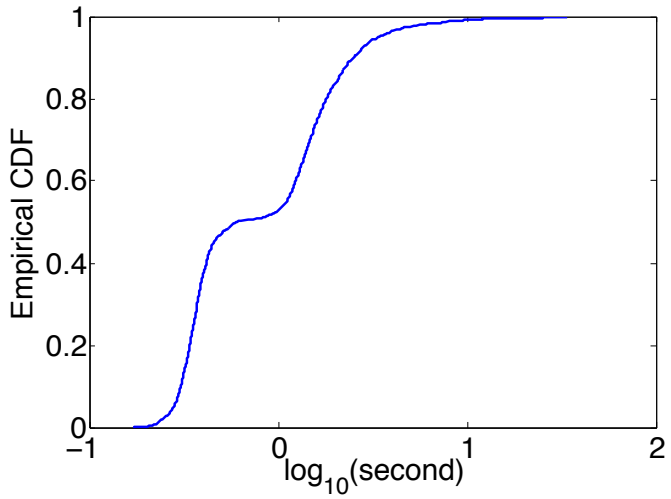
sampling rate of 44.1 kHz, using a Marantz PMD 670 with a Telinga parabolic reflector and a Sennheiser omni-directional microphone. Manual annotation was performed using software "Praat" [122] to note the phrase class, and the start and end time of each phrase in the song. The phrases are categorized into one of the 65 phrase classes based on both visual examination of their spectrograms and auditory recognition. There are total 852 phases of Cassin's Vireo annotated in the recordings that are not severely overlapped with other species' calls. For the notable calls from other species, the start and end time are labeled and are all classified as "others".

8.3.2 Experimental Results

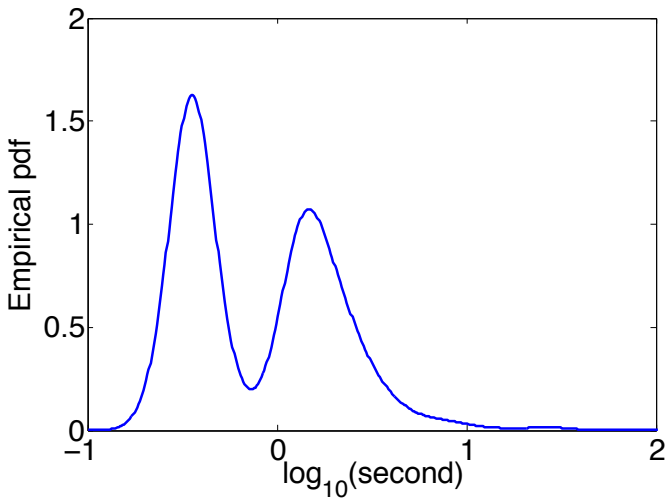
The spectrogram of the recordings are obtained by performing STFT with a Hamming window applied, FFT size 512, and the frame hop size 102 (20% of the FFT frame size). The time length t_w of the time-frequency block for calculating entropy is set to 138.8ms. The frequency range of the block is from $f_1 = 1.5$kHz to $f_F = 7$kHz. The block rate is 144Hz.

To improve the accuracy of the change point detection, the knowledge of $F_T(t)$, the distribution of the length between change points, that defines the run length transition probability as shown in (8.17) is needed. To this end, we record the lengths of each human-labeled phrase and quiet period in the recordings and find the empirical CDF by Kaplan-Meier estimator [123]. The empirical CDF and the empirical pdf computed by the kernel density estimation [124] are shown in Fig. 8.4. Note that the time is in log scale. It can be seen that the distribution is bimodal. This is because the phrase length of Cassin's Vireo is generally much shorter than the time length between calls. With the distribution information and (8.17), we can obtain the run length transition probability by a look-up table.

First, we evaluate the proposed method by the effectiveness of capturing bird



(a)



(b)

Figure 8.4: (a) The empirical CDF by Kaplan-Meier estimator and (b) the empirical pdf by kernel density estimation of the time T between change points.

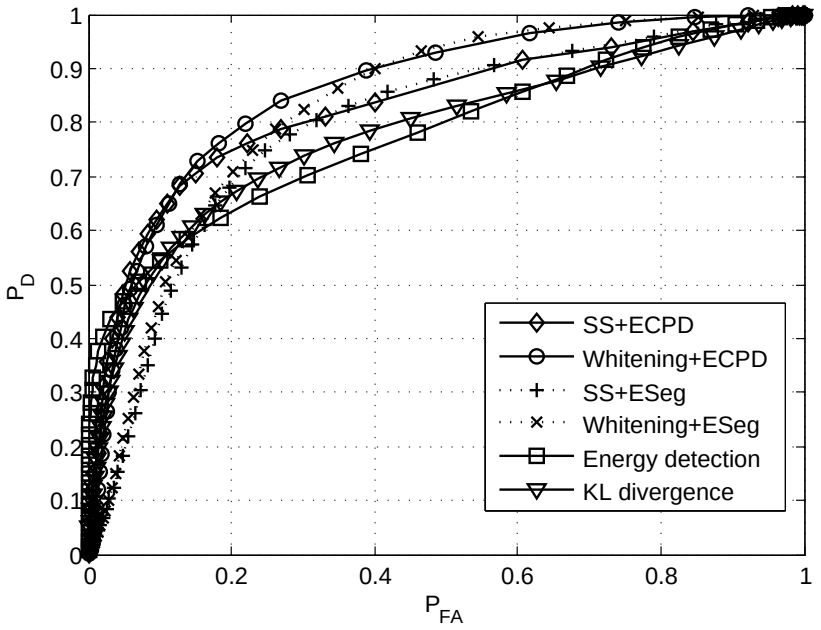


Figure 8.5: ROC curves of the entropy-based CPD segmentation (SS+ECPD, Whitening+EPD), the entropy-based segmentation without CPD (SS+ESeg, Whitening+ESeg), energy detection and KL divergence segmentation.

calls. Let $L(\cdot)$ be the length of a given time interval, and denote I_m and I_a as the bird phrase intervals (including the class "other") labeled by human and by the proposed method, respectively. Also, let I_m^C be those time intervals without any human labels. Define the detection rate P_D and the false alarm rate P_{FA} as

$$P_D = \frac{L(I_m \cap I_a)}{L(I_m)} \quad \text{and} \quad P_{FA} = \frac{L(I_m^C \cap I_a)}{L(I_m^C)}. \tag{8.20}$$

The intersection here means the overlap between two intervals. Based on (8.20), the receiver operating characteristic (ROC curve)[1] of the proposed method is plotted in Fig. 8.5. The "SS+ECPD" and "Whitening+ECPD" represent the entropy-based CPD segmentation with spectral subtraction and spectral whitening filter as the front-end processing, respectively. Using whitening filter as the front-end detects calls better than using spectral subtraction. This is because spectral

[1]Note that to obtain each point in the ROC curve of every method, the threshold is fixed at all time without adaptively changing, while the thresholds used to obtain different operating points are different.

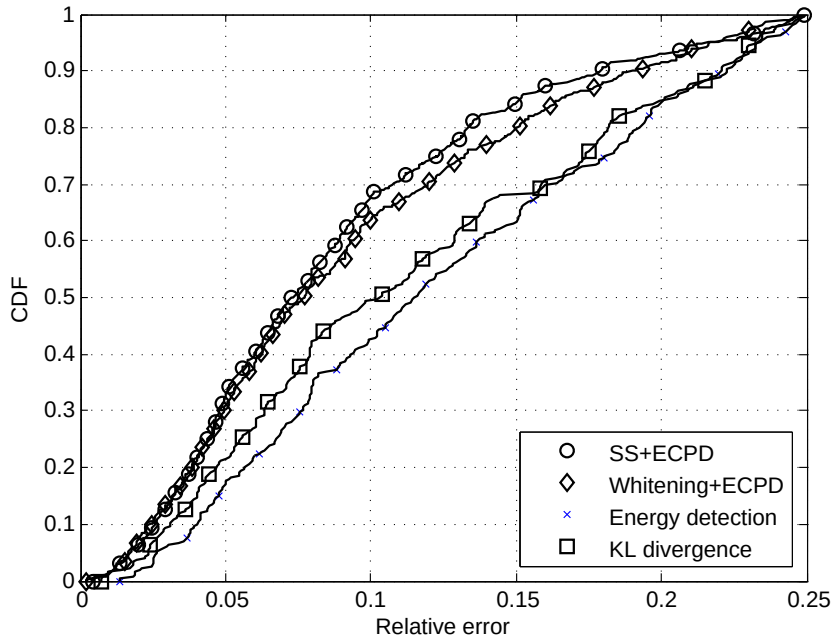


Figure 8.6: The cumulative distribution of relative error of the "matched" phrases by different automated segmentation methods.

subtracting is a spectral cleaning technique and it tends to remove those relatively low-energy background noise. Some weak bird calls may also being removed from the spectrogram. In order to verify the advantage of using CPD over using a hard threshold to detect the change of the entropy sequence, the results without CPD are also shown as "SS+ESeg" and "Whitening+ESeg." It is clear that at the fixed P_{FA} (below 0.4 for the whitening filter front-end and below 0.25 for the spectral subtraction front-end), the detection rate of using CPD is significantly higher than using a hard threshold. For high false alarm rate and high detection rate region which is usually not the desired operating region, there is almost no difference between using CPD or a hard threshold. This shows that by using CPD the system is able to detect more bird calls. The time-domain energy detection and the KL divergence method [77] are also shown for comparison. Both entropy-based CPD segmentation results outperform these two methods in every region of the ROC curves.

	SS+ECPD	Whitening +ECPD	Energy detection	KL divergence
Matched phrases count (rate)	641 (75.23%)	597 (70.07%)	268 (31.46%)	317 (37.21%)
Average RE	8.82%	9.28%	12.29%	11.41%

Table 8.1: The number of detected phrases of Cassin's Vireo and the average relative errors.

In the following experiments, we focus on the phrases of Cassin's Vireo only. Take the 852 human-labeled phrases as the ground true, the relative error (RE) of a segmented phrase can be defined as

$$\mathrm{RE} = \frac{|t_{\mathrm{start,seg}} - t_{\mathrm{start,true}}| + |t_{\mathrm{end,seg}} - t_{\mathrm{end,true}}|}{t_{\mathrm{end,true}} - t_{\mathrm{start,true}}}, \tag{8.21}$$

where $t_{\mathrm{start,seg}}$ and $t_{\mathrm{end,seg}}$ are the start and end labels by the automated segmentation methods, and $t_{\mathrm{start,true}}$ and $t_{\mathrm{end,true}}$ are the start and end time of the human-labeled phrase that is closest to the segmented phrase. Not every segmented phrase matches to the true phrase of Cassin's Vireo, so we define that a segmented phrase is "matched" to a true phrase if the relative error is less than 25%. For those matched phrases, the accuracy of the start and end time labeled by the proposed method are evaluated. The number of matched phrases and the average RE are shown in Table 8.1. The cumulative distribution of RE is also shown in Fig. 8.6. The proposed methods not only have over twice of the matched phrases than the energy detection and the KL divergence segmentation, the average REs among those matched phrases are also lower. Since in energy detection and KL divergence detection, they are more sensitive to the instantaneous energy variation so they sometimes capture the bird vocalization in the syllable level not in the phrase level. Additionally, the proposed method is more robust to the border effect [74, 117, 118], which usually causes troubles for the other two methods on determine the start and end time. The ECPD with spectral whitening has a better

accuracy than the one with spectral subtraction. This may because the low-energy noise around the beginning and the end of a phrase in the spectrogram has been removed or mitigated by spectral subtraction, so the segmentation time labels are more accurate.

The segmented phrases by the proposed method are also tested on the bird phrase classifier. The sparse representation-based (SR) classifier [125] and the support vector machine (SVM) classifer [126] are considered in the experiment. SR classifier performs phrase classification through representing the test feature vector by a sparse linear combination of feature vectors in the training set. The sparse linear combination can be found by solving for a sparse vector via the l_1 minimization convex optimization problem. The advantage of SR classifier is that it only requires a small training set. In our experiment, we choose 7 training tokens per phrase. The training set and the testing set are all chosen from the matched phrases generated by the proposed method. Since not every phrase class has enough tokens for training and testing, 27 out of 65 classes are considered in the classification experiment. The 7 testing tokens for each phrase class are randomly chosen from the matched phrases set, while all the remaining tokens are used for testing. The dimension of the feature vector is set to 128. The multi-class SVM classifier is implemented using the LibSVM [127], which uses a one-against-one decomposition strategy. The selected kernel function is the Gaussian radial basis function (RBF). The classifier for each training set is trained using a five-fold cross-validation to search for an optimal pair of regularization factor and the RBF parameter.

In each classifier, two different training scenarios were considered. The first training set is chosen from the phrases segmented by the proposed ECPD method, and the other training set is chosen from the human-labeled phrases. The testing set is always selected from the phrases generated by ECPD. The classification results are listed in Table 8.2. As expected, the classification rates of the experiment

	Trained by ECPD		Trained by HA	
Testing set	SR	SVM	SR	SVM
SS+ECPD	84.76%	79.74%	80.67%	79.93%
Whitening+ECPD	80.23%	77.91%	77.33%	75.58%

Table 8.2: Phrase classification rates of the sparse representation-based (SR) and the support vector machine (SVM) classifiers training by ECPD and human-annotated phrases (HA)

using ECPD training set is generally higher than the ones using human-annotated training set, since there are less mismatches between and training and testing data. However, the differences between the results of these two scenarios are no more than 5%, which implies that the ECPD phrases stay fairly close to the human annotated phrases. The rates of "SS+ECPD" are higher than "Whitening+ECPD" in all cases by 1.8∼4.5%. This matches to the results in Table 8.1 where the "SS+ECPD" has the lowest relative errors. The classification rate in Table 8.2 is up to 84.76% which shows that the combining the proposed method with phrase classifiers is promising in providing a automated solution to segment and classify phrases reliably.

CHAPTER 9

Conclusion

In this dissertation, we covered wide range of issues in the wireless communication systems. We also applied our solution used on wireless fading channel to biological applications and obtained numerous satisfying results. In the following sections, we conclude our work and summarize the contributions of this dissertation.

9.1 Summary and Contribution

First, we advocate the use of systolic array in MIMO receiver in order to achieve higher processing speed. Systolic array is a network of processing elements, working in parallel, communicating with only neighboring elements, and repeating the same operations. Due to its parallel processing behavior, the overall processing time can be greatly reduced and higher throughput is achieved. In Chapters 3 to 5, we described a systolic array performing LLL-based lattice-reduction-aided detection for MIMO receivers. Lattice reduction and the ensuing linear detection or successive spatial-interference cancellation can be executed by the same array, with minimum global access to each processing element. The classical LLL algorithm was not originally designed for parallel processing. Therefore, we introduced two variants algorithms to facilitate the design of systolic array. The proposed systolic array with external logic controller can work with two different lattice-reduction algorithms. One is the LLL algorithm with full size reduction, in which we alter the way of size reduction and make it more suitable for parallel processing. The second one is an all-swap complex lattice-reduction algorithm, which not only uses

full size reduction, but swaps all columns within single iteration. Compared to the FSR-LLL, the ASLR operates on a whole matrix, rather than on its single columns, during the column-swap and Givens-rotation steps. To reduce the complexity of data communications between processing elements in the systolic array and the complexity of hardware, we replace the Lovász' condition in the LLL algorithm by the Siegel's condition. Even though the Siegel's condition is a relaxation of the Lovász' condition, the BER performance of LR-aided linear detections based on our two algorithm versions appears to be as good as using the conventional LLL, and the computational complexity is reduced by the relaxation. Based on the BER performance and time-efficiency comparisons, the ASLR should be preferred to the FSR-LLL, especially for an MIMO system with a large number of antennas. In FPGA implementation, we limit the μ value in order to simplify the hardware design. The LRAD performance and the orthogonality defect of the reduced channel matrix is still maintained under this approximation. The FPGA emulation results show that our proposed systolic architecture for lattice reduction algorithms run about 1.6$\times$ faster than the conventional LLL, at the cost of moderate increases of hardware complexity. Additionally, due to the high- throughput property of systolic arrays, our design appears very promising for high-data-rate systems, such as in a MIMO-OFDM system.

Second, we proposed a precise way to characterize the abrupt temporal variations in the wireless fading channel. The fading channel is usually stationary only for a short period of time. In order to relate the variation of the channel statistics with the changes in the transmission environment, the channel model and the corresponding statistics for each short stationary interval need to be obtained. Conventionally, the channel measurement were partitioned into short intervals of a the same length. The channel model and statistics were estimated from those intervals. However, the variation in the environment, such as moving pedestrians, street traffic and even when users alter their ways of holding the hand-held devices,

are random, so that a better approach to determine the true location of changes is needed. We proposed using a Bayesian change point detection method to detect the statistical changes in the fading channel gain and envelope. In the CPD algorithm, the key part is the recursive computation for the joint probability of the run length and the observations, which requires the models for run length transition probability and the data predictive probability. The data predictive probability is obtained using Bayesian inference and the computational complexity can be greatly reduced by using conjugate priors of the postulated data model. The common fading channel models are mostly exponential families, for which the conjugate priors always exist. In Chapter 6, we derive the conjugate priors, the updating equations of the hyperparameter sets of the conjugate priors, and the predictive probabilities for the common fading channel envelope models, Rayleigh, Nakagami-m, Weibull and lognormal, and the fading channel gain models, Gaussian and multivariate Gaussian distributions. Another main contribution of this work is performing parameter estimation along with change point detection. Since the posterior probability of the channel model parameters are obtained during the computation of the predictive probability, the MAP and MMSE estimators for the channel parameters are derived accordingly. In Chapter 7, the change point detection based on the derived predictive probabilities and the accuracy of the parameter estimators are first verified using the computer-generated random sequences of different channel models. We also tested the proposed method using the 802.11n channel simulator and the channel measurement conducted in an indoor lab environment. The results show that the change point detection based on Nakagami-m distribution provides the most general solution as it covers wide range of fading conditions including LOS and NLOS scenarios. Based on the detected change points, the empirical distributions of different segments are significantly different, and by the results of Komolgorov-Smirnov goodness-of-fit test shows the accuracy of the estimated parameters based on the proposed methods.

Last but not least, we contribute the usage of the change point method on bird phrase segmentation and propose a bird phrase segmentation method by entropy-based change point detection. Since the spectrogram of the bird vocalization is usually sparse, the entropy calculated from sliding time-frequency block of the spectrogram is relatively low when there is a bird call in the block and is relatively high when there is no call in the block. To enlarge the difference between the entropy of a call period and the one of a quiet period, a polynomial-based whitening filter is proposed as the front-end of the segmentation to whiten the spectrogram of the background noise. Alternatively, spectral subtraction which reduce the noise interferences can also be applied to the spectrogram prior to calculating the entropy. Change point detection is used to monitor the statistical change in the entropy sequence. The detected change points are the starting and end points of a bird phrase. Comparing to using a hard threshold for detecting the change in the entropy level, change point detection is more robust to the noise interference and the border effect. Experimental results also shows that the segmented phrases by our proposed methods achieved a better similarity to the human-labeled phrases. It is also shown to be practical to combine the proposed segmentation method with sparse representation-based classifier. This automated system would facilitate the analysis of long field recordings.

Appendix A

Derivations of Predictive Probabilities and Updating Equations of the Conjugate Prior Hyperparameter sets for Various Distributions

A.1 One-Dimensional Gaussian Distribution

As shown in (6.6) and (6.7), the parameter set $\boldsymbol{\eta}^{r_t}$ relates to $\boldsymbol{\eta}^{r_{t-1}}$ through $P(x_t|\mu,\sigma^2)$. Since the Gaussian-inverse-Gamma conjugate prior is used for the Gaussian likelihood $P(x_t|\mu,\sigma^2)$, the posterior probability $P(\mu,\sigma^2|m^{r_t},\tau^{r_t},\alpha^{r_t},\beta^{r_t})$, which is proportional to

$$P(x_t|\mu,\sigma^2)P(\mu,\sigma^2|m^{r_{t-1}},\tau^{r_{t-1}},\alpha^{r_{t-1}},\beta^{r_{t-1}}), \tag{A.1}$$

should still be in the Gaussian-inverse-Gamma family. To show this, (A.1) can be written as

$$\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(x_t-\mu)^2}{2\sigma^2}\right\}\right] \times\left[\frac{(\sigma^2)^{-\alpha^{r_{t-1}}-\frac{3}{2}}\left(\beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}}}{\Gamma(\alpha^{r_{t-1}})\sqrt{2\pi\tau^{r_{t-1}}}}\exp\left\{-\frac{(\mu-m^{r_{t-1}})^2+2\tau^{r_{t-1}}\beta^{r_{t-1}}}{2\tau^{r_{t-1}}\sigma^2}\right\}\right], \tag{A.2}$$

where inside the first brackets is the Gaussian form and inside the second brackets is the Gaussian-inverse-Gamma form. It can be rearranged as

$$K \cdot \left[\frac{1}{\sqrt{2\pi \frac{\tau^{r_{t-1}}}{\tau^{r_{t-1}}+1}\sigma^2}} \exp\left\{ -\frac{(\tau^{r_{t-1}}+1)\left(\mu - \frac{\tau^{r_{t-1}}x_t + m^{r_{t-1}}}{\tau^{r_{t-1}}+1}\right)^2}{2\tau^{r_{t-1}}\sigma^2} \right\} \right]$$
$$\times \left[\frac{(\sigma^2)^{-\left(\alpha^{r_{t-1}}+\frac{1}{2}\right)-1}\left[\beta^{r_{t-1}} + \frac{(x_t - m^{r_{t-1}})^2}{2(\tau^{r_{t-1}}+1)}\right]^{\alpha^{r_{t-1}}+\frac{1}{2}}}{\Gamma(\alpha^{r_{t-1}}+\frac{1}{2})} \exp\left\{ -\frac{\left[\beta^{r_{t-1}} + \frac{(x_t - m^{r_{t-1}})^2}{2(\tau^{r_{t-1}}+1)}\right]}{\sigma^2} \right\} \right] \tag{A.3}$$

where

$$K = \sqrt{\frac{1}{2\pi\beta^{r_{t-1}}(\tau^{r_{t-1}}+1)}} \frac{\Gamma(\alpha^{r_{t-1}}+\frac{1}{2})}{\Gamma(\alpha^{r_{t-1}})} \left(1 + \frac{(x_t - m^{r_{t-1}})^2}{2\beta^{r_{t-1}}(\tau^{r_{t-1}}+1)}\right) \tag{A.4}$$

It can be easily seen that (A.3) is

$$K \cdot \mathcal{N}\left(\mu \middle| \frac{\tau^{r_{t-1}}x_t + m^{r_{t-1}}}{\tau^{r_{t-1}}+1}, \frac{\tau^{r_{t-1}}}{\tau^{r_{t-1}}+1}\right) \mathcal{IG}\left(\sigma^2 \middle| \alpha^{r_{t-1}} + \frac{1}{2}, \beta^{r_{t-1}} + \frac{(x_t - m^{r_{t-1}})^2}{2(\tau^{r_{t-1}}+1)}\right), \tag{A.5}$$

which is also a Gaussian-inverse-gamma distribution in the form of (6.20). Hence, from (A.5) we can conclude the updating equations of the parameter set as shown in (6.23) to (6.26). Additionally, the predictive probability $P(x_t|\mathbf{x}^{r_{t-1}})$ is

$$\int_{\mu}\int_{\sigma^2} P(x_t|\mu,\sigma^2)P(\mu,\sigma^2|m^{r_{t-1}},\tau^{r_{t-1}},\alpha^{r_{t-1}},\beta^{r_{t-1}})d\sigma^2 d\mu. \tag{A.6}$$

From (A.5), $P(x_t|\mathbf{x}^{r_{t-1}}) = K$ and K is in the form of a Student-t distribution as shown in (6.21).

A.2 Rayleigh Distribution

The predictive probability calculated from Rayleigh likelihood and its conjugate prior is

$$
\begin{aligned}
&P(h_t|r_{t-1}, \mathbf{h}^{r_{t-1}}) \\
&= \int_{\sigma^2} P(h_t|\sigma^2) P(\sigma^2|\alpha^{r_{t-1}}, \beta^{r_{t-1}}) d\sigma^2 && \text{(A.7)} \\
&= \int_{\sigma^2} \left\{ \frac{h_t}{\sigma^2} \exp\left(-\frac{h_t^2}{2\sigma^2}\right) \right\} \left\{ \frac{(\beta^{r_{t-1}})^{\alpha^{r_{t-1}}}}{\Gamma(\alpha^{r_{t-1}})} (\sigma^2)^{-\alpha^{r_{t-1}}-1} \exp\left(-\frac{\beta^{r_{t-1}}}{\sigma^2}\right) \right\} d\sigma^2 && \text{(A.8)} \\
&= \frac{\Gamma(\alpha^{r_{t-1}}+1)}{\Gamma(\alpha^{r_{t-1}})} \frac{h_t \left(\beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}}}{\left(\frac{h_t^2}{2} + \beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}+1}} \\
&\quad \times \int_{\sigma^2} \frac{\left(\frac{h_t^2}{2} + \beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}+1}}{\Gamma(\alpha^{r_{t-1}}+1)} (\sigma^2)^{-(\alpha^{r_{t-1}}+1)-1} \exp\left(-\frac{\frac{h_t^2}{2} + \beta^{r_{t-1}}}{\sigma^2}\right) d\sigma^2 && \text{(A.9)}
\end{aligned}
$$

The integrand in (A.9) is an inverse-gamma pdf of σ^2 with parameter set $\{\alpha^{r_{t-1}} + 1, h_t^2/2 + \beta^{r_{t-1}}\}$. This inverse-gamma pdf is in fact the posterior probability $P(\sigma^2|\mathbf{h}^{r_t}) = P(\sigma^2|\alpha^{r_t}, \beta^{r_t})$. Therefore, we can conclude that

$$\alpha^{r_t} = \alpha^{r_{t-1}} + 1, \text{ and } \beta^{r_t} = \beta^{r_{t-1}} + \frac{h_t^2}{2}. \tag{A.10}$$

Additionally, the integral in (A.9) becomes 1 since it is integrated over all possible σ^2. Hence, the predictive probability becomes

$$\frac{\Gamma(\alpha^{r_{t-1}}+1)}{\Gamma(\alpha^{r_{t-1}})} \frac{h_t \left(\beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}}}{\left(\frac{h_t^2}{2} + \beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}+1}}. \tag{A.11}$$

Using the property of gamma function, $\Gamma(x+1) = x\Gamma(x)$ for positive x, (A.11) is equal to

$$\frac{h_t \alpha^{r_{t-1}} \left(\beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}}}{\left(\frac{h_t^2}{2} + \beta^{r_{t-1}}\right)^{\alpha^{r_{t-1}}+1}}. \tag{A.12}$$

A.3 Nakagami-m Distribution

The predictive probability calculated from the Nakagami-m likelihood and its conjugate prior is

$$
\begin{aligned}
&P(h_t|r_{t-1}, \mathbf{h}^{r_{t-1}}) \\
&= \int_m \int_\gamma P(h_t|m,\gamma) P(m,\gamma|n^{r_{t-1}}, \nu^{r_{t-1}}, s^{r_{t-1}}, p^{r_{t-1}}) d\gamma dm \qquad \text{(A.13)} \\
&= \int_m \int_\gamma \left\{ \frac{2}{\Gamma(m)} (m\gamma)^m h_t^{(2m-1)} \exp\left\{-m\gamma h_t^2\right\} \right\} \\
&\qquad \times \left\{ \frac{(m\gamma)^{\nu^{r_{t-1}} m}}{K(\boldsymbol{\eta}^{r_{t-1}}) \Gamma^{n^{r_{t-1}}}(m)} \left(p^{r_{t-1}}\right)^{2m-1} \exp\left\{-s^{r_{t-1}} m\gamma\right\} \right\} d\gamma dm, \qquad \text{(A.14)}
\end{aligned}
$$

where K is defined in (6.81) and $\boldsymbol{\eta} = \{n, \nu, s, p\}$. By rearranging the integrand, (A.14) can be written as

$$
\frac{2}{K(\boldsymbol{\eta}^{r_{t-1}})} \int_m \int_\gamma \frac{(m\gamma)^{(\nu^{r_{t-1}}+1)m}}{\Gamma^{n^{r_{t-1}}+1}(m)} \left(h_t p^{r_{t-1}}\right)^{2m-1} \exp\left\{-\left(s^{r_{t-1}} + h_t^2\right) m\gamma\right\} d\gamma dm. \qquad \text{(A.15)}
$$

Compare (A.15) with (A.14), we can obtain the updating equation of the hyperparameter set $\boldsymbol{\eta}^{r_t}$,

$$
n^{r_t} = n^{r_{t-1}} + 1, \quad \nu^{r_t} = \nu^{r_{t-1}} + 1, \qquad \text{(A.16)}
$$

$$
p^{r_t} = p^{r_{t-1}} \cdot h_t, \quad s^{r_t} = s^{r_{t-1}} + h_t^2. \qquad \text{(A.17)}
$$

By the definition of K, it is clear that (A.15) is equal to

$$
\frac{2K(\boldsymbol{\eta}^{r_t})}{K(\boldsymbol{\eta}^{r_{t-1}})}. \qquad \text{(A.18)}
$$

A.4 Weibull Distribution

The predictive probability calculated from Weibull likelihood (with fixed α) and its conjugate prior is

$$\begin{aligned} & P(h_t|r_{t-1}, \mathbf{h}^{r_{t-1}}) \\ &= \int_\beta P(h_t|\beta) P(\beta|p^{r_{t-1}}, q^{r_{t-1}}) d\beta \qquad \text{(A.19)} \\ &= \int_\beta \left\{ \frac{\alpha}{\beta} h_t^{\alpha-1} e^{-\frac{h_t^\alpha}{\beta}} \right\} \left\{ \frac{(q^{r_{t-1}})^{p^{r_{t-1}}}}{\Gamma(p^{r_{t-1}})} \beta^{-p^{r_{t-1}}-1} e^{-\frac{q^{r_{t-1}}}{\beta}} \right\} d\beta \qquad \text{(A.20)} \end{aligned}$$

Rearranging the integrand in (A.20), we can form an inverse-gamma pdf multiplied by a scaling factor,

$$\begin{aligned} & \frac{\Gamma(p^{r_{t-1}}+1)}{\Gamma(p^{r_{t-1}})} \frac{\alpha h_t^{\alpha-1} (q^{r_{t-1}})^{p^{r_{t-1}}}}{(h_t^\alpha + q^{r_{t-1}})^{p^{r_{t-1}}+1}} \times \\ & \qquad \int_\beta \frac{(h_t^\alpha + q^{r_{t-1}})^{p^{r_{t-1}}+1}}{\Gamma(p^{r_{t-1}}+1)} \beta^{-(p^{r_{t-1}}+1)-1} \exp\left(-\frac{h_t^\alpha + q^{r_{t-1}}}{\beta} \right) d\beta \qquad \text{(A.21)} \end{aligned}$$

The integrand in (A.21) is an inverse-gamma pdf of β with parameter set $\{p^{r_{t-1}} + 1, h_t^\alpha + q^{r_{t-1}}\}$. Therefore, we can conclude that the updating equations of $\{p^{r_t}, q^{r_t}\}$ are

$$p^{r_t} = p^{r_{t-1}} + 1, \text{ and } q^{r_t} = h_t^\alpha + q^{r_{t-1}}. \qquad \text{(A.22)}$$

Additionally, (A.21) can be simplified to

$$\frac{\alpha h_t^{\alpha-1} p^{r_{t-1}} (q^{r_{t-1}})^{p^{r_{t-1}}}}{(h_t^\alpha + q^{r_{t-1}})^{p^{r_{t-1}}+1}}, \qquad \text{(A.23)}$$

since integrating the inverse-gamma distribution over all possible β is equal to 1 and $\Gamma(x+1) = x\Gamma(x)$ for positive x.

References

[1] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.

[2] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO Wireless Communications*. New York, NY, USA: Cambridge University Press, 2007.

[3] Z. Guo and P. Nilsson, "A VLSI implementation of MIMO detection for future wireless communications," in *Proc. IEEE Personal, Indoor and Mobile Radio Communications*, vol. 3, 2003, pp. 2852–2856.

[4] M. Myllyla, J. Hintikka, J. Cavallaro, M. Juntti, M. Limingoja, and A. Byman, "Complexity analysis of MMSE detector architectures for MIMO OFDM systems," in *Proc. the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, 2005, pp. 75–81.

[5] M. Karkooti, J. Cavallaro, and C. Dick, "FPGA implementation of matrix inversion using QRD-RLS algorithm," in *Proc. Asilomar Conference on Signals, Systems and Computers*, 2005, pp. 1625–1629.

[6] H. Yao and G. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *IEEE Global Telecommunications Conference, GLOBECOM*, vol. 1, 2002, pp. 424–428.

[7] D. Seethaler, G. Matz, and F. Hlawatsch, "Low-complexity MIMO data detection using seysen's lattice reduction algorithm," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, vol. 3, 2007, pp. III–53–III–56.

[8] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice reduction," in *Proc. IEEE International Conference on Communications*, vol. 2, 2004, pp. 798–802.

[9] D. Wübben, D. Seethaler, J. Jaldén, and G. Matz, "Lattice reduction," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 70 –91, May 2011.

[10] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.

[11] P. Nguyen and B. Vallee, *The LLL Algorithm Survey and Applications*. Heidelberg: Springer, 2010.

[12] Y. H. Gan, C. Ling, and W. H. Mow, "Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection," *IEEE Trans. on Signal Processing*, vol. 57, no. 7, pp. 2701–2710, 2009.

[13] X. Ma and W. Zhang, "Performance analysis for MIMO systems with lattice-reduction aided linear equalization," *IEEE Trans. on Communications*, vol. 56, no. 2, pp. 309–318, 2008.

[14] M. Taherzadeh, A. Mobasher, and A. Khandani, "LLL reduction achieves the receive diversity in MIMO decoding," *IEEE Trans. on Inform. Theory*, vol. 53, no. 12, pp. 4801–4805, 2007.

[15] J. Jaldén, D. Seethaler, and G. Matz, "Worst- and average-case complexity of LLL lattice reduction in MIMO wireless systems," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2008, pp. 2685–2688.

[16] B. Gestner, W. Zhang, X. Ma, and D. Anderson, "VLSI implementation of a lattice reduction algorithm for low-complexity equalization," in *Proc. IEEE International Conference on Circuits and Systems for Communications, ICCSC*, 2008, pp. 643–647.

[17] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical Programming*, vol. 66, no. 1-3, pp. 181–199, 1994.

[18] J. Jaldén and P. Elia, "DMT optimality of LR-aided linear decoders for a general class of channels, lattice designs, and system models," *IEEE Trans. on Information Theory*, vol. 56, no. 10, pp. 4765–4780, 2010.

[19] H. Vetter, V. Ponnampalam, M. Sandell, and P. Hoeher, "Fixed complexity LLL algorithm," *IEEE Trans. on Signal Processing*, vol. 57, no. 4, pp. 1634–1637, 2009.

[20] L. Luzzi, G. Othman, and J. Belfiore, "Augmented lattice reduction for MIMO decoding," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2853–2859, 2010.

[21] C. Ling, S. Liu, L. Luzzi, and D. Stehl, "Decoding by embedding: Correct decoding radius and DMT optimality," in *Proceedings of the IEEE International Symposium on Information Theory*, Saint-Petersburg, Russie, Fdration De, Jul. 2011, pp. 1106–1110.

[22] C. Ling, "On the proximity factors of lattice reduction-aided decoding," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2795–2808, 2011.

[23] A. Burg, D. Seethaler, and G. Matz, "VLSI implementation of a lattice-reduction algorithm for multi-antenna broadcast precoding," in *IEEE International Symposium on Circuits and Systems, ISCAS*, 2007, pp. 673–676.

[24] L. Barbero, D. Milliner, T. Ratnarajah, J. Barry, and C. Cowan, "Rapid prototyping of Clarkson's lattice reduction for MIMO detection," in *IEEE International Conference on Communications, ICC*, 2009, pp. 1–5.

[25] L. Bruderer, C. Studer, M. Wenk, D. Seethaler, and A. Burg, "VLSI implementation of a low-complexity LLL lattice reduction algorithm for MIMO detection," in *IEEE International Symposium on IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 3745–3748.

[26] A. Youssef, M. Shabany, and P. Gulak, "VLSI implementation of a hardware-optimized lattice reduction algorithm for WiMAX/LTE MIMO detection," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 3541–3544.

[27] H. Kung and C. Leiserson, "Algorithms for VLSI processor arrays," *Introduction to VLSI systems*, pp. 271–292, 1980.

[28] S. Y. Kung, "VLSI array processors," *IEEE ASSP Magazine*, vol. 2, no. 3, pp. 4–22, 1985.

[29] W. M. Gentleman and H. T. Kung, "Matrix triangulation by systolic arrays," in *Proc. of SPIE: Real-time Signal Processing IV*, vol. 298, 1981, pp. 19–26.

[30] A. El-Amawy and K. Dharmarajan, "Parallel VLSI algorithm for stable inversion of dense matrices," *IEE Proc. Computers and Digital Techniques*, vol. 136, no. 6, pp. 575–580, 1989.

[31] C. Rader, "VLSI systolic arrays for adaptive nulling," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 29–49, 1996.

[32] K. Liu, S.-F. Hsieh, K. Yao, and C.-T. Chiu, "Dynamic range, stability, and fault-tolerant capability of finite-precision RLS systolic array based on givens rotations," *IEEE Trans. on Circuits and Systems*, vol. 38, no. 6, pp. 625–636, 1991.

[33] D. Boppana, K. Dhanoa, and J. Kempa, "FPGA based embedded processing architecture for the QRD-RLS algorithm," in *Proc. IEEE Symposium on Field-Programmable Custom Computing Machines*, vol. 0, 2004, pp. 330–331.

[34] K. Yao and F. Lorenzelli, "Systolic algorithms and architectures for high-throughput processing applications," *Journal of Signal Processing Systems*, vol. 53, no. 1-2, pp. 15–34, 2008.

[35] J. Wang and B. Daneshrad, "A universal systolic array for linear MIMO detections," in *Proc. IEEE Wireless Communications and Networking Conference*, 2008, pp. 147–152.

[36] K. Seki, T. Kobori, J. Okello, and M. Ikekawa, "A CORDIC-Based reconfigrable systolic array processor for MIMO-OFDM wireless communications," in *IEEE Workshop on Signal Processing Systems*, 2007, pp. 639–644.

[37] Y. Hu, "CORDIC-based VLSI architectures for digital signal processing," *IEEE Signal Processing Magazine*, vol. 9, no. 3, pp. 16–35, 1992.

[38] B. Cerato, G. Masera, and P. Nilsson, "Hardware architecture for matrix factorization in mimo receivers," in *Proc. ACM Great Lakes symposium on VLSI*, New York, NY, USA, 2007, pp. 196–199.

[39] C. Heckler and L. Thiele, "A parallel lattice basis reduction for mesh-connected processor arrays and parallel complexity," in *Proc. IEEE Symposium on Parallel and Distributed Processing*, 1993, pp. 400–407.

[40] J. W. S. Cassels, *Rational quadratic forms.* London; New York: Academic Press, 1978.

[41] H. Hashemi, "The indoor radio propagation channel," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943–968, Jul. 1993.

[42] E. Biglieri, *Coding for Wireless Channels.* Springer, May 2005.

[43] U. Charash, "Reception through Nakagami fading multipath channels with random delays," *IEEE Transactions on Communications*, vol. 27, no. 4, pp. 657–670, Apr. 1979.

[44] J. Cavers, "Optimized use of diversity modes in transmitter diversity systems," in *Vehicular Technology Conference, 1999 IEEE 49th*, vol. 3, Jul. 1999, pp. 1768–1773.

[45] Y.-C. Ko and M.-S. Alouini, "Estimation of Nakagami-m fading channel parameters with application to optimized transmitter diversity systems," *IEEE Transactions on Wireless Communications*, vol. 2, no. 2, pp. 250–259, Mar. 2003.

[46] S. Catreux, V. Erceg, D. Gesbert, and J. Heath, R.W., "Adaptive modulation and MIMO coding for broadband wireless data networks," *IEEE Communications Magazine*, vol. 40, no. 6, pp. 108 –115, Jun. 2002.

[47] J. Cheng, C. Tellambura, and N. Beaulieu, "Performance of digital linear modulations on Weibull slow-fading channels," *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1265–1268, Aug. 2004.

[48] V. A. Aalo, T. Piboongungon, and G. P. Efthymoglou, "Another look at the performance of MRC schemes in Nakagami-m fading channels with arbitrary parameters," *IEEE Transactions on Communications*, vol. 53, no. 12, pp. 2002–2005, Dec. 2005.

[49] M. Ismail and M. Matalgah, "Performance of dual maximal ratio combining diversity in nonidentical correlated Weibull fading channels using pade acute; approximation," *IEEE Transactions on Communications*, vol. 54, no. 3, pp. 397–402, Mar. 2006.

[50] K. A. Saaifan and E. K. Al-Hussaini, "Performance of MIMO system through Nakagami fading channel with arbitrary fading parameters," *Wireless Personal Communications*, vol. 50, no. 3, pp. 367–380, Oct. 2008.

[51] A. Maaref and S. Aissa, "Exact error probability analysis of rectangular QAM for single- and multichannel reception in Nakagami-m fading channels," *IEEE Transactions on Communications*, vol. 57, no. 1, pp. 214–221, Jan. 2009.

[52] J. Cheng and N. Beaulieu, "Maximum-likelihood based estimation of the Nakagami-m parameter," *IEEE Communications Letters*, vol. 5, no. 3, pp. 101–103, Mar. 2001.

[53] J. Gaeddert and A. Annamalai, "Further results on Nakagami-m parameter estimation," *IEEE Communications Letters*, vol. 9, no. 1, pp. 22–24, Jan. 2005.

[54] R. Ross, "Bias and standard deviation due to weibull parameter estimation for small data sets," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 3, no. 1, pp. 28–42, Feb. 1996.

[55] H. Hashemi, M. McGuire, T. Vlasschaert, and D. Tholl, "Measurements and modeling of temporal variations of the indoor radio propagation channel," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 733–737, Aug. 1994.

[56] P. Hafezi, A. Nix, and M. Beach, "An experimental investigation of the impact of human shadowing on temporal variation of broadband indoor radio channel characteristics and system performance," in *Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd*, vol. 1, 2000, pp. 37–42.

[57] P. Pagani and P. Pajusco, "Experimental assessment of the UWB channel variability in a dynamic indoor environment," in *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004*, vol. 4, Sep. 2004, pp. 2973–2977.

[58] L. Ahumada, R. Feick, R. Valenzuela, and C. Morales, "Measurement and characterization of the temporal behavior of fixed wireless links," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 6, pp. 1913–1922, Nov. 2005.

[59] L. Ahumada, R. Feick, and R. Valenzuela, "Characterization of temporal fading in urban fixed wireless links," *IEEE Communications Letters*, vol. 10, no. 4, pp. 242–244, Apr. 2006.

[60] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. Van Herwegen, and W. Vantomme, "The industrial indoor channel: large-scale and temporal fading at 900, 2400, and 5200 MHz," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2740–2751, Jul. 2008.

[61] L. Greenstein, S. Ghassemzadeh, V. Erceg, and D. Michelson, "Ricean k-factors in narrow-band fixed wireless channels: Theory, experiments, and statistical models," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4000–4012, Oct. 2009.

[62] R. P. Adams and D. J. C. MacKay, "Bayesian online changepoint detection," *Technical Report, University of Cambridge*, Oct. 2007. [Online]. Available: http://arxiv.org/abs/0710.3742

[63] A. Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim, "A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods," vol. 54, no. 9, pp. 3372–3382, 2006.

[64] Y. Chen, K. Hwang, and W.-S. Ku, "Collaborative detection of DDoS attacks over multiple network domains," vol. 18, no. 12, pp. 1649–1662, 2007.

[65] M. A. Osborne, R. Garnett, and S. J. Roberts, "Active data selection for sensor networks with faults and changepoints," in *International Conference on Advanced Information Networking and Applications*. Los Alamitos, CA, USA: IEEE Computer Society, 2010, pp. 533–540.

[66] M. Staudacher, S. Telser, A. Amann, H. Hinterhuber, and M. Ritsch-Marte, "A new method for change-point detection developed for on-line analysis of the heart beat variability during sleep," *Physica A: Statistical Mechanics and its Applications*, vol. 349, no. 3, pp. 582–596, Apr. 2005.

[67] V. Spokoiny, "Multiscale local change point detection with applications to value-at-risk," *The Annals of Statistics*, vol. 37, no. 3, pp. 1405–1436, Jun. 2009.

[68] D. J. Mennill, "Individual distinctiveness in avian vocalizations and the spatial monitoring of behaviour," *Ibis*, vol. 153, no. 2, pp. 235–238, 2011.

[69] B. C. Pijanowski, L. J. Villanueva-Rivera, S. L. Dumyahn, A. Farina, B. L. Krause, B. M. Napoletano, S. H. Gage, and N. Pieretti, "Soundscape ecology: The science of sound in the landscape," *BioScience*, vol. 61, no. 3, pp. 203–216, Mar. 2011.

[70] V. M. Trifa, A. N. G. Kirschel, C. E. Taylor, and E. E. Vallejo, "Automated species recognition of antbirds in a Mexican rainforest using hidden markov models," *The Journal of the Acoustical Society of America*, vol. 123, no. 4, pp. 2424–2431, 2008.

[71] A. Harma, "Automatic identification of bird species based on sinusoidal modeling of syllables," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*, vol. 5, Apr. 2003, pp. V545–V548.

[72] C.-H. Lee, C.-C. Han, and C.-C. Chuang, "Automatic classification of bird species from their sounds using two-dimensional cepstral coefficients," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1541–1550, Nov. 2008.

[73] P. Somervuo and A. Harma, "Bird song recognition based on syllable pair histograms," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04)*, vol. 5, May 2004, pp. V825–V828.

[74] S. Fagerlund, "Bird species recognition using support vector machines," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, Jan. 2007.

[75] A. Selin, J. Turunen, and J. T. Tanttu, "Wavelets in recognition of bird sounds," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, Jan. 2007.

[76] P. Somervuo, A. Harma, and S. Fagerlund, "Parametric representations of bird sounds for automatic species recognition," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 6, pp. 2252–2263, Nov. 2006.

[77] B. Lakshminarayanan, R. Raich, and X. Fern, "A syllable-level probabilistic framework for bird species identification," in *International Conference on Machine Learning and Applications, 2009*, Dec. 2009, pp. 53–59.

[78] L. Neal, F. Briggs, R. Raich, and X. Fern, "Time-frequency segmentation of bird song in noisy acoustic environments," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 2012–2015.

[79] C. Elemans, K. Heeck, and M. Muller, "Spectrogram analysis of animal sound production," *Bioacoustics : the International Journal of Animal Sound and its Recording*, vol. 18, no. 2, pp. 183–212, 2008.

[80] C. K. Catchpole and P. J. B. Slater, *Bird Song: Biological Themes and Variations*. Cambridge University Press, Oct. 2003.

[81] B. Hassibi, "An efficient square-root algorithm for BLAST," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2000, pp. II737–II740.

[82] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. on Inform. Theory*, vol. 48, no. 8, pp. 2201–2214, 2002.

[83] P. Wolniansky, G. Foschini, G. Golden, and R. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," in *URSI International Symposium on Signals, Systems, and Electronics, ISSSE*, 1998, pp. 295–300.

[84] R. Böhnke, D. Wübben, V. Kühn, and K.-D. Kammeyer, "Reduced complexity MMSE detection for BLAST architectures," in *IEEE Global Telecommunications Conference, GLOBECOM*, vol. 4, 2003, pp. 2258–2262 vol.4.

[85] L. Babai, "On lovász' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[86] R. Döhler, "Squared givens rotation," *IMA Journal of Numerical Analysis*, vol. 11, no. 1, pp. 1 –5, Jan. 1991.

[87] P. Luethi, A. Burg, S. Haene, D. Perels, N. Felber, and W. Fichtner, "VLSI implementation of a high-speed iterative sorted MMSE QR decomposition," in *Proc. IEEE International Symposium on Circuits and Systems*, 2007, pp. 1421–1424.

[88] F. Lorenzelli, P. Hansen, T. Chan, and K. Yao, "A systolic implementation of the Chan/Foster RRQR algorithm," *IEEE Trans. on Signal Processing*, vol. 42, no. 8, pp. 2205–2208, 1994.

[89] A. Wenzler and E. Luder, "New structures for complex multipliers and their noise analysis," in *, 1995 IEEE International Symposium on Circuits and Systems, 1995. ISCAS '95*, vol. 2, May 1995, pp. 1432–1435.

[90] C. V. Ramamoorthy, J. R. Goodman, and K. H. Kim, "Some properties of iterative square-rooting methods using high-speed multiplication," *IEEE Trans. on Computers*, vol. C-21, no. 8, pp. 837–847, 1972.

[91] D. M. Markovic, "A power/area optimal approach to VLSI signal processing," Ph.D. dissertation, EECS Department, University of California, University of California at Berkeley, Berkeley, California, May 2006.

[92] M. Schulte and K. Wires, "High-speed inverse square roots," in *14th IEEE Symposium on Computer Arithmetic, 1999. Proceedings*, 1999, pp. 124–131.

[93] N. Takagi, "Powering by a table look-up and a multiplication with operand modification," *IEEE Transactions on Computers*, vol. 47, no. 11, pp. 1216–1222, Nov. 1998.

[94] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, Sep. 1959.

[95] S.-F. Hsiao and J.-M. Delosme, "Householder CORDIC algorithms," *IEEE Transactions on Computers*, vol. 44, no. 8, pp. 990–1001, Aug. 1995.

[96] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis, Second Edition*. Chapman and Hall/CRC, Jul. 2003.

[97] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall PTR, 1993.

[98] C. Bishop, *Pattern recognition and machine learning*. Springer, Oct. 2006.

[99] H. Lütkepohl, *Handbook of Matrices*. John Wiley & Sons, Jan. 1997.

[100] R. Vaughan, J. Bach-Anderson, and J. B. Andersen, *Channels, Propagation and Antennas for Mobile Communications*. IET, 2003.

[101] M. Nakagami, "The m-distribution-a general formula of intensity distribution of rapid fading," *Statistical Method of Radio Propagation*, 1960.

[102] L. Tierney and J. B. Kadane, "Accurate approximations for posterior moments and marginal densities," *Journal of the American Statistical Association*, vol. 81, no. 393, p. 82, Mar. 1986.

[103] J. M. Bernardo, "Algorithm AS 103: Psi (Digamma) function," *Applied Statistics*, vol. 25, no. 3, p. 315, 1976.

[104] W. Weibull, "A statistical distribution function of wide applicability," *Journal of Applied Mechanics*, vol. 18, pp. 293–297, 1951.

[105] N. Shepherd, "Radio wave loss deviation and shadow loss at 900 MHz," in *26th IEEE Vehicular Technology Conference, 1976*, vol. 26, Mar. 1976, pp. 63–66.

[106] G. Tzeremes and C. Christodoulou, "Use of Weibull distribution for describing outdoor multipath fading," in *IEEE Antennas and Propagation Society International Symposium, 2002*, vol. 1, 2002, pp. 232–235.

[107] U. Schuster and H. Bolcskei, "Ultrawideband channel modeling on the basis of information-theoretic criteria," *IEEE Transactions on Wireless Communications*, vol. 6, no. 7, pp. 2464–2475, Jul. 2007.

[108] M. Ismail and M. Matalgah, "BER analysis of diversity M-PSK receivers over the Weibull fading channel with cochannel interference," in *IEEE International Conference on Communications, 2006. ICC '06*, vol. 11, Jun. 2006, pp. 5134–5139.

[109] D. Fink, *A Compendium of Conjugate Priors*, 1997.

[110] M. Zonoozi and P. Dassanayake, "Shadow fading in mobile radio channel," in *, Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96*, vol. 2, Oct. 1996, pp. 291–295.

[111] J. Foerster, "Channel modeling sub-committee report final," 2002.

[112] A. Molisch, J. Foerster, and M. Pendergrass, "Channel models for ultrawideband personal area networks," *IEEE Wireless Communications*, vol. 10, no. 6, pp. 14–21, Dec. 2003.

[113] V. Erceg, L. Schumacher, and P. Kyritsi, et al., "IEEE 802.11 document 03/940r4 TGn channel models," May 2004.

[114] J. Kermoal, L. Schumacher, K. Pedersen, P. Mogensen, and F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 6, pp. 1211–1226, Aug. 2002.

[115] "MATLAB implementation of the IEEE 802.11 HTSG channel model special committee proposal." [Online]. Available: http://www.info.fundp.ac.be/~lsc/Research/IEEE_80211_HTSG_CMSC/distribution_terms.html

[116] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press, 2004.

[117] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee, "Classification of general audio data for content-based retrieval," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 533–544, Apr. 2001.

[118] M. Spina and V. Zue, "Automatic transcription of general audio data: preliminary analyses," in *, Fourth International Conference on Spoken Language, 1996. ICSLP 96. Proceedings*, vol. 2, Oct. 1996, pp. 594 –597 vol.2.

[119] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," vol. 27, no. 2, pp. 113–120, Apr. 1979.

[120] N. Virag, "Single channel speech enhancement based on masking properties of the human auditory system," vol. 7, no. 2, pp. 126–137, Mar. 1999.

[121] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, May 2002, pp. IV–4164–IV–4164.

[122] P. Boersma and D. Weenink, "Praat: doing phonetics by computer (Version 5.1.05)," May 2009. [Online]. Available: http://www.praat.org/

[123] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American Statistical Association*, vol. 53, no. 282, pp. 457–481, Jun. 1958.

[124] W. L. Martnez and A. R. Martnez, *Computational statistics handbook with MATLAB*. CRC Press, 2002.

[125] L. N. Tan, K. Kaewtip, M. L. Cody, C. E. Taylor, and A. Alwan, "Evaluation of a sparse representation-based classifier for bird phrase classification under limited data conditions," in *Proc. of 13th Annual Conference of the Iternational Speech Communication Association (IterSpeech 2012)*, 2012.

[126] M. A. Acevedo, C. J. Corrada-Bravo, H. Corrada-Bravo, L. J. Villanueva-Rivera, and T. M. Aide, "Automated classification of bird and amphibian calls using machine learning: A comparison of methods," *Ecological Informatics*, vol. 4, no. 4, pp. 206–214, Sep. 2009.

[127] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.