DIGITAL COMMUNICATION RECEIVER ALGORITHMS AND ARCHITECTURES FOR REDUCED COMPLEXITY AND HIGH THROUGHPUT

BY

JUN WON CHOI

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Doctoral Committee:

Professor Andrew C. Singer, Chair
Professor Yoram Bresler
Associate Professor Minh N. Do
Assistant Professor Olgica Milenkovic
ABSTRACT

In this dissertation, efficient receiver algorithms and architectures for digital communications are studied. As the demand for higher data communication rate increases, the dimension of communication systems is rapidly growing, thereby requiring computationally efficient detection and decoding algorithms in the receiver. Hence, it is crucial to develop receiver algorithms that can offer good performance-complexity trade-offs in high dimensional communication systems such as multi-input multi-output (MIMO) systems and systems with a large delay spread. In this dissertation, computationally efficient receiver algorithms and low-power implementation of receiver architectures are investigated.

First, a low-complexity near maximum-likelihood (ML) detector, called the reduced-dimension ML search (RD-MLS), is proposed. The main idea of the RD-MLS is based on reduction of search space dimension. That is, a solution is searched over a subset of symbols to reduce the search complexity. In order to minimize the inevitable performance loss due to the search space reduction, a list tree search (LTS) algorithm is employed, which finds the best $K$ candidates over the reduced search space. A final solution is chosen among the $K$ candidates after extension to the full dimension via an MMSE decision-feedback (MMSE-DF) detector. Through computer simulations, we demonstrate that the RD-MLS algorithm achieves significant complexity reduction over the existing near ML detectors while limiting performance loss to within one dB from ML detection.

Second, a low complexity MIMO tree detector, called the improved soft-input soft-output $M$-algorithm (ISS-MA), is presented. The proposed detector is developed for iterative detection and decoding (IDD) systems, which are known to achieve near-optimal detection performance for MIMO channels. In order to improve the performance of tree detection, a look-ahead path metric is employed that accounts for the impact of unvisited paths of
the tree via an unconstrained linear MMSE estimator. Based on an analysis of the probability of correct path loss, we show that the improved path metric offers better detection performance than the conventional path metric. We also demonstrate through simulations that the ISS-MA provides a better performance-complexity trade-off than existing soft-input soft-output detection algorithms.

Third, a computationally efficient turbo equalization algorithm for underwater acoustic communications is studied. The performances of two popular linear turbo equalizers, a channel estimate-based minimum mean square error TEQ (CE-based MMSE-TEQ) and a direct-adaptive TEQ (DA-TEQ) technique, are compared in the presence of channel estimation errors and adjustment errors of a least mean square (LMS) adaptive algorithm. Next, an underwater receiver architecture built upon the LMS DA-TEQ technique is introduced. To maintain a performance gains over time-varying channels, the convergence speed of the LMS algorithm is improved via two methods: (1) data reusing and gear-shifting LMS and (2) reducing the length of the equalizer by capturing the sparse structure of underwater acoustic channels. Experimental results show great promise for this approach, as data rates in excess of 15 kbit/s could readily be achieved without error.

Lastly, an energy efficient estimation and detection problem is formulated for low-power digital filtering. Building on the soft digital signal processing technique that combines algorithmic noise tolerance and voltage scaling to reduce power, a minimum power soft error cancellation (MP-SEC) technique detects, estimates and corrects transient errors that arise from voltage over-scaling. These timing violation-induced errors, called soft errors, can be detected and corrected by exploiting the correlation structure induced by the filtering operation being protected, together with a reduced-precision replica of the protected operation. By exploiting a spacing property of soft errors in certain architectures, MP-SEC can achieve up to 30% power savings with no SNR loss and up to 55% power savings with less than 1 dB SNR loss, according to logic-level simulations performed for an example 25-tap frequency-selective filter.
TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION ........................................ 1
  1.1 Background ............................................. 1
  1.2 Purpose of this Study ............................... 4
  1.3 Key Contribution of Dissertation ............... 5
  1.4 Organization of Dissertation ................... 9

CHAPTER 2 NOTATION ........................................... 10

CHAPTER 3 LOW COMPLEXITY REDUCED DIMENSION MAX-
IMUM LIKELIHOOD SEARCH ............................. 12
  3.1 Introduction ........................................ 12
  3.2 Sphere Decoding Algorithm ................. 16
  3.3 Reduced Dimension ML Search (RD-MLS) Algorithm .... 19
  3.4 Discussion ...................................... 27
  3.5 Simulations .................................... 32
  3.6 Conclusions .................................. 39

CHAPTER 4 EFFICIENT SOFT-INPUT SOFT-OUTPUT TREE
DETECTION VIA AN IMPROVED PATH METRIC .......... 40
  4.1 Introduction .................................... 40
  4.2 Problem Description ......................... 43
  4.3 Improved Soft-Input Soft-Output M-Algorithm (ISS-MA) .... 46
  4.4 Performance Analysis ....................... 53
  4.5 Simulation and Discussion ............... 60
  4.6 Conclusions ................................ 69

CHAPTER 5 ADAPTIVE LINEAR TURBO EQUALIZATION
OF LARGE DELAY SPREAD TIME-VARYING CHANNEL .... 70
  5.1 Introduction .................................... 70
  5.2 System Description ............................ 74
  5.3 Review of Turbo Equalizer with Linear Structure .... 76
  5.4 Comparison of CE-Based MMSE-TEQ versus LMS DA-TEQ .... 80
  5.5 Underwater Acoustic Communication Receiver Based on
LMS DA-TEQ Algorithm .................................. 86
  5.6 Experimental Results ....................... 90
# Table of Contents

5.7 Conclusions ............................................................... 100

CHAPTER 6  LOW-POWER FILTERING VIA MINIMUM POWER
SOFT ERROR CANCELLATION (MP-SEC) .................................... 102
6.1 Introduction ................................................................. 102
6.2 Soft Error Cancellation Approach ..................................... 104
6.3 Energy Minimum Soft Error Cancellation .......................... 118
6.4 Discussion ................................................................. 124
6.5 Conclusions ................................................................. 134

CHAPTER 7  CONCLUSIONS ................................................ 135

APPENDIX A  SUMMARY OF THE RD-MLS ALGORITHM ................. 138
APPENDIX B  ASYMPTOTIC NORMALITY OF THE LMMSE
ESTIMATION ERROR PLUS NOISE ........................................ 140
APPENDIX C  PROOF OF (3.53) .............................................. 141
APPENDIX D  PROOF OF THEOREM 4.3.3 ............................... 145
APPENDIX E  PROOF OF (4.38) .............................................. 147
APPENDIX F  PROOF OF LEMMA 4.4.1 ................................. 148
APPENDIX G  PROOF OF THEOREM 4.4.2 ............................... 149
APPENDIX H  PROOF OF COROLLARY 4.4.3 ......................... 151
APPENDIX I  PROOF OF LEMMA 4.4.4 ................................. 152
APPENDIX J  DERIVATION OF (5.23) .................................... 153
APPENDIX K  DERIVATION OF (5.22) .................................... 155
REFERENCES ................................................................. 156
CHAPTER 1

INTRODUCTION

1.1 Background

A goal of digital communications is to send the largest amount of information over a communication channel possible with the fewest possible errors, under resource constraints such as transmitted power and bandwidth. The information sent from the transmitter is corrupted by the channel and the receiver attempts to recover the transmitted data at the lowest error probability. In this dissertation, computationally efficient and power-efficient receiver algorithms and architectures for wireless communications are studied.

Wireless channels have several distinct properties. A transmitted signal may experience reflection and scattering from objects in the path from the transmitter to the receiver so that multiple replicas of transmitted signal appear at the receiver with different propagation delays and power levels. Since the transmitted signals pass through independent paths, aggregate channel responses exhibit random characteristics. In particular, when the transmitter or receiver moves fast, time-varying channel responses often result. Such time-varying channels are often modeled by random processes and are called *fading channels*. The fading effect is detrimental to receiver performance since the receiver cannot recover the data with low error probability easily when the channel experiences a deep fade. A useful method to fight against a fading channel is to employ a diversity technique, which combines independent replicas of the signal appropriately. This diversity technique reduces the probability that the equivalent channel gains are in deep fade and improves receiver performance [1]. Sometimes, the maximum difference of propagation delays in multi-path signals is larger than a symbol period so that the channel response spans multiple symbol periods. Such channels are called *frequency-selective fading channels*. Since consecutive symbols interfere with
each other due to the large delay spread, this intersymbol interference (ISI) degrades the receiver performance. In order to recover the performance loss due to ISI, equalization techniques are frequently employed to mitigate the effects of the channel impulse response [2]. A typical example of a frequency-selective channel with large delay spread is an *underwater acoustic channel*. Due to the slow propagation speed, the underwater channel has long channel response. Furthermore, due to the dynamic motion of the water and waves, the channel response changes rapidly in time. These properties make reliable equalization of underwater channels difficult.

Recently, multi-input multi-output (MIMO) communication techniques have received much attention due to their ability to increase channel capacity without increasing the bandwidth. In MIMO techniques, the achievable data rate is increased by multiplexing independent streams over the additional spatial dimensions offered by multiple transmit and receive antennas. Since multiple data streams are transmitted simultaneously, they interfere with each other, requiring the receiver to mitigate this interference for reliable communication. When each transmitter-receiver link of the MIMO channel undergoes a fading, diversity techniques or multi-channel equalization methods are needed in the MIMO receiver. In Chapters 3, 4, and 5, efficient receiver structures are presented for various communication modules including both frequency-nonselective (flat) fading MIMO and frequency-selective fading MIMO systems.

In the early 1990s, it was shown that performance close to the Shannon capacity could be achieved for an additive white Gaussian noise (AWGN) channel via a particular concatenation of two convolutional codes. This concatenated code structure is called a *turbo code* [3]. In the original turbo code, parallel concatenation of two recursive systematic convolutional (RSC) codes was considered. In fact, this near-optimal performance could be achieved via an *iterative decoding* algorithm, wherein two constituent decoders exchanged soft information in an iterative manner. Since then, the turbo principle that embodies iterative decoding between two receiver components has been applied to various digital communication receivers, including channel equalization, multi-user detection, and MIMO detection. In these systems, the channel encoder and interference channels are considered as the *outer* code and the *inner* code of a serially concatenated turbo code, respectively, and signal detection and channel decoding are carried out iteratively to approach
the performance of optimal joint detection and decoding. These receiver systems based on the turbo principle are referred to as iterative detection and decoding (IDD) systems. In order to perform the IDD operation, the symbol detector and channel decoder should be implemented such that they can process soft inputs and produce soft outputs. In Chapters 4 and 5, efficient soft-input soft-output symbol detection techniques are studied for wireless MIMO detection and equalization of underwater acoustic communications.

As wireless communication devices become increasingly pervasive and essential, their low-power implementation is a crucial factor in battery life. Recently, a variety of low-power techniques have been proposed in various implementation levels, e.g., in circuit, logic, and system (algorithmic) levels. At the circuit level, two common approaches are used to reduce power: (1) decreasing the supply voltage and (2) reducing the switching capacitance in the system. In fact, dynamic power dissipation in DSP architectures is a quadratic function of the supply voltage, denoted $V_{dd}$, i.e. $P = C_L V_{dd}^2 f_s$, where $C_L$ is the effective switching capacitance and $f_s$ is the clock frequency [4]. Due to the quadratic effect on power, a supply voltage reduction scheme can be a powerful approach to achieving significant power savings. In this dissertation, power reduction via supply voltage scaling is studied in depth. Traditionally, the supply voltage was determined such that the critical path delay (the worst case delay of the architecture) was strictly less than a clock cycle, to ensure correct timing operations. However, this choice might be considered too conservative, since such worst case paths are excited rarely. Dynamic voltage scaling techniques are proposed to control the supply voltage by monitoring a workload of the system in real time. Recently, a more aggressive voltage reduction approach, called voltage overscaling has been introduced that scales the supply voltage beyond the level corresponding to this critical path delay. This voltage overscaling technique can be combined with algorithmic noise tolerance (ANT) techniques [5], which protect the main system from hardware faults occurring when timing constraints are violated. In Chapter 6, an energy-efficient ANT system is discussed in detail.
1.2 Purpose of this Study

The increases in capacity available from using multi-input multi-output (MIMO) communication techniques promise enormous gains in next-generation wireless systems. This may be achieved by performing spatial multiplexing of data streams over a high dimensional signal space. To push the throughput limit of such wireless systems, the system dimensionality is growing fast and hence rapidly becoming a computational burden. As such, efficient receiver (detection) algorithms must be developed. For example, the sphere decoding algorithm that is known as a powerful maximum likelihood (ML) detection technique for MIMO systems, exhibits an exponentially growing complexity in terms of problem dimension. This makes implementation of the ML detector infeasible for large-size systems. In Chapter 3, computationally efficient implementations of ML detection are investigated for uncoded MIMO systems. In Chapters 4 and 5, more emphasis is put on IDD receiver algorithms. An efficient soft-input soft-output tree detector is developed for wireless MIMO systems, and a low-complexity adaptive linear turbo equalizer is introduced for underwater acoustic communications. The primary goal of this study is to introduce low-complexity receiver structures that maintain near-optimal performance. It should be noted that the receiver algorithms presented in this dissertation are not restricted to a particular communication setup, but can be generalized to a variety of digital communication systems that can be modeled by the equation $y = Hx + n$, where $y$, $x$ and $n$ are the observation, transmitted symbols, and noise vector, respectively, and $H$ is the channel matrix. A variety of digital communication systems can be described through this model.

In addition to low-complexity implementation, power-efficient design of wireless systems is also important. In Chapter 6, effective power reduction techniques based on voltage overscaling are investigated for DSP systems. In particular, the ANT technique is studied in depth, which detects and corrects hardware errors occurring due to low supply voltage. Since timing violation errors, called soft errors, tend to have a large magnitude, the impact on system performance is often catastrophic. Correct cancellation of soft errors is crucial for proper system operations. In addition, it is important to reduce the power overhead of the ANT block. A constrained optimization problem for ANT design is formulated such that power consumed by the ANT system
is minimized under a given performance constraint, e.g., signal to noise (SNR) constraint.

1.3 Key Contribution of Dissertation

Each chapter of this dissertation addresses a number of topics for achieving a common goal: designing computationally and power-efficient receiver algorithms and architectures. In this section, the key contributions of this dissertation are described.

1.3.1 Low Complexity Maximum Likelihood (ML) Detection

Bit error rate (BER) optimal performance in uncoded MIMO systems is achieved by ML detection techniques [6–8]. Among a variety of detection algorithms that achieve ML performance, the sphere decoding technique has attracted much attention due to its efficient search mechanism [9]. Contrary to the exhaustive search that enumerates all symbol combinations, a sphere decoding efficiently reduces the search space into the symbol vectors inside a hyper-sphere with a certain radius. In spite of significant complexity reductions, it is known that the (average) complexity of sphere decoding grows exponentially in terms of the search space dimension [10]. In this study, a low-complexity ML detector is proposed based on a dimension reduction approach. The dimension of the original search space is reduced via a partitioned search. Specifically, the symbol vector is partitioned into two parts, strong symbols and weak symbols, according to an appropriate detection ordering [11]. Then, a tree search is performed over an enumeration of all combinations of the strong symbols. Before the tree search, an MMSE dimensionality reduction operator is applied to suppress the impact of the weak symbols on the received vector. Reduction of the search space dimension leads to an inevitable performance loss, as compared to the full dimensional search. To compensate the performance loss, multiple promising candidates are found via a list tree search (LTS) algorithm [12]. The main contribution of this study is to show that the LTS algorithm for finding the best $K$ symbol candidates can successfully lead to near-ML performance at a small increase in complexity. An LTS technique called the closest-$K$ list
stack algorithm is developed, which employs a stoping criterion to adjust the size of the candidate list adaptively. Asymptotic performance analysis shows that this multiple candidate search offers a significant algorithmic gain in error performance. In addition, simulation results confirm that the proposed technique achieves a better performance-complexity trade-off than existing near-ML detection algorithms.

1.3.2 Efficient Soft-input Soft-output Tree Detection

The iterative receiver algorithm based on the IDD principle consists of two components: a symbol detector and a channel decoder. An efficient soft-input soft-output symbol detector is studied in depth in this dissertation. In general, the soft-input soft-output symbol detector produces extrinsic information on symbols based on the channel observations and a priori information on the transmitted symbols. The a priori information is obtained from the channel decoder. To compute the extrinsic information, the symbol detector computes the a posteriori probability (APP) of the transmitted symbols, which requires marginalization over all symbol combinations. A tree detection technique is often used to reduce the complexity of computing the APP [13]. A tree search is performed to find a small number of promising symbol candidates and compute approximate APPs by marginalizing over those candidates. However, as mentioned above, the complexity of tree detection grows considerably with dimension and becomes impractical for high dimensional systems.

In this study, an efficient tree search algorithm is developed for soft-input soft-output symbol detection. A breadth-first search is adopted since its pipelined structure is suited for multiple candidate search. Among various breadth-first search techniques, a sub-optimal fixed complexity detection algorithm, called the $M$-algorithm, is chosen to prevent the complexity from growing for higher dimensions. Since the $M$-algorithm does not allow for back-tracking, it achieves substantial complexity reduction. Because the conventional path metric used in tree search methods accounts for only the information on the visited path, the path metric at early detection stages does not capture sufficient information about the likelihood that the true path lies on the path visited. As a result, a correct path is often rejected.
from the candidate selection phase of an \( M \)-algorithm in an early stage, thereby making subsequent search efforts inefficient. In order to improve the performance-complexity trade-off of the \( M \)-algorithm, an improved path metric, called the linear estimate-based look-ahead path metric is proposed, which accounts for the information on the unvisited part of the tree. By employing this metric, the proposed soft-input soft-output \( M \)-algorithm performs better sorting in candidate selection. A theoretical analysis of the probability of correct path loss is presented that demonstrates the advantage of the linear estimate-based look-ahead path metric when applied to the soft-input soft-output \( M \)-algorithm.

### 1.3.3 Adaptive Linear Turbo Equalization for Underwater Acoustic Communications

Underwater acoustic channels are doubly selective channels which exhibit large spread both in delay and Doppler. Due to their large delay spread, equalization of underwater channels typically leads to many equalizer taps, which causes high computational complexity and leads to poor tracking performance. In order to alleviate the high complexity, an orthogonal frequency division multiplexing (OFDM) system was considered [14]. Unfortunately, for fast time-varying channels, the OFDM approach suffers from inter-carrier interference (ICI), which deteriorates equalization performance. In this study, low-complexity equalization for single-carrier transmission is considered.

A linear turbo equalization technique is studied for underwater acoustic communications to achieve significant performance gains over the conventional decision-feedback equalizer (DFE). The complexity and performance trade-offs of a variety of turbo-equalization (TEQ)-based receiver architectures are explored. First, two popular linear turbo equalizers are reviewed: (1) a channel estimate-based MMSE turbo equalizer [15] and (2) an LMS direct adaptive turbo equalizer [16]. The channel estimate-based MMSE turbo equalizer incorporates an explicit channel estimate in the MMSE equalizer. In contrast, the LMS direct adaptive turbo equalizer estimates symbol directly using well-known adaptive algorithms. Since both turbo equalizers are suboptimal without a knowledge of channel, it is meaningful to compare the performance of two approaches. Mean square error (MSE) analysis as well
as extrinsic information transfer (EXIT) chart analysis are used for performance comparison. An underwater receiver architecture based on an LMS direct-adaptive turbo equalizer is introduced for underwater acoustic channels. The main contribution of this study is to show that the adaptive linear turbo equalizer achieves substantial performance gains over conventional decision feedback equalizers at reasonable complexity. In addition, experimental results are provided to evaluate the performance of these turbo equalizers in real underwater channels. The LMS-based turbo equalizer yields more than an order of magnitude performance gain for various configurations and distances from a transmitter.

1.3.4 Lower Power DSP Architecture via Minimum Power Soft Error Cancellation (MP-SEC)

A power-optimized algorithmic noise tolerance (ANT) technique is proposed to detect, estimate, and cancel soft errors using an ML criterion. The contribution of this study is two-fold. Most arithmetic units in DSP systems are based on least significant bit (LSB)-first computation. Hence, when the voltage overscaling technique is applied, most timing violation errors are likely to occur in most significant bits (MSBs). For a fixed voltage overscaling factor, soft errors occur only at a few designated MSBs. Hence, the magnitude of soft errors reflected at the output of a DSP block takes values in a discrete set, i.e., a multiple of $2^M$, where $M$ is the number of LSBs where soft errors do not occur. This discrete property of the soft errors is referred to as a spacing property. Soft error estimation is formulated as pulse amplitude modulation (PAM) signal detection problem from digital communications and the spacing property is exploited in deriving the ML estimate of the soft errors. Second, in order to reduce the overall power dissipation of the ANT-based DSP system, the power overhead occupied by the ANT block should be minimized. Towards this end, a constrained optimization problem is formulated, where power dissipation of the ANT block is minimized under a performance constraint. The solution is sought via a search over the precisions and the number of active taps of the soft error canceller. For a frequency-selective filter with fixed coefficients, a branch and bound (BB) technique is employed to search for the best resources of the soft error canceller. For an adaptive
filter, an automatic power control algorithm is developed that dynamically switches on and off the taps of the SEC filter.

1.4 Organization of Dissertation

The dissertation is organized as follows. In Chapter 2, notation used in the rest of the dissertation is introduced. In Chapter 3, a low-complexity ML detection algorithm, called reduced dimension ML search (RD-MLS), is described. First, an MMSE dimension reduction operator is introduced, which reduces the dimension of the search space. Then, the description of a closest $K$ list stack algorithm is provided. The asymptotic error analysis and simulation results are presented to evaluate the performance of the RD-MLS. In Chapter 4, an efficient soft-input soft-output tree detector for a wireless MIMO system is introduced. The improved path metric that accounts for the contribution of unvisited paths is derived for the soft-input soft-output M-algorithm. The probability of correct path loss is analyzed to demonstrate the performance gain of the new path metric. Simulation results are presented to compare the performance of the proposed detector with the existing soft-input soft-output detectors. In Chapter 5, practical application of an adaptive linear turbo equalizer to underwater acoustic communications is studied. Two popular but sub-optimal linear turbo equalizers are compared via MSE and EXIT chart analysis. An underwater receiver architecture based on the LMS directive-adaptive turbo equalizer is described. The experimental results are provided to demonstrate the performance gain of the LMS turbo receiver over state-of-the-art conventional receivers. In Chapter 6, the power-optimum design of an ANT system is presented. ML estimation of timing errors (called soft errors) is formulated and a power-optimum design of the ANT system is presented. Simulation results are provided to evaluate the performance of the proposed ANT system. In Chapter 7, some conclusions are presented.
We briefly summarize the notation used in this paper.

- Uppercase and lowercase letters written in boldface are used for matrix and vector notation, respectively.

- The superscripts $(\cdot)^T$ and $(\cdot)^H$ denote transpose and conjugate transpose, respectively.

- $\| \cdot \|^2$ denotes an $L_2$-norm square of a vector.

- $\text{diag}\{x_1, \cdots, x_n\}$ is a diagonal matrix whose diagonals are $x_1, \cdots, x_n$.

- $\Re(x)$ and $\Im(x)$ denote the real and imaginary part of $x$, respectively.

- $f_{x_1, x_2, \cdots, x_n}(a_1, a_2, \cdots, a_n)$ denotes a joint probability density function (PDF) for the random variables $x_1, x_2, \cdots, x_n$.

- $\chi^2_k$ denotes a chi-square distribution with $k$ degrees of freedom (DOF).

- $F_\chi(\cdot; k)$ and $F^{-1}_\chi(\cdot; k)$ are the cumulative density function (CDF) and the inverse CDF of the $\chi^2$-random variable with $k$ DOF, respectively.

- $Q(x)$ denotes the $Q$-function defined as $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$.

- $\mathcal{CN}(m, \sigma^2)$ denotes a circular symmetric complex Gaussian with a mean $m$ and variance $\sigma^2$.

- $\mathcal{CN}(\mathbf{m}, \Sigma)$ is a complex Gaussian with mean $\mathbf{m}$ and covariance matrix $\Sigma$, and $\mathcal{N}(\mathbf{m}, \Sigma)$ is a real Gaussian.

- $\mathbf{1}_{i \times j}$ is the $i \times j$ matrix where entries are all one. $\mathbf{0}_{i \times j}$ is a zero matrix defined similarly. Note that $\mathbf{1}_i$ and $\mathbf{0}_i$ represent an $i \times i$ square matrix. The subscript for these matrices is omitted without risk of confusion.
\begin{itemize}
  \item $\text{Var}(x) = E[x^2] - E[x]^2$. $\text{Cov}(x) = E[(x - E[x])(x - E[x])^H]$ and $\text{Cov}(x, y) = E[(x - E[x])(y - E[y])^H]$.
  \item $\text{tr}(\cdot)$ denotes a trace operation.
  \item For a Hermitian matrix $A$, $A \succeq 0$ (or $A \succ 0$) means that $A$ is semi-positive definite (or positive definite).
\end{itemize}
CHAPTER 3
LOW COMPLEXITY REDUCED DIMENSION MAXIMUM LIKELIHOOD SEARCH

3.1 Introduction

The complex-domain relationship between the transmitted symbol and received signal vector in many communication systems can be expressed as

\[ y = Hx + w, \]  

(3.1)

where \( x \) is the transmitted vector whose entries are chosen from a finite symbol alphabet, \( y \) and \( w \) are the received signal and noise vectors, respectively, and \( H \) is a channel matrix. Multiple-input multiple-output (MIMO) links are a typical example described by this model. In order to achieve the diversity and multiplexing gains promised by MIMO technologies [17, 18], a powerful MIMO detection scheme for recovering the transmitted symbol with minimal error is indispensable. In particular, a maximum likelihood (ML) tree search algorithm referred to as sphere decoding (SD) has received much attention in recent years [6–9, 19]. The ML search algorithm searches over the lattice points spanned by noiseless channel outputs \( Hx \) to find the one with minimum value of \( \| y - Hx \|^2 \), where a Gaussian noise assumption has been made. Instead of enumerating all lattice points, the SD algorithm restricts the search space to within a sphere centered at the received vector, thereby achieving a considerable reduction in computational complexity. In spite of this benefit, the computational burden of the SD algorithm is still a major concern, since its expected complexity remains exponential with respect to problem size for a fixed signal-to-noise ratio (SNR) [10]. Considering the growing demand for high data rate services in next generation wireless systems, it remains a challenge to apply the SD algorithm to MIMO systems of large dimension and high-order constellations.
There have been a number of approaches to reduce the complexity of the SD algorithm, such as the Schnorr-Euchner enumeration [20–22], descending probabilistic ordering [23], increasing radius sphere decoding [24] and the parallel competing branch algorithm [25]. Other approaches trading performance for complexity include the radius scheduling method [26], $K$-best sphere decoder [27], probabilistic tree pruning algorithm [28, 29], sequential Fano decoders [30], $M$-algorithm [31], $K$-algorithm [32], and semi-definite relaxation [33].

In this chapter, we introduce a near-ML detection technique, referred to as a reduced dimension ML search (RD-MLS) that provides significant complexity reduction, yet maintains near-ML performance. By reducing the dimension of the search space from $n_t$ to $n_1$ ($n_1 < n_t$), the RD-MLS directly achieves a significant reduction in the number of lattice points that must be searched from $M^{n_t}$ to $M^{n_1}$. Owing to the direct benefit on complexity, there have been a number of studies [34–42] on partitioned search techniques where a subset of the symbol vector is estimated by sub-optimal methods and the remaining symbols are more carefully searched. In [34,35], a symbol estimate is obtained by concatenation of the elements obtained by an exhaustive search and those obtained via zero-forcing or decision feedback estimation. However, due to imperfect decisions made by linear detectors, these schemes suffer a performance loss. In order to mitigate the performance loss caused by dimension reduction, techniques generating multiple candidates for a portion of the symbols and choosing a solution among those concatenated with linear estimates have been proposed. It is shown in [36,37] that enumeration of full candidates achieves the diversity gain of the exact ML detector. Similar methods employing more refined post-detection schemes are found in [38] and [40]. In [39], an approach to selecting symbol candidates that are close to the minimum mean square error (MMSE) estimate was proposed. Such a partitioned search idea was extended to soft-output maximum a posteriori (MAP) detection in [41] and [42]. These schemes allow for a fixed complexity [36–38,40] or reduced worst-case complexity [39] but often require a large number of symbol candidates to achieve near-ML performance, resulting in considerable complexity.

Our RD-MLS technique is distinct from these approaches in two respects. First, rather than performing an exhaustive or ad hoc enumeration of candidates as in [36–40], we employ a list tree search (LTS) method [12,13] to find
promising symbol candidates. The LTS is employed after applying an MMSE dimension reduction operator that performs soft cancellation of interference (see the illustration in Figure 3.1).

While the LTS has been used to perform soft output decoding [12,13], its application to partitioned search has not yet been explored to our knowledge. In fact, owing to the LTS, the number of candidates generated to achieve near-ML performance reduces significantly compared to previous schemes [36–40]. Second, we introduce an efficient LTS algorithm, called a closest-K list stack algorithm (K-LSA), which finds a flexible, but limited number of closest lattice points. Contrary to previous LTS algorithms visiting a large number of lattice points to obtain accurate a posteriori probabilities [12,13], the K-LSA reduces the number of lattice points to visit by employing a stopping criterion which terminates the candidate search adaptively as well as a probabilistic bias for pruning additional unnecessary branches. As a result, the RD-MLS can maintain modest complexity for various channel and noise conditions.

Through an asymptotic performance analysis, we observe that the diversity gain of the RD-MLS is at most \( n_r + n_1 - n_t \), compared to \( n_r \) of the full dimensional ML search. We show that the K-LSA can bring an improvement in the effective SNR by a factor proportional to the size of candidate list, thereby compensating for the diversity reduction. We observe through simulation that the performance loss due to the diversity gain reduction is partially mitigated by the additional algorithmic gains offered by the K-LSA, leading to performance that appears close to that of the ML detector. In addition, it is shown that the RD-MLS achieves significant complexity reduction over the SD algorithm as well as previous near-ML approaches [39].

The rest of this chapter is organized as follows. After describing the system model in Section 3.2, we briefly review the SD algorithm and its computational complexity in 3.3. We present an asymptotic performance analysis in Section 3.4. Simulation results are provided in Section 3.5.
Figure 3.1: Illustration of the previous partitioned search schemes and the RD-MLS detection: (a) fixed-complexity sphere decoder [37], (b) B-Chase detector (l) [39], and (c) RD-MLS detector. The RD-MLS detector finds the candidates using the LTS after reducing the problem size via the MMSE dimension reduction operator while the schemes in [37] and [39] fully or partially enumerate the candidates. Refer to Section 3.3 for the details of the RD-MLS algorithm.
3.2 Sphere Decoding Algorithm

3.2.1 Sphere Tree Search

After the real conversion of complex matrices and vectors, the ML detection problem can be written by

\[ x_{\text{ml}} = \arg \min_{x \in \mathbb{F}} \left\| \begin{bmatrix} \Re(y) \\ \Im(y) \end{bmatrix} - \begin{bmatrix} \Re(H) & -\Im(H) \\ \Im(H) & \Re(H) \end{bmatrix} \begin{bmatrix} \Re(x) \\ \Im(x) \end{bmatrix} \right\|^2, \]  

(3.2)

where \( H \) is the \( n_r \times n_t \) channel matrix and \( x \) is the \( n_t \times 1 \) vector comprising elements of the \( M \)-quadrature amplitude modulation (QAM) set \( \mathbb{F} \) defined as

\[ \mathbb{F} = \left\{ x_r + j x_i \mid x_r, x_i \in \left\{ \frac{-\sqrt{M} + 1}{\lambda}, \frac{-\sqrt{M} + 3}{\lambda}, \ldots, \frac{\sqrt{M} - 3}{\lambda}, \frac{\sqrt{M} - 1}{\lambda} \right\} \right\}, \]  

(3.3)

where \( \lambda \) is chosen to satisfy the normalization condition \( E[xx^H] = I_{nt} \). For example, \( \lambda = \sqrt{10} \) for 16-QAM and \( \lambda = \sqrt{42} \) for 64-QAM modulation, respectively. The SD algorithm searches the lattice points inside a hypersphere of radius \( \sqrt{B} \), centered at the received vector \( y \) [7,9]. The sphere constraint is expressed as

\[ \|y_r - H_r x_r\|^2 \leq B, \]  

(3.4)

where \( y_r = \begin{bmatrix} \Re(y) \\ \Im(y) \end{bmatrix} \), \( x_r = \begin{bmatrix} \Re(x) \\ \Im(x) \end{bmatrix} \), and \( H_r = \begin{bmatrix} \Re(H) & -\Im(H) \\ \Im(H) & \Re(H) \end{bmatrix} \). In the sequel, we let \( n'_r = 2n_r \) and \( n'_t = 2n_t \). In order to perform a systematic tree search, following [7], we perform the QR decomposition of the channel matrix \( H_r \), i.e.,

\[ H_r = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}, \]  

(3.5)

where \( Q_1 \) and \( Q_2 \) are \( n'_r \times n'_t \) and \( n'_t \times (n'_r - n'_t) \) matrices and \( R \) is an upper triangular matrix whose diagonal elements are non-negative. Since a norm operation is invariant to orthogonal transform, the sphere constraint of (3.4)
can be rewritten

\[ \|y'_r - R x_r\|^2 = \sum_{i=1}^{n'_t} \left| y'_i - \sum_{l=i}^{n'_i} r_{i,l} x_l \right|^2 \leq B', \quad (3.6) \]

where \( y'_r = [y'_1, \ldots, y'_{n'_t}]^T, x_r = [x_1, \ldots, x_{n_t}]^T, y'_r = Q^T_1 y_r, B' = |B - \|Q^T_2 y_r\|^2|, \)
and \( r_{i,j} \) is the \((i, j)\)th entry of \( R \). Emphasizing that each term in the summation in (3.6) is a function of \( x_i, \ldots, x_{n'_t} \), (3.6) becomes

\[ B_1 (X^{n'_t}_1) + B_2 (X^{n'_t}_2) + \cdots + B_{n_t} (X^{n'_t}_{n'_t}) \leq B', \quad (3.7) \]

where \( X^j_i \) denotes a set of variables \( x_i, \ldots, x_j \) and \( B_i(X^{n'_t}_i) = \left| y'_i - \sum_{l=i}^{n'_i} r_{i,l} x_l \right|^2 \).

The SD algorithm can be interpreted as a tree search, where each node is associated with the variables \( X^{n'_t}_i \) (see [6,7] for details). A path metric assigned for each node of the tree is defined as

\[ d_k (X^{n'_t}_k) = \sum_{i=k}^{n_t} B_i (X^{n'_t}_i). \quad (3.8) \]

A complete path starting at root and ending at the bottom of the tree represents a realization of a symbol vector and the path having minimum path metric among all complete paths becomes the ML solution of the tree search. In order to find the ML path, the following relationship between parent-child pair nodes is employed:

\[ d \left( X^{n'_t}_k \right) = d \left( X^{n'_t}_{k+1} \right) + B_k (X^{n'_t}_k). \quad (3.9) \]

By additivity, we have that \( d (X^{n'_t}_l) \geq d (X^{n'_t}_m) \) for \( l \leq m \) and hence, the path metric \( d (X^{n'_t}_k) \) monotonically increases with tree depth \( k \). Hence, for a node whose path metric violates the sphere condition, i.e., \( d \left( X^{n'_t}_k \right) > B' \), all leaf nodes of its subtree violate the condition as well, so that the node \( X^{n'_t}_k \) and its subtree are removed from the tree without loss of optimality, so long as at least one leaf satisfies the sphere condition.

Two popular methods for searching the nodes in a branch are Pohst enumeration [9] and SE enumeration [20]. In Pohst enumeration, natural span-
ning from the minimal to the maximal value is used within the interval,
\[ X_k^{\text{min}} \leq x_k \leq X_k^{\text{max}} \] (3.10)
where
\[ X_k^{\text{max}} = \left\lfloor \frac{1}{r_{k,k}} \left( y_k' - \xi_k + \sqrt{B' + d \left( X_{nt}^{n_t} + 1 \right)} \right) \right\rfloor, \quad (3.11) \]
\[ X_k^{\text{min}} = \left\lceil \frac{1}{r_{k,k}} \left( y_k' - \xi_k + \sqrt{B' - d \left( X_{nt}^{n_t} \right)} \right) \right\rceil, \quad (3.12) \]
where \( \xi_k = \sum_{i=k+1}^{n_t} r_{k,j} x_i \). In contrast, the SE method enumerates the admissible points \( x_k \) in a zig-zag order from the mid-point \( x_{k,\text{mid}} = \left\lfloor \frac{1}{r_{k,k}} (y_k' - \xi_k) \right\rfloor \).
That is, the SE enumeration spans \( x_{k,\text{mid}}, x_{k,\text{mid}} + 1, x_{k,\text{mid}} - 1, x_{k,\text{mid}} + 2, \ldots \), when \( y_k' - \xi_k - r_{k,k} x_{k,\text{mid}} \geq 0 \), and \( x_{k,\text{mid}}, x_{k,\text{mid}} - 1, x_{k,\text{mid}} + 1, x_{k,\text{mid}} - 2, \ldots \), otherwise. By traversing the tree with this branch ordering mechanism, all lattice points inside the sphere are visited and the final lattice point having the minimum path metric becomes the ML point.

### 3.2.2 Complexity of SD Algorithm

Due to the data-driven nature of the search, computational complexity of the SD algorithm is non-deterministic. Expected complexity has been widely considered for assessing the relative computational complexity of various approaches to the SD algorithm [7, 10]. Assuming a uniform distribution of computational cost across the nodes, a lower bound on the expected number of nodes visited by the search algorithm [10] becomes [7, 8]
\[ E\left[ N \right] \geq \frac{M^{m_t} - 1}{\sqrt{M} - 1}, \quad (3.13) \]
where \( N \) is the number of the visited nodes, \( M \) is a modulation order, and \( \eta \) is the complexity exponent given by
\[ \eta = \frac{1}{2} \left( 1 + \frac{4(M - 1)}{3\lambda^2} \text{SNR} \right)^{-1}, \quad (3.14) \]
where \( \lambda \) is defined in (4.2). Since \( E\left[ N \right] \) increases exponentially with \( n_t \), search dimension reduction poses a clear strategy for reducing complexity. However,
simple reduction of the search dimension might cause significant performance loss, so that a careful mechanism for mitigating such performance loss is needed. In the following section, we propose a dimension reduction algorithm that attempts to mitigate such performance loss, while enabling substantial complexity reduction.

3.3 Reduced Dimension ML Search (RD-MLS) Algorithm

The structure of the RD-MLS system is depicted in Figure 3.2. The dimension reduction operator reduces the system dimension by suppressing interference, that is, the contribution of symbols not participating in the LTS operation. New observation $z$ and system matrix $G$ obtained from the dimension reduction operator are delivered to the $K$-LSA block, which performs the closest lattice point search over the reduced search space. Since the closest point of the reduced-dimension system is not necessarily equal to the ML solution of the original system, we find multiple candidates via LTS. Then, each candidate of the list (denoted as $L_1$) found by the $K$-LSA is extended to the full symbol dimension via MMSE-decision feedback (MMSE-DF) estimation. Among the extended list $L$, a final estimate is chosen based on $L_2$-norm criterion.
3.3.1 Dimension Reduced ML Problem

As the first step for dimension reduction, the symbol vector $x$ is divided into two vectors $x_1 \in F^{n_1}$ and $x_2 \in F^{n_2}$ ($n_2 = n_t - n_1$). With knowledge of the received data $y$ and channel $H$, the ML solution becomes

$$ x_{ml} = \arg \min_{x \in F^n} \| y - Hx \|^2 $$

$$ = \arg \min_{x_1 \in F^{n_1}, x_2 \in F^{n_2}} \| y - H_2 x_2 - H_1 x_1 \|^2, \quad (3.15) $$

where $H_1$ and $H_2$ are the sub-matrices constructed by $n_1$ and $n_2$ columns of $H$, respectively. Denoting $x_{ml}^T = [x_{1,ml}^T \ x_{2,ml}^T]^T$, $x_{1,ml}$ can be expressed as

$$ x_{1,ml} = \arg \min_{x_1 \in F^{n_1}} \| y - H_2 g(y, x_1) - H_1 x_1 \|^2, \quad (3.16) $$

$$ g(y, x_1) = \arg \min_{x_2 \in F^{n_2}} \| y - H_2 x_2 - H_1 x_1 \|^2. \quad (3.17) $$

Insertion of (3.17) into (3.16) will return to (3.15) and no dimension reduction is therefore achieved. In order to restrict the search space within that spanned by $x_1$, we use a linear estimate of $x_2$ instead of $g(y, x_1)$. Employing a linear minimum mean square (MMSE) estimate of $x_2$, i.e., $\hat{x}_2$, for a given $x_1$, we obtain the approximate ML estimate,

$$ \tilde{x}_{1,ml} = \arg \min_{x_1 \in F^{n_1}} \| y - H_2 \hat{x}_2 - H_1 x_1 \|^2. \quad (3.18) $$

Assuming that $x_1$ is given, the linear LMMSE estimate of $x_2$ is

$$ \hat{x}_2 = f(y, x_1) = R_{y|x_1} R_{x_1|x_1}^{-1} (y - E[y|x_1]) \quad (3.19) $$

$$ = H_2^H (H_2 H_2^H + \sigma^2_w I)^{-1} (y - H_1 x_1). \quad (3.20) $$

Using (3.18) and (3.20), the approximate ML estimate $\tilde{x}_{1,ml}$ becomes

$$ \tilde{x}_{1,ml} = \arg \min_{x_1 \in F^{n_1}} \| y - H_2 H_2^H (H_2 H_2^H + \sigma^2_w I)^{-1} (y - H_1 x_1) - H_1 x_1 \|^2. \quad (3.21) $$

Defining the projection operator

$$ Z = \sigma^2_w (H_2 H_2^H + \sigma^2_w I)^{-1}, \quad (3.22) $$
then, (3.21) is written as

$$\tilde{x}_{1,ml} = \arg\min_{x_1 \in F^n_1} \|Zy - ZH_1 x_1\|^2.$$  

(3.23)

Further, by denoting \(z = Zy\) and \(G = ZH_1\), we obtain the integer least squares problem

$$\tilde{x}_{1,ml} = \arg\min_{x_1 \in F^n_1} \|z - Gx_1\|^2.$$  

(3.24)

### 3.3.2 MMSE Dimension Reduction Operator

The preprocessing operation consists of (1) an application of the linear operator \(Z\) and (2) a tree search over the transformed system. From the relationship \(y = H_1 x_1 + H_2 x_2 + w\), we can express \(z\) as

$$z = Zy = Gx_1 + Zr,$$  

(3.25)

where \(r = (H_2 x_2 + w)\). As discussed, \(H_2 x_2\) is an interference term to detect \(x_1\) and the contribution of this term is minimized by the preprocessing. In fact, from the definition of \(Z\), one can show that (3.22) can be written as

$$Z = \arg\min_{Z'} E \left[ \|w - Z'r\|^2 \right] = R_{wr} R_{rr}^{-1},$$  

(3.26)

which is the LMMSE estimate of \(w\), i.e., \(\hat{w} = Zr\). Hence, (3.25) becomes

$$z = Gx_1 + \hat{w},$$  

(3.27)

where \(\hat{w} = w + e\) and \(e\) is the LMMSE estimation error. In Section 3.4.1, we will show that the performance of the RD-MLS detector is limited by the detection performance of \(x_1\). A potentially useful choice for \(H_1\) and \(H_2\) would be that which maximizes the receiver SNR for detecting \(x_1\) in (3.27), i.e.,

$$(H_1, H_2) = \arg\max_{H_1', H_2'} \text{SNR}(H_1', H_2'),$$  

(3.28)
where
\[
\text{SNR} \left( \mathbf{H}'_1, \mathbf{H}'_2 \right) \triangleq \frac{E(||\mathbf{Gx}_1||^2)}{E(||\mathbf{w}||^2)} \quad (3.29)
\]
\[
= \frac{\text{tr} \left( \mathbf{ZH}'_1 (\mathbf{H}'_1)^H \mathbf{Z}^H \right)}{\text{tr} \left( \sigma_w^4 \left( \mathbf{H}'_2 (\mathbf{H}'_2)^H + \sigma_w^2 \mathbf{I} \right)^{-1} \right)}. \quad (3.30)
\]

To obtain an optimal partition from (3.28), \( \binom{n_t}{n_1} \) choices would need to be examined, which is clearly burdensome for a large \( n_t \). Thus, a simple scheme such as V-BLAST symbol ordering [11] or probabilistic symbol ordering [23] can be an alternative, where the symbols associated with \( \mathbf{H}_1 \) are detected first.

### 3.3.3 Closest-\( K \) List Stack Algorithm (\( K \)-LSA)

As mentioned, due to the reduced dimensionality of the search, \( \widehat{\mathbf{x}}_{1,\text{ml}} \) is not guaranteed to be the true ML solution \( \mathbf{x}_{1,\text{ml}} \), and thus performance loss is unavoidable. In order to mitigate the loss, a list tree search generating multiple candidates for \( \mathbf{x}_1 \) is employed. In [13], the list SD (LSD) algorithm to find \( N \) best lattice points was proposed. Contrary to the SD algorithm where the radius of the sphere is updated dynamically for each candidate found, the LSD algorithm maintains a fixed radius until it finds the \( N \) best points. The radius is updated only when the list is full and a new candidate replacing an existing one is found. In many cases, therefore, excessive numbers of lattice points are visited, which can easily reduce the benefits of dimension reduction. To maintain the complexity gains of the reduced dimension search while pursuing the performance gain of the LTS, we employ a closest-\( K \) list stack algorithm.

As a best-first tree search technique, the stack algorithm (SA) [24, 43] extends the node in a tree with the minimum cost metric. For every node extension, node information is stored in the stack and the best node is chosen based on a cost metric of the nodes in the stack. In the reduced system (3.24) described in Section 3.3.1, the cost metric for a tree node \( X'_{i}^{n'_i} \) after
the conversion to the real domain is defined by [30]

\[ a_i \left( X_{n_i}^{n'} \right) = \min_{x_{i-1} \in C_i} d_{i-1} \left( X_{i-1}^{n_i} \right) + \eta_{i-1}, \]

(3.31)

where \( d_{i-1} \left( X_{i-1}^{n_i} \right) \) is the path metric of the node \( X_{i-1}^{n_i} \) (defined in Section 3.2.1), \( C_i \) is the set of child nodes of \( X_{i}^{n_i} \) not generated, and \( \eta_i \) is a bias term penalizing short paths [12, 30, 43]. Notice that \( n'_1 = 2n_1 \) due to the real conversion. While the search of the SA is finished once it arrives at the first leaf node corresponding to the ML point, the \( K \)-LSA continues the search until it finds \( K - 1 \) additional closest points. However, due to the multiple-lattice-point search, large numbers of back-tracking operations occur. In order to alleviate the complexity increase, the \( K \)-LSA employs two measures, a stopping criterion and probabilistic bias, which will be described as follows.

The number of points collected in the candidate list directly impacts the complexity of the RD-MLS detector. In order to adjust the candidate list size effectively, we terminate the search before filling the list via a stopping criterion. After finding the first closest point, the stopping criterion checks if the path metric of the subsequently found closest points \( d_1 \left( X_{1}^{n_i} \right) \) satisfies the condition, i.e., \( d_1 \left( X_{1}^{n_i} \right) > D \). Then, the search is stopped if the stopping condition is satisfied. In order to choose \( D \), we consider the path metric of the actual transmitted symbols denoted as \( \mathbf{x}_1 \), i.e.,

\[ d_1 \left( X_{1}^{n_i} \right) = \| \mathbf{z} - \mathbf{Gx}_1 \|^2 = \| \hat{\mathbf{w}} \|^2, \]

(3.32)

where \( \hat{\mathbf{w}} \) was defined in Section 3.3.2. The parameter \( D \) can be chosen such that the probability of the path metric for the transmitted symbols being less than \( D \) equals some probability \( P_\epsilon \), i.e.,

\[ Pr \left( \| \hat{\mathbf{w}} \|^2 < D \right) = P_\epsilon. \]

(3.33)

Since it is hard to find \( D \) analytically for non-Gaussian \( \hat{\mathbf{w}} \), we introduce a heuristic for choosing \( D \). Using the path metric of first found closest point \( \hat{X}_1^{n_i} \), we let \( D = mD_1 \left( \hat{X}_1^{n_i} \right) \), where \( m(>1) \) is called a stopping parameter. Using this condition, the \( K \)-LSA collects the points whose path metric is less than \( m \) times of that first found. The rationale behind this choice is
that for the case of benign channel/noise conditions (e.g., small $\|\hat{w}\|^2$), the path metric of the transmitted symbol $\overline{x}_1$ is significantly smaller than that of other lattice points, so that it is highly possible to find $\overline{x}_1$ with only a small number of candidates. In the opposite conditions, many paths have a similar path metric so that the number of candidates should increase in efforts to keep the true solution in the list. Refer to the illustration in Figure 3.3. Due to the stopping criterion, the candidate list size is adjusted to the channel and noise condition and lattice points with dominant metric are stored in the list. In Section 3.4.1, we will show that with this stopping criterion, the LTS offers performance improvement over detection without LTS.

Next, we introduce a method to choose the bias term $\eta_i$ in (3.31). To compensate the path metric of short paths such that the most likely path chosen to extend appropriately accounts for the differing path lengths of the nodes visited, a proportionate bias term has been used in the traditional stack algorithm [30, 43]. While the bias term $\eta_i$ in these approaches can be approximate, $\eta_i$ in our method is chosen by taking into account the contributions of random noise $w$ in the unvisited paths. Specifically, we model $\eta_i$ to represent the noise contributions from the unvisited levels of the tree, $|w_1|^2 + \cdots + |w_{i-1}|^2$, and then assign the probabilistic condition $Pr (|w_1|^2 + \cdots + |w_{i-1}|^2 < \eta_i) = P_{\text{prun}}$, where $P_{\text{prun}}$ is the pruning probability. For a specific $P_{\text{prun}}$, $\eta_i$ is given by

$$\eta_i = F_{\chi}^{-1} (P_{\text{prun}}; i - 1).$$

(3.34)

In general, $\eta_i$ decreases with the tree depth $n_1' - i$ assessing larger bias to short paths. By using an appropriate value for $P_{\text{prun}}$ (such as $P_{\text{prun}} \sim 0.2$ through empirical simulations), we can achieve substantial reduction in backtracking operations with negligible performance loss. A similar approach has been applied to the SD algorithm in [29] and to SA in [44]. Our approach differs from that of [44] in that rather than using the expected noise power to obtain a bias term, we employ a probabilistic condition (3.34) to choose $\eta_i$. Hence, our bias term can be controlled more flexibly through the parameter $P_{\text{prun}}$. 

24
Figure 3.3: Illustration of the stopping criterion for the $2 \times 2$ system $G$. The dark circle $O$ corresponds to the transmitted symbol vector $Hx_1$. The open circles are the observed signal vectors for two scenarios of (a) bad and (b) good noise conditions, i.e., $\|\hat{w}_A\|^2 > \|\hat{w}_B\|^2$. As shown in the figure, for bad noise realizations, the points near the observed vector $z_A$ are likely to have a similar distance metric so that many lattice points are collected. On the other hand, for good noise realizations, only a few lattice points have small distance metric. With high probability, the ML point will be found among them. The stopping criterion with $m = 1.5$ selects the points inside the gray circle as candidates, and hence it collects $\{A, B, C, O\}$ for the case (a) and only $\{O\}$ for the case (b).
3.3.4 Postprocessing

The postprocessing operates in two steps. First, for each of the candidates for $x_1$ obtained by the LTS, we generate the symbol vectors of full dimension by concatenating them with the MMSE-DF estimates of $x_2$. Then, among those vectors, we choose one minimizing the Euclidean distance metric as the final output.

We assume that the columns of $H_2$ and entries of $x_2$ are arranged based on the detection ordering provided in [23] or [11]. We let $\hat{x}_i^1$ be the $i$th element in the candidate list and $\hat{x}_1^2 = [\hat{x}_{2,1}^i, \cdots, \hat{x}_{2,n_2}^i]^T$ be the corresponding MMSE-DF estimate of $x_2$ given $\hat{x}_1^i$. Further, we let $h_{2,k}$ be the $k$th column of $H_2$. To obtain the MMSE-DF estimate of $x_2$, for each candidate of $\mathcal{L}$, we first subtract the effect of $\hat{x}_1^i$ from $y$, i.e., $\hat{y}_i = y - H_1 \hat{x}_1^i$, and then obtain the estimates $\hat{x}_{2,1}^i, \cdots, \hat{x}_{2,n_2}^i$ successively. The MMSE-DF detection steps can be summarized as follows:

**STEP 1**: Compute $\hat{y}_i = y - H_1 \hat{x}_1^i$ for all $i = 1, \cdots, K$.

**STEP 2**: (Iteration) for all $i = 1, \cdots, K$, for $k = 1 : n_2$,

compute

$$\hat{x}_{2,k}^i = f_k^H \hat{y}_i^{(k)} \quad (3.35)$$
$$\hat{x}_{2,k}^i = \text{slicer} \left( \hat{x}_{2,k}^i \right) \quad (3.36)$$
$$\hat{y}_i^{(k+1)} = \hat{y}_i^{(k)} - h_{2,k} \hat{x}_{2,k}^i \quad (3.37)$$

end

Here “slicer” denotes the function of mapping the complex-valued $\hat{x}_{2,k}^i$ to the nearest transmitted constellation point. The MMSE-DF filter coefficient $f_k$ is given by [45, 46]

$$f_k = \left( \sum_{j=k}^{n_2} h_{2,j} h_{2,j}^H + \sigma_w^2 I \right)^{-1} h_{2,k} \quad (3.38)$$

Once $\hat{x}_2^i$ is obtained by (3.35)-(3.37), $\hat{x}_1^i$ and $\hat{x}_2^i$ are concatenated for the
final list search as

$$L = \left\{ \hat{x}_{\text{ext}}^1, \ldots, \hat{x}_{\text{ext}}^K \right\} = \left\{ \begin{bmatrix} \hat{x}_1^1 \\ \hat{x}_2^1 \end{bmatrix}, \ldots, \begin{bmatrix} \hat{x}_1^K \\ \hat{x}_2^K \end{bmatrix} \right\}$$ (3.39)

and the element of $L$ minimizing the cost function (3.40) becomes the final output of the RD-MLS

$$\hat{x}_{\text{ml}} = \arg \min_{a \in L} \| y - Ha \|^2.$$ (3.40)

The whole detection procedure of the RD-MLS algorithm is summarized in Appendix A.

### 3.4 Discussion

The two key parameters affecting the complexity and performance of the RD-MLS are the dimension $n_1$ and stopping parameter $m$. In this section, we present the performance analysis for the RD-MLS and show how these parameters affect error performance.

#### 3.4.1 Performance Analysis

The aim of this subsection is to derive an upper bound on the detection error probability for the RD-MLS. To make the analysis tractable, we consider the case where the system matrix $H$ is partitioned without any column ordering so that the elements of $H_1$ and $H_2$ are assumed to be random i.i.d. complex Gaussian. That is, we disregard the partitioning criterion described in Section 3.3.2 for the simplicity of analysis. The detection error probability of the RD-MLS detector, $P_{\text{err}}$, is defined as

$$P_{\text{err}} = \sum_{x_A} P_{\text{cer}}(x_A) P(x_A),$$ (3.41)

where $P(x_A)$ is the a priori probability that $x_A$ was transmitted and $P_{\text{cer}}(x_A)$ is the conditional error rate (CER) that $x_A$ is not detected by the RD-MLS given that $x_A$ was transmitted. Recalling that the final output chosen from
the list $\mathcal{L}$ is denoted as $\tilde{x}^{\text{ml}}$, $P_{\text{cer}}(x_A)$ becomes

$$P_{\text{cer}}(x_A) = Pr_A(\tilde{x}^{\text{ml}} \neq x_A) \tag{3.42}$$

$$= Pr_A(\tilde{x}^{\text{ml}} \neq x_A \mid x_A \in \mathcal{L}) Pr_A(x_A \in \mathcal{L})$$

$$+ Pr_A(\tilde{x}^{\text{ml}} \neq x_A \mid x_A \notin \mathcal{L}) Pr_A(x_A \notin \mathcal{L}), \tag{3.43}$$

where $Pr_A(B)$ refers to the conditional probability of event $B$ under the condition that $x_A$ was transmitted. Let $x^{\text{ml}}$ be an exact ML solution. We denote $P_e^{\text{ML}}(x_A) = Pr_A(x^{\text{ml}} \neq x_A)$, which is the ML detection error probability given $x_A$ was sent. Then, we have

$$Pr_A(\tilde{x}^{\text{ml}} \neq x_A \mid x_A \in \mathcal{L})$$

$$\leq Pr_A(x^{\text{ml}} \neq x_A \mid x_A \in \mathcal{L}) \tag{3.44}$$

$$\leq \frac{Pr_A(x^{\text{ml}} \neq x_A, x_A \in \mathcal{L}) + Pr_A(x^{\text{ml}} \neq x_A, x_A \notin \mathcal{L})}{Pr(x_A \in \mathcal{L})} \tag{3.45}$$

$$= \frac{P_e^{\text{ML}}(x_A)}{Pr(x_A \in \mathcal{L})}, \tag{3.46}$$

where (3.44) follows from the fact that the RD-MLS chooses a closest point from $\mathcal{L}$ while the ML detector does from all candidate points. From (3.46) and the fact that $Pr_A(\tilde{x}^{\text{ml}} \neq x_A \mid x_A \notin \mathcal{L}) = 1$, (3.43) becomes

$$P_{\text{cer}}(x_A) \leq P_e^{\text{ML}}(x_A) + Pr_A(x_A \notin \mathcal{L}). \tag{3.47}$$

The second term in the right-hand side of (3.47) illustrates the primary source of sub-optimality of the RD-MLS. Following symbol partitioning, let $x_A$ be divided into $x_{A1}$ and $x_{A2}$. Let the corresponding partitioned candidate sets be $\mathcal{L}_1 = \{\hat{x}^1_1, \cdots, \hat{x}^1_K\}$ and $\mathcal{L}_2 = \{\hat{x}^2_1, \cdots, \hat{x}^2_K\}$. We can show that (3.47) becomes

$$P_{\text{cer}}(x_A) \leq P_e^{\text{ML}}(x_A) + Pr_A(x_A \notin \mathcal{L} \mid x_{A1} \notin \mathcal{L}_1) Pr_{A}(x_{A1} \notin \mathcal{L}_1)$$

$$+ Pr_A(x_A \notin \mathcal{L} \mid x_{A1} \in \mathcal{L}_1) Pr_{A}(x_{A1} \in \mathcal{L}_1) \tag{3.48}$$

$$\leq P_e^{\text{ML}}(x_A) + Pr_A(x_{A1} \notin \mathcal{L}_1) + Pr_A(x_{A2} \notin \mathcal{L}_2 \mid x_{A1} \in \mathcal{L}_1). \tag{3.49}$$
where (3.49) follows from $Pr_A (x_A \notin L | x_{A1} \in L_1) = Pr_A (x_{A2} \notin L_2 | x_{A1} \in L_1)$ and $Pr_A (x_A \notin L | x_{A1} \notin L_1) = 1$. The second and third terms on the right-hand side in (3.49) are the probability that the candidate list found by the LTS does not contain $x_{A1}$ and the conditional probability that the MMSE-DF estimation makes a decision error given the LTS yielded $x_{A1}$ in $L_1$, respectively. We denote these terms as $P_{eLTS}$ and $P_{eMMSE-DF}$, respectively. From (3.41) and (3.49), an upper bound of $P_{err}$ is written

$$P_{err} \leq P_{eML} + \sum_{x_{A1}} \sum_{x_{A2}} P_{R_{A1,A2}} (x_{A1} \notin L_1) P (x_{A1}, x_{A2}) + \sum_{x_A} P_{R_A} (x_{A2} \notin L_2 | x_{A1} \in L_1) P (x_A)$$

$$= P_{eML} + \sum_{x_{A1}} P_{R_{A1}} (x_{A1} \notin L_1) P (x_{A1}) + \sum_{x_A} P_{R_A} (x_{A2} \notin L_2 | x_{A1} \in L_1) P (x_A),$$

(3.50)

where $P_{R_{A1}} (\cdot)$, $P (x_{A1})$, and $P (x_{A1}, x_{A2})$ are defined similarly as $P_{R_A} (\cdot)$ and $P (x_A)$.

To investigate the asymptotic behavior of $P_{eLTS}$, we consider the input to the LTS in (3.27), $z = Gx_{A1} + \hat{w}$. We remind the reader that $G = ZH_1$, $\hat{w} = w + e$ and $e = Z (H_2x_2 + w) - w$. Using these, we can write $z$ by

$$z = Gx_{A1} + ZH_2x_2 + Z\hat{w}.\quad (3.52)$$

Though $x_2$ is discrete in $\mathcal{F}^{n_2}$, we approximate the LMMSE estimation error plus noise, $\hat{w} (= w + e)$ as a complex Gaussian vector with zero mean and the covariance matrix $\Phi = E[(\hat{w})(\hat{w})^H] = \sigma_w^4 (H_2H_2^H + \sigma_w^2 I)^{-1}$. For large SNR, we note that $ZH_2$ becomes zero matrix while $Z$ does not vanish as $\sigma_w^2 \to 0$ (see Appendix B). From (3.52) and also using this result, $\hat{w}$ approximates to a Gaussian distribution as SNR increases. Asymptotic normality of this type of interference plus noise term for MMSE detection was also validated in [47] and [48].

In the LTS with the stopping parameter $m$ and using the Gaussian approx-
imation, we show in Appendix C that $Pr_{A_1}(x_{A_1} \notin L_1)$ is upper-bounded by

$$Pr_{A_1}(x_{A_1} \notin L_1) \leq cE_{H_2} \left[ \prod_{i=1}^{n_r-P} \frac{\lambda_i}{\lambda_i^2 + \frac{m}{\lambda_i}} \prod_{i=n_r-P+1}^{n_r} \frac{\lambda_i}{\lambda_i^2 + \frac{m}{\lambda_i}} \right], \quad (3.53)$$

where $\lambda$ is the normalization factor of the QAM modulation, $\lambda_1 \geq \cdots \geq \lambda_{n_r}$ are the eigenvalues of $(H_2H_2^H + \sigma_w^2I)^{-1}$, and $c = M^{n_1} - 1$. The notation $E_{H_2}[\cdot]$ implies an expectation taken over $H_2$. We consider the high SNR regime of $(3.53)$ as $\sigma_w^2 \to 0$. If we let $P(\leq n_2)$ be the rank of $H_2$, it follows that $\lambda_1 = \cdots = \lambda_{n_r-P} = 1/\sigma_w^2$ and $\lambda_i = 1/(q_{n_r-i+1} + \sigma_w^2)$ for $n_r - P + 1 \leq i \leq n_r$, where $q_1, \cdots, q_P$ are the (non-zero) eigenvalues of $H_2H_2^H$. Then, we have

$$Pr_{A_1}(x_{A_1} \notin L_1) \leq cE_{H_2} \left[ \prod_{i=1}^{n_r-P} \frac{\lambda_i}{\lambda_i^2 + \frac{m}{\lambda_i}} \prod_{i=n_r-P+1}^{n_r} \frac{\lambda_i}{\lambda_i^2 + \frac{m}{\lambda_i}} \right], \quad (3.54)$$

$$\leq cE_{H_2} \left[ \prod_{i=1}^{n_r-n_2} \frac{\lambda_i}{\lambda_i^2 + \frac{m}{\lambda_i}} \right], \quad (3.55)$$

$$\leq c \prod_{i=1}^{n_r-n_2} \left( \frac{\sigma_w^2}{\sigma_w^2 + \frac{m}{\lambda_i}} \right), \quad (3.56)$$

$$= c \left( \frac{1}{1 + \frac{m}{\lambda_i \sigma_w^2}} \right)^{n_r-n_2}, \quad (3.57)$$

$$\leq c \left( \frac{\lambda_2^2 \sigma_w^2}{m} \right)^{n_r-n_2}, \quad (3.58)$$

where the inequality in $(3.55)$ gets tighter as $\sigma_w^2 \to 0$ since $\frac{\lambda_1}{\sigma_w^2}, \cdots, \frac{\lambda_{n_r}}{\sigma_w^2} \to \infty$ and $(3.56)$ holds because $\lambda_1 = \cdots = \lambda_{n_r-n_2} = \frac{1}{\sigma_w^2}$ regardless of realizations of $H_2$. Since this upper bound does not depend on $x_{A_1}$, it becomes an upper bound of $P_{e,\text{LTS}}$.

From $(3.58)$, we can provide some important observations on the asymptotic behavior of the RD-MLS. First, we observe that the upper bound is a decreasing function of the list stopping parameter $m$. This follows our intuition as, when $m$ increases, the number of candidates in the list also increases, and the effect of additional loss decreases. Since $m$ multiplies the ratio of the minimum squared distance $4/\lambda^2$ to the noise power $\sigma_w^2$, it provides additional effective SNR gains over the single candidate search. Second, we observe that the diversity gain $(= \lim_{\text{SNR} \to \infty} \frac{\log BER}{\log \text{SNR}})$ of the term related
Table 3.1: Complexity of RD-MLS algorithm.

<table>
<thead>
<tr>
<th>Block</th>
<th># of real additions</th>
<th># of real multiplications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension reduction step</td>
<td>4</td>
<td>2n_r^2 - 2n_t</td>
</tr>
<tr>
<td>Postprocessing step</td>
<td>K(8n_r - 4n_t - 4)n_t - K(2n_t - 2n_r)</td>
<td>K(8n_r - 4n_t - 4)n_t</td>
</tr>
<tr>
<td>N_k for the K-LSA</td>
<td>2n_r - k + 7</td>
<td>2n_r - k + 5</td>
</tr>
</tbody>
</table>

to \( P_e^{\text{LTS}} \) is larger than or equal to \( n_r - n_2 (= n_r - n_t + n_1) \).

We next take a look at the probability \( P_e^{\text{MMSE-DF}} \). This is the probability that the MMSE-DF estimation makes a decision error for the system \( H_2x_2 + w \) resulting from perfect cancellation of \( x_{A1} \) in (3.35). It is well known that the MMSE-DF detector for the \( n_r \)-by-\( n_2 \) system achieves a diversity of \( n_r - n_2 + 1 \) both with and without the ordering based on post-detection SNR [11,49].

Based on (3.51), at high SNR, the diversity gain would be dominated by \( n_r + n_1 - n_t \), which is the minimum of those for \( P_e^{\text{ML}} \), \( P_e^{\text{LTS}} \), and \( P_e^{\text{MMSE-DF}} \), i.e., \( \min(n_r, n_r + n_1 - n_t, n_r + n_1 - n_t + 1) \). In fact, this diversity gain is the same as that achieved by the scheme employing a single candidate search [35]. However, owing to the performance gains offered by the LTS, performance loss from the ML search becomes moderate over a wide range of operational regimes. This will be shown in the simulation results in Section 3.5.

3.4.2 Comments on Complexity

The overall complexity of the RD-MLS algorithm comprises those of (1) the dimension reduction operator, (2) the tree search, and (3) the postprocessing operations. In our analysis, the complexity is evaluated by the number of floating-point operations (FLOPS) counted per channel use. The number of FLOPS \( N \) needed for the tree search operation is expressed as

\[
N = \sum_{k=1}^{n_r'} \sum_{X_r'^{n_i}} N_k I \left( X_r'^{n_i} \right),
\]

where \( I \left( X_r'^{n_i} \right) \) evaluates to 1 if the path \( X_r'^{n_i} \) is visited, and 0 otherwise. \( N_k \) is the number of FLOPS per node at tree level \( k \). The total number of FLOPS depends on the number of nodes visited. Table 3.1 shows the number
of FLOPS in the dimension reduction and postprocessing steps and per node computations of the $K$-LSA. The FLOPS required for the preprocessing steps are derived from the calculation of $\mathbf{z} = \mathbf{Z}\mathbf{y}$ while those for the postprocessing step are derived from the calculation of $\mathbf{y}_i = \mathbf{y} - \mathbf{H}_i\hat{\mathbf{x}}_i$ in (3.35) and (3.37). The derivation of $N_k$ is obtained by inspecting the computation of the residual signal $y'_i - \sum_{l=1}^{n_1} r_{i,l}x_l$ and path metric update as in (3.9). From this and the node visitation information, averaged numbers of FLOPS (per channel use) for the RD-MLS detector can be measured.

### 3.5 Simulations

In this section, we observe the complexity and performance of the proposed RD-MLS as compared to that of the full-dimensional SD and other near-ML detectors. The simulation setup is based on $M$-QAM transmission over MIMO systems in quasi-static Rayleigh fading channels where the elements of $\mathbf{H}$ are modeled by independent complex Gaussian random variables. As a measure of the performance and complexity, bit error rate (BER) and the average number of FLOPS are used. To obtain the average number of FLOPS through simulations, the average number of real additions is counted per signal transmission. In the simulation, the following algorithms are compared:

1. The state-of-the-art full dimensional SD (FD-SD) algorithm: *Algorithm II* in [6], which guarantees exact ML performance.

2. RD-MLS $(n_1, m)$: $K$-LSA with the pruning parameter $P_{\text{prun}} = 0.2$. $(n_1, m)$ indicates the parameter sets of RD-MLS; $n_1$ is the reduced dimension and $m$ is the stopping parameter defined in Section 3.3.3.

3. B-Chase ($l$) detector of [39]: $l$ is the parameter specifying the number of first layer symbols that are picked up to generate the candidate list.

4. Probabilistic tree pruning (PTP) sphere decoder ($P_e$) [29]: $P_e$ indicates the pruning probability that controls the radius of sphere search.

5. $K$-best sphere decoder ($K$) [27]: $K$ indicates the maximum number of candidates picked up every tree level. We set the parameter $\gamma$ to 5 when SNR $< 25$ dB and to 10 when SNR $\geq 25$ dB as suggested in [27].
We do not consider the dimension reduction methods in [34, 35] in our comparison due to their performance gap from ML detection. We instead consider the B-Chase \((l)\) detector, which contains several partitioned search detectors [37, 38] as a special case. In addition, we include two low-complexity sphere decoding algorithms [27, 29]. Assuming that the block fading length is large, the complexities of preprocessing such as QR decomposition or STEP 2 in [39] are ignored since they can be shared in the block. In the RD-MLS setup, we set the maximum candidate size of the list to 16 based on intensive simulations.

First, we consider 16-QAM transmission for \(8 \times 8\) MIMO systems, where \((n_1, m)\) parameters of the RD-MLS are set to \((5, 2)\) and \((5, 3)\). In Figure 3.4, we provide the BER results and average FLOPS over the SNR range between 13 dB and 23 dB. As shown in Figure 3.4, the RD-MLS algorithm exhibits far lower complexity than the full dimensional SD while maintaining performance within 1 dB for most of the SNR range of interest. In particular, at 19 dB of SNR, the RD-MLS achieves up to 25\% and 55\% complexity reduction for \((n_1, m) = (5, 3)\) and \((n_1, m) = (5, 2)\) over the FD-SD. It is also clear that the RD-MLS has the best performance/complexity trade-off among all low-complexity detection schemes considered. In particular, RD-MLS shows a clear complexity benefit in the mid and high SNR regime since the stopping criterion prevents the LTS from finding unnecessarily many candidates. In Figure 3.5 we provide the results of \(10 \times 10\) systems with two configurations \((n_1, m) = (6, 4)\) and \((n_1, m) = (7, 4)\) of RD-MLS. Overall, we observe that the complexity gain of RD-MLS over the FD-SD improves dramatically compared to the \(8 \times 8\) case. The complexity of RD-MLS is lower than those of the PTP-SD \((P_e = 0.8)\) and \(K\)-best SD \((K = 12)\) algorithms. Although the B-Chase \((l = 16)\) achieves lower complexity than RD-MLS in this scenario, it suffers substantial performance loss from the ML detector. In contrast, the performance loss of the RD-MLS is maintained within 1 dB.

In Figure 3.6, the performance and complexity curves for a \(6 \times 6\) MIMO system with 64-QAM transmission are provided. In the RD-MLS, two configurations \((n_1, m) = (4, 2), (4, 3)\) are considered. In general, we observe that the performance loss of the RD-MLS over exact ML performance is within 1 dB in the BER range of \(10^{-1}\) to \(10^{-4}\). For this high-order modulation format, we observe that the RD-MLS algorithm achieves a substantial reduction in complexity over the full-dimensional SD while performing close to the ML
Figure 3.4: Plots of average BER and FLOPS of the $8 \times 8$ 16-QAM system.
Figure 3.5: Plots of average BER and FLOPS of the $10 \times 10$ 16-QAM system.
Figure 3.6: Plots of average BER and FLOPS of the $6 \times 6$ 64-QAM system.
detector over the SNR range of interest. The increased complexity reduction as compared to the 16-QAM case matches our intuition that the search complexity would be scaled by a factor of $M^{n_1-n_1}$ due to the reduction in the search space dimension.

In Figure 3.7, we test how the stopping criterion adjusts the size of the candidate list for different SNRs. In order to observe this, the average size of the candidate list collected by the $K$-LSA for each transmission is measured. A 6×6 16-QAM MIMO system is employed with the parameters $(n_1, m)$ of the RD-MLS set to (4, 4). The stopping criterion exploits the property that only a small fraction of candidates are necessary to maintain good performance at high SNR. In fact, we observe that the average size of the candidate list decreases with SNR.

Figure 3.8 shows how complexity and performance are traded off through the parameter set $m$ and $n_1$. We consider several combinations of $m$ and $n_1$ in a 6×6 system with 64-QAM modulation. From the BER and FLOPS curves in Figure 3.8, we observe a clear tradeoff between complexity and performance. As $n_1$ or $m$ increases, the performance gap from the full dimensional SD decreases at the expense of complexity increase. This demonstrates how
Figure 3.8: The complexity and performance of RD-MLS for several configurations of \((n_1, m)\). The 6 \times 6 64-QAM case is considered.
the RD-MLS can bridge the gap between the high complexity ML detector and a linear sub-optimal detector.

3.6 Conclusions

A low-complexity near-ML detection technique referred to as RD-MLS detection is presented in this chapter. In addition to dimensionality reduction, which directly impacts complexity, two main ideas are proposed for mitigating the performance loss incurred. First, detection in the reduced dimension system is modified from an ML search to a list tree search. Therefore, instead of detecting the one best symbol, multiple candidates are found in the LTS stage, enabling errors introduced by the dimension reduction step to be mitigated. Second, for each of the symbol candidates found by the LTS, the rest of the symbols are estimated via an MMSE-DF algorithm. After concatenating two symbol estimates, the final output is chosen as the unique minimizer of the ML cost function. We have found, from asymptotic performance analysis, that performance gains that increase the effective SNR can be achieved by the LTS. We observe from simulations that the BER performance of the RD-MLS represents a good balance between the performance of the full dimensional SD and the complexity of linear receivers.
CHAPTER 4

EFFICIENT SOFT-INPUT SOFT-OUTPUT TREE DETECTION VIA AN IMPROVED PATH METRIC

4.1 Introduction

Recall from Chapter 3 that the relationship between the transmitted symbol and the received signal vector in many communication systems can be expressed in the form

\[ y_o = Hx + n_o, \]  

(4.1)

where \( x \) is the transmitted vector whose entries are chosen from a finite symbol alphabet, \( y_o \) and \( n_o \) are the received signal and noise vectors, respectively, and \( H \) is a channel matrix. As a practical decoding scheme when a code constraint is imposed, the iterative detection and decoding (IDD) method, has received much attention [13]. Motivated by the turbo principle [3], a IDD receiver exchanges soft information between a symbol detector and a channel decoder to achieve performance close to the channel capacity. The symbol detector exchanges soft information with the channel decoder in the form of a posteriori probabilities (APP) on the bits comprising \( x \), using a priori probabilities provided by the channel decoder and the observation \( y \). In the sequel, we refer to this detector as an APP detector.

Direct computation of the APP involves marginalization over all configurations of the vector \( x \), leading to an exponential complexity in the system size (number of antenna elements). As means of approximately performing APP detection at reduced complexity, tree detection techniques have received much attention [12,13,42,50–55]. The essence of these approaches is to produce a set of promising symbol candidates via a tree search for estimating the APP over this reduced set. Thus far, a variety of tree detection algorithms have been proposed. In [13], the sphere decoding algorithm (SDA) [6,9] with a fixed radius was used to find symbol candidates. In [50], a priori information obtained from the channel decoder was exploited to improve the search
efficiency of the SDA. In [51], a hard sphere decoder was employed to find a single maximum a posteriori probability (MAP) symbol estimate maximizing $P(x|y)$ and a candidate list was generated by flipping bits in the MAP estimate. In [53], the APPs of all bits in $x$ are obtained simultaneously by a modified bound tightening of a single sphere search. Additionally, a more sophisticated extension of this idea was introduced in [52]. The computational complexity of these tree detection algorithms varies depending on the channel and noise realizations so that the worst case complexity is the same as that of exhaustive search. On the other hand, tree detection algorithms limiting the worst complexity via a fixed-complexity tree search have also been proposed. In [54], an $M$-algorithm was employed to find a fixed number of candidates and in [12], the stack algorithm was exploited for list generation in combination with soft augmentation of tail bits of stack elements. Other fixed-complexity tree detection algorithms also exist [42,55].

The $M$-algorithm [31], which stores the $M$ best candidates for each symbol layer in the decoding tree, is an attractive method for soft-input soft-output detection due to its parallel and pipelined structure. In spite of this benefit, the $M$-algorithm suffers from a poor performance-complexity trade-off due to the greedy nature of the algorithm. That is, the algorithm checks the validity of paths in a forward direction but never traverses back for reconsideration. Once a correct path is rejected, it will never be selected again in subsequent selections, resulting in a considerable amount of wasteful search effort. Moreover, these erroneous decisions often occur in early candidate selection stages where the accumulated path metric considers only a few symbol spans, thereby increasing the chance of rejecting promising paths.

In this chapter, we put forth a new tree detection algorithm, referred to as the improved soft-in soft-out $M$-algorithm (ISS-MA), which enhances the sorting process of the $M$-algorithm by employing a path metric deliberately designed to capture the contribution of the entire symbol path. While the conventional path metric accounts for the contributions of symbols along the visited path only, the path metric of the ISS-MA looks ahead to the unvisited paths and estimates their contributions through a soft unconstrained linear symbol estimate. Towards this end, a bias term reflecting the information from as-yet undecided symbols is incorporated into the conventional path metric. In order to distinguish this improved path metric from the conventional path metric and other look-ahead path metrics, we refer to it as
a linear estimate-based look-ahead (LE-LA) path metric. By sorting paths based on the LE-LA path metric, the ISS-MA lessens the chance of rejecting the correct path from candidate list, thereby improving the performance of soft-input soft-output detection. Indeed, from an analysis of the probability of correct path loss (CPL), we show that an upper bound on the CPL probability of the ISS-MA is strictly smaller than that of the conventional path metric.

The idea of look-ahead path metric has been explored in artificial intelligence search problems [56] and can also be found in soft decoding of linear block codes [57]. While these approaches search for a legitimate bias term that guarantees the optimality of the sequential search, our approach exploits a linear estimator to obtain a useful bias term at low complexity. It is worth emphasizing the difference between the proposed path metric and the Fano metric, which exploits the a posteriori probability of each path as its path metric [58]. For a binary symmetric channel, the Fano metric introduces a bias term proportional to the path length to penalize paths of short length. The extension of the Fano metric to channels with memory or MIMO channels is not straightforward, since it involves marginalization over the distribution of the undecided symbols. Modification of the Fano metric is considered for equalization of intersymbol interference (ISI) channels in [59] and for multi-input multi-output detection in [28–30]. In [12], the probability density of an observed signal is used as a bias term and a separate tree search is employed to find a proper bias term. While these approaches assign an equal bias term for paths of the same length, the proposed ISS-MA provides a distinct bias term for each path in the tree, allowing for the application of a breadth-first search, such as the $M$-algorithm. Henceforth, our path metric can be readily combined with any tree-based soft-input soft-output detection algorithm.

The rest of this chapter is organized as follows. In Section 4.2, we briefly introduce the IDD system and describe the tree detection algorithm. In Section 4.3, we present the LE-LA path metric along with its efficient computation. We employ the LE-LA path metric to improve the performance of the soft-input soft-output $M$-algorithm. In Section 4.4, we present the performance analysis for the ISS-MA. In Section 4.5, we provide simulation results and conclude in Section 4.6.
4.2 Problem Description

In this section, we briefly review the iterative detection and decoding framework, and then we introduce the approach taken in many soft-input soft-output tree detection algorithms.

4.2.1 Iterative Detection and Decoding (IDD)

In a transmitter, a rate $R_c$ convolutional encoder is used to convert a sequence of i.i.d. binary information bits $\{b_i\}$ to an encoded sequence $\{c_i\}$. The bit sequence $\{c_i\}$ is permuted using a random interleaver, $\Pi$ and then mapped into a symbol vector using a $2^Q$-ary quadrature amplitude modulation (QAM) symbol alphabet. We label the interleaved bits associated with the $k$th symbol $x_k$ by $\bar{c}_{k,1}, \cdots, \bar{c}_{k,Q}$. Due to the interleaver, we can assume that these interleaved bits are mutually uncorrelated.

In the system model (5.2), $y_0$ and $n_0$ are the $L \times 1$ received signal and noise vectors, respectively, and $H$ is an $L \times N$ channel matrix. Each entry of the $N \times 1$ symbol vector $x$ is drawn from a finite alphabet

$$\mathcal{F} = \{ x_r + j x_i \mid x_r, x_i \in \left\{ -\frac{\sqrt{M} + 1}{\lambda}, -\frac{\sqrt{M} + 3}{\lambda}, \cdots, \frac{\sqrt{M} - 3}{\lambda}, \frac{\sqrt{M} - 1}{\lambda} \right\} \},$$

(4.2)

where $\lambda$ is chosen to satisfy the normalization condition $E[xx^H] = I_n$. For example, $\lambda = \sqrt{10}$ for 16-QAM and $\lambda = \sqrt{42}$ for 64-QAM modulation, respectively.

Figure 4.1 depicts the basic structure of an IDD system. The receiver consists of two main blocks: the APP detector and the channel decoder. The APP detector generates the a posteriori log-likelihood ratio (LLR) of $\bar{c}_{k,i}$ using the observation $y_o$ and a priori information delivered from the
channel decoder. The \textit{a posteriori} LLR is defined as

$$L_{\text{post}}(c_{k,i}) = \ln \frac{Pr(c_{k,i} = +1 | y)}{Pr(c_{k,i} = -1 | y_o)},$$

(4.3)

where we take $c_{k,i} \in \{1, -1\}$ rather than $\{1, 0\}$ by convention. With the standard noise model $n_o \sim \mathcal{CN}(0, \sigma^2_n I)$, (4.3) can be rewritten

$$L_{\text{post}}(c_{k,i}) = \ln \frac{\sum_{x \in X_{k,i}^+} \exp (\lambda (x))}{\sum_{x \in X_{k,i}^-} \exp (\lambda (x))},$$

(4.4)

where

$$\lambda(x) = -\frac{1}{\sigma^2_n} ||y - Hx||^2 + \sum_{i=1}^{N} \sum_{j=1}^{Q} \ln Pr(\bar{c}_{i,j}),$$

$$Pr(\bar{c}_{i,j}) = \frac{1}{2} \left( 1 + \bar{c}_{i,j} \tanh \left( \frac{L_{\text{pri}}(\bar{c}_{i,j})}{2} \right) \right).$$

(4.5)

The set $X_{k,i}^+$ is the set of all configurations of the vector $x$ satisfying $\bar{c}_{k,i} = +1$ ($X_{k,i}^-$ is defined similarly), and $L_{\text{pri}}(\bar{c}_{k,i})$ is the \textit{a priori} LLR defined as $L_{\text{pri}}(\bar{c}_{k,i}) = \ln Pr(c_{k,i} = +1) - \ln Pr(c_{k,i} = -1)$. Once $L_{\text{post}}(\bar{c}_{k,i})$ is computed, the extrinsic LLR is obtained from $L_{\text{ext}}(\bar{c}_{k,i}) = L_{\text{post}}(\bar{c}_{k,i}) - L_{\text{pri}}(\bar{c}_{k,i})$. These extrinsic LLRs are de-interleaved and then delivered to the channel decoder. The channel decoder computes the extrinsic LLR for the coded bits $\{c_i\}$ and feeds them back to the APP detector. These operations are repeated until a suitably chosen convergence criterion is achieved [13].

4.2.2 Soft-Input Soft-Output Tree Detection

The direct computation of the \textit{a posteriori} LLR in (4.4) involves marginalization over $2^{NQ}$ symbol candidates, which easily becomes infeasible for large systems employing high order modulations. A tree detection algorithm addresses this problem by searching a small set of promising symbol candidates over which \textit{a posteriori} LLRs are estimated. Specifically, a small number of symbol vectors with large $\lambda(x)$, equivalently, small $-\sigma^2_n \lambda(x)$ are sought. In the sequel, we refer to $d_{\text{APP}}(x) = -\sigma^2_n \lambda(x)$ as a \textit{cost metric} for tree detection. The goal of the tree detection algorithm is to find symbol vectors of small cost metric and the best among them corresponds to the \textit{maximum a
posteriori (MAP) solution (denoted by $\mathbf{x}_{\text{MAP}}$).

The tree detection algorithm relies on the tree representation of the search space spanned by $\mathbf{x} = (x_1, \cdots, x_N) \in \mathcal{F}^N$. Tree construction is performed from the root node as follows. First, representing the symbol realizations for $x_N$, we extend $2^Q$ branches from the root (recall that we assume $2^Q$-ary QAM modulation). For each such branch, $2^Q$ child branches are extended for the possible realization of the next symbol $x_{N-1}$. These branch extensions are repeated until all branches corresponding to $x_N, \cdots, x_1$ are generated. This yields a tree of the depth $N$, where each “complete” path from the root to a leaf corresponds to a realization of $\mathbf{x}$. In order to find the complete paths of small cost metric, the tree detection algorithm searches the tree using a systematic node visiting rule. For notational simplicity, we henceforth denote a path associated with a set of symbols $x_i, \cdots, x_j, (i < j)$ by a column vector $\mathbf{x}_{ij}^T = [x_i, \cdots, x_j]^T$. Also, we call a level of tree associated with the symbol $x_i$ “the $i$th level” (e.g., the bottom level associated with $x_1$ is the first level).

For a systematic search of symbol candidates, a path metric is assigned to each path $\mathbf{x}_{ij}^N$. Towards this end, a QR decomposition of $\mathbf{H}$ is performed

$$\mathbf{H} = \mathbf{Q} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = [\mathbf{Q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix},$$

where $\mathbf{R}$ has an $N \times N$ upper-triangular matrix whose diagonals are non-negative and $\mathbf{Q}$ is an $L \times N$ unitary matrix. Using the invariance of the norm to unitary transformation, we can express $d_{\text{APP}}(\mathbf{x})$ as

$$d_{\text{APP}}(\mathbf{x}) = \|\mathbf{y}_o - \mathbf{H}\mathbf{x}\|^2 - \sigma_n^2 \sum_{i=1}^N \sum_{j=1}^Q \ln Pr(\bar{c}_{i,j})$$

$$= \|\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 - \sigma_n^2 \sum_{i=1}^N \sum_{j=1}^Q \ln Pr(\bar{c}_{i,j})$$

$$= \sum_{i=1}^N b(x_{ij}^N) + C,$$

where $b(x_{ij}^N) = \left| y_i - \sum_{j=1}^N r_{i,j} x_j \right|^2 - \sigma_n^2 \sum_{i=1}^Q \ln Pr(\bar{c}_{k,i})$, and $\mathbf{y} = [y_1, \cdots, y_N]^T = \mathbf{Q}_1^H \mathbf{y}_o$ and $C = \|\mathbf{Q}_2^H \mathbf{y}\|^2$. The path metric associated with the path $\mathbf{x}_{ij}^N$ can
be defined as a partial sum in the cost metric \([6,7]\)
\[
p^{(c)}(x^N_k) = \sum_{i=k}^{N} b(x^N_i).
\] (4.10)

Whenever a new node is visited, the term \(b(x^N_i)\), referred to as a branch metric, is added to the path metric of the parent node. Since the branch metric is non-negative for all \(i\), the path metric \(p^{(c)}(x^N_k)\) becomes a lower bound of the cost metric \(d_{\text{APP}}(x)\). Using \(p^{(c)}(x^N_k)\), the tree detection algorithm compares the reliability of distinctive paths and chooses the surviving paths. Since the path metric is determined by the visited path, we henceforth denote \(p^{(c)}(x^N_k)\) as a causal path metric.

According to predefined node visiting rule \([30]\), the tree detection algorithm finds the complete paths associated with smallest cost metric. Denoting the set of the corresponding symbol candidates as \(L\), an approximate APP can be expressed as
\[
L_{\text{post}}(\bar{c}_{k,i}) \approx \ln \frac{\sum_{x \in L \cap X^{k+1}_{k,i}} \exp(\lambda(x))}{\sum_{x \in L \cap X^{k-1}_{k,i}} \exp(\lambda(x))}.
\] (4.11)

Further simplification can be achieved using max-log approximation \([60]\)
\[
L_{\text{post}}(\bar{c}_{k,i}) \approx \left( \max_{x \in L \cap X^{k+1}_{k,i}} \lambda(x) - \max_{x \in L \cap X^{k-1}_{k,i}} \lambda(x) \right).
\] (4.12)

4.3 Improved Soft-Input Soft-Output \(M\)-Algorithm (ISS-MA)

In this section, we present the ISS-MA which improves the candidate selection process of soft-input soft-output tree detection algorithms. We first describe a genie-aided path metric that motivates our approach to improve the conventional path metric and then we introduce the new path metric that accounts for the information on the unvisited paths using an unconstrained linear estimate of the non-causal symbols. In order to minimize the complexity overhead due to the new path metric, we investigate methods to compute it efficiently.
4.3.1 Motivation

We begin our discussion with the following path metric:

**Definition 4.3.1.** A genie-aided path metric \( p^{(g)}(x_k^N) \) is defined as

\[
p^{(g)}(x_k^N) = p^{(e)}(x_k^N) + \min_{x_1^{k-1}} \left( \sum_{i=1}^{k-1} b(x_i^N) \right).
\]

(4.13)

The genie-aided path metric is obtained by minimizing the sum of \( b(x_i^N)(1 \leq i \leq k - 1) \) over all combinations of undecided symbols \( x_1^{k-1} \). This minimal term, which can be considered as a bias term, is added into the causal path metric.

**Lemma 4.3.2.** The modified \( M \)-algorithm employing the genie-aided path metric finds the closest (best) path only with minimal number of node visitations (i.e., with \( M = 1 \)).

This lemma can be readily shown using that the genie-aided path metric provides the smallest cost metric among all tail paths. Note that using the genie-aided path metric, the best path can be found with probability one for an arbitrary \( M \) value.

**Theorem 4.3.3.** Given the actual transmitted symbol vector \( x_k^N \) (i.e., \( x_k^N = \tilde{x}_k^N \)), the bias term of the genie-aided path metric is

\[
\min_{x_1^{k-1}} \left( \sum_{i=1}^{k-1} b(x_i^N) \right) = \sum_{i=1}^{k-1} b(x_i^N) \bigg|_{x_1^{k-1} = \hat{x}_1^{k-1}},
\]

(4.14)

where the minimizer \( \hat{x}_1^{k-1} \) is the MAP estimate of \( x_1^{k-1} \), i.e.,

\[
\hat{x}_1^{k-1} = \arg \max_{x_1^{k-1}} \ln Pr \left( x_1^{k-1} \Big| y, x_k^N = \tilde{x}_k^N \right).
\]

(4.15)

**Proof.** See Appendix D.

**Theorem 3.3** implies that the bias term of the genie-aided path metric is obtained by computing \( \sum_{i=1}^{k-1} b(x_i^N) \) using the MAP estimate of \( x_1^{k-1} \). This MAP estimate is derived under the condition that the path associated with the actual transmitted symbols, \( \tilde{x}_k^N \) is given. Though the genie-aided path
metric offers a substantial performance gain, it is impractical due to the high complexity of obtaining the MAP estimate in the bias term.

4.3.2 Derivation of LE-LA Path Metric

In order to alleviate the complexity associated with MAP detection of $x_{k-1}^1$ in the genie-aided path metric, we relax the finite alphabet constraint of $x_{k-1}^1$ and then replace the MAP estimate by the linear MMSE estimate $\hat{x}_{k-1}^1$. Note that when $\hat{x}_{k-1}^1$ is assumed to be Gaussian, the MAP estimate is identical to the linear MMSE solution [61]. For a particular path visited $x_k^N$, we first define the LE-LA path metric.

**Definition 4.3.4.** The linear estimate-based look-ahead path metric, denoted by $p^{(l)} (x_k^N)$, is defined as

$$p^{(l)} (x_k^N) \triangleq p^{(c)} (x_k^N) + \sum_{j=1}^{k-1} b (x_j^N) \bigg|_{x_{k-1}^1 = \hat{x}_{k-1}^1}, \quad (4.16)$$

where $\hat{x}_{k-1}^1$ is the linear MMSE estimate of $x_{k-1}^1$.

Note that $\hat{x}_{k-1}^1$ is obtained under the condition that $x_k^N = \tilde{x}_k^N$. In the sequel, we will denote this bias term $p^{(b)} (x_k^N)$.

To derive the linear MMSE estimate, we partition the vectors $y$ and $n$ to $(k - 1) \times 1$ and $(N - k + 1) \times 1$ vectors, i.e.,

$$y = \begin{bmatrix} y_{k-1}^1 \\ y_k^N \end{bmatrix} = \begin{bmatrix} R_{11,k} & R_{12,k} \\ 0 & R_{22,k} \end{bmatrix} \begin{bmatrix} x_{k-1}^1 \\ x_k^N \end{bmatrix} + \begin{bmatrix} n_{k-1}^1 \\ n_k^N \end{bmatrix},$$

(4.17)

where $R_{11,k}$, $R_{12,k}$, and $R_{22,k}$ are the adequately partitioned sub-matrices of $R$. Using (4.17), $p^{(l)} (x_k^N)$ can be expressed as

$$p^{(l)} (x_k^N) = p^{(c)} (x_k^N) + p^{(b)} (x_k^N), \quad (4.18)$$

where

$$p^{(c)} (x_k^N) = \| y_k^N - R_{22,k} x_k^N \|^2 + \xi (x_k^N),$$

(4.19)

$$p^{(b)} (x_k^N) = \| y_{k-1}^1 - R_{11,k} \hat{x}_{k-1}^1 - R_{12,k} x_k^N \|^2,$$

(4.20)
and \( \xi (\mathbf{x}_k^N) = -\frac{\sigma_n^2}{2} \sum_{i=k}^N \sum_{j=1}^Q \ln Pr (\bar{c}_{i,j}) \). Note that the term generated by \textit{a priori} information, \( \xi (\mathbf{x}_1^N) \), considers only \( \mathbf{x}_k^N \) since the symbols \( \mathbf{x}_1^{k-1} \) are undecided. Note also that the linear MMSE estimate of the non-causal symbols \( \mathbf{x}_1^{k-1} \) is given by [61]

\[
\hat{\mathbf{x}}_1^{k-1} = \mathbf{F}_k \left( \mathbf{y}_1^{k-1} - E \left[ \mathbf{y}_1^{k-1} \bigg| \mathbf{x}_k^N = \bar{\mathbf{x}}_k^N \right] \right) + E \left[ \mathbf{x}_1^{k-1} \bigg| \mathbf{x}_k^N = \bar{\mathbf{x}}_k^N \right] \tag{4.21}
\]

\[
= \mathbf{F}_k \left( \mathbf{y}_1^{k-1} - \mathbf{R}_{11,k} \bar{\mathbf{x}}_1^{k-1} - \mathbf{R}_{12,k} \mathbf{x}_k^N \right) + \mathbf{x}_1^{k-1}, \tag{4.22}
\]

where \( \bar{\mathbf{x}}_1^{k-1} = E[\mathbf{x}_1^{k-1}] \) and \( \mathbf{F}_k = \text{Cov}(\mathbf{x}_1^{k-1}, \mathbf{y}_1^{k-1}|\mathbf{x}_k^N = \bar{\mathbf{x}}_k^N) \text{ Cov}^{-1}(\mathbf{y}_1^{k-1}|\mathbf{x}_k^N = \bar{\mathbf{x}}_k^N) \). We can obtain \( \bar{\mathbf{x}}_1^{k-1} \) and \( \mathbf{F}_k \) from \textit{a priori} LLRs as [62]

\[
\bar{\mathbf{x}}_1^{k-1} = \left[ \sum_{\theta \in \Theta} \theta \prod_{j=1}^Q \frac{1}{2} \left( 1 + \bar{c}_{i,j} \tanh \left( \frac{L_{\text{pri}} (\bar{c}_{i,j})}{2} \right) \right) \right] \tag{4.23}
\]

\[
\mathbf{F}_k = \mathbf{A}_k (\mathbf{R}_{11,k})^H \left( (\mathbf{R}_{11,k}) \mathbf{A}_k (\mathbf{R}_{11,k})^H + \sigma_n^2 \mathbf{I} \right)^{-1}, \tag{4.24}
\]

where \( \mathbf{A}_k = \text{diag}(\lambda_1, \ldots, \lambda_{k-1}) \) and

\[
\lambda_i = \sum_{\theta \in \Theta} |\theta - \bar{x}_i| \prod_{q=1}^Q \frac{1}{2} \left( 1 + \bar{c}_{i,q} \tanh \left( \frac{L_{\text{pri}} (\bar{c}_{i,q})}{2} \right) \right). \tag{4.25}
\]

The set \( \Theta \) includes all possible constellation points. In the first iteration where \textit{a priori} LLRs are unavailable, \( \mathbf{A}_k = \mathbf{I} \) and \( \mathbf{x}_1^{k-1} = \mathbf{0} \).

Using (4.20) and (4.22), \( p^{(b)} (\mathbf{x}_k^N) \) can be rewritten

\[
p^{(b)} (\mathbf{x}_k^N) = \left\| (\mathbf{I} - \mathbf{R}_{11,k} \mathbf{F}_k) \left( \mathbf{y}_1^{k-1} - \mathbf{R}_{11,k} \bar{\mathbf{x}}_1^{k-1} - \mathbf{R}_{12,k} \mathbf{x}_k^N \right) \right\|^2 \tag{4.26}
\]

\[
= \left\| \mathbf{Z}_k \left( \mathbf{y}_1^{k-1} - \mathbf{R}_{11,k} \bar{\mathbf{x}}_1^{k-1} - \mathbf{R}_{12,k} \mathbf{x}_k^N \right) \right\|^2, \tag{4.27}
\]

where

\[
\mathbf{Z}_k = \mathbf{I} - \mathbf{R}_{11,k} \mathbf{F}_k \tag{4.28}
\]

\[
= \sigma_n^2 (\mathbf{R}_{11,k} \mathbf{A}_k (\mathbf{R}_{11,k})^H + \sigma_n^2 \mathbf{I})^{-1}. \tag{4.29}
\]

Further, denoting \( \mathbf{q}_k = \mathbf{Z}_k (\mathbf{y}_1^{k-1} - \mathbf{R}_{11,k} \bar{\mathbf{x}}_1^{k-1}) \) and \( \mathbf{P}_k = \mathbf{Z}_k \mathbf{R}_{12,k} \), \( p^{(i)} (\mathbf{x}_k^N) \)
can be simply expressed

\[ p^{(l)} (x_k^N) = p^{(c)} (x_k^N) + \| q_k - P_k x_k^N \|_2. \]  

(4.30)

The bias term \( \| q_k - P_k x_k^N \|_2 \) of the LE-LA path metric is simply computed with linear operations without searching over combinations of \( x_1^{k-1} \). Note that \textit{a priori} information obtained from the channel decoder is reflected through \( \bar{x}_1^{k-1} \) and \( \Lambda_k \) in the bias term.

### 4.3.3 Comments on the Bias Term

Using (4.17), the signal \( q_k \) is expressed

\[ q_k = Z_k (R_{12,k} \bar{x}_k^N + R_{11,k} (\bar{x}_1^{k-1} - \bar{x}_1^{k-1}) + n_1^{k-1}) \]  

(4.31)

\[ = P_k \bar{x}_k^N + Z_k b_k, \]  

(4.32)

where \( b_k = R_{11,k} (\bar{x}_1^{k-1} - \bar{x}_1^{k-1}) + n_1^{k-1} \), and \( \bar{x}_1^{k-1} \) and \( \bar{x}_k^N \) are the vectors associated with the actual transmitted symbols. We can observe that \( Z_k \) in (4.29) corresponds to

\[ Z_k = \text{Cov} (n_1^{k-1}, b_k) \text{Cov}^{-1} (b_k) \]  

(4.33)

which implies that \( Z_k \) is the operator generating the linear MMSE estimate of \( n_1^{k-1} \) based on \( b_k \), i.e., \( \hat{n}_1^{k-1} = Z_k b_k \). Since \( n_1^{k-1} \) and \( R_{11,k} (\bar{x}_1^{k-1} - \bar{x}_1^{k-1}) \) here play the role of the signal to be estimated and the noise, respectively, the effect of \( R_{11,k} (\bar{x}_1^{k-1} - \bar{x}_1^{k-1}) \) on \( b_k \) is suppressed by \( Z_k \). Denoting the MMSE estimation error as \( e_1^{k-1} = (n_1^{k-1} - \hat{n}_1^{k-1}) \), (4.32) becomes

\[ q_k = P_k \bar{x}_k^N + (n_1^{k-1} - e_1^{k-1}). \]  

(4.34)

We see that \( q_k \) is expressed as a sum of the signal \( P_k \bar{x}_k^N \) and a perturbation vector \( n_1^{k-1} - e_1^{k-1} \). In this sense, the bias term in (4.30) becomes the Euclidean distance between the corrupted signal \( P_k \bar{x}_k^N + (n_1^{k-1} - e_1^{k-1}) \) and a candidate vector \( P_k x_k^N \). We can show that the covariance matrices of the
perturbations \(e_1^{k-1}\) and \((n_1^{k-1} - e_1^{k-1})\) are, respectively,

\[
\text{Cov}(e_1^{k-1}) = \sigma_n^2 I - \sigma_n^4 (R_{11,k} A_k (R_{11,k})^H + \sigma_n^2 I)^{-1}
\]  

(4.35)

\[
\text{Cov}(n_1^{k-1} - e_1^{k-1}) = \sigma_n^4 (R_{11,k} A_k (R_{11,k})^H + \sigma_n^2 I)^{-1}
\]  

(4.36)

Note that the perturbation vector \(n_1^{k-1} - e_1^{k-1}\) is correlated due to the MMSE estimation errors. The impact of this bias term on the detection performance will be analyzed in Section 4.4.

### 4.3.4 Efficient Computation of Path Metric

In this subsection, we discuss how the LE-LA path metric can be computed efficiently. Recalling that the bias term is expressed as \(\|Z_k(y_1^{k-1} - R_{11,k} x_1^{k-1} - R_{12,k} x_k^N)\|^2\), computation of the path metric is divided into two steps: (1) computation of \(Z_k\) for all \(k\) prior to the tree search and (2) recursive update of the path metric for each branch extension during the search.

First, using a matrix inversion formula for block matrices [63], the operators \(Z_k\), \((k = 1, \cdots, N)\) in (4.29) can be computed recursively. Denoting

\[
R_{11,k+1} = \begin{bmatrix} R_{11,k} & r_{k+1} \\ 0 & r_{k+1,k+1} \end{bmatrix}, \quad A_{k+1} = \begin{bmatrix} A_k & 0 \\ 0 & \lambda_{k+1} \end{bmatrix},
\]

(4.37)

and \(r_{k+1} = [r_{1,k+1}, \cdots, r_{k,k+1}]^T\), then \(Z_{k+1}\) is expressed as a function of \(Z_k\) as

\[
Z_{k+1} = \begin{bmatrix} Z_k - K \lambda_{k+1} Z_k r_{k+1}^H Z_k \\
-K \lambda_{k+1} r_{k+1,k+1}^H Z_k K (\lambda_{k+1} + \sigma_n^2 Z_k r_{k+1}^H Z_k + \sigma_n^2) \\
-K \lambda_{k+1} r_{k+1,k+1}^H Z_k K (\lambda_{k+1} + \sigma_n^2 Z_k r_{k+1}^H Z_k + \sigma_n^2) \end{bmatrix},
\]

(4.38)

where

\[
K = \frac{1}{\lambda_{k+1} (r_{k+1}^H (Z_k r_{k+1}^H + r_{k+1,k+1}^2 r_{k+1,k+1}) + \sigma_n^2).}
\]

(4.39)

In particular, \(Z_2 = \frac{\sigma_n^2}{\lambda_{1} r_{1,1} + \sigma_n^2}\). See Appendix E for the derivation of (4.38). If the \textit{a priori} LLRs are all zero, the complexity for computing \(Z_k\) can be shared over the coherent period over which channel state is constant [1]. If the \textit{a priori} LLRs are non-zero, these steps are performed for each symbol. However, the required computations can be further reduced by replacing the instantaneous covariance matrix \(A_k\) by its time-average over a coherent
time [64].

Next, the LE-LA path metric can be recursively updated for each tree extension. At the root node, a vector is defined as \( \mathbf{a}_{N+1} = \mathbf{y} - \mathbf{R} \mathbf{x}_1^N \). The vector \( \mathbf{a}_k \) is updated from that of its parent node as 

\[
\begin{bmatrix}
\mathbf{a}_k \\
\mathbf{v}_k
\end{bmatrix} = \mathbf{a}_{k+1} - \begin{bmatrix}
r_{1,k} & \cdots & r_{k,k}
\end{bmatrix}^T (x_k - \bar{x}_k),
\]

(4.40)

where \( \mathbf{v}_k \) is a scaler variable. Using the vector updated for each path \( \mathbf{x}_k^N \), the LE-LA path metric can be obtained as

\[
p^{(l)} (\mathbf{x}_k^N) = p^{(c)} (\mathbf{x}_k^N) + p^{(b)} (\mathbf{x}_k^N)
\]

(4.41)

\[
p^{(c)} (\mathbf{x}_k^N) = p^{(c)} (\mathbf{x}_{k+1}^N) + |v_k|^2 + \xi (x_k)
\]

(4.42)

\[
p^{(b)} (\mathbf{x}_k^N) = \| \mathbf{Z}_k \cdot \mathbf{a}_k \|^2,
\]

(4.43)

where \( p^{(c)} (\mathbf{x}_{N+1}^N) = 0 \). Noting that the dimension of the matrix \( \mathbf{Z}_k \) is \((k-1) \times (k-1)\), the number of complex multiplications for the bias term computation is proportional to \((k-1)^2\) at level \( k \). In order to reduce complexity, we can look ahead only \( N_l < k - 1 \) symbols instead of all non-causal symbols. Towards this goal, we set \( \alpha = \max(0, k - N_l) \) and repartition the system as

\[
\begin{bmatrix}
\mathbf{y}_k^{k-1} \\
\mathbf{y}_k^N
\end{bmatrix} = \begin{bmatrix}
\mathbf{R}_{11,k} & \mathbf{R}_{12,k} \\
0 & \mathbf{R}_{22,k}
\end{bmatrix} \begin{bmatrix}
\mathbf{x}_k^{k-1} \\
\mathbf{x}_k^N
\end{bmatrix} + \begin{bmatrix}
\mathbf{n}_k^{k-1} \\
\mathbf{n}_k^N
\end{bmatrix},
\]

(4.44)

where \( \mathbf{R}_{11,k} \) and \( \mathbf{R}_{12,k} \) are the redefined sub-matrices of (4.17), respectively. In this case, the bias term defined in Section 4.3.2 needs to be modified based on this partitioning. In doing so, the dimension of \( \mathbf{R}_{11,k} \) and \( \mathbf{Z}_k \) is reduced to \( N_l \times N_l \) from \((k-1) \times (k-1)\). The recursive computation of \( \mathbf{Z}_k \) employing the new partitioning can be derived without matrix inversion (see [15, Section III. A]). In addition, in (4.43), we only need to multiply \( \mathbf{Z}_k \) with the last \( \alpha \) elements of \( \mathbf{a}_k \). Overall, by using only \( N_l \) non-causal symbols for the bias term, the number of operations for bias computation can be reduced to \( MN \cdot N_l^2 \) from \( M \sum_{k=1}^N (k-1)^2 (= (M/6)(2N^2 - 3N^2 + N)) \).
4.3.5 Application to APP Detection

In this section, we introduce the soft-input soft-output tree detection algorithm employing the LE-LA path metric. In each level of the tree, \( p^{(i)}(x_k^N) \) of \( 2^{2Q}M \) survival paths are compared and then the \( M \) best paths are selected. Starting from the root node, this candidate selection procedure continues to the bottom level and \( 2^{2Q}M \) complete paths are chosen. The symbol vectors corresponding to these complete paths generate a candidate list \( \mathcal{L} \), over which the extrinsic LLR for each bit is calculated. In the event that a particular bit in each of the candidates takes the same value (all one or zero), the magnitude of the generated LLR might become unduly large, limiting the error-correction capability of the channel decoder [65]. In order to prevent this situation, whenever this occurs for the \( k \)th bit of the candidate list, the \( k \)th bits of the best \( J \) candidates (\( J \leq 2^{2Q}M \)) are flipped and added into the candidate list \( \mathcal{L} \), generating an extended list \( \mathcal{L}_{\text{ext}}^k \). We have seen that list extension based on this modification can alleviate performance losses with small values of \( J \), e.g., \( J = 4 \). Over the list \( \mathcal{L}_{\text{ext}}^k \), the approximated APP is computed as

\[
L_{\text{post}}(c_{k,i}) \approx \max_{x \in \mathcal{L}_{\text{ext}}^k \cap X^{+1}_{k,i}} \lambda(x) - \max_{x \in \mathcal{L}_{\text{ext}}^k \cap X^{-1}_{k,i}} \lambda(x). \tag{4.45}
\]

A summary of the ISS-MA is provided in Table 4.1.

4.4 Performance Analysis

We discussed in the previous section that the transmitted symbols are always found with \( M = 1 \) if the genie-aided path metric is used. Relaxation of the finite alphabet constraint and the Gaussian approximation were made for undecided symbols to derive the LE-LA path metric. In this section, we show that this path metric offers performance gains over the conventional path metric when applied to the soft-input soft-output \( M \)-algorithm. As a measure for performance, we consider the probability of a CPL event, i.e., the probability that the tree search rejects a path associated with the transmitted symbols. To make the analysis tractable, we focus on the case when \( M = 1 \).

Given the channel matrix \( \mathbf{R} \) and the \textit{a priori} LLRs, the probability of CPL
Table 4.1: Summary of ISS-MA.

Output: \(\{L_{\text{post}}(\tau_{k,i})\}_{k=[1:N], i=[1:Q]}\)
Input: \(y, H, \{L_{\text{pre}}(\tau_{k,i})\}_{k=[1:N], i=[1:Q]}\), and \(J\)

**STEP 1**: (Preprocessing) Perform V-BLAST symbol ordering and QR decomposition of \(H\). Obtain \(Z_1\) to \(Z_N\) for all \(k\) levels.

**STEP 2**: (Initialization) Initialize \(i = N + 1\) and start the tree search from the root node.

**STEP 3**: (Loop) Extend \(2^Q\) branches for each of \(M\) paths that have survived at the \((i + 1)\)th level. This generates \(2^QM\) paths at the \(i\)th level.

**STEP 4**: If \(i > 1\), choose the best \(M\) paths with the smallest \(p^{(i)}(x_N)\) and go to **STEP 3** with \(i = i - 1\). Otherwise, store all \(2^QM\) survival candidates into the list \(L\) and go to **STEP 5**.

**STEP 5**: (List extension & APP calculation) For each value of \(k\) and \(i\), compute \(\{L_{\text{post}}(\tau_{k,i})\}\) based on \(L\). If the value of \(\tau_{k,i}\) for all elements of \(L\) is either +1 or -1, the value of \(\tau_{k,i}\) of the best \(J\) candidates is flipped and these counter-hypothesis candidates are added to \(L\) to generate the extended \(L^+_k\). The APP is calculated over the extended list based on (4.45).

\[
P_{\text{CPL}} = 1 - \Pr(\tilde{x} \in L | \tilde{x} \text{ is sent})
= 1 - \prod_{k=1}^{N} \left( 1 - \overline{\Pr}(\tilde{x}_k^N \notin L_k | \tilde{x}_{k+1}^N \in L_{k+1}) \right)
\]

where \(L_k\) denotes the set of paths selected at the \(k\)th level and \(\overline{\Pr}(\cdot)\) is the probability given that \(\tilde{x}\) is sent. Since we consider the case of \(M = 1\), \(\tilde{x}_{k+1}^N \in L_{k+1}\) implies that a correct path has been selected up to the \(k + 1\)th level. With this setup and from (4.17), (4.19), and (4.27), we can show that...
The SINR in (4.53) is bounded by Lemma 4.4.1.

\[
\text{SINR} = \frac{(r_k^H Z_k r_k + |r_{k,k}|^2)^2}{r_k^H Z_k^2 E[b_k b_k^H] Z_k^2 r_k + \sigma_n^2 |r_{k,k}|^2} \leq 1 \left( \frac{r_k^H (\sigma_n^4 \Sigma_k^{-2}) r_k + |r_{k,k}|^2}{\sigma_n^2} \right) \leq \text{SINR} \leq \frac{r_k^H \Sigma_k^{-1} r_k + |r_{k,k}|^2}{\sigma_n^2}. \tag{4.54}
\]
Proof. See Appendix F.

Taking similar steps, we can show that the SINR for the causal path metric, 
\[ p^c(x^N_k) = \frac{|r_{k,k}|^2}{\sigma_n^2}. \]
Hence, \( r_k^H \Sigma_k^{-1} r_k^H \) and \( \sigma_n^2 r_k^H \Sigma_k^{-2} r_k^H \) can be regarded as upper and lower bounds on the SINR gain achieved by the LE-LA path metric, respectively. It is of interest to check the behavior of the upper and lower bound of SINR gain for high dimensional systems. Suppose that \( N, L \to \infty \) with a fixed aspect ratio \( \beta = N/L \) (0 < \( \beta \leq 1 \)), and let \( \lambda_{\min} \) and \( \lambda_{\max} \) be smallest and largest diagonals of \( \Lambda_k \), respectively. Then, we attain a loose bound on the SINR

\[
\sigma_n^2 r_k^H (\sigma_n^2 I + \lambda_{\max} R_{11,k} R_{11,k}^H)^{-2} r_k + \frac{|r_{k,k}|^2}{\sigma_n^2} \leq \text{SINR}
\]

\[
\leq r_k^H (\sigma_n^2 I + \lambda_{\min} R_{11,k} R_{11,k}^H)^{-1} r_k + \frac{|r_{k,k}|^2}{\sigma_n^2}.
\]

(4.55)

This can be shown by the relationship \( B \preceq \Sigma_k \preceq A \) (equivalently, \( \Sigma_k^{-1} \preceq B^{-1} \) and \( A^{-2} \preceq \Sigma_k^{-2} \)), where \( A = \sigma_n^2 I + \lambda_{\max} R_{11,k} R_{11,k}^H \) and \( B = \sigma_n^2 I + \lambda_{\min} R_{11,k} R_{11,k}^H \).

**Theorem 4.4.2.** For an \( L \times N \) matrix \( H \) whose elements are i.i.d. random variables with zero mean and variance \( \frac{1}{L} \), the upper and lower bound of the SINR gain for \( k = \gamma N + 1 \) (0 < \( \gamma < 1 \)) converge as

\[
B_k^{\text{upper}} \longrightarrow B_k^{\text{upper,}\infty} = \frac{1}{2\lambda_{\min}} \left( -1 - (1 - \gamma \beta) \frac{\lambda_{\min}}{\sigma_n^2} + G \left( \frac{\lambda_{\min}}{\sigma_n^2}, \gamma \beta \right) \right)
\]

(4.56)

\[
B_k^{\text{lower}} \longrightarrow B_k^{\text{lower,}\infty} = \frac{1}{2\sigma_n^2} \left( -(1 - \gamma \beta) + \frac{1 + \gamma \beta + (1 - \gamma \beta)^2 \lambda_{\max}}{G \left( \frac{\lambda_{\max}}{\sigma_n^2}, \gamma \beta \right)} \right)
\]

(4.57)

as \( N, L \to \infty \) with \( \beta = N/L \), where \( G(x, b) = \sqrt{1 + 2(1 + b)x + (1 - b)^2 x^2} \).

Proof. See Appendix G.
Corollary 4.4.3. As $\sigma_n^2 \rightarrow 0$, we have

$$B_{k,\infty}^{\text{upper}} \rightarrow \lambda \min \frac{\gamma \beta}{(1 - \gamma \beta)}$$

(4.58)

$$B_{k,\infty}^{\text{lower}} \rightarrow 0.$$  (4.59)

In addition, as $\sigma_n^2 \rightarrow 0$, $B_{k,\infty}^{\text{upper}}$ monotonically increases and approaches $\lambda \min \frac{\gamma \beta}{(1 - \gamma \beta)}$.

Proof. See Appendix H. \qed

We can deduce from (4.58) and (4.59) that the actual SINR gain approaches a deterministic value between $[0, \lambda \min \frac{\gamma \beta}{(1 - \gamma \beta)}]$. One can also show that both $B_{k,\infty}^{\text{upper}}$ and $B_{k,\infty}^{\text{lower}}$ are increasing functions of $\gamma \beta \in (0, 1)$. Noting that $\gamma$ indicates an index for tree depth, the SINR bounds achieve their maximum at the top tree level.

Next, we analyze the probability of CPL using the SINR obtained. In order to make the derivation more tractable, we make a Gaussian approximation for the MMSE estimation error $e_i^{k-1}$ in (4.34), equivalently, $Z_b = n_i + e_i^{k-1}$. Under this approximation, we can assume that the interference plus noise of the scalar channel is Gaussian as well. The validity of this approximation has been supported in many asymptotic scenarios in [47] and [66]. In particular, it is shown that the Gaussian approximation is highly accurate for large problem size $N$ [67].

Using the SINR in (4.53), the probability of CPL for the $k$th level detection can be expressed as [2]

$$\Pr \left( x_k^N \notin L_k \mid x_{k+1}^N \in L_{k+1}, H \right) \leq 4 \left( 1 - \frac{1}{\sqrt{2Q}} \right) Q \left( \sqrt{K \frac{1}{\sigma_n^2} \left( \frac{r_k^H \left( \sigma_n^2 \Sigma_k^{-2} \right) r_k + |r_{k,k}|^2}{r_k^H \left( \sigma_n^2 \Sigma_k^{-3} \right) r_k + |r_{k,k}|^2} \right)} \right),$$

(4.60)

where $K = \frac{3}{(2^{\frac{1}{2}} - 1)}$. The inequality in (4.60) follows from the existence of a priori terms in (4.51), which lowers the actual CPL probability. From (4.54),
we have
\[
\Pr \left( \tilde{x}_k^N \notin \mathcal{L}_k \middle| \tilde{x}_{k+1}^N \in \mathcal{L}_{k+1}, H \right) \\
\leq 4 \left( 1 - \frac{1}{\sqrt{2Q}} \right) Q \left( \sqrt{K} \left( \sigma_n^2 r_k^H \Sigma_k^{-2} r_k + \frac{\mathbb{E}_{k,k}}{\sigma_n^2} \right) \right). \tag{4.61}
\]

Using (4.61), we next analyze an average probability of CPL for a random channel matrix $H$ whose elements are independent complex Gaussian with $\mathcal{CN}(0, 1)$. The average probability of CPL, denoted as $P_{\text{CPL}}$, is given by
\[
P_{\text{CPL}} = 1 - E_H \left[ \prod_{k=1}^N \left( 1 - \Pr \left( \tilde{x}_k^N \notin \mathcal{L}_k \middle| \tilde{x}_{k+1}^N \in \mathcal{L}_{k+1}, H \right) \right) \right] \tag{4.62}
\]
\[
= \sum_{k=1}^N E_H \left[ \Pr \left( \tilde{x}_k^N \notin \mathcal{L}_k \middle| \tilde{x}_{k+1}^N \in \mathcal{L}_{k+1}, H \right) \right] + \text{higher order terms}, \tag{4.63}
\]
where $E_H[\cdot]$ denotes the expectation operation in terms of $H$. The higher order terms are ignored since they become negligible in the high SNR regime. Using the relationship $Q(\sqrt{x+y}) \leq Q(\sqrt{x}) \exp \left( -\frac{y}{2} \right)$ for $x, y > 0$ and from (4.61), we have
\[
E_H \left[ \Pr \left( \tilde{x}_k^N \notin \mathcal{L}_k \middle| \tilde{x}_{k+1}^N \in \mathcal{L}_{k+1}, H \right) \right] \\
\leq LE_H \left[ Q \left( \sqrt{K} \frac{|r_{k,k}|^2}{\sigma_n^2} \right) \exp \left( -\frac{K\sigma_n^2 r_k^H \Sigma_k^{-2} r_k}{2} \right) \right] \tag{4.64}
\]
\[
= LE_H \left[ Q \left( \sqrt{K} \frac{|r_{k,k}|^2}{\sigma_n^2} \right) \right] E_H \left[ \exp \left( -\frac{K\sigma_n^2 r_k^H \Sigma_k^{-2} r_k}{2} \right) \right], \tag{4.65}
\]
where (4.66) follows from independence of $r_{k,k}$ and $r_k$. Noting that $r_{k,k}$ has a Chi-square distribution with $2(L - k + 1)$ degrees of freedom and $r_k$ has
independent complex Gaussian elements [68, Lemma 2.1], we have [1]

\[
E_H \left[ Q \left( \sqrt{\frac{K}{\sigma^2_n}} |r_{k,k}|^2 \right) \right] \\
= \left( \frac{1}{2} - \frac{1}{2} \sqrt{\frac{K}{K + 2\sigma^2_n}} \right)^{L-k+1} \sum_{l=0}^{L-k} \binom{L - k + l}{l} \left( \frac{1}{2} + \frac{1}{2} \sqrt{\frac{K}{K + 2\sigma^2_n}} \right)^l.
\]

(4.67)

**Lemma 4.4.4.** An upper bound on the scaling gain in (4.66) is given by

\[
E_H \left[ \exp \left( -\frac{K\sigma^2_n r^H_k \Sigma^{-2} r_k}{2} \right) \right] \leq \int_0^\infty \cdots \int_0^\infty \left( \prod_{i=1}^{k-1} \frac{1}{1 + \frac{K}{2} \frac{\sigma^2_n}{(\lambda_{\max}x_i + \sigma^2_n)^2}} \right) \\
\times f_{\eta_1, \cdots, \eta_{k-1}} (x_1, \cdots, x_{k-1}) \, dx_1 \cdots dx_{k-1},
\]

(4.68)

where

\[
f_{\eta_1, \cdots, \eta_{k-1}} (x_1, \cdots, x_{k-1}) = \frac{1}{(k-1)!} \exp \left( -\sum_{i=1}^{k-1} x_i \right) \prod_{i=1}^{k-1} \frac{x_i^{L-k+1}}{(k-1-i)!} \prod_{i<j} (x_i - x_j)^2.
\]

(4.69)

**Proof.** See Appendix I. \qed

While \( \exp \left( -\frac{K\sigma^2_n r^H_k \Sigma^{-2} r_k}{2} \right) \) tends to one as \( \sigma^2_n \to 0 \), (4.67) decreases to zero with a slope \( \lim_{\sigma^2_n \to 0} \ln(P_e) / \ln(\sigma^2_n) = L - k + 1 \). Therefore, at high SNR, the probability of CPL for the top level \( (k = N) \) would dominate, i.e.,

\[
\overline{P}_{\text{CPL}} \lesssim LE_H \left[ Q \left( \sqrt{\frac{K|r_{N,N}|^2}{\sigma^2_n}} \right) \right] E_H \left[ \exp \left( -\frac{K\sigma^2_n r^H_N \Sigma^{-2} r_N}{2} \right) \right],
\]

(4.70)

where the right-hand side is obtained from (4.67) and (4.68). On the other hand, an upper bound of the average CPL probability for the causal path metric becomes

\[
\overline{P}_{\text{CPL}}^{\text{causal}} \lesssim LE_H \left[ Q \left( \sqrt{\frac{K|r_{N,N}|^2}{\sigma^2_n}} \right) \right].
\]

(4.71)

59
We observe from (4.70) that the average CPL probability of the LE-LA path metric is smaller than that of the causal path metric by the factor of $E_H \left[ \exp \left( -\frac{K}{2} \sigma_n^2 r_N^H \Sigma_N^{-2} r_N \right) \right]$. Since this term is strictly less than unity, we can refer to this as a scaling gain.

In Figure 4.2 and 4.3, we provide the plot of the average CPL probability versus SNR for several system sizes ($N = 5, 10, 15,$ and $20$) and for an uncoded system. The average CPL rate and its upper bound are obtained from (4.60) and (4.61). For a comprehensive view, the average CPL rate for the causal path metric in (4.71) is included as well. For all cases considered, the CPL expression in (4.60) is quite close to that obtained from simulation results, supporting the accuracy of the analytic bound we obtained. In particular, the upper bound of the average CPL rate appears tight at high SNR. Figure 4.4 shows how the scaling gain in (4.68) varies as a function of SNR and system size. We observe that the performance gain of the LE-LA path metric improves with system size and the maximum is achieved in low to moderate SNR range ($10 \text{ dB} \sim 20 \text{ dB}$). This behavior is desirable for IDD, since the performance in the low-to-mid SNR range is critical in triggering performance improvement though iterations [69].

4.5 Simulation and Discussion

In this section, we evaluate the performance of the ISS-MA through computer simulations. First, we observe the performance of the soft-input soft-output M-algorithm employing the LE-LA path metric and that employing the conventional path metric. Then, we compare the performance and complexity of the ISS-MA and existing tree detection algorithms.

4.5.1 Simulation Setup

The simulation setup for the IDD system is as follows. A rate $R = 1/2$ turbo code with the generation matrix $(15, 17)$ is used. For the channel decoding, a log-MAP decoder [60] is employed and a total of 10 inner iterations are carried out for each code block.\(^1\) We use a random interleaver of a block of 12,000

\(^1\)For convenience, we call “inner iteration” for decoding of turbo code and “outer iteration” for iterative detection and decoding.
Figure 4.2: Average CPL rate versus SNR for the (a) 5 \times 5 and (b) 10 \times 10 systems. QPSK uncoded transmission is considered. The curves for the “CPL rate (anal.)” are obtained by Monte-Carlo averaging of (4.60) over i.i.d. Gaussian channels.
Figure 4.3: Average CPL rate versus SNR for the (a) $15 \times 15$ and (b) $20 \times 20$ systems. QPSK uncoded transmission is considered. The curves for the “CPL rate (anal.)” are obtained by Monte-Carlo averaging of (4.60) over i.i.d. Gaussian channels.
Figure 4.4: Scaling gain versus SNR for different system sizes $N = 5, 10, 15, 20$.

coded bits. A total of $10^5$ information bits are randomly generated. Perfect knowledge of the channel state at the receiver is assumed. We focus on a $20 \times 20$ QPSK system and a $10 \times 10$ 16-QAM system, both of which would require quite demanding detection and decoding complexity. The channel state is assumed to be constant over a block consisting of ten symbols but changes independently across each individual block. The entries of $H$ are modeled as i.i.d. complex Gaussian variables $\mathcal{CN}(0, 1)$. The spectral efficiency of the MIMO transmission is given by $RQN$ bit/Hz/s and the complexity is measured by counting the average number of complex multiplications per symbol period and per outer iteration.\footnote{The complexity for QR decomposition and detection ordering is not considered since they are common in all detection algorithms under consideration.}

4.5.2 Simulation Results

In this subsection, we compare the performance between the conventional $M$-algorithm using the causal path metric \cite{54} and the proposed ISS-MA employing the LE-LA path metric. For both algorithms, we employ the
Figure 4.5: Comparison between the causal path metric and LE-LA path metric for (a) the 20 × 20 QPSK system and (b) the 10 × 10 16-QAM system. For both cases, $M$ is set to 12.

same candidate extension strategy described in Section 4.3.5 and set the parameter $J$ to 4 for fair comparison.

In Figure 4.5 (a) and (b), we provide BER performance for the 20 × 20 QPSK and the 10 × 10 16-QAM systems, respectively. The parameter $N_l$ is set to 8 and 4, respectively, and the parameter $M$ is set to 12. For the 20 × 20
Table 4.2: The number of complex multiplications per channel use and per iteration for the ISS-MA and the conventional $M$ algorithm.

<table>
<thead>
<tr>
<th>System</th>
<th>Complexity of conventional $M$-alg.</th>
<th>Complexity of ISS-MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$20 \times 20$ QPSK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M = 12$</td>
<td>81k</td>
<td>94k</td>
</tr>
<tr>
<td>$M = 20$</td>
<td>97k</td>
<td>115k</td>
</tr>
<tr>
<td>$M = 56$</td>
<td>175k</td>
<td>222k</td>
</tr>
<tr>
<td>$10 \times 10$ 16-QAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M = 12$</td>
<td>51k</td>
<td>72k</td>
</tr>
<tr>
<td>$M = 20$</td>
<td>74k</td>
<td>108k</td>
</tr>
<tr>
<td>$M = 48$</td>
<td>156k</td>
<td>232k</td>
</tr>
</tbody>
</table>

QPSK system, we observe that the ISS-MA achieves about 1 dB performance gain over the conventional $M$-algorithm after 7th outer iteration. For the $10 \times 10$ 16-QAM system, the ISS-MA achieves 0.5 dB performance gain. Note that the performance gain is more significant after fewer (< 7) outer iterations are carried out. In particular, with no outer iteration (first outer iteration), the ISS-MA achieves more than 1.5 dB gain. We can deduce from these results that the ISS-MA would outperform the conventional $M$-algorithm for the uncoded systems as well. In addition, the performance of the ISS-MA converges within 5 outer iterations while the conventional $M$-algorithm needs more iterations for convergence.

Figure 4.6 shows how the performance of the ISS-MA and the conventional detector behave in terms of $M$. In Figure 4.6 (a), the performance for the $20 \times 20$ QPSK system is provided for $M = 12, 20, \text{and } 48$. The performance is measured after the 7 outer iterations. Note that the performance gain of the ISS-MA remains for all values of $M$. For the $20 \times 20$ system, even $M = 48$ is not sufficient to capture promising candidates for the conventional $M$-algorithm. In contrast, the ISS-MA finds the promising candidates with small $M$ values due to the enhanced sorting process. Figure 4.6 (b) shows the results for the $10 \times 10$ 16-QAM system. Note that 0.5 dB performance gain of the ISS-MA is maintained for all $M$ values ($M = 12, 20, \text{and } 56$).

Table 4.2 summarizes the search complexity of the ISS-MA in comparison with that of the conventional $M$-algorithm. With $N_l = 8$ for the $20 \times 20$ QPSK system and $N_l = 4$ for the $10 \times 10$ 16-QAM system, the complexity overhead of the ISS-MA is kept small. For $M = 12$, the complexity overhead due to the LE-LA path metric is 16% for the $20 \times 20$ QPSK system and 41% for the $10 \times 10$ 16-QAM system. Note that the performance of the
Figure 4.6: Comparison between the causal path metric and LE-LA path metric for several $M$ values for (a) the $20 \times 20$ QPSK system and (b) the $10 \times 10$ 16-QAM system. The performance is measured after the 7th iteration.
conventional $M$-algorithm with $M = 48$ or $M = 56$ is worse than that of the ISS-MA with $M = 12$. Nevertheless, the conventional $M$-algorithm requires twice the complexity of the ISS-MA. This result clearly demonstrates that the ISS-MA provides a better performance-complexity trade-off. Furthermore, the performance of the ISS-MA converges faster than the conventional $M$-algorithm, which would reduce the complexity of the ISS-MA further by early termination of outer iterations.

Since the parameter $N_l$ affects the performance and complexity of the ISS-MA, it is of interest to observe the impact of $N_l$ on the BER performance. Figure 4.7 (a) and (b) show how the BER changes with $N_l$ for $20 \times 20$ QPSK and $10 \times 10$ 16-QAM systems, respectively. The BER is measured after the 7 iterations. As $N_l$ decreases, the SNR value at which the BER steeply drops to zero remains unchanged until $N_l$ becomes 4 and 2, respectively (we refer to this SNR value as SNR threshold). For the $20 \times 20$ system, the SNR threshold remains 4 dB until $N_l$ becomes 8. Similar behavior is observed for the $10 \times 10$ system. From this result, it is seen that the choice of $N_l = 8$ and $N_l = 4$ is sufficient to achieve the maximal performance gain for these cases.

Next, we compare the ISS-MA with the existing soft-input soft-output tree detection algorithms. Along with the ISS-MA and the conventional $M$-algorithm, we consider the following algorithms:

1. LISS algorithm ($|S|, |S_x|$) [12] - List sequential stack algorithm. It is characterized by the size of stack $|S|$ and that of auxiliary stack $|S_x|$ for computing bias term.

2. Hard sphere decoding (HSD) [51] - After applying hard sphere search, the candidate list is generated by flipping each bit of the MAP estimate. Only single bit flipping is considered for the APP generation.

The SNR threshold and average complexity of the detectors are provided in Table 4.3. Since the complexity of the sphere search grows exponentially as the system size increases, the HSD has significantly larger complexity than that of the other tree detectors. While both the ISS-MA and the HSD achieve the best performance among all candidate detectors, the complexity of the ISS-MA is much lower than that of the HSD. Due to the limited stack size ($|S| = 4096$), the LISS also offers lower complexity compared to the HSD. However, the performance of the LISS is not as good as those of the HSD and
20 x 20 system, QPSK (M=20)

(a)

10 x 10 system, 16-QAM (M=20)

(b)

Figure 4.7: BER versus SNR for several $N_t$ values for (a) 20 x 20 QPSK system and (b) 10 x 10 16-QAM system.

Table 4.3: Performance comparison of several soft-input soft-output tree detectors.

<table>
<thead>
<tr>
<th></th>
<th>20 x 20 QPSK</th>
<th>10 x 10 16-QAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR thres.</td>
<td>SNR thres.</td>
<td></td>
</tr>
<tr>
<td>SNR thres.</td>
<td>SNR thres.</td>
<td></td>
</tr>
<tr>
<td>ISS-MA ($M = 12$)</td>
<td>4 dB</td>
<td>94 dB</td>
</tr>
<tr>
<td>SNR thres.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR thres.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conventional $M$-alg. ($M = 12$)</td>
<td>5 dB</td>
<td>81k dB</td>
</tr>
<tr>
<td>SNR thres.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR thres.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LISS ($</td>
<td>S</td>
<td>= 4096,</td>
</tr>
<tr>
<td>SNR thres.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR thres.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSD</td>
<td>4 dB</td>
<td>1322k</td>
</tr>
</tbody>
</table>

68
ISS-MA because the stack memory used in the LISS easily gets full before reaching a leaf of the tree so that it does not often end up with reliable symbol candidates. On the other hand, the ISS-MA consistently produces a fixed number of candidates while providing good BER performance. From the above results, we confirm that the ISS-MA achieves the best performance-complexity trade-off among all candidate detectors.

4.6 Conclusions

In this chapter, we discussed a new path metric that shows great promise in terms of its performance-complexity trade-off for soft-input soft-output tree detection in an IDD system. The performance gains over the existing causal path metric is achieved by accounting for non-causal symbols in the linear estimate-based look-ahead path metric. This improved path metric is applied to the soft-input soft-output $M$-algorithm, which produces a candidate list needed to compute the APP value. By adopting the sorting mechanism exploiting the LE-LA path metric, we dramatically improve the chance of selecting the correct path, thereby achieving good detection and decoding performance with fewer iterations. From CPL probability analysis, we observed that the LE-LA path metric reflects the reliability of path much better than causal path metric. Computer simulations confirm that the proposed ISS-MA can be a promising candidate for soft-input soft-output detection in high dimensional systems.
CHAPTER 5

ADAPTIVE LINEAR TURBO
EQUALIZATION OF LARGE DELAY
SPREAD TIME-VARYING CHANNEL

5.1 Introduction

In order to achieve reliable communication over wireless channels with large delay spread, inter-symbol interference (ISI) caused by multi-path arrivals of a transmitted signal should be mitigated by an equalization technique. For decades, equalization techniques for a variety of ISI channels have been established in the literature [70]. However, performance of equalization is often hampered by poor channel qualities such as rapid time-variation of channel gains and non-linear system response. A typical example of such challenging channels is the underwater acoustic channel. Figure 5.1 shows a time-delay response of an underwater channel for 1 km distance from the transmitter. It is observed that at a symbol rate of 9.77 ksymbol/s, the channel response spans around 70-90 symbols. When a hydrophone array with multiple elements is deployed, the number of unknown parameters for the equalization task can run into the thousands, which demands a computationally intensive equalization algorithm. In addition, the increase in the number of unknown parameters makes the fast tracking of time-varying channel responses difficult. One approach to lower the complexity is to use an orthogonal frequency division multiplexing (OFDM) transmission, which transforms a taped-delay-line channel response to parallel scalar channel gains [14]. By performing separate scalar equalization for each frequency bin, the complexity of the receiver can be significantly reduced. Unfortunately, for rapidly varying channels such as an underwater acoustic channel, these approaches often suffer from an intercarrier interference (ICI), which degrades the system performance [71].

To approach the performance of the joint equalization and decoding, the turbo equalization (TEQ) technique, motivated by the turbo principle [3], was
introduced, which exploits the information obtained from a channel decoder for better equalization [15, 16, 64, 72]. In [72], a turbo equalizer algorithm is derived by computing a posteriori log-likelihood ratio (LLR) of the data bits directly using a priori LLRs obtained from the channel decoder. Since direct computation of a posteriori LLR demands high complexity, various low-complexity TEQ techniques are proposed. In [16], a linear adaptive TEQ is introduced, where soft symbol estimates are fed back to the equalizer and the equalizer coefficients are adjusted to minimize mean square error (MSE) in an adaptive fashion. A linear minimum mean square error (MMSE) TEQ was investigated in [15, 64], where an optimal linear MMSE equalizer was derived given exact knowledge of the channels. Without knowledge of the channel state, the channel could be estimated and the estimates are incorporated into the equalizer. Since channel estimation errors deteriorate equalization performance, the covariance matrix of the channel estimation errors was accounted for in the derivation of the linear MMSE equalizer to reduce the impact of channel estimation errors in [73]. In [74] and [75], a low-complexity implementation of joint channel estimation and equalization approaches are investigated. The application of similar iterative detection schemes can be found in other digital communication problems such as multi-
user detection [76], multi-input multi-output (MIMO) detection [13, 62, 77], and ICI cancellation for OFDM system [71].

Recently, the TEQ techniques have been applied to underwater acoustic communications [78–84]. Iterative equalization algorithms were derived for multi-channel equalization in [78, 79] and the extensive experimental results for the turbo receiver are presented in [80]. A low-complexity turbo equalizer was presented for systems employing space-time codes in [81] and a non-linear TEQ technique followed by a prefiltering (channel-shaping) filter is proposed in [82]. In [83], multi-carrier transmission is considered to reduce the effective channel length for low-complexity equalization. In [84], a Markov Chain Monte Carlo (MCMC) turbo equalizer is also developed to bridge the gap between the complex nonlinear MAP equalizer and linear TEQ algorithm.

In this chapter, we study the practical application of linear TEQ techniques to underwater acoustic communications. The main contribution of this chapter is to demonstrate the practical feasibility of turbo equalization for single-carrier data transmission. Towards this end, we provide the extensive experimental results that confirm a significant performance gain achieved by TEQ techniques.

First, we investigate two popular TEQ algorithms of linear structure: (1) a channel estimate-based MMSE TEQ (CE-based MMSE-TEQ) technique and (2) a direct-adaptive TEQ (DA-TEQ) technique. The CE-based MMSE-TEQ algorithm exploits an explicit channel estimate to determine the equalizer coefficients. In contrast, the DA-TEQ algorithm directly estimates a symbol using well-known adaptive algorithms [85]. Since both TEQ algorithms are suboptimal without knowledge of the channel, it is of interest to take a close look at their differences. Basically, two TEQ algorithms operate on different criteria: While the CE-based MMSE-TEQ algorithm minimizes a conditional MSE for given a priori information on symbols and the estimated channel information, the DA-TEQ minimizes an unconditional MSE due to the autonomous nature of adaptive algorithms. As a result, the coefficients of the DA-TEQ do not capture time-varying second-order symbol statistics, and hence its optimal Wiener solution is different from that of the optimal MMSE-TEQ. We compare the performance behavior of the CE-based MMSE-TEQ and DA-TEQ algorithms in the presence of channel estimation errors and adjustment errors of a least mean square (LMS) adaptive algorithm used in the implementation. In spite of the discrepancy of their optimal
forms, we show through extrinsic information transfer (EXIT) chart analysis that the performance gap between the DA-TEQ and CE-based MMSE-TEQ techniques is small after convergence.

Next, we introduce an underwater acoustic communication receiver architecture based on the DA-TEQ technique. An LMS adaptive algorithm is used to maintain low equalization complexity. To facilitate good tracking performance for the LMS DA-TEQ algorithm, we improve a convergence speed of the LMS algorithm by two methods: (1) repeating the weight update for the same set of data (data reuse LMS) with decreasing step size (gear shifting LMS) and (2) reducing the number of active equalizer taps based on the sparse structure of the underwater acoustic channels. With the aid of these means, the LMS DA-TEQ technique can show better decoding results than the original form of the LMS turbo equalizer [16] in real underwater channels. The performance of our receiver architecture was tested via the SPACE 08 experiment conducted off the coast of Martha’s Vineyard, MA, USA. We show that a substantial performance gain can be achieved over the conventional decision feedback equalizer widely adopted for underwater acoustic communication modems [86].

It is worth mentioning that we have attempted a challenging transmission setup, i.e., signal transmission beyond the Nyquist signalling rate. To increase the data rate under the restriction of bandwidth, we pushed a symbol rate beyond the level guaranteeing no ISI, that is, twice the transmission bandwidth $2W$ [2]. This would generate intentional ISI at the transmitter, increasing an effective channel span at the receiver and degrading the receiver performance. We will show that the LMS DA-TEQ algorithm maintains strong performance even in this challenging setup.

The rest of this chapter is organized as follows. In Section 5.2, the overall system is described. In Section 5.3, the TEQ algorithms with linear structure are briefly described, and in Section 5.4 the performances of two practical linear TEQ algorithms, CE-based MMSE-TEQ and DA-TEQ, are compared. In Section 5.5, an underwater receiver architecture based on the LMS DA-TEQ algorithm is introduced. In Section 5.6, the experimental results are discussed and in Section 5.7 some conclusions are discussed.
5.2 System Description

5.2.1 Transmitter Description

The transmitter structure considered here is based on space-time bit-interleaved coded modulation (ST-BICM) [87]. The ST-BICM allows us to perform coding over different temporal and spatial dimensions. First, the information bits \( \{ b_k \} \) are encoded by a rate \( R_c \) channel encoder, producing the coded bit sequence \( \{ c_i \} \). The sequence \( \{ c_i \} \) is permuted using a random interleaver, and \( Q \) interleaved bits are mapped to \( 2^Q \)-ary quadrature amplitude modulation (QAM) symbols. The sequence of QAM symbols is divided into \( N_t \) parallel streams using a serial to parallel converter. These parallel data streams are transmitted through a transmitter array with \( N_t \) elements. In the sequel, we denote \( x_{m,n} \) as a symbol sent by the \( m \)th array element at time \( n \).

Data transmission is carried out frame-by-frame. As shown in Figure 5.2, a frame consists of \( T \) sets of the training symbols followed by the data symbols. The preamble symbols are inserted in front of every multiple frame for data acquisition and synchronization purposes. The training symbols are used to aid tracking performance of the adaptive equalizer or channel estimator. The periods of training, data, and preamble sequences are \( M_t \), \( M_d \), and \( M_p \) symbol times, respectively.
5.2.2 Receiver Description

Assuming perfect synchronization and symbol-spaced sampling, the signal received at the hydrophone array with \( N_r \) elements is expressed as

\[
\mathbf{r}_n = \sum_{k=-K_f}^{K_p} \mathbf{H}_{n,k} \mathbf{x}_{n-k} + \mathbf{w}_n, \tag{5.1}
\]

where \( n \) and \( k \) are time and delay indices, respectively. Note that \( \mathbf{r}_n \) and \( \mathbf{w}_n \) are the \( N_r \times 1 \) noise and received vectors, respectively, \( \mathbf{x}_n = [x_{1,n}, \ldots, x_{N_t,n}]^T \), and \( \mathbf{H}_{n,k} \) is the \( N_r \times N_t \) channel matrix whose \((l,m)\)th element is the channel gain from the \( m \)th transmitter element to the \( l \)th receiver element. The channel length is assumed to be at most \( K_f + K_p + 1 \), where \( K_f \) is the length of the precursor and \( K_p \) is that of the postcursor response. We also assume that channel gains change in time \( n \). The signal vector \( \mathbf{x}_n \) is normalized to have unit energy and assumed to be uncorrelated due to the interleaver, i.e., \( E[\mathbf{x}_n \mathbf{x}_n^H] = \mathbf{I}_{N_t} \). Considering an observation window containing \( L_f + L_p + 1 \) received vectors, i.e., \( \mathbf{r}_{n-L_p}, \ldots, \mathbf{r}_{n+L_f} \), we can write

\[
\mathbf{y}_n = \mathbf{H}_n \mathbf{s}_n + \mathbf{n}_n \tag{5.2}
\]

where

\[
\mathbf{y}_n = \begin{bmatrix} \mathbf{r}_{n-L_p}^T, \ldots, \mathbf{r}_{n+L_f}^T \end{bmatrix}^T, \quad \mathbf{n}_n = \begin{bmatrix} \mathbf{w}_{n-L_p}^T, \ldots, \mathbf{w}_{n+L_f}^T \end{bmatrix}^T, \\
\mathbf{s}_n = \begin{bmatrix} \mathbf{x}_{n+K_f+L_f}^T, \ldots, \mathbf{x}_{n-K_p-L_p}^T \end{bmatrix}^T, \\
\mathbf{H}_n = \begin{bmatrix} \mathbf{H}_{n,L_f-L_p} & \ldots & \mathbf{H}_{n,L_f+K_p} & 0 & 0 \\
0 & \ddots & \ddots & \ddots & 0 \\
0 & 0 & \mathbf{H}_{n-L_p,L_f-L_p} & \ldots & \mathbf{H}_{n-L_p,L_f+K_p} \end{bmatrix}.
\]

The vector \( \mathbf{y}_n \) contains all received vectors in the observation window and the noise vector \( \mathbf{n}_n \) is assumed to be complex Gaussian, i.e., \( \mathbf{n}_n \sim \mathcal{CN}(\mathbf{0}, \mathbf{R}) \). In the sequel, we will denote \( K = N_t(K_p + K_f + L_p + L_f + 1) \) and \( L = N_r(L_p + L_f + 1) \) so that the size of \( \mathbf{H}_n \) becomes \( L \times K \). For convenience, we also denote \( \mathbf{s}_n = [s_{n,1}, \ldots, s_{n,K}]^T, \mathbf{y}_n = [y_{n,1}, \ldots, y_{n,L}]^T, \) and \( \mathbf{H}_n = [\mathbf{h}_{n,1}, \ldots, \mathbf{h}_{n,K}] \). In addition, we let \( \mathbf{H}_{n,i;j} = [\mathbf{h}_{n,i}, \mathbf{h}_{n,i+1}, \ldots, \mathbf{h}_{n,j}] \) for \( i < j \).

We denote the \( Q \) coded (and interleaved) bits corresponding to the symbol...
Figure 5.3: Block diagram of the receiver architecture. $\Pi$ and $\Pi^{-1}$ denote interleaving and deinterleaving operations, respectively.

$s_{n,k}$ by $\{\tau_{n,k}^{1}, \cdots, \tau_{n,K}^{Q}\}$.

The block diagram of the TEQ system is shown in Figure 5.3. The equalizer produces the extrinsic log-likelihood ratios (LLRs) on the coded bits $\{\tau_{n,k}^{1}, \cdots, \tau_{n,K}^{Q}\}$ using the received vector $y_n$ and the a priori information on $s_n$ [15, 64]. The extrinsic LLR $L_{\text{ext}}(\tau_{n,k}^{q})$ can be obtained from $L_{\text{ext}}(\tau_{n,k}^{q}) = L_{\text{post}}(\tau_{n,k}^{q}) - L_{\text{pri}}(\tau_{n,k}^{q})$, where $L_{\text{pri}}(\tau_{n,k}^{q})$ is the a priori LLR defined as $L_{\text{pri}}(\tau_{n,k}^{q}) = \ln Pr(\tau_{n,k}^{q} = +1) - \ln Pr(\tau_{n,k}^{q} = -1)$ and $L_{\text{post}}(\tau_{n,k}^{q})$ is a posteriori LLR defined as $L_{\text{post}}(\tau_{n,k}^{q}) = \ln Pr(\tau_{n,k}^{q} = +1 | \text{observation}) - \ln Pr(\tau_{n,k}^{q} = -1 | \text{observation})$. Note that the observation can be any function of $y_n$ and the a priori LLRs are obtained from the channel decoder. The extrinsic LLRs obtained in the equalizer are delivered to the channel decoder through the deinterleaver. They are used as a priori LLRs to produce the additional extrinsic LLRs in the channel decoder. Finally, the extrinsic LLRs obtained by the channel decoder are fed back to the equalizer to aid the equalization task. These steps complete one cycle of turbo iteration and are repeated until a suitably chosen convergence criterion is achieved.

5.3 Review of Turbo Equalizer with Linear Structure

In what follows, we briefly review three different linear TEQ techniques: (1) MMSE-TEQ, (2) CE-based MMSE-TEQ, and 3) DA-TEQ. The MMSE-TEQ algorithm minimizes an MSE criterion under the condition that the channel state is perfectly known. When the channel information is not known, two alternative TEQ algorithms are available: the CE-based MMSE-TEQ technique and the DA-TEQ technique. Though these two algorithms are suboptimal, they are widely adopted for practical systems due to relatively low complexity.
5.3.1 MMSE Turbo Equalizer

With perfect knowledge of the channel matrix $H_n$, a linear MMSE estimate of the symbol $s_{n,k}$ for $k \in [M(K_f+N_f)+1, M(K_f+N_f)+M]$ is given by [64]

$$\hat{s}_{n,k}^\text{MMSE} = z_{n,k}^H (y_n - H_n \bar{s}_{n,k})$$

(5.3)

$$z_{n,k} = (H_n \Sigma_{n,k} H_n^H + R)^{-1} h_{n,k},$$

where

$$\bar{s}_{n,k} = [\bar{s}_{n,1}, \cdots, \bar{s}_{n,k-1}, 0, \bar{s}_{n,k+1}, \cdots, \bar{s}_{n,K}]^T$$

$\Sigma_{n,k} = \text{diag}(\lambda_{n,1}, \cdots, \lambda_{n,k-1}, 1, \lambda_{n,k+1}, \cdots, \lambda_{n,K})$

are a mean vector and covariance matrix of $s_n$ for given a priori LLRs. To prevent early limit-cycle behavior, the linear MMSE estimate in (5.3) does not rely on a priori knowledge of $s_{n,k}$, i.e., $\bar{s}_{n,k} = 0$ and $\lambda_{n,k} = 1$. We can obtain $\bar{s}_{n,k}$ and $\Sigma_{n,k}$ from a priori LLRs as [64]

$$\bar{s}_{n,i} = \sum_{\theta \in \Theta} \theta \prod_{q=1}^{Q} \frac{1}{2} \left( 1 + \bar{c}_i^q \tanh \left( \frac{L_{\text{pri}}(c_i^q)}{2} \right) \right)$$

(5.4)

$$\lambda_{n,i} = \sum_{\theta \in \Theta} |\theta|^2 \prod_{q=1}^{Q} \frac{1}{2} \left( 1 + \bar{c}_i^q \tanh \left( \frac{L_{\text{pri}}(c_i^q)}{2} \right) \right) - |\bar{s}_{n,i}|^2.$$  (5.5)

A simpler expression of $\bar{s}_{n,k}$ and $\Sigma_{n,k}$ is available for a particular signal mapper [15]. By treating the MMSE estimate $\hat{s}_{n,k}^\text{MMSE}$ as an observation, we can compute the extrinsic LLR for $\bar{c}_{n,k}^q$ as follows. From (5.2) and (5.3), we can express $\hat{s}_{n,k}^\text{MMSE}$ as a sum of a signal part and a residual error part, i.e., $\hat{s}_{n,k}^\text{MMSE} = \mu_{n,k} s_{n,k} + \eta_{n,k}$, where

$$\mu_{n,k} = z_{n,k}^H h_{n,k}$$

(5.6)

$$\eta_{n,k} = z_{n,k}^H (H_n (s_n - \bar{s}_{n,k}) - h_{n,k} s_{n,k}).$$

(5.7)
We can show \( \text{Var}(\eta_{n,k}) \triangleq \sigma_{n,k}^2 = \mu_{n,k}(1 - \mu_{n,k}) \) \[64\]. If we assume that \( \eta_{n,k} \) is Gaussian, i.e., \( \eta_{n,k} \sim \mathcal{CN}(0, \sigma_{n,k}^2) \), the extrinsic LLR of \( \tau^q_{n,k} \) is given by

\[
L_{\text{MMSE}} (\tau^q_{n,k}) = \ln \frac{P(\hat{s}^{\text{MMSE}}_{n,k} | \tau^q_{n,k} = +1)}{P(\hat{s}^{\text{MMSE}}_{n,k} | \tau^q_{n,k} = -1)}
\]

\[
= \ln \frac{\sum_{\theta \in \Theta_{k,q}^+} \exp \left( -\frac{\left| \hat{s}^{\text{MMSE}}_{n,k} - \mu_{n,k} \theta \right|^2}{\sigma_{n,k}^2} \right)}{\sum_{\theta \in \Theta_{k,q}^-} \exp \left( -\frac{\left| \hat{s}^{\text{MMSE}}_{n,k} - \mu_{n,k} \theta \right|^2}{\sigma_{n,k}^2} \right)},
\]

where \( \Theta_{k,q}^+ \) and \( \Theta_{k,q}^- \) are the set of all constellation points such that \( \tau^q_{n,k} \) is +1 and -1, respectively.

### 5.3.2 Channel Estimate-based MMSE Turbo Equalizer (CE-based MMSE-TEQ)

When perfect channel knowledge is not available, the channel response can be estimated and incorporated into the coefficients of the MMSE-TEQ algorithm. Define the channel estimate and the corresponding channel estimation error as \( \hat{H}_n \) and \( E_n = H_n - \hat{H}_n \), respectively. We assume that \( E_n \) has zero mean and is uncorrelated with \( \hat{H}_n \). We assume that \( E_n \) is also uncorrelated with \( s_n \). Then, we can write (5.2) by \( y_n = \hat{H}_n s_n + (E_n s_n + n_n) \). Given the channel estimate \( \hat{H}_n \), the linear MMSE estimate of \( s_{n,k} \) is given by

\[
\hat{s}_{n,k}^{\text{CE-MMSE}} = \hat{z}_{n,k}^H \left( y_n - \hat{H}_n \bar{s}_{n,k} \right)
\]

\[
\hat{z}_{n,k} = \left( \hat{H}_n \Sigma_{n,k} \hat{H}_n^H + E \left[ E_n E_n^H \right] + R \right)^{-1} \hat{h}_{n,k}.
\]

Note that the term \( E \left[ E_n E_n^H \right] \) can be obtained from an some approximate analysis of the channel estimator. Alternatively, we can estimate the sum \( E \left[ E_n E_n^H \right] + R \) by

\[
E \left[ E_n E_n^H \right] + R \approx \frac{1}{M_t} \sum_{n=1}^{M_t} \left( y_n - \hat{H}_n s_n \right) \left( y_n - \hat{H}_n s_n \right)^H,
\]

\[78\]
where \( s_n \) is obtained from the training data. Denoting \( \hat{\mu}_{n,k} = \hat{z}_{n,k}^H \hat{h}_{n,k} \), we express the extrinsic LLR for \( \tau_{n,k}^0 \) as

\[
L_{\text{ext}}^{\text{CE-MMSE}}(\tau_{n,k}^0) = \ln \frac{\sum_{\theta \in \Theta_q^+} \exp \left( -\frac{\|s_{n,k}^{\text{CE-MMSE}} - \hat{\mu}_{n,k}\theta\|^2}{\hat{\mu}_{n,k}(1 - \hat{\mu}_{n,k})} \right)}{\sum_{\theta \in \Theta_q^-} \exp \left( -\frac{\|s_{n,k}^{\text{CE-MMSE}} - \hat{\mu}_{n,k}\theta\|^2}{\hat{\mu}_{n,k}(1 - \hat{\mu}_{n,k})} \right)}.
\] (5.12)

### 5.3.3 Direct-Adaptive Turbo Equalizer (DA-TEQ)

We can estimate the symbol \( s_k \) directly via an adaptive algorithm. First, we restrict the estimate of \( s_k \) to be a linear combination of observations and a priori symbol estimates, i.e., [16]

\[
\hat{s}_{n,k}^{\text{DA}} = f_{n,k}^H y_n + g_{n,k}^H \left[ s_{n,1:k-1} \right] \left[ s_{n,k+1:K} \right]^T,
\] (5.13)

where \( s_{n,i:j} = [s_{n,i}, \ldots, s_{n,j}]^T \) for \( i < j \). The vectors \( f_{n,k} \) and \( g_{n,k} \) represent the \( L \times 1 \) feedforward and \( (K-1) \times 1 \) feedback coefficients, respectively. These coefficients are adjusted such that the MSE \( E[|s_{n,k} - \hat{s}_{n,k}^{\text{DA}}|^2] \) is minimized. We can update these coefficients via an LMS algorithm, i.e.,

\[
\begin{bmatrix}
\begin{bmatrix} f_{n+1,k} \\ g_{n+1,k} \end{bmatrix}
\end{bmatrix} = \begin{bmatrix}
\begin{bmatrix} f_{n,k} \\ g_{n,k} \end{bmatrix}
\end{bmatrix} + \xi \left( s_{n,k} - \hat{s}_{n,k}^{\text{DA}} \right) \begin{bmatrix} y_n \\ s_{n,1:k-1} \\ s_{n,k+1:K} \end{bmatrix},
\] (5.14)

where \( \xi \) is a step size. When \( s_{n,k} \) is not available, a tentative symbol estimate \( Q(\hat{s}_{n,k}^{\text{DA}}) \) can be used in place of \( s_{n,k} \) in (5.14), where \( Q(\cdot) \) is the slicer operation that maps the input to the nearest constellation point. Note that to obtain \( \hat{s}_{n,k}^{\text{DA}} \) for all \( k \in [N_t(K_f + L_f) + 1, N_t(K_f + L_f) + N_t] \), the DA-TEQ updates \( N_t \) feedforward and feedback filters separately.

Computation of the extrinsic LLR from the output of the DA-TEQ is not straightforward since the channel response, and equivalently \( \mu_{n,k} \) and \( \sigma_{n,k}^2 \) in (5.9) are not known. An approximate soft demapping algorithm for obtaining the extrinsic LLR in the DA-TEQ is proposed in [16]. In Section 5.5, we will present a different soft demapping algorithm for our receiver architecture.
5.4 Comparison of CE-Based MMSE-TEQ versus LMS DA-TEQ

In this section, we compare the performance of the CE-based MMSE-TEQ algorithm and the LMS DA-TEQ algorithm. To simplify the analysis, we assume a time-invariant channel, i.e., $H_n = H$ for all $n$. Following the methodology used in [69], we model the a priori LLR as a conditional Gaussian variable, i.e., $L_{pri}(\overline{c}_{n,k}^l) \sim \mathcal{N}(\sigma_g^2/2, \sigma_g^2)$ for $\overline{c}_{n,k}^l = +1$ and $L_{pri}(\overline{c}_{n,k}^l) \sim \mathcal{N}(-\sigma_g^2/2, \sigma_g^2)$ for $\overline{c}_{n,k}^l = -1$. Hence, the distribution of the a priori LLR is characterized by the single parameter $\sigma_g^2$. In the analysis that follows, a priori LLR is seen as a random signal that is correlated with $\overline{c}_{n,k}^l$. Since $\tilde{s}_n$ and $\Sigma_n$ are functions of the a priori LLR (see (5.4) and (5.5)), they are also random signals characterized by $\sigma_g^2$. We will consider in the analysis that a priori LLRs provide side information on $\overline{c}_{n,k}^l$ denoted as $\mathcal{L}$. As the parameter $\sigma_g^2$ increases, the reliability of this side information becomes stronger. Denoting $\overline{s}_n = E[s_n|\mathcal{L}]$, we have $E[\overline{s}_n] = 0_{K \times 1}$ and $\text{Cov}(s_n - \overline{s}_n, \overline{s}_n') = 0_K$ due to the orthogonality principle. In addition, with $\text{Cov}(\overline{s}_n) \triangleq \sigma_g^2 I_K$, we have $\text{Cov}(s_n - \overline{s}_n) = (1 - \sigma_g^2)I_K$. The framework under which the analysis is performed is depicted in Figure 5.4.

![Figure 5.4: Setup for performance analysis.](image)

5.4.1 Mean Square Error (MSE) Analysis

First, we look at the performance of the CE-based MMSE-TEQ algorithm. Given the side information $\mathcal{L}$ and the channel estimate $\tilde{H}_n$, the CE-based
The MMSE-TEQ algorithm minimizes the conditional MSE, i.e.,

$$E \left[ |s_{n,k} - \hat{s}^{CE-MMSE}_{n,k}|^2 \mid \mathcal{L}, \hat{H}_n \right]$$

$$= 1 - \hat{h}^H_{n,k} \left( \hat{H}_n \Sigma_{n,k} \hat{H}_n^H + E \left[ E_n E_n^H \right] + R \right)^{-1} \hat{h}_{n,k}. \quad (5.15)$$

We assume that the channel estimation error, $(\hat{H}_n - H)$, is a random matrix uncorrelated with $\hat{H}_n$. To obtain an average MSE, (5.15) should be averaged over $\Sigma_{n,k}$ and $\hat{H}_n$, i.e.,

$$E \left[ |s_{n,k} - \hat{s}^{CE-MMSE}_{n,k}|^2 \right]$$

$$= 1 - E \left[ \hat{h}^H_{n,k} \left( \hat{H}_n \Sigma_{n,k} \hat{H}_n^H + E \left[ E_n E_n^H \right] + R \right)^{-1} \hat{h}_{n,k} \right]. \quad (5.16)$$

Consider a standard LMS channel estimator with a step size $\xi_{\text{ch}}$ estimating $K_p + K_f + 1$ channel taps. The covariance matrix of the channel estimation error is given by $\frac{\xi_{\text{ch}} \text{tr}(R)}{L} I_{K_p + K_f + 1}$ (see Appendix J). Assuming that $\hat{H}_n - H$ is complex Gaussian, we can obtain (5.16) by generating random samples of $\hat{H}_n$ and performing Monte Carlo averaging.

Next, we take a look at the performance of the LMS DA-TEQ algorithm. Contrary to the CE-based MMSE-TEQ algorithm, the LMS DA-TEQ algorithm minimizes the unconditional MSE, i.e., $E[|s_{n,k} - \hat{s}^{LMS-DA}_{n,k}|^2]$ via the LMS algorithm. At first, we look at the Wiener solution to which the coefficients of the LMS DA-TEQ converge in mean square sense. Let the converged feedback and feedforward coefficients be $g^o_k$ and $f^o_k$, respectively. Using the orthogonality principle between $s_{n,k} - \hat{s}^{LMS-DA}_{n,k}$ and $[s^T_{n,1:k-1}, \hat{s}^T_{n,k+1:K}]^T$, we have

$$g_{n,k} \rightarrow g^o_k = - \left[ H_{n,1:k-1} \ H_{n,k+1:K} \right]^H f^o_k. \quad (5.17)$$

In a similar manner, using that $s_{n,k} - \hat{s}^{LMS-DA}_{n,k}$ and $y_n$ are uncorrelated, we get

$$f_{n,k} \rightarrow f^o_k = (h_{n,k} h_{n,k}^H + (1 - \sigma_s^2) H_{n,1:k-1} H_{n,1:k-1}^H + (1 - \sigma_s^2) H_{n,k+1:K} H_{n,k+1:K}^H + R)^{-1} h_{n,k}. \quad (5.18)$$

From (5.13), (5.17), and (5.18), we can show that the output of LMS DA-
TEQ converges to

\[ \hat{s}_{n,k}^{\text{LMS-DA}} \rightarrow (f_k^o)^H (y_n - H_n \bar{s}_{n,k}). \]  

(5.19)

We can compare this Wiener solution with the optimal MMSE solution in (5.3). Clearly, the optimal MMSE-TEQ and DA-TEQ share the same structure; the soft feedback estimates are subtracted from the observation, i.e., \( y_n - H_n \bar{s}_{n,k} \) and then interference suppressing filters \( z_{n,k} \) and \( f_k^o \) are applied. However, the coefficients \( z_{n,k} \) and \( f_k^o \) have different values. While \( z_{n,k} \) of the CE-based MMSE-TEQ depends on an instantaneous estimate of covariance matrix, \( \Sigma_{n,k} \), \( f_k^o \) of the LMS DA-TEQ depends on an (ensemble) covariance matrix of the signal \( \bar{s}_n \). The reason for this discrepancy is that two TEQ algorithms rely on different criteria: conditional versus unconditional MSE. As a result, the DA-TEQ algorithm does not fully exploit time-varying (second-order) statistics of the symbols \( s_{n,k} \) obtained from the channel decoder. Note that two filter coefficients, \( z_{n,k} \) and \( f_k^o \), turn out to be equal only when \( \sigma^2_g = 0 \) or \( \sigma^2_g \to \infty \). Now, we derive the MSE of the LMS DA-TEQ algorithm. In general, the steady-state MSE of an adaptive equalizer can be separated into a minimum MSE \( \mathcal{M}_k^{\text{min}} \) determined by the optimal Wiener solution and an excess MSE \( \mathcal{M}_k^{\text{excess}} \) due to misadjustment of filter adaptation [85]. Accordingly, the MSE for the LMS DA-TEQ technique is given by

\[ E[|s_{n,k} - \hat{s}_{n,k}^{\text{LMS-DA}}|^2] = \mathcal{M}_k^{\text{min}} + \mathcal{M}_k^{\text{excess}} \]  

(5.20)

where

\[ \mathcal{M}_k^{\text{min}} = 1 - h_{n,k}^H (h_{n,k} h_{n,k}^H + (1 - \sigma^2_s)H_{n,1:k-1}^H H_{n,1:k-1}^H) \]

\[ + (1 - \sigma^2_s) H_{n,k+1:K}^H H_{n,k+1:K}^{-1} h_{n,k}, \]  

(5.21)

and

\[ \mathcal{M}_k^{\text{excess}} \approx \frac{\xi \mathcal{M}_k^{\text{min}} (\text{tr} (H_n H_n^H + R) + \sigma^2_s (K - 1))}{2}, \]  

(5.22)

where \( \xi \) is a step size of the LMS DA-TEQ algorithm. Derivation of (5.22) is presented in Appendix K.

In Figure 5.5, we investigate how the MSEs in (5.16) and (5.20) behave in terms of \( \sigma^2_g \). We consider the single-input single-output setup, i.e., \( L = \)
Figure 5.5: (a) Average MSEs of MMSE-TEQ and DA-TEQ for the step size $\mu = 0.001$ and (b) those for $\mu = 0.0002$. The SNR is set to 0 dB.
$M = 1$ and QPSK modulation. We also consider a channel response shown in Figure 5.6 (a). The SNR is set to 0 dB. We assume that the step-size of the LMS channel estimator and the LMS DA-TEQ are set to $\xi = \xi_{ch}$ equally. In Figure 5.5 (a), the plots of MSE versus $\sigma_g^2$ are provided for the step size $\xi = \xi_{ch} = 0.001$. Note that the MSE gap between two TEQ algorithms decreases when the a priori LLRs delivered from the channel decoder become stronger, i.e., $\sigma_g^2 \to \infty$. Hence, we expect that the performance gap between two TEQs would diminish after a sufficient number of iterations. Figure 5.5 (b) shows the plot of MSE for $\xi = \xi_{ch} = 0.0002$. Due to smaller step size, the channel estimation errors and excess MSE of the DA-TEQ algorithm get smaller. In this case, the performance gap between two TEQ algorithms is reduced further. This implies that the DA-TEQ algorithm is more sensitive to the LMS adaptation noise than the CE-based MMSE-TEQ. Nevertheless, the MSE gap approaches zero as $\sigma_g^2 \to \infty$ for both cases.

### 5.4.2 Extrinsic Information Transfer (EXIT) Chart Analysis

The iterative behavior of TEQ algorithms can be well illustrated by an EXIT chart [64,69]. In the EXIT chart, mutual information (MI) transfer curves of an equalizer and a channel decoder are drawn in the same chart to show the exchange of the MI between them. Note that the output MI ($I_{Eq,\text{output}}$) is drawn in terms of the input MI ($I_{Eq,\text{out}}$) for the equalizer while the output MI ($I_{Dec,\text{in}}$) is drawn in terms of the output MI ($I_{Ec,\text{out}}$) for the channel decoder.

---

1. This channel is obtained from a 10,000th snapshot of underwater channels in Figure 5.1.

---

84
In Figure 5.7 (a) and (b), the EXIT charts of two equalizers are provided along with that of (133, 171) recursive systematic convolutional (RSC) code. We use the same setup as for the MSE analysis except that the step size is set to $\xi = \xi_{ch} = 0.001$. As for the DA-TEQ technique, the steady-state coefficients are expressed as $f_{n,k} = f^o_k + \Delta f_{n,k}$ and $g_{n,k} = g^o_k + \Delta g_{n,k}$, where $f^o_k$ and $g^o_k$ are Wiener solutions and $\Delta f_{n,k}$ and $\Delta g_{n,k}$ are the perturbation noise due to misadjustment errors. The coefficients of the LMS DA-TEQ are

---

**Figure 5.7**: Exit chart graphs for (a) SNR = −0.25 dB and (b) SNR = −0.4 dB.
generated from the distribution
\[
\begin{bmatrix}
    f_{n,k} \\
    g_{n,k}
\end{bmatrix}
\sim \mathcal{CN}
\left(
\begin{bmatrix}
    f_k^r \\
    g_k^o
\end{bmatrix}, \frac{\mu}{2^M_k} I_{N+K-1}
\right).
\] (5.23)

See Appendix J for the derivation of (5.23). For each value of input MI, the output MI is obtained via a histogram-based numerical method [64]. Comparing the trajectories stepping between the equalizer curve and decoder curve in Figure 5.7, we observe that the MI transfer curves of both TEQs approach similar output MI values as the input MI approaches one (see the points A and B in the charts). As long as SNR is high enough to create a tunnel between equalizer curve and decoder curve, actual performance achieved by both TEQs would not be much different after a sufficient number of iterations. On the other hand, the LMS DA-TEQ algorithm requires more iterations for convergence and a higher SNR to achieve convergence beyond the waterfall region, i.e., it has a higher SNR threshold.

5.5 Underwater Acoustic Communication Receiver
Based on LMS DA-TEQ Algorithm

In this section, we introduce a receiver architecture built upon the LMS DA-TEQ technique. The block diagrams of the underwater receiver for the \(N_r \times 1\) SIMO and the \(N_r \times 2\) MIMO systems are depicted in Fig. 5.8 (a) and (b), respectively. The received data at each hydrophone array element is processed by the synchronization block. A symbol timing, a start of processing frame, and a channel length are estimated by thresholding the correlation between the received samples and preamble data. The threshold is determined based on the relative magnitude of correlations measured in the silent period and preamble period. In order to align the data frame of all channels, we obtain a rough estimate of channel support by thresholding the magnitude of correlation. Once each data stream is aligned, the LMS DA-TEQ algorithm is applied. The symbol estimates are converted to the extrinsic LLRs in the soft-input soft-output demapper block. When there are multiple transmitter array elements, i.e., \(N_t > 1\), the multiple streams of the extrinsic LLRs are multiplexed into a single stream via the parallel-to-serial converter. The
Figure 5.8: Structure of underwater receiver based on the LMS DA-TEQ for (a) $N_r \times 1$ SIMO and (b) $N_r \times 2$ MIMO systems.
remaining steps were explained in Section 5.2.2.

5.5.1 Tracking Performance Enhancement

Generally, the LMS adaptive equalizer suffers from a slow convergence speed. In shallow water, channel dynamics tend to change rapidly in a local scale so that the statistic of channel gains is highly non-stationary [88]. Consequently, the LMS algorithm often fails to converge to a steady state within the period over which the channel is stationary and hence the equalization performance is degraded. In order to improve the slow convergence of the LMS algorithm, we employ two methods.

First, we partition the whole frame into multiple sub-blocks over each of which the channel is assumed to be stationary. For each sub-block, we repeat the weight update of the (normalized) LMS algorithm until the filter weights converge to a steady state. At each repetition (called ‘pass’), the step size $\mu$ is decreased by the factor of $\rho (< 1)$. For each pass, the equalizer coefficients are initialized by the last update in the previous pass. Denoting $B$ as the size of the sub-block and $J$ as a total number of repetitions per sub-block, the repeated weight update algorithm is summarized as follows:

for $t = 1 : D$,
Set $\xi = \xi_{\text{init}}$.
for $i = 1 : J$
    for $n = (t - 1)B + 1 : tB$
        Let $e_{n,k} = s_{n,k} - \hat{s}_{LMS-DA}^{n,k}$.
        For all $k$, compute
        \[
        \begin{bmatrix}
        f_{n+1,k} \\
        g_{n+1,k}
        \end{bmatrix}
        = \begin{bmatrix}
        f_{n,k} \\
        g_{n,k}
        \end{bmatrix}
        + \frac{\xi e_{n,k} \begin{bmatrix}
        y_n & s_{n,1:k-1} & s_{n,k+1:K}
        \end{bmatrix}^T}{\epsilon + \|y_n\|^2 + \|s_{n,1:k-1}\|^2 + \|s_{n,k+1:K}\|^2},
        \]
    end
    Set $\xi \leftarrow \rho \xi$. For all $k$, set $f_{(t-1)B+1,k} \leftarrow f_{tB,k}$, and $g_{(t-1)B+1,k} \leftarrow g_{tB,k}$.
end

$DB$ becomes the length of the whole frame ($= T(M_t + M_d)$) and $\xi_{\text{init}}$ is the
initial step size. For better tracking performance, we employ an \(\epsilon\)-normalized LMS algorithm [89]. A training symbol is used in place of \(s_{n,k}\) in the training mode, and a tentative symbol estimate \(Q(\hat{s}_{n,k}^{\text{LMS-DA}})\) in the detection mode. Note that the size of sub-block \(B\) should be chosen carefully such that the channel is stationary within the sub-block and sufficient data statistics are captured for equalizing the channel.

Next, we exploit the sparse structure of the underwater channel to reduce the number of active equalizer coefficients. By reducing the number of unknown parameters that should be tracked by the equalizer, convergence speed of the equalizer can be enhanced. We extend the sparse equalization technique developed for the conventional DFE in [90] to the LMS DA-TEQ algorithm. Specifically, we first employ an LMS channel estimator to obtain a rough estimate of channels during the training period and order the channel taps in terms of the average power. We select the \(\gamma\) strongest taps, where \(\gamma\) is chosen such that the difference between the original MSE and the MSE obtained after active channel tap selection is less than \(P\) percent of the original MSE for some \(\gamma\). After the sparse channel structure is identified, a part of feedforward and feedback taps are switched off depending on the sparse structure found. The sparse structure directly indicates which part of feedback weights should be switched off. More specifically, the feedback taps not contributing to the output due to zero channel gains, are deactivated. In contrast, the feedforward equalizer taps are switched off at the cost of performance loss. We preserve the matched filter property of the equalizer so that the feedforward coefficients corresponding to zero channel taps reflected on the observation \(r_n\) are turned off to reduce the active equalizer taps.

5.5.2 Soft-Input Soft-Output Demapper

The LMS DA-TEQ technique cannot directly compute \(\mu_{n,k}\) and \(\sigma^2_{n,k}\) in (5.9) due to the absence of the channel estimate. The soft-input soft-output demapper estimates \(\mu_{n,k}\) and \(\sigma^2_{n,k}\) and produces the extrinsic LLRs using them. From (5.2) and (5.13), the output of the LMS DA-TEQ algorithm is
expressed as
\[ s_{n,k}^{\text{LMS-DA}} = f_{n,k}^H (H_n s_n + n_n) + g_{n,k}^H \left[ \bar{s}_{n,1:k-1} \right] + g_{n,k}^H \left[ \bar{s}_{n,k+1:K} \right], \] (5.24)
\[ = f_{n,k}^H h_{n,k} s_{n,k} + \left( f_{n,k}^H \left[ H_{n,1:k-1} \ H_{n,k+1:K} \right] s_{n,1:k-1} + g_{n,k}^H \left[ \bar{s}_{n,1:k-1} \right] \right). \] (5.25)

Note that the first and second terms in (5.25) are signal and residual noise components of the equalizer output, respectively. As long as the channel state remains constant and the coefficients \( f_{n,k} \) and \( g_{n,k} \) are in a steady state, these terms are assumed to be stationary. During each subblock, we can estimate \( \mu_{n,k} \) and \( \sigma^2_{n,k} \) by time-averaging:
\[ \hat{\mu}_{k}^{\text{LMS-DA}} = \frac{1}{B} \sum_{i=(t-1)B+1}^{tB} \frac{\hat{s}_{i,k}^{\text{LMS-DA}}}{Q (\hat{s}_{i,k}^{\text{LMS-DA}})} \] (5.26)
\[ (\hat{\sigma}_{k}^{\text{LMS-DA}})^2 = \frac{1}{B} \sum_{i=(t-1)B+1}^{tB} \left| \hat{s}_{i,k}^{\text{LMS-DA}} - \hat{\mu}_{k}^{\text{LMS-DA}} Q (\hat{s}_{i,k}^{\text{LMS-DA}}) \right|^2, \] (5.27)
where \( \hat{\mu}_{k}^{\text{LMS-DA}} \) and \( (\hat{\sigma}_{k}^{\text{LMS-DA}})^2 \) are the estimates of \( \mu_{n,k} \) and \( \sigma^2_{n,k} \) for the \( t \)th subblock. Then, we compute the extrinsic LLR for \( \hat{c}_{n,k}^d \) as
\[ L_{\text{ext}}^{\text{LMS-DA}} (\hat{c}_{n,k}^d) = \ln \frac{\sum_{\theta \in \Theta q+1} \exp \left( -\frac{|s_{n,k}^{\text{LMS-DA}} - \hat{\mu}_{k}^{\text{LMS-DA}} \theta|^2}{(\hat{\sigma}_{k}^{\text{LMS-DA}})^2} \right)}{\sum_{\theta \in \Theta q-1} \exp \left( -\frac{|s_{n,k}^{\text{LMS-DA}} - \hat{\mu}_{k}^{\text{LMS-DA}} \theta|^2}{(\hat{\sigma}_{k}^{\text{LMS-DA}})^2} \right)}. \] (5.28)

5.6 Experimental Results

In this section, we evaluate the performance of the LMS DA-TEQ-based receiver in real underwater environments. The SPACE 08 experiment was conducted off the coast of Martha’s Vineyard, MA, during Oct. 14th - Nov 2nd, 2008. We compare the performance of the LMS DA-TEQ with that of the conventional LMS decision feedback equalizer (LMS-DFE) for a variety of configurations. In addition, the performance of the CE-based MMSE-TEQ
5.6.1 Experiment Description

Over the whole experiment, we employ a $R_c = 1/2$ rate (131, 171) RSC code and a random interleaver of the size of 19200–38400 bits for data generation. Four transducers spaced by 50 cm are used for signal transmission and each of them has a frequency bandwidth of 9 kHz ($W = 4.5 \times 10^3$). The transducers and hydrophone array are all located on fixed tripods and hence there is little movement of the transmitter and receiver. The carrier frequency is set to 13 kHz. The sampling rate of the transducer is 39.0625 ksamples/second and anti-aliasing filters with a cut-off frequency of about 18 kHz are used. Four one-minute data files are transmitted every two hours and data transmissions are carried out 149 times during 14 days. For convenience, each transmission is indexed with “epoch 1–149”. Each of four data files contains 6 “chunks” wherein each chunk corresponds to 8-second samples of the received data (associated with each configuration). The transmitted data is received by several hydrophone arrays which are located at the range of 60, 200, and 1000 m. Once data transmission is finished every two hours, ambient noise is recorded separately.

5.6.2 Parameter Setup

A square-root raised cosine filter with a roll-off factor $\beta = 0.2$ is used both in the transmitter and receiver. A frame consists of 6 set of training symbols and data symbols. Turbo iteration is carried out on a frame-by-frame basis. The length of training period is $M_t = 400$ and the length of data period is $M_d = 1600$. The length of preamble period is $M_p = 1000$. The preamble data is inserted at the start of each chunk. The data rate of the system can be obtained as

$$\frac{M_d}{M_t + M_d} \frac{39.06 \cdot 10^3}{OVF} N_t Q R_c, \quad (5.29)$$

In the SPACE 08 experiment, each epoch is distinguished by (Julian) date and time (in GMT) of the start of transmission. For example, the first epoch data corresponds the data transmitted on the date 288 and the time 00:00. For the sake of brevity, we use the epoch index from 1 to 149 in the sequel.
where OVF is the oversampling factor that represents the number of samples per symbol, i.e., sampling rate = OVF × symbol rate.\(^3\) The system bandwidth is given by \(\min((1+\beta)39.06/2\text{OVF}, 4.5)\) kHz. Hence, the over-Nyquist setup corresponds to the case where the symbol rate is larger than twice of bandwidth, i.e., \(39.06/\text{OVF} > 9\) Hz, equivalently for \(\text{OVF} = 2, 3\). The size of a sub-block is set to \(B = 200\) (symbol periods) and a total number of repetitions of the weight update is set to \(J = 5\). The initial step size is set to \(\xi_{\text{init}} = 0.01\) and it is decreased by \(\rho = 0.8\) in every pass. The value of \(P\) for sparse equalization is set to 20\%. We choose the equalizer length such that \(L_p = K_f + 7\) and \(L_f = K_p + 7\), where \(K_f + K_p + 1\) are a maximum channel length. A noise power is estimated during the silent period.

### 5.6.3 Experimental Results

#### Underwater Channel Responses versus Distance

In Figure 5.9 and 5.10, underwater channel responses are shown for several distances. A recursive least square (RLS) channel estimator with a forgetting factor of 0.999 is employed. The symbol period is 100 \(\mu\)s. We observe that the main arrival paths appear at around 2 ms delay. The channel gains for the secondary arrivals fluctuate more rapidly. In addition, the time-variation gets more significant for longer distance. On the other hand, for the 60 meter channel, the channel gains appear less fluctuating and the response looks sparse. For most cases, the actual channel span was around 6-7 ms corresponding to 60-70 channel taps for 9.77 ksym/s transmission.

#### Iterative Behavior of TEQ Receiver

First, we take a look at how performance of the TEQ algorithms behaves with iteration in real underwater channels. To evaluate the performance over different SNR levels, the separately recorded ambient noise is added to the received data. The symbol rate (\(=1/\text{symbol period}\)) is 9.77 ksym/s. Only 1000 m data for the epoch 24 is used for performance evaluation. Figure 5.11 and 5.12 show the BER performance versus SNR for the \(10 \times 1\) SIMO QPSK.

---

\(^3\)For data transmission over a frequency band of \([−W, W]\) Hz, we define a bandwidth of the system as \(W\) Hz.
Figure 5.9: Magnitude of channel impulse responses over time-delay domain for the distance of (a) 60 m and (b) 200 m.
Figure 5.10: Magnitude of channel impulse responses over time-delay domain for the distance of 1000 m.

The 10 × 1 SIMO 16-QAM, and the 10 × 2 MIMO QPSK transmissions, respectively. The data rates achieved by these systems are 3.66 kbit/s for the 10 × 1 QPSK transmission and 7.32 kbit/s for both 10 × 1 16-QAM and 10 × 2 QPSK transmissions. For comparison, the performance of the conventional LMS-DFE [90] is included. The performance LMS DA-TEQ algorithm dramatically improves with iterations so that significant performance gain is achieved over the conventional LMS-DFE algorithm after the 6 iterations. The LMS DA-TEQ outperforms the LMS-DFE even before the first iteration due to the LMS-DFE suffering from severe error propagation in the decision-feedback process, especially for long channel responses. Comparing the 10 × 1 16-QAM and 10 × 2 QPSK setups achieving the same data rate (=7.32 kbit/s), the performance of the LMS DA-TEQ algorithm is better for the 10 × 2 QPSK case. This suggests that increasing spatial dimension offers better spectral efficiency than increasing modulation order. However, the performance for the 10 × 2 MIMO setup converges at the slowest speed among three cases.

\[\footnote{In this LMS-DFE algorithm, we did not limit the number of symbols that are fed back to the equalizer. Instead, the actual symbols to be fed back are chosen depending on a sparse structure of underwater channels. In fact, all post-cursor symbols are considered as a candidate for feedback.}\]
Figure 5.11: Performance of LMS-DFE and LMS DA-TEQ for (a) $10 \times 1$ SIMO QPSK and (b) $10 \times 1$ SIMO 16-QAM transmissions.
Figure 5.12: Performance of LMS-DFE and LMS DA-TEQ for (a) 10 x 2 MIMO QPSK transmissions.

Long-Term Experimental Results

Next, we present more extensive results measured for long time duration (the epochs from 9 to 31). The symbol rate is fixed to 9.77 ksym/s, i.e., OVF = 4. Tables 5.1-5.3 present the BER performance for every four epochs of three equalizers: (1) LMS-DFE, (2) LMS DA-TEQ, and (3) CE-based MMSE-TEQ.\(^5\) The environment logs including wave height and wind speed are also provided in Table 5.4.

Different sizes of hydrophone arrays are used for each distance data, i.e., \( N_r = 6, 8, \) and 12 for 60, 200, and 1000 m, respectively. As for the LMS DA-TEQ and CE-based MMSE-TEQ, the performance is measured after the 6th iteration. Due to the tremendous amount of simulation time in the CE-based MMSE-TEQ algorithm, we could not provide the results for all setups. In all cases considered, the LMS DA-TEQ algorithm outperforms the conventional LMS-DFE by more than an order of magnitude. In many epochs, the LMS DA-TEQ could decode the data without errors while the LMS-DFE could not. In particular, the performance gain is remarkable for the \( N_r \times 2 \) MIMO setup where the problem size is twice that in the other setups. In most cases, the performance of the LMS DA-TEQ algorithm is comparable to that of the CE-based MMSE-TEQ algorithm. For some setups, the CE-based MMSE-

\(^5\)The BER in the tables is averaged over each set of four epochs.
Table 5.1: Experimental results for 60 m distance ($N_r = 6$).

<table>
<thead>
<tr>
<th>System</th>
<th>Setup</th>
<th>Data rate (kbit/s)</th>
<th>LMS-DFE</th>
<th>LMS DA-TEQ</th>
<th>CE MMSE-TEQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epoch 9 - 12</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$4.21 \times 10^{-4}$</td>
<td>0</td>
<td>$1.98 \times 10^{-4}$</td>
</tr>
<tr>
<td>Epoch 13 - 16</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>$2.97 \times 10^{-7}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$1.56 \times 10^{-4}$</td>
<td>$5.38 \times 10^{-4}$</td>
<td>$6.25 \times 10^{-4}$</td>
</tr>
<tr>
<td>Epoch 17 - 20</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>$1.77 \times 10^{-4}$</td>
<td>0</td>
<td>$1.12 \times 10^{-4}$</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$1.92 \times 10^{-4}$</td>
<td>$1.67 \times 10^{-4}$</td>
<td>$1.5/ \times 10^{-4}$</td>
</tr>
<tr>
<td>Epoch 21 - 24</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>$9.48 \times 10^{-8}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$1.31 \times 10^{-4}$</td>
<td>$4.67 \times 10^{-4}$</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 25 - 28</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>$1.71 \times 10^{-8}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$1.82 \times 10^{-4}$</td>
<td>$2.31 \times 10^{-4}$</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 29 - 31</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$1.13 \times 10^{1}$</td>
<td>$4.73 \times 10^{3}$</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 5.2: Experimental results for 200 m distance ($L = 8$).

<table>
<thead>
<tr>
<th>System</th>
<th>Setup</th>
<th>Data rate (kbit/s)</th>
<th>LMS-DFE</th>
<th>LMS DA-TEQ</th>
<th>CE MMSE-TEQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epoch 9 - 12</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$2.03 \times 10^{-2}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 13 - 16</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$1.66 \times 10^{-3}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 17 - 20</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$1.96 \times 10^{-4}$</td>
<td>0</td>
<td>$1.55 \times 10^{-4}$</td>
</tr>
<tr>
<td>Epoch 21 - 24</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$5.17 \times 10^{-2}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 25 - 28</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$6.82 \times 10^{-3}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 29 - 31</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$1.32 \times 10^{-2}$</td>
<td>0</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Table 5.3: Experimental results for 1000 m distance ($L = 10$).

<table>
<thead>
<tr>
<th>System</th>
<th>Setup</th>
<th>Data rate (kb/s)</th>
<th>LMS-DFE</th>
<th>LMS DA-TEQ</th>
<th>CE MMSE-TEQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epoch 9 - 12</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 1$ 16-QAM</td>
<td>7.32</td>
<td>$6.55 \times 10^{-4}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$3.70 \times 10^{-3}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 13 - 16</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>$2.38 \times 10^{-4}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 1$ 16-QAM</td>
<td>7.32</td>
<td>$1.75 \times 10^{-4}$</td>
<td>$5.26 \times 10^{-4}$</td>
<td>$1.30 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$8.48 \times 10^{-2}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 17 - 20</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>$7.66 \times 10^{-3}$</td>
<td>0</td>
<td>$1.72 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 1$ 16-QAM</td>
<td>7.32</td>
<td>$2.33 \times 10^{-4}$</td>
<td>$1.44 \times 10^{-4}$</td>
<td>$7.08 \times 10^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$2.22 \times 10^{-1}$</td>
<td>$1.17 \times 10^{-1}$</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 21 - 24</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>$2.38 \times 10^{-3}$</td>
<td>$3.79 \times 10^{-4}$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 1$ 16-QAM</td>
<td>7.32</td>
<td>$4.44 \times 10^{-4}$</td>
<td>$2.44 \times 10^{-4}$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$7.04 \times 10^{-2}$</td>
<td>$4.56 \times 10^{-2}$</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 25 - 28</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>$1.52 \times 10^{-4}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 1$ 16-QAM</td>
<td>7.32</td>
<td>$1.53 \times 10^{-4}$</td>
<td>$1.96 \times 10^{-2}$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$6.26 \times 10^{-2}$</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>Epoch 29 - 31</td>
<td>$N_r \times 1$ QPSK</td>
<td>3.66</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 1$ 16-QAM</td>
<td>7.32</td>
<td>$1.03 \times 10^{-4}$</td>
<td>$2.79 \times 10^{-2}$</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>$N_r \times 2$ QPSK</td>
<td>7.32</td>
<td>$2.57 \times 10^{-2}$</td>
<td>0</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 5.4: Experiment environments and conditions.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6933</td>
<td>0.6917</td>
<td>0.9583</td>
<td>0.6500</td>
<td>0.4417</td>
<td>0.4111</td>
</tr>
<tr>
<td>Wind speed (m/s)</td>
<td>1.1917</td>
<td>5.9803</td>
<td>6.0750</td>
<td>5.4017</td>
<td>6.0333</td>
<td>4.7111</td>
</tr>
</tbody>
</table>

98
TEQ algorithm is even worse than the LMS DA-TEQ, which is a counter-intuitive result according to our analysis in Section 5.4. This appears to be due to the inaccurate estimation of the covariance matrix of the noise plus residual signal in the CE-based MMSE-TEQ, which was not accounted for in the analysis. The CE-based MMSE-TEQ estimates $E[E_nE_n^H] + R$, using (5.11) and this step is performed only in the training period to keep the covariance matrix estimation errors and the data detection errors from causing error propagations. Due to the insufficient training symbols and inability to track the change of the covariance matrix, accurate estimate of covariance matrix could not be obtained. We observed that the CE-based MMSE-TEQ is, in particular, sensitive to various mismatches in terms of channel length estimation, frame alignment, and covariance matrix estimation compared to the LMS DA-TEQ.

Another interesting point observed in these results is that the performance of the equalizers for the 200 m distance is better than that for the 60 m distance. This observation can be explained by the shape of the channel responses shown in Figure 5.9 and 5.10. Though the channel for 60 m distance looks relatively calm, actual channel length is longer and magnitude of secondary arrivals is larger. This makes the 60 m channel more challenging. For 1000 m distance, the LMS DA-TEQ performs well for most of cases. However, the performance is not good during the epochs 17-20, where the wave height and wind speed are highest among all epochs (see Table 5.4).

It is shown again that the performance of the LMS DA-TEQ for the $N_r \times 2$ QPSK case is better that for the $N_r \times 1$ 16-QAM case (both setups support the same data rate). On the other hand, the performance of the conventional LMS-DFE is not satisfactory for both setups.

Performance Versus Symbol Rate

We look at how the throughput and performance of the LMS DA-TEQ vary in terms of different symbol rates and modulation orders. The BER averaged over a long period (epochs 109-114) is provided exclusively for the LMS DA-TEQ in Table 5.5. Only the $10 \times 1$ SIMO case and 1000 m distance are considered. As shown in the table, after five iterations, the LMS DA-TEQ can retrieve the data without errors for all BPSK and QPSK cases. This implies that our system can achieve 14.65 kbit/s rate (corresponding to the
Table 5.5: Performance of the LMS DA-TEQ for different symbol rates and modulations (1000 m).

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Symbol rate (ksym/s)</th>
<th>Data rate (kbit/s)</th>
<th>No Iteration</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
<th>Iteration 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td>6.51 (OVF=6)</td>
<td>2.44</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>9.77 (OVF=4)</td>
<td>3.66</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>19.53 (OVF=2)</td>
<td>7.32</td>
<td>0.0002</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>QPSK</td>
<td>6.51 (OVF=6)</td>
<td>4.88</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>9.77 (OVF=4)</td>
<td>7.32</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>19.53 (OVF=2)</td>
<td>14.65</td>
<td>0.310</td>
<td>0.004</td>
<td>0.002</td>
<td>0.0</td>
</tr>
<tr>
<td>16-QAM</td>
<td>6.51 (OVF=6)</td>
<td>9.77</td>
<td>0.01</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>9.77 (OVF=4)</td>
<td>14.65</td>
<td>0.0211</td>
<td>0.0138</td>
<td>0.0019</td>
<td>0.0098</td>
</tr>
<tr>
<td></td>
<td>19.53 (OVF=2)</td>
<td>29.27</td>
<td>0.48</td>
<td>0.49</td>
<td>0.51</td>
<td>0.55</td>
</tr>
</tbody>
</table>

case of QPSK, 19.53 ksym/s) with low error probability. On the other hand, for the 16-QAM case, we could achieve perfect data recovery for 6.51 ksym/s case and 0.01 BER for 9.77 ksym/s case. However, for higher symbol rate, the data decoding was not successful. Note that the QPSK, 19.53 ksym/s case corresponds to the over-Nyquist setup, where the symbol rate is larger than the system bandwidth 9 kHz (we mark these setups using * in the table). It is of particular interest to compare two cases: the QPSK, 19.53 ksym/s case and the 16-QAM, 9.77 ksym/s. The highest data throughput (14.65 kbit/s) is achieved in these setups. We observe that the over-Nyquist transmission yields better decoding performance than that with the increased modulation order and the LMS DA-TEQ could maintain good decoding performance even for over-Nyquist setups.

5.7 Conclusions

In this Chapter, we have investigated linear turbo equalizers for underwater acoustic communications. Specifically, we studied a low-complexity TEQ technique to the single-carrier equalization of underwater channels. The two linear TEQ techniques, CE-based MMSE-TEQ and LMS DA-TEQ, were considered. Clearly, the complexity of the LMS DA-TEQ is much lower than the CE-based MMSE-TEQ. We showed that both TEQ techniques would show comparable performance as the number of iterations increases. Inspired by small performance gap of both TEQs, we built the underwater receiver architecture based on the LMS DA-TEQ technique. According to the actual in-water experiments conducted off the coast Massachusetts, the LMS DA-
TEQ technique could achieve the data throughput of 15 kbit/s with low probability of error. In addition, the performance of the LMS DA-TEQ is an order of magnitude better than that of the conventional LMS decision feedback equalizer [86] for a variety of configurations such as different times, distances, and symbol rates.
CHAPTER 6

LOW-POWER FILTERING VIA MINIMUM POWER SOFT ERROR CANCELLATION (MP-SEC)

6.1 Introduction

Reliability and power efficiency in digital signal processing (DSP) systems are important yet often conflicting goals in complex systems. A wide variety of techniques have been developed in the last decade to reduce power in DSP systems [4, 5, 91–97]. In general, dynamic power dissipation in a DSP architecture is a quadratic function of the supply voltage, denoted $V_{dd}$, i.e.

$$P = C_L V_{dd}^2 f_s,$$  

(6.1)

where $C_L$ is the effective switching capacitance and $f_s$ is the clock frequency [4]. Due to the quadratic effect on power, a supply voltage reduction scheme, called dynamic voltage scaling, is often used to achieve significant power savings. Techniques to minimize $V_{dd}$ include variable voltage scaling [91,92], multiple supply voltages [93], and retiming techniques [94].

In practice, due to increased execution delay at reduced voltage, the extent of supply voltage reduction is limited by the worst case path delay in a given architecture. Specifically, a system is designed such that the critical path delay at the given supply voltage is less than the clock period to void timing errors. Therefore, existing voltage scaling methods [91–95] have performed supply voltage reduction up to the point that the critical path delays in the architecture and the sampling period are nearly equal. We refer to this as a critically scaled system, and the supply voltage as the critical supply voltage.

However, in [5], the authors suggested that the supply voltage might be scaled further, below the critical supply voltage for additional power savings, i.e.

$$V_{dd} = k_{vos} V_{dd\text{-crit}}, \quad 0 < k_{vos} < 1,$$  

(6.2)
where $V_{dd \text{- crit}}$ is the critical supply voltage. This technique, referred to as voltage overscaling (VOS), is motivated by the possibility of controlling the transient errors caused by timing violations, within a tolerable margin, via algorithmic noise tolerance [5]. These algorithmic errors are called soft errors and their mitigation is a key factor for enabling VOS-based DSP architectures.

Previous work to mitigate soft errors include: prediction-based error-correction methods [5], which estimate the current output sample of the system from previous samples by using a reduced-length forward linear predictor such that a corrupted sample can be replaced whenever an error is detected; reduced precision replica methods [96], which approximately calculate the current output to detect and correct errors; and adaptive error correction methods [97], which attempt to estimate soft errors directly in a minimum mean square error (MMSE) sense.

In this chapter, we propose a new soft error cancellation technique, called minimum power soft error cancellation (MP-SEC), which can detect, estimate and correct soft errors. A statistical detection and estimation problem is formulated for the soft errors. We show that the best, in an MMSE sense, unbiased linear estimator, followed by a local maximum likelihood (ML) detector, provides accurate estimates of soft errors, under some mild assumptions. This formulation enables power dissipation of the soft error canceller to be traded off against error resilience. For this setup, observable signals at the input and output of the main filter to be protected are collected as shown in Fig 6.1. While the main filter is operating in a VOS regime to reduce power, an error cancellation unit will not suffer soft errors for much of this regime, due to the reduced complexity of the MP-SEC units used to detect and correct any soft errors induced by VOS. Necessarily, the power consumed by the SEC unit will be small compared to the savings achieved through VOS. We explore the minimum power configuration for such an SEC unit and develop an adaptive power control algorithm, which optimizes the power dissipation of the SEC unit with respect to the selection of which observations to use and their numerical precision in the SEC unit.

An important observation that makes this approach possible is that soft errors can be characterized as discrete, i.e. finite alphabet, signals. As most arithmetic units perform least significant bit (LSB)-first computation, erroneous bits due to VOS will occur largely for bits near the most significant
Figure 6.1: Proposed MP-SEC soft error estimator applied to an FIR filter. The vector $x_n$ contains past and present samples of $x_n$ and the vector $\hat{y}_{n-1}$ contains past values of the sequence of $\hat{y}_n$.

bit (MSB). While this may seem problematic, any resulting soft errors will take on a small set of large-amplitudes as their possible outcomes and these possible amplitudes will be spaced apart, such that soft error estimation can be treated as an $M$-ary pulse amplitude modulation (PAM) signal detection problem [2].

The remainder of this chapter is organized as follows. In Section 6.2, we derive the soft error estimation and detection algorithm and provide performance analysis. In Section 6.3, we present a power-optimized algorithm for the soft error canceller. In Section 6.4, we describe a hardware design and some simulation results, and in Section 6.5, some conclusions are given.

6.2 Soft Error Cancellation Approach

In this section, we investigate statistical estimation and detection of soft errors. We first describe the framework where soft errors arise and their statistical description, and then derive the soft error estimator and detector.
Figure 6.2: A $4 \times 4$ carry-save multiplier for inputs (in two’s complement) $0111_2 \times 0011_2$. For this $B = 8$-bit example, the clock period $T_s$ is shown to be less than the critical path $T_6$. Three of the resulting bits become error-prone, while five of the bits are “safe.” ($B = 8$ and $M = 5$.)

6.2.1 Soft Error Model

By appropriate design, soft errors can be constrained to appear near the MSB in the binary representation of the signal samples for LSB-first arithmetic units used in many structures for computation. For example, we will assume a two’s complement number representation such that $x = -b_0 + \sum_{i=1}^{B-1} b_i2^{-i}$ in $B$-bit precision. As a simple illustration, consider the $4 \times 4$ carry-save multiplier shown in Figure 6.2. Let $T_s$ and $T_i$ be the sampling period and the worst path delay to $i$th output bit, respectively. When $k_{vos} = 1$ (no VOS), it is evident from the figure that $T_8 \geq \cdots \geq T_1$, due to the use of LSB-first computation. However, as we scale the supply voltage below $V_{dd-crit}$, the worst-case delays $T_i$ increase for all $i$, so that output bits become divided into two sets: error prone bits (EB) where the timing conditions may be violated ($T_i > T_s$) and safe bits (SB) where the timing relation is guaranteed. If we let $B$ and $M$ be the number of output bits and safe bits in the multiplier, then the soft errors are expressed as a combination of bits from the EB region, and their magnitudes become a multiple of $2^M/2^B$. This implies that the possible magnitudes of soft errors are equally spaced by $2^M/2^B$. This property, which
we refer to as a spacing property, plays a key role in estimating soft errors.

To illustrate the impact of soft errors on FIR filters, we consider an $N_1$-tap causal FIR filter whose direct-form I implementation is shown in Figure 6.3 (a). Under VOS, the processing units in the main filter violating the timing requirement may suffer from soft errors. As shown in Figure 6.3 (b), the system can be described by an equivalent linear additive model. In this model, a soft error, denoted $\alpha_{i,n}$ for the $i$th multiplier and $\beta_{j,n}$ for the $j$th adder, are injected at the output of each arithmetic unit. If soft errors do not appear, $\alpha_{i,n} = 0$ and $\beta_{j,n} = 0$. These sources of soft error can be collected together and merged into one signal source $e_n$ at the node $c'$, where $e_n$ is given by

$$e_n = \sum_{i=1}^{N_1} \alpha_{i,n} + \sum_{j=0}^{N_1-1} \beta_{j,n} + \gamma_n,$$

(6.3)

where $\gamma_n$ represents the errors which might occur due to overflow from the adders. Due to the spacing property, $e_n$ takes on values in

$$\Omega = \left\{ k2^{M-B} \bigg| k \in \mathcal{Z}, k \in [-2^B/2^M, 2^B/2^M) \right\},$$

where $B$ and $M$ are the numerical precision of output and the smallest number of SBs of all arithmetic units, respectively. The noisy output $z_n$ in the
presence of a soft error is given by

\[ z_n = y_n + e_n = \sum_{k=1}^{N_1} h_k x_{n-k+1} + e_n, \]  

(6.4)

where \( x_n \) and \( y_n \) are the \( n \)th sample of the input and the error-free output, respectively, and \( h_k \) is the \( k \)th FIR filter coefficient. In the sequel, we will use the vector notation, \( \mathbf{x}_n = [x_n, \ldots, x_{n-N_1+1}]^T \), \( \mathbf{y}_n = [y_{n-1}, \ldots, y_{n-N_2}]^T \) and \( \mathbf{h} = [h_1, \ldots, h_{N_1}]^T \), for convenience, where a boldface vector denotes a vector of random variables and an overbar denotes a deterministic vector.

### 6.2.2 Soft Error Cancellation

The objective of soft error cancellation is to subtract an estimate of the soft error from the erroneous output if necessary, i.e.,

\[ \hat{y}_n = z_n - \hat{e}_n = y_n + (e_n - \hat{e}_n), \]  

(6.5)

where \( \hat{y}_n \) is the error-corrected output, and \( \hat{e}_n \) is an estimate of the soft error. Hence, ideal soft error cancellation provides that \( (e_n - \hat{e}_n) \) is zero such that \( \hat{y}_n = y_n \).

#### Soft Error Estimation

Assume that the input and therefore the output signals, \( x_n \) and \( y_n \), are zero-mean stationary random processes. As mentioned in the previous section, the soft error estimator makes decisions based on the observations of a subset of the sets, \( \{z_n, \{x_n, \ldots, x_{n-N_1+1}\}, \{y_{n-1}, \ldots, y_{n-N_2}\}\} \), where the elements are selected to trade off performance of the estimation for the added power of the soft error cancellation. To limit the added complexity or power drawn by the estimator, we restrict the precision used to describe the sets \( \{x_n, \ldots, x_{n-N_1+1}\} \) and \( \{\hat{y}_{n-1}, \ldots, \hat{y}_{n-N_2}\} \) to only \( p \) bits, producing the vectors \( \mathbf{x}_{q,n} = [x_{q,n}, \ldots, x_{q,n-N_1+1}]^T \) and \( \mathbf{\hat{y}}_{q,n} = [\hat{y}_{q,n-1}, \ldots, \hat{y}_{q,n-N_2}]^T \), where \( x_n \) and \( \hat{y}_n \) are quantized to \( \mathbf{x}_{q,n} \) and \( \mathbf{\hat{y}}_{q,n} \). Then, we mask the vectors \( \mathbf{x}_{q,n} \) and \( \mathbf{\hat{y}}_{q,n} \) using the switching vectors \( \mathbf{c} = [c_1, \ldots, c_{N_1}]^T \) and \( \mathbf{d} = [d_1, \ldots, d_{N_2}]^T \) producing \( \mathbf{x}_{c,n} = [c_1 x_{q,n}, \ldots, c_{N_1} x_{q,n-N_1+1}]^T \) and \( \mathbf{\hat{y}}_{c,n} = [d_1 \hat{y}_{q,n-1}, \ldots, d_{N_2} \hat{y}_{q,n-N_2}]^T \).
where \( c_i \) and \( d_i \) take on values 0 or 1. As a result, the reduced observation takes the form
\[
\begin{bmatrix}
z_n \\
x_{c,n} \\
\hat{y}_{c,n}
\end{bmatrix} = \begin{bmatrix}
y_n \\
x_{c,n} \\
\hat{y}_{c,n}
\end{bmatrix} + e_n \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.
\] (6.6)

Note that the resolution and selection of observations are controlled by \( p, \bar{c} \) and \( \bar{d} \). For the time being, we assume that \( p, \bar{c} \) and \( \bar{d} \) are given. Based on these reduced observations, a linear (affine) unbiased estimate of \( e_n \) can be expressed
\[
\tilde{e}_n = a_0 z_n + \left[ \begin{array}{c} \bar{w}^T \\ \bar{\nu}^T \end{array} \right] \begin{bmatrix} x_{c,n} \\ \hat{y}_{c,n} \end{bmatrix} + A,
\] (6.7)

where \( \bar{w}^T = [w_1, \ldots, w_{N_1}]^T \) and \( \bar{\nu}^T = [v_1, \ldots, v_{N_2}]^T \). To satisfy the unbiased constraint, \( E[\tilde{e}_n] = e_n \) for all \( e_n \in \Omega \), \( a_0 = 1 \), and
\[
A = - \left[ \begin{array}{c} \bar{w}^T \\ \bar{\nu}^T \end{array} \right] E \begin{bmatrix} x_{c,n} \\ \hat{y}_{c,n} \end{bmatrix},
\] (6.8)

where \( E[\cdot] \) denotes a statistical expectation. The vectors \( \bar{w} \) and \( \bar{\nu} \) are determined to minimize the variance of the estimate,
\[
E \left[ (\tilde{e}_n - e_n)^2 \right] = E \left[ \left( y_n + \left[ \begin{array}{c} \bar{w}^T \\ \bar{\nu}^T \end{array} \right] \begin{bmatrix} x_{c,n} \\ \hat{y}_{c,n} \end{bmatrix} + A \right)^2 \right].
\] (6.9)

The estimator coefficients \( \bar{w} \) and \( \bar{\nu} \) are obtained by finding a linear minimum mean square error (LMMSE) estimate of \( y_n \) based on \( x_{c,n} \) and \( \hat{y}_{c,n} \). Substituting (6.8) into (6.9), we write this variance as
\[
E \left[ (\tilde{e}_n - e_n)^2 \mid \bar{c}, \bar{d} \right] = E \left[ \left( y_n + \left[ \begin{array}{c} \bar{w}_c^T \\ \bar{\nu}_c^T \end{array} \right] \begin{bmatrix} x_{q,n} \\ \hat{y}_{q,n} \end{bmatrix} - E \begin{bmatrix} x_{q,n} \\ \hat{y}_{q,n} \end{bmatrix} \right)^2 \right],
\] (6.10)

where \( \bar{w}_c = [c_1 w_1, \ldots, c_{N_1} w_{N_1}]^T \), and \( \bar{\nu}_c = [d_1 v_1, \ldots, d_{N_2} v_{N_2}]^T \). Note that the entries \( w_{c,i} \) and \( v_{c,j} \) of \( \bar{w}_c \) and \( \bar{\nu}_c \) are constrained to be zero when \( c_i \) and \( d_j \) are zero. The coefficients of \( \bar{w}_c \) and \( \bar{\nu}_c \), that minimize (6.10) subject to
the coefficient constraint, are given by

\[
\begin{bmatrix}
w_{c,i_1} \\
\vdots \\
w_{c,i_{N_1'}} \\
v_{c,j_1} \\
\vdots \\
v_{c,j_{N_2'}}
\end{bmatrix} = - \text{Cov} \left( \begin{bmatrix} x_{q,n-i_1+1} \\ \vdots \\ x_{q,n-I_1'+1} \\ \hat{y}_{q,n-j_1} \\ \vdots \\ \hat{y}_{q,n-j_{N_2'}} \end{bmatrix}, \mathbf{x}_n \right) \mathbf{h},
\]

(6.11)

where \(i_1 \cdots i_{N_1'} \in \Lambda_1 = \{i : c_i = 1\}\), \(j_1 \cdots j_{N_2'} \in \Lambda_2 = \{j : d_j = 1\}\), and \(\text{Cov}(a, b) = \mathbb{E}[ab^T] - \mathbb{E}[a] \mathbb{E}[b]^T\) and \(\text{Cov}(a) = \mathbb{Cov}(a, a)\). Note that \(w_{c,i} = 0\) for \(i \in \Lambda_c^1\) and \(v_{c,j} = 0\) for \(j \in \Lambda_c^2\), where the superscript \(c\) denotes a set complement. Let the quantization error be \(\Delta x_n = x_n - x_{q,n}\) and \(\Delta \hat{y}_n = \hat{y}_n - \hat{y}_{q,n}\). When a random variable \(x\) has a two’s complement representation with independent and identically distributed (i.i.d.) bits and is truncated from \(B\) bits to \(p\) bits, the mean and variance of the resulting quantization errors can be shown to be given by

\[
\mu_{\Delta x} = 2^{-p} - 2^{-B}
\]

(6.12)

\[
\text{Var}_{\Delta x} = \frac{1}{3} \left( 2^{-2p} - 2^{-2B} \right).
\]

(6.13)

When the observations are quantized from \(B\) bits to \(p\) bits, and using (6.12), the parameter \(A\) is given by

\[
A = \left( 2^{-p} - 2^{-B} \right) \left( \sum_{i=1}^{N_1} w_{c,i} + \sum_{i=1}^{N_2} v_{c,i} \right),
\]

(6.14)
and using (6.13), the coefficients $w_{c,i_1}, \ldots, w_{c,i_{N_1}'}$ and $v_{c,j_1}, \ldots, v_{c,j_{N_2}'}$ are given by

$$
\begin{bmatrix}
w_{c,i_1} \\
\vdots \\
w_{c,i_{N_1}'} \\
v_{c,j_1} \\
\vdots \\
v_{c,j_{N_2}'}
\end{bmatrix} = - \text{Cov} \begin{pmatrix}
x_{n-i_1+1} \\
\vdots \\
x_{n-i_{N_1}'+1} \\
y_{n-j_1} \\
\vdots \\
y_{n-j_{N_2}'}
\end{pmatrix} + \frac{1}{3} \left(2^{-2p} - 2^{-2B}\right) I
$$

where $I$ is an $(N_1' + N_2')$-by-$(N_1' + N_2')$ identity matrix, and it has been assumed that $\Delta x_n$ and $\Delta \hat{y}_n$ are mutually uncorrelated and uncorrelated with $x_n$ and $\hat{y}_n$, which is reasonable for moderate values of $p$. The coefficients $\overline{w}_c$ and $\overline{v}_c$ can be implemented with linear filters, and hence we refer to $\overline{w}_c$ and $\overline{v}_c$ as the main estimation filter (MEF), and specifically to $\overline{w}_c$ as the feed-forward MEF (FF-MEF) and to $\overline{v}_c$ as the feed-back MEF (FB-MEF), respectively.

The resulting soft error estimate is given by

$$
\tilde{e}_n = z_n + \bigg[\overline{w}_c^T \overline{v}_c^T\bigg] \begin{bmatrix}x_{q,n} \\
y_{q,n}\end{bmatrix} + A.
$$

Although we have obtained an unbiased estimate of $e_n$, it will not generally satisfy the constraint that the estimate of $e_n$ lie in $\Omega$, while we know that the true $e_n$ must lie in $\Omega$. We consider

$$
\tilde{e}_n = e_n + \left(y_n + \bigg[\overline{w}_c^T \overline{v}_c^T\bigg] \begin{bmatrix}x_{q,n} \\
y_{q,n}\end{bmatrix} + A\right),
$$

where we refer to the term in (6.17) in braces as the estimation error of the MMSE estimator, which will be called the residual error. Given the
unbiased estimate $\tilde{e}_n$, we may be able to refine the estimate by maximizing the log-likelihood function of $\tilde{e}_n$ with respect to $e_n$, i.e.,

$$\hat{e}_n = \arg \max_{e_n \in \Omega} \ln p(\tilde{e}_n; e_n). \quad (6.18)$$

When the PDF of the residual error is symmetric about zero and unimodal, then a local maximum likelihood estimate (MLE) of soft error is given by

$$\hat{e}_n = T(\tilde{e}_n) \quad (6.19)$$

where

$$T(x) = \frac{2^M}{2^B} i, \quad \text{where} \quad \left| x - \frac{2^M}{2^B} i \right| \leq \left| x - \frac{2^M}{2^B} j \right|, \quad \text{for all} \quad j \quad (6.20)$$

for $i$ and $j$ are integers between $-2^B/2^M$ and $2^B/2^M - 1$. This MLE is based on the statistic $\tilde{e}$, not on $[z_n, x_n, \hat{y}_n]^T$. We see that $\hat{e}_n$ would become the MLE obtained based on $[z_n, x_n, \hat{y}_n]^T$, if $z_n$, $x_n$, and $\hat{y}_n$ were jointly Gaussian and not truncated. In the general case, the estimate $\hat{e}_n$ would be suboptimal, but can be practically implemented with low complexity.

Based on (6.17) and (6.19), the total estimation error $\hat{e}_n - e_n$ is given by

$$\hat{e}_n - e_n = T \left( y_n + \begin{bmatrix} \bar{w}^T_c & \bar{v}^T_c \end{bmatrix} \begin{bmatrix} x_{q,n} \hat{y}_{q,n} \end{bmatrix} + A \right) \quad (6.21)$$

Equation (6.21) implies that when the residual error is smaller than $2^M/2^B$, the estimate would be accurate, i.e. $\hat{e}_n = e_n$. This means that wider soft error spacing leads to better estimation. We may be able to reduce the variance of residual error by increasing $p$, i.e., the resolution of the quantizer or the number of ones in $\bar{c}, \bar{d}$. Accordingly, we can assume that we set the values of $\bar{c}, \bar{d}$ and $p$ such that

$$E \left[ (\hat{e}_n - e_n)^2 \right] \ll \sigma^2_y, \quad (6.22)$$

where $\sigma^2_y$ is the variance of $y_n$. As long as this assumption holds, the impact of soft error correction errors on the statistics of $y_n$ can be assumed to be negligible. This low-error regime allows the use of $y_{n-k}$, instead of $\hat{y}_{n-k}$, in
the estimate formulation (6.15).

The soft error estimator derived herein consists of two parts: (1) the feed-forward and feedback linear filters which enhance the quality of estimation, and (2) the sequent maximum likelihood detector (MLD) which maps the input to the nearest soft error candidate, as shown in Figure 6.4. This estimation mechanism shows an interesting analogy to that of a PAM receiver which consists of a matched filter correlator that improves the estimation SNR and the maximum likelihood symbol detector.

Soft Error Detection

To determine when an error has occurred and enable error correction, the following hypothesis test for soft error detection is used:

\[
\begin{align*}
H_1 & : z_n = \mathbf{h}^T \mathbf{x}_n + e_n \\
H_0 & : z_n = \mathbf{h}^T \mathbf{x}_n,
\end{align*}
\]

where \(e_n\) takes a value from \(\Omega\). As the parameter, \(e_n\) is unknown, this problem can be interpreted as a composite hypothesis test for which a generalized likelihood ratio test (GLRT) is often used [61]. Subject to complexity constraints, we may also base the detection on the test statistic, \(\tilde{e}_n\), not on \([z_n, \mathbf{x}_n, \mathbf{y}_n]^T\). We compare the log-likelihood ratio, maximized over \(e_n\) with a threshold, \(\tau\), i.e.,

\[
\Lambda = \max_{e_n \in \Omega} \ln \frac{p(\tilde{e}_n | H_1; e_n)}{p(\tilde{e}_n | H_0)} \overset{H_1}{\geq} \tau, \quad (6.24)
\]
where \( \tau \) may be chosen using a constant false alarm rate (CFAR) criterion. To simplify the development, we assume the condition (6.22), and that the residual error is well approximated by a zero mean Gaussian, then the maximizer of the log-likelihood ratio becomes the MLE given in (6.19), and we can substitute \( e_n \) by \( \hat{e}_n \) in (6.24). The resulting approximated detection rule is given by

\[
\frac{\hat{e}_n^2 - (\hat{e}_n - \hat{e}_n)^2}{\sigma_r^2} \begin{cases} H_1 & \tau \geq 2 \sigma_r^2, \\ H_0 & \end{cases}
\]

(6.25)

where \( \sigma_r^2 \) is the variance of the residual error and is given by

\[
\sigma_r^2 = \sigma_y^2 - \left[ \frac{w_c}{v_c} \right]^T \left( \text{Cov} \left[ \frac{x_n}{y_n} \right] + \frac{1}{3} \left( 2^{-2p} - 2^{-2B} \right) I \right) \left[ \frac{w_c}{v_c} \right].
\]

(6.26)

In practice, this soft error detector may not be necessary when the error spacing is large, because the quantization function \( T(\cdot) \) in (6.19) performs the task of detecting the errors if the residual error exceeds \( 2^M/2^B \).

Algorithmic Performance Measure

In order to analyze the performance of the soft error canceller, we use the signal power to soft error power ratio (SSR), defined as

\[
\text{SSR} = \frac{\text{power of desired signal}}{\text{power of residual soft error}}.
\]

(6.27)

From (6.5), the SSR at the output of the main filter is given by

\[
\text{SSR} = 10 \log_{10} \frac{\sigma_y^2}{E \left[ (\hat{e}_n - e_n)^2 \right]}.
\]

(6.28)

We may also use other measures such as the signal to noise ratio (SNR). As an example, consider an application in which the output signal \( y_n \) consists of both a desired signal \( d_n \) and an undesired noise signal \( \eta_n \), i.e.,

\[
y_n = d_n + \eta_n.
\]

(6.29)
After applying the soft error canceller, the SNR is given by

$$\text{SNR} = 10 \log_{10} \frac{\sigma_d^2}{\sigma_n^2 + E[(\hat{e}_n - e_n)^2]}.$$  \hspace{1cm} (6.30)

Note that both measures in (6.28) and (6.30) depend on the power of the estimation error, \(\hat{e}_n - e_n\), or residual mean square error (RMSE). Hence, the algorithmic performance of MP-SEC can be thoroughly analyzed by deriving the RMSE.

RMSE Analysis

In this subsection, we provide an analysis of RMSE when employing the soft error estimator. In deriving the estimate, we neglected the effect of previous decisions assuming the condition (6.22) to hold. However, in practice, inaccurate estimates of soft errors may cause subsequent errors in estimating soft errors, permitting error propagation. Hence, we need to analyze the RMSE considering the consequences of using tentative decisions.

First, we assume that the soft error detector is not employed. We assume that the residual error has a Gaussian distribution \(\mathcal{N}(0, \sigma_r^2)\) for analysis. If the previous errors are essentially correct, the probability that the soft error estimate is not correct is

$$P(\hat{e}_n \neq e_n) = 1 - 2Q\left(\sqrt{\frac{\lambda^2}{4\sigma_r^2}}\right),$$  \hspace{1cm} (6.31)

where \(Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp(-t^2/2)\), and \(\lambda = 2^M/2^B\). Note that the probability of error will increase as \(\lambda\) increases or \(\sigma_r^2\) decreases. Hence, the ratio, \(\lambda^2/\sigma_r^2\) is a crucial factor that affects the quality of soft error estimation, even when error propagation happens. The RMSE is expressed as

$$E[(e_n - \hat{e}_n)^2] = \sum_{k=-\infty}^{\infty} (\lambda k)^2 P(\hat{e}_n - e_n = \lambda k).$$  \hspace{1cm} (6.32)
When ignoring the effect of previous decision errors, the RMSE is given by

\[
RMSE = 2 \sum_{k=1}^{\infty} \lambda^2 k^2 \left\{ Q \left( \frac{\lambda^2}{\sigma_r^2} \left( k - \frac{1}{2} \right) \right) - Q \left( \frac{\lambda^2}{\sigma_r^2} \left( k + \frac{1}{2} \right) \right) \right\}. \tag{6.33}
\]

Note that RMSE also increases when \( \sigma_r^2 \) increases.

Now, assume that the previous decision errors are no longer negligible. We use a Markov chain model to describe the sequence of previous errors, and evaluate the RMSE in the steady state. This approach has been employed in the analysis of decision feedback equalization [98, 99]. Let the estimation error at time \( n - k \) be \( s_{n-k} \), i.e., \( s_{n-k} = e_{n-k} - \hat{e}_{n-k} \). Without loss of generality, we consider the feedback errors to come from the length \( N_2 \) sequence, \( \hat{y}_{n-1}, \cdots, \hat{y}_{n-N_2} \). Then, we define the state \((i_1, \cdots, i_{N_2})\) at time \( n \) to be

\[
\text{state}_n(i_1, \cdots, i_{N_2}) = \{ s_{n-1} = i_1 \lambda, \cdots, s_{n-N_1} = i_{N_2} \lambda \}. \tag{6.35}
\]

where \( i_1, \cdots, i_{N_2} \) are integers in \([-2 \cdot 2^{B-M} + 1, 2 \cdot 2^{B-M} - 1]\]. The number of possible decision errors is \((4 \cdot 2^{B-M} - 1)\), and the total number of states should be \((4 \cdot 2^{B-M} - 1)^{N_2}\). However, for moderate configurations of the estimator, the possibility that a decision error occurs with large magnitude is small as shown in Figure 6.5, which is derived under modest assumptions in the following analysis. This reduces the number of states by counting only \(2m + 1(< 2^{B-M})\) error magnitudes near zero, where \( m \) is a small integer, and hence the total number of states becomes \((2m + 1)^{N_2}\). Then, we can present the following conditional probabilities:

\[
P (s_n = i \lambda | \text{state}_n(i_1, \cdots, i_{N_2})) = Q \left( \frac{(2i - 1) \lambda - 2 \sum_{k=1}^{N_2} v_{c,k} i_k \lambda}{2\sigma_r} \right) - Q \left( \frac{(2i + 1) \lambda - 2 \sum_{k=1}^{N_2} v_{c,k} i_k \lambda}{2\sigma_r} \right), \tag{6.36}
\]

\((i = -m + 1, \cdots, m - 1)\).
Figure 6.5: Analytically modeled PDF of the estimation error, or $\hat{e}_n - e_n$, when $\lambda^2/\sigma_r^2$ is (a) 0 dB, (b) 4 dB, (c) 8 dB, and (d) 12 dB. Note that the time-axis is scaled by $1/\lambda$. 
and

\[
P(s_n = \pm m \lambda | \text{state}_n (i_1, \cdots, i_{N_2}))
= Q \left( \frac{(\pm 2m - 1) \lambda \mp 2 \sum_{k=1}^{N_2} v_{c,k} i_k \lambda}{2 \sigma_r} \right).
\] (6.37)

We list from \(\text{state}_n (-m, \cdots, -m)\) to \(\text{state}_n (m, \cdots, m)\) in an appropriate but arbitrary order and number them from \(n = 1\) to \(n = (2m + 1)^{N_2}\). The \((2m + 1)^{N_2}\)-by-\((2m + 1)^{N_2}\) state-transition matrix \(T\) from time \(n\) to time \(n+1\) is given by

\[
T (\text{state}_n (i_1, \cdots, i_{N_2}), \text{state}_{n+1} (j_1, \cdots, j_{N_2})) =
\begin{cases} 
P (s_n = j_1 \lambda | \text{state}_n (i_1, \cdots, i_{N_2})) , \\
\text{if } i_1 = j_2, i_2 = j_3, \cdots, i_{N_2-1} = j_{N_2} \\
0, \text{ otherwise}.
\end{cases}
\]

Let the probability of staying in the \(i\)th state in the steady state be \(\pi_i\). The vector, \([\pi_1, \cdots, \pi_{(2m+1)^{N_2}}]\) can be found by solving the following simultaneous equations:

\[
\begin{bmatrix} 
\pi_1 & \cdots & \pi_{(2m+1)^{N_2}} 
\end{bmatrix} T = \begin{bmatrix} 
\pi_1 & \cdots & \pi_{(2m+1)^{N_2}} 
\end{bmatrix},
\]

\[
\sum_{k=1}^{(2m+1)^{N_2}} \pi_k = 1.
\] (6.39)

The solution to these equations is obtained by finding the null vector of the matrix \((T^T - I)\) and normalizing the null vector to satisfy (6.39). The probability that the decision error equals \(\lambda k\) in the steady state is given by

\[
P (\hat{e}_n - e_n = \lambda k) = \sum_{\{j: i_1 \text{ of } j\text{th state} = k\}} \pi_j.
\] (6.40)

for \(-m \leq k \leq m\).

Figure 6.5 shows the PDFs of \(\hat{e}_n - e_n\) using this result, in the case that two feedback observations are employed, i.e., \(N_2 = 2, B = 16\) and \(M = 12\). Note that the PDF is defined on the discrete event space due to the soft error spacing property and more concentrated around zero with increasing \(\lambda^2/\sigma_r^2\). Finally, the RMSE can be obtained by substituting (6.40) into (6.32). In
Figure 6.6: Analytically derived and simulated RMSEs are shown versus \( \lambda^2/\sigma^2_r \).

Figure 6.6, we plot the RMSE versus \( \lambda^2/\sigma^2_r \) when \( N_2 \) is 2 and 3. The RMSE decreases as \( \lambda^2/\sigma^2_r \) increases and as \( N_2 \) decreases. The experimental values of RMSE are close to the analytically derived curves. It should be noted that the performance of the soft error canceller depends on \( \sigma^2_r \), or equivalently how well the MMSE estimator estimates the desired output \( y_n \) using the given information inferred in the observations.

6.3 Energy Minimum Soft Error Cancellation

In this section, we introduce two approaches to optimize the power dissipation in the soft error canceller. First, an energy-minimum design is provided, which assumes stationary statistics of the signals of interest. The key feature of this strategy is enabling “one shot” design which can be fixed, e.g. in a VLSI design. Secondly, we introduce a dynamic power optimization technique, which controls the configuration of the SEC unit in real time to cope with time-varying environments.
6.3.1 Power Optimization Criterion

In general, the power dissipation $P_v$ of a system may follow

$$P_v \propto (C_0 + C_s) (k \times V_{dd-crit})^2$$

(6.41)

where $C_0$ and $C_s$ are the switched capacitances in the main filter block and SEC block, respectively. This relationship implies that for maximum power savings, we have to reduce the complexity of the SEC block as much as possible. A rather general approach to optimizing power consumption has been addressed in [100, 101] in the form of a constrained optimization that minimizes an estimate of the power dissipation subject to particular performance constraints. We express this formulation in a form relevant to our framework,

Minimize : $P_{SEC}(\bar{c}, \bar{d}, p)$
Subject to : $E[(\hat{e}_n - e_n)^2 | \bar{c}, \bar{d}, p] \leq D,$

(6.42)

where $D$ is the desired RMSE given as a system requirement, and $P_{SEC}(\bar{c}, \bar{d}, p)$ is the power consumed in the SEC unit. In the following, we search for a feasible solution to this problem with respect to $\bar{c}, \bar{d}$, and $p$.

6.3.2 Static Power-Optimum Design

In this subsection, we find the selection of observations and their precisions to optimize the constrained objective function in (6.42).

Observation Selection

An important result from the previous section is that the algorithmic performance of the system depends on the MSE of the MMSE estimator, or $\sigma_i^2$. Taking more observations leads to smaller $\sigma_i^2$, or a better quality soft error estimate, however it will require higher complexity, and thus higher power. Therefore, we need to limit the number of observations, and more systematically, we should select the best combination of observations on the basis of (6.42). In the following, we will describe an efficient observation selection method which uses a tree search procedure based on the branch and bound
principle [102,103]. Let \( \xi_m \) be the set of \( m \) samples of observations that are masked by the vector \( c \) and \( d \). The variance \( \sigma_r^2 \) based on the observation set, \( \xi_m \) is given by

\[
\sigma_r^2(\xi_m) = \sigma_y^2 - \text{Cov} \left( y_n, \left[ \xi_m \right] \right)^T \left( \text{Cov} \left( \left[ \xi_m \right] \right) \right)^{-1} \text{Cov} \left( y_n, \left[ \xi_m \right] \right),
\]

(6.43)

where \( \left[ \xi_m \right] \) is a column vector associated with \( \xi_m \). If the signal subsets \( \xi_1, \cdots, \xi_m \) are nested such that

\[
\xi_1 \subset \xi_2 \subset \cdots \subset \xi_m,
\]

(6.44)

then

\[
\sigma_r^2(\xi_1) \geq \sigma_r^2(\xi_2) \geq \cdots \geq \sigma_r^2(\xi_m),
\]

(6.45)

since \( (y_n - E[y_n|\xi_{m+1}]) \) and \( (E[y_n|\xi_{m+1}] - E[y_n|\xi_m]) \) are orthogonal so that

\[
\sigma_r^2(\xi_m) = \sigma_r^2(\xi_{m+1}) + E \left[ \left( E \left[ y_n|\xi_m \right] - E \left[ y_n|\xi_{m+1} \right] \right)^2 \right].
\]

(6.46)

On the other hand, the power dissipation \( P_{\text{SEC}} \), given \( c \) and \( d \), may be estimated via the multiplier energy model [100] or any available power modelling technique [104]. For the nested sets in (6.44), it is reasonable to assume that

\[
P_{\text{SEC}}(\xi_1) \leq P_{\text{SEC}}(\xi_2) \leq \cdots \leq P_{\text{SEC}}(\xi_m),
\]

(6.47)

where \( P_{\text{SEC}}(\xi_i) \) is the power dissipation associated with \( \xi_i \). The two monotonicity properties in (6.45) and (6.47) enable the use of the branch and bound technique to solve for the optimal observation selection.

As a simple example, assume that we select a combination from five candidates, denoted \( \{ a_1 = x_{q,n}, a_2 = x_{q,n-1}, a_3 = x_{q,n-2}, a_4 = \hat{y}_{q,n-1}, a_5 = \hat{y}_{q,n-2} \} \), which constitutes the full set of observations. We can construct a tree as shown in Figure 6.7. The search begins from the root with “no observation,” or \( [\bar{c}^T, \bar{d}^T] = [0, \cdots, 0]^T \), and traverses down when adding each “new observation.” Beginning from the rightmost branch, we calculate \( P_{\text{SEC}} \) and the resulting RMSE and save them at each branch (see “A” in Fig 6.7). We continue traversing the tree while decreasing RMSE, and stop when the
RMSE begins to be less than $D$. We call the first node at which the traversal stops the *initial best subset* and the corresponding $P_{SEC}$ a *bound*. Next, we search the left-side branches and cut off any branches and subbranches whose $P_{SEC}$’s are larger than the current bound. However, if a leaf of the tree is reached with no pruning, we replace the bound by the current $P_{SEC}$ and update the performance of the branch and bound with the current subset. In this example, the final best subset is \( \{ x_{q,n}, x_{q,n-2}, \hat{y}_{q,n-2} \} \).

In order to improve the algorithm, it is common to position the “good” signals, i.e., those that cause significant RMSE decrease, to the right hand side of the tree (note the signals at the first level ordered “$a_4, a_3, a_2, a_5, a_1$”). This ordering reduces the average search path. The initial best subset under this ordering provides a good, but not optimal, solution. Since the initial subset improves successively as the update proceeds, we can terminate the procedure early to obtain a good solution if the search time is excessive.

**Precision Selection**

The precision parameter $p$ has a significant impact on the power dissipation of the SEC block. To obtain the jointly optimal values of $\bar{c}$, $\bar{d}$, and $p$, we can construct the search tree for each value of $p$ over a nominal range. After finding the optimal $\bar{c}$ and $\bar{d}$ for each $p$, we select the $p$ which results in the minimum $P_{SEC}$ as the optimal $p$, and the corresponding $\bar{c}$ and $\bar{d}$ as optimal
switching vectors.

### 6.3.3 Dynamic Power-Optimum Configuration

In this subsection, we introduce an automatic power control algorithm that adapts the control vectors $\overline{c}$ and $\overline{d}$ to the variation of input statistics or a given target performance.

**Adaptive Soft Error Cancellation**

To develop a procedure to control the vectors $\overline{c}$ and $\overline{d}$, we need to update the estimator weights $w_{c,i}$, $v_{c,i}$ and $A$ automatically, for each $\overline{c}$ and $\overline{d}$. We can employ the *least mean square* (LMS) algorithm, which adapts to minimize the MSE (6.10) over the weights:

\[
w_{c,i}^{(n+1)} = c_i \left( w_{c,i}^{(n)} + \mu \epsilon_n x_{q,n-i+1} \right), \quad \text{for } i = 1, \cdots, N_1 \tag{6.48}
\]

\[
v_{c,i}^{(n+1)} = d_i \left( v_{c,i}^{(n)} + \mu \epsilon_n \hat{y}_{q,n-i} \right), \quad \text{for } i = 1, \cdots, N_2 \tag{6.49}
\]

\[
A^{(n+1)} = A^{(n)} + \mu \epsilon_n, \tag{6.50}
\]

where $w_{c,i}^{(n)}$, $v_{c,i}^{(n)}$, and $A^{(n)}$ are the values of $w_{c,i}$, $v_{c,i}$ and $A$ at time $n$, respectively, and $\mu$ is the step size. We denote $\epsilon_n$ as the result of subtracting the MEF output from $y_n$. Since the value of $y_n$ is not available, we use the current restored output instead of $y_n$ as a training symbol, and hence $\epsilon_n$ is given by

\[
\epsilon_n = \hat{y}_n + \sum_{i=1}^{N_1} w_{c,i}^{(n)} x_{q,n-i+1} + \sum_{i=1}^{N_2} v_{c,i}^{(n)} \hat{y}_{q,n-i} + A^{(n)}. \tag{6.51}
\]

Whenever there is a change in $\overline{c}$ or $\overline{d}$, the adaptive algorithm begins operation until it converges to the correspondingly optimal weight vector. Since $\overline{c}$ and $\overline{d}$ can power down the update algorithm for each weight, the power consumption of weight update block (WUB) can be reduced depending on $\overline{c}$ and $\overline{d}$. We can now address the control of $\overline{c}$ and $\overline{d}$.
Automatic Power Control Algorithm

We assume that the computation of each tap \( w_{c,i} \) or \( v_{c,i} \) and its weight update algorithm consumes the same power \( E_s \). Then, the power dissipation in the SEC block is given by

\[
P_{\text{SEC}} = \left( \sum_{i=1}^{N_1} c_i + \sum_{i=1}^{N_2} d_i \right) E_s + \mathcal{H},
\]

where \( \mathcal{H} \) includes the power dissipation in the MLD block. This means that as we power down more taps, the power dissipation in SEC will proportionally decrease. Hence, we can rewrite the energy optimization problem (6.42) by

\[
\text{Minimize : } \sum_{i=1}^{N_1} c_i + \sum_{i=1}^{N_2} d_i \\
\text{Subject to : } \sigma_r^2 \leq J,
\]

where the RMSE constraint can be replaced by a new constraint, \( \sigma_r^2 \leq J \), since RMSE is an increasing function in \( \sigma_r^2 \). In [100], the authors presented a solution to a similar problem for an adaptive equalizer, using a Lagrange multiplier method under the assumption that the input signals to the adaptive equalizer are white. The solution suggested that the best strategy involves powering down the taps of the equalizer with less contribution to the performance metric if each tap consumes the same energy. Unfortunately, it is considerably more difficult to find the associated control vectors for the case with correlated input. Hence, we adopt the strategy of switching off the taps with the smallest coefficients.

The performance estimate (P-estimate) \( P_n \) is monitored in real time and compared with two preset thresholds \( \tau_1 \) and \( \tau_2 \) where \( \tau_1 > \tau_2 \). The P-estimate is computed by averaging the square of the LMS update error, \( \epsilon_n^2 \), i.e.,

\[
P_n = (1 - \rho)P_{n-1} + \rho \epsilon_n^2,
\]

where \( 0 < \rho < 1 \) is a constant for experimental averaging. The automatic power control algorithm (PCA) changes \( \overline{c} \) and \( \overline{d} \) only when the P-estimate is larger than \( \tau_1 \) or smaller than \( \tau_2 \). Starting from the initial setup, \( \overline{c}_0 \) and \( \overline{d}_0 \), which are preset to provide a small P-estimate, we set \( c_i = 0 \) or \( d_i = 0 \) if the
Table 6.1: Power optimization algorithm.

| STEP 1 | Start with $c_0$ and $d_0$ preset to yield P-estimate smaller than $\tau_2$. |
| STEP 2 | Wait until the estimator coefficients converge. |
| STEP 3 | Monitor P-estimate: if $P_n - \tau_2 \leq 0$ go to STEP 4, and if $P_n - \tau_1 \geq 0$ go to STEP 5. |
| STEP 4 | For $|w_{c,i}| < |w_{c,j}| \forall i \neq j$ and $|v_{c,k}| < |v_{c,l}| \forall k \neq l$, set $c_i = 0$ if $|w_{c,i}| < |v_{c,k}|$, else set $d_k = 0$. Go to STEP 2. |
| STEP 5 | Set $c_i = 1$ or $d_i = 1$ for the mask coefficient subject to the last change. Go to STEP 2. |

P-estimate is smaller than $\tau_2$, where $i$ corresponds to smallest coefficients. In contrast, we set $c_i = 1$ or $d_i = 1$, if $P_n$ exceeds $\tau_1$ in the reverse order. Once the control vector, $\tilde{c}$ or $\tilde{d}$, has been changed, the PCA waits until the adaptive algorithm converges. When the P-estimate lies within $[\tau_2, \tau_1]$, we power down the PCA and keep monitoring the P-estimate to detect any changes. This procedure is summarized in Table 6.1.

6.4 Discussion

In this section, we discuss a hardware design of an MP-SEC system, and present a simulation framework and some results.

6.4.1 Hardware Design

In Figure 6.8, we depict an implementation of the MP-SEC system, which protects a 26-tap FIR frequency selective filter. The system is designed via the static design methodology described in Section 6.3.2. It should be noted that the critical path delay of the MEF is shorter than the main filter due to its reduced complexity, and therefore soft errors cannot occur in the MEF for values of up to $k_{vos} = 0.6$.

Since the detection rule (6.25) requires the use of a multiplier, it is desirable to further simplify the detector structure complexity. Under $H_1$, it follows that $\tilde{e}_n^2 \gg (\tilde{e} - \tilde{e}_n)^2$, and under $H_0$, $(\tilde{e} - \tilde{e}_n)^2$ is close to zero. As such, the
The detection rule can be simplified to

$$|\tilde{e}_n| < \sqrt{2\gamma \sigma_r^2}. \quad (6.55)$$

This rule is called a double-sided test (DST) which compares \(\tilde{e}_n\) to both positive and negative thresholds. This rule can be expressed as

$$I_b(\tilde{e}_n)\tilde{e}_n - \sqrt{2\gamma \sigma_r^2} \begin{cases} H_1 \\ H_2 \end{cases} \geq 0, \quad (6.56)$$

where \(I_b(\cdot)\) outputs 1 for a positive input and \(-1\), otherwise. As a result, the hardware design of the detector in (6.56) requires only one inverter for computing \(I_b(\cdot)\) and one subtractor.

Next, consider the hardware design of the MLD block. We can write (6.18) as

$$\hat{e}_n = \text{round} \left( \frac{2^{B-M-1} \cdot \tilde{e}_n}{2^{B-M-1}} \right). \quad (6.57)$$

Figure 6.9 (a) illustrates the computation of \(\hat{e}_n\) based on \(\tilde{e}_n\), when \(B = 9\) and \(M = 5\). Figure 6.9 (b) describes its implementation. The MLD block can be implemented with only one MUX and one full adder.

Next, we depict an SEC system that employs dynamic PCA in Figure 6.10.
$2^{B-M-1} \hat{e}_n = 1010.01110_2$

Figure 6.9: Implementation of MLD block.
Table 6.2: Hardware units for MP-SEC system.

<table>
<thead>
<tr>
<th>Block</th>
<th>Sub-blocks</th>
<th>Necessary units</th>
<th>Number of full adders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main filter</td>
<td></td>
<td>26 multipliers (16 bit), 27 adders (21 bit)</td>
<td>7223</td>
</tr>
<tr>
<td>SEC Block</td>
<td>MEF</td>
<td>9 multipliers (7 bit), 8 adders (12 bit)</td>
<td>537</td>
</tr>
<tr>
<td></td>
<td>DST</td>
<td>1 adder (5 bit), 1 inverter</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>MLD ect</td>
<td>1 MUX, 1 adder (4 bit)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 adders (21 bit)</td>
<td>42</td>
</tr>
</tbody>
</table>

To implement the power control block in hardware, we devise the stack-based power control which memorizes the tap-positions of the inactive MEF coefficients in the stack on a first input last output (FILO) basis. When the P-estimate drops below $\tau_2$, the tap-position of the smallest MEF weight enters the stack, deactivating the coefficient. When the P-estimate rises above $\tau_1$, the tap at the top of the stack is released. Though the hardware implementation including the WUB and PCA blocks appears complicated, the algorithm powering down the SEC sub-blocks dramatically reduces power consumption.

Table 6.2 summarizes the number of basic arithmetic units required for the SEC block depicted in Figure 6.8, compared with that of the main filter when the PCA is not employed. Note that the hardware complexity of the SEC block is simple compared to that of the main filter.
6.4.2 Energy Saving Measure

The energy savings $E_{\text{sav}}(\%)$ of the ML-EC system is defined by

$$E_{\text{sav}} = \frac{P_{\text{ORG}} - P_{\text{VOS}}}{P_{\text{ORG}}} \times 100,$$

(6.58)

where $P_{\text{ORG}}$ and $P_{\text{VOS}}$ are the power dissipation before and after applying MP-SEC technique, respectively. Note that the power required by SEC block, $P_{\text{SEC}}$, is included in calculating $P_{\text{VOS}}$.

6.4.3 Simulation Setup

The simulation setup used in our simulations is illustrated in Fig 6.11. The context that we chose for experiments is a low-pass FIR filter (LPF) that removes the out-of-band noise in front of a speech recognizer. A sequence of 10,000 speech samples of bandwidth 8 kHz are filtered to remove the out of band corruption from additive white Gaussian noise (AWGN). A 25-tap linear-phase FIR LPF with cut-off frequency $\pi/4$ is used as the main filter to be protected.

We assume a 0.25$\mu$m, 2.5 V CMOS process technology and that 16 $\times$ 16 bit Baugh-Wooley multipliers [105] are used in the main filter. We also assume that the supply voltage, set to 2.5 V, is scaled down by $k_{\text{vos}} = 0.9, 0.8, 0.7$ and 0.6. First, we compute the logic gate delay for each $k_{\text{vos}}$ via a circuit-level simulator, HSPICE, and obtain the worst path delays reaching the intermediate bits of each processing unit via a logic-level simulator. The intermediate bits with larger path delay than the sampling period exhibit a
Table 6.3: Soft error rate, spacing and SNR degradation vs. VOS factor.

<table>
<thead>
<tr>
<th>$k_{vos}$</th>
<th>1.0</th>
<th>0.9</th>
<th>0.8</th>
<th>0.7</th>
<th>0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft error rate, $P(H_1)$ (%)</td>
<td>0%</td>
<td>2.69%</td>
<td>8.13%</td>
<td>29.88%</td>
<td>54.55%</td>
</tr>
<tr>
<td>Number of SBs, $M$</td>
<td>None</td>
<td>13</td>
<td>12</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Output SNR (dB)</td>
<td>22.91</td>
<td>-0.16</td>
<td>-4.34</td>
<td>-8.24</td>
<td>-10.92</td>
</tr>
</tbody>
</table>

Table 6.4: Design specification and energy savings of energy-minimum MP-SEC system.

<table>
<thead>
<tr>
<th>$k_{vos}$</th>
<th>FF-MEF Length ($N'_1$)</th>
<th>FB-MEF Length ($N'_2$)</th>
<th>$p$</th>
<th>SNR Achievement</th>
<th>$E_{\text{snr}}$ (%)</th>
<th>$P_{\text{SEC}}/P_{\text{VOS}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>2 tap</td>
<td>2 tap</td>
<td>4 bit</td>
<td>22.91 dB</td>
<td>17.76%</td>
<td>2.10%</td>
</tr>
<tr>
<td>0.8</td>
<td>4 tap</td>
<td>2 tap</td>
<td>6 bit</td>
<td>22.91 dB</td>
<td>33.40%</td>
<td>4.69%</td>
</tr>
<tr>
<td>0.7</td>
<td>13 tap</td>
<td>1 tap</td>
<td>7 bit</td>
<td>22.46 dB</td>
<td>42.23%</td>
<td>14.03%</td>
</tr>
<tr>
<td>0.6</td>
<td>15 tap</td>
<td>1 tap</td>
<td>9 bit</td>
<td>22.08 dB</td>
<td>57.57%</td>
<td>23.90%</td>
</tr>
</tbody>
</table>

timing violation, thereby causing a soft error. The power dissipation of the system is obtained via a gate-level power simulation tool, MED [106].

6.4.4 Simulation Results

Table 6.3 tabulates the soft error rate, number of SB, $M$, and output SNR versus $k_{vos}$, when the speech samples are used as the input. As $k_{vos}$ decreases, the error rate increases, and the number of SBs decreases or equivalently $\lambda$ decreases. The original output SNR before applying VOS was 22.91 dB, but the system experiences a catastrophic SNR drop with $k_{vos}$.

Figure 6.12 (a) shows the original 400 samples of the desired speech signal, $y_n$.

The output signal which is corrupted by soft errors when $k_{vos} = 0.7$ is shown in Figure 6.12 (b). We employ the MP-SEC unit to restore the degraded signal. The predesigned 13-tap FF-MEF and 1-tap FB-MEF are employed to meet the target SNR, or 22 dB. The signal $\hat{e}_n$ and restored $\hat{y}_n$ are shown in Figure 6.12 (c) and Figure 6.13 (a). Note that the MEF modifies the noisy signal to $\hat{e}_n$ to readily estimate $e_n$. The resulting estimation error, $\hat{y}_n - y_n$, after error correction is shown in Figure 6.13 (b).

Table 6.4 summarizes the design specifications of the MP-SEC system for each $k_{vos}$ when the power-optimum design strategy described in Section 6.3.2, is employed. The table also includes the resulting SNR and achieved power savings. When $k_{vos}$ is 0.9 and 0.8, the degraded SNR is completely restored.
Figure 6.12: Shown are 400 samples of (a) the clean output $y_n$, (b) the noisy output $z_n$, (c) $\tilde{e}_n$ when $k_{vos}$ is set to 0.7.
Figure 6.13: Shown are 400 samples of (a) the corrected output $\hat{y}_n$, and (b) the estimation error, $\hat{y}_n - y_n$, when $k_{vos}$ is set to 0.7.

with at most 6 coefficients, and 6 bit precision, since wide soft error spacing allows for relatively loose MSE requirements for error correction. As $k_{vos}$ is scaled to 0.7 and then to 0.6, we can achieve power savings of 57%, with at most 0.83 dB SNR loss. This is why the quadratic effect of $k_{vos}$ on power dissipation becomes dominant even when the complexity overhead in the SEC block is increased.

Table 6.5 tabulates the design specifications and the resulting energy savings depending on various input signal and filter bandwidths. To generate the input with a particular bandwidth, random white noise is shaped by a linear filter with the given cut-off frequency. The SEC block is designed to allow at most 1 dB SNR loss. For brevity, we present the result only for $k_{vos} = 0.7$, since similar behavior trends were observed when $k_{vos}$ was set to 0.6, 0.8, and 0.9. The higher the input bandwidth, the less power savings. However, the decrease is slight, and we gain relatively consistent power savings for all input bandwidths. Furthermore, the filter bandwidths hardly appear to influence the power saving, which may be a desirable feature compared with the prediction-based error-correction technique [5].

Figure 6.14 shows the nature of the adaptation performed by the automatic
Figure 6.14: Shown are (a) the signal, $x_n$, (b) the P-estimate, and (c) $N'_1 + N'_2$. 

132
Table 6.5: Design specification and energy savings depending on input and filter bandwidth. \((k_{vos} = 0.7).\)

<table>
<thead>
<tr>
<th>Filter bandwidth</th>
<th>Input bandwidth</th>
<th>FF-MEF length</th>
<th>FB-MEF length</th>
<th>(p)</th>
<th>SNR loss (dB)</th>
<th>Energy savings (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2π</td>
<td>0.2π</td>
<td>14</td>
<td>2</td>
<td>5 bit</td>
<td>0.99</td>
<td>47.13</td>
</tr>
<tr>
<td></td>
<td>0.4π</td>
<td>18</td>
<td>1</td>
<td>5 bit</td>
<td>0.79</td>
<td>46.18</td>
</tr>
<tr>
<td></td>
<td>0.6π</td>
<td>20</td>
<td>1</td>
<td>5 bit</td>
<td>1.00</td>
<td>45.94</td>
</tr>
<tr>
<td></td>
<td>0.8π</td>
<td>21</td>
<td>1</td>
<td>5 bit</td>
<td>0.86</td>
<td>45.71</td>
</tr>
<tr>
<td>0.4π</td>
<td>0.2π</td>
<td>6</td>
<td>3</td>
<td>6 bit</td>
<td>0.76</td>
<td>48.00</td>
</tr>
<tr>
<td></td>
<td>0.4π</td>
<td>12</td>
<td>1</td>
<td>5 bit</td>
<td>0.97</td>
<td>47.84</td>
</tr>
<tr>
<td></td>
<td>0.6π</td>
<td>10</td>
<td>3</td>
<td>6 bit</td>
<td>0.82</td>
<td>46.70</td>
</tr>
<tr>
<td></td>
<td>0.8π</td>
<td>21</td>
<td>1</td>
<td>5 bit</td>
<td>0.94</td>
<td>45.71</td>
</tr>
<tr>
<td>0.6π</td>
<td>0.2π</td>
<td>6</td>
<td>3</td>
<td>6 bit</td>
<td>0.86</td>
<td>48.66</td>
</tr>
<tr>
<td></td>
<td>0.4π</td>
<td>7</td>
<td>1</td>
<td>5 bit</td>
<td>0.97</td>
<td>49.04</td>
</tr>
<tr>
<td></td>
<td>0.6π</td>
<td>7</td>
<td>3</td>
<td>6 bit</td>
<td>0.98</td>
<td>47.68</td>
</tr>
<tr>
<td></td>
<td>0.8π</td>
<td>12</td>
<td>3</td>
<td>5 bit</td>
<td>0.95</td>
<td>45.37</td>
</tr>
<tr>
<td>0.8π</td>
<td>0.2π</td>
<td>5</td>
<td>1</td>
<td>6 bit</td>
<td>0.94</td>
<td>48.99</td>
</tr>
<tr>
<td></td>
<td>0.4π</td>
<td>5</td>
<td>3</td>
<td>6 bit</td>
<td>0.68</td>
<td>48.33</td>
</tr>
<tr>
<td></td>
<td>0.6π</td>
<td>12</td>
<td>2</td>
<td>5 bit</td>
<td>0.90</td>
<td>47.61</td>
</tr>
<tr>
<td></td>
<td>0.8π</td>
<td>13</td>
<td>1</td>
<td>5 bit</td>
<td>0.93</td>
<td>47.61</td>
</tr>
</tbody>
</table>

PCA described in Section 6.3.3 to changes in the input signal characteristics. Figure 6.14 (a) contains a plot of the input signal \(x_n\). The input \(x_n\) exhibits a dramatic change in statistics at time 10,000 and 20,000. Specifically, over the intervals [0, 10000], [10000, 20000], and [20000, 30000], the signal has bandwidth of 0.2\(\pi\), 0.5\(\pi\), and 0.9\(\pi\) and variance of 0.04, 0.0025, and 0.06, respectively. Figure 6.14 (b) shows the P-estimate in Eq. (6.54) when \(\rho\) is set to 0.999. The evolution of the number of powered-up coefficients is plotted in Figure 6.14 (c). The largest power savings is achieved during the interval between the samples of 10,000 and 20,000, where only 7 coefficients are used for soft error cancellation. Note that the coefficient adaptation and power control operate only when the P-estimate remains outside of the range \([\tau_2, \tau_1]\). Hence, when the P-estimate lies within \([\tau_2, \tau_1]\), the power overhead of automatic PCA comes from the task of computing the P-estimate and comparing it with the thresholds.

Figure 6.15 compares the performance and power trade-off of MP-SEC with those of the prediction-based error-correction [5] and reduced precision replica [96] techniques, when the speech samples are used as the system input. The MP-SEC technique yields better SNR performance over the range of 0% to 50% power savings, and is 9 dB better than the prediction-based error-correction method and 1 dB better from the reduced precision replica method at 40% power savings. When employing the PCA, the MP-SEC achieves 8% power savings with no SNR loss and 15% power savings within 1 dB SNR loss.
6.5 Conclusions

In this chapter, we have addressed two problems: (1) estimation and detection of soft errors induced by VOS in low-power digital filtering, and (2) an approach to energy minimum design and adaptive power control for re-configuration. Our derivation of the soft error estimator and detection algorithm is based on the observation that soft errors are created by higher order bits in the representation of the signals being processed. As such, through tracking the signal correlation over time, and using a reduced-precision replica of the filtering operation to be protected, such high-order bit errors can be readily detected and ultimately corrected. Through a low-power implementation of the error detection and correction unit, the MP-SEC approach shows promise for achieving significant power savings for digital filtering as well as a variety of DSP applications.
In this dissertation, efficient receiver algorithms and architectures for wireless RF and underwater acoustic communication were investigated. The different approaches to receiver design were presented for several different scenarios such as uncoded and coded MIMO systems, and systems with a frequency selective channel. Various design principles were introduced, including dimension reduction, look-ahead path metrics and power-optimum optimization. Though these principles were separately addressed in the dissertation, they can be combined together to build a host of efficient receiver architectures.

In Chapter 3, a low-complexity near-ML detector was proposed. In order to reduce the size of the search space, the received symbol vector was partitioned into two parts, strong symbols and weak symbols, according the V-BLAST detection ordering. Then, the tree search was performed over the combinations of strong symbols. Towards this end, the MMSE dimension operator was developed, which suppresses the impact of weak symbols. However, this dimension reduction approach causes unavoidable loss in diversity gain since there is no search over the combinations of possible weak symbols. To compensate for the performance loss, the best $K$ candidates were searched via a list tree search and the final candidate was chosen among them after extending each candidate to the full dimension. An efficient LTS algorithm called the closest $K$ list stack algorithm was developed that can control the size of candidate list adaptively. Asymptotic error analysis of this RD-MLS technique showed that performance gains can be achieved via the list tree search. Furthermore, simulation results confirmed that the RD-MLS yields a better performance-complexity trade-off than existing near-ML detectors.

In Chapter 4, an efficient soft-input soft-output tree detector was proposed. The soft-input soft-output $M$-algorithm was investigated since it is well suited for the APP detection for high dimension systems. In spite of low complexity, the $M$-algorithm is sub-optimal due to its greedy nature.
The performance of the $M$-algorithm was improved by developing a linear estimate-based look-ahead (LE-LA) path metric, which accounts for the contribution of unvisited paths. By looking ahead to the unvisited paths, the new path metric reflects the reliability of a path better than the conventional path metric. The analysis of the probability of correct path loss was derived to support the advantage of the look-ahead path metric. Simulation results showed that the proposed ISS-MA promises 0.5-1.0 dB performance gain after convergence at the expense of a small complexity overhead.

In Chapter 5, the receiver design issues for underwater acoustic communication were discussed. Since an underwater acoustic channel is doubly selective, the receiver should be able to handle long channel impulse responses as well as track rapidly varying channel gains. The application of the turbo equalization technique to underwater channels was studied in this chapter. Two linear turbo equalization techniques, channel-estimate-based MMSE turbo equalizer and direct-adaptive turbo equalizer, were compared. It was shown through EXIT chart analysis that after sufficient number of iterations, the performance gap between them is small. Due to the use of the LMS algorithm, the DA-TEQ could be implemented at low computational complexity. The structure of an underwater receiver based on the LMS DA-TEQ technique was presented. According to the analysis based on experiments conducted off the coast of Martha’s Vineyard, the LMS DA-TEQ receiver achieved an order of magnitude performance gain over the conventional decision feedback equalizer widely used in practice for such system.

In Chapter 6, the design of a reliable and power-optimum ANT system was presented. The proposed ANT technique detects, estimates, and cancels the soft errors from the system output based on an ML criterion. The discrete property of soft errors was exploited in deriving the ML-based soft error canceler. Since the ANT system requires some amount of power overhead in the overall system, its power dissipation was minimized via a constrained optimization problem. Specifically, power consumption of the FIR filter structure used for ANT was minimized under SNR performance constraint. Logic-level simulations were performed to evaluate the proposed MP-SEC technique. It was shown that by exploiting a spacing property of soft errors in certain architectures, MP-SEC can achieve up to 30% power savings with no SNR loss and up to 55% power savings with less than 1 dB SNR loss, according to logic-level simulations performed for an example 25-tap frequency-selective
filter.
APPENDIX A

SUMMARY OF THE RD-MLS ALGORITHM

The following definitions are used.

- **STACK**: memory storing the generated nodes.
- **XLIST**: memory storing the candidate points found.
- **\( \bar{X}^{n_1}_k \)**: current best node (called top node) that has a minimum cost in the stack.
- **bcn (\( x^{n_1}_k \))**: best child node of \( x^{n_1}_k \) not generated.

In addition, we assume that the symbol is ordered according to the V-BLAST detection ordering and the dimension reduction operator \( Z \) and the matrix \( G (= ZH_1) \) are computed prior to the routine that follows.

- Input: \( y, Z, G \) and, \( (n_1, m) \),
- Output: \( \bar{x}_{ml} \)

**STEP 1**: (Preprocessing) Compute \( z = Zy \)

**STEP 2**: (Initialization for \( K \)-LSA)

STACK ← root node,

**STEP 3**: (Main routine of \( K \)-LSA)

Let \( \bar{X}^{n_1}_k \) be the current top node.

(Node extension)

if \( k = 2 \)

\[ \text{XLIST} \leftarrow (\text{bcn (} \bar{X}^{n_1}_k, \bar{X}^{n_1}_k) \) \]

else

\[ \text{STACK} \leftarrow (\text{bcn (} \bar{X}^{n_1}_k, \bar{X}^{n_1}_k) \)

For a new extension, update \( a_{k-1} (\{ \text{bcn (} \bar{X}^{n_1}_k, \bar{X}^{n_1}_k) \}) \) and \( \text{bcn (} \{ \text{bcn (} \bar{X}^{n_1}_k, \bar{X}^{n_1}_k) \}) \).

end

(Parent node update)

if all child nodes of \( \bar{X}^{n_1}_k \) have been generated,
remove $\check{X}^{n_1}_k$ from STACK.

else

update $\text{bcn} (\check{X}^{n_1}_k)$, and $a_k (\check{X}^{n_1}_k)$. (extend to the next sibling node [30])

end

STEP 4 : (Stack sorting)

Compare the path metrics of all STACK elements and nominate one with minimum path metric as a top node $\check{X}^{n_1}_k$.

STEP 5 : (Check the stopping criterion)

If $a_k (\check{X}^{n_1}_k)$ satisfies the stopping criterion, go to STEP 6,

else, go to STEP 3.

end

STEP 6 : (Postprocessing)

Using (3.35) and (3.37), extend all elements of XLIST to full dimension.

Output the best candidate by comparing the $L_2$ norm distance of them.
APPENDIX B

ASYMPTOTIC NORMALITY OF THE LMMSE ESTIMATION ERROR PLUS NOISE

We assume that the partitioned matrices $H_1$ and $H_2$ are given for now. Via singular value decomposition (SVD), we can decompose $H_2$ to $PVH_2$, where

$$\Sigma = \text{diag}\{\sigma_1, \cdots, \sigma_{nr}\} \quad \text{and} \quad \sigma_1 \geq \cdots \geq \sigma_{nr}. $$

Then, $ZH_2$ can be expressed as

$$ZH_2 = \sigma_w^2 \left( H_2 H_2^H + \sigma_w^2 I \right)^{-1} H_2 \quad \text{(B.1)}$$

$$= P \left( \frac{1}{\sigma_w^2} \Sigma^2 + I \right)^{-1} P^H (PVH_2) \quad \text{(B.2)}$$

$$= P \begin{bmatrix} \frac{\sigma_1 \sigma_w^2}{\sigma_1^2 + \sigma_w^2} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{\sigma_{nr} \sigma_w^2}{\sigma_{nr}^2 + \sigma_w^2} \end{bmatrix} V^H. \quad \text{(B.3)}$$

If we let $K(\leq n_2)$ be the rank of $H_2$, it follows that $\sigma_{K+1} = \cdots = \sigma_{nr} = 0$. Hence, we have $\frac{\sigma_i \sigma_w^2}{\sigma_i^2 + \sigma_w^2} = 0$ for $K+1 \leq i \leq nr$. On the other hand, since $\sigma_i > 0$ for $1 \leq i \leq K$, we have $\frac{\sigma_i \sigma_w^2}{\sigma_i^2 + \sigma_w^2} \to 0$ as $\sigma_w^2 \to 0$. Therefore, $ZH_2$ vanishes as $\sigma_w^2 \to 0$. In the similar manner, we can express $Z$ as

$$Z = P \begin{bmatrix} \frac{\sigma_w^2}{\sigma_1^2 + \sigma_w^2} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{\sigma_w^2}{\sigma_{nr}^2 + \sigma_w^2} \end{bmatrix} P^H. \quad \text{(B.4)}$$

Note that for $K+1 \leq i \leq nr$, $\frac{\sigma_w^2}{\sigma_i^2 + \sigma_w^2} = \frac{\sigma_w^2}{\sigma_i^2} = 1$. Since $K+1 \leq nr \quad (K \leq n_2 \quad \text{and} \quad n_2 < n_r)$, the projection operator $Z$ never vanishes.
APPENDIX C
PROOF OF (3.53)

Let \(\hat{x}_1^{\text{max}}\) and \(\hat{x}_1^{\text{min}}\) be the elements of \(L_1\) that correspond to the maximizer and minimizer of the cost function \(J(x) = \|z - Gx\|^2\), respectively. Given a candidate list \(L_1\), the probability \(Pr_{A_1}(x_{A_1} \notin L_1 | H, L_1)\) is expressed as

\[
Pr_{A_1}(x_{A_1} \notin L_1 | H, L_1) = Pr_{A_1}(\|z - Gx_{A_1}\|^2 > \|z - G\hat{x}_1^{\text{max}}\|^2 | H, L_1) \quad (C.1)
\]

\[
= Pr(\|\hat{w}\|^2 > \|G(x_{A_1} - \hat{x}_1^{\text{max}}) + \hat{w}\|^2 | H, L_1) \quad (C.2)
\]

\[
= Pr(\Re\{v\} < -\frac{1}{2} \|G(x_{A_1} - \hat{x}_1^{\text{max}})\|^2 | H, L_1), \quad (C.3)
\]

where \(v = (G(x_{A_1} - \hat{x}_1^{\text{max}}))^H \hat{w}\). An affine transform of a proper Gaussian variable remains proper Gaussian [107] so that \(v\) is also proper Gaussian. Recalling \(\Phi = E[(\hat{w})(\hat{w})^H] = \sigma_w^4 (H_2H_2^H + \sigma_w^2 I)^{-1}\), the variance of its real part is given by

\[
\frac{1}{2} E[|v|^2 | H, L_1]
\]

\[
= \frac{1}{2} (G(x_{A_1} - \hat{x}_1^{\text{max}}))^H \Phi (G(x_{A_1} - \hat{x}_1^{\text{max}}))
\]

\[
= \frac{1}{2} \|\Psi G(x_{A_1} - \hat{x}_1^{\text{max}})\|^2
\]

where \(\Psi\) is a square root of \(\Phi\), i.e., \(\Phi = \Psi^H \Psi\). Recall that \(Pr(z < -\beta) = Pr(z > \beta) = Q(\frac{\beta}{\sigma})\) for real Gaussian variable \(z\) with zero mean and variance \(\sigma^2\). Then \(Pr_{A_1}(x_{A_1} \notin L_1 | H, L_1)\) is expressed as

\[
Pr_{A_1}(x_{A_1} \notin L_1 | H, L_1) = Q\left(\frac{\|G(x_{A_1} - \hat{x}_1^{\text{max}})\|^2}{\sqrt{2} \|\Psi G(x_{A_1} - \hat{x}_1^{\text{max}})\|^2}\right). \quad (C.4)
\]
From the definition of a matrix norm, \( \|A\|_2 = \max_x \|Ax\| / \|x\| \), i.e., \( \|Ax\| / \|x\| \leq \|A\|_2 \), we have

\[
Pr_{A1} \left( x_{A1} \notin L_1 \middle| H, L_1 \right) \leq Q \left( \sqrt{\frac{\|G (x_{A1} - \hat{x}_{1}^{\text{max}})\|^2}{2 \|\Psi\|^2_2}} \right). \tag{C.5}
\]

The candidate search is stopped when a lattice point whose distance is larger than \( md_1 \left( y | \hat{x}_1^{(1)} \right) (= m \| z - G\hat{x}_{1}^{\text{min}}\|^2) \) is found. In addition, to simplify the analysis, we include the final lattice point in the list, which is the first lattice point violating the stopping criterion. With this assumption, the list can be expressed as \( L_1 = \{ \hat{x}_1^1, \hat{x}_1^2, \ldots, \hat{x}_1^{u-1}, \hat{x}_1^u \} \) where \( \hat{x}_1^{\text{min}} = \hat{x}_1^1 \) and \( \hat{x}_1^{\text{max}} = \hat{x}_1^u \). Due to the stopping criterion, we have

\[
\|z - G\hat{x}_1^{\text{max}}\|^2 \geq m \|z - G\hat{x}_1^{\text{min}}\|^2. \tag{C.6}
\]

By taking an expectation of (C.6) with respect to \( \hat{w} \) and plugging \( z = Gx_{A1} + \hat{w} \), we have

\[
E \left[ \|G (x_{A1} - \hat{x}_1^{\text{max}}) + \hat{w}\|^2 \middle| H, L_1 \right] \geq m E \left[ \|G (x_{A1} - \hat{x}_1^{\text{min}}) + \hat{w}\|^2 \middle| H, L_1 \right]. \tag{C.7}
\]

After some manipulation, (C.7) becomes

\[
\|G (x_{A1} - \hat{x}_1^{\text{max}})\|^2 \geq m \|G (x_{A1} - \hat{x}_1^{\text{min}})\|^2 + (m - 1) \text{tr} (\Phi) \geq m \|G (x_{A1} - \hat{x}_1^{\text{min}})\|^2 \tag{C.8}
\]

for given \( H \) and \( L_1 \), where (C.9) follows from that \( m > 1 \) and \( \Phi (= \Psi H \Psi) \) is positive semi-definite. Using \( \|\Phi\|_2 = \|\Psi\|^2_2 \) and from (C.5) and (C.9),

\[
Pr_{A1} \left( x_{A1} \notin L_1 \middle| H, L_1 \right) \leq Q \left( \sqrt{\frac{m \|G (x_{A1} - \hat{x}_1^{\text{min}})\|^2}{2 \|\Phi\|^2_2}} \right). \tag{C.10}
\]

Let \( (H_2 H_2^H + \sigma_w^2 I)^{-1} \) be decomposed into \( U \Lambda U^H \), where \( U \) is a unitary matrix and \( \Lambda \) is the diagonal matrix whose diagonal entries are the eigenvalues of \( (H_2 H_2^H + \sigma_w^2 I)^{-1} \), \( \lambda_1 \geq \cdots \geq \lambda_n \). From (3.22), \( Z \) is expressed
as \( U(\sigma_w^2 \mathbf{A}) U^H \). In addition, since \( \Phi = \sigma_w^4 (\mathbf{H}_2 H_2^H + \sigma_w^2 \mathbf{I})^{-1} \), \( \|\Phi\|_2 \) equals \( \sigma_w^4 \lambda_1 \) and hence

\[
P_{R_{A1}} \left( x_{A1} \notin \mathcal{L}_1 \mid \mathbf{H}, \mathcal{L}_1 \right) \leq Q \left( \sqrt{m \| \Lambda \mathbf{H}_1 (x_{1A} - \hat{x}_1^{\min}) \|_2^2 / 2\lambda_1} \right), \quad (C.11)
\]

where \( \tilde{\mathbf{H}}_1 = U^H \mathbf{H}_1 \).

To remove conditioning, \( P_{R_{A1}} \left( x_{A1} \notin \mathcal{L}_1 \mid \mathbf{H}, \mathcal{L}_1 \right) \) needs to be averaged over \( \mathbf{H} \) and \( \mathcal{L}_1 \), i.e.,

\[
P_{R_{A1}} (x_{A1} \notin \mathcal{L}_1) = \sum_{\{\mathcal{L}_1\}} E_{\mathbf{H}} \left[ P_{R_{A1}} \left( x_{A1} \notin \mathcal{L}_1 \mid \mathbf{H}, \mathcal{L}_1 \right) \right] P_{R_{A1}} (\mathcal{L}_1), \quad (C.12)
\]

where \( P_{R_{A1}} (\mathcal{L}_1) \) is the probability that \( \mathcal{L}_1 \) is observed given \( x_{A1} \). We can divide the set of candidate lists \( \{\mathcal{L}_1\} \) into \( \{e_{\mathcal{L}_1}\} \) and \( \{\bar{\mathcal{L}}_1\} \) which contain \( x_{A1} \) and do not, respectively. It is clear that for a candidate list \( \mathcal{L}_1 \in \{\bar{\mathcal{L}}_1\} \), we have \( P_{R_{A1}} \left( x_{A1} \notin \mathcal{L}_1 \mid \mathbf{H}, \mathcal{L}_1 \right) = 0 \). Hence, the summation in (C.12) reduces to that over \( \{\bar{\mathcal{L}}_1\} \). Further, employing the property \( Q(x) \leq \exp \left( -\frac{x^2}{2} \right) \) and \( E[f(x,y)] = E_y [E_x [f(x,y)|y]] \), we have

\[
E_{\mathbf{H}} \left[ P_{R_{A1}} \left( x_{A1} \notin \mathcal{L}_1 \mid \mathbf{H}, \mathcal{L}_1 \right) \right] \\
\leq E_{\tilde{\mathbf{H}}_1} \left[ E_{\mathbf{H}_2} \left[ \exp \left( -m \sum_{i=1}^{n_r} \lambda_i^2 |p_i|^2 / 4\lambda_1 \right) \right] \right], \quad (C.13)
\]

where \( p_i = \tilde{h}_{i_2}^H (x_{1A} - \hat{x}_1^{\min}) \) and \( \tilde{h}_{i_2}^H \) is the \( i \)th row vector of \( \tilde{\mathbf{H}}_1 \). With the assumption that the elements of \( \tilde{\mathbf{H}}_1 \) (equivalently \( \mathbf{H}_1 \)) are i.i.d. complex Gaussian, \( \mathcal{CN}(0,1) \), \( p_i \) follows \( \mathcal{CN}(0,\|x_{1A} - \hat{x}_1^{\min}\|^2) \). Hence, we can show that (C.13) can be expressed as [1]

\[
E_{\mathbf{H}} \left[ P_{R_{A1}} \left( x_{A1} \notin \mathcal{L}_1 \mid \mathbf{H}, \mathcal{L}_1 \right) \right] \\
\leq E_{\mathbf{H}_2} \left[ \prod_{i=1}^{n_r} \frac{4\lambda_i}{4\lambda_1 + m\lambda_i^2 \|x_{A1} - \hat{x}_1^{\min}\|^2} \right]. \quad (C.14)
\]

Since the minimum distance between two constellation points is \( 2/\lambda \) from
(4.2), for \( \mathcal{L}_1 \in \{ \mathcal{L}_1 \} \), it holds that \( \| x_{1A} - \hat{x}_{1}^{\min} \|^2 \geq 4/\lambda^2 \). Therefore, (C.12) becomes

\[
Pr_{A1} (x_{A1} \notin \mathcal{L}_1) \leq \sum_{x_1 \neq x_{A1}} E_H \left[ \prod_{i=1}^{n_r} \frac{4\lambda_1}{4\lambda_1 + m\lambda^2_1 \| x_{A1} - x_1 \|^2} \right] \quad (C.15)
\]

\[
\leq \sum_{x_1 \neq x_{A1}} E_H \left[ \prod_{i=1}^{n_r} \frac{\lambda_1}{\lambda_1^* + \frac{m}{\lambda^2}} \right] \quad (C.16)
\]

\[
=(M^{n_1} - 1) E_H \left[ \prod_{i=1}^{n_r} \frac{\lambda_1}{\lambda_1^* + \frac{m}{\lambda^2}} \right]. \quad (C.17)
\]
The transformed vector $y$ can be expressed as $y = Rx + n$, where $n = Q_1 n_o$. Letting $k$ be the current layer being searched, then $y$, $x$, and $n$ can be partitioned into two $(k - 1) \times 1$ and $(N - k + 1) \times 1$ vectors, i.e.,

$$y = \begin{bmatrix} y_{1}^{k-1} \\ y_{k}^{N} \end{bmatrix}$$

$$= \begin{bmatrix} R_{11,k} & R_{12,k} \\ 0 & R_{22,k} \end{bmatrix} \begin{bmatrix} x_{1}^{k-1} \\ x_{k}^{N} \end{bmatrix} + \begin{bmatrix} n_{1}^{k-1} \\ n_{k}^{N} \end{bmatrix},$$

where the upper-triangular matrix $R$ is partitioned into four sub-matrices.

Given the transmitted symbol $x_{k}^{N} = \tilde{x}_{k}^{N}$, a posteriori probability of $x_{1}^{k-1}$ is given by

$$\ln Pr \left( x_{1}^{k-1} \bigg| y, x_{k}^{N} = \tilde{x}_{k}^{N} \right)$$

$$= \ln Pr \left( y \bigg| x_{1}^{k-1}, x_{k}^{N} = \tilde{x}_{k}^{N} \right) + \ln Pr \left( x_{1}^{k-1} \right)$$

$$= - \ln \left( \sqrt{2\pi \sigma_n} \right) - \frac{1}{\sigma_n^2} \left\| y - R \begin{bmatrix} x_{1}^{k-1} \\ \tilde{x}_{k}^{N} \end{bmatrix} \right\|^2 + \ln Pr \left( x_{1}^{k-1} \right)$$

$$= - \ln \left( \sqrt{2\pi \sigma_n} \right) - \frac{1}{\sigma_n^2} \left\| y_{1}^{k-1} - R_{12,k} \tilde{x}_{k}^{N} - R_{11,k} x_{1}^{k-1} \right\|^2$$

$$- \frac{1}{\sigma_n^2} \left\| y_{k}^{N} - R_{22,k} \tilde{x}_{k}^{N} \right\|^2 + \ln Pr \left( x_{1}^{k-1} \right).$$

(3)
Hence, we can show that

\[ \hat{x}_{k-1} = \arg \max \ln \Pr \left( x_{k-1} | y, x_1^N = \bar{x}_N^k \right) \]  
\[ = \arg \min_{x_{k-1}} \left\| y_{k-1} - R_{12,k} \bar{x}_N^k - R_{11,k} x_{k-1} \right\|^2 - \frac{2}{n} \ln \Pr \left( x_{k-1} \right) \]  
\[ = \arg \min_{x_{k-1}} \sum_{i=1}^{k-1} \left( y_i - \sum_{j=k}^{N} r_{i,j} \bar{x}_j - \sum_{j=i}^{k-1} r_{i,j} x_j \right)^2 - \frac{2}{n} \sum_{j=1}^{Q} \ln \Pr \left( \bar{c}_{i,j} \right) \]  
\[ = \arg \min_{x_{k-1}} \sum_{i=1}^{k-1} b \left( x_i^N \right), \]  

where the equation (D.10) follows from the definition of the branch metric. Hence, for \( x_k^N = \bar{x}_N^k \), \( \min_{x_i} \sum_{i=1}^{k-1} b \left( x_i^N \right) = \sum_{i=1}^{k-1} b \left( x_i^N \right) \bigg|_{x_i = x_i^N}. \)
APPENDIX E

PROOF OF (4.38)

We can express $Z_{k+1}$ in (4.29) as

\[ Z_{k+1} = \sigma_n^2 \left[ R_{11,k+1} A_{k+1} (R_{11,k+1})^H + \sigma_n^2 I \right]^{-1} \]

\[ = \sigma_n^2 \left( \begin{bmatrix} R_{11,k} & r_{k+1} \\ 0 & r_{k+1,k+1} \end{bmatrix} \begin{bmatrix} A_k & 0 \\ 0 & \lambda_{k+1} \end{bmatrix} \begin{bmatrix} R_{11,k} & r_{k+1} \\ 0 & r_{k+1,k+1} \end{bmatrix}^H + \begin{bmatrix} \sigma_n^2 I_k & 0 \\ 0 & \sigma_n^2 \end{bmatrix} \right)^{-1} \]

\[ = \sigma_n^2 \left( \sigma_n^2 (Z_k)^{-1} + \lambda_{k+1} r_{k+1,k+1}^T Z_k + \lambda_{k+1}^2 r_{k+1,k+1}^T + \sigma_n^2 \right)^{-1} \]

To obtain the update formula, for partitioned matrices, $A$ is given by [63]

\[
A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}
\]

we have

\[
A^{-1} = \begin{bmatrix} (A_{11} - A_{12} A_{22}^{-1} A_{21})^{-1} & - (A_{11} - A_{12} A_{22}^{-1} A_{21})^{-1} A_{12} A_{22}^{-1} \\ - (A_{22} - A_{21} A_{11}^{-1} A_{12})^{-1} A_{21} A_{11}^{-1} & (A_{22} - A_{21} A_{11}^{-1} A_{12})^{-1} \end{bmatrix}.
\]

Let $A_{11} = \sigma_n^2 (Z_k)^{-1} + \lambda_{k+1} r_{k+1,k+1}^T$, $A_{12} = \lambda_{k+1} r_{k+1,k+1}^T$, $A_{21} = \lambda_{k+1}^2 r_{k+1,k+1}^T$, and $A_{22} = \lambda_{k+1}^2 r_{k+1,k+1}^2 + \sigma_n^2$, then (E.3) becomes

\[ Z_{k+1} = \begin{bmatrix} Z_k - K \lambda_{k+1} Z_k r_{k+1}^H Z_k & -K \lambda_{k+1} r_{k+1,k+1} Z_k r_{k+1} \\ -K \lambda_{k+1} r_{k+1,k+1}^H Z_k & K \left( \lambda_{k+1} r_{k+1}^H Z_k r_{k+1} + \sigma_n^2 \right) \end{bmatrix}, \]

where

\[ K = \frac{1}{\lambda_{k+1} r_{k+1}^T Z_k r_{k+1} + \lambda_{k+1}^2 r_{k+1,k+1}^2 + \sigma_n^2}. \]
Let $\Sigma_k$ be decomposed to $U \Phi_k U^H$, where $\phi_1 \geq \phi_2 \geq \cdots \geq \phi_k$ are the eigenvalues of $\Sigma_k$. Then, the upper bound of the SINR is given by

$$\text{SINR} = \frac{1}{\sigma_n^2} \frac{r_k^H \left( \sigma_n^4 \Sigma_k^{-2} \right) r_k + |r_{k,k}|^2}{r_k^H \left( \sigma_n^6 \Sigma_k^{-3} \right) r_k + |r_{k,k}|^2}$$

\hspace{1cm} (F.1)

$$= \frac{1}{\sigma_n^2} \frac{\sum_{i=1}^{k-1} \sigma_n^4 \phi_i^{-2} |r'_{i,k}|^2 + |r_{k,k}|^2}{\sum_{i=1}^{k-1} \sigma_n^6 \phi_i^{-3} |r'_{i,k}|^2 + |r_{k,k}|^2}$$

\hspace{1cm} (F.2)

$$\leq \sum_{i=1}^{k-1} \phi_i^{-1} |r'_{i,k}|^2 + \frac{|r_{k,k}|^2}{\sigma_n^2}$$

\hspace{1cm} (F.3)

$$= r_k^H \Sigma_k^{-1} r_k + \frac{|r_{k,k}|^2}{\sigma_n^2},$$

\hspace{1cm} (F.4)

where $r'_k = [r'_{1,k}, \cdots, r'_{k-1,k}]^T = U r_k$ and (F.3) is from the Cauchy-Schwarz inequality.

Next, with $A = r_k^H \left( \sigma_n^4 \Sigma_k^{-2} \right) r_k + |r_{k,k}|^2$ and $B = r_k^H \left( \sigma_n^6 \Sigma_k^{-3} \right) r_k + |r_{k,k}|^2$, we can show

$$A - B = r_k^H \left( \sigma_n^4 \Sigma_k^{-2} - \sigma_n^6 \Sigma_k^{-3} \right) r_k$$

\hspace{1cm} (F.5)

$$= \sum_{i=1}^{k-1} \left( \sigma_n^4 \phi_i^{-2} - \sigma_n^6 \phi_i^{-3} \right) |r'_{i,k}|^2 \geq 0,$$

\hspace{1cm} (F.6)

where (F.6) follows from $\sigma_n^2 \phi_i^{-1} = \sigma_n^2 / (\sigma_n^2 + \epsilon) \leq 1$. Hence,

$$\text{SINR} = \frac{1}{\sigma_n^2} \frac{A^2}{B} \geq \frac{1}{\sigma_n^2} A = \frac{\sigma_n^2 r_k^H \Sigma_k^{-2} r_k + |r_{k,k}|^2}{\sigma_n^2}.$$ 

\hspace{1cm} (F.7)

This becomes the lower bound of the SINR.
APPENDIX G

PROOF OF THEOREM 4.4.2

Let $H_{1:k-1}$ be a matrix generated from the first $k - 1$ columns of $H$. Since $H_{1:k-1} = Q \begin{bmatrix} R_{11,k} \\ 0 \end{bmatrix}$, the matrices $R_{11,k}R_{11,k}^H$ and $H_{1:k-1}^H H_{1:k-1}$ share same eigenvalues. For an i.i.d. random matrix $H$, the elements of $r_k$ are zero-mean and independent with variance of $\frac{1}{L}$. According to [68, Lemma 2.29], as $N, L \to \infty$ with $\beta = L/N$,

$$B_{k}^{\text{upper}} = r_k^H (\sigma_n^2 I + \lambda_{\min} R_{11,k} R_{11,k}^H)^{-1} r_k \to \frac{\gamma \beta}{\sigma_n^2} \int_0^\infty \frac{1}{1 + \frac{\lambda_{\min}}{\sigma_n} x} f_\eta(x) dx$$

where $f_\eta(x)$ is an empirical eigenvalue distribution of $H_{1:k-1}^H H_{1:k-1}$. According to the Marcenko-Pastur law [68, Theorem 2.35], as $N, L \to \infty$ with $\beta = L/N$, $f_\eta(x)$ converges to

$$f_\eta(x) \to f_\eta^*(x) = \frac{\sqrt{(x - (1 - \sqrt{\gamma \beta})^2)^+ (1 + \sqrt{\gamma \beta})^2 - x)^+}}{2\pi \gamma \beta x},$$

where $(x)^+ = \max(0, x)$. Hence, from (G.1) and (G.2), we obtain

$$B_{k}^{\text{upper}} \to \frac{\gamma \beta}{\sigma_n^2} \int_0^\infty \frac{1}{1 + \frac{\lambda_{\min}}{\sigma_n} x} f_\eta^*(x) dx \approx \frac{1}{2\lambda_{\min}} \left( -1 - (1 - \gamma \beta) \frac{\lambda_{\min}}{\sigma_n^2} + G \left( \frac{\lambda_{\min}}{\sigma_n^2}, \gamma \beta \right) \right).$$

(G.3)
In a similar manner, the lower bound converges to

\[ B_{\text{lower}}^k = \sigma_n^2 r_k^H (\sigma_n^2 I + \lambda_{\text{max}} R_{11,k} R_{11,k}^H)^{-2} r_k \]

\[ \longlongrightarrow \gamma \beta \int_0^\infty \frac{1}{(1 + \frac{\lambda_{\text{max}}}{\sigma_n^2} x)} f_\eta(\eta) dx \]

\[ = \frac{1}{2\sigma_n^2} \left( - (1 - \gamma \beta) + \frac{1 + \gamma \beta + (1 - \gamma \beta)^2 \lambda_{\text{max}}}{G \left( \frac{\lambda_{\text{max}}}{\sigma_n^2}, \gamma \beta \right)} \right). \]
APPENDIX H

PROOF OF COROLLARY 4.4.3

Using the Taylor series expansion, we have

\[ B_{upper, \infty} = \frac{1}{2\lambda_{\min}} \left( -1 - (1 - \gamma \beta) \frac{\lambda_{\min}}{\sigma^2_n} + G \left( \frac{\lambda_{\min}}{\sigma^2_n}, \gamma \beta \right) \right) = \frac{1}{\sigma^2_n} \sum_{i=0}^{\infty} \frac{a_i}{i!} \frac{2^i}{\sigma^2_n}, \]  

\[(H.1)\]

where

\[ a_i = \frac{1}{2\lambda_{\min}} \frac{d^i}{dx^i} \left( -x - (1 - \gamma \beta) \lambda_{\min} + x G \left( \frac{\lambda_{\min}}{x}, \gamma \beta \right) \right) \bigg|_{x=0}. \]  

\[(H.2)\]

Note that \(a_0 = 0\) and \(a_1 = \frac{1}{\lambda_{\min}} \frac{\gamma \beta}{1 - \gamma \beta}\). Hence, \(B_{upper, \infty}\) approaches \(a_1 = \frac{1}{\lambda_{\min}} \frac{\gamma \beta}{1 - \gamma \beta}\). In addition, letting

\[ \mu(x) = -1 - (1 - \gamma \beta) \lambda_{\max} x + \sqrt{1 + 2(1 + \gamma \beta) \lambda_{\max} x + (1 - \gamma \beta)^2 \lambda_{\max}^2 x^2}, \]  

\[(H.3)\]

then we can show that \(d\mu(x)/dx \geq 0\) for \(x \geq 0\). Since the term \(B_{upper, \infty} \propto \mu \left( \frac{1}{\sigma^2_n} \right)\), \(B_{upper, \infty}\) monotonically increases as a function of \(\frac{1}{\sigma^2_n}\). When \(\sigma^2_n \rightarrow 0\), the right-hand side of (H.1) monotonically increases and approaches \(\frac{1}{\lambda_{\min}} \frac{\gamma \beta}{1 - \gamma \beta}\).

On the other hand, the term \(B_{lower, \infty}\) in (4.57) is expressed as

\[ B_{lower, \infty} = \frac{1}{2\sigma^2_n} \left( - (1 - \gamma / \beta) + \frac{1 + \gamma / \beta + (1 - \gamma / \beta)^2 \lambda_{\max}}{\sigma^2_n G \left( \frac{\lambda_{\max}}{\sigma^2_n}, \gamma \beta \right)} \right) = \frac{1}{2\sigma^2_n} \sum_{i=0}^{\infty} \frac{b_i}{i!} \frac{2^i}{\sigma^2_n}. \]  

\[(H.4)\]

Since \(b_0 = b_1 = 0\), the lower bound converges to zero as \(\sigma^2_n \rightarrow 0\).
APPENDIX I

PROOF OF LEMMA 4.4.4

Let $\eta_1, \eta_2, \ldots, \eta_{k-1}$ be the unordered eigenvalues of $\mathbf{R}_{11,k}\mathbf{R}_{11,k}^H$. The scaling gain in (4.66) can be expressed as

$$E_{\mathbf{H}} \left[ \exp \left( -\frac{K}{2} \sigma_n^2 \mathbf{r}_k^H \Sigma_k^{-1} \mathbf{r}_k \right) \right]$$

$$\leq E_{\mathbf{H}} \left[ \exp \left( -\frac{K}{2} \sigma_n^2 \mathbf{r}_k^H \left( \sigma_n^2 \mathbf{I} + \lambda_{\text{max}} \mathbf{R}_{11,k} \mathbf{R}_{11,k}^H \right)^{-2} \mathbf{r}_k \right) \right] \quad (I.1)$$

$$= E_{\mathbf{H}} \left[ \prod_{i=1}^{k-1} \exp \left( -\frac{K}{2} \frac{\sigma_n^2}{(\lambda_{\text{max}} \eta_i + \sigma_n^2)^2} |r_{i,k}|^2 \right) \right] \quad (I.2)$$

$$= E_{\mathbf{R}_{11,k}} \left[ \prod_{i=1}^{k-1} E_{r_{i,k}} \left[ \exp \left( -\frac{K}{2} \frac{\sigma_n^2}{(\lambda_{\text{max}} \eta_i + \sigma_n^2)^2} |r_{i,k}|^2 \right) \right] \right] \mathbf{R}_{11,k} \right] \quad (I.3)$$

$$= E_{\mathbf{R}_{11,k}} \left[ \prod_{i=1}^{k-1} \frac{1}{1 + \frac{K}{2} \frac{\sigma_n^2}{(\lambda_{\text{max}} \eta_i + \sigma_n^2)^2}} \right], \quad (I.4)$$

where (I.3) is from $E[x] = E[E[x|y]]$ and (I.4) follows from the fact that $r_{i,k}$ is $\mathcal{CN}(0,1)$ and independent of $\mathbf{R}_{11,k}$. Letting $\mathbf{H}_{1:k-1}$ be a matrix generated from the first $k-1$ columns of $\mathbf{H}$, then the matrices $\mathbf{R}_{11,k}\mathbf{R}_{11,k}^H$ and $\mathbf{H}_{1:k-1}^H \mathbf{H}_{1:k-1}$ share same eigenvalues. The pdf of the unordered eigenvalues of $\mathbf{H}_{1:k-1}^H \mathbf{H}_{1:k-1}$ is given by [108]

$$f_{\eta_1, \ldots, \eta_{k-1}}(x_1, \ldots, x_{k-1})$$

$$= \frac{1}{(k-1)!} \exp \left( -\sum_{i=1}^{k-1} x_i \right) \prod_{i=1}^{k-1} \frac{x_i^{L-k+1}}{(k-1-i)!(L-i)} \prod_{i<j} (x_i - x_j)^2, \quad (I.5)$$

which completes the proof.
We derive a covariance matrix of the perturbation noise in LMS filter coefficients following the convergence analysis in [85]. Let filter coefficients, input signal, and desired signal of an LMS adaptive filter be $w_n$, $u_n$, and $d_n$, respectively. The LMS update rule is given by

$$w_{n+1} = w_n + \xi u_n (d_n - w_n^H u_n)^*.$$  \hspace{1cm} (J.1)

A Wiener solution of this adaptive filter is given by $w_o = \text{Cov}(u_n)^{-1}\text{Cov}(u_n,d_n)$. Denoting an error signal as $\epsilon_n = d_n - w_o^H u_n$, we can show that $E[|\epsilon_n|^2] = \sigma_d^2 - \text{Cov}(u_n,d_n)^H \text{Cov}(u_n)^{-1}\text{Cov}(u_n,d_n)$ [85]. Hence, $w_n - w_o$ becomes a perturbation vector due to misadjustment errors of the adaptive filter. We define the covariance matrix $K_n = \text{Cov}(w_n - w_o)$. If we subtract $w_o$ from both sides of (J.1), then the LMS weight update rule is written

$$w_{n+1} - w_o = w_n - w_o + \xi u_n (d_n - w_n^H u_n)^*$$  \hspace{1cm} (J.2)

$$= w_n - w_o + \xi u_n (\epsilon_n - (w_n - w_o)^H u_n)^*$$  \hspace{1cm} (J.3)

$$= (I - \xi u_n u_n^H) (w_n - w_o) + \xi u_n \epsilon_n^*.$$  \hspace{1cm} (J.4)

Using the direct-averaging method [85], we can approximate the solution of the difference equation in (J.4) as that of the equation

$$w_{n+1} - w_o = (I - \xi \text{Cov}(u_n)) (w_n - w_o) + \xi u_n \epsilon_n^*.$$  \hspace{1cm} (J.5)

From (J.5), we have

$$K_{n+1} = (I - \xi \text{Cov}(u_n)) K_n (I - \xi \text{Cov}(u_n))$$

$$+ \xi^2 \text{Cov}(u_n) E[|\epsilon_n|^2].$$ \hspace{1cm} (J.6)
Since $K_{n+1} = K_n$ in steady state, (J.6) becomes

$$
\xi \text{Cov}(u_n)K_n \text{Cov}(u_n) - \text{Cov}(u_n)K_n - K_n \text{Cov}(u_n)
= -\xi \text{Cov}(u_n)E[|\epsilon_n|^2].
$$

(J.7)

When the input signal $u_n$ is white, i.e., $\text{Cov}(u_n) = \alpha I$, $K_n$ is obtained as

$$
K_n = \frac{\xi}{2 - \alpha \xi} E[|\epsilon_n|^2]I
$$

(J.8)

$$
\approx \frac{\xi}{2} E[|\epsilon_n|^2]I,
$$

(J.9)

where (J.9) follows from the assumption that the step size $\xi$ is small. Due to the assumption on the white input, this results can be used for the analysis of the adaptive channel estimator in Section 5.4.1.

On the other hand, for a correlated input and with $K_n = \frac{\xi}{2} E[|\epsilon_n|^2]I$, the left-hand side of (J.7) is approximated to $\left(1 - \xi/2\right)(-\xi \text{Cov}(u_n)E[|\epsilon_n|^2]) \approx -\xi \text{Cov}(u_n)E[|\epsilon_n|^2]$ for small $\xi$, which equals to the right-hand side. Hence, the solution (J.9) approximately holds for the correlated input. Hence, we can also use this result for the analysis of the LMS DA-TEQ in Section 5.4.2.
APPENDIX K

DERIVATION OF (5.22)

We derive the excess MSE of the LMS DA-TEQ using the results in [89, equation 6.5.6]. For sufficiently small $\xi$,

$$\mathcal{M}_k^{\text{excess}} \approx \xi \mathcal{M}_k^{\text{min}} \left( \frac{\text{tr} \left( \text{Cov} \left( \begin{bmatrix} y_n \\ \bar{s}_{n,k} \end{bmatrix} \right) \right)}{2} \right).$$  \hspace{1cm} (K.1)

Since

$$\text{Cov} \left( \begin{bmatrix} y_n \\ \bar{s}_{n,k} \end{bmatrix} \right) = \begin{bmatrix} H_n H_n^H + R & \sigma_s^2 \left[ H_{n,1:k-1} \quad H_{n,k+1:K} \right] \\ \sigma_s^2 \left[ H_{n,1:k-1} \quad H_{n,k+1:K} \right]^H & \sigma_s^2 I_{K-1,K-1} \end{bmatrix}$$  \hspace{1cm} (K.2)

we have

$$\mathcal{M}_k^{\text{excess}} \approx \xi \mathcal{M}_k^{\text{min}} \left( \frac{\text{tr} \left( H_n H_n^H + R \right) + \sigma_s^2 (K-1)}{2} \right).$$  \hspace{1cm} (K.3)
REFERENCES


[47] H. V. Poor and S. Verdu, “Probability of error in MMSE multiuser
1997.

SINR for the MMSE MIMO receiver and performance analysis,” IEEE

BLAST,” Proc. IEEE Global Telecommunication Conference (GLOBE-

channels via modified sphere decoding,” IEEE Trans. Wireless Commu-

with soft detection based on hard sphere decoding,” IEEE Trans. Com-

algorithms and VLSI implementation,” IEEE Journal on Selected Areas

[53] J. Jalden and B. Ottersten, “Parallel implementation of a soft output
sphere decoder,” Proc. IEEE Ashilomar Conference on Signals, Systems,

[54] Y. L. C. de Jong and T. J. Wilink, “Iterative tree search detection
930-935, June 2005.

soft-MIMO detector based on the fixed-complexity sphere decoder,” Proc.
IEEE International Conference on Acoustics, Speech, and Signal Process-
ing (ICASSP), April 2008, pp. 2669-2672.


[57] Y. S. Han, C. R. P. Hartmann, and C. Chen, “Efficient priority-first
search maximum-likelihood soft-decision decoding of linear block codes,”


