Title
Noncoherent Detection Schemes for Multi-User/Multi-Node Communication Systems

Permalink
https://escholarship.org/uc/item/4rj187qt

Author
Poorkasmaei, Sina

Publication Date
2014

License
CC BY-NC-ND 4.0

Peer reviewed|Thesis/dissertation
Noncoherent Detection Schemes
for Multi-User/Multi-Node Communication Systems

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Sina Poorkasmaei

Dissertation Committee:
Professor Hamid Jafarkhani, Chair
Professor Ender Ayanoglu
Professor A. Lee Swindlehurst

2014
DEDICATION

To my parents
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................. v
LIST OF TABLES ................................................................. viii
LIST OF ALGORITHMS ......................................................... ix
ACKNOWLEDGMENTS ............................................................ x
CURRICULUM VITAE ............................................................. xi
ABSTRACT OF THE DISSERTATION .......................................... xiii

## 1 Introduction
1.1 Point-to-Point Differential Space-Time Modulation ............... 1
1.2 Multi-User Differential Space-Time Modulation .................. 2
1.3 Distributed Beamforming ............................................... 3
   1.3.1 Adaptive DBF with Random Phase Adjustments .............. 3
   1.3.2 Adaptive DBF with Deterministic Phase Adjustments ....... 4
1.4 Overview of Thesis ...................................................... 5
1.5 Notation ................................................................. 5

## 2 Synchronous Orthogonal Differential Decoding for MAC Systems
2.1 System Model ......................................................... 8
2.2 Differential Encoding and Decoding ................................ 9
   2.2.1 Differential Encoding ........................................ 9
   2.2.2 Differential Decoding ....................................... 10
2.3 Extension to Any Number of Receive Antennas .................... 19
2.4 Diversity Analysis ................................................... 22
2.5 Simulation Results .................................................... 24

## 3 Asynchronous Orthogonal Differential Decoding for MAC Systems
3.1 System Model .......................................................... 32
3.2 Differential Encoding ............................................... 36
3.3 Differential Decoding ............................................... 37
   3.3.1 Low Complexity Decoding Schemes ......................... 38
   3.3.2 Optimal Multiple Partition Decoding Schemes ............ 45
3.4 Diversity Analysis .................................................. 47
3.5 Simulation Results .................................................. 49

4 Distributed Beamforming Using Feedback Control Based on TSVQ 55
4.1 System Model ....................................................... 56
4.2 TSVQ-Based DBF Algorithms for Static Channels ............... 58
  4.2.1 Algorithm for \( N = 2 \) Transmitters .......................... 59
  4.2.2 Algorithm for a Larger Number of Transmitters .............. 64
4.3 TSVQ-Based DBF Algorithms for Time-Varying Channels ....... 73
  4.3.1 Algorithm for \( N = 2 \) Transmitters .......................... 74
  4.3.2 Algorithm for a Larger Number of Transmitters .............. 76
4.4 Simulation Results .................................................. 80

5 Conclusion .......................................................... 85

Appendices 93
A Proof of Proposition 2.1 ............................................. 93
B Noise Power Approximation .......................................... 95
C Proof of Proposition 3.1 ............................................. 96
D Proof of Theorem 3.1 ................................................ 97
E Proof of Theorem 3.3 ................................................ 98
F Proof of Theorem 4.2 ................................................ 102
G Proof of Proposition 4.2 ............................................. 103
H Proof Sketch of Theorem 4.3 ....................................... 109
I Proof of Corollary 4.1 ............................................... 110
J Proof of Theorem 4.4 ................................................ 111
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Block diagram of differential encoder.</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Chain corresponding to the decoding of $P^l$.</td>
<td>15</td>
</tr>
<tr>
<td>2.3</td>
<td>Block diagram of the IUIF differential decoders.</td>
<td>18</td>
</tr>
<tr>
<td>2.4</td>
<td>Performance of coherent and differential schemes at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.</td>
<td>26</td>
</tr>
<tr>
<td>2.5</td>
<td>Performance of coherent and differential schemes at a rate of 2 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.</td>
<td>26</td>
</tr>
<tr>
<td>2.6</td>
<td>Performance of coherent and differential schemes at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 3 receive antennas.</td>
<td>27</td>
</tr>
<tr>
<td>2.7</td>
<td>Performance of coherent and differential schemes at a rate of 2 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 3 receive antennas.</td>
<td>27</td>
</tr>
<tr>
<td>2.8</td>
<td>Comparison of our schemes and the heuristic differential scheme in [20] at a rate of 1 b/(s Hz) for 2 users each with 1 transmit antenna and 1 receiver with 2 receive antennas.</td>
<td>28</td>
</tr>
<tr>
<td>3.1</td>
<td>Block diagram of differential encoder.</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Chain corresponding to the decoding of $P^l_j$.</td>
<td>43</td>
</tr>
<tr>
<td>3.3</td>
<td>Block diagram of differential decoders.</td>
<td>46</td>
</tr>
<tr>
<td>3.4</td>
<td>Performance of the proposed asynchronous differential schemes for $\tau_2 - \tau_1 = T_s/2$, the synchronous differential schemes in Chapter 2, and the synchronous coherent schemes using ZF and ML decoding at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.</td>
<td>50</td>
</tr>
<tr>
<td>3.5</td>
<td>Performance of the proposed asynchronous differential schemes for $\tau_2 - \tau_1 = T_s/2$, the synchronous differential schemes in Chapter 2, and the synchronous coherent schemes using ZF and ML decoding at a rate of 2 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.</td>
<td>50</td>
</tr>
<tr>
<td>3.6</td>
<td>Performance of the proposed asynchronous differential schemes for $\tau_2 - \tau_1 = T_s/2$, the synchronous differential schemes in Chapter 2, and the synchronous coherent schemes using ZF and ML decoding at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 3 receive antennas.</td>
<td>51</td>
</tr>
<tr>
<td>3.7</td>
<td>Performance of the proposed asynchronous differential schemes for $\tau_2 - \tau_1 = T_s/2$, the synchronous differential schemes in Chapter 2, and the synchronous coherent schemes using ZF and ML decoding at a rate of 2 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 3 receive antennas.</td>
<td>51</td>
</tr>
</tbody>
</table>
3.8 Performance of the proposed asynchronous differential schemes for $\tau_2 - \tau_1 = T_s/2$ and the synchronous coherent scheme using ML decoding at a rate of 1 b/(s Hz) for 2 users each with 4 transmit antennas and 1 receiver with 1 receive antenna. ................................................................. 52

3.9 Performance of the proposed asynchronous differential schemes for $\tau_{j+1} - \tau_j = T_s/3$, $\forall j$, and the synchronous coherent scheme using ML decoding at a rate of 1 b/(s Hz) for 3 users each with 2 transmit antennas and 1 receiver with 2 receive antennas. ................................................................. 53

3.10 Comparison of the proposed asynchronous differential schemes using Method 3 for different relative time delays $\Delta \tau = \tau_2 - \tau_1$ at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas. ................................................................. 53

4.1 Proposition 4.1. ................................................................. 59

4.2 Example of phase synchronization using Algorithm 1 for $K = 5$ and $B = 1$. The blue and red segments represent the vectors corresponding to Nodes 1 and 2, respectively, after each stage. The tails of both vectors are at point $(0,0)$. The green dashes display the boundaries of the quantization regions after each stage. ................................................................. 63

4.3 Proposition 4.3. ................................................................. 65

4.4 Example of phase synchronization using Algorithm 2 for $N = 1000$, $K = 5$, $B = 1$ and $L = 25$. The blue segments represent the vectors corresponding to the nodes, and the red segment represents the sum of the vectors corresponding to the nodes after each stage. The tails of all the vectors are at point $(0,0)$. The green dashes display the boundaries of the quantization regions. ................................................................. 71

4.5 Histogram of the angles corresponding to the vectors in a sample run of Algorithm 2 for $N = 1000$, $K = 5$, $B = 1$ and $L = 25$. ................................................................. 72

4.6 Example of phase synchronization using Algorithm 3 for $K = 3$ and $B = 1$ where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/8$. The blue and red segments represent the vectors corresponding to Nodes 1 and 2, respectively, after each stage/iteration. The tails of both vectors are at point $(0,0)$. ................................................................. 78

4.7 Example of phase synchronization using Algorithm 4 for $N = 100$, $K = 3$ and $B = 1$ where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/35$. The blue segments represent the vectors corresponding to the nodes after each iteration. The tails of all the vectors are at point $(0,0)$. ................................................................. 80

4.8 The values of $\mathbb{E}\{\rho_N[t]\}$ achieved using Algorithm 1 for $K = 15$ and $B = 1, 2, 3$, the corresponding lower bounds based on Theorem 4.2, and the values of $\mathbb{E}\{\rho_N[t]\}$ achieved using the existing algorithms for static channels. ................................................................. 81

4.9 The values of $\mathbb{E}\{\rho_N[t]\}$ achieved using Algorithm 2 for $N = 1000$, $K = 5$, $B = 1, 2, 3$ and $L = 25$, the corresponding asymptotic values based on Theorem 4.4, and the values of $\mathbb{E}\{\rho_N[t]\}$ achieved using the existing algorithms for static channels. ................................................................. 82

4.10 The values of $\mathbb{E}\{\rho_N[t]\}$ achieved using Algorithm 3 for $K = 3$ and $B = 1, 2, 3$ and using the existing algorithms for time-varying channels where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/8$. ................................................................. 83
4.11 The values of $E\{\rho_N[t]\}$ achieved using Algorithm 4 for $N = 100$, $K = 3$ and $B = 1, 2, 3$ and using the existing algorithms for time-varying channels where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/35$. 83
LIST OF TABLES

2.1 Execution time of decoding one frame (128 data symbols) in seconds using Matlab for 2 users with 2 transmit antennas and a receiver with 2 receive antennas .................................................. 30
# LIST OF ALGORITHMS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

My deepest gratitude goes to my advisor Professor Hamid Jafarkhani not only for his support and guidance, but also for helping me get through some difficult times I faced during my studies.

I would like to express my sincere appreciation to my committee members, Professor Ender Ayanoglu and Professor A. Lee Swindlehurst, for their constructive comments and criticisms.

I would also like to thank my labmates for creating an enjoyable environment.

Last but not least, I would like to thank my family for their unconditional love and support. None of my achievements throughout my life would have been possible without their encouragement.

This research was supported in part by the NSF Award CCF-0963925.
EDUCATION

Doctor of Philosophy in Electrical and Computer Engineering 2014
University of California, Irvine Irvine, CA

Master of Science in Electrical and Computer Engineering 2008
University of California, Irvine Irvine, CA

Bachelor of Science in Computer Engineering 2007
American University in Dubai Dubai, UAE

RESEARCH EXPERIENCE

Graduate Research Assistant 2007–2014
University of California, Irvine Irvine, CA

TEACHING EXPERIENCE

Teaching Assistant 2009–2014
University of California, Irvine Irvine, CA
PREPRINTS

Distributed Beamforming Using Feedback Control: A Quantization Approach
Submitted

JOURNAL PUBLICATIONS

Asynchronous Orthogonal Differential Decoding for Multiple Access Channels
IEEE Transactions on Wireless Communications
Orthogonal Differential Modulation for MIMO Multiple Access Channels with Two Users
IEEE Transactions on Communications

CONFERENCE PUBLICATIONS

Asynchronous Orthogonal Differential Modulation for MAC Systems
IEEE Global Communications Conference (GLOBECOM)
Differential Distributed Space-Time Coding with Imperfect Synchronization
IEEE Global Communications Conference (GLOBECOM)
We design and analyze noncoherent detection schemes for multi-user/multi-node communication systems where neither the transmitters nor the receiver knows the channel. First, we propose differential decoding schemes for two-user MIMO systems based on orthogonal space-time block codes (OSTBCs). We derive low complexity differential decoders for users with two transmit antennas. We also present differential decoding schemes that achieve full diversity, perform significantly better than the existing schemes, and work for any square OSTBC, but need higher decoding complexity compared to our low complexity decoders. Moreover, we analyze the diversity of the proposed schemes. To the best of our knowledge, our low complexity schemes are the first low complexity differential schemes for multi-user systems.

We then propose differential decoding schemes for asynchronous multi-user MIMO systems based on OSTBCs. We derive novel low complexity differential decoders by performing interference cancelation in time. The decoding complexity of these schemes grows linearly with the number of users. We also present differential decoding schemes that perform significantly better than our low complexity decoders and outperform the existing synchronous differential schemes but require higher decoding complexity compared to our low complexity decoders.
decoders. The proposed schemes work for any square OSTBC, any number of users, and any number of receive antennas. Furthermore, we analyze the diversity of the proposed schemes and derive conditions under which our schemes provide full diversity. To the best of our knowledge, the proposed differential detection schemes are the first differential schemes for asynchronous multi-user systems.

Finally, we present novel distributed beamforming (DBF) algorithms using feedback control based on Tree-Structured Vector Quantization (TSVQ). We develop TSVQ-based DBF algorithms for static channels. To the best of our knowledge, the proposed algorithms are the first deterministic DBF methods that can feed back more than 1 bit per time slot for faster phase synchronization. We analytically prove that our TSVQ-based DBF algorithms attain phase synchronization in probabilistic senses. Moreover, we modify our TSVQ-based DBF algorithms to enable them to track time-varying channels without the knowledge of the channel. Simulation results demonstrate that our algorithms significantly outperform the existing adaptive DBF algorithms for static and time-varying channels.
1.1 Point-to-Point Differential Space-Time Modulation

Various space-time modulation techniques to achieve transmit diversity have been proposed in the literature [1]. In most cases, it is assumed that the channel state information (CSI) is perfectly known at the receiver [2], [3]. This is a reasonable assumption when the channel changes slowly and can be estimated by transmitting known training symbols. However, this is not always possible, and there is a tradeoff between frame length and accuracy of the channel estimation [4]. Therefore, the effects of channel estimation error make it desirable to use schemes that avoid such an estimation.

Many differential space-time coding schemes have been proposed in the literature where neither the transmitter nor the receiver knows the CSI. A differential coding scheme based on orthogonal designs with two transmit antennas was proposed in [5]. This differential detection scheme was generalized to multiple transmit antennas in [6]. There is about 3-dB loss in performance for this differential detection scheme compared to the corresponding coherent detection. In [7], a differential unitary space-time code was proposed. A differential
modulation scheme based on unitary group codes was independently presented in [8]. In these schemes, the constellation of unitary matrices or group codes are utilized to encode the symbols. While these schemes can be theoretically used for any number of transmit and receive antennas and any signal constellation, their exponential decoding complexity in data rate and number of transmit antennas make them impractical for high data rates or a large number of transmit antennas. Using space-time block codes (STBCs), a differential modulation method with linear decoding complexity was proposed in [9]. This scheme, which is a special case of the scheme in [5], was generalized to constellations with multiple amplitudes in [10] and [11] and to quasi-orthogonal space-time block codes (QOSTBCs) of [12] in [13].

1.2 Multi-User Differential Space-Time Modulation

Multi-user detection schemes with simple coherent detection structures for multiple access channels (MACs) have garnered significant attention. When there is no channel information available at the transmitters, for \( J \) users equipped with \( N \) transmit antennas, [14] shows how to cancel the interference using \( NJ \) receive antennas. To reduce the number of required receive antennas to \( J \), [15] presents a scheme for users with two transmit antennas using the properties of orthogonal space-time block codes (OSTBCs) [3]. This method is extended in [16] to a higher number of transmit antennas but only for \( J = 2 \) users. Unfortunately, these methods do not work for a general case of complex constellations, \( N > 2 \) transmit antennas, and \( J > 2 \) users [17]. In order to address these shortcomings, [17] suggests a method based on QOSTBCs and extends the above multi-user detection schemes to any constellation, any number of users, and any number of transmit antennas. This is achieved by a moderate increase of decoding complexity due to the complexity tradeoff between OSTBCs and QOSTBCs. In [18], based on the concept of dominant error event regions
introduced in [19], space-time/frequency code design criteria are derived for fading MIMO MACs. Moreover, for the specific case of a two-user system, a code construction based on the modification of the Alamouti code is presented.

A differential modulation scheme for a two-user SISO system was proposed in [20]. This scheme was extended to a two-user MIMO system in [21]. However, these schemes have a high decoding complexity, and to the best of our knowledge, a low complexity differential modulation scheme for multi-user MIMO systems does not exist in the literature.

1.3 Distributed Beamforming

Distributed beamforming (DBF) is a form of cooperative communication where a cluster of transmitters create a virtual antenna array in order to send a common message to a remote destination. In other words, nodes combine their antenna resources and transmit their signals in a way that they coherently add up in a desired direction in space. Such an aggregation results in power gains proportional to the number of cooperating devices. In order to perform beamforming with distributed transmitters, unknown channel gains from each transmitter to the receiver and unknown phase offsets between the transmitters need to be compensated for. However, providing this information to the nodes is costly and impractical for a large number of nodes. Alternatively, adaptive beamforming using feedback control can be used to eliminate the need for this information.

1.3.1 Adaptive DBF with Random Phase Adjustments

Prior literature offers numerous different approaches to distributed phase synchronization. An iterative distributed phase synchronization algorithm using one bit of feedback at each time slot was proposed in [22]. In each iteration, all nodes first apply random phase ad-
justments to their signals independently and then transmit them to the receiver. The receiver measures the Received Signal Strength (RSS) and sends one bit of feedback indicating whether the random adjustments have improved the RSS or worsened it. Depending on whether the feedback is positive or negative, the nodes then maintain or discard the introduced phase adjustments. This procedure is iteratively repeated until the received phases attain a desired level of coherence. It is shown in [22] that this procedure leads to asymptotic coherence almost surely for a broad class of distributions for the random phase adjustments. By considering this algorithm as a local random search algorithm, [23] provides a comprehensive analysis of the fast convergence and linear scalability of the algorithm. In [24], an extension of this algorithm to the case of time-varying channels was introduced. In [25] and [26], an enhancement of this method was presented by exploiting negative feedback information to speed up convergence. In this algorithm, when negative feedback is received by the nodes, they invert the sign of the last phase adjustments and apply them before making the next phase adjustments. Other examples of feedback-based synchronization procedures include [27]-[31]. In [32], the original feedback-based iterative DBF algorithm was generalized to a multi-user scenario where $M$ separate clusters of nodes communicate with $M$ distinct receivers.

### 1.3.2 Adaptive DBF with Deterministic Phase Adjustments

In [33], an iterative distributed phase synchronization algorithm was introduced using a deterministic phase adjustment approach. In this method, nodes take turns and test a number of phase adjustments that are predefined in a set. After each adjustment, all nodes transmit their signals to the receiver simultaneously, and the receiver measures the RSS and sends one bit of feedback indicating whether the adjustments have improved the best measured RSS or not. After testing all possible phase adjustments in the predefined set, the node will apply the phase adjustment that resulted in the best RSS value. This process is
performed for all nodes in a one-by-one fashion. This algorithm was extended to the case of time-varying channels in [34]. Simulation results show that the deterministic approaches outperform the random methods in the case of static and time-varying channels. In [35], the authors presented a phase synchronization algorithm that requires each node to transmit only once during the synchronization process. This reduces the amount of power consumed to attain phase synchronization. They also analyze the performance of this algorithm and study the effect of noise on it. However, in their scheme, cooperative transmission of information is not possible during the phase synchronization stage. Thus, their scheme is different from the above deterministic DBF algorithms and the proposed DBF algorithms in this thesis.

1.4 Overview of Thesis

In this thesis, we design and analyze noncoherent detection schemes for multi-user/multi-node communication systems. The main goal is to design detection schemes where neither the transmitters nor the receiver requires to know the CSI in order to avoid the cost of performing channel estimation. The thesis is organized as follows. In Chapter 2, we introduce differential decoding schemes for two-user MIMO systems based on OSTBCs. In Chapter 3, we propose differential decoding schemes for asynchronous multi-user MIMO systems based on OSTBCs. In Chapter 4, we present novel DBF algorithms using feedback control based on Tree-Structured Vector Quantization (TSVQ). Chapter 5 concludes the thesis.

1.5 Notation

We use super-scripts \((\cdot)^*\) and \((\cdot)^\dagger\) to denote conjugate and conjugate transpose, respectively. \(\| \cdot \|_F\) indicates the Frobenius norm, and \(E[\cdot]\) represents the expected value. Also, we use \(I_n\) and \(0_n\) to denote the \(n \times n\) identity and zero matrices, respectively, and \(0_{m \times n}\) to denote the
$m \times n$ zero matrix. For a sequence of random variables $\{X_n\}_{n=1}^{\infty}$ and a random variable (or a fixed number) $X$, we write $X_N \xrightarrow{p} X$ as $N \to \infty$ if and only if $\lim_{n \to \infty} P(|X_n - X| > \epsilon) = 0$ for any $\epsilon > 0$. For any $a = (a_1, \cdots, a_n)^T \in \mathbb{R}^n$ and any $b > 0$, we define $a \mod b = (r_1, \cdots, r_n)^T$ where for $i = 1, \cdots, n$, $r_i \in (-b/2, b/2]$ and there exists an integer $q_i$ such that $a_i = bq_i + r_i$. Also, for functions $f(t)$ and $g(t)$, $f(t) \in o(g(t))$ if and only if $\lim_{t \to \infty} \frac{f(t)}{g(t)} = 0$, and $f(t) \in \omega(g(t))$ if and only if $\lim_{t \to \infty} \frac{f(t)}{g(t)} = \infty$. 
Chapter 2

Synchronous Orthogonal Differential Decoding for MAC Systems

In this chapter, we design differential detection schemes for two-user MIMO systems where neither the transmitters nor the receiver has knowledge of the channel. With a slow Rayleigh fading channel model, we present a differential encoding algorithm and derive low complexity noncoherent decoders for users with two transmit antennas based on three distinct decoding methods. These decoders achieve full transmit diversity. We also present additional differential decoding schemes that provide full diversity, outperform the existing differential schemes, and work for any square OSTBC. Simulation results show that our differential detection schemes provide good performance.

The rest of the chapter is organized as follows. In Section 2.1, we introduce the system model. In Section 2.2, we present the differential encoding and decoding for our differential modulation schemes. In Section 2.3, we show that our low complexity schemes can be extended to a system with two users each with two transmit antennas and one receiver with more than two receive antennas. We analyze the diversity of our schemes in Section 2.4, and
2.1 System Model

Consider a wireless communication system with two users each with \( N \) transmit antennas and one receiver with \( M \) receive antennas with a quasi-static flat Rayleigh fading channel. We define \( T \) as the block size of the system and \( K \) as the number of data symbols transmitted during one block. In the \( l \)th block, we define \( \mathbf{Y}^l \) as an \( M \times T \) received signal matrix whose \((m, t)\)th element \( y_{mt}^l \) is the received signal by antenna \( m \) at time slot \( t \). Also, we define \( \mathbf{C}^l \) and \( \mathbf{S}^l \) as \( N \times T \) transmitted signal matrices whose \((n, t)\)th elements \( c_{nt}^l \) and \( s_{nt}^l \) are the signals transmitted from the \( n \)th antenna at time slot \( t \) for Users 1 and 2, respectively. Finally, we define \( \mathbf{H} \) and \( \mathbf{G} \) as \( M \times N \) channel fading matrices whose \((m, n)\)th elements \( h_{mn} \) and \( g_{mn} \) are the channel fading coefficients from transmit antenna \( n \) to receive antenna \( m \) for Users 1 and 2, respectively, and \( \mathbf{N}^l \) as an \( M \times T \) white noise matrix. The channels are assumed to be unknown at both the transmitter and the receiver. The input-output relationship of the system can be modeled as

\[
\mathbf{Y}^l = \mathbf{H} \mathbf{C}^l + \mathbf{G} \mathbf{S}^l + \mathbf{N}^l. \tag{2.1}
\]

The entries of \( \mathbf{H} \) and \( \mathbf{G} \) are samples of independent zero-mean complex Gaussian random variables with variance 0.5 per real dimension, and the entries of \( \mathbf{N}^l \) are samples of independent complex Gaussian noises with mean 0 and variance \( 1/(2\text{SNR}) \) per real dimension. In this chapter, it is assumed that the data transmitted by the users in time block \( l \) reaches the base-station simultaneously and their transmissions are perfectly synchronized in time, phase, and frequency. Moreover, both users utilize the same frequency band for their transmission and do not use any orthogonal signature.
2.2 Differential Encoding and Decoding

2.2.1 Differential Encoding

In this section, we describe our differential encoding scheme for the two users each with $N$ transmit antennas. The block diagram of the differential encoder is shown in Fig. 2.1. First, we describe the encoding procedure for User 1. At a transmission rate of $b$ bits/(s Hz), we use a unit-length signal constellation that is symmetric with respect to the origin with $2^b$ elements such as $2^b$-PSK. Similar to the case of a single user, extension to arbitrary constellations is possible. For each block of $Kb$ bits, User 1 selects $K$ symbols and transmits them using a square OSTBC. This transmitted codeword also depends on the codeword and symbols transmitted in the previous block.

The encoding starts with the transmission of arbitrary square OSTBCs $C^0$ and $C^1$. As in the case of a single user, we could transmit only one OSTBC instead of two and the system would still work with minor changes. Because of space limitations, we will not discuss the details.
of this alternative. For block \( l \), we use the \( Kb \) input bits to pick \( K \) symbols \( p_1^l, \cdots, p_K^l \) from the unit-length constellation and construct the corresponding square OSTBC, \( P^l \). Assuming that \( C^{l-1} \) is the codeword of User 1 for the \( (l - 1) \)th block, we calculate \( C^l \) by

\[
C^l = C^{l-1} \cdot P^l
\]

and then transmit it at block \( l \). Note that the generated codeword \( C^l \) will be orthogonal as well.

The encoding procedure for User 2 is similar. The only difference is that in order for the decoder to be able to distinguish between the two users’ codewords, for User 2, we need to use a rotated version of the unit-length constellation\(^1\).

### 2.2.2 Differential Decoding

In this section, we present differential decoding schemes for both users. First, we derive low complexity decoders for users with two transmit antennas based on three different decoding methods. These decoders provide full transmit diversity. Then, we present additional decoding schemes based on these three methods that achieve full diversity, perform significantly better than the existing schemes, and work for any square OSTBC. We assume that the channel is unchanged within three consecutive time blocks.

#### Low Complexity Differential Decoding Schemes

In this subsection, we present low complexity decoders for users with two transmit antennas based on three different decoding methods. In the first method, we derive a low complexity decoder by obtaining a relationship between the received signals and the transmitted data

\(^1\)Using computer search we obtained the optimized rotation angles for all the proposed low complexity decoders using BPSK and QPSK to be \( \pi/2 \) and \( \pi/4 \), respectively.
signals for each user via canceling the interference of the other user and removing the channel matrices from the equations. Then, based on the decoder in the first method, we present two additional decoding methods to improve the performance. These methods use dynamic programming (DP) to efficiently decode the transmitted data signals. For our low complexity decoders, we assume that each user has two transmit antennas, i.e. $N = 2$, and the data matrices $P^l, Q^l$, and thus $C^l, S^l$, are Alamouti codes. For the sake of simplicity, we first assume that the receiver has two receive antennas, i.e. $M = 2$. Later in Section 2.3, we show how to extend our low complexity schemes to more than two receive antennas.

**Method 0:** With a small abuse of the notation, we consider $c^l_1, c^l_2$ and $s^l_1, s^l_2$ as the constellation points that are used in $C^l$ and $S^l$, respectively. Then using the channel equation in (2.1), the input-output relationship for receive antenna $m = 1, 2$ can be written as

$$
\left( y^l_{m1} \ y^l_{m2} \right) = \left( h_{m1} \ h_{m2} \right) \left( \begin{array}{cc} c^l_1 \ -c^l_2^* \\ c^l_2 \ c^l_1^* \end{array} \right) + \left( g_{m1} \ g_{m2} \right) \left( \begin{array}{cc} s^l_1 \ -s^l_2^* \\ s^l_2 \ s^l_1^* \end{array} \right) + \left( n^l_{m1} \ n^l_{m2} \right).
$$

(2.3)

We can equivalently write (2.3) using orthogonal matrices as

$$
Y^l_m = H_m C^l + G_m S^l + N^l_m
$$

(2.4)

where

$$
Y^l_m = \left( \begin{array}{cc} y^l_{m1} \ y^l_{m2} \\ y^l_{m2}^* \ -y^l_{m1}^* \end{array} \right), \quad H_m = \left( \begin{array}{cc} h_{m1} \ h_{m2} \\ h_{m2}^* \ -h_{m1}^* \end{array} \right), \quad G_m = \left( \begin{array}{cc} g_{m1} \ g_{m2} \\ g_{m2}^* \ -g_{m1}^* \end{array} \right), \quad N^l_m = \left( \begin{array}{cc} n^l_{m1} \ n^l_{m2} \\ n^l_{m2}^* \ -n^l_{m1}^* \end{array} \right).
$$

(2.5)

To design the decoder, first we need the following proposition:

**Proposition 2.1.** For any $l \geq 2$, the following relationship holds almost surely between the received signals and the transmitted data signals of User 1

$$
0_2 = P^{l-1}P^l A^l_0 + P^{l-1}A^l_1 + A^l_2 + \hat{N}^l
$$

(2.6)

Equation (2.6) does not hold when any of the determinants are zero, but that event has a zero probability.
where $A^t_0, A^t_1, A^t_2$ are functions of the received signals defined as

$$
A^t_0 = |\det(Y^t_1 \Gamma^t)|^{-1} \cdot Y^t_1 \Gamma^t, \quad A^t_1 = -\sum_{i=0}^{1} |\det(Y^t_{1-i} \Gamma^{t-i})|^{-1} \cdot Y^t_{1-i} \Gamma^{t-i},
$$

$$
A^t_2 = \left| \det(Y^t_{1-i} \Gamma^{t-i}) \right|^{-1} \cdot Y^t_{1-i} \Gamma^{t-i}.
$$

(2.7)

where $\Gamma^t = \sum_{i=0}^{1} (-1)^i |\det(Y^t_{1-i})|^{-1} \cdot Y^t_{1-i} Y^t_{1-i} \Gamma^{t-i}, \ell \geq 1$, and $\bar{N}^t$ is the effective noise matrix given by

$$
\bar{N}^t = C^t - 2 \sum_{i=0}^{2} \left( \Psi_2 N^t_{1-i} - \Psi_1 N^t_{1-i} \right) A^t_i.
$$

(2.8)

where $\Psi_i = |\det(FG_i)|^{-1} \cdot F^\dagger G_i^\dagger, i = 1, 2$, with $F = |\det(G_2)|^{-1} \cdot G_2^\dagger H_2 - |\det(G_1)|^{-1} \cdot G_1^\dagger H_1$.

**Proof.** See Appendix A.

Equation (2.6) is the key to our differential decoding algorithm. It does not include any channel matrices. It only contains the transmitted signals from User 1, the received signals and noise components. In other words, the interference from User 2 and the channel matrices are canceled in an exchange with the inter-block interference. Therefore, Equation (2.6) can be utilized to decode the transmitted signals without any channel knowledge and interference. Also, notice that from Eq. (2.8) $\bar{N}^t$ is not Gaussian since $A^t_i, i = 0, 1, 2$, is a function of the received signals and thus dependent on the noise terms $N^t_{1-i}, N^t_{2-i}, i = 0, 1, 2$; however, in what follows we assume that $\bar{N}^t$ is Gaussian by ignoring the dependency between the received signals and the noise terms $N^t_{1-i}, N^t_{2-i}, i = 0, 1, 2$. This is a common practice in differential modulation schemes as usually the noise products are ignored in the analysis [1].

Noticing that $\det(C^{t-2\dagger}) = 1$ and using Eq. (2.8), the effective noise power at block $l$, $\lambda^l$, when using Eq. (2.6) for decoding, can be derived as

$$
\lambda^l = (|\det(\Psi_1)| + |\det(\Psi_2)|) \sum_{i=0}^{2} \left| \det(A^t_i) \right|.
$$

(2.9)

In order to compute the effective noise power using Eq. (2.9), we now need to determine the values of $|\det(\Psi_1)|$ and $|\det(\Psi_2)|$. Since they both depend on the channel matrices, which
are unknown, we use the following formulas to approximate their values:

\[ |\text{det}(\Psi_1)| \approx |\text{det}(\bar{\Gamma}')|^{-1} \cdot \left( |\text{det}(Y_2^l)|^{-1} + |\text{det}(Y_2^{l-1})|^{-1} \right), \]  

(2.10)

\[ |\text{det}(\Psi_2)| \approx |\text{det}(\Gamma')|^{-1} \cdot \left( |\text{det}(Y_1^l)|^{-1} + |\text{det}(Y_1^{l-1})|^{-1} \right) \]  

(2.11)

where

\[ \bar{\Gamma}' = \sum_{i=0}^{1} (-1)^i |\text{det}(Y_2^{l-i})|^{-1} \cdot Y_1^{l-i}Y_2^{l-i}. \]  

(2.12)

The detailed derivation of the above approximations is presented in Appendix B. We denote the approximation for \( \lambda' \) using the above approximations by \( \hat{\lambda}' \).

We are now ready to present our first low complexity differential decoding scheme. One approach is to decode \( P_{l-1} \) and \( P_l \) jointly using Eq. (2.6). Therefore, for any block \( l \geq 2 \), we define the Inter-User Interference Free (IUIF) decoding using Method 0 as

\[ \{ \hat{P}_{l-1}, \hat{P}_l \} = \arg\min_{P_{l-1}, P_l} \| \Lambda^l(P_{l-1}, P_l) \|_F^2 \]  

(2.13)

where \( \Lambda^l(P_{l-1}, P_l) \) is given by

\[ \Lambda^l(P_{l-1}, P_l) = P_{l-1}P_lA_0 + P_{l-1}A_1 + A_2. \]  

(2.14)

Notice that for \( l = 2 \) in Eq. (2.14), \( P_1 = C_0^\dagger C_1 \) is the arbitrary data matrix at block 1 and is known at both the encoder and the decoder. Using this scheme, the information provided by Eq. (2.6) at time blocks other than \( l \) is ignored, and some performance is lost. Another approach to decode \( P_l \) at block \( l \) is to solve Eq. (2.6) using the decoded value for \( P_{l-1} \) at block \( l - 1 \). One may easily observe that by using this approach, a symbol-by-symbol decoding is possible. However, the decoded signals for \( P_{l-1} \) at block \( l - 1 \) may be erroneous, which can lead to error propagation and thus performance degradation. To avoid such performance losses, in this chapter, we also propose two novel methods to efficiently decode the transmitted signals by Users 1 and 2 as follows. Note that since \( HC_l \) and \( GS_l \) have the same role in (2.1), we can decode the signals of User 2 by replacing \( P_l \) with \( Q_l \) and switching \( H \) and \( G \) in the derivation of (2.6).
**Method 1 (Causal):** In Method 1, we decode $P_l$ based on Eq. (2.6) for all blocks $\ell = 2, \ldots, l$ together. Since the decoded signals at block $l$ using Method 1 do not depend on the received signals at blocks $\ell > l$, the decoder using Method 1 can be considered a causal system where the received signals at blocks $\ell \leq l$ are the input of the system and the decoded signals at block $l$ are the output of the system. For an efficient decoder, we utilize DP to find the best possible data matrix that maximizes the approximate joint probability distribution function (pdf) of the data matrices up to the current block. Using Eq. (2.6), we consider the following approximate joint pdf

$$
\hat{f}_1(P^2, \ldots, P^l) \propto \prod_{\ell=2}^{l} \exp \left\{ - (\hat{\lambda}^\ell)^{-1} \| \Lambda^\ell (P^{\ell-1}, P^\ell) \|_F^2 \right\} 
$$

(2.15)

Note that, as explained before, we have ignored the correlations of $\hat{N}^\ell$ at different blocks $\ell = 2, \ldots, l$ and assumed that they are independent. To maximize the above pdf, we only need to minimize $\sum_{\ell=2}^{l} (\hat{\lambda}^\ell)^{-1} \| \Lambda^\ell (P^{\ell-1}, P^\ell) \|_F^2$. For any block $l \geq 2$, we define the IUIF decoding using Method 1 as

$$
\hat{P}^l = \arg\min_{P^l} \Phi^l(P^l) 
$$

(2.16)

where $\Phi^l(P^l)$ is defined as

$$
\Phi^l(P^l) \triangleq \begin{cases} 
(\hat{\lambda}^2)^{-1} \| \Lambda^2(P^1, P^2) \|_F^2, & l = 2 \\
\min_{P^2, \ldots, P^{l-1}} \sum_{\ell=2}^{l-1} (\hat{\lambda}^\ell)^{-1} \| \Lambda^\ell (P^{\ell-1}, P^\ell) \|_F^2, & \text{otherwise.} 
\end{cases} 
$$

(2.17)

The optimization problem in (2.17) can be efficiently solved by utilizing DP. Using (2.17), for $l > 2$ we have

$$
\Phi^l(P^l) = \min_{P^{l-1}} \left\{ \min_{P^2, \ldots, P^{l-2}} \left[ \sum_{\ell=2}^{l-1} (\hat{\lambda}^\ell)^{-1} \| \Lambda^\ell (P^{\ell-1}, P^\ell) \|_F^2 \right] + (\hat{\lambda}^l)^{-1} \| \Lambda^l (P^{l-1}, P^l) \|_F^2 \right\} 
$$

$$
= \min_{P^{l-1}} \left\{ \sum_{\ell=2}^{l-1} (\hat{\lambda}^\ell)^{-1} \| \Lambda^\ell (P^{\ell-1}, P^\ell) \|_F^2 \right\} + (\hat{\lambda}^l)^{-1} \| \Lambda^l (P^{l-1}, P^l) \|_F^2 
$$

$$
= \min_{P^{l-1}} \left\{ \Phi^{l-1}(P^{l-1}) + (\hat{\lambda}^{l-1})^{-1} \| \Lambda^{L-1} (P^{l-1}, P^l) \|_F^2 \right\}. 
$$

(2.18)
Therefore, DP can be used to solve the optimization problem in (2.18) efficiently. As a result of storing the cost function of the previous block, $\Phi_{l-1}(P_{l-1})$, we only need to perform an optimization over $P_{l-1}$ for each time block $l$. In other words, instead of solving the optimization problem in (2.17) over all data matrices for the previous blocks, $P^2, \ldots, P^{l-1}$, we can solve the optimization problem in (2.18) over the data matrix of only one block, $P^{l-1}$. This is illustrated in Fig. 2.2. The path shown in black corresponds to the optimization in (2.18), which uses the results of the optimization for the previous blocks corresponding to the path in gray. This is the key element of dynamic programming that enables us to make the decoding process fast. Additionally, the operations in (2.18) can also be performed in parallel to make the decoding even faster. The process for decoding the signals for User 2 proceeds similarly.

**Method 2 (Non-Causal):** In Method 2, we consider some specific time blocks $k_1 < k_2 < \cdots$ and decode all the previously undecoded signals within those blocks. In other words, we decode the signals transmitted at blocks $2, 3, \cdots, k_1$ once the signals for block $k_1$ are received. Then, we decode the signals transmitted at blocks $k_1 + 1, k_1 + 2, \cdots, k_2$ once the signals for block $k_2$ are received, and so on. Since the decoded signals at some blocks using Method 2 depend on the received signals at future blocks, the decoder using Method 2 can be considered a non-causal system. Therefore, this method increases the decoding delay, but as we will see later in Section 2.5, the performance improves using this method. This is due.
to the fact that Eq. (2.6) depends on both $P^l$ and $P^{l-1}$, and thus provides some information about the data matrix of the previous block, $P^{l-1}$, as well as that of the current block, $P^l$. Therefore, all these equations at different time blocks are in some way connected. By also considering the equations for future blocks, more information is provided about the data matrix at the current block, which is to be decoded, and the performance of our decoder improves.

Using Method 2, in the $j$th stage of decoding, $j \geq 1$, we decode the data matrices at blocks $k_{j-1} + 1, \ldots, k_j$ where $k_0 = 1$. We consider the following pdf

$$
\hat{f}_2(P^2, \cdots, P^{k_j}) \propto \prod_{\ell=2}^{k_j} \exp \left\{ -(\lambda^\ell)^{-1} \| \Lambda^\ell(P^{\ell-1}, P^\ell) \|_F^2 \right\}
$$

$$
= \exp \left\{ -\sum_{\ell=2}^{k_j} (\lambda^\ell)^{-1} \| \Lambda^\ell(P^{\ell-1}, P^\ell) \|_F^2 \right\}.
$$

Then, in order to decode the data matrix for any block $l$ ($k_{j-1} < l \leq k_j$), we use DP to find the best estimate of $P^l$ that maximizes $\hat{f}_2(P^2, \cdots, P^{k_j})$ in (2.19). In order to maximize the above pdf, we only need to minimize $\sum_{\ell=2}^{k_j} (\lambda^\ell)^{-1} \| \Lambda^\ell(P^{\ell-1}, P^\ell) \|_F^2$. Therefore, for any $j \geq 1$, we define the $j$th stage of the IUIF decoding using Method 2 as

$$
\left\{ \hat{P}^{k_{j-1}+1}, \ldots, \hat{P}^{k_j} \right\} = \arg\min_{P^{k_{j-1}+1}, \ldots, P^{k_j}} \left\{ \min_{P^2, \cdots, P^{k_j-1}} \sum_{\ell=2}^{k_j} (\lambda^\ell)^{-1} \| \Lambda^\ell(P^{\ell-1}, P^\ell) \|_F^2 \right\}.
$$

Solving the optimization problem in (2.20) using exhaustive search is computationally expensive. In what follows, we will show how DP can be applied to do this efficiently and describe all the steps of the decoding process. Let us denote the minimizing arguments of $\sum_{\ell=2}^{k_j} (\lambda^\ell)^{-1} \| \Lambda^\ell(P^{\ell-1}, P^\ell) \|_F^2$ by $\hat{P}^2, \cdots, \hat{P}^{k_j}$. Then, if we know $\hat{P}^{l+1}$ ($k_{j-1} < l \leq k_j - 1$),
we can write $\hat{P}^l$ using (2.20) as

$$
\hat{P}^l = \text{argmin}_{P^l} \left\{ \min_{P^2, \ldots, P^{l-1}} \sum_{\ell=2}^{k_j} \left( \hat{\lambda}^\ell \right)^{-1} \left\| \Lambda^\ell \left( P^{\ell-1}, P^\ell \right) \right\|_F^2 \right\}
$$

$$
= \text{argmin}_{P^l} \left\{ \min_{P^2, \ldots, P^{l-1}} \left[ \sum_{\ell=2}^{l} \left( \hat{\lambda}^\ell \right)^{-1} \left\| \Lambda^\ell \left( P^{\ell-1}, P^\ell \right) \right\|_F^2 \right] + \left( \hat{\lambda}^{l+1} \right)^{-1} \left\| \Lambda^{l+1} \left( P^l, \hat{P}^{l+1} \right) \right\|_F^2
\right.
\left. + \min_{P^{l+2}, \ldots, P^{k_j}} \left[ \left( \hat{\lambda}^{l+2} \right)^{-1} \left\| \Lambda^{l+2} \left( \hat{P}^{l+1}, P^{l+2} \right) \right\|_F^2 \right] + \sum_{\ell=l+3}^{k_j} \left( \hat{\lambda}^\ell \right)^{-1} \left\| \Lambda^\ell \left( P^{\ell-1}, P^\ell \right) \right\|_F^2 \right\}.
$$

(2.21)

Since the last two lines in (2.21) are independent of $P^l$, we can remove them and write (2.21) as

$$
\hat{P}^l = \text{argmin}_{P^l} \left\{ \Phi^l \left( P^l \right) + \left( \hat{\lambda}^{l+1} \right)^{-1} \left\| \Lambda^{l+1} \left( P^l, \hat{P}^{l+1} \right) \right\|_F^2 \right\}
$$

(2.22)

Therefore, if we know $\hat{P}^{l+1}$ and $\Phi^l \left( P^l \right)$, we can compute $\hat{P}^l$ using (2.22). This is the main property that we use in decoding.

In the $j$th stage of decoding, using (2.17) and similar to Method 1, we start by computing and storing $\Phi^l \left( P^l \right), \ell = k_{j-1} + 1, \cdots, k_j$, for any possible data matrix $P^l$ using the stored values of $\Phi^l \left( P^l \right)$ for the previous block. We can compute $\Phi^l \left( P^l \right)$ once the received signals for block $\ell$ are received, just as in Method 1, and there is no additional delay. Then, note that $\hat{P}^{kj}$ is exactly the same as that of Method 1 since (2.15) and (2.19), and thus the resulting cost functions using Methods 1 and 2, are the same for decoding block $l = k_j$. So, at block $k_j$, we compute $\hat{P}^{kj} = \text{argmin}_{P^{kj}} \Phi^{kj} \left( P^{kj} \right)$ as the best estimate of the data matrix $P^{kj}$ and provide the decoded bits. Then, we move backwards and decode the remaining matrices one at a time starting from $P^{kj-1}$ and ending at $P^{kj-1+1}$ based on (2.22), i.e. using the last decoded matrix and the stored values of $\Phi^l \left( P^l \right), \ell = k_{j-1} + 1, \cdots, k_j - 1$. Then we provide the decoded bits for each time block. The block diagram of the IUIF differential decoders is shown in Fig. 2.3.
Differential Decoding Schemes That Achieve Full Diversity

In this subsection, we present additional decoding schemes based on the above methods that achieve full diversity. Letting $\tilde{Y}_l = [Y_l^{l-2}, Y_l^{l-1}, Y_l^l]$, it is easy to see from (2.1) that conditioned on the data matrices $P_l^{l-1}, P_l^l, Q_l^{l-1}, Q_l^l$, the matrix $\tilde{Y}_l$ is Gaussian with conditional pdf

$$P\left(\tilde{Y}_l \mid P_l^{l-1}, P_l^l, Q_l^{l-1}, Q_l^l\right) \propto \exp\left\{-\text{Tr}\left[\tilde{Y}_l \cdot (\Sigma_l')^{-1} \cdot (\tilde{Y}_l)^\dagger\right]\right\}$$

where $\Sigma_l'$ is the covariance matrix given by $\Sigma_l' = (U_1')^\dagger \cdot U_1' + (U_2')^\dagger \cdot U_2' + (SNR)^{-1} \cdot I_{3N}$ with $U_1' = [I_N, P_l^{l-1}, P_l^l]$ and $U_2' = [I_N, Q_l^{l-1}, Q_l^l]$. Based on (2.23), we can define the Maximum 3 Block Likelihood (M3BL) decoding using Method 0 as

$$\left\{\hat{P}_l^{l-1}, \hat{P}_l^l, \hat{Q}_l^{l-1}, \hat{Q}_l^l\right\} = \arg\min_{P_l^{l-1}, P_l^l, Q_l^{l-1}, Q_l^l} \left\{M \cdot \ln\left[\det(\Sigma_l')\right] + \text{Tr}\left[\tilde{Y}_l \cdot (\Sigma_l')^{-1} \cdot (\tilde{Y}_l)^\dagger\right]\right\}. \tag{2.24}$$

The cost function of the M3BL decoder using Method 0 is a function of $P_l^{l-1}, P_l^l, Q_l^{l-1}, Q_l^l$, while the cost function of the IUIF decoder using Method 0 is only a function of $P_l^{l-1}, P_l^l$. The DP procedures in Methods 1 and 2 can be used with the cost function of the M3BL decoder in (2.24) as well. However, instead of $\Phi^l(P_l^l)$ defined in (2.17), a function of $P_l^l, Q_l^l$ needs to be computed and stored. The two algorithms can therefore be changed accordingly. As another example of such cost functions, we can consider the cost function of the differential scheme in [21], which is also a function of $P_l^{l-1}, P_l^l, Q_l^{l-1}, Q_l^l$. Therefore, our methods can be similarly applied to the scheme in [21]. Note that all these schemes, unlike the IUIF decoding schemes, work for any square orthogonal STBC with any number of transmit or
receive antennas. In Section 2.3, we will show that the IUIF decoding schemes can also be extended to the case where the receiver has more than two receive antennas.

A variety of algorithms for decoders with cost functions like those of Methods 1 and 2 have been proposed in the literature that provide a tradeoff between performance, decoding complexity, memory requirements, and so on. For example, under reasonable noise conditions, the lazy Viterbi algorithm [36] provides a faster decoding than the Viterbi algorithm (VA) without affecting performance; however, the lazy Viterbi algorithm requires more memory compared to the VA. All such algorithms can be utilized in our schemes, and we expect to observe similar tradeoffs.

2.3 Extension to Any Number of Receive Antennas

In this section, we show that the IUIF decoding schemes can also be extended to two users with two transmit antennas and one receiver with more than two receive antennas. When there are $M > 2$ receive antennas, the same differential encoding procedure presented in Section 2.2.1 can be used. For differential decoding, we use maximum ratio combining where the received signals from all diversity branches are combined and the instantaneous SNR is maximized at the combiner output [1]. First, we consider Eqs. (2.3), (2.4), and (2.5), for $m = 1, \cdots, M$. Then, we cancel the interference of User 2 and proceed similarly to the procedure described in Appendix A. Therefore, similar to Proposition 2.1, we can obtain the following property:

Proposition 2.2. For $m = 2, \cdots, M$ and any $l \geq 2$, the following relationship holds almost surely between the received signals and the transmitted data signals of User 1

$$
0_2 = P^{l-1}P^lA_{m0}^l + P^{l-1}A_{m1}^l + A_{m2}^l + \tilde{N}_m^l \quad (2.25)
$$
where \(A_{m0}^l, A_{m1}^l, A_{m2}^l\) are functions of the received signals defined as

\[
A_{m0}^l = \left| \det(Y_i^l \Gamma_m^l) \right|^{-1} \cdot Y_i^l \Gamma_m^l, \quad A_{m1}^l = -\sum_{i=0}^1 \left| \det(Y_i^{l-1} \Gamma_m^{l-i}) \right|^{-1} \cdot Y_i^{l-1} \Gamma_m^{l-i},
\]

\[
A_{m2}^l = \left| \det(Y_1^{l-2} \Gamma_m^{l-1}) \right|^{-1} \cdot Y_1^{l-2} \Gamma_m^{l-1},
\]

where \(\Gamma_m^l = \sum_{i=0}^1 (-1)^i \left| \det(Y_1^{l-i}) \right|^{-1} \cdot Y_m^{l-i} Y_1^{l-i}, \ell \geq 1\), and \(\bar{\mathbf{N}}_m^l\) is the effective noise matrix given by

\[
\bar{\mathbf{N}}_m^l = \mathbf{C}^l \left[ \sum_{i=0}^{2} \left( \mathbf{P}_m \mathbf{N}_{m1}^{i-l} - \mathbf{P}_m \mathbf{N}_{m1}^{i-l} \right) \right] A_{mi}^l
\]

where \(\mathbf{P}_m = |\det(\mathbf{F}_m G_1)|^{-1} \cdot \mathbf{F}_m \mathbf{G}_1^\dagger\) and \(\mathbf{P}_m = |\det(\mathbf{F}_m G_m)|^{-1} \cdot \mathbf{F}_m \mathbf{G}_m^\dagger\) with \(\mathbf{F}_m = |\det(\mathbf{G}_m)|^{-1} \cdot \mathbf{G}_m^\dagger \mathbf{H}_m - |\det(\mathbf{G}_1)|^{-1} \cdot \mathbf{G}_1^\dagger \mathbf{H}_1\).

**Proof.** The proof is similar to that of Proposition 2.1 in Appendix A.

Thus, we can obtain the effective noise power at block \(l\) and receive antenna \(m = 2, \ldots, M\), \(\lambda_m^l\), when using Eq. (2.25) for decoding as

\[
\lambda_m^l = \left( |\det(\mathbf{P}_m)| + |\det(\mathbf{P}_m)| \right) \sum_{i=0}^2 \left| \det(A_{mi}^l) \right|.
\]

(2.28)

Similarly, we can also obtain approximations for \(|\det(\mathbf{P}_m)|\) and \(|\det(\mathbf{P}_m)|\) as

\[
|\det(\mathbf{P}_m)| \approx |\det(\bar{\mathbf{P}}_m)| \left( |\det(Y_m^l)|^{-1} + |\det(Y_m^{l-1})|^{-1} \right),
\]

\[
|\det(\mathbf{P}_m)| \approx |\det(\bar{\mathbf{P}}_m)| \left( |\det(Y_1^l)|^{-1} + |\det(Y_1^{l-1})|^{-1} \right).
\]

(2.29)

(2.30)

where \(\bar{\mathbf{P}}_m^l = \sum_{i=0}^1 (-1)^i |\det(Y_m^{l-i})|^{-1} \cdot Y_m^{l-i} Y_1^{l-i}\). We denote the approximation for \(\lambda_m^l\) using the above approximations by \(\hat{\lambda}_m^l\). For \(l \geq 2\), we define the UIIF decoding using Method 0 as

\[
\left\{ \hat{\mathbf{P}}_m^{l-1}, \hat{\mathbf{P}}_m^l \right\} = \arg\min_{\mathbf{P}_m^{l-1}, \mathbf{P}_m^l} \sum_{m=2}^{M} \left( \lambda_m^l \right)^{-1} \left\| A_m^l \left( \mathbf{P}_m^{l-1}, \mathbf{P}_m^l \right) \right\|_F^2
\]

(2.31)

where \(\mathbf{A}_m^l \left( \mathbf{P}_m^{l-1}, \mathbf{P}_m^l \right) = \mathbf{P}_m^{l-1} \mathbf{A}_m^l \mathbf{P}_m^{l-1} + \mathbf{P}_m^{l-1} \mathbf{A}_m^l + \mathbf{A}_m^l \). When using Method 1 for decoding the signals transmitted by User 1, we consider the following approximate joint pdf

\[
\hat{f}_1(\mathbf{P}_m^2, \ldots, \mathbf{P}_m^l) \propto \prod_{\ell=2}^{l} \exp \left\{- \sum_{m=2}^{M} \left( \lambda_m^\ell \right)^{-1} \left\| \mathbf{A}_m^\ell \left( \mathbf{P}_m^{\ell-1}, \mathbf{P}_m^\ell \right) \right\|_F^2 \right\}
\]

\[
= \exp \left\{- \sum_{\ell=2}^{l} \sum_{m=2}^{M} \left( \lambda_m^\ell \right)^{-1} \left\| \mathbf{A}_m^\ell \left( \mathbf{P}_m^{\ell-1}, \mathbf{P}_m^\ell \right) \right\|_F^2 \right\}.
\]

(2.32)
For any block \( l \geq 2 \), we define the IUIF decoding using Method 1 as

\[
\hat{P}_l = \arg\min_{P_l} \Phi_l(P_l)
\]

where

\[
\Phi_l(P_l) \triangleq \begin{cases} 
\sum_{m=2}^{M} (\hat{\lambda}_m^l)^{-1} \left\| \Lambda_m^l(P_1, P_2) \right\|^2_F, & l = 2 \\
\min_{P_2, \ldots, P_{l-1}} \sum_{\ell=2}^{M} \sum_{m=2}^{\hat{\lambda}_m^{l-1}} \left\| \Lambda_m^\ell(P_{\ell-1}, P_\ell) \right\|^2_F, & \text{otherwise.}
\end{cases}
\]

Again, for \( l > 2 \), one may show that

\[
\Phi_l(P_l) = \min_{P_l} \left\{ \Phi_{l-1}(P_{l-1}) + \sum_{m=2}^{M} (\hat{\lambda}_m^{l-1})^{-1} \left\| \Lambda_m^l(P_{l-1}, P_l) \right\|^2_F \right\}.
\]

Then we can apply DP to decode the signals transmitted by User 1. For Method 2, we consider the following pdf

\[
\hat{f}_2(P_2, \ldots, P_k) \propto \prod_{\ell=2}^{k_j} \exp \left\{ - \sum_{m=2}^{M} (\hat{\lambda}_m^{\ell-1})^{-1} \left\| \Lambda_m^\ell(P_{\ell-1}, P_\ell) \right\|^2_F \right\}
\]

\[
= \exp \left\{ - \sum_{\ell=2}^{k_j} \sum_{m=2}^{M} (\hat{\lambda}_m^{\ell-1})^{-1} \left\| \Lambda_m^\ell(P_{\ell-1}, P_\ell) \right\|^2_F \right\}.
\]

For any \( j \geq 1 \), we define the \( j \)th stage of the IUIF decoding using Method 2 as

\[
\{\hat{P}_{k_j-1+1}, \ldots, \hat{P}_{k_j}\} = \arg\min_{P_{k_j-1+1}, \ldots, P_{k_j}} \left\{ \min_{P_2, \ldots, P_{k_j}} \sum_{\ell=2}^{k_j} \sum_{m=2}^{M} (\hat{\lambda}_m^{\ell-1})^{-1} \left\| \Lambda_m^\ell(P_{\ell-1}, P_\ell) \right\|^2_F \right\}.
\]

Now, let \( \hat{P}_2, \ldots, \hat{P}_{k_j} \) denote the minimizing arguments of \( \sum_{\ell=2}^{k_j} \sum_{m=2}^{M} (\hat{\lambda}_m^{\ell-1})^{-1} \left\| \Lambda_m^\ell(P_{\ell-1}, P_\ell) \right\|^2_F \). Then, if we know \( \hat{P}_{l+1} \) \((k_{j-1} < l \leq k_j - 1)\), we can write \( \hat{P}_l \) as

\[
\hat{P}_l = \arg\min_{P_l} \left\{ \Phi_l(P_l) + \sum_{m=2}^{M} (\hat{\lambda}_m^{l+1})^{-1} \left\| \Lambda_m^{l+1}(P_l, \hat{P}_{l+1}) \right\|^2_F \right\}.
\]

Using (2.38), we can apply the same differential decoding algorithm described in Section 2.2.2 to decode the signals. Due to space limitations, we leave the extension of our differential schemes to more than two users as our future work.
2.4 Diversity Analysis

Diversity is usually defined as the exponent of SNR in the error rate expression at high SNRs

\[ d \triangleq \lim_{{\text{SNR} \to \infty}} \frac{\log P_{{\text{err}}}}{\log \text{SNR}} \]  \hspace{1cm} (2.39)

where \( P_{{\text{err}}} \) represents the probability of error. In this section, we first prove that if one of our schemes using Method 0 provides full diversity, then the corresponding schemes using Methods 1 and 2 will provide full diversity as well. Then, we analyze the diversity of the proposed schemes in Section 2.2.2.

Consider one of the proposed differential schemes using Methods 0, 1 and 2. Throughout this section we let \( E^l_i, i = 0, 1, 2 \), denote the event when an error occurs at the time of decoding the data matrix for block \( l \) using the considered scheme and Method \( i \). Note that when using Method 0, for any \( n \in \mathbb{N} \), the data matrices at blocks \( 2n - 1 \) and \( 2n \) are decoded jointly. On the other hand, when using Method 1 or 2, the data matrix at each block \( l \) is decoded by itself as a result of deploying DP. Therefore, for any block \( l \), \( E^l_0 \) denotes the event when at least one of the decoded matrices for the two consecutive blocks is in error, while \( E^l_i, i = 1, 2 \), denotes the event when the data matrix at block \( l \) is in error. Additionally, note that for any blocks \( l_1, l_2 \) we have \( P(E^{l_1}_0) = P(E^{l_2}_0) \). This is because \( E^{l_1}_0 \) and \( E^{l_2}_0 \) are the error events using Method 0, which uses a cost function based on only three consecutive time blocks. Thus, the set of possible noise values, channels, and data matrices that result in the occurrence of \( E^{l_1}_0 \) will also result in the occurrence of \( E^{l_2}_0 \), except that the values in the two cases may correspond to different time blocks and the channel matrices in the latter case may be rotated to cancel out the previously transmitted unitary matrices. Therefore, \( P(E^l_0) \) is independent of \( l \), and we will simply denote it by \( P(E_0) \). In the following lemma, we obtain an upper bound for \( P(E_i^l) \), \( i = 1, 2 \), with respect to \( P(E^l_0) \). Later, we will use this bound to analyze the diversity of our schemes.
Lemma 2.1. For any time block $l \geq 2$, we have
\[ P(E^l_1) \leq (l-1) \cdot P(E_0), \quad P(E^l_2) \leq (k_j - 1) \cdot P(E_0) \] (2.40)
where $k_j$ is the last block in the $j$th stage of decoding using Method 2 and $j$ is such that the signals for block $l$ are decoded in the $j$th stage, that is, $k_{j-1} < l \leq k_j$.

Proof. We show the first inequality (for Method 1) by induction on $l$; the proof for the second inequality (for Method 2) is similar. For $l = 2$, the cost function of Method 0 will be the same as that of Method 1. Even if in Method 0 only the first transmitted matrix for each user is arbitrary and known at both the encoder and decoder, and in Method 1 there is an additional arbitrary transmitted matrix for each user, it can be easily seen from the definitions of $E^2_1$ and $E^2_0$ that $E^2_1 \subseteq E^2_0$ and thus $P(E^2_1) \leq P(E_0)$. Now for some $l \geq 2$ assume that $P(E^l_1) \leq (l-1) \cdot P(E_0)$. We first note that $P(E^{l+1}_1)$ can be written as $P(E^{l+1}_1) = P(E^{l+1}_1, E^l_1) + P(E^{l+1}_1, \bar{E}^l_1)$. Then, we note that $P(E^{l+1}_1, E^l_1) \leq P(E^{l+1}_0, E^l_1)$ because if $E^{l+1}_1$ and $E^l_1$ both occur then $E^{l+1}_0$ must occur. This can be easily shown by observing that $E^l_1$ occurs when the data matrix at block $l$ is correctly decoded using Method 1, and thus it is the matrix that minimizes the cost function. Combining the above two facts we obtain
\[ P(E^{l+1}_1) \leq P(E^{l+1}_1, E^l_1) + P(E^{l+1}_0, \bar{E}^l_1) \leq P(E^l_1) + P(E_0) \] (2.41)
where the last inequality is obtained noting that $P(E^{l+1}_1, E^l_1) \leq P(E^l_1)$ and $P(E^{l+1}_0, \bar{E}^l_1) \leq P(E^{l+1}_0) = P(E_0)$. Then invoking the inductive assumption, we have
\[ P(E^{l+1}_1) \leq (l-1) \cdot P(E_0) + P(E_0) = l \cdot P(E_0). \] (2.42)
This ends the proof.

We will now present the main theorem that enables us to analyze the diversity of our schemes:

Theorem 2.1. If one of the proposed differential schemes using Method 0 provides full diversity, then the corresponding differential schemes using Methods 1 and 2 will provide full diversity as well.
Proof. We show the result for Method 1; the proof for Method 2 is similar. Using Lemma 2.1, the average probability of error in time blocks $2, \cdots, L$ (one frame) when using Method 1 is bounded by

$$P_{err} = \frac{1}{L-1} \sum_{\ell=2}^{L} P(E_1^\ell) \leq \frac{1}{L-1} \sum_{\ell=2}^{L} (\ell - 1) \cdot P(E_0) = \frac{L}{2} \cdot P(E_0).$$  \hfill (2.43)

Therefore, we have

$$\lim_{\text{SNR} \to \infty} \frac{\log P_{err}}{\log \text{SNR}} \leq \lim_{\text{SNR} \to \infty} \frac{\log \left( \frac{L}{2} \cdot P(E_0) \right)}{\log \text{SNR}}$$

$$= \lim_{\text{SNR} \to \infty} \frac{\log \left( \frac{L}{2} \right) + \log P(E_0)}{\log \text{SNR}}$$

$$= \lim_{\text{SNR} \to \infty} \frac{\log P(E_0)}{\log \text{SNR}}$$  \hfill (2.44)

where the last equality results from the fact that $\log \text{SNR}$ grows faster than $\log(L/2)$. Therefore, the diversity of the scheme using Method 1 must be greater than or equal to that of the scheme using Method 0. Since the scheme using Method 0 provides the highest possible diversity (full diversity), the scheme using Method 1 must provide full diversity as well. \qed

It is shown in [21] that using the cost function in [21] (Method 0) full diversity is achieved. Therefore, by Theorem 2.1 if we use our differential scheme based on Method 1 or 2 using the cost function in [21], full diversity is achieved as well. Moreover, since the M3BL decoding using Method 0 achieves the minimal probability of error among the differential decoding schemes with cost functions that use only the received signals for 3 consecutive time blocks, the M3BL decoding scheme using Method 0 must perform at least as good as the scheme in [21]. Thus, the M3BL decoding scheme using Method 0 must achieve full diversity. Then, by Theorem 2.1 the M3BL decoding using Method 1 or 2 provides full diversity as well.

### 2.5 Simulation Results

In this section, we provide simulation results for the performance of the proposed differential modulation schemes based on Methods 1 and 2 using the IUIF decoders, the M3BL decoders,
and the “exact” decoding scheme in [21]. Note that there are several decoding schemes presented in [21], and the one with the best performance is called “exact”. We compare the performance of the proposed schemes to the “exact” decoding scheme presented in [21] (using Method 0) and the coherent detection scheme for two users using Zero-Forcing (ZF) and Maximum Likelihood (ML) decoding. We also provide the simulation results for users with a single transmit antenna using the IUIF decoders and Methods 1 and 2, the M3BL decoders and Methods 0, 1 and 2, and the heuristic differential modulation scheme in [20] (using Method 0) for the purpose of comparison. In our differential schemes when using Method 2 for decoding, we decode all the signals within each frame after receiving the last signal in that frame. In our simulations, the channel is quasi-static flat Rayleigh fading where the fading is constant within one frame and varies independently from one frame to another. Every frame includes the transmission of 128 data symbols from each antenna. We use BPSK and QPSK as the unit-length constellations for the simulations of our differential schemes at transmission rates 1 b/(s Hz) and 2 b/(s Hz), respectively. For the IUIF decoding schemes, we use rotation angles of $\pi/2$ and $\pi/4$ for User 2 at transmission rates 1 b/(s Hz) and 2 b/(s Hz), respectively, and for all the other differential schemes we use the rotation angles suggested in [21]. In each figure, the curves for Users 1 and 2 are identical.

Figs. 2.4 and 2.5 show BER as a function of SNR at transmission rates 1 b/(s Hz) and 2 b/(s Hz), respectively, for 2 users each equipped with 2 transmit antennas and a receiver with 2 receive antennas. In Figs. 2.6 and 2.7 we present similar results for 3 receive antennas. All simulation results demonstrate that the IUIF decoding schemes achieve full transmit diversity like the corresponding coherent schemes using ZF. Moreover, the proposed scheme using the cost function in [21] and Method 1 or 2 and the M3BL decoding using Method 1 or 2 provide full diversity like the corresponding decoding schemes using Method 0 and the corresponding coherent schemes using ML decoding. Note that by increasing the number of receive antennas from 2 to 3, the performance loss of the IUIF decoding schemes compared to the coherent cases increases by about 1 dB in each case due to the
Figure 2.4: Performance of coherent and differential schemes at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.

Figure 2.5: Performance of coherent and differential schemes at a rate of 2 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.
Figure 2.6: Performance of coherent and differential schemes at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 3 receive antennas.

Figure 2.7: Performance of coherent and differential schemes at a rate of 2 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 3 receive antennas.
Figure 2.8: Comparison of our schemes and the heuristic differential scheme in [20] at a rate of 1 b/(s Hz) for 2 users each with 1 transmit antenna and 1 receiver with 2 receive antennas.

fact that even though we have normalized the power of the effective noises for each receive antenna, they are still correlated and thus some performance is lost. The performance loss between the IUIF differential decoding schemes and the corresponding coherent detection schemes using ZF is approximately 6-9 dB using Method 1 and about 4-7 dB using Method 2 depending on the rate and the number of receive antennas. Additionally, the performance loss between the M3BL decoding schemes and the corresponding coherent detection schemes using ML decoding is approximately 6-8 dB using Method 1 and about 4-5 dB using Method 2 depending on the rate and the number of receive antennas. Similar results are achieved for a higher number of receive antennas.

In Fig. 2.8, we compare the performance of our differential schemes using different decoding algorithms with the performance of the heuristic scheme in [20] at a transmission rate of 1 b/(s Hz). Note that using a similar argument to that of Section 2.2.2 one may easily observe that the IUIF differential schemes can also be used for a system with two users
each with a single transmit antenna by simply replacing each matrix in Eq. (2.5) with its corresponding entry in the first row and first column and replacing any \( \det(\cdot) \) with \( |\cdot|^2 \) in all the equations. The M3BL decoding schemes can easily be extended to the case of users with a single transmit antenna as well. We present the simulation results for a system with 2 users each equipped with 1 transmit antenna and a receiver with 2 receive antennas. Similar to the case of 2 transmit antennas, the IUIF decoding schemes provide full transmit diversity, while all the other differential schemes provide full diversity.

Note that in the IUIF differential schemes, one of the receive antennas is used to cancel the interference of one of the users. Therefore, similar to the case of interference cancelation in a coherent system, the additional receive antenna does not provide a higher receive diversity order as shown in Figs. 2.4-2.8. However, canceling the interference of one of the users results in a lower decoding complexity than that of the differential schemes based on Methods 0, 1 and 2 using the schemes in [20], the scheme in [21], and the M3BL decoding since the signals of both users are decoded jointly in all of them. Therefore, the IUIF differential decoders are faster than the other schemes. Additionally, our differential schemes using Methods 1 and 2 with the same decoding complexity as the differential schemes in [20] and [21] perform up to 8 dB better than them depending on the choice of the cost function. Therefore, compared to the differential decoders in [20] and [21], our schemes using Methods 1 and 2 with similar complexities to the schemes in [20] and [21] provide significant performance improvement and the possibility of a tradeoff between decoding complexity and the receive diversity.

Finally, we compare the execution time of decoding one frame (128 data symbols) in Matlab for our differential schemes using different decoders based on Methods 1 and 2 with that of the differential scheme of [21] in Table 2.1 at transmission rates 1 and 2 b/(s Hz). We have considered a system with 2 users each equipped with 2 transmit antennas and a receiver with 2 receive antennas. We have measured the execution time in seconds on a 3 GHz Core 2 Duo with 4 GB of RAM. It is evident from the table that the execution time of the IUIF schemes
Table 2.1: Execution time of decoding one frame (128 data symbols) in seconds using Matlab for 2 users with 2 transmit antennas and a receiver with 2 receive antennas

<table>
<thead>
<tr>
<th>transmission rate bits/(s Hz)</th>
<th>differential scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0379</td>
</tr>
<tr>
<td>2</td>
<td>0.3329</td>
</tr>
</tbody>
</table>

and the M3BL decoding using Methods 1 and 2 are much less than the execution time of the scheme in [21] and its corresponding decoders using Methods 1 and 2. Moreover, as the transmission rate increases, the execution time of the IUIF schemes increases much more slowly than all the other decoding schemes. Similar results are observed for the execution time of our schemes using a single transmit antenna and the scheme in [20].
Chapter 3

Asynchronous Orthogonal Differential Decoding for MAC Systems

All the existing multi-user differential schemes assume the transmission of the data by the users to be perfectly synchronized in time. To the best of our knowledge, a differential modulation scheme for asynchronous multi-user systems does not exist in the literature. In this chapter, we design differential detection schemes for asynchronous multi-user MIMO systems where neither the transmitters nor the receiver knows the channel. Our main results are as follows:

- With a slow Rayleigh fading channel model for an asynchronous multi-user system, we present a differential encoder and derive novel low complexity differential decoders by performing interference cancelation in time and employing different decoding methods. The decoding complexity of these schemes grows linearly with the number of users.

- We also present additional differential decoding schemes that perform significantly better than our low complexity decoders and outperform the existing synchronous differential schemes, but need higher decoding complexity compared to our low complexity
decoders.

- All the proposed decoders work for any square OSTBC, any constant amplitude constellation, any number of users, and any number of receive antennas.

- We analyze the diversity of our schemes and derive conditions under which the proposed schemes provide full diversity. For the cases of two and four transmit antennas, we provide examples of PSK constellations to achieve full diversity. Simulation results show that the proposed differential detection schemes provide good performance.

The rest of the chapter is organized as follows. In Section 3.1, we introduce the system model. In Section 3.2, we present the differential encoding for our asynchronous differential modulation schemes. The differential decoding schemes are put forward in Section 3.3. We analyze the diversity of our schemes in Section 3.4. Simulation results are provided in Section 3.5.

### 3.1 System Model

We consider a wireless communication system with $J$ users each with $N$ transmit antennas and one receiver with $M$ receive antennas with a quasi-static flat Rayleigh fading channel. We define $H_j$, $j = 1, \cdots, J$, as $M \times N$ channel fading matrices whose $(m,n)$th elements $h_{j,m,n}$ are the channel fading coefficients from transmit antenna $n$ to receive antenna $m$ for User $j$. The entries of $H_j$, $j = 1, \cdots, J$, are samples of independent zero-mean complex Gaussian random variables with a variance of 0.5 per real dimension.

In a practical set-up, the transmitters use pulse-shaping filters, and the receiver usually utilizes a matched filter to maximize the SNR. In such a scenario, the role of sampling is to provide a set of sufficient statistics for the detection of the received signals. Consider the
$N \times 1$ signal vector transmitted by the $j$th transmitter

$$x_j(t) = \sum_k s_j(k)\psi(t - kT_s)$$  \hspace{1cm} (3.1)$$

where $s_j(\cdot)$ is the $N \times 1$ symbol vector, $T_s$ is a symbol duration, and $\psi(\cdot)$ is the pulse-shaping filter with a non-zero duration of at most $LT_s$ for some $L \in \mathbb{N}$ (i.e., $\psi(t) = 0$, $|t| > \frac{L}{2}T_s$). We assume the average transmit power of each user is unity. The $M \times 1$ received signal vector is

$$y(t) = \sum_{j=1}^J H_j x_j(t - \tau_j) + n(t)$$

$$= \sum_{j=1}^J H_j \sum_k s_j(k)\psi(t - kT_s - \tau_j) + n(t)$$  \hspace{1cm} (3.2)$$

where $n(t)$ is the $M \times 1$ complex white Gaussian noise vector, and the symbol vectors $s_j(k)$ for the $j$th user are transmitted through the channel matrix $H_j$ and received with a relative delay of $\tau_j$. We assume $\tau_j$ is fixed within a frame. Then, considering the transmission of a frame of $D$ symbol vectors $s_j(1), \cdots, s_j(D)$ and assuming $s_j(k) = 0$ for $k \notin \{1, \cdots, D\}$, the optimum maximum-likelihood (ML) receiver uses the log-likelihood cost function given by

$$\Lambda = \int \left\| y(t) - \sum_{j=1}^J H_j \sum_{k=1}^D s_j(k)\psi(t - kT_s - \tau_j) \right\|^2_F dt$$

$$= \int \left\| y(t) \right\|^2_F dt + \int \left\| \sum_{j=1}^J H_j \sum_{k=1}^D s_j(k)\psi(t - kT_s - \tau_j) \right\|^2_F dt$$

$$- 2 \text{Re} \left\{ \text{Tr} \left[ \sum_{j=1}^J \sum_{k=1}^D \int y(t)\psi^*(t - kT_s - \tau_j) dt \cdot s_j^\dagger(k) \cdot H_j^\dagger \right] \right\}.$$  \hspace{1cm} (3.3)$$

Now, suppose that $\psi(\cdot)$, $T_s$ and $\tau_j$ are all known at the receiver and consider the RHS of the last equality in (3.3). The first integral depends only on $y(t)$, which is the same for all possible information sequences, and thus can be ignored for ML decoding. Also, for a given sequence $s_j(k)$, since all other quantities are known in coherent detection, the second integral can be calculated independent of the received signal. Finally, in terms of the received signal, it is sufficient to know only the last integral in order to perform ML decoding. Therefore, the output of the matched filter can be sampled at different sampling times associated with
different transmitters to construct $y_i(k)$ as follows

$$y_i(k) = \int_{(k-\frac{L}{2})T_s+\tau_i}^{(k+\frac{L}{2})T_s+\tau_i} y(t)\psi^*(t-kT_s-\tau_i)dt, \quad i = 1, \ldots, J, \quad k = 1, \ldots, D.$$  \hfill (3.4)

Clearly, the operations in (3.4) do not destroy any information that is valuable in deciding which symbols were transmitted, and thus these samples constitute a set of sufficient statistics for detecting all symbols. To simplify the notation, we assume that $\tau_1 = 0$, $\tau_1 < \tau_2 < \cdots < \tau_J < T_s$, and $\tau_{(i_1+i_2,J)} = \tau_{i_1} + i_2 \cdot T_s$ (\forall $i_1, i_2 \in \mathbb{Z}$). We can write each integral in (3.4) as the sum of multiple integrals on smaller intervals. Then, we can scale the resulting integrals for simplification in notation and construct a new set consisting of all these integrals to obtain another set of sufficient statistics for detection of all symbols as

$$y_i(d) = \frac{T_s}{\tau_{i+1,i}} \int_{(d-\frac{L}{2})T_s+\tau_i}^{(d+\frac{L}{2})T_s+\tau_{i+1}} y(t)\psi^*(t-dT_s-\tau_i)dt$$

$$= \sum_{j=1}^{J} H_j \sum_{r=0}^{L} s_j(d-r)\alpha_{j,i}(r) + n_i(d),$$ \hfill (3.5)

$i = 1, \ldots, J, \quad d = 1, \ldots, D + L,$

where $\tau_{i_1,i_2} = \tau_{i_1} - \tau_{i_2}, \forall i_1, i_2,$

$$n_i(d) = \frac{T_s}{\tau_{i+1,i}} \int_{(d-\frac{L}{2})T_s+\tau_i}^{(d-\frac{L}{2})T_s+\tau_{i+1}} n(t)\psi^*(t-dT_s-\tau_i)dt,$$

$$\alpha_{j,i}(r) = \frac{T_s}{\tau_{i+1,i}} \int_{(d-\frac{L}{2})T_s+\tau_i}^{(d-\frac{L}{2})T_s+\tau_{i+1}} \psi(t-(d-r)T_s-\tau_j) \cdot \psi^*(t-dT_s-\tau_i)dt$$ \hfill (3.6)

$$= \frac{T_s}{\tau_{i+1,i}} \int_{\tau_{i+1,i}-\frac{L}{2}T_s}^{\tau_{i+1,i}+\frac{L}{2}T_s} \psi(t+rT_s-\tau_{j,i})\psi^*(t)dt.$$

Note that the last element of the set, $y_J(D + L)$, is not obtained by splitting and scaling the integrals in (3.4). However, we make the notation simpler by adding it to the set, and the result is still a set of sufficient statistics. Also, notice that $\alpha_{j,i}(r) = 0$ for $r \notin \{0, \cdots, L\}$. Therefore, the index $r$ in (3.5) and (3.6) ranges from 0 to $L$. Moreover, $n_i(d)$, $\forall i, d$, are independent zero-mean complex Gaussian random vectors with covariance matrices
\[ E \left[ n_i(d) n_i^*(d) \right] = \frac{\left( \text{SNR} \right)^{-1} T_s \alpha_{i,0}(0)}{\tau_{i+1,i}} \cdot I_M \] where SNR is the ratio of the average transmit power to the noise power. Let

\[
Y(d) = (y_1(d), \ldots, y_J(d)), \quad N(d) = (n_1(d), \ldots, n_J(d)),
\]

\[ \alpha_j(r) = (\alpha_{j,1}(r), \ldots, \alpha_{j,J}(r)). \] (3.7)

Then, the received samples can be written in a matrix form as

\[
Y = \sum_{j=1}^{J} H_j S_j A_j + N
\] (3.8)

where \( Y = (Y(1), \ldots, Y(D+L)), S_j = (s_j(1), \ldots, s_j(D)), N = (N(1), \ldots, N(D+L)) \) are \( M \times (D+L)J \), \( N \times D \) and \( M \times (D+L)J \) matrices, respectively, and \( A_j \) is a \( D \times (D+L)J \) matrix given by

\[
A_j = \begin{pmatrix}
\alpha_{j,0} & \alpha_{j,1} & \ldots & \alpha_{j,L} & 0_{1 \times J} & 0_{1 \times J} & \ldots & 0_{1 \times J} & \ldots & 0_{1 \times J} \\
0_{1 \times J} & \alpha_{j,0} & \alpha_{j,1} & \ldots & \alpha_{j,L} & 0_{1 \times J} & \ldots & 0_{1 \times J} & \ldots & 0_{1 \times J} \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0_{1 \times J} & \ldots & 0_{1 \times J} & \ldots & 0_{1 \times J} & \alpha_{j,0} & \alpha_{j,1} & \ldots & \alpha_{j,L} & 0_{1 \times J} \\
0_{1 \times J} & \ldots & 0_{1 \times J} & \ldots & 0_{1 \times J} & 0_{1 \times J} & \alpha_{j,0} & \alpha_{j,1} & \ldots & \alpha_{j,L}
\end{pmatrix} \] (3.9)

For the sake of simplicity, in this chapter we consider the case where \( L = 1 \) and the pulse-shaping filter is a rectangular pulse

\[
\psi(t) = \begin{cases}
1/\sqrt{T_s} & -T_s/2 \leq t \leq T_s/2 \\
0 & \text{otherwise}
\end{cases}.
\] (3.10)

Then, it can be easily seen from (3.6) that

\[
\alpha_{j,i}(0) = \begin{cases}
1 & j \leq i \\
0 & \text{otherwise}
\end{cases}, \quad \alpha_{j,i}(1) = \begin{cases}
1 & j > i \\
0 & \text{otherwise}
\end{cases}. \] (3.11)

Therefore, in this case, using (3.7), (3.9) and (3.11), \( A_j \) becomes

\[
A_j = \begin{pmatrix}
0 \cdots 0 & 1 \cdots 1 & 0 \cdots 0 & \ldots & 0 \cdots 0 & 0 \cdots 0 \\
0 \cdots 0 & 0 \cdots 0 & 1 \cdots 1 & \ldots & 0 \cdots 0 & 0 \cdots 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 \cdots 0 & 0 \cdots 0 & 0 \cdots 0 & 1 \cdots 1 & \ldots & 0 \cdots 0 \\
0 \cdots 0 & 0 \cdots 0 & 0 \cdots 0 & \ldots & 1 \cdots 1 & \ldots & 0 \cdots 0
\end{pmatrix} \] (3.12)
and $n_i(d), \forall i, d$, become independent zero-mean complex Gaussian random vectors with covariance matrices $E[n_i(d)n_i^\dagger(d)] = \frac{(\text{SNR})^{-1}T_s}{\tau_{i+1,i}} \cdot I_M$. Note that we have not made any assumption about the values of delay differences. However, because of the scaling factor of $\frac{T_s}{\tau_{i+1,i}}$ used in (3.5), $A_j$ includes only 0s and 1s. Therefore, the values of $\tau_1, \tau_2, \cdots, \tau_J$ only appear in the noise covariance matrices in our system model.

In what follows, we consider the received signals in size $TJ$ blocks of $(y_1(Tl+1), \cdots, y_J(Tl+1), \cdots, y_1(Tl+T), \cdots, y_J(Tl+T))$, for $l = 0, 1, \cdots$, and with a small abuse of the notation, we denote them as $(y_{l,1,1}, \cdots, y_{l,j,1}, \cdots, y_{l,T,1}, \cdots, y_{l,T,j})$. Similarly, we denote the noise terms $(n_1(Tl+1), \cdots, n_J(Tl+1), \cdots, n_1(Tl+T), \cdots, n_J(Tl+T))$ as $(n_{l,1,1}, \cdots, n_{l,1,j}, \cdots, n_{l,T,1}, \cdots, n_{l,T,j})$, for $l = 0, 1, \cdots$. We define $K$ as the number of data symbols transmitted during one block. The channels are assumed to be unknown at both the transmitters and the receiver.

### 3.2 Differential Encoding

In this section, we describe our differential encoding scheme for User $j = 1, \cdots, J$. The block diagram of the differential encoder is the same as that of a synchronized system and is shown in Fig. 3.1. The main difference with the synchronous case [21], [37] is that different users do not need to employ different constellations. At a transmission rate of $b$ bits/(s Hz), we use a constant amplitude signal constellation with $2^b$ elements such as $2^b$-PSK with an
appropriate normalization to make the transmitted codewords unitary. Similar to the case of a single user, extension to other constellations is possible. For each block of $Kb$ bits, User $j$ selects $K$ symbols and transmits them using an $N \times N$ OSTBC. This transmitted codeword also depends on the codeword and symbols transmitted in the previous block. We assume the input bits are the outputs of independent uniformly distributed random variables.

The encoding starts with the transmission of arbitrary $N \times N$ OSTBCs $S_j^0$ and $S_j^1$. As in the case of a single user, we could transmit only one OSTBC instead of two and the system would still work with minor changes. For block $l$, we use the $Kb$ input bits to pick $K$ symbols $p_{j,1}^l, \cdots, p_{j,K}^l$ from the signal constellation and construct the corresponding square OSTBC, $P_j^l$. Assuming that $S_{j}^{l-1}$ is the codeword of User $j$ for the $(l-1)$th block, we calculate $S_j^l$ by

$$S_j^l = S_j^{l-1} \cdot P_j^l$$  \hspace{1cm} (3.13)

and then transmit it at block $l$. Note that the generated codeword $S_j^l$ will be orthogonal as well. Later, in Section 3.4, we analyze the diversity of the proposed schemes and derive conditions under which our schemes provide full diversity.

### 3.3 Differential Decoding

In this section, we present differential decoding schemes for all users. First, we derive novel low complexity decoders by performing interference cancelation in time and employing different decoding methods. The decoding complexity of these decoders increases linearly with the number of users. We then present additional decoding schemes that perform significantly better compared to our low complexity decoders and outperform the existing synchronous differential schemes. All the proposed decoders work for any square OSTBC, any constant amplitude constellation, any number of users, and any number of receive antennas. We
assume that the channel is unchanged within three consecutive time blocks.\footnote{As will become clear later, the channel could be assumed to be unchanged within a shorter period of time and our schemes would still work with minor changes.}

### 3.3.1 Low Complexity Decoding Schemes

In this subsection, we introduce low complexity decoders for J users with N transmit antennas through several decoding methods. First, we start with a simple example for \( J = 2 \) users and \( N = 2 \) transmit antennas to illustrate the main ideas behind our low complexity decoders. In what follows, we describe the decoding procedure for User 2. We use a subscript 2 for the quantities used in decoding the signals of User 2 to distinguish them from those of User 1.

Note that the input-output relationship in (3.8) contains the signals for the entire frame. We can rewrite (3.8) for a single time block \( l > 0 \) as

\[
\begin{pmatrix}
  y_{1,1}^l, y_{1,2}^l, y_{2,1}^l, y_{2,2}^l
\end{pmatrix} = H_1 \begin{pmatrix}
  s_{1}^{l-1}, s_{1}^l
\end{pmatrix} \begin{pmatrix}
  0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 \\
  1 & 1 & 0 & 0 \\
  0 & 0 & 1 & 1
\end{pmatrix} + H_2 \begin{pmatrix}
  s_{2}^{l-1}, s_{2}^l
\end{pmatrix} \begin{pmatrix}
  0 & 0 & 0 & 0 \\
  1 & 0 & 0 & 0 \\
  0 & 1 & 1 & 0 \\
  0 & 0 & 0 & 1
\end{pmatrix} + \begin{pmatrix}
  n_{1,1}^l, n_{1,2}^l, n_{2,1}^l, n_{2,2}^l
\end{pmatrix} \tag{3.14}
\]

Then, note that the interference of User 1 on User 2 can be canceled by subtracting \( y_{t,1}^l \) from \( y_{t,2}^l \) for \( t = 1, 2 \) as follows

\[
\begin{pmatrix}
  \bar{y}_{1,2}^l, \bar{y}_{2,2}^l
\end{pmatrix} = H_2 \begin{pmatrix}
  s_{2}^{l-1}, s_{2}^l
\end{pmatrix} \begin{pmatrix}
  0 & 0 \\
  -1 & 0 \\
  1 & -1 \\
  0 & 1
\end{pmatrix} + \begin{pmatrix}
  \bar{n}_{1,2}^l, \bar{n}_{2,2}^l
\end{pmatrix} \tag{3.15}
\]

where \( \bar{y}_{t,2}^l = y_{t,2}^l - y_{t,1}^l, \bar{n}_{t,2}^l = n_{t,2}^l - n_{t,1}^l \) for \( t = 1, 2 \). Considering (3.15) for more consecutive
time slots and using simple algebra, one may obtain

\[
\left( \bar{y}_l^{i-2}, \bar{y}_l^{i-1}, \bar{y}_l^i, \bar{y}_l^{i+1} \right) = H_2 \cdot \left( S_l^{i-2}, S_l^{i-1}, S_l^i \right) \cdot \bar{A} + \left( \bar{n}_l^{i-2}, \bar{n}_l^{i-1}, \bar{n}_l^i, \bar{n}_l^{i+1} \right) \]

\[
= H_2 S_l^{i-2} \left( I_2, P_l^{i-1}, P_l^{i-1} P_l^i \right) \bar{A} + \bar{N}_l^i \]

where

\[
\bar{A} = \begin{pmatrix}
-1 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 \\
0 & 1 & -1 & 0 \\
0 & 0 & 1 & -1 \\
0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

Now, to obtain our low complexity decoders, we note that when conditioned on \( P_l^{i-1}, P_l^i \), the matrix \( \bar{Y}_l^i \) is Gaussian with conditional probability density function (pdf)

\[
P \left( \bar{Y}_l^i \mid P_l^{i-1}, P_l^i \right) \propto \exp \left\{ -\text{Tr} \left[ \bar{Y}_l^i \cdot (\bar{V}_l^i)^{-1} \cdot (\bar{Y}_l^i)^\dagger \right] \right\} / \left[ \det(\bar{V}_l^i) \right]^M \]

where \( \bar{V}_l^i \) is the covariance matrix given by

\[
\bar{V}_l^i = (U_l^i \bar{A})^\dagger \cdot (U_l^i \bar{A}) + (\text{SNR})^{-1} T_s (\tau_{3,2}^{-1} + \tau_{2,1}^{-1}) \cdot I_5.
\]

Therefore, we can define our first low complexity decoder as

\[
\left\{ \hat{P}_l^{i-1}, \hat{P}_l^i \right\} = \arg \min_{P_l^{i-1}, P_l^i} \Lambda_2^l \left( P_l^{i-1}, P_l^i \right) \]

where \( \Lambda_2^l \left( P_l^{i-1}, P_l^i \right) \) is given by

\[
\Lambda_2^l \left( P_l^{i-1}, P_l^i \right) = M \cdot \ln \left[ \det(\bar{V}_l^i) \right] + \text{Tr} \left[ \bar{Y}_l^i \cdot (\bar{V}_l^i)^{-1} \cdot (\bar{Y}_l^i)^\dagger \right].
\]

We now consider the general case of \( J \) users with \( N \) transmit antennas and present our low complexity decoders. We illustrate the decoding process for User \( j = 1, \cdots, J \). In Method 0, we derive a low complexity decoder by canceling the interference of all users on User \( j \) and then performing ML decoding. Based on the decoder in Method 0, we then use Methods 1 and 2 presented in Chapter 2 to improve the performance. These methods use dynamic programming (DP) to efficiently decode the transmitted data signals. As we will see later,
the tradeoff for better performance of our differential schemes using Method 2 compared to that of Method 1 is the decoding delay (i.e., the number of time blocks it takes until the transmitted signals at a given time block are decoded by the receiver). Finally, using the decoder in Method 0, we present another decoding method (Method 3) to further reduce the decoding complexity while maintaining good performance.

**Method 0:** We use the following proposition to design our low complexity decoders:

**Proposition 3.1.** For any $l \geq 2$, the following relationship holds between the received signals and the transmitted signals of User $j = 1, \cdots, J$

$$\bar{Y}_j^l = H_j S_j^{l-2} U_j^l \bar{A} + \bar{N}_j^l$$

(3.21)

where $\bar{A}$ is a $3T \times 3T - 1$ matrix given by

$$\bar{A} = \begin{pmatrix}
-1 & 0 & 0 & \cdots & 0 & 0 \\
1 & -1 & 0 & \cdots & 0 & 0 \\
0 & 1 & -1 & \cdots & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & \cdots & -1 & 0 \\
0 & 0 & 0 & \cdots & 1 & -1 \\
0 & 0 & 0 & \cdots & 0 & 1 \\
\end{pmatrix}$$

(3.22)

$$\bar{Y}_j^l = \left(\bar{y}_{2,j}^{l-2}, \cdots, \bar{y}_{T,j}^{l-2}, \bar{y}_{1,j}^{l-1}, \cdots, \bar{y}_{T,j}^{l-1}, \bar{y}_{1,j}^l, \cdots, \bar{y}_{T,j}^l \right),$$

$$\bar{N}_j^l = \left(\bar{n}_{2,j}^{l-2}, \cdots, \bar{n}_{T,j}^{l-2}, \bar{n}_{1,j}^{l-1}, \cdots, \bar{n}_{T,j}^{l-1}, \bar{n}_{1,j}^l, \cdots, \bar{n}_{T,j}^l \right),$$

$$U_j^l = \left(I_N, P_j^{l-1}, P_j^{l-1} P_j^l \right),$$

where $\bar{y}_{i,j}^t = y_{i,j}^t - y_{i,j-1}^t$, $\bar{n}_{i,j}^t = n_{i,j}^t - n_{i,j-1}^t$ for $t = 1, \cdots, T$ and $\forall \ l$ (assuming that $y_{i,0}^l, n_{i,0}^l$, respectively, denote $y_{i-1,j}^1, n_{i-1,j}^1$ if $t \neq 1$, and $y_{T,j}^{l-1}, n_{T,j}^{l-1}$ if $t = 1$).

**Proof.** See Appendix A.

Equation (3.21) is the main property used to design our low complexity differential decoding algorithm, where the interference of all users on User $j$ is completely canceled. Therefore, it can be utilized to decode the transmitted signals without interference. Notice that $\bar{Y}_j^l$ starts from $\bar{y}_{2,j}^{l-2}$ instead of $\bar{y}_{1,j}^{l-2}$. We could consider using $\bar{y}_{1,j}^{l-2}$ and other previously received
signals to improve the performance of our scheme. However, that would cause additional inter-block interference from the previously transmitted signals of User $j$, which would then increase the decoding complexity. It is easy to see from (3.21) that when conditioned on $P_{j}^{l-1}, P_{j}^{l}$, the matrix $\bar{Y}_{j}^{l}$ is Gaussian with conditional pdf

$$P\left(\bar{Y}_{j}^{l} \mid P_{j}^{l-1}, P_{j}^{l}\right) \propto \exp\left\{-\text{Tr}\left[\bar{Y}_{j}^{l} \cdot (\bar{V}_{j}^{l})^{-1} \cdot (\bar{Y}_{j}^{l})^\dagger\right]\right\} \left[\det(\bar{V}_{j}^{l})\right]^{M/2}. \tag{3.23}\right.$$ 

where $\bar{V}_{j}^{l}$ is the covariance matrix given by $\bar{V}_{j}^{l} = (U_{j}^{l} \tilde{A})^\dagger \cdot (U_{j}^{l} \tilde{A}) + (\text{SNR})^{-1} T_{s} (\tau_{j+1, j}^{-1} + \tau_{j, j-1}^{-1}) \cdot I_{3T-1}$. We are now prepared to present our first low complexity differential decoding scheme. One approach is to decode $P_{j}^{l-1}$ and $P_{j}^{l}$ jointly based on (3.23). Therefore, we define the Inter-Time Interference Cancelation (ITIC) decoding using Method 0 as

$$\{\hat{P}_{j}^{l-1}, \hat{P}_{j}^{l}\} = \arg\min_{P_{j}^{l-1}, P_{j}^{l}} \Lambda_{j}^{l} \left(P_{j}^{l-1}, P_{j}^{l}\right) \tag{3.24}$$

where $\Lambda_{j}^{l} \left(P_{j}^{l-1}, P_{j}^{l}\right)$ is given by

$$\Lambda_{j}^{l} \left(P_{j}^{l-1}, P_{j}^{l}\right) = M \cdot \ln \left[\det(\bar{V}_{j}^{l})\right] + \text{Tr}\left[\bar{Y}_{j}^{l} \cdot (\bar{V}_{j}^{l})^{-1} \cdot (\bar{Y}_{j}^{l})^\dagger\right]. \tag{3.25}$$

Notice that for $l = 2$ in $U_{j}^{l}$, $P_{j}^{1} = (S_{j}^{l})^\dagger \cdot S_{j}^{1}$ is the arbitrary data matrix at block 1 and is known at both the encoder and decoder. When using this scheme, information provided by (3.23) at time blocks other than $l$ is ignored, and thus some performance is lost. To avoid such losses, we also propose additional decoding schemes using Methods 1 and 2 presented in Chapter 2 to efficiently decode the signals transmitted by the users. Note that we use the cost function of the ITIC decoder using Method 0 as described above, and thus the corresponding decoders using Methods 1 and 2 as presented in this chapter are different from the decoders presented in Chapter 2. In what follows, we summarize the description of the ITIC decoders using Methods 1 and 2 based on the cost function of the ITIC decoder using Method 0. We refer the interested reader to Chapter 2 for the details on derivations.

**Method 1 (Causal DP):** In Method 1, we decode $P_{j}^{l}$ based on (3.23) for all blocks $\ell = 2, \cdots, l$ together. We utilize DP to efficiently find the best possible data matrix that maximizes an
approximation for the conditional pdf of $\tilde{Y}_j^2, \ldots, \tilde{Y}_j^l$ given the data matrices $P_j^2, \ldots, P_j^l$. Using (3.23) and ignoring the correlations of $\tilde{Y}_j^\ell$ at different blocks $\ell = 2, \ldots, l$ given the data matrices, we consider the following:

$$f_1(P_j^2, \ldots, P_j^l) \propto \prod_{\ell=2}^l \exp \left\{ -\Lambda_j^\ell(P_j^{\ell-1}, P_j^\ell) \right\}$$

(3.26)

In order to maximize the above function, we only need to minimize $\sum_{\ell=2}^l \Lambda_j^\ell(P_j^{\ell-1}, P_j^\ell)$. For any block $l \geq 2$, we define the ITIC decoding using Method 1 as

$$\hat{P}_j^l = \arg\min_{P_j^l} \Phi_j^l(P_j^l)$$

(3.27)

where $\Phi_j^l(P_j^l)$ is defined as

$$\Phi_j^l(P_j^l) \triangleq \begin{cases} 
\Lambda_j^2(P_j^1, P_j^2), & l = 2 \\
\min_{P_j^2, \ldots, P_j^{l-1}} \sum_{\ell=2}^l \Lambda_j^\ell(P_j^{\ell-1}, P_j^\ell), & \text{otherwise}
\end{cases}$$

(3.28)

The optimization problem in (3.28) can be efficiently solved by utilizing DP. Using (3.28), it is easy to show that for $l > 2$, we have

$$\Phi_j^l(P_j^l) = \min_{P_j^{l-1}} \left\{ \Phi_j^{l-1}(P_j^{l-1}) + \Lambda_j^l(P_j^{l-1}, P_j^l) \right\}.$$  

(3.29)

As a result of storing the cost function of the previous block, $\Phi_j^{l-1}(P_j^{l-1})$, we only need to perform an optimization over $P_j^{l-1}$ for each possible data matrix $P_j^l$ at time block $l$. That is, for each possible data matrix $P_j^l$, in lieu of solving the optimization problem in (3.28) over all data matrices for the previous blocks, $P_j^2, \ldots, P_j^{l-1}$, we can solve the optimization problem in (3.29) over the data matrix of only one block, $P_j^{l-1}$, as illustrated in Fig. 3.2. The optimization in (3.29) corresponds to the black path, while the optimization for the previous blocks corresponds to the gray path.

**Method 2 (Non-Causal DP):** In Method 2, we consider some non-overlapping windows of blocks and decode the transmitted symbols within each window together. Note that since
the decoding of each block may depend on future blocks in the same window, this method will cause some additional delay. However, since more information is used, the performance will improve as well.

Using Method 2, in the $m$th stage of decoding, $m \geq 1$, we decode the data matrices at blocks $k_{m-1} + 1, \ldots, k_m$ where $k_0 = 1$ and $k_0 < k_1 < k_2 < \cdots$. We consider the following:

\[
\begin{align*}
\hat{f}_2(P^2_j, \ldots, P^{k_m}_j) &\propto \prod_{\ell=2}^{k_m} \exp\left\{-\Lambda^\ell_j(P^{\ell-1}_j, P^\ell_j)\right\} \\
&= \exp\left\{-\sum_{\ell=2}^{k_m} \Lambda^\ell_j(P^{\ell-1}_j, P^\ell_j)\right\}.
\end{align*}
\]

Then, in order to decode the data matrix for any block $l$ ($k_{m-1} < l \leq k_m$), we use DP to find the best estimate of $P^l_j$ that maximizes $f_2(P^2_j, \ldots, P^{k_m}_j)$ in (3.30). In order to maximize the above function, we only need to minimize $\sum_{\ell=2}^{k_m} \Lambda^\ell_j(P^{\ell-1}_j, P^\ell_j)$. Therefore, for any $m \geq 1$, we define the $m$th stage of the ITIC decoding using Method 2 as

\[
\begin{align*}
\{(\hat{P}^l_j)^{k_{m-1}+1}, \ldots, (\hat{P}^l_j)^{k_m}\} &= \operatorname*{arg\,min}_{P^l_j^{k_{m-1}+1}, \ldots, P^l_j^{k_m}} \left\{ \min_{P^l_j^{k_{m-1}+1}, \ldots, P^l_j^{k_{m-1}}} \sum_{\ell=2}^{k_m} \Lambda^\ell_j(P^{\ell-1}_j, P^\ell_j) \right\}.
\end{align*}
\]

To reduce the complexity of the exhaustive search in (3.31), we use DP as described below. Let us denote the minimizing arguments of $\sum_{\ell=2}^{k_m} \Lambda^\ell_j(P^{\ell-1}_j, P^\ell_j)$ by $\hat{P}^2_j, \ldots, \hat{P}^{k_m}_j$. If we know $\hat{P}^{l+1}_j$ ($k_{m-1} < l \leq k_m - 1$), it can be easily shown that $\hat{P}^l_j$ can be written as

\[
\hat{P}^l_j = \operatorname*{arg\,min}_{P^l_j} \left\{ \Phi^l_j(P^l_j) + \Lambda^{l+1}_j(P^l_j, \hat{P}^{l+1}_j) \right\}.
\]
Therefore, if we know $\hat{P}_{l+1}^j$ and $\Phi_{l}^j(P_{l}^j)$, we can compute $\hat{P}_{l}^j$ using (3.32). This is the key element of our low complexity decoder using Method 2.

In the $m$th stage of decoding, similar to Method 1, we begin by employing (3.28) and (3.29) to compute and store $\Phi_{\ell}^j(P_{\ell}^j)$, $\ell = k_m - 1, \cdots, k_m$, for any possible data matrix $P_{\ell}^j$ using the stored values of $\Phi_{\ell}^j(P_{\ell}^j)$ from the previous block. As in Method 1, once the signals for block $\ell$ are received, we can compute $\Phi_{\ell}^j(P_{\ell}^j)$ with no additional delay. Note that $\hat{P}_{j}^{k_m}$ is then exactly the same as in Method 1 because (3.26) and (3.30) (and therefore the resulting cost functions) are identical for decoding block $l = k_m$. Thus, at block $k_m$, we compute $\hat{P}_{j}^{k_m} = \text{argmin}_{P_{j}^{k_m}} \Phi_{j}^{k_m}(P_{j}^{k_m})$ as the best estimate of the data matrix $P_{j}^{k_m}$, which then determines the decoded bits. We then move backwards, decoding the remaining matrices one at a time beginning from $P_{j}^{k_m-1}$ and ending at $P_{j}^{k_m+1}$ using (3.32), that is, utilizing the last decoded matrix and the stored values of $\Phi_{j}^\ell(P_{j}^\ell)$, $\ell = k_m - 1, \cdots, k_m - 1$. Finally, we supply the decoded bits for each time block.

**Method 3 (Decision Feedback):** An alternative approach to decoding $P_{l}^j$ at block $l$ is to use the decoded matrix for $P_{l}^{l-1}$ at block $l - 1$ in (3.24). Therefore, we define the ITIC decoding using Method 3 as

$$\hat{P}_{j}^l = \text{argmin}_{P_{j}^l} A_{j}^l(\hat{P}_{j}^{l-1}, P_{j}^l)$$  \hspace{1cm} (3.33)

where $\hat{P}_{j}^{l-1}$ is the decoded matrix for $P_{j}^{l-1}$ at block $l - 1$. Notice that by using this approach, in order to decode $P_{j}^l$ we only need to solve an optimization over $P_{j}^l$. Therefore, the decoding complexity is significantly reduced compared to the previous three decoding methods. However, the decoded signals for $P_{j}^{l-1}$ at block $l - 1$ may be erroneous, which can lead to error propagation and thus performance degradation. We study the effect of error propagation in Section 3.5 and show that it is not significant.
3.3.2 Optimal Multiple Partition Decoding Schemes

In this subsection, we present additional decoding schemes that achieve significantly higher coding gains compared to our low complexity schemes. In order to do this, we need the following proposition:

**Proposition 3.2.** For any \( l \geq 2 \), the following relationship holds

\[
\tilde{Y}^l = \sum_{j=1}^{J} H_j S_j^{l-2} U_j^l \tilde{A}_j + \tilde{N}^l
\]  
(3.34)

where \( \tilde{A}_j \) is a \( 3T \times 3T(J - J + 1) \) matrix given by

\[
\tilde{A}_j = \begin{pmatrix}
1 \cdots 0 \cdots 0 \cdots 0 \cdots 0 \\
0 \cdots 1 \cdots 0 \cdots 0 \cdots 0 \\
0 \cdots 0 \cdots 1 \cdots 0 \cdots 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 \cdots 0 \cdots 0 \cdots 1 \cdots 0 \\
\end{pmatrix}
\]  
(3.35)

\[
\tilde{Y}^l = \left( y_{1,l}^l, y_{2,l}^l, y_{3,l}^l, \ldots, y_{1,l}^{l-1}, y_{2,l}^{l-1}, y_{3,l}^{l-1}, \ldots, y_{T,l}^{l-1}, y_{1,l}, y_{2,l}, \ldots, y_{T,l} \right),
\]

\[
\tilde{N}^l = \left( n_{1,l}^l, n_{2,l}^l, n_{3,l}^l, \ldots, n_{1,l}^{l-1}, n_{2,l}^{l-1}, n_{3,l}^{l-1}, \ldots, n_{T,l}^{l-1}, n_{1,l}, n_{2,l}, \ldots, n_{T,l} \right),
\]

\[
U_j^l = \left( I_N, P_{j-1}^l, P_j^{l-1} P_j^l \right), \quad j = 1, \ldots, J.
\]

**Proof.** The result follows from the input-output relationship for any time block \( l > 0 \) available in Appendix A and using simple algebra. \( \square \)

Again, notice that \( \tilde{Y}^l \) starts from \( y_{l-2}^{l-2} \) instead of \( y_{l-1}^{l-2} \). Other previously received signals could be considered to improve performance, but that would cause additional inter-block interference from previously transmitted signals and would increase decoding complexity. It is easy to see from Proposition 3.2 that when conditioned on the data matrices \( P_1^{l-1}, P_1^l, \ldots, P_j^{l-1}, P_j^l \), the matrix \( \tilde{Y}^l \) is Gaussian with conditional pdf

\[
P \left( \tilde{Y}^l \mid P_1^{l-1}, P_1^l, \ldots, P_j^{l-1}, P_j^l \right) \propto \exp \left\{ -\operatorname{Tr} \left[ \tilde{Y}^l \cdot (\tilde{V}^l)^{-1} \cdot (\tilde{Y}^l)^\dagger \right] \right\} / \left[ \det(\tilde{V}^l) \right]^{M/2}
\]  
(3.36)
where $\tilde{V}^l$ is the covariance matrix given by $\tilde{V}^l = \sum_{j=1}^J (U^l_j \tilde{A}_j)^\dagger \cdot (U^l_j \tilde{A}_j) + (\text{SNR})^{-1} T_s \cdot \tilde{D}$ and $\tilde{D} = \text{diag}(\tau^{-1}_{1,0}, \tau^{-1}_{2,1}, \cdots, \tau^{-1}_{3TJ-J-1,3TJ-J})$ is a $3TJ-J+1 \times 3TJ-J+1$ diagonal matrix. Based on (3.36), we can define the Maximum Multiple Partition Likelihood (MMPL) decoding using Method 0 as

\[
\begin{align*}
\{ \hat{P}_{l-1}^1, \hat{P}_1^1, \cdots, \hat{P}_j^1, \cdots, \hat{P}_{l-1}^J, \hat{P}_j^J \} &= \arg\min_{P_{l-1}^1, P_1^1, \cdots, P_j^1, \cdots, P_{J-1}^1, P_j^J} \left\{ M \cdot \ln \left[ \det(\tilde{V}^l) \right] + \text{Tr} \left[ \tilde{Y}^l \cdot (\tilde{V}^l)^{-1} \cdot (\tilde{Y}^l)^\dagger \right] \right\}.
\end{align*}
\]

(3.37)

The cost function of the MMPL decoder using Method 0 is a function of $P_{l-1}^1, P_1^1, \cdots, P_j^1, \cdots, P_{J-1}^1, P_j^J$, whereas the cost function of the ITIC decoder for User $j = 1, \cdots, J$ using Method 0 is only a function of $P_j^{l-1}, P_j^l$. We can use the DP procedures in Methods 1 and 2 with the cost function of the MMPL decoder in (3.37) just as with the cost function of the ITIC decoder in (3.24). However, we need to compute and store a function of $P_1^l, \cdots, P_j^l$ instead of $\Phi_j^l(P_j^l)$ defined in (3.28). Similarly, Method 3 can be applied to the cost function of the MMPL decoder in (3.37) by using the decoded matrices for $P_{l-1}^1, \cdots, P_{l-1}^J$ at block $l-1$ in (3.37) to decode $P_1^l, \cdots, P_j^l$ at block $l$. The three algorithms can therefore be changed accordingly. The block diagram of the proposed differential decoders is shown in Fig. 3.3.

The corresponding coherent decoders for the ITIC and MMPL decoders can be derived using similar procedures to the ones described above as well. Due to space limitations, we do not provide the details of the coherent ITIC and MMPL decoders.
3.4 Diversity Analysis

With a small abuse of the notation, for data matrices $P_1, P_2, P_3, P_4$, let us define

$$G(P_1, P_2, P_3, P_4) \triangleq \begin{pmatrix} I_N & P_1 P_2 \\ I_N & P_3 P_4 \end{pmatrix} \cdot \bar{A}$$

(3.38)

where $\bar{A}$ is the $3T \times 3T - 1$ matrix given in (3.22). Suppose that we choose the signal constellation such that for any possible data matrices $P_1, P_2, P_3, P_4$ with $(P_1, P_2) \neq (P_3, P_4)$, the matrix $G(P_1, P_2, P_3, P_4)$ has full row rank (i.e., $G(P_1, P_2, P_3, P_4)$ is of rank $2N$). We prove that under this condition all the proposed schemes achieve a diversity order of $MN$ (full diversity). We also derive an equivalent condition, which can be easily verified using simple matrix operations. Furthermore, for the cases of two and four transmit antennas, we provide examples of PSK constellations to achieve full diversity.

**Theorem 3.1.** The proposed ITIC and MMPL decoders using Method 0 achieve full diversity.

**Proof.** See Appendix B.

The following theorem extends the result of Theorem 3.1 to all the proposed methods:

**Theorem 3.2.** If one of the proposed differential schemes using Method 0 provides full diversity, then the corresponding differential schemes using Methods 1, 2 and 3 will provide full diversity as well.

**Proof.** The proof is very similar to that of Theorem 2.1 in Chapter 2.

Therefore, by Theorems 3.1 and 3.2, all the proposed differential schemes (i.e., ITIC and MMPL decoders using Methods 0, 1, 2 and 3) provide full diversity.

As mentioned above, in order to guarantee full diversity, we need to make sure that $G(P_1, P_2, P_3, P_4)$ has full row rank for any possible data matrices $P_1, P_2, P_3, P_4$ with $(P_1, P_2) \neq (P_3, P_4)$.
In the following theorem we derive an equivalent condition, which can be easily verified using simple matrix operations:

**Theorem 3.3.** \(G(P_1, P_2, P_3, P_4)\) has full row rank for any possible data matrices \(P_1, P_2, P_3, P_4\) with \((P_1, P_2) \neq (P_3, P_4)\) if and only if

\[
\begin{align*}
& w \cdot \left[ P_1 \bar{P}_2 + N \left( I_N - ar{P}_1 \right) \cdot \left( \frac{(\bar{P}_3 - \bar{P}_1)^\dagger}{\|\bar{P}_3 - P_1\|^2_F} \right) \cdot \left( \bar{P}_3 \bar{P}_4 - P_1 \bar{P}_2 \right) \right] \neq w \\
& \text{for any possible data matrices } \bar{P}_1, \bar{P}_2, \bar{P}_3, \bar{P}_4 \text{ with } \bar{P}_1 \neq \bar{P}_3,
\end{align*}
\]

\(w = (1, 1, \cdots, 1)\).

**Proof.** See Appendix C.

For instance, consider the case when the Alamouti code is used to construct the data matrices \(P^t_j\). Then, one can use Theorem 3.3 to verify that when the BPSK constellation \(\left\{ \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right\} \) or the QPSK constellation \(\left\{ \frac{\sqrt{2}}{2}, j \left( \frac{\sqrt{2}}{2} \right), -\frac{\sqrt{2}}{2}, -j \left( \frac{\sqrt{2}}{2} \right) \right\} \) is used, \(G(P_1, P_2, P_3, P_4)\) will have full row rank for any possible data matrices \(P_1, P_2, P_3, P_4\) with \(P_1, P_2 \neq P_3, P_4\). As another example, consider the case when the following 4 × 4 rate-one STBC [1] is used to construct the data matrices:

\[
P^t_j = \begin{pmatrix}
  p^t_{j,1} & -p^t_{j,2} & -p^t_{j,3} & p^t_{j,4} \\
  p^t_{j,2} & p^t_{j,1} & p^t_{j,4} & -p^t_{j,3} \\
  p^t_{j,3} & -p^t_{j,4} & p^t_{j,1} & p^t_{j,2} \\
  p^t_{j,4} & p^t_{j,3} & -p^t_{j,2} & p^t_{j,1}
\end{pmatrix}.
\]

Note that the above STBC is orthogonal for the BPSK constellation \(\left\{ \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right\} \). Again, one may use Theorem 3.3 to verify that when the BPSK constellation \(\left\{ \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right\} \) is used, \(G(P_1, P_2, P_3, P_4)\) will have full row rank for any possible data matrices \(P_1, P_2, P_3, P_4\) with \(P_1, P_2 \neq P_3, P_4\).
3.5 Simulation Results

In this section, we provide simulation results for the performance of the proposed differential modulation schemes using the ITIC and MMPL decoders based on Methods 1, 2 and 3. We compare the performance of our schemes to the IUIF and M3BL differential schemes presented in Chapter 2 and the synchronous coherent schemes using Zero-Forcing (ZF) and ML decoding. When using Method 2 for decoding, we decode all the signals within each frame after receiving the last signal in that frame. In our simulations, the channel is quasi-static flat Rayleigh fading where the fading is constant within one frame and varies independently from one frame to another. Depending on the number of transmit antennas, we use either the Alamouti code or the $4 \times 4$ OSTBC in (3.40) for all users to encode and transmit 64 data matrices per user in each frame. Also, we use the BPSK and QPSK constellations described in Section 3.4 as the signal constellations for the simulations of our differential schemes at transmission rates $1 \text{ b/(s Hz)}$ and $2 \text{ b/(s Hz)}$, respectively. In Figs. 3.4-3.9, we consider the relative time delays between the received signals of consecutive users to be equal (i.e., $\tau_{j+1} - \tau_j = T_s / J$, $\forall j$). We study the effect of other relative time delays on performance in Fig. 3.10. In each figure, the curves for all users are identical.

Figs. 3.4 and 3.5 show BER as a function of SNR at transmission rates $1 \text{ b/(s Hz)}$ and $2 \text{ b/(s Hz)}$, respectively, for 2 users each equipped with 2 transmit antennas and a receiver with 2 receive antennas. In Figs. 3.6 and 3.7, we present similar results for 3 receive antennas. In Fig. 3.8, we provide simulation results at a transmission rate of $1 \text{ b/(s Hz)}$ for 2 users each equipped with 4 transmit antennas and a receiver with 1 receive antenna. Note that all our schemes work for any number of receive antennas, while the low complexity differential schemes in Chapter 2 require at least $J$ receive antennas. All simulation results demonstrate that all the proposed schemes achieve full diversity like the corresponding coherent schemes using ML decoding. On the other hand, the low complexity differential schemes in Chapter 2 only provide full transmit diversity. Additionally, compared to the differential schemes
Figure 3.4: Performance of the proposed asynchronous differential schemes for $\tau_2 - \tau_1 = T_s/2$, the synchronous differential schemes in Chapter 2, and the synchronous coherent schemes using ZF and ML decoding at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.

Figure 3.5: Performance of the proposed asynchronous differential schemes for $\tau_2 - \tau_1 = T_s/2$, the synchronous differential schemes in Chapter 2, and the synchronous coherent schemes using ZF and ML decoding at a rate of 2 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.
Figure 3.6: Performance of the proposed asynchronous differential schemes for $\tau_2 - \tau_1 = T_s/2$, the synchronous differential schemes in Chapter 2, and the synchronous coherent schemes using ZF and ML decoding at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 3 receive antennas.

Figure 3.7: Performance of the proposed asynchronous differential schemes for $\tau_2 - \tau_1 = T_s/2$, the synchronous differential schemes in Chapter 2, and the synchronous coherent schemes using ZF and ML decoding at a rate of 2 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 3 receive antennas.
in Chapter 2, the MMPL decoding schemes provide significant performance improvement. Therefore, the proposed schemes provide the possibility of a tradeoff between decoding complexity and the coding gain.

In Fig. 3.9, we show BER as a function of SNR at a transmission rate of 1 b/(s Hz) for 3 users each equipped with 2 transmit antennas and a receiver with 2 receive antennas. With the assumption of equal relative time delays, it can be seen from Proposition 3.1 and the covariance matrices for the noise vectors given in Section 3.1 that the effect of changing the number of users from $J_1$ to $J_2$ on the performance of the ITIC decoders is the same as that of multiplying the SNR by $J_1/J_2$. This corresponds to a change of $10 \log_{10}(J_1/J_2)$ dB in performance. As expected, the performances of the ITIC decoders in Fig. 3.4 for 2 users are $10 \log_{10}(3/2) \approx 1.8$ dB better than those of Fig. 3.9 for 3 users. All simulations show that the effect of error propagation on the performance of the proposed schemes using Method 3 is very small. Our schemes using Method 3 have lower decoding complexity compared
Figure 3.9: Performance of the proposed asynchronous differential schemes for $\tau_{j+1} - \tau_j = T_s/3$, $\forall j$, and the synchronous coherent scheme using ML decoding at a rate of 1 b/(s Hz) for 3 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.

Figure 3.10: Comparison of the proposed asynchronous differential schemes using Method 3 for different relative time delays $\Delta \tau = \tau_2 - \tau_1$ at a rate of 1 b/(s Hz) for 2 users each with 2 transmit antennas and 1 receiver with 2 receive antennas.
to their corresponding schemes using Method 1, yet the proposed schemes using Method 3 provide almost the same performance as their corresponding schemes using Method 1.

Finally, we compare the performance of our differential schemes with different relative time delays between the received signals. Again, we consider a system with 2 users each equipped with 2 transmit antennas and a receiver with 2 receive antennas. Fig. 3.10 shows the performance of the ITIC and MMPL decoders using Method 3 for different values of $\Delta \tau = \tau_2 - \tau_1$ at a transmission rate of $1 \text{ b/(s Hz)}$. The results for our decoding schemes using Methods 0, 1 and 2 are similar. It is evident from the simulations that the proposed schemes perform best when $\Delta \tau = T_s/2$, that is, when the signals of the two users are received with a time difference of half a symbol. Moreover, for values of $\Delta \tau$ close to $T_s/2$, the performance of our schemes is close to the best performance for $\Delta \tau = T_s/2$ and deviates from the best performance more quickly as $\Delta \tau$ deviates from $T_s/2$. This is in line with capacity results reported in [38] where $\Delta \tau = T_s/2$ provides the highest value of channel capacity in a two-user MAC.
Chapter 4

Distributed Beamforming Using Feedback Control Based on TSVQ

As mentioned in Chapter 1, prior literature offers numerous different approaches to distributed phase synchronization. In order to provide a boost to network performance in terms of delay, latency, and energy efficiency, further reduction in the time for phase synchronization is required. Our main results are as follows:

- We put forth novel DBF algorithms using feedback control based on TSVQ with static channels:
  - We first propose a simple TSVQ-based DBF algorithm for the case of \( N = 2 \) transmitters.
  - We then present a TSVQ-based DBF algorithm that works for a larger number of transmitters.

- The proposed TSVQ-based DBF algorithms can feed back at any rate. To the best of our knowledge, they are the first deterministic DBF methods that can feed back more
than 1 bit per time slot for faster phase synchronization.

- We analyze the behavior of our TSVQ-based DBF algorithms:
  - We analytically show that the phase asynchrony in our TSVQ-based DBF algorithm for $N = 2$ transmitters converges to zero with probability 1.
  - We also prove that the probability that phase synchronization is attained in the other proposed TSVQ-based DBF algorithm goes to 1 as the number of transmitters goes to infinity.

- We modify our TSVQ-based DBF algorithms to enable them to track time-varying channels without the knowledge of channel state information (CSI). Unlike the existing adaptive DBF algorithms for time-varying channels, our methods have no continuous parameters to be optimized.

- Simulation results demonstrate that our algorithms significantly outperform the existing adaptive DBF algorithms for static and time-varying channels.

The rest of the chapter is organized as follows. In Section 4.1, we introduce the system model. In Section 4.2, we design and analyze the TSVQ-based DBF algorithms for static channels. The proposed TSVQ-based DBF algorithms are extended to time-varying channels in Section 4.3, and simulation results are provided in Section 4.4.

### 4.1 System Model

We consider a system with $N$ transmitting nodes that collaboratively transmit a common message to a receiver. The transmissions of the transmitters are assumed to be perfectly synchronized in time and frequency. Our goal is to adjust the phase offsets of the transmitters to achieve phase coherence at the receiver. For $n = 1, \cdots, N$, we denote the channel
coefficient from Node \( n \) to the receiver at time slot \( t \) as \( h_n[t] = a_ne^{j\psi_n[t]} \) where \( a_n \) and \( \psi_n[t] \) represent the attenuation and phase response of the channel from Node \( n \) to the receiver at time slot \( t \). Now, let \( \gamma_n \) and \( \phi_n[t] \) denote the unknown phase offset at Node \( n \) and the phase component adjusted by Node \( n \) at time \( t \), respectively. \( \psi_n[t], \gamma_n, n = 1, \cdots, N \), are assumed to be independent random variables with a uniform distribution over \((-\pi, \pi] \). We define the RSS at time \( t \) as
\[
\text{RSS}_N[t] \triangleq \left| \sum_{n=1}^{N} a_ne^{j\theta_n[t]} \right| \tag{4.1}
\]
where \( \theta_n[t] = \psi_n[t] + \gamma_n + \phi_n[t] \). Therefore, at any time slot \( t \), any Node \( n \) is associated with a particular vector \( a_n e^{j\theta_n[t]} \). Note that the RSS is maximized if and only if \( \theta_1[t] = \cdots = \theta_N[t] \).

We denote this maximum value by \( \text{RSS}_{N,\text{max}} \) defined as
\[
\text{RSS}_{N,\text{max}} \triangleq \sum_{n=1}^{N} a_n. \tag{4.2}
\]

Therefore, our objective is to maximize the RSS by adjusting the phase components \( \theta_n[t] \), \( n = 1, \cdots, N \) to achieve phase coherence. This adjustment is performed by adapting \( \phi_n[t] \) for \( n = 1, \cdots, N \) based on up to \( B \) bits of feedback from the receiver at each time slot. We denote the ratio of the RSS to its maximum possible value by \( \rho_N \) defined as
\[
\rho_N[t] \triangleq \frac{\text{RSS}_N[t]}{\text{RSS}_{N,\text{max}}}. \tag{4.3}
\]

Since the proposed algorithms for static channels can be divided into several stages consisting of multiple time slots, for the sake of simplicity, we use \( \text{RSS}_N\{K\} \) and \( \rho_N\{K\} \) to refer to the RSS and the ratio of the RSS to its maximum possible value, respectively, at the last time slot of Stage \( K \) of the algorithms. Also, in some parts of the chapter, we suppress the subscript \( N \) in \( \text{RSS}_N[t] \) when there is no confusion.

In the static scenario, the channel coefficients, \( h_n[t] \), are fixed at different time slots. Therefore, the phase response of the channel, \( \psi_n[t] \), remains unchanged over time and does not de-
pend on time. However, in a time-varying scenario, channel fluctuations are time-dependent causing the channel phase responses to change over time. Therefore, to avoid performance deterioration, these changes need to be tracked as well. For the time-varying case, we use the channel drift model described in [24]. In [24], the channel phase response for Node $n$ at time slot $t$ is modeled as

$$\psi_n[t] = \psi_n[t - 1] + \Delta[t]$$

(4.4)

where the drift process $\Delta[t]$ is independent and identically distributed (i.i.d.) across sensors, and stationary and uncorrelated in time with a uniform distribution in $(-\Delta_{\text{max}}, \Delta_{\text{max}}]$ where $\Delta_{\text{max}}$ is the parameter that determines the maximum and minimum possible values of the drift process.

In what follows, $\theta = (\theta_1, \cdots, \theta_N)$ denotes the last updated phase vector, $\theta_{TX} = (\theta_{TX,1}, \cdots, \theta_{TX,N})$ denotes the phase vector at the transmission time, and $\theta_{\text{new}} = (\theta_{\text{new},1}, \cdots, \theta_{\text{new},N})$ denotes the new phase vector that will be used to update $\theta$.

### 4.2 TSVQ-Based DBF Algorithms for Static Channels

In this section, we present TSVQ-based DBF algorithms for static channels. We first design and analyze a simple TSVQ-based DBF algorithm for the case of $N = 2$ transmitters. We then develop a TSVQ-based DBF algorithm that works for a larger number of transmitters and analyze its behavior. The main idea behind our algorithms is to compute (or estimate) the angles that the vectors corresponding to the nodes make relative to a vector with an unchanged (or asymptotically unchanged) phase. Then, these angles are quantized and fed back to the transmitters, and the nodes adjust their phases by subtracting angles corresponding to the reproduction points from their phases. For the sake of simplicity, to quantize a $1 \times n$ vector of angles, in this chapter we uniformly quantize the phases. The
space is partitioned with $n$-dimensional hypercubes with no gaps and no overlaps such that one of these cells is centered at the origin (i.e., one of the reproduction points is the point $(0, \ldots, 0)$).

### 4.2.1 Algorithm for $N = 2$ Transmitters

In this subsection, we present a simple TSVQ-based DBF algorithm for the case of $N = 2$ transmitters with static channels. We use the following proposition depicted in Fig. 4.1 to design and analyze the algorithm:

**Proposition 4.1.** Let $A$, $B$, $B'$ and $B''$ be four distinct points in the plane such that $AB = AB' = AB''$. Suppose that $C$ is a point distinct from $A$, and let $\alpha$ be the angle that $\overrightarrow{AB}$ makes relative to $\overrightarrow{AC}$ and $\beta$ and $\gamma$ be the angles that $\overrightarrow{AB'}$ and $\overrightarrow{AB''}$ make, respectively, relative to $\overrightarrow{AB}$. Let points $D$, $D'$ and $D''$ be such that $\overrightarrow{AD} = \overrightarrow{AB} + \overrightarrow{AC}$, $\overrightarrow{AD'} = \overrightarrow{AB'} + \overrightarrow{AC}$ and $\overrightarrow{AD''} = \overrightarrow{AB''} + \overrightarrow{AC}$, and let $d_1 = AD^2 - AD^2$ and $d_2 = AD''^2 - AD^2$. If $d_2 \sin \beta \neq d_1 \sin \gamma$, then

\[
\tan \alpha = \frac{d_2(\cos \beta - 1) - d_1(\cos \gamma - 1)}{d_2 \sin \beta - d_1 \sin \gamma}. \tag{4.5}
\]
Proof. Let \( r = AB = AB' = AB'' \) and \( r' = AC \). Then, by the law of cosines we have

\[
AD^2 = AB^2 + AC^2 - 2 \cdot AB \cdot AC \cdot \cos(180^\circ - \alpha) = r^2 + r'^2 + 2rr' \cos \alpha, \tag{4.6}
\]

\[
AD^{r_2} = AB^{r_2} + AC^2 - 2 \cdot AB' \cdot AC \cdot \cos(180^\circ - \alpha - \beta) = r^2 + r'^2 + 2rr' \cos(\alpha + \beta), \tag{4.7}
\]

\[
AD^{r_2} = AB^{r_2} + AC^2 - 2 \cdot AB'' \cdot AC \cdot \cos(180^\circ - \alpha - \gamma) = r^2 + r'^2 + 2rr' \cos(\alpha + \gamma). \tag{4.8}
\]

Using (4.6), (4.7) and (4.8), we have

\[
d_1 = AD^{r_2} - AD^2 = 2rr' [\cos(\alpha + \beta) - \cos \alpha] = 2rr' [\cos \alpha (\cos \beta - 1) - \sin \alpha \sin \beta], \tag{4.9}
\]

\[
d_2 = AD^{r_2} - AD^2 = 2rr' [\cos(\alpha + \gamma) - \cos \alpha] = 2rr' [\cos \alpha (\cos \gamma - 1) - \sin \alpha \sin \gamma]. \tag{4.10}
\]

Then, by multiplying (4.9) by \( d_2 \) and (4.10) by \( d_1 \), we find that

\[
2rr' d_1 [\cos \alpha (\cos \gamma - 1) - \sin \alpha \sin \gamma] = 2rr' d_2 [\cos \alpha (\cos \beta - 1) - \sin \alpha \sin \beta]. \tag{4.11}
\]

Note that \( r, r' \neq 0 \) since \( B \) and \( C \) are distinct from \( A \). Also, since \( d_2 \sin \beta \neq d_1 \sin \gamma \), it can be easily shown using (4.11) that \( \cos \alpha \neq 0 \). Thus, by dividing the two sides of (4.9) by \( 2rr' \cos \alpha \), we obtain

\[
d_1 [(\cos \gamma - 1) - \tan \alpha \sin \gamma] = d_2 [(\cos \beta - 1) - \tan \alpha \sin \beta]. \tag{4.12}
\]

Then, solving (4.12) for \( \tan \alpha \), we attain (4.5). This completes the proof. \( \square \)

Our first algorithm, Algorithm 1, consists of \( K + 1 \) stages (Stages 0, \cdots, \( K \)) where \( K \) is the parameter that determines the depth of the TSVQ and thus the maximum possible phase misalignment. In Stage 0, a reference RSS corresponding to the initial phases is measured. In Stage 1, the RSS after applying a phase adjustment of \( \beta = \pi \) to Node 1 is measured, and
Algorithm 1 TSVQ-based DBF algorithm for static channels with $N = 2$ transmitters

```
procedure TSVQ_DBF_STAT_2TX($K, B$

$\beta \leftarrow \pi$  \Comment{Initialization}
$\gamma \leftarrow \pi/2$

Stage 0  
($t = 0$)

$\theta_{TX} \leftarrow \theta$

Transmitters transmit to receiver with $\theta_{TX}$
$r_0 \leftarrow \text{RSS}[0]$ \Comment{Receiver computes RSS[0] and stores it in $r_0$}

Stage 1  
($t = 1$)

$\theta_{TX} \leftarrow \theta$
$\theta_{TX,1} \leftarrow \theta_{TX,1} + \beta$

Transmitters transmit to receiver with $\theta_{TX}$
$r_1 \leftarrow \text{RSS}[1]$ \Comment{Receiver computes RSS[1] and stores it in $r_1$}

if $r_1 > r_0$
then
Receiver feeds back a 0 bit to transmitters \Comment{First bit of TSVQ}
$\theta_1 \leftarrow \theta_{TX,1}$
$r_0 \leftrightarrow r_1$ \Comment{Swap values of $r_0$ and $r_1$}
$\beta \leftarrow -\beta$ \Comment{Switch $\beta$ to $-\beta$}

else
Receiver feeds back a 1 bit to transmitters \Comment{First bit of TSVQ}
end if

Stage 2  
($t = 2$)

$\theta_{TX} \leftarrow \theta$
$\theta_{TX,1} \leftarrow \theta_{TX,1} + \gamma$

Transmitters transmit to receiver with $\theta_{TX}$
$r_2 \leftarrow \text{RSS}[2]$ \Comment{Receiver computes RSS[2] and stores it in $r_2$}

$d_1 \leftarrow (r_1)^2 - (r_0)^2$
$d_2 \leftarrow (r_2)^2 - (r_0)^2$

$\alpha \leftarrow \arctan \left[ \frac{d_2 (\cos \beta - 1) - d_1 (\cos \gamma - 1)}{d_2 \sin \beta - d_1 \sin \gamma} \right]$ \Comment{Receiver computes the desired angle}

Quantize $\alpha$ and feed back the next $B$ bits to transmitters
Update $\theta_1$ using the received feedback bits

for $k \leftarrow 3$ to $K$ do
Feed back the next $B$ bits to transmitters
Update $\theta_1$ using the received feedback bits
end for

end procedure
```
an appropriate phase adjustment to Node 1 (corresponding to the phase that resulted in a larger RSS in Stages 0 and 1) is chosen to reduce the maximum possible phase misalignment from $\pi$ to $\pi/2$. In Stage 2, another RSS after applying a phase adjustment of $\gamma = \pi/2$ to Node 1 is measured, and the phase difference between the two nodes is exactly computed using Proposition 4.1. Then, the result is quantized, the next $B$ bits (i.e., after the one bit corresponding to the adjustment in Stage 1) are fed back to the transmitters, and the phase of Node 1 is adjusted. Similarly, in Stages $k = 3, \cdots, K$, the next $B$ bits are fed back to the transmitters, and the phase of Node 1 is adjusted. In what follows, we describe the operations performed by the transmitters and the receiver in each stage of the algorithm:

- **Stage 0**: In Stage 0, the angles $\beta$ and $\gamma$ are first initialized to $\beta = \pi$ and $\gamma = \pi/2$ radians. The transmitters then transmit to the receiver, and the receiver measures and stores $r_0 = \text{RSS}[0]$ (i.e., the RSS corresponding to the initial phases).

- **Stage 1**: In Stage 1, Node 1 first applies a phase adjustment by adding $\beta = \pi$ radians to its phase. Then, the transmitters transmit to the receiver, and the receiver measures and stores the corresponding RSS, $r_1 = \text{RSS}[1]$. The receiver then compares $r_1$ with $r_0$. If $r_1 > r_0$, it feeds back a 0 bit to the transmitters indicating that the RSS has increased, and Node 1 updates its phase with the adjusted phase in Stage 1. Also, the receiver swaps the values of $r_0$ and $r_1$ and assigns $-\beta$ to $\beta$. As will become clear, these operations simplify the procedure in Stage 2. If $r_1 \leq r_0$, the receiver feeds back a 1 bit to the transmitters, and no action is taken by the transmitters.

- **Stage 2**: In Stage 2, Node 1 first applies a phase adjustment by adding $\gamma = \pi/2$ radians to its phase. Then, the transmitters transmit to the receiver, and the receiver measures and stores the corresponding RSS, $r_2 = \text{RSS}[2]$. The receiver then computes the resulting angle between the vectors of the two nodes, $\alpha$, using Proposition 4.1 as follows:

$$\alpha = \arctan \left[ \frac{d_2(\cos \beta - 1) - d_1(\cos \gamma - 1)}{d_2 \sin \beta - d_1 \sin \gamma} \right]$$

(4.13)
Figure 4.2: Example of phase synchronization using Algorithm 1 for $K = 5$ and $B = 1$. The blue and red segments represent the vectors corresponding to Nodes 1 and 2, respectively, after each stage. The tails of both vectors are at point $(0, 0)$. The green dashes display the boundaries of the quantization regions after each stage.

where $d_1 = (r_1)^2 - (r_0)^2$ and $d_2 = (r_2)^2 - (r_0)^2$. The value of $\alpha$ is then quantized, and the next $B$ bits are fed back to the transmitters. Therefore, the phase of Node 1 is adjusted, and the tree is traversed accordingly.

- **Stages 3, $\cdots$, $K$:** In Stage $k = 3, \cdots, K$, the next $B$ bits are fed back to the transmitters, and the phase of Node 1 is adjusted. Thus, the tree is traversed accordingly.

Fig. 4.2 displays an example of phase synchronization using Algorithm 1 for $K = 5$ and $B = 1$. As shown in the figure, the phase difference for the two nodes will not be greater than $\pi/2^k$ radians after Stage $k$ of the algorithm. Therefore, as the algorithm continues, the phases of the two nodes are synchronized and phase coherence is achieved at the receiver. In the following theorem, we analyze the behavior of Algorithm 1:
**Theorem 4.1.** After Stage $K \geq 1$ of Algorithm 1, the angle between the vectors of the two nodes will have a uniform distribution in $\left( -\pi/2^{B(K-1)+1}, \pi/2^{B(K-1)+1} \right)$.

*Proof.* The result follows from using the law of cosines in Stage 1 and Proposition 4.1 in Stages $k = 2, \cdots, K$.

Therefore, as $K \to \infty$, the phase asynchrony in Algorithm 1 converges to zero with probability 1. The following theorem provides a lower bound for $E[\rho_2\{K\}]$ in Stage $K \geq 3$ of Algorithm 1:

**Theorem 4.2.** In Stage $K \geq 3$ of Algorithm 1, we have

$$E[\rho_2\{K\}] \geq \frac{1}{2} \left[ 1 + \frac{2^{B(K-2)+1}}{\pi} \sin \left( \frac{\pi}{2^{B(K-2)+1}} \right) \right].$$ \hspace{1cm} (4.14)

*Proof.* See Appendix F.

Using Theorem 4.2, it can be easily shown that $E[\rho_2\{K\}] = E[RSS_2\{K\}/RSS_{2,\text{max}}] \to 1$ as $K \to \infty$.

### 4.2.2 Algorithm for a Larger Number of Transmitters

In this subsection, we propose a TSVQ-based DBF algorithm that works for $N > 2$ transmitters with static channels. One approach is to use Algorithm 1 for each Node $n = 1, \cdots, N$ to calculate the angle between its corresponding vector and the sum of all the initial vectors. Then, these angles can be quantized and fed back to the transmitters. Although this approach is possible, it is not desirable. In order to achieve higher RSS values in the initial stages of the algorithm, we propose another approach to estimate the phases using a coarse synchronization (with a phase adjustment of $\pi$ for each node) followed by a finer one (with
a phase adjustment of $\pi/2$ for each node). We use the following propositions to design and analyze the algorithm:

**Proposition 4.2.** For any $r > 0$, let $E_{n,r}$, $n = 1, \ldots, N$, denote the event that $\frac{\text{RSS}[0]}{a_n} > r$ where $a_n$ is the attenuation of the channel from Node $n$ to the receiver. Then, for any $r > 0$ we have

$$
\lim_{N \to \infty} P \left( \bigcap_{n=1}^{N} E_{n,r} \right) = 1.
$$

**(4.15)**

*Proof.* See Appendix G.

Proposition 4.2 states that the probability that the sum of the initial vectors (RSS[0]) is arbitrarily larger than the length of all the vectors $(a_n, n = 1, \ldots, N)$ goes to 1 as $N \to \infty$. In the following proposition, we consider this fact and derive a formula that can be used to estimate the phase differences, as depicted in Fig. 4.3:

**Proposition 4.3.** For a fixed point $A$ in the plane, let $B$ and $B'$ be variable points such that $AB = AB'$. Suppose that $C$ is another variable point, and let $\alpha = \angle BAC$ and $\beta = \angle B'AC$ denote the corresponding variable angles. Let points $D$ and $D'$ be such that $\overrightarrow{AD} = \overrightarrow{AB} + \overrightarrow{AC}$ and $\overrightarrow{AD'} = \overrightarrow{AB'} + \overrightarrow{AC}$. Then, $\frac{\overrightarrow{AD} - \overrightarrow{AD'}}{AB} \to \cos \alpha - \cos \beta$ as $\frac{AC}{AB} \to \infty$.

*Proof.* Let $C'$ and $C''$ be the points on rays $AD$ and $AD'$, respectively, such that $AC =
\[ AC' = AC''. \] Then, we have

\[
\frac{AD - AD'}{AB} = \frac{(AC' + C'D) - (AC'' + C''D')}{AB} = \frac{C'D}{AB} - \frac{C''D'}{AB}. \tag{4.16}
\]

Now, consider triangle \( CAD \), and let \( \gamma = \angle CAD \), \( \theta = \angle CDA \), and \( H \) be the foot of the altitude from \( D \). Note that \( AB = CD \) and \( \angle DCH = \alpha \). Thus, \( \frac{AH}{AB} = \frac{AC + CH}{AB} = \frac{AC}{AB} + \cos \alpha \).

Since \( |\cos \alpha| \leq 1 \) is finite, this implies that as \( \frac{AC}{AB} \to \infty \), we have \( \frac{AH}{AB} \to \infty \). On the other hand, \( \frac{DH}{AB} = |\sin \alpha| \leq 1 \) is finite. Therefore, as \( \frac{AC}{AB} \to \infty \), we have \( \tan \gamma = \frac{DH}{AH} = \frac{DH/AB}{AH/AB} \to 0 \), and thus, \( \gamma \to 0 \). Additionally, we have \( \theta + \gamma = \angle DCH = \alpha \). Therefore, as \( \frac{AC}{AB} \to \infty \), \( \theta = \alpha - \gamma \to \alpha \).

Since \( AC = AC' \), \( ACC' \) is an isosceles triangle. Therefore, \( \angle CC'A = 90^\circ - \frac{\gamma}{2} \), and thus, \( \angle CC'D = 90^\circ + \frac{\gamma}{2} \). This implies that as \( \frac{AC}{AB} \to \infty \), \( \angle CC'D \to 90^\circ \), and thus, \( \angle C'D \to 90^\circ - \theta \). Therefore, as \( \frac{AC}{AB} \to \infty \), using the law of sines we have

\[
\frac{C'D}{AB} = \frac{C'D}{CD} \to \frac{\sin(90^\circ - \theta)}{\sin(90^\circ)} = \cos \theta \to \cos \alpha. \tag{4.17}
\]

Using a similar argument to the above, one may show that as \( \frac{AC}{AB} \to \infty \), we have

\[
\frac{C''D'}{AB} \to \cos \beta. \tag{4.18}
\]

Therefore, using (4.16), (4.17) and (4.18), we conclude that \( \frac{AD - AD'}{AB} \to \cos \alpha - \cos \beta \) as \( \frac{AC}{AB} \to \infty \).

We are now ready to present our second algorithm, Algorithm 2. Similar to Algorithm 1, Algorithm 2 consists of \( K + 1 \) stages (Stages 0, \cdots, K) where \( K \) is the parameter that determines the depth of the TSVQ and thus the maximum possible phase misalignment. In Stage 0, a reference RSS corresponding to the initial phases is measured. In Stage 1, for any Node \( n = 1, \cdots, N \), the RSS after applying a phase adjustment of \( \beta = \pi \) only to
Algorithm 2: TSVQ-based DBF algorithm for static channels with $N > 2$ transmitters

```
procedure TSVQ_DBF_STAT_NTX($K, B, L$)
  \( \beta \leftarrow \pi \) \Comment{Initialization}
  \( \gamma \leftarrow \pi / 2 \)
  \( t \leftarrow 0 \) \Comment{Current time}
  \( \theta_{TX} \leftarrow \theta \)
  Transmitters transmit to receiver with \( \theta_{TX} \)
  \( r_t \leftarrow \text{RSS}[t] \Comment{Receiver computes RSS[t] and stores it in } r_t \)
  \( \text{RSS}_{last} \leftarrow r_t \Comment{RSS value for the last updated phases} \)

Stage 0
\((t = 0)\)

\( \theta_{\text{new}} \leftarrow \theta \Comment{Used to update } \theta \text{ after } n = \left\lceil \frac{N}{2} \right\rceil, \ l = 1, \cdots, L, \text{ transmissions for the nodes} \)
for \( n \leftarrow 1 \) to \( N \) do
  \( t \leftarrow t + 1 \)
  \( \theta_{TX} \leftarrow \theta \)
  \( \theta_{TX,n} \leftarrow \theta_{TX} + \beta \)
  Transmitters transmit to receiver with \( \theta_{TX} \)
  \( r_t \leftarrow \text{RSS}[t] \Comment{Receiver computes RSS[t] and stores it in } r_t \)
  if \( r_t > \text{RSS}_{last} \) then
    Receiver feeds back a 0 bit to transmitters \Comment{First bit of TSVQ for Node } n
    \( \theta_{\text{new},n} \leftarrow \theta_{TX,n} \)
    \( \beta_n \leftarrow -\beta \)
    \( \delta_{1,n} \leftarrow \text{RSS}_{last} - r_t \)
  else
    Receiver feeds back a 1 bit to transmitters \Comment{First bit of TSVQ for Node } n
    \( \beta_n \leftarrow \beta \)
    \( \delta_{1,n} \leftarrow r_t - \text{RSS}_{last} \)
  end if
end for

Stage 1
\((1 \leq t \leq N+L)\)

if \( n = \left\lceil \frac{N}{2} \right\rceil \) then
  \( t \leftarrow t + 1 \)
  \( l \leftarrow l + 1 \)
  \( \theta \leftarrow \theta_{\text{new}} \Comment{Transmitters update their phases} \)
  \( \theta_{TX} \leftarrow \theta_{\text{new}} \)
  Transmitters transmit to receiver with \( \theta_{TX} \)
  \( \text{RSS}_{last} \leftarrow \text{RSS}[t] \Comment{Receiver computes and stores the RSS for the last updated phases} \)
end if

Stage 2
\((N+L+1 \leq t \leq 2N+L)\)

if \( n = \left\lceil \frac{N}{2} \right\rceil \) then
  \( t \leftarrow t + 1 \)
  \( l \leftarrow l + 1 \)
  \( \theta \leftarrow \theta_{\text{new}} \Comment{Transmitters update their phases} \)
  \( \theta_{TX} \leftarrow \theta_{\text{new}} \)
  Transmitters transmit to receiver with \( \theta_{TX} \)
  \( \text{RSS}_{last} \leftarrow \text{RSS}[t] \Comment{Receiver computes and stores the RSS for the last updated phases} \)
end if

for \( k = 3, \cdots, K \) do
  for \( n \leftarrow 1 \) to \( N \) do
    Feed back the next \( B \) bits for Node \( n \)
    Update \( \theta_n \) using the received feedback bits
  end for
end for
end procedure
```
Node $n$ is measured. Also, appropriate phase adjustments (corresponding to the phases that resulted in larger RSS values in Stages 0 and 1) are chosen and applied only at $L < N$ specific time slots where $L$ is a parameter. As will become clear later, if these adjustments were applied after each of the $N$ RSS measurements, the resulting RSS values after applying each of these adjustments would need to be measured immediately as well, and thus the convergence time would increase. Therefore, these adjustments are not applied after each of the $N$ RSS measurements. In Stage 2, for any Node $n = 1, \cdots, N$, another RSS after applying a phase adjustment by $\gamma = \pi/2$ only to Node $n$ is measured, and the angle between the vector corresponding to Node $n$ after Stage 1 and the sum of all the initial vectors in Stage 0 is estimated using Proposition 4.3. Then, for any Node $n = 1, \cdots, N$, the result is quantized, the next $B$ bits (i.e., after the one bit per node corresponding to the adjustments in Stage 1) are fed back to the transmitters, and the phases of the nodes are adjusted. Similarly, in Stages $k = 3, \cdots, K$, for any Node $n = 1, \cdots, N$, the next $B$ bits are fed back to the transmitters, and the phases of the nodes are adjusted. In the following, we describe the operations performed by the transmitters and the receiver in each stage of the algorithm. We use the variable $t$ to indicate the transmission time and assume that the first transmission occurs at time $t = 0$.

- **Stage 0**: In Stage 0, the angles $\beta$ and $\gamma$ are first initialized to $\beta = \pi$ and $\gamma = \pi/2$ radians. The transmitters then transmit to the receiver, and the receiver measures and stores $r_t = RSS[t]$ at time $t = 0$ (i.e., the RSS corresponding to the initial phases). The receiver also initializes $RSS_{\text{last}}$ to $RSS_{\text{last}} = RSS[0]$. The value of this variable indicates the RSS corresponding to the last updated phases and will be used and updated in the next stage.

- **Stage 1**: In Stage 1, $\theta_{\text{new}}$ and $l$ are initialized to $\theta_{\text{new}} = \theta$ and $l = 1$ at the transmitters. All the nodes update their phases using the elements of $\theta_{\text{new}}$, which holds the most recent adjusted phase values. These updates occur at transmission times $\left\lceil \frac{Nt}{L} \right\rceil$ where
$L$ is one of the algorithm’s parameters and the value of $l$ is incremented by 1 each time all the phases are updated. Then, the following operations are performed for any Node $n = 1, \cdots, N$:

- Node $n$ first applies a phase adjustment by adding $\beta = \pi$ radians to its phase. Then, the transmitters transmit to the receiver, and the receiver measures and stores the corresponding RSS, $r_t = \text{RSS}[t]$.

- The receiver then compares $r_t$ with $\text{RSS}_{\text{last}}$. If $r_t > \text{RSS}_{\text{last}}$, it feeds back a 0 bit to the transmitters indicating that the RSS has increased, and Node $n$ updates $\theta_{\text{new}, n}$ with the adjusted phase in Stage 1. Also, the receiver assigns $-\beta$ to $\beta_n$ and $\text{RSS}_{\text{last}} - r_t$ to $\delta_{1,n}$ where $\beta = (\beta_1, \cdots, \beta_N)$ and $\delta_1 = (\delta_{1,1}, \cdots, \delta_{1,N})$ are $1 \times N$ vectors and are used in Stage 2. If $r_t \leq \text{RSS}_{\text{last}}$, the receiver feeds back a 1 bit to the transmitters, it assigns $\beta$ to $\beta_n$ and $r_t - \text{RSS}_{\text{last}}$ to $\delta_{1,n}$, and no action is taken by the transmitters.

- Finally, the transmitters check if $n = \lceil \frac{Nl}{L} \rceil$. If that is the case, they increment the value of $l$ by 1, update their phases with $\theta_{\text{new}}$, and then transmit to the receiver. Then, the receiver measures $\text{RSS}[t]$ and assigns it to $\text{RSS}_{\text{last}}$.

- **Stage 2:** In Stage 2, $\theta_{\text{new}}$ and $l$ are initialized to $\theta_{\text{new}} = \theta$ and $l = 1$ at the transmitters. Then, the following operations are performed for any Node $n = 1, \cdots, N$:

  - Node $n$ first applies a phase adjustment by adding $\gamma = \pi/2$ radians to its phase. Then, the transmitters transmit to the receiver, and the receiver measures and stores the corresponding RSS, $r_t = \text{RSS}[t]$. The receiver also assigns $r_t - \text{RSS}_{\text{last}}$ to $\delta_{2,n}$ where $\delta_2 = (\delta_{2,1}, \cdots, \delta_{2,N})$ is a $1 \times N$ vector and will be used to estimate the phase differences. Let $v_n$ denote the length of the sum of the vectors corresponding to all nodes other than Node $n$ in Stage 0, and let $\tilde{\alpha}_n$ denote the angle that the vector corresponding to Node $n$ after Stage 1 makes relative to this sum.
vector. Using Proposition 4.2, it can be easily shown that the probability that 
$|v_n|/a_n \to \infty$, \ $\forall n$, goes to 1 as $N \to \infty$. Then, using $\delta_{1,n},\delta_{2,n}$ and by applying
Proposition 4.3, it can be shown that the following events occur with a probability 
that goes to 1 as $N \to \infty$:

$$\frac{\delta_{1,n}}{a_n} \rightarrow \cos(\tilde{\alpha}_n + \beta_n) - \cos(\alpha_n), \quad \frac{\delta_{2,n}}{a_n} \rightarrow \cos(\tilde{\alpha}_n + \gamma) - \cos(\alpha_n). \quad (4.19)$$

Noting that $a_n \neq 0$ with probability 1, if we multiply the two sides of the two 
relationships in (4.19) by $a_n$, the resulting relationships will be similar to Eqs. 
(4.9) and (4.10) in the proof of Proposition 4.1. Then, using a similar procedure 
to that of the proof for Proposition 4.1, we can obtain an estimate for $\tilde{\alpha}_n$. It can 
also be shown that $\tilde{\alpha}_n$ converges to the angle that the vector corresponding to 
Node $n$ after Stage 1 makes relative to the sum of all the initial vectors in Stage 
0 ($\alpha_n$) in probability as $N \to \infty$. Therefore, the receiver estimates $\alpha_n$ as follows:

$$\alpha_n = \arctan \left[ \frac{\delta_{2,n} \cdot (\cos(\beta_n) - 1) - \delta_{1,n} \cdot (\cos(\gamma) - 1)}{\delta_{2,n} \cdot \sin(\beta_n) - \delta_{1,n} \cdot \sin(\gamma)} \right]. \quad (4.20)$$

The value of $\alpha_n$ is then quantized, and the next $B$ bits are fed back to the 
transmitters. Therefore, Node $n$ updates $\theta_{\text{new},n}$, which will be used to update the 
phase of Node $n$ and traverse the tree.

- Finally, the transmitters check if $n = \lceil \frac{Nl}{L} \rceil$. If that is the case, they increment the 
value of $l$ by 1, update their phases with $\theta_{\text{new}}$, and then transmit to the receiver.

Then, the receiver measures RSS$[t]$ and assigns it to RSS$\text{last}$.

- Stages 3, ⋅⋅⋅, $K$: In Stage $k = 3, \cdots, K$, for $n = 1, \cdots, N$, the next $B$ bits for Node $n$ 
are fed back to the transmitters, and the phase of Node $n$ is adjusted. Thus, the tree 
is traversed accordingly.

Note that after estimating the angles in Stage 2, any quantization method can be used to
quantize the angles and feed back the bits to the transmitters. Depending on the choice of quantization method, the RSS values will change at different time slots. Fig. 4.4 displays an example of phase synchronization using Algorithm 2 for $N = 1000$, $K = 5$, $B = 1$ and $L = 25$. As shown in the figure, the phase difference between the sum of the vectors corresponding to the nodes and the vector corresponding to almost any node is not greater than $\pi/2^k$ radians after Stage $k$ of the algorithm. Thus, the phases of the nodes are synchronized and phase coherence is achieved at the receiver. In the following theorem, we analyze the behavior of Algorithm 2:

**Theorem 4.3.** For any $\epsilon > 0$, let $S_\epsilon$ denote the set of nodes whose vectors are initially in a distance of greater than $\epsilon$ radians from all the boundaries of the quantization regions. Then, for any $\epsilon > 0$, the probability that all the phases corresponding to the nodes in $S_\epsilon$ are moved
Figure 4.5: Histogram of the angles corresponding to the vectors in a sample run of Algorithm 2 for $N = 1000$, $K = 5$, $B = 1$ and $L = 25$.

into a single quantization region of length $\pi/2^{K-1}$ radians after Stage $K \geq 1$ of Algorithm 2 goes to 1 as $N \to \infty$.

Proof Sketch. See Appendix H.

Therefore, the probability that phase synchronization is attained in Algorithm 2 goes to 1 as $N \to \infty$. The histogram of the angles corresponding to the vectors in a sample run of Algorithm 2 for $N = 1000$, $K = 5$, $B = 1$ and $L = 25$ is shown in Fig. 4.5. As can be seen in the figure, the phase distribution of the nodes is almost a uniform distribution in an interval of length $\pi/2^{k-1}$ after completing Stage $k$ of the algorithm. This is consistent with the result of Theorem 4.3 and the following claim that states the ratio of the number of nodes whose phases are moved into the quantization region described in Theorem 4.3 to $N$ converges to 1 in probability as $N \to \infty$. 

72
Corollary 4.1. For $N$ transmitting nodes, let $Z_N$ be a random variable denoting the number of nodes whose phases are moved into the quantization region described in Theorem 4.3 after Stage $K \geq 1$ of Algorithm 2. Then,

$$\frac{Z_N}{N} \xrightarrow{p} 1 \text{ as } N \to \infty.$$  \hfill (4.21)

Proof. See Appendix I. \qed

The following theorem quantifies the value of $\rho_N\{K\}$ as $N \to \infty$ in Stage $K \geq 1$ of Algorithm 2:

**Theorem 4.4.** In Stage $K \geq 1$ of Algorithm 2, we have

$$\rho_N\{K\} \xrightarrow{p} \frac{2^{B(K-1)+1}}{\pi} \sin \left( \frac{\pi}{2^{B(K-1)+1}} \right) \text{ as } N \to \infty.$$  \hfill (4.22)

Proof. See Appendix J. \qed

Using Theorem 4.4, it can be easily shown that $\rho_N\{K\} = \frac{\text{RSS}_N\{K\}}{\text{RSS}_{N,\text{max}}} \xrightarrow{p} 1$ as $N \to \infty$ and $K \to \infty$.

### 4.3 TSVQ-Based DBF Algorithms for Time-Varying Channels

In this section, we modify our TSVQ-based DBF algorithms to enable them to track time-varying channels without the knowledge of CSI. First, we introduce our algorithm for the case of $N = 2$ transmitters. Then, we present an algorithm that works for a larger number of transmitters.
4.3.1 Algorithm for $N = 2$ Transmitters

In this subsection, we introduce our algorithm for the case of $N = 2$ transmitters. Our third algorithm, Algorithm 3, consists of multiple parts as well. The main idea behind Algorithm 3 is to first perform the $K + 1$ stages of Algorithm 1 as if the channel is static, and then start a process that adaptively chases and approximates the random phase changes. In other words, while the random phase changes are insignificant, Algorithm 1 can be used to approximate and adjust the phases. When the phases start to drift from each other due to the time-dependent channel variations, a new process is used to track the phase changes. This process starts with measuring the new RSS corresponding to the phases after Stage $K$ of the algorithm and then iteratively adjusting the phase of Node 1 by $\pm 2\Delta_{\text{max}}$. The sign of this adjustment is updated at each time slot. If the new phase results in a lower RSS than the RSS at the previous time slot, the sign of this adjustment is switched. Otherwise, the next iteration is performed using the same value for phase adjustment. Note that parameter $K$ should be chosen based on the value of $\Delta_{\text{max}}$, which determines the maximum and minimum possible values of the drift process $\Delta[t]$. A larger value for $\Delta_{\text{max}}$ would result in larger potential values for the drift process and thus faster changes in phases. Therefore, for a large value of $\Delta_{\text{max}}$, a small integer value for $K$ should be picked such that the algorithm starts the chasing process sooner to chase and approximate the random phase changes. In what follows, we describe the operations performed by the transmitters and the receiver in each part of the algorithm. Again, we use the variable $t$ to indicate the transmission time and assume that the first transmission occurs at time $t = 0$.

- **Stages 0, $\cdots$, $K$:** Stages 0, $\cdots$, $K$ of Algorithm 3 are the same as Stages 0, $\cdots$, $K$ of Algorithm 1. However, in Stage 2 of Algorithm 3, the resulting angle between the vectors of the two nodes, $\alpha$, is only an approximation, while that angle is exactly computed in Stage 2 of Algorithm 1.
Algorithm 3 TSVQ-based DBF algorithm for time-varying channels with $N = 2$ transmitters

\begin{verbatim}
procedure TSVQ_DBF_TV.2TX(K, B)
    \( \beta \leftarrow \pi \) \quad \triangleright \text{Initialization}
    \( \gamma \leftarrow \pi / 2 \)
    \( \theta_{TX} \leftarrow \theta \)
    Transmitters transmit to receiver with \( \theta_{TX} \)
    \( r_0 \leftarrow \text{RSS}[0] \) \quad \triangleright \text{Receiver computes RSS}[0] and stores it in \( r_0 \)

    if \( r_1 > r_0 \) then
        First bit of TSVQ
        \( \theta_1 \leftarrow \theta_{TX,1} \)
        \( r_0 \leftrightarrow r_1 \) \quad \triangleright \text{Swap values of } r_0 \text{ and } r_1
        \( \beta \leftarrow -\beta \) \quad \triangleright \text{Switch } \beta \text{ to } -\beta
    else
        Receiver feeds back a 1 bit to transmitters \quad \triangleright \text{First bit of TSVQ}
    end if

    for \( k \leftarrow 3 \) to \( K \) do
        \( d_i \leftarrow \frac{(r_1)^2 - (r_0)^2}{(r_2)^2 - (r_0)^2} \)
        \( \alpha \leftarrow \arctan \left[ \frac{d_1 (\cos \beta - 1) - d_1 (\cos \gamma - 1)}{d_1 \sin \beta - d_1 \sin \gamma} \right] \) \quad \triangleright \text{Receiver approximates the desired angle}
        Quantize } \alpha \text{ and feed back the next } B \text{ bits to transmitters}
        Update } \theta_1 \text{ using the received feedback bits
    end for

    for \( k \leftarrow 3 \) to \( K \) do
        Feed back the next } B \text{ bits to transmitters
        Update } \theta_1 \text{ using the received feedback bits
    end for

    for \( t = K + 1 \) to \( t = K + 2 \) do
        RSS Update
        Transmitters transmit to receiver with } \theta_{TX}
        \( r_t \leftarrow \text{RSS}[t] \) \quad \triangleright \text{Receiver computes RSS}[t] and stores it in } r_t
        \( d \leftarrow 1 \) \quad \triangleright \text{ } d \in \{-1, 1\} \text{ indicates the direction}
    end for

    Iteration } i \geq 1 \quad (t \geq K + 2)
    loop
        \( t \leftarrow t + 1 \)
        \( \theta_{TX} \leftarrow \theta \)
        Transmitters transmit to receiver with } \theta_{TX}
        \( r_t \leftarrow \text{RSS}[t] \) \quad \triangleright \text{Receiver computes RSS}[t] and stores it in } r_t
        if \( r_t < r_{t-1} \) then
            Receiver feeds back a 0 bit to transmitters
            \( d \leftarrow -d \) \quad \triangleright \text{Change direction}
        else
            Receiver feeds back a 1 bit to transmitters
        end if
        \( \theta_1 \leftarrow \theta_1 + 2d \Delta_{\text{max}} \) \quad \triangleright \text{Update } \theta_1 \end{verbatim}
• RSS Update: Next, the transmitters transmit to the receiver, and the receiver measures and stores the corresponding RSS, \( r_t = \text{RSS}[t] \) where \( t = K + 1 \). At the transmitters, \( d \) is also initialized to \( d = 1 \). The variable \( d \in \{-1, 1\} \) indicates the direction for the phase adjustment and is used in the following.

• Iteration \( i \geq 1 \): The following stages of the algorithm are executed repeatedly:
  
  – The transmitters transmit to the receiver, and the receiver measures and stores the corresponding RSS, \( r_t = \text{RSS}[t] \).
  
  – If \( r_t < r_{t-1} \), the receiver feeds back a 0 bit to the transmitters indicating that the RSS has decreased, and Node 1 assigns \(-d\) to \( d \). Otherwise, the receiver feeds back a 1 bit to the transmitters, and no action is taken by the transmitters.
  
  – Finally, Node 1 applies a phase adjustment by adding \( 2d\Delta_{\text{max}} \) radians to its phase.

Fig. 4.6 displays an example of phase synchronization using Algorithm 3 for \( K = 3 \) and \( B = 1 \) where the drift process described in Section 4.1 is used with a parameter \( \Delta_{\text{max}} = \pi/8 \). As shown in the figure, the two nodes attain a desired level of phase coherence with some misalignment due to the variations in the channel.

4.3.2 Algorithm for a Larger Number of Transmitters

In this subsection, we present an algorithm that works for a larger number of transmitters, and we refer to it as Algorithm 4. Algorithm 4 consists of multiple stages as well. In Stage 0, a reference RSS corresponding to the initial phases is measured. Then, for any Node \( n = 1, \cdots, N \), an approximation to the phase of Node \( n \) is calculated using a similar procedure to that of Algorithm 1. The latter operations are then repeated iteratively. In other words, Stage 0 of Algorithm 4 is executed once, while all other stages are repeated continuously. Note that the method used in Algorithm 2 (i.e., coarse synchronization followed by a finer
Algorithm 4 TSVQ-based DBF algorithm for time-varying channels with $N > 2$ transmitters

```
procedure TSVQ_DBF_TV_NTX($K, B$)
    \( \beta \leftarrow \pi \) \quad \text{Initialization}
    \( \gamma \leftarrow \pi/2 \)
    Stage 0 \quad (t = 0) \quad \text{Current time}
    \( \theta_{TX} \leftarrow \theta \)
    Transmitters transmit to receiver with \( \theta_{TX} \)
    \( r_t \leftarrow \text{RSS}[t] \) \quad \text{Receiver computes RSS}[t] and stores it in \( r_t \)

    loop
        for \( n \leftarrow 1 \) to \( N \) do
            \( t \leftarrow t + 1 \)
            \( \theta_{TX} \leftarrow \theta \)
            \( \theta_{TX,n} \leftarrow \theta_{TX,n} + \beta \)
            Transmitters transmit to receiver with \( \theta_{TX} \)
            \( r_t \leftarrow \text{RSS}[t] \) \quad \text{Receiver computes RSS}[t] and stores it in \( r_t \)
            if \( r_t > r_{t-1} \) then
                Receiver feeds back a 0 bit to transmitters \quad \text{First bit of TSVQ for Node } n
                \( \theta_n \leftarrow \theta_{TX,n} \)
                \( r_t \leftrightarrow r_{t-1} \) \quad \text{Swap values of } r_t \text{ and } r_{t-1}
                \( \beta_{\text{tmp}} \leftarrow -\beta \) \quad \text{Assign } -\beta \text{ to } \beta_{\text{tmp}}
            else
                Receiver feeds back a 1 bit to transmitters \quad \text{First bit of TSVQ for Node } n
                \( \beta_{\text{tmp}} \leftarrow \beta \) \quad \text{Assign } \beta \text{ to } \beta_{\text{tmp}}
            end if
        end for
        Iteration \( i \geq 1 \):
        Stage 1 \quad (t = (i-1)KN+2n) \quad n = 1, \cdots, N)
        \( r_t \leftarrow \text{RSS}[t] \) \quad \text{Receiver computes RSS}[t] and stores it in \( r_t \)
        \( d_1 \leftarrow (r_{t-1})^2 - (r_{t-2})^2 \)
        \( d_2 \leftarrow (r_t)^2 - (r_{t-2})^2 \)
        \( \alpha[n] \leftarrow \arctan \left[ \frac{d_2(\cos(\beta_{\text{tmp}})-1)-d_1(\cos(\gamma)-1)}{d_2 \sin(\beta_{\text{tmp}})-d_1 \sin(\gamma)} \right] \) \quad \text{Receiver approximates the desired angle}
        Quantize \( \alpha[n] \) and feed back the next \( B \) bits for Node \( n \)
        Update \( \theta_n \) using the received feedback bits
    end for

    Iteration \( i \geq 1 \):
    Stage \quad (t = (i-1)KN+2n+1 \quad n = 1, \cdots, N)
    \( r_t \leftarrow \text{RSS}[t] \) \quad \text{Receiver computes RSS}[t] and stores it in \( r_t \)
    for \( k \leftarrow 3 \) to \( K \) do
        for \( n \leftarrow 1 \) to \( N \) do
            Feed back the next \( B \) bits for Node \( n \)
            Update \( \theta_n \) using the received feedback bits
        end for
    end for
end loop
end procedure
```
Figure 4.6: Example of phase synchronization using Algorithm 3 for $K = 3$ and $B = 1$ where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/8$. The blue and red segments represent the vectors corresponding to Nodes 1 and 2, respectively, after each stage/iteration. The tails of both vectors are at point $(0, 0)$.

one) cannot be used in the time-varying case when there are significant random phase changes occurring from Stage 1 to Stage 2. The operations performed by the transmitters and the receiver in each part of the algorithm are described in the following:

- **Stage 0**: In Stage 0, the angles $\beta$ and $\gamma$ are first initialized to $\beta = \pi$ and $\gamma = \pi/2$ radians. The transmitters then transmit to the receiver, and the receiver measures and stores $r_t = \text{RSS}[t]$ at time $t = 0$ (i.e., the RSS corresponding to the initial phases).

- **Iteration $i \geq 1$**: The following stages of the algorithm are executed for any Node $n = 1, \ldots, N$ repeatedly:
  
  - **Stage 1**: Node $n$ first applies a phase adjustment by adding $\beta = \pi$ radians to its phase. Then, the transmitters transmit to the receiver, and the receiver measures and stores the corresponding RSS, $r_t = \text{RSS}[t]$. The receiver then compares $r_t$ with $r_{t-1}$. If $r_t > r_{t-1}$, it feeds back a 0 bit to the transmitters indicating that the RSS has increased, and Node $n$ updates its phase with the adjusted phase in Stage 1 of the current iteration. Also, the receiver swaps the values of $r_t$ and $r_{t-1}$.
and assigns $-\beta$ to $\beta_{\text{tmp}}$, which is used in Stage 2. If $r_t \leq r_{t-1}$, the receiver feeds back a 1 bit to the transmitters, it assigns $\beta$ to $\beta_{\text{tmp}}$, and no action is taken by the transmitters.

- **Stage 2**: In Stage 2, Node $n$ first applies a phase adjustment by adding $\gamma = \pi/2$ radians to its phase. Then, the transmitters transmit to the receiver, and the receiver measures and stores the corresponding RSS, $r_t = \text{RSS}_t$. Using Proposition 4.1, the receiver estimates the angle between the vector corresponding to Node $n$ after Stage 1 and the sum of all the initial vectors in Stage 0 as follows:

$$
\alpha[n] = \arctan \left[ \frac{d_2(\cos(\beta_{\text{tmp}}) - 1) - d_1(\cos(\gamma) - 1)}{d_2 \sin(\beta_{\text{tmp}}) - d_1 \sin(\gamma)} \right]
$$

(4.23)

where $d_1 = (r_{t-1})^2 - (r_{t-2})^2$ and $d_2 = (r_t)^2 - (r_{t-2})^2$. The value of $\alpha[n]$ is then quantized, and the next $B$ bits are fed back to the transmitters. Therefore, the phase of Node $n$ is adjusted, and the tree is traversed accordingly.

- **Stages 3, \cdots, K**: In Stage $k = 3, \cdots, K$, for $n = 1, \cdots, N$, the next $B$ bits for Node $n$ are fed back to the transmitters, and the phase of Node $n$ is adjusted. Thus, the tree is traversed accordingly.

Again, note that parameter $K$ should be chosen based on the value of $\Delta_{\text{max}}$. For a large value of $\Delta_{\text{max}}$, a small integer value for $K$ should be selected to make each iteration of the algorithm shorter and thus increase the number of times phase estimation is performed in the algorithm. Fig. 4.7 displays an example of phase synchronization using Algorithm 4 for $N = 100$, $K = 3$ and $B = 1$ where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/35$. As shown in the figure, all the nodes attain a desired level of phase coherence with some misalignment due to the variations in the channel.
Figure 4.7: Example of phase synchronization using Algorithm 4 for $N = 100$, $K = 3$ and $B = 1$ where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/35$. The blue segments represent the vectors corresponding to the nodes after each iteration. The tails of all the vectors are at point $(0, 0)$.

4.4 Simulation Results

In this section, we provide simulation results for the expected value of the ratio of the RSS to its maximum possible value, $E\{\rho_N[t]\}$, for the proposed TSVQ-based DBF algorithms for $B = 1, 2, 3$ bits of feedback per time slot. We compare the expected value of the ratio of the RSS to its maximum possible value for our algorithms with that of the algorithms in [22], [25] and [33] in the case of static channels, and to that of the algorithms in [24], [25] and [34] in the case of time-varying channels. For static channels, we use the hybrid approach in [25] and the hybrid approach in [39] and [33] where the proposed Deterministic Joint Activation (DJA) methods with parameters $K = 2, 4, 8, \cdots$ are combined sequentially. For time-varying channels, we use the hybrid approach in [25] and the DJA method in [39] and [34]. The methods selected in all simulations provide the best results with our simulations' parameters among all similar methods presented in the above papers. Additionally, we provide the curves for the lower bounds and the asymptotic values derived in Theorems 4.2 and 4.4, respectively, for the purpose of comparison. The channel magnitudes are i.i.d.
Figure 4.8: The values of $E\{\rho_N[t]\}$ achieved using Algorithm 1 for $K = 15$ and $B = 1, 2, 3$, the corresponding lower bounds based on Theorem 4.2, and the values of $E\{\rho_N[t]\}$ achieved using the existing algorithms for static channels.

random variables with a Rayleigh distribution, and the initial phases before synchronization are independent and uniformly distributed in $(-\pi, \pi)$. In all simulations, the values chosen for the parameters of the existing algorithms are numerically optimized.

Fig. 4.8 shows $E\{\rho_N[t]\}$ as a function of $t$ using Algorithm 1 for $K = 15$ and $B = 1, 2, 3$ in the case of static channels. Fig. 4.9 displays similar results for Algorithm 2 for $N = 1000$, $K = 5$, $B = 1, 2, 3$ and $L = 25$. The convergence time of the proposed TSVQ-based DBF algorithms is linear in $K$, while that of the deterministic DBF algorithms in [33] is exponential in $K$. This is because the algorithms in [33] require testing almost all the $2^K$ possible rotation angles per node and selecting the best one. Therefore, as shown in Figs. 4.8 and 4.9, the convergence time of our algorithms will be significantly lower. Moreover, the proposed TSVQ-based DBF algorithms can feed back more than 1 bit per time slot for faster phase synchronization, while the existing deterministic DBF algorithms can only feed back 1 bit per time slot. Also, as depicted in Fig. 4.8, the deterministic DBF algorithms presented
Theorem 3.4: $B=1, K=5, L=25$
Algorithm 2: $B=2, K=5, L=25$
Algorithm 2: $B=3, K=5, L=25$

Hybrid [39], [33]
Hybrid [25]

Figure 4.9: The values of $E\{\rho_N[t]\}$ achieved using Algorithm 2 for $N = 1000$, $K = 5$, $B = 1, 2, 3$ and $L = 25$, the corresponding asymptotic values based on Theorem 4.4, and the values of $E\{\rho_N[t]\}$ achieved using the existing algorithms for static channels.

in [33] perform poorly during the synchronization process for a small number of nodes since they require testing several bad rotation angels to find the best one. All simulation results demonstrate that the proposed algorithms provide significant improvement compared to the existing algorithms in [22], [25] and [33]. As shown in Fig. 4.8, the lower bounds obtained in Theorem 4.2 are tight at time slots $t \geq 3$. Moreover, the asymptotic values derived in Theorem 4.4 closely match the values of $E\{\rho_N[t]\}$ for the proposed TSVQ-based DBF algorithms displayed in Fig. 4.9.

In Fig. 4.10, we show $E\{\rho_N[t]\}$ as a function of $t$ using Algorithm 3 for $K = 3$ and $B = 1, 2, 3$ in the case of time-varying channels where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/8$. Fig. 4.11 displays similar results for Algorithm 4 for $N = 100$, $K = 3$ and $B = 1, 2, 3$ in the case of time-varying channels where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/35$. As can be seen, all the proposed TSVQ-based DBF algorithms yield a ceiling due to the variations in the channel over time.
Figure 4.10: The values of $E\{\rho_N[t]\}$ achieved using Algorithm 3 for $K = 3$ and $B = 1, 2, 3$ and using the existing algorithms for time-varying channels where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/8$.

Figure 4.11: The values of $E\{\rho_N[t]\}$ achieved using Algorithm 4 for $N = 100$, $K = 3$ and $B = 1, 2, 3$ and using the existing algorithms for time-varying channels where the drift process described in Section 4.1 is used with a parameter $\Delta_{\text{max}} = \pi/35$. 
It is evident from the simulations that the proposed algorithms significantly outperform the existing algorithms in [24], [25] and [34].
Chapter 5

Conclusion

In this thesis, we presented noncoherent detection schemes for multi-user/multi-node communication systems. The main goal has been to design detection schemes where neither the transmitters nor the receiver knows the CSI in order to avoid the cost of performing channel estimation.

Initially, we introduced differential detection schemes for two-user MIMO systems based on orthogonal STBCs. We first proposed schemes with a simple differential encoding and low complexity differential decoding algorithms for users with two transmit antennas based on three different decoding methods. In these methods, we deployed dynamic programming for efficient decoding. Simulation results showed that our differential schemes using the proposed low complexity decoders provide full transmit diversity and good performance. We also presented additional differential decoding schemes based on the three decoding methods that work for any square OSTBC. Although these schemes have a higher decoding complexity, they achieve full diversity and outperform the existing differential schemes. We extended our differential schemes to a system with a receiver with any number of receive antennas. Furthermore, we compared the performance of our differential modulation schemes with that
of the corresponding coherent modulation schemes (using Zero-Forcing and Maximum Likelihood decoding) and other existing differential schemes. The tradeoff for better performance of our differential schemes using Method 2 compared to that of Method 1 is the decoding delay. To the best of our knowledge, the proposed low complexity schemes are the first low complexity differential modulation schemes for multi-user MIMO communication systems.

We also proposed differential detection schemes for asynchronous multi-user MIMO systems based on orthogonal STBCs. We first presented schemes with simple differential encoding and low complexity differential decoding algorithms by performing interference cancelation in time and employing different decoding methods. The decoding complexity of these schemes increases linearly with the number of users. We then presented additional differential decoding schemes that achieve significantly higher coding gains compared to our low complexity schemes. Simulation results show that they also outperform the existing synchronous differential schemes. The proposed schemes work for any square OSTBC, any constant amplitude constellation, any number of users, and a receiver with any number of receive antennas. Similar to the case of a single user, our schemes can be extended to work with other STBCs with higher rates, such as QOSTBCs, through minor changes. Furthermore, we derived conditions under which our schemes provide full diversity. For the cases of two and four transmit antennas, we also provided examples of PSK constellations to achieve full diversity. To the best of our knowledge, the proposed differential modulation schemes are the first differential schemes for asynchronous multi-user communication systems.

Finally, we introduced novel deterministic DBF algorithms using feedback control based on TSVQ. First, we proposed a simple TSVQ-based DBF algorithm for the case of \( N = 2 \) transmitters and static channels. Then, we presented a TSVQ-based DBF algorithm that works for a larger number of transmitters with static channels. In contrast with the existing deterministic DBF algorithms, the proposed TSVQ-based DBF algorithms can feed back more than 1 bit per time slot for faster phase synchronization. We analytically proved that
the phase asynchrony in our TSVQ-based DBF algorithm for $N = 2$ transmitters converges to zero with probability 1. We also proved that the probability that phase synchronization is attained in the other proposed TSVQ-based DBF algorithm goes to 1 as the number of transmitters goes to infinity. Furthermore, the proposed TSVQ-based DBF algorithms were modified to enable them to track time-varying channels without the knowledge of CSI. Simulation results were provided for the performance of our algorithms, and it was shown that our algorithms significantly outperform the existing adaptive DBF algorithms for static and time-varying channels.
Bibliography


Appendices

A Proof of Proposition 2.1

We make use of the following properties in the proof of Proposition 2.1:

**Lemma A.1.** Let $r$ be a real number and $Z_1$, $Z_2$ and $\tilde{Z}_1$, $\tilde{Z}_2$ be two pairs of $2 \times 2$ complex matrices with the structure of the Alamouti matrices and the orthogonal matrices in (2.5), respectively, i.e.

$$Z_i = \begin{pmatrix} z_{i1} & -z_{i2}^* \\ z_{i2} & z_{i1}^* \end{pmatrix}, \quad \tilde{Z}_i = \begin{pmatrix} \tilde{z}_{i1} & \tilde{z}_{i2} \\ \tilde{z}_{i2}^* & -\tilde{z}_{i1}^* \end{pmatrix}, \quad i = 1, 2$$ (A.1)

where $z_{ij}, \tilde{z}_{ij}$ are complex numbers for $i, j \in \{1, 2\}$. Then $Z_1 \cdot Z_2$, $\tilde{Z}_1 \cdot \tilde{Z}_2$, $Z_1 + Z_2$, $r \cdot Z_1$, and $Z_1^\dagger$ have the structure of the Alamouti matrices, while $\tilde{Z}_1 \cdot Z_2$, $Z_1 \cdot \tilde{Z}_2$, $\tilde{Z}_1 + \tilde{Z}_2$, $r \cdot \tilde{Z}_1$, and $\tilde{Z}_1^\dagger$ have the structure of the orthogonal matrices in (2.5).

**Proof.** The results follow from simple algebra. \qed

As mentioned in Section 2.2.2, the proof of Proposition 2.1 is divided into two steps, namely interference cancelation and channel cancelation:

**Step I:** In the first step, we perform interference cancelation by removing the effect of User 2. Since $G_m$, $m = 1, 2$, is an orthogonal matrix, we have $G_m^\dagger \cdot G_m = |\det(G_m)| \cdot I_2$. Thus, by multiplying the two sides of (2.4) by $|\det(G_m)|^{-1} \cdot G_m^\dagger$ from the left, for $m = 1, 2$ we
obtain

\[
|\text{det}(G_m)|^{-1} \cdot G_m \dagger Y_m^l = |\text{det}(G_m)|^{-1} \cdot G_m \dagger H_m C^l + S^l + |\text{det}(G_m)|^{-1} \cdot G_m \dagger N_m^l. \tag{A.2}
\]

Note that by subtracting the two sides of (A.2) for receive antenna one, i.e., \( m = 1 \), from the two sides of (A.2) for receive antenna two, i.e., \( m = 2 \), we can eliminate the transmitted codeword by User 2, \( S^l \), from the equation. This yields

\[
|\text{det}(G_2)|^{-1} \cdot G_2 \dagger Y_2^l - |\text{det}(G_1)|^{-1} \cdot G_1 \dagger Y_1^l = FC^l + \left( |\text{det}(G_2)|^{-1} \cdot G_2 \dagger N_2^l - |\text{det}(G_1)|^{-1} \cdot G_1 \dagger N_1^l \right). \tag{A.3}
\]

After canceling the interference of User 2, the channel matrices \( G_1 \) and \( G_2 \), which are unknown at both the transmitter and the receiver, are still present in (A.3). In the next step, we remove the channel matrices from the equation to obtain a relationship between the received signals and the transmitted data matrices for User 1.

**Step II:** Since \( Y_m^l, m = 1, 2 \), are orthogonal matrices, we have \( Y_m^l \dagger \cdot Y_m^l = |\text{det}(Y_m^l)| \cdot I_2 \). Thus, by multiplying the two sides of (A.3) by \( |\text{det}(Y_1^l)|^{-1} \cdot Y_1^l \dagger \) from the right we get

\[
|\text{det}(G_2 Y_1^l)|^{-1} \cdot G_2 \dagger Y_2^l Y_1^l \dagger - |\text{det}(G_1)|^{-1} \cdot G_1 \dagger = |\text{det}(Y_1^l)|^{-1} \cdot FC^l Y_1^l \dagger + |\text{det}(Y_1^l)|^{-1} \cdot \left( |\text{det}(G_2)|^{-1} \cdot G_2 \dagger N_2^l - |\text{det}(G_1)|^{-1} \cdot G_1 \dagger N_1^l \right) Y_1^l \dagger. \tag{A.4}
\]

Now consider Eq. (A.4) for time blocks \( l \) and \( l-1 \). By subtracting the equation for time block \( l-1 \) from the equation for time block \( l \), we can eliminate the unknown term \( |\text{det}(G_1)|^{-1} \cdot G_1 \dagger \) from the left side of the equation. This results in

\[
|\text{det}(G_2)|^{-1} \cdot G_2 \dagger \Gamma^l = F \left( |\text{det}(Y_1^l)|^{-1} \cdot C^l Y_1^l \dagger - |\text{det}(Y_1^{l-1})|^{-1} \cdot C^{l-1} Y_1^{l-1} \dagger \right) + \tilde{N}^l \tag{A.5}
\]

where \( \tilde{N}^l \) is the effective noise matrix. From the results of Lemma A.1, \( F \) and \( \Gamma^l \) have the structure of the Alamouti matrices and thus \( F^\dagger \cdot F = |\text{det}(F)| \cdot I_2 \) and \( \Gamma^l \dagger \cdot \Gamma^l = |\text{det}(\Gamma^l)| \cdot I_2 \). By multiplying the two sides of (A.5) by \( |\text{det}(F)|^{-1} \cdot F^\dagger \) from the left and then by \( |\text{det}(\Gamma^l)|^{-1} \cdot \Gamma^l \dagger \) from the right we obtain

\[
|\text{det}(FG_2)|^{-1} \cdot F^\dagger G_2^\dagger = |\text{det}(\Gamma^l)|^{-1} \cdot \left( |\text{det}(Y_1^l)|^{-1} \cdot C^l Y_1^l \dagger - |\text{det}(Y_1^{l-1})|^{-1} \cdot C^{l-1} Y_1^{l-1} \dagger \right) \Gamma^l \dagger + |\text{det}(F\Gamma^l)|^{-1} \cdot F^\dagger \tilde{N}^l \Gamma^l \dagger. \tag{A.6}
\]
Again consider Eq. (A.6) for time blocks $l$ and $l - 1$. By subtracting the equation for time block $l - 1$ from the equation for time block $l$, we can eliminate the unknown term $|\det(FG_2)|^{-1} \cdot F^\dagger G_2^\dagger$ from the left side of the equation. Therefore, we have

$$0_2 = C^l A_0^l + C^{l-1} A_1^l + C^{l-2} A_2^l + \tilde{N}^l$$

where $\tilde{N}^l$ is the effective noise matrix and $A_0^l, A_1^l, A_2^l$ are given in (2.7). At the end of Step II, noting that $C^{l-2\dagger} \cdot C^{l-2} = I_2$ and that from Eq. (2.2) we have $C^l = C^{l-1} \cdot P^l$ and thus $C^{l-2\dagger} \cdot C^{l-1} = P^{l-1}$ and $C^{l-2\dagger} \cdot C^l = P^{l-1}P^l$, we multiply the two sides of Eq. (A.7) by $C^{l-2\dagger}$ from the left to obtain Eq. (2.6). Since all the terms that we multiplied by the equations throughout the proof were almost surely defined, Eq. (2.6) holds almost surely. Following the above procedure, one may easily obtain $\tilde{N}^l$ in terms of $N_1^{l-i}, N_2^{l-i}$, $i = 0, 1, 2$, as given in Eq. (2.8).

### B Noise Power Approximation

We will approximate $|\det(\Psi_2)|$ using Eq. (A.6). Note that the left hand side of Eq. (A.6) is equal to $\Psi_2$. Then ignoring the noise term on the right hand side of Eq. (A.6) at high SNRs we have

$$|\det(\Psi_2)| \approx |\det(F^\dagger)|^{-1} \cdot |\det\left(|\det(Y_1^l)|^{-1} \cdot C^l Y_1^l \dagger - |\det(Y_1^{l-1})|^{-1} \cdot C^{l-1} Y_1^{l-1\dagger}\right)|.$$  \hspace{1cm} (B.8)

Again from the results of Lemma A.1, $|\det(Y_1^l)|^{-1} \cdot C^l Y_1^l \dagger - |\det(Y_1^{l-1})|^{-1} \cdot C^{l-1} Y_1^{l-1\dagger}$ also has the structure of the orthogonal matrices in Eq. (2.5), and thus we can use the Frobenius norm to derive

$$\left|\det\left(|\det(Y_1^l)|^{-1} \cdot C^l Y_1^l \dagger - |\det(Y_1^{l-1})|^{-1} \cdot C^{l-1} Y_1^{l-1\dagger}\right)\right|$$

$$= \frac{1}{2} \left|\left|\det(Y_1^l)|^{-1} \cdot C^l Y_1^l \dagger - |\det(Y_1^{l-1})|^{-1} \cdot C^{l-1} Y_1^{l-1\dagger}\right|\right|^2_{F}$$

$$= |\det(Y_1^l)|^{-1} + |\det(Y_1^{l-1})|^{-1} - |\det(Y_1^{l-1\dagger})|^{-1} \cdot \operatorname{Tr}\left\{Y_1^l P^{l\dagger} Y_1^{l-1\dagger}\right\}$$

where the last equality is easily obtained from the definition of the Frobenius norm of a matrix $Z$ given by $\|Z\|_{F}^2 = \operatorname{Tr}\{Z^\dagger \cdot Z\}$. The expected value of $Y_1^l P^{l\dagger} Y_1^{l-1\dagger}$ over all possible
data matrices $P^l$ is the zero matrix $0_2$ since the expected value of $P^l\dagger$ is $0_2$ due to the symmetry of the constellation. Therefore, we will replace the last term in (B.9) with $0_2$ and substitute the result back into (B.8) to obtain an approximation for $|\det (\Psi_2)|$ as given in (2.11).

Now in order to approximate $|\det (\Psi_1)|$, we note that to obtain Eq. (A.6), we multiplied the two sides of Eq. (A.3) by $|\det(Y^i)\dagger|\cdot Y^i\dagger$ from the right, then removed the unknown term $|\det(G_2)|\cdot G_2\dagger$ in the next step, and finally multiplied the two sides of Eq. (A.5) by $|\det(F)^i|\cdot F^i$ from the left and then by $|\det(\Gamma^i)|\cdot \Gamma^i\dagger$ from the right to achieve (A.6). We may instead multiply the two sides of Eq. (A.3) by $|\det(Y^2)\dagger|\cdot Y^2\dagger$ from the right, then remove the unknown term $|\det(G_2)|\cdot G_2\dagger$ in the next step, and finally multiply the two sides of the resulting equation by $|\det(F)^i|\cdot F^i$ from the left and then by $|\det(\Gamma^i)|\cdot \Gamma^i\dagger$ from the right where $\tilde{\Gamma}^i$ is given in Eq. (2.12). Then proceeding similarly to the above argument, we can obtain an approximation for $|\det (\Psi_1)|$ given by Eq. (2.10).

### C Proof of Proposition 3.1

Using the input-output relationship in (3.8) and (3.12), we can write the input-output relationship for a single time block $l > 0$ as

$$
(y^i_{1,J}, \ldots, y^i_{T,1}, \ldots, y^i_{T,J}) = \sum_{i=1}^{J} H_i \left( S^i_{s^{-1}}, S^i_t \right) \left( Z^i_{s,0} \right) + \left( n^i_{1,J}, \ldots, n^i_{T,1}, \ldots, n^i_{T,J} \right)
$$

where $Z^i_{s,0}, Z^i_{s,1}, i = 1, \ldots, J$, are $T \times TJ$ matrices given by

$$
Z^i_{s,0} = \begin{pmatrix}
0 & \cdots & 0 & 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 1 & \cdots & 1 \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0
\end{pmatrix},
Z^i_{s,1} = \begin{pmatrix}
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 \\
0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0
\end{pmatrix}.
$$
Then, note that the interference of all users on User $j$ can be canceled by subtracting $y_{t,j-1}^j$ from $y_{t,j}^j$ for $t = 1, \cdots, T$ as follows

$$
(y_{1,j}^j - y_{1,j-1}^j, \cdots, y_{T,j}^j - y_{T,j-1}^j) = H_j \left( S_j^{t-1}, S_j^t \right) \left( \bar{Z}_0 \right) + \left( n_{1,j}^j - n_{1,j-1}^j, \cdots, n_{T,j}^j - n_{T,j-1}^j \right)
$$

(C.12)

where $\bar{Z}_0, \bar{Z}_1$ are $T \times T$ matrices given by

$$
\bar{Z}_0 = 
\begin{pmatrix}
1 & -1 & 0 & \cdots & 0 & 0 \\
0 & 1 & -1 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -1 & 0 \\
0 & 0 & 0 & \cdots & 1 & -1 \\
0 & 0 & 0 & \cdots & 0 & 1
\end{pmatrix}
\quad 
\bar{Z}_1 = 
\begin{pmatrix}
0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 \\
-1 & 0 & 0 & \cdots & 0 & 0
\end{pmatrix}
$$

(C.13)

Considering (C.12) for more consecutive time slots and using simple algebra, one may easily show that

$$
\bar{Y}_j^t = H_j \cdot \left( S_j^{t-2}, S_j^{t-1}, S_j^t \right) \cdot \bar{A} + \bar{N}_j^t = H_j S_j^{t-2} U_j^t \bar{A} + \bar{N}_j^t.
$$

(C.14)

## D Proof of Theorem 3.1

In the ITIC decoder using Method 0, we used the relationship in (3.21) and performed noncoherent ML detection. In (3.21), $H_j S_j^{t-2}, U_j^t \bar{A}$, and $\bar{N}_j^t$ can be considered as the equivalent channel, signal, and noise terms, respectively. Note that the entries of $H_j S_j^{t-2}$ and $\bar{N}_j^t$ are samples of independent zero-mean complex Gaussian random variables. With a small abuse of the notation, let $U_{j,1}^t = (I_N, P_{j,1}^{t-1}, P_{j,1}^{t-1} P_{j,1}^t), U_{j,2}^t = (I_N, P_{j,2}^{t-1}, P_{j,2}^{t-1} P_{j,2}^t)$ for some arbitrary data matrices $P_{j,1}^{t-1}, P_{j,1}^t, P_{j,2}^{t-1}, P_{j,2}^t$ such that $U_{j,1}^t \neq U_{j,2}^t$. Then, in order to prove that the ITIC decoder using Method 0 achieves a diversity order of $MN$, by Proposition
4 of [40], it suffices to show that for any \( \mathbf{U}_{j,1}^l \neq \mathbf{U}_{j,2}^l \), the following has full row rank\(^1\):

\[
\begin{pmatrix}
\mathbf{U}_{j,1}^l \cdot \bar{\mathbf{A}} \\
\mathbf{U}_{j,2}^l \cdot \bar{\mathbf{A}}
\end{pmatrix} = \begin{pmatrix}
\mathbf{I}_N & \mathbf{P}^{-1}_{j,1} \mathbf{P}^{-1}_{j,1} \\
\mathbf{I}_N & \mathbf{P}^{-1}_{j,2} \mathbf{P}^{-1}_{j,2}
\end{pmatrix} \cdot \bar{\mathbf{A}} = \mathbf{G}(\mathbf{P}_{j,1}^{-1}, \mathbf{P}_{j,1}^l, \mathbf{P}_{j,2}^{-1}, \mathbf{P}_{j,2}^l).
\] (D.15)

By our assumption, \( \mathbf{G}(\mathbf{P}_{j,1}^{-1}, \mathbf{P}_{j,1}^l, \mathbf{P}_{j,2}^{-1}, \mathbf{P}_{j,2}^l) \) has full row rank when \( (\mathbf{P}_{j,1}^{-1}, \mathbf{P}_{j,1}^l) \neq (\mathbf{P}_{j,2}^{-1}, \mathbf{P}_{j,2}^l) \) (or equivalently, \( \mathbf{U}_{j,1}^l \neq \mathbf{U}_{j,2}^l \)). Thus, the ITIC decoder using Method 0 provides full diversity.

Now, note that the MMPL decoder using Method 0 is optimal among the decoders using the same set of (or a subset of) the time partitions it uses. Since the ITIC decoder using Method 0 uses a subset of the time partitions the MMPL decoder using Method 0 uses, the MMPL decoder using Method 0 must perform at least as good as the ITIC decoder using Method 0. Thus, the MMPL decoder using Method 0 must achieve full diversity as well.

### E Proof of Theorem 3.3

We need the following property to prove the theorem:

**Lemma E.2.** Let \( \mathbf{X}_1, \mathbf{X}_2 \) be distinct \( N \times N \) matrices such that \( (\mathbf{X}_2 - \mathbf{X}_1)^\dagger \cdot (\mathbf{X}_2 - \mathbf{X}_1) = \frac{\|\mathbf{X}_2 - \mathbf{X}_1\|^2}{N} \cdot \mathbf{I}_N \). Then,

\[
\begin{pmatrix}
\mathbf{I}_N & \mathbf{X}_1 \\
\mathbf{I}_N & \mathbf{X}_2
\end{pmatrix}^{-1} = \begin{pmatrix}
\mathbf{I}_N + \mathbf{X}_1 \bar{\mathbf{X}} - \mathbf{X}_1 \bar{\mathbf{X}} \\
-\bar{\mathbf{X}} & \bar{\mathbf{X}}
\end{pmatrix}
\] (E.16)

where \( \bar{\mathbf{X}} = \frac{\mathbf{N}(\mathbf{X}_2 - \mathbf{X}_1)^\dagger}{\|\mathbf{X}_2 - \mathbf{X}_1\|^2} \).

**Proof.** The result can be easily proven by showing that

\[
\begin{pmatrix}
\mathbf{I}_N + \mathbf{X}_1 \bar{\mathbf{X}} - \mathbf{X}_1 \bar{\mathbf{X}} \\
-\bar{\mathbf{X}} & \bar{\mathbf{X}}
\end{pmatrix} \cdot \begin{pmatrix}
\mathbf{I}_N & \mathbf{X}_1 \\
\mathbf{I}_N & \mathbf{X}_2
\end{pmatrix} = \begin{pmatrix}
\mathbf{I}_N & \mathbf{0}_N \\
\mathbf{0}_N & \mathbf{I}_N
\end{pmatrix} = \mathbf{I}_{2N}.
\] (E.17)

---

\(^2\)The channel model used in [40] is the transposed version of ours. We have modified their results based on our channel model. We have also used the fact that \( \text{rank}(\mathbf{X}^\dagger \mathbf{X}) = \text{rank}(\mathbf{X}) \) for any matrix \( \mathbf{X} \) with complex elements.
To prove Theorem 3.3, we consider two cases:

**Case 1:** We first consider the case when \( P_1 \neq P_3 \). Since \( P_1, P_3 \) are constructed using the same OSTBC and thus \((P_3-P_1)^\dagger \cdot (P_3-P_1) = \frac{\|P_3-P_1\|_F^2}{N} \cdot I_N\), by Lemma E.2, \( \begin{pmatrix} I_N & P_1 \\ I_N & P_3 \end{pmatrix} \) is invertible. Also, since its inverse must be a full rank matrix, multiplying its inverse by \( G(P_1, P_2, P_3, P_4) \) must result in a matrix with the same rank as \( G(P_1, P_2, P_3, P_4) \). Therefore, using Lemma E.2 and the definition of \( G(P_1, P_2, P_3, P_4) \) in (3.38), by multiplying \( G(P_1, P_2, P_3, P_4) \) by \( \begin{pmatrix} I_N & P_1 \\ I_N & P_3 \end{pmatrix}^{-1} \) from the left we obtain

\[
\begin{pmatrix} I_N & P_1 \\ I_N & P_3 \end{pmatrix}^{-1} \cdot G(P_1, P_2, P_3, P_4) = \begin{pmatrix} I_N + P_1 \left( \frac{N(P_3-P_1)^\dagger}{\|P_3-P_1\|_F^2} \right) & -P_1 \left( \frac{N(P_3-P_1)^\dagger}{\|P_3-P_1\|_F^2} \right) \\ -P_1 \left( \frac{N(P_3-P_1)^\dagger}{\|P_3-P_1\|_F^2} \right) & I_N \\ 0_N & N \left( \frac{\|P_3-P_1\|_F^2}{\|P_3-P_1\|_F^2} \right) \cdot (P_3P_4 - P_1P_2) \end{pmatrix} \cdot \tilde{A}
\]

which must be of the same rank as \( G(P_1, P_2, P_3, P_4) \). Now, let \( B_1 \) and \( B_1^{-1} \) be \( 3N-1 \times 3N-1 \) matrices given by

\[
B_1 = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \quad B_1^{-1} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}
\]

Note that \( B_1^{-1} \) is the inverse of \( B_1 \). Again, since \( B_1^{-1} \) is a full rank matrix, multiplying it by (E.18) will result in a matrix with the same rank as (E.18). Therefore, multiplying (E.18) by \( B_1^{-1} \) from the right yields a matrix with the same rank as \( G(P_1, P_2, P_3, P_4) \), given by

\[
\begin{pmatrix} I_N & P_1 \\ I_N & P_3 \end{pmatrix}^{-1} \cdot G(P_1, P_2, P_3, P_4) \cdot B_1^{-1} = \begin{pmatrix} I_N & 0_N & P_1P_2 - NP_1 \cdot \left( \frac{P_3-P_1}{\|P_3-P_1\|_F} \right) \cdot (P_3P_4 - P_1P_2) \\ 0_N & I_N & N \left( \frac{\|P_3-P_1\|_F^2}{\|P_3-P_1\|_F^2} \right) \cdot (P_3P_4 - P_1P_2) \end{pmatrix} \cdot B_2
\]
where $B_2$ is the $3N \times 3N - 1$ matrix

$$B_2 = \tilde{A} \cdot B_1^{-1} = \begin{pmatrix}
-1 & -1 & -1 & \cdots & -1 & -1 & -1 \\
1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 1 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & 1
\end{pmatrix}.$$  \hfill (E.21)

Now, consider the RHS of (E.20) and let

$$\begin{pmatrix}
P_1 P_2 - \hat{N} P_1 : \left( \frac{(P_3 - P_3)^\dagger}{\|P_3 - P_1\|_F} \right) \cdot (P_3 P_4 - P_1 P_2) \\
N \left( \frac{(P_3 - P_3)^\dagger}{\|P_3 - P_1\|_F} \right) \cdot (P_3 P_4 - P_1 P_2)
\end{pmatrix} = \begin{pmatrix}
\beta_{1,1} & \beta_{1,2} & \beta_{1,3} & \cdots & \beta_{1,N} \\
\beta_{2,1} & \beta_{2,2} & \beta_{2,3} & \cdots & \beta_{2,N} \\
\beta_{3,1} & \beta_{3,2} & \beta_{3,3} & \cdots & \beta_{3,N} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\beta_{2N,1} & \beta_{2N,2} & \beta_{2N,3} & \cdots & \beta_{2N,N}
\end{pmatrix}. \hfill (E.22)
$$

By plugging (E.22) into (E.20) and using simple algebra, we can write (E.20) as

$$\begin{pmatrix}
I_N & P_1 \\
I_N & P_3
\end{pmatrix}^{-1} \cdot G(P_1, P_2, P_3, P_4) \cdot B_1^{-1} = \begin{pmatrix}
\beta_{1,1} - 1 & \beta_{1,2} - 1 & \beta_{1,3} - 1 & \cdots & \beta_{1,N} - 1 \\
1 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
\end{pmatrix}.$$  \hfill (E.23)

Let $r_i, i = 1, \cdots, 2N$, denote the $i$th row of (E.23). Then, the linear combination of $r_1, \cdots, r_{2N}$ with coefficients $\lambda_1, \lambda_2, \cdots, \lambda_{2N}$, which are not all zero, is given by

$$r = \sum_{i=1}^{2N} \lambda_i r_i = \left( \lambda_2 - \lambda_1, \cdots, \lambda_{2N} - \lambda_1, -\lambda_1 + \sum_{i=1}^{2N} \lambda_i \beta_{i,1}, \cdots, -\lambda_1 + \sum_{i=1}^{2N} \lambda_i \beta_{i,N} \right). \hfill (E.24)$$

Note that $r$ is equal to the zero vector if and only if $\lambda_1 = \lambda_2 = \cdots = \lambda_{2N}$ and $\sum_{i=1}^{2N} \beta_{i,1} = \sum_{i=1}^{2N} \beta_{i,2} = \cdots = \sum_{i=1}^{2N} \beta_{i,N} = 1$. This means that the rows of (E.23) are linearly dependent if and only if $\sum_{i=1}^{2N} \beta_{i,1} = \sum_{i=1}^{2N} \beta_{i,2} = \cdots = \sum_{i=1}^{2N} \beta_{i,N} = 1$. Using (E.22), this implies that (E.23), and thus $G(P_1, P_2, P_3, P_4)$, has full row rank if and only if

$$\begin{pmatrix}
P_1 P_2 - \hat{N} P_1 : \left( \frac{(P_3 - P_3)^\dagger}{\|P_3 - P_1\|_F} \right) \cdot (P_3 P_4 - P_1 P_2) \\
N \left( \frac{(P_3 - P_3)^\dagger}{\|P_3 - P_1\|_F} \right) \cdot (P_3 P_4 - P_1 P_2)
\end{pmatrix} \neq (1, 1, \cdots, 1). \hfill (E.25)$$
Then, it is easy to see that (E.25) holds, and thus $G(P_1, P_2, P_3, P_4)$ has full row rank, for any possible data matrices $P_1, P_2, P_3, P_4$ with $P_1 \neq P_3$ if and only if (3.39) holds for any possible data matrices $\tilde{P}_1, \tilde{P}_2, \tilde{P}_3, \tilde{P}_4$ with $\tilde{P}_1 \neq \tilde{P}_3$. This means that (3.39) is a necessary and sufficient condition for $G(P_1, P_2, P_3, P_4)$ to have full row rank in Case 1.

**Case 2:** We now consider the case when $P_1 = P_3$. Since $(P_1, P_2) \neq (P_3, P_4)$, this implies that $P_2 \neq P_4$. Also, since $P_2, P_4$ are constructed using the same OSTBC and thus $[P_1(P_4 - P_2)]^\dagger \cdot [P_1(P_4 - P_2)] = \|P_1(P_4 - P_2)\|_F^2 N \cdot I_N$, by Lemma E.2, $\begin{pmatrix} I_N & P_1 P_2 \\ I_N & P_1 P_4 \end{pmatrix}$ is invertible. Again, since its inverse must be a full rank matrix, multiplying its inverse by $G(P_1, P_2, P_3, P_4)$ must result in a matrix with the same rank as $G(P_1, P_2, P_3, P_4)$. Therefore, by multiplying $G(P_1, P_2, P_3, P_4)$ by $\begin{pmatrix} I_N & P_1 P_2 \\ I_N & P_1 P_4 \end{pmatrix}^{-1}$ from the left we obtain

\[
\begin{pmatrix} I_N & P_1 P_2 \\ I_N & P_1 P_4 \end{pmatrix}^{-1} \cdot G(P_1, P_2, P_3, P_4)
= \begin{pmatrix} I_N + P_1 P_2 \left( \frac{N[|P_1(P_4 - P_2)|]}{\|P_1(P_4 - P_2)\|_F^2} \right) - P_1 P_2 \left( \frac{N[|P_1(P_4 - P_2)|]}{\|P_1(P_4 - P_2)\|_F^2} \right) \right) \cdot \begin{pmatrix} I_N & P_1 P_2 \\ I_N & P_1 P_4 \end{pmatrix} \cdot A
= \begin{pmatrix} I_N & P_1 & 0_N \\ 0_N & 0_N & I_N \end{pmatrix} \cdot A,
\]

(E.26)

which must be of the same rank as $G(P_1, P_2, P_3, P_4)$. Once again, since $B_1^{-1}$ is a full rank matrix, multiplying (E.26) by $B_1^{-1}$ from the right yields a matrix with the same rank as (E.26), and thus $G(P_1, P_2, P_3, P_4)$, given by

\[
\begin{pmatrix} I_N & P_1 P_2 \\ I_N & P_1 P_4 \end{pmatrix}^{-1} \cdot G(P_1, P_2, P_3, P_4) \cdot B_1^{-1} = \begin{pmatrix} I_N & P_1 & 0_N \\ 0_N & 0_N & I_N \end{pmatrix} \cdot B_2.
\]

(E.27)

Then proceeding similarly to the procedure described in (E.22)-(E.25) for Case 1, we find that $G(P_1, P_2, P_3, P_4)$ has full row rank if and only if $w \cdot P_1 \neq w$. Note that this condition is a special case of (3.39) when $\tilde{P}_2 = \tilde{P}_4 = P_1$. Therefore, (3.39) is a sufficient condition for $G(P_1, P_2, P_3, P_4)$ to have full row rank in Case 2. Also, we showed that (3.39) is a necessary and sufficient condition for $G(P_1, P_2, P_3, P_4)$ to have full row rank in Case 1. Thus, (3.39) is a necessary and sufficient condition in the general case for $G(P_1, P_2, P_3, P_4)$ to have full row rank for any possible data matrices $P_1, P_2, P_3, P_4$ with $(P_1, P_2) \neq (P_3, P_4)$. 101
**F  Proof of Theorem 4.2**

Let us consider a Cartesian coordinate system where the x-axis is along the vector of the second node, and suppose that $\tilde{h}_1, \tilde{h}_2$ denote the vectors corresponding to Nodes 1 and 2, respectively, in this coordinate system. We have

$$\text{RSS}_2\{K\} = |\tilde{h}_1 + \tilde{h}_2| = \sqrt{[\text{Re}\{\tilde{h}_1 + \tilde{h}_2\}]^2 + [\text{Im}\{\tilde{h}_1 + \tilde{h}_2\}]^2} \quad \text{(F.28)}$$

$$\geq |\text{Re}\{\tilde{h}_1 + \tilde{h}_2\}|.$$

Note that $\text{Re}\{\tilde{h}_2\} = |\tilde{h}_2|$ is a Rayleigh random variable with parameter $\sigma = \sqrt{2}/2$ and $\text{Re}\{\tilde{h}_1\} = |\tilde{h}_1| \cdot \cos(\angle \tilde{h}_1)$ is the x-coordinate of a random vector whose angle, $\angle \tilde{h}_1$, and length, $|\tilde{h}_1|$, are independent and have a uniform distribution over $(-\pi/2^B(K-2)+1, \pi/2^B(K-2)+1]$ and a Rayleigh distribution with parameter $\sigma = \sqrt{2}/2$, respectively. The uniform distribution of $\angle \tilde{h}_1$ in $(-\pi/2^B(K-2)+1, \pi/2^B(K-2)+1]$ follows directly from Theorem 4.1 and noting that the phases of the two nodes in Stage $K \geq 3$ (i.e., at transmission time) are the same as the phases of the two nodes at the end of Stage $K - 1$ (i.e., after phase adjustment). Therefore, using (F.28) we have

$$\mathbb{E}[\rho^2\{K\}] = \mathbb{E}\left[\frac{\text{RSS}_2\{K\}}{\text{RSS}_2,\text{max}}\right]$$

$$\geq \mathbb{E}\left[\frac{|\tilde{h}_2| + |\tilde{h}_1| \cdot \cos \angle \tilde{h}_1}{|\tilde{h}_1| + |\tilde{h}_2|}\right]. \quad \text{(F.29)}$$

Also, since $|\tilde{h}_1|$, $|\tilde{h}_2|$, $\angle \tilde{h}_1$ are independent and $|\tilde{h}_1|$, $|\tilde{h}_2|$ have the same distribution, we can switch $|\tilde{h}_1|$ and $|\tilde{h}_2|$ on the RHS of the inequality in (F.29) to obtain

$$\mathbb{E}[\rho^2\{K\}] \geq \mathbb{E}\left[\frac{|\tilde{h}_1| + |\tilde{h}_2| \cdot \cos \angle \tilde{h}_1}{|\tilde{h}_2| + |\tilde{h}_1|}\right]. \quad \text{(F.30)}$$
Then, we can take the average of the RHSs of the inequalities in (F.29) and (F.30) to attain a lower bound for $E[\rho_2\{K\}]$ as follows

$$E[\rho_2\{K\}] \geq \frac{1}{2} E\left[ \frac{(|\tilde{h}_1|+|\tilde{h}_2|)(1+\cos \tilde{h}_1)}{|\tilde{h}_1|+|\tilde{h}_2|} \right]$$

$$= \frac{1}{2} \left[ 1 + \int_{\frac{2B(K-2)}{\pi}}^{\frac{2B(K-2)+1}{\pi}} \cos x dx \right]$$

$$= \frac{1}{2} \left[ 1 + \frac{2B(K-2)+1}{\pi} \sin \left( \frac{\pi}{2B(K-2)+1} \right) \right].$$

(F.31)

This completes the proof.

\section{Proof of Proposition 4.2}

We start with some useful lemmas that we use throughout the proof:

\begin{lemma}
Let $\Theta_1, \ldots, \Theta_n$ be independent random variables with a uniform distribution over $(-\pi, \pi]$, and let $S_n = (\sum_{i=1}^{n} \Theta_i) \mod 2\pi$. Then, $S_n$ is uniformly distributed over $(-\pi, \pi]$.
\end{lemma}

\begin{proof}
We prove the result by induction on $n$. For $n = 1$, the result is trivial. For $n = 2$, the pdf of $\Theta_1 + \Theta_2$ is given by

$$f_{\Theta_1+\Theta_2}(x) = \int_{-\infty}^{\infty} f_{\Theta_1}(x-y)f_{\Theta_2}(y)dy = \frac{1}{2\pi} \int_{-\pi}^{\pi} f_{\Theta_1}(x-y)dy = \begin{cases} \frac{2\pi-x}{4\pi^2}, & 0 \leq x \leq 2\pi \\ \frac{2\pi+x}{4\pi^2}, & -2\pi \leq x < 0 \\ 0, & \text{otherwise} \end{cases} \quad (G.32)$$

103
Then, we can obtain the cdf of $S_2$ as follows

$$F_{S_2}(x) = P((\Theta_1 + \Theta_2) \mod 2\pi \leq x) = \begin{cases} 1, & x > \pi \\ \frac{\pi + x}{2\pi}, & |x| \leq \pi \\ 0, & \text{otherwise} \end{cases} \quad (G.33)$$

By differentiating the cdf of $S_2$, we can find that the pdf of $S_2$ is given by

$$f_{S_2}(x) = \begin{cases} \frac{1}{2\pi}, & |x| \leq \pi \\ 0, & \text{otherwise} \end{cases} \quad (G.34)$$

Now, assume $S_n$ is uniformly distributed over $(-\pi, \pi]$ for some $n \geq 2$. Then we have

$$f_{S_{n+1}}(x) = \left(\sum_{i=1}^{n+1} \Theta_i\right) \mod 2\pi = (\Theta_{n+1} + S_n) \mod \pi \quad (G.35)$$

Invoking the inductive assumption and using the same argument as above for $n = 2$, we find that $S_{n+1}$ is also uniformly distributed over $(-\pi, \pi]$. This ends the proof. \qed

**Lemma G.4.** Let $R$ and $\Theta$ be independent random variables that have a Rayleigh distribution with parameter $\sigma$ and a uniform distribution over $(-\pi, \pi]$, respectively, and assume $X = R \cos \Theta$ and $Y = R \sin \Theta$. Then, $X$ and $Y$ are independent normal random variables with mean 0 and variance $\sigma^2$.

**Proof.** Since $R$ and $\Theta$ are independent, their joint pdf is given by

$$f_{R,\Theta}(r, \theta) = f_R(r)f_\Theta(\theta) = \begin{cases} \frac{r}{2\pi\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right), & r \in [0, \infty), \theta \in (-\pi, \pi] \\ 0, & \text{otherwise} \end{cases} \quad (G.36)$$

Using the Jacobian technique for transformation of two random variables and simple algebra,
it can be easily shown that

\[ f_{X,Y}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \] (G.37)

Hence, the marginal densities of \( X \) and \( Y \) can be found as

\[ f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y) dy = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{x^2}{2\sigma^2}\right), \] (G.38)

\[ f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x, y) dx = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{y^2}{2\sigma^2}\right). \]

Thus, \( X \) and \( Y \) are independent normal random variables. \( \square \)

**Lemma G.5.** Let \( \{E_{1,j}\}_{j=1}^{\infty}, \cdots, \{E_{n,j}\}_{j=1}^{\infty} \) be \( n \) possibly dependent sequences of infinite events, and suppose that \( \lim_{j \to \infty} P(E_{i,j}) = 1 \) for \( i = 1, \cdots, n \). Then

\[ \lim_{j \to \infty} P \left( \bigcap_{i=1}^{n} E_{i,j} \right) = 1. \] (G.39)

**Proof.** We prove the result by induction on \( n \). For \( n = 1 \), the result is trivial. Now, let us assume \( \lim_{j \to \infty} P \left( \bigcap_{i=1}^{k} E_{i,j} \right) = 1 \) for some \( k \geq 1 \). We have

\[ \lim_{j \to \infty} P \left( \bigcap_{i=1}^{k+1} E_{i,j} \right) = \lim_{j \to \infty} P \left[ \bigcap_{i=1}^{k} E_{i,j} \right] \cap E_{k+1,j} \]

\[ = \lim_{j \to \infty} P \left( \bigcap_{i=1}^{k} E_{i,j} \right) + \lim_{j \to \infty} P(E_{k+1,j}) - \lim_{j \to \infty} P \left( \bigcup_{i=1}^{k} E_{i,j} \right) \cup E_{k+1,j} \] (G.40)

where the last equality is obtained by applying the relationship \( P(X_1 \cap X_2) = P(X_1) + P(X_2) - P(X_1 \cup X_2) \) for any events \( X_1, X_2 \). Now, consider the RHS of the last equality in (G.40). By invoking the inductive assumption, the first limit must be equal to 1. The second limit is also equal to 1 by the assumptions. Finally, noting that the last limit cannot
be greater than 1, we attain

\[
\lim_{j \to \infty} P \left( \bigcap_{i=1}^{k+1} E_{i,j} \right) \geq 1 + 1 - 1 = 1. \tag{G.41}
\]

Since \( \lim_{j \to \infty} P \left( \bigcap_{i=1}^{k+1} E_{i,j} \right) \) cannot be greater than 1, using (G.41) we conclude that it is equal to 1. This completes the proof. \( \square \)

The proof of Proposition 4.2 is divided into three steps:

**Step I:** In the first step, we show that for any function \( g_1 : \mathbb{R}^+ \to \mathbb{R}^+ \) with \( g_1(t) \in o(t^{1/2}) \), we have

\[
\lim_{N \to \infty} P \{ \text{RSS}_N[0] > g_1(N) \} = 1. \tag{G.42}
\]

At time \( t = 0 \), the phase component adjusted by any Node \( n \), \( \phi_n[0] \), is 0, and thus \( \theta_n[0] = \psi_n[0] + \gamma_n \). Note that \( \psi_n[0] \) and \( \gamma_n \) are independent random variables with a uniform distribution over \((-\pi, \pi]\). Therefore, by Lemma G.3, \( \theta_n[0] \mod 2\pi \) is uniformly distributed over \((-\pi, \pi]\) as well. Now, let \( X_n = \text{Re}\{h_n[0]\} = a_n \cos \theta_n[0] \) and \( Y_n = \text{Im}\{h_n[0]\} = a_n \sin \theta_n[0] \) denote the projections of \( h_n[0] \) over the x-axis and y-axis, respectively. Then, we have

\[
\text{RSS}_N[0] = \left| \sum_{n=1}^{N} h_n[0] \right| = \sqrt{\left( \sum_{n=1}^{N} X_n \right)^2 + \left( \sum_{n=1}^{N} Y_n \right)^2}. \tag{G.43}
\]

Since \( |h_n[0]| = a_n \) has a Rayleigh distribution with parameter \( \sigma = \sqrt{2}/2 \) and is independent of \( \theta_n[0] \), by applying Lemma G.4, we find that \( X_n \) and \( Y_n \) are independent normal random variables with variance \( \sigma^2 = 1/2 \). Therefore, \( \tilde{X}_N = \sum_{n=1}^{N} X_n \sim \mathcal{N}(0, N/2) \) and \( \tilde{Y}_N = \sum_{n=1}^{N} Y_n \sim \mathcal{N}(0, N/2) \). Thus, \( \text{RSS}_N[0] = \sqrt{ (\tilde{X}_N)^2 + (\tilde{Y}_N)^2 } \) has a Rayleigh distribution given by

\[
f_{\text{RSS}_N[0]}(r) = \frac{2r}{N} \cdot \exp \left( -\frac{r^2}{N} \right), \quad r \geq 0. \tag{G.44}
\]
Therefore, using (G.44) we have

\[
\lim_{N \to \infty} P \{ \text{RSS}_N[0] > g_1(N) \} = \lim_{N \to \infty} \int_{g_1(N)}^{\infty} \frac{2r}{N} \exp \left( -\frac{r^2}{N} \right) \, dr \\
= -\lim_{N \to \infty} \exp \left( -\frac{r^2}{N} \right) \bigg|_{g_1(N)}^{\infty} \\
= \lim_{N \to \infty} \exp \left\{ -\left[ \frac{g_1(N)}{N^{1/2}} \right]^2 \right\} .
\]

(G.45)

Since \( g_1(N) \in o(N^{1/2}) \), we have \( \lim_{N \to \infty} \frac{g_1(N)}{N^{1/2}} = 0 \). Thus, using (G.45) we attain (G.42).

**Step II:** Let \( g_2 : \mathbb{R}^+ \to \mathbb{R}^+ \) be a function such that \( \exp \{ [g_2(t)]^2 \} \in \omega(t) \), and let \( F_{n,g_2(N)} \), \( n = 1, \ldots, N \), denote the event that \( a_n < g_2(N) \). In the second step, we show that

\[
\lim_{N \to \infty} P \left( \bigcap_{n=1}^{N} F_{n,g_2(N)} \right) = 1. \tag{G.46}
\]

Since \( a_n, n = 1, \ldots, N \), are independent Rayleigh random variables with parameter \( \sigma = \sqrt{2/2} \), we have

\[
\lim_{N \to \infty} P \left( \bigcap_{n=1}^{N} F_{n,g_2(N)} \right) = \lim_{N \to \infty} \prod_{n=1}^{N} P \left( F_{n,g_2(N)} \right) \\
= \lim_{N \to \infty} \left[ \int_{0}^{g_2(N)} 2r \cdot \exp \left( -r^2 \right) \, dr \right]^N \tag{G.47}
\]

Now, let \( \tilde{g}_2(N) = \exp \{ [g_2(N)]^2 \} \). Then, from (G.47) and noting that \( \lim_{N \to \infty} \tilde{g}_2(N) = \infty \),
we obtain
\[
\lim_{N \to \infty} P \left( \bigcap_{n=1}^{N} F_{n,g_2(N)} \right) = \lim_{N \to \infty} \left[ 1 - \frac{1}{\tilde{g}_2(N)} \right]^N
\]
\[
= \lim_{N \to \infty} \left\{ \left[ 1 - \frac{1}{\tilde{g}_2(N)} \right]^{\frac{N}{\tilde{g}_2(N)}} \right\}^{\frac{N}{\tilde{g}_2(N)}}
\]
\[
= \lim_{N \to \infty} \exp \left[ -\frac{N}{\tilde{g}_2(N)} \right]
\]
where we have used the fact that \( \lim_{t \to \infty} \left( 1 - \frac{1}{t} \right)^t = \exp(-1) \) in the last equality. Since by assumption \( \exp \left\{ [g_2(N)]^2 \right\} \in \omega(N) \), we have \( \lim_{N \to \infty} \frac{\tilde{g}_2(N)}{N} = \infty \). Using this in (G.48), we attain
\[
\lim_{N \to \infty} P \left( \bigcap_{n=1}^{N} F_{n,g_2(N)} \right) = 1,
\]
as desired.

**Step III:** In the final step, we use the results of Steps I and II to prove Proposition 4.2.

Let \( g_1(t) = t^{1/3} \) and \( g_2(t) = t^{1/4} \) in Steps I and II, respectively. Note that \( g_1(t) \in o(t^{1/2}) \) since \( \lim_{t \to \infty} t^{1/3} = 0 \), and \( \exp \left\{ [g_2(t)]^2 \right\} \in \omega(t) \) since \( \lim_{t \to \infty} \frac{\exp(t^{1/2})}{t} = \infty \). By the result of Step I, we have
\[
\lim_{N \to \infty} P \left\{ \text{RSS}_N[0] > N^{1/3} \right\} = 1. \tag{G.49}
\]
Also, by the result of Step II, we have
\[
\lim_{N \to \infty} P \left( \bigcap_{n=1}^{N} F_{n,N^{1/4}} \right) = 1. \tag{G.50}
\]
Then, by applying Lemma G.5, we have
\[
\lim_{N \to \infty} P \left\{ \bigcap_{n=1}^{N} F_{n,N^{1/4}} \cap \tilde{F}_{N^{1/3}} \right\} = 1 \tag{G.51}
\]
where \( \tilde{F}_{N^{1/3}} \) denotes the event that \( \text{RSS}_N[0] > N^{1/3} \). This implies that the probability that for all \( n = 1, \ldots, N \) we have \( \frac{\text{RSS}_N[0]}{\delta_n} > \frac{N^{1/3}}{N^{1/4}} = N^{1/12} \) goes to 1 as \( N \to \infty \). Then, noting that \( \lim_{N \to \infty} N^{1/12} = \infty \), we obtain the result in (4.15).
H Proof Sketch of Theorem 4.3

By Proposition 4.2, the probability that the length of the sum of the initial vectors (RSS$_N[0]$) is arbitrarily larger than the length of all the vectors ($a_n$, $n = 1, \cdots, N$) goes to 1 as $N \to \infty$. Now, for $n = 1, \cdots, N$, let $v_n$ denote the sum of the vectors for all the nodes except for Node $n$ in Stage 0. Then, using Proposition 4.2 and the law of sines, it can be easily shown that for any $\epsilon > 0$, the probability that the angle between $v_n$ and the sum of all the vectors in Stage 0 for all $n = 1, \cdots, N$ is less than $\epsilon$ goes to 1 as $N \to \infty$.

In Stage 1, we first consider the case when $L = 1$ (i.e., when all the phases are updated at the end of Stage 1). Similar to Algorithm 1, it can be easily shown using the law of cosines that from the rotation angles $0$ and $\beta = \pi$ radians applied to the vector for Node $n = 1, \cdots, N$, the one that results in a larger RSS makes the angle between the vector for Node $n$ and $v_n$ smaller. Therefore, that rotation angle is chosen to traverse the tree. Then, using the above claim about the angle between $v_n$ and the sum of the vectors for all the nodes in Stage 0, the phases of the nodes whose vectors are initially in a distance of greater than $\epsilon$ radians from the boundaries of the quantization regions in this stage are moved into a single quantization region of length $\pi$ radians. When $L > 1$, by induction on $L$ and the weak law of large numbers (WLLN), the same result can be shown to hold for the updated phases after each phase update. Then, by applying Lemma G.5, for any $\epsilon > 0$, the probability that all the phases corresponding to the nodes in $S_\epsilon$ are moved into a single quantization region of length $\pi$ radians after Stage $K = 1$ of Algorithm 2 goes to 1 as $N \to \infty$.

In Stage 2, Propositions 4.2 and 4.3 are applied to estimate the angle between the vector for any Node $n = 1, \cdots, N$ after completing Stage 1 and the sum of the initial vectors for all the nodes in Stage 0. As mentioned above, by Proposition 4.2, the probability that the length of the sum of the initial vectors (RSS$_N[0]$) is arbitrarily larger than the length of all the vectors ($a_n$, $n = 1, \cdots, N$) goes to 1 as $N \to \infty$. Then, using Proposition 4.3, when
\( L = 1 \), it can be easily shown that the estimations for the angles in Stage 2 tend to the exact angles as \( N \to \infty \). Similar to the argument in Stage 1, when \( L > 1 \), by induction on \( L \) and the WLLN, the same result can be shown to hold for the updated phases after each phase update. Then, the result of the theorem is evident since these estimated angles are quantized and fed back to the transmitters in Stages \( k = 2, \ldots, K \).

I Proof of Corollary 4.1

For \( N \) transmitting nodes and any \( \epsilon > 0 \), let \( S_{N,\epsilon} \) be the set of nodes whose vectors are initially in a distance of greater than \( \epsilon \) radians from all the boundaries of the quantization regions. Also, let \( E_{N,\epsilon} \) denote the event that all the phases of the nodes in \( S_{N,\epsilon} \) are moved into a single quantization region of length \( \pi/2^{K-1} \) radians after Stage \( K \geq 1 \) of Algorithm 2. Then, by Theorem 4.3, for any \( \epsilon > 0 \)

\[
\lim_{N \to \infty} P(E_{N,\epsilon}) = 1. \tag{I.52}
\]

Now, let \( Y_{N,\epsilon} \) be the number of elements in \( S_{N,\epsilon} \). Then, (I.52) implies that for any \( \epsilon > 0 \)

\[
\lim_{N \to \infty} P\left\{ Z_N \geq Y_{N,\epsilon} \right\} = 1. \tag{I.53}
\]

Suppose that \( Q = 2^{B(K-1)+1} \) is the number of boundaries of the quantization regions, and assume \( \epsilon \) is smaller than half of the angle created by the boundaries of each quantization region (i.e., \( \epsilon < \frac{\pi}{Q} \)). Then, since the angles for the vectors are initially independently and uniformly distributed in \((-\pi, \pi]\), by the WLLN, for any \( \epsilon' > 0 \) we have

\[
\lim_{N \to \infty} P\left\{ \left| \frac{Y_{N,\epsilon}}{N} - \left(1 - \frac{Q \cdot \epsilon}{\pi}\right) \right| \leq \epsilon' \right\} = 1. \tag{I.54}
\]
Using (I.53) and (I.54) and applying Lemma G.5, by letting $\epsilon \to 0$ and $\epsilon' \to 0$, it can be easily shown that for any $\epsilon'' > 0$

$$
\lim_{N \to \infty} P \left\{ \frac{Z_N}{N} \geq 1 - \epsilon'' \right\} = 1. \tag{I.55}
$$

Since $\frac{Z_N}{N} \leq 1$, (I.55) implies that for any $\epsilon'' > 0$

$$
\lim_{N \to \infty} P \left\{ \left| \frac{Z_N}{N} - 1 \right| \geq \epsilon'' \right\} = 0. \tag{I.56}
$$

This implies the desired result.

\section*{J \ Proof of Theorem 4.4}

Using Eq. (4.2) and the WLLN, it can be easily shown that $\frac{\text{RSS}_{N, \text{max}}}{N} \xrightarrow{p} \sqrt{\pi}$ as $N \to \infty$. Therefore, in order to prove the theorem, we only need to show that

$$
\frac{\text{RSS}_N[K]}{N} \xrightarrow{p} 2^{B(K-1)} \frac{\pi}{\sqrt{\pi}} \sin \left( \frac{\pi}{2B(K-1)+1} \right) \quad \text{as} \quad N \to \infty. \tag{J.57}
$$

Thus, we first show that

$$
\frac{\text{RSS}_N[K] - \tilde{\text{RSS}}_N[K]}{N} \xrightarrow{p} 0 \quad \text{as} \quad N \to \infty \tag{J.58}
$$

where $\tilde{\text{RSS}}_N[K]$ is the RSS when the angles corresponding to all the vectors are quantized and moved into a single quantization region of length $\pi/2^{K-1}$ radians accordingly. Then, to prove the theorem, we show that

$$
\frac{\tilde{\text{RSS}}_N[K]}{N} \xrightarrow{p} 2^{B(K-1)} \frac{\pi}{\sqrt{\pi}} \sin \left( \frac{\pi}{2B(K-1)+1} \right) \quad \text{as} \quad N \to \infty. \tag{J.59}
$$
For any $\epsilon > 0$, let $S_\epsilon$ denote the set of nodes whose vectors are initially in a distance of greater than $\epsilon$ radians from all the boundaries of the quantization regions. By Theorem 4.3, for any $\epsilon > 0$, the probability that all the angles corresponding to the nodes in $S_\epsilon$ are moved into a single quantization region of length $\pi/2^{K-1}$ radians after Stage $K \geq 1$ of Algorithm 2 goes to 1 as $N \to \infty$. In what follows, we show that, for any $\epsilon > 0$, $1/N$ times the sum of the length of the vectors that are not in $S_\epsilon$ converges to $2B(K-1)\sqrt{\frac{\pi}{2}}$ in probability as $N \to \infty$. Then, (J.58) can be easily obtained by noting that $\frac{2B(K-1)\epsilon}{\sqrt{\pi}} \to 0$ as $\epsilon \to 0$ and applying the triangle inequality. Let $X_n, n = 1, \cdots, N$, be random variables defined as

$$X_n = \begin{cases} a_n, & n \notin S_\epsilon \\ 0, & \text{otherwise} \end{cases} \quad (J.60)$$

Notice that the sum of the length of the vectors that are not in $S_\epsilon$ is given by $\sum_{n=1}^{N} X_n$. Then, by the WLLN, as $N \to \infty$ we have

$$\frac{1}{N} \sum_{n=1}^{N} X_n \xrightarrow{p} \mathbb{E}[X_1]$$

$$= \mathbb{E}[a_1] \cdot \left[ 2^{B(K-1)+1} \int_{-\frac{\epsilon}{2\pi}}^{\frac{\epsilon}{2\pi}} \frac{1}{2\pi} dx \right]$$

$$= \frac{2^{B(K-1)}\epsilon}{\sqrt{\pi}}. \quad (J.61)$$

Finally, we employ the WLLN to show (J.59). Using the WLLN, as $N \to \infty$ we have

$$\frac{\text{RSS}_N[K]}{N} \xrightarrow{p} \frac{\sqrt{\pi}}{2} \int_{-\frac{2^{B(K-1)+1}}{\pi}}^{\frac{2^{B(K-1)+1}}{\pi}} 2B(K-1) \frac{\pi}{\cos x} \cos x dx$$

$$= \frac{2^{B(K-1)}\sqrt{\pi}}{2^{B(K-1)+1}} \sin \left( \frac{\pi}{2^{B(K-1)+1}} \right). \quad (J.62)$$

This completes the proof.