

國立交通大學

電信工程學系

博士論文

高速移動單輸入單輸出/多輸入多輸出正交
分頻多工系統的子載波間干擾消除：
低複雜度演算法及效能分析

ICI Mitigation for High-mobility SISO/MIMO
OFDM Systems: Low-complexity Algorithms
and Performance Analysis

研究生：許兆元

指導教授：吳文榕

中華民國 98 年 2 月

高速移動單輸入單輸出/多輸入多輸出正交分頻多工系
統的子載波間干擾消除:低複雜度演算法及效能分析

ICI Mitigation for High-mobility SISO/MIMO OFDM
Systems: Low-complexity Algorithms and Performance
Analysis

研究生：許兆元

Student: Chao-Yuan Hsu

指導教授：吳文榕 博士

Advisor: Dr. Wen-Rong Wu



A Dissertation
Submitted to Institute of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in
Communication Engineering
Hsinchu, Taiwan

2009 年 2 月

高速移動單輸入單輸出/多輸入多輸出正交 分頻多工系統的子載波間干擾消除:低複雜 度演算法及效能分析

研究生：許兆元

指導教授：吳文榕 博士

國立交通大學

電信工程學系博士班



在正交分頻多工系統中，一個基本的假設是在一個正交分頻多工符元時間內通道是靜止不變的。然而，在高速移動的環境下，這假設就不再成立了。因此會造成子載波間干擾且使系統效能降低。強制歸零(zero-forcing, ZF)及最小均方差(minimum mean square error, MMSE)等化器是兩個簡單的干擾消除方法。不幸的，強制歸零等化器需要執行 $N \times N$ 子載波間干擾矩陣的反矩陣運算，此處 N 是正交分頻多工系統的子載波數目。當子載波數目變大時，計算複雜度將會變得很高。對於最小均方差等化器，除了子載波間干擾矩陣的反矩陣運算之外，還需要一個矩陣與矩陣的乘法運算。這將使得最小均方差等化器的複雜度變得比強制歸零等化器還高。在本論文中，我們首先提出一個低複雜度的強制歸零等化器來解決單輸入單輸出正交分頻多工系統中的問題。主要的概念是探究子載波間干擾矩陣的特殊結構及應用牛頓反矩陣疊代法。依據我們的演算法結構，快速傅立葉轉換(fast Fourier transform, FFT)可以被結合到疊代過程中，進而使得複雜度

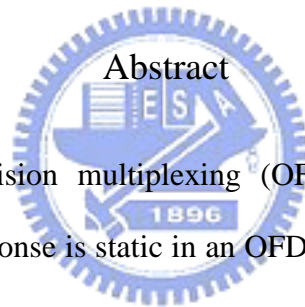
從 $O(N^3)$ 降到 $O(N \log_2 N)$ 。此外，疊代次數約略為一或兩次。我們亦分析所提方法的收斂行為及推導其訊號干擾雜訊比(signal to interference noise ratio, SINR)。對於最小均方差方法，我們首先改寫其數學表示式，使矩陣與矩陣的乘法運算可以被避免。與先前提出的低複雜度強制歸零方法相似，我們也將探究子載波間干擾矩陣的特殊結構及應用牛頓反矩陣疊代法來降低最小均方差方法中反矩陣的高運算複雜度。在多輸入多輸出正交分頻多工系統中，強制歸零及最小均方差等化器所需的複雜度問題將變得更難以解決。有鑑於此，我們將延伸在單輸入單輸出正交分頻多工系統中所提出的演算法至多輸入多輸出正交分頻多工系統中。這樣的延伸應用，使得所降低的複雜度比在單輸入單輸出正交分頻多工系統中還大。模擬結果顯示，所提出的低複雜度強制歸零及最小均方差等化器效能跟直接強制歸零(direct ZF)及直接最小均方差(direct MMSE)等化器的效能相當，但是所需的複雜度卻是大幅降低。最後，我們也將所提的高速移動干擾消除方法再延伸應用到正交分頻多工存取(OFDMA)上傳系統中並將此概念進一步應用來消除子載波偏移所引起的干擾。模擬結果顯示，所提方法可以大幅降低所需複雜度。

ICI Mitigation for High-mobility SISO/MIMO OFDM Systems: Low-complexity Algorithms and Performance Analysis

Student : Chao-Yuan Hsu

Advisor : Dr. Wen-Rong Wu

Institute of Communication Engineering
National Chiao Tung University



In orthogonal frequency-division multiplexing (OFDM) systems, it is generally assumed that the channel response is static in an OFDM symbol period. However, the assumption does not hold in high-mobility environments. As a result, intercarrier interference (ICI) is induced and the system performance is degraded. A simple remedy for this problem is the application of the zero-forcing (ZF) and minimum mean square error (MMSE) equalizers. Unfortunately, the direct ZF method requires the inversion of an $N \times N$ ICI matrix, where N is the number of subcarriers. When N is large, the computational complexity can become prohibitively high. As for the direct MMSE method, in addition to an $N \times N$ matrix inverse, it requires an extra $N \times N$ matrix multiplication, making the required computational complexity higher compared to the direct ZF method. In this dissertation, we first propose a low-complexity ZF method to solve the problem in single-input-single-output (SISO) OFDM systems. The main idea is to explore the special structure inherent in the ICI

matrix and to apply Newton's iteration for matrix inversion. With our formulation, fast Fourier transforms (FFTs) can be used in the iterative process, reducing the complexity from $O(N^3)$ to $O(N \log_2 N)$. Also, the required number of the iteration is typically one or two. We also analyze the convergence behavior of the proposed method and derive the theoretical output signal-to-interference-noise-ratio (SINR). For the MMSE method, we first reformulate the MMSE solution in a way that the extra matrix multiplication can be avoided. Similar to the ZF method, we then exploit the structure of the ICI matrix and apply Newton's iteration to reduce the complexity of the matrix inversion. For a multiple-input-multiple-output (MIMO) OFDM system, the required complexity of the ZF and MMSE methods becomes more intractable. We then manage to extend the proposed ZF and MMSE methods for SISO-OFDM systems to MIMO-OFDM systems. It turns out that the computational complexity can be reduced even more significantly. Simulation results show that the performance of the proposed methods is almost as good as that of the direct ZF and MMSE methods, while the required computational complexity is reduced dramatically. Finally, we explore the application of the proposed methods in mobility-induced ICI mitigation for OFDM multiple access (OFDMA) systems, and in carrier frequency offset (CFO) induced ICI mitigation for OFDMA uplink systems. As that in OFDM systems, the proposed methods can reduce the required computational complexity, effectively.

Acknowledgements

During the Ph.D. program, I would like to show my gratitude to many people. First, I would like to thank my advisor, Prof. Wen-Rong Wu, for his guidance and patience. He spends a lot of time in discussing the problems I encounter in my research, providing valuable suggestions, and teaching me how to write technical papers. In addition to academic research, Prof. Wu also helps me a lot in my daily life. I do learn a lot from Prof. Wu. I really appreciate what he has done for me. I have to say that without Prof. Wu's help, I cannot complete my Ph.D. degree smoothly. I also appreciate the suggestions of the oral defense committee members.

Second, I am grateful to all the members in Prof. Wu's lab. for their valuable discussion and help in academic research including Shou-Sheu Lin, Chun-Fang Lee, Yu-Tao Hsieh, Ren-Jr Chen, Hua-Lung Yang, Yinman Lee, Fan-Shuo Tseng, Hung-Dau Hsieh, Chun-Tao Lin, Pei-Ju Chung, and so on. Besides, I want to thank Shou-Sheu Lin for his encouragement and help during the period of the Ph.D. program. I would like to thank all my friends who ever encourage or help me, especially Kuo-Chun Huang and Nai-Fang Hsu.

Finally, I would like to show my deep gratitude to my family, especially my best-loved parents, Hua-Chung Hsu and Mei-Hua Lee, for their support and encouragement in the Ph.D. program period.



Contents

Abstract	iii
Acknowledgements	v
Contents	vi
List of Tables	x
List of Figures	xiii
1 Introduction	1
1.1 ICI Problem	1
1.2 ICI Mitigation	3
1.3 Proposed Approach	6
1.4 Organization of the Dissertation	8
2 Mobility-induced ICI Mitigation for SISO-OFDM Systems	11
2.1 Signal Model	11
2.2 ZF Method	12
2.2.1 Proposed Newton-ZF Method	13
2.2.2 Derivation of the Initial Matrix	15
2.2.3 Complexity Analysis	20



2.2.4	Performance Analysis	22
2.2.5	Simulation Results	28
2.3	MMSE Method	36
2.3.1	Proposed Newton-MMSE Method	37
2.3.2	Derivation of the Initial Matrix	40
2.3.3	Complexity Analysis	43
2.3.4	Simulations	45
3	Mobility-induced ICI Mitigation for MIMO-OFDM Systems	51
3.1	Signal Model	51
3.2	ZF Method	53
3.2.1	Proposed Newton-ZF Method	53
3.2.2	Derivation of the Initial Matrix	55
3.2.3	Complexity Analysis	57
3.2.4	Simulations	59
3.3	MMSE Method	64
3.3.1	Proposed Newton-MMSE Method	64
3.3.2	Derivation of the Initial Matrix	65
3.3.3	Complexity Analysis	67
3.3.4	Simulations	69
4	Mobility-induced ICI Mitigation for SISO/MIMO-OFDMA Systems	75
4.1	SISO-OFDMA Signal Model	75
4.1.1	Simulations	78
4.2	MIMO-OFDMA Signal Model	83
4.2.1	Simulations	86
5	CFO-induced ICI Mitigation for OFDMA Uplink Systems	93
5.1	Signal Model	93

5.2	Previous Methods	95
5.2.1	Conventional Method	95
5.2.2	CLJL Method	96
5.2.3	CLJL-PIC Method	97
5.3	ZF Method	98
5.3.1	Proposed Newton-ZF Method	98
5.3.2	Pre-compensation Method	102
5.3.3	Complexity Analysis	102
5.3.4	Performance Analysis	104
5.3.5	Simulations	114
6	Conclusions	127
	Bibliography	129





List of Tables

2.1	Complexity comparison among N-ZF, PSE, and direct ZF methods in a SISO-OFDM system.	22
2.2	Complexity of the initial matrix calculation for the N-ZF method in a SISO-OFDM system.	22
2.3	Complexity comparison among N-ZF, PSE, and direct ZF methods in a SISO-OFDM system ($N = 128$ and $S = 2$).	37
2.4	Complexity comparison between the N-MMSE and direct MMSE methods in a SISO-OFDM system.	45
2.5	Complexity of the initial matrices calculation for the N-MMSE method in a SISO-OFDM system.	45
2.6	Complexity comparison between the N-MMSE and direct MMSE methods in a SISO-OFDM system ($N = 128$, and $\{D_1, D_2\} = \{1, 1\}$).	50
3.1	Complexity comparison between N-ZF and direct ZF methods in a 2×2 MIMO-OFDM system.	58
3.2	Complexity of the initial matrix calculation for the N-ZF method in a 2×2 MIMO-OFDM system.	59
3.3	Complexity comparison between N-ZF and direct ZF methods in a 2×2 MIMO-OFDM system ($N = 128$ and $S = 2$).	63
3.4	Complexity comparison between the N-MMSE and direct MMSE methods in a 2×2 MIMO-OFDM system.	69

3.5	Complexity of the initial matrices calculation for the N-MMSE method in a 2×2 MIMO-OFDM system.	69
3.6	Complexity comparison between the N-MMSE and direct MMSE methods in a 2×2 MIMO-OFDM system ($N = 128$, and $\{D_1, D_2\} = \{1, 1\}$).	74
4.1	Complexity comparison between direct ZF and N-ZF methods in a SISO-OFDMA system ($N = 128$, $D = 0$, and $S = 2$).	83
4.2	Complexity comparison between direct MMSE and N-MMSE methods in a SISO-OFDMA system ($N = 128$, $D_1 = D_2 = 0$, and $S_1 = S_2 = 2$).	83
4.3	Complexity comparison between direct ZF and N-ZF methods in a 2×2 MIMO-OFDMA system ($N = 128$, $D = 0$, and $S = 2$).	91
4.4	Complexity comparison between the N-MMSE and direct MMSE methods in a 2×2 MIMO-OFDMA system ($N = 128$, $\{D_1, D_2\} = \{0, 0\}$, and $\{S_1, S_2\} = \{2, 2\}$).	91
5.1	Complexity comparison among proposed method, CLJL-PIC method, and direct ZF method.	104
5.2	Complexity comparison among direct ZF method, CLJL-PIC method, and proposed method when $N = 64$ and $Q = 4$	121
5.3	Complexity comparison of the direct ZF method, the banded ZF method, and the proposed method when $N = 2048$ and $Q = 16$	121

List of Figures

2.1	An example of the structure of a banded initial matrix for $N = 8$ and $D = 1$. The elements in the shaded area are non-zeros, while the others are zeros. . . .	16
2.2	An example of \mathbf{A}_i for $N = 8$ and $D = 1$. Note that \mathbf{A}_i overlaps with \mathbf{A}_{i-1} and \mathbf{A}_{i+1}	18
2.3	SINR analysis of N-ZF method ($D = 0$ and $S = 2$) for case 1, where $f_d = 0.1$ and SNR = 35 dB.	32
2.4	SINR analysis of N-ZF method ($D = 0$ and $S = 2$) for case 2, where $f_d = 0.1$ and SNR = 35 dB.	32
2.5	BER comparison among one-tap FEQ, PSE, N-ZF ($D = 0$ and $S = 2$), direct ZF, and direct MMSE methods in a SISO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.	33
2.6	BER comparison among one-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a SISO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.	33
2.7	BER comparison between the direct ZF and MMSE methods using the exact Jakes and LTV channels in a SISO-OFDM system; $f_d = 0.1$ and 16-QAM mod- ulation.	34
2.8	BER comparison among one-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a SISO-OFDM system; $f_d = 0.2$ and 16-QAM modulation.	34

2.9	BER comparison among one-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a SISO-OFDM system; $f_d = 0 \sim 0.2$, 16 QAM modulation, and SNR = 30 dB.	35
2.10	Complexity comparison between N-ZF ($D = 1$, $S = 2$, and $k = 2$) and direct ZF methods in a SISO-OFDM system for various N	35
2.11	Complexity comparison between N-ZF ($D = 1$, $S = 2$, and $k = 3$) and direct ZF methods in a SISO-OFDM system for various N	36
2.12	The structure of a circular band initial matrix is depicted for $N = 8$ and $D = 1$. The elements in the shaded area are non-zeros, while the others are zeros. . . .	41
2.13	The structure of \mathbf{A}_i is presented for $N = 8$ and $D = 1$. Note that \mathbf{A}_i overlaps \mathbf{A}_{i+1} for modulo-8 i (i.e., the relationship is circular).	42
2.14	BER comparison among the one-tap FEQ, N-MMSE ($\{D_1, D_2\} = \{1, 1\}$ and $\{S_1, S_2\} = \{2, 2\}$), and direct MMSE methods in a SISO-OFDM system; $f_d = 0.05$ and 16-QAM modulation.	47
2.15	BER comparison among the one-tap FEQ, N-MMSE ($\{D_1, D_2\} = \{1, 1\}$ and $\{S_1, S_2\} = \{5, 5\}$), and direct MMSE methods in a SISO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.	48
2.16	BER comparison among one-tap FEQ, N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 5$, and $k = 3$), and direct MMSE methods in a SISO-OFDM system; $f_d = 0 \sim 0.2$, 16 QAM modulation, and SNR = 30 dB.	48
2.17	Complexity comparison between N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 5$, and $k = 2$) and direct MMSE methods in a SISO-OFDM system for various N . . .	49
2.18	Complexity comparison between N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 10$, and $k = 3$) and direct MMSE methods in a SISO-OFDM system for various N . . .	49
3.1	BER comparison among two-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0.05$ and 16-QAM modulation.	61

3.2	BER comparison among two-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.	61
3.3	BER comparison among one-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a 2×2 SISO-OFDM system; $f_d = 0 \sim 0.2$, 16 QAM modulation, and SNR = 30 dB.	62
3.4	Complexity comparison between N-ZF ($D = 1$, $S = 2$, and $k = 2$) and direct ZF methods in a 2×2 MIMO-OFDM system for various N	62
3.5	Complexity comparison between N-ZF ($D = 1$, $S = 2$, and $k = 3$) and direct ZF methods in a 2×2 MIMO-OFDM system for various N	63
3.6	BER comparison among the two-tap FEQ, N-MMSE ($\{D_1, D_2\} = \{1, 1\}$ and $\{S_1, S_2\} = \{2, 2\}$), and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0.05$ and 16-QAM modulation.	71
3.7	BER comparison among the two-tap FEQ, N-MMSE ($\{D_1, D_2\} = \{1, 1\}$ and $\{S_1, S_2\} = \{10, 10\}$), and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.	72
3.8	BER comparison among one-tap FEQ, N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 10$, and $k = 3$), and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0 \sim 0.2$, 16 QAM modulation, and SNR = 30 dB.	72
3.9	Complexity comparison between N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 5$, and $k = 2$) and direct MMSE methods in a 2×2 MIMO-OFDM system for various N	73
3.10	Complexity comparison between N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 10$, and $k = 3$) and direct MMSE methods in a 2×2 MIMO-OFDM system for various N	73

4.1	BER comparison among one-tap FEQ, direct ZF, and N-ZF ($D = 0, S = 2$) methods in a SISO-OFDMA system; $f_d = \{0.05, 0.02, 0.04, 0.03\}$ and 16-QAM modulation.	81
4.2	BER comparison among one-tap FEQ, direct ZF, and N-ZF ($D = 0, S = 2$) methods in a SISO-OFDMA system; $f_d = \{0.05, 0.1, 0.04, 0.03\}$ and 16-QAM modulation.	81
4.3	BER comparison among one-tap FEQ, direct MMSE, and N-MMSE ($D_1 = D_2 = 0, S_1 = S_2 = 2$) methods in a SISO-OFDMA system; $f_d = \{0.05, 0.02, 0.04, 0.03\}$ and 16-QAM modulation.	82
4.4	BER comparison among one-tap FEQ, direct MMSE, and N-MMSE ($D_1 = D_2 = 0, S_1 = S_2 = 2$) methods in a SISO-OFDMA system; $f_d = \{0.05, 0.1, 0.04, 0.03\}$ and 16-QAM modulation.	82
4.5	BER comparison among one-tap FEQ, direct ZF, and N-ZF ($D = 0, S = 2$) methods in a 2×2 MIMO-OFDMA system; $f_d = \{0.02, 0.05, 0.03, 0.04\}$ and 16-QAM modulation.	89
4.6	BER comparison among one-tap FEQ, direct ZF, and N-ZF ($D = 0, S = 2$) methods in a 2×2 MIMO-OFDMA system; $f_d = \{0.04, 0.1, 0.08, 0.07\}$ and 16-QAM modulation.	89
4.7	BER comparison among two-tap FEQ, direct MMSE, and N-MMSE ($D_1 = D_2 = 0, S_1 = S_2 = 2$) methods in a 2×2 MIMO-OFDMA system; $f_d = \{0.02, 0.05, 0.03, 0.04\}$ and 16-QAM modulation.	90
4.8	BER comparison among two-tap FEQ, direct MMSE, and N-MMSE ($D_1 = D_2 = 0, S_1 = S_2 = 2$) methods in a 2×2 MIMO-OFDMA system; $f_d = \{0.04, 0.1, 0.08, 0.07\}$ and 16-QAM modulation.	90
5.1	Theoretical average SINR for the proposed method, exact and approximated (Case 1: CFOs = $\{0.1, -0.2, -0.05, 0.2\}$ and SNR = 30 dB; Case 2: CFOs = $\{0.49, 0.49, 0.15, 0.4\}$ and SNR = 15 dB).	120

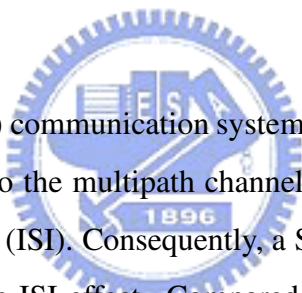
5.2	Theoretical average SINR for the proposed method (exact).	120
5.3	Theoretical subcarrier SINR for the proposed method (exact; SNR = 30 dB). . .	122
5.4	BER performance comparison for conventional, CLJL, CLJL-PIC, proposed, and direct ZF methods (16-QAM modulation, and CFOs = {0.1, -0.2, -0.05, 0.2}).	122
5.5	BER performance comparison for conventional, CLJL, CLJL-PIC, proposed, and direct ZF methods (64-QAM modulation, and CFOs = {0.1, -0.2, -0.05, 0.2}).	123
5.6	BER performance comparison for conventional, CLJL, CLJL-PIC, proposed, and direct ZF methods (16-QAM modulation, and CFOs = {0.1, -0.4, -0.05, 0.2}).	123
5.7	BER performance comparison for CLJL-PIC ($p = 3$), proposed, and direct ZF methods (16-QAM modulation, and CFO of the fourth user increases from 0 to 0.5).	124
5.8	BER performance comparison for CLJL-PIC ($p = 3$), proposed ($k = 2$), and direct ZF methods (16-QAM modulation, CFOs = {0.1, -0.2, -0.05, 0.2}, and near-far scenario).	124
5.9	BER performance comparison for the proposed method with and without PC (16-QAM modulation, and CFOs = {0.49, 0.49, 0.1, 0.4}).	125
5.10	BER performance comparison for the proposed method with PC (16-QAM modulation and CFOs = {-0.1, 0.3, 0.4, -0.2}).	125
5.11	BER performance comparison for the conventional, CLJL, banded ZF, and di- rect ZF and proposed methods (16-QAM modulation, and CFOs = {0.1, -0.2, -0.05, 0.2, -0.3, 0, -0.1, 0.4, -0.3, 0.05, 0, -0.1, 0.05, -0.1, 0.3, 0.15}).	126



Chapter 1

Introduction

§ 1.1 ICI Problem



IN wireless single-carrier (SC) communication systems, data transmission occupies the whole available bandwidth. Due to the multipath channel, a SC system usually suffers from the severe intersymbol interference (ISI). Consequently, a SC system requires a complicated time-domain equalizer to combat the ISI effect. Compared to a SC system, a conventional multi-carrier (MC) system divides the whole available bandwidth into many non-overlapped narrow subchannels and subcarriers are used simultaneously to transmit data on these narrow subchannels. Since each data stream is transmitted on a narrow subchannel, it is subject to little ISI which makes the channel equalizer simpler. Moreover, since the data streams are transmitted on independent subchannels, different modulation schemes can be used for the subchannels. Since the subchannels are non-overlapped in conventional MC systems, guard bands are required between these subchannels to avoid inter-channel interference. Owing to the extra guard bands, the conventional MC system is bandwidth-inefficient.

To solve the problem, a bandwidth-efficient MC technique, called orthogonal frequency-division multiplexing (OFDM), was developed. The technique, dating back to the 1960's, overlaps subchannels in an orthogonal way such that bandwidth efficiency can be greatly improved.

In 1971, Weinstein and Ebert used IDFTs/DFTs to perform the OFDM baseband modulation and demodulation instead of a bank of subcarrier oscillators [1]. This method provides an efficient digital implementation of OFDM systems. In 1980, Peled and Ruiz introduced the concept of cyclic prefix (CP) which fills the vacant guard interval with a cyclic extension of an OFDM symbol [2]. This results in a circular convolution between the transmit data and the channel response. With the CP, OFDM can convert a frequency-selective channel into a set of frequency non-selective channels, and only a one-tap frequency-domain equalizer is required for each subcarrier signal. This greatly reduces the complexity of the channel equalization in the OFDM receiver. Nowadays, OFDM is known to be an effective and successful technique to cope with the multipath channel effect in wireless communications [3]. Since all subcarrier signals overlap orthogonally in the spectrum, an ideal OFDM system has no intercarrier interference (ICI). Thus, OFDM can be easily developed as a frequency-division multiple access (FDMA) scheme. An OFDM-based FDMA system is generally referred to as an OFDMA system [4], [5]. In an OFDMA system, subcarriers are divided into exclusive groups, and each group is assigned to a user for simultaneous data transmission. The OFDM technique has been adopted in many systems, e.g. Asymmetric Digital Subscriber Line (ADSL), IEEE 802.11a/g, IEEE 802.16e-2005 [6], IEEE 802.16m, 3GPP Long Term Evolution (LTE), Digital Audio Broadcasting (DAB), and Terrestrial Digital Video Broadcasting (DVB-T).

For conventional OFDM systems, it is usually assumed that the channel is static during an OFDM symbol. However, in high-speed mobile environments, this assumption does not hold anymore. If the channel is time-variant in an OFDM symbol period, orthogonality will be destroyed. As a result, ICI is induced and the system performance is degraded. The behavior of mobility-induced ICI has been extensively investigated in the literature [7], [8], [9], [10], [11], [12]. In [7], [8], it is shown that the interference on a subcarrier mainly comes from neighboring subcarriers. Also, the interference level is proportional to the Doppler frequency.

Another factor that affects orthogonality in an OFDM system is carrier frequency offset (CFO). In OFDM systems, CFO is always present due to imperfect oscillators. In the presence

of CFO, the orthogonal property of an OFDM system is also destroyed and the ICI is induced, degrading the system performance significantly [13]. Different from the CFO-induced ICI in OFDM systems, CFO in OFDMA uplink systems causes not only the self-interference but also the multiuser interference (MUI), degrading the system performance even more severely [14], [15]. ICI mitigation has been studied by many researchers and this will be the focus of the dissertation.

§ 1.2 ICI Mitigation

Since an OFDM system is vulnerable to mobility and CFO, various techniques have been proposed to cope with these two kinds of ICI. First, we discuss the mobility-induced ICI problem. Two algorithms are well-known, namely, 1) the zero-forcing (ZF) method and 2) minimum mean square error (MMSE) method. Unfortunately, these methods require the inversion of an $N \times N$ ICI matrix, where N is the number of subcarriers. Except for a matrix inversion, the MMSE method also needs to conduct an extra $N \times N$ matrix multiplication. Thus, its computational complexity is even higher than that of the ZF method. The payoff for the higher complexity is its enhanced performance. If N is large, the computational complexity of both algorithms can become prohibitively high. Systems with a lot of subcarriers are not uncommon in real-world applications. For example, for the application of DVB, the number of subcarriers can be as large as 8192. To solve the problem of a large ICI matrix inversion, a simpler ICI equalizer for the ZF method was developed in [16]. As mentioned, ICI on a subcarrier mainly comes from a few neighboring subcarriers. Thus, ICI from the other subcarriers can then be ignored. This method has good performance in low-mobility environments. In high-mobility environments, however, the number of insignificant ICI terms will be decreased and the computational complexity will be significantly increased.

Successive interference cancellation (SIC) and parallel interference cancellation (PIC) are two well-known multiuser interference (MUI) cancellation techniques in code-division-multiple-

access (CDMA) systems. Since the characteristic of ICI is similar to that of MUI, these methods can be directly applied to ICI mitigation in OFDM systems. A method combining the MMSE and SIC techniques was first proposed in [17]. Later, it was improved with a recursive method in [18], reducing the required complexity further. Although good performance can be achieved with these methods, the required complexity is still high and the time delay can be intolerably large. The PIC technique was then employed to solve the problem [19], [20], [21], [22], [23]. Although the processing delay is greatly reduced, the performance is discounted as well. Other approaches use transmitter frequency-domain coding or beamforming to reduce ICI or to enhance the received signal-to-interference-noise-ratio (SINR). Interested readers may see [24], [25], [26], [27].

Apart from the processing in the frequency domain, some researchers also explore that in the time domain. In [28], a time-domain filtering technique maximizing the signal-to-ICI-plus-noise ratio was proposed for single-input-single-output(SISO)/multiple-input-multiple-output (MIMO) OFDM systems. One disadvantage of this method is that it requires matrix operations to solve a generalized eigenvalue problem. Another approach involves the use of a time-variant time-domain equalizer, making the time-variant channel less variant. Transferring the equalizer from time-domain to frequency-domain, one can obtain a frequency-domain per-tone equalizer (PTEQ). The PTEQ was originally proposed to deal with the insufficient CP problem in OFDM systems. Lately, it is extended to suppress ICI in SISO/MIMO-OFDM systems [29], [30], [31], [32], [33]. The PTEQ is well-known for its good performance; however, its implementation complexity and storage requirement can be high. In [34], a two-stage equalizer was proposed. In the first stage, a time-domain windowing technique is used to shorten the ICI response in the frequency domain. In the second stage, an iterative MMSE method is used to suppress the residual ICI. Although the windowing approach is simple, the iterative MMSE processing is not trivial. To further enhance the system performance, another approach called turbo equalization can be applied to mitigate ICI [35], [36], [37]. In [37], a block turbo MMSE method was proposed. The main feature is that this method uses the whole ICI matrix to obtain the MMSE

solution although it ignores some insignificant ICI terms.

Next, we discuss the CFO-induced ICI mitigation problem. For OFDM and OFDMA downlink systems, the CFO can be easily estimated and compensated in the receiver [38], [39]. However, for OFDMA uplink systems, the problem is more involved. In the literature, various ICI mitigation methods have been proposed to solve the problem. One direct method is to estimate CFO in the base station and transmit the information back to mobile stations for CFO correction. Another approach is to transmit redundant information in subcarriers such that ICI can be cancelled with a simple method in the receiver end. This approach is called the self-ICI-cancellation [24], [40], [41], [42], [43], [44]. However, these methods mentioned above will sacrifice the transmission rate.

Yet another viable approach eliminates the need for extra transmission overhead by compensating for ICI in the receiver. CFO compensation methods for OFDMA uplink systems have been reported [45], [46], [47], [48], [49], [50], [51]. The simplest method is to treat the CFO-induced ICI as that in OFDM systems and to compensate for ICI with a time-domain phase de-rotation operation for each user [45]. This approach can suppress self-ICI, but it does not take MUI into account. In [46], a post-FFT CFO compensation method was proposed, improving the performance of the phase de-rotation approach. Unfortunately, the MUI problem still remains. In [47], a scheme combining the method in [46] with the PIC technique was developed. Other PIC-related works can be found in [48], [49]. It is simple to observe that the CFO-induced ICI on a subcarrier mainly comes from neighboring subcarriers. Thus, the method in [50] modifies the CFO-induced ICI matrix into a banded matrix, and reduces the computational complexity of the ZF and MMSE methods. However, its performance may be compromised due to the simplification. Taking advantage of an interleaved-OFDMA structure, the authors in [51] proposed a method that divides the whole system into several smaller subsystems, after which the MMSE method was applied to the subsystems. This method has good performance, and it requires low computational complexity; however, it is only applicable to an ideal interleaved structure (i.e., uniform subcarrier-spacing for each user). The aforementioned methods were developed

for CFO-compensation. CFO estimation methods have also been reported for OFDMA uplink systems [52], [53], [54], [55].

§ 1.3 Proposed Approach

As mentioned, the main problem in the ZF and MMSE methods is the matrix inversion. Thus, how to conduct this operation efficiently becomes the main concern. It is found that some iterative methods can be much more efficient than the direct matrix inversion method. We first discuss the mobility-induced ICI problem. In [56], the Gauss-Seidel iteration was used to conduct the matrix inversion. However, it still needs a matrix inverse in its iterative process. Another method called operator-perturbation was recently proposed [57]. Similar to [56], this method also requires a matrix inverse in its iterations. Thus, the computational complexity for the methods in [56] and [57] is still high. In [58], it was discovered that the ICI matrix for a linear time-variant (LTV) channel model exhibits a special structure, allowing the application of fast Fourier transforms (FFTs) in the matrix inversion. The LTV channel model was proposed in [16] and its original purpose is for the time-variant channel estimation [59]. Exploiting this structure, a power-series expansion (PSE) method was proposed for the ICI matrix inversion [58], [60]. Although the PSE method can greatly reduce the computational complexity, it does not perform well in high-mobility environments.

In this dissertation, we propose low-complexity ZF and MMSE methods to solve the mobility-induced ICI problem in SISO/MIMO-OFDM(A). Similar to [58], we exploit the special structure inherent in the LTV channel model. For the ZF method, we first develop a method that can implement Newton's iteration for the ICI matrix inversion in SISO-OFDM systems. With our specially designed architecture, FFTs can be used in the iterative process, reducing the computational complexity effectively. We also propose a method for the calculation of initial values. With those values, Newton's iteration can converge very fast, usually within a couple of iterations. Unlike the PSE method [58], our method works well even in high-mobility en-

vironments. Simulation results show that the performance of the proposed low-complexity ZF method can be as good as that of the direct ZF method. However, the required computational complexity is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(N\log_2 N)$. We also analyze the convergence behavior of the proposed low-complexity ZF algorithm and derive the theoretical output SINR. Using a new MIMO-OFDM system formulation, we then extend the proposed method to ICI mitigation in MIMO-OFDM systems. It is shown that in MIMO-OFDM systems, the computational complexity can be reduced even more significantly. For an $M \times M$ system, where M is the number of transmit (receive) antennas, the proposed algorithm can reduce the computational complexity from $\mathcal{O}(M^3 N^3)$ to $\mathcal{O}(MN\log_2 N)$.

As mentioned, the matrix inversion is the main obstacle in the ZF method, and some researchers try to use iterative methods to overcome this problem [56], [57], [58]. Although these methods can reduce the computational complexity of the ZF method, they are not applicable for the MMSE method. As mentioned above, the MMSE method has to conduct an extra $N \times N$ matrix multiplication which cannot be avoided in these approaches. Using the basic idea in the proposed low-complexity ZF method, we further develop an efficient low-complexity MMSE method. The main contribution in the proposed ZF method is to develop an efficient ICI mitigation scheme using Newton's iteration. With the approach, we can use FFTs/IFFTs in the computation of the matrix inversion, dramatically reducing the computational complexity. The proposed MMSE algorithm inherits this property, and further eliminates the requirement of the $N \times N$ matrix multiplication. Simulation results show that the performance of the proposed low-complexity MMSE method is similar to that of the direct MMSE method. However, the required computational complexity is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(2N\log_2 N)$. We also extend the proposed method to ICI mitigation in MIMO-OFDM systems. For an $M \times M$ system, the proposed algorithm can reduce the computational complexity from $\mathcal{O}(M^3 N^3)$ to $\mathcal{O}(2MN\log_2 N)$. It is simple to see that for MIMO-OFDM systems, the proposed method can reduce the computational complexity even more significantly. Moreover, we apply the proposed ZF and MMSE methods to ICI mitigation in SISO/MIMO-OFDMA systems.

Next we discuss the CFO-induced ICI problem in OFDMA uplink systems. The ZF method is known to be a simple yet effective method for CFO compensation. However, it has to invert the CFO-induced ICI matrix whose dimension equals the number of subcarriers. As a result, the computational complexity can become prohibitively high when the number of subcarriers is large, a case commonly found in OFDMA systems. As we can see, this problem is similar to ICI mitigation in high-mobility environments. We then propose a low-complexity iterative ZF method to cope with the problem. Following the idea described above, we use Newton's method to iteratively perform the matrix inversion. Taking advantage of the special structure of the CFO-induced ICI matrix, we develop a method that can implement Newton's method with FFTs. With our specially designed initial matrix, the proposed iterative method can stop within two to three iterations. From simulation results, we find that the performance of the proposed method is similar to that of the direct ZF method. However, the required computational complexity is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log_2 N)$. We also analyze the theoretical SINR enhancement of the proposed algorithm. Two approaches are used for the analysis; one is simple but approximated, and the other is complicated but exact. The issue of convergence is also discussed.

§ 1.4 Organization of the Dissertation

The rest of this dissertation is organized as follows. In Chapter 2, we first describe an LTV-based SISO-OFDM signal model. This signal model involves DFT/IDFT and diagonal matrices which can be used to develop low-complexity ICI mitigation algorithms. Since the PSE method also exploits the LTV-based SISO-OFDM signal model, it is briefly reviewed. Then we describe the proposed ZF method and present the complexity and performance analysis. Lastly, we show simulation results to corroborate the proposed algorithm. Except for the ZF method, we also consider the MMSE method. We reformulate the MMSE solution and extend the proposed ZF method to avoid the extra matrix multiplication and matrix inversion.

In Chapter 3, we focus on ICI mitigation in a MIMO-OFDM system. For this system, conducting the matrix inversion is more difficult since the dimension of the ICI matrix can be huge. To solve the problem, we derive a MIMO-OFDM signal model that allows the application of the proposed low-complexity ZF and MMSE methods developed in Chapter 2. It can be shown that the reduction in computational complexity is even more significant in MIMO-OFDM systems.

In Chapter 4, we further consider the ICI mitigation problem in an OFDMA uplink system. Based on the LTV channel model, we derive a SISO-OFDMA signal model from which the low-complexity algorithms developed in Chapter 2 can be applied. This signal model can be viewed as a generalized SISO-OFDM signal model. When the number of users is one, the SISO-OFDMA signal model is degenerated to the SISO-OFDM signal model. We also extend the model from SISO-OFDMA systems to MIMO-OFDMA systems. Chapters 2, 3 and 4 give a complete treatment for mobility-induced ICI mitigation in SISO/MIMO-OFDM(A) systems.

In Chapter 5, we discuss low-complexity algorithms for CFO-induced ICI mitigation in OFDMA uplink systems. We first describe the OFDMA uplink signal model that is composed of diagonal and DFT/IDFT matrices. Applying the low-complexity algorithms developed in Chapter 2, we obtain low-complexity CFO-induced ICI mitigation methods. We also propose a pre-compensation method to further enhance the performance of the low-complexity methods. Complexity and performance analyses are also provided to verify the effectiveness of the proposed method.

In Chapter 6, we draw some conclusions for the dissertation and outline possible topics for further research.



Chapter 2

Mobility-induced ICI Mitigation for SISO-OFDM Systems

§ 2.1 Signal Model

Consider a mobile OFDM system whose channel variation is large such that the mobility-introduced ICI cannot be ignored. It was shown in [59] that the LTV channel model can be used to approximate a time-variant channel for the normalized Doppler frequency up to 20%, where the normalized Doppler frequency is defined as the maximum Doppler frequency divided by subcarrier spacing. Using the LTV channel model, we can approximate the time-variant channel in a specific OFDM symbol period as

$$h_j(n) = h_{0,j} + n \times h_{1,j}, \quad (2.1)$$

where n is the time index, $h_j(n)$ is the j th-tap channel response at time instant n , $h_{0,j}$ is its constant term, and $h_{1,j}$ is its variation slope. We assume that n is 0 at the midpoint of an OFDM symbol. Let $\mathbf{h}_0 = [h_{0,0}, h_{0,1}, \dots, h_{0,N-1}]^T$, $\mathbf{h}_1 = [h_{1,0}, h_{1,1}, \dots, h_{1,N-1}]^T$, $\mathbf{H}_0 = \text{cir}(\mathbf{h}_0)$, and $\mathbf{H}_1 = \text{cir}(\mathbf{h}_1)$, where $\text{cir}(\mathbf{c})$ denotes a circulant matrix with the first column vector being \mathbf{c} . Also, we define $\mathbf{v}_1 = [(-N+1)/2, (-N+3)/2, \dots, (N-1)/2]^T$ and $\mathbf{V}_1 = \text{diag}(\mathbf{v}_1)$, where

the notation, $diag(\mathbf{d})$, denotes a diagonal matrix with the diagonal vector of \mathbf{d} . According to (2.1), we can express the receive time-domain signal in the OFDM symbol (after CP removal) as

$$\mathbf{y} = (\mathbf{H}_0 + \mathbf{V}_1\mathbf{H}_1)\mathbf{x} + \mathbf{z}, \quad (2.2)$$

where \mathbf{y} and \mathbf{x} are the receive and transmit time-domain $N \times 1$ signal vectors, respectively, and \mathbf{z} is the noise vector (additive white Gaussian). Let \mathbf{G} be a unitary discrete Fourier transform (DFT) matrix with the property that $\mathbf{G}\mathbf{G}^H = \mathbf{I}_N$, where \mathbf{I}_N is an $N \times N$ identity matrix. Moreover, let $\tilde{\mathbf{y}} = \sqrt{N}\mathbf{G}\mathbf{y}$, $\tilde{\mathbf{x}} = \sqrt{N}\mathbf{G}\mathbf{x}$, $\tilde{\mathbf{z}} = \sqrt{N}\mathbf{G}\mathbf{z}$, $\tilde{\mathbf{h}}_0 = \sqrt{N}\mathbf{G}\mathbf{h}_0$, $\tilde{\mathbf{h}}_1 = \sqrt{N}\mathbf{G}\mathbf{h}_1$, $\tilde{\mathbf{H}}_0 = diag(\tilde{\mathbf{h}}_0)$, and $\tilde{\mathbf{H}}_1 = diag(\tilde{\mathbf{h}}_1)$. Multiplying both sides of (2.2) by $\sqrt{N}\mathbf{G}$, we can express the receive signal in the frequency domain as

$$\begin{aligned} \tilde{\mathbf{y}} &= \sqrt{N}\mathbf{G}\mathbf{y} \\ &= \sqrt{N}\mathbf{G}(\mathbf{H}_0 + \mathbf{V}_1\mathbf{H}_1)\mathbf{G}^H\mathbf{G}\mathbf{x} + \sqrt{N}\mathbf{G}\mathbf{z} \\ &= \left[\tilde{\mathbf{H}}_0 + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{H}}_1 \right] \tilde{\mathbf{x}} + \tilde{\mathbf{z}} \\ &= \tilde{\mathbf{M}}\tilde{\mathbf{x}} + \tilde{\mathbf{z}}, \end{aligned} \quad (2.3)$$

where $\tilde{\mathbf{M}} = \tilde{\mathbf{H}}_0 + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{H}}_1$ is the so-called ICI matrix. Note that $\tilde{\mathbf{M}}$ can also be rewritten as $\tilde{\mathbf{M}} = \tilde{\mathbf{H}}_0 + \tilde{\mathbf{V}}_1\tilde{\mathbf{H}}_1$, where $\tilde{\mathbf{V}}_1 = \mathbf{G}\mathbf{V}_1\mathbf{G}^H = [cir(\tilde{\mathbf{v}}_1)]^T$ and $\tilde{\mathbf{v}}_1 = (1/\sqrt{N})\mathbf{G}^H\mathbf{v}_1$. Since the ICI matrix is not a diagonal matrix, the ICI exists. If the channel is time-invariant, the frequency-domain ICI term, $\tilde{\mathbf{V}}_1$, will disappear. Thus, the signal model will become the traditional OFDM signal model.

§ 2.2 ZF Method

Among the ICI mitigation methods, the simplest remedy for ICI is the ZF method. Denote the ZF equalized signal as $\bar{\mathbf{x}}_{ZF}$. Then, we can obtain the equalized signal as $\bar{\mathbf{x}}_{ZF} = \tilde{\mathbf{M}}^{-1}\tilde{\mathbf{y}}$. From the above formulation, it is simple to see that direct implementation of the ZF method

will require high computational complexity if N is large. Thus, it is a critical problem for the ZF method. Then, the PSE method was introduced in [58] to solve the problem. The idea is to express $\widetilde{\mathbf{M}}^{-1}$ as

$$\begin{aligned}\widetilde{\mathbf{M}}^{-1} &= \left[\left(\mathbf{I}_N + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\widetilde{\mathbf{H}}_1\widetilde{\mathbf{H}}_0^{-1} \right) \widetilde{\mathbf{H}}_0 \right]^{-1} \\ &= \widetilde{\mathbf{H}}_0^{-1} (\mathbf{I}_N - \mathbf{P})^{-1},\end{aligned}\quad (2.4)$$

where $\mathbf{P} = \mathbf{G}\mathbf{V}_1\mathbf{G}^H\widetilde{\mathbf{H}}$ and $\widetilde{\mathbf{H}} = -\widetilde{\mathbf{H}}_1\widetilde{\mathbf{H}}_0^{-1}$. Next, $(\mathbf{I}_N - \mathbf{P})^{-1}$ is expanded with a power series and the high order terms are truncated, i.e., $(\mathbf{I}_N - \mathbf{P})^{-1} \approx \sum_{i=0}^U \mathbf{P}^i$, where U is the highest order retained in the expansion. The convergence condition for this expansion is that $\|\mathbf{P}\| < 1$ [58], where $\|\mathbf{P}\|$ indicates the p-norm of \mathbf{P} [60]. Finally, the equalized $\widetilde{\mathbf{x}}$, denoted as $\widetilde{\mathbf{x}}_{PSE}$, is equal to $\widetilde{\mathbf{H}}_0^{-1} \sum_{i=0}^U \mathbf{a}_i$, where $\mathbf{a}_i = \mathbf{P}^i \widetilde{\mathbf{y}}$. Note that $\mathbf{a}_{i+1} = \mathbf{P}\mathbf{a}_i = \mathbf{G}\mathbf{V}_1\mathbf{G}^H(\widetilde{\mathbf{H}}\mathbf{a}_i)$. Thus, \mathbf{a}_i can be recursively calculated. Also, with the special structure of \mathbf{P} , FFTs/IFFTs can be used to calculate \mathbf{a}_i . Although the computational complexity can be reduced effectively, the performance of the PSE method is unsatisfactory in high-mobility environments. This will be verified from simulation results. In the following subsection, we will present the proposed method to solve the problem.

§ 2.2.1 Proposed Newton-ZF Method

As mentioned, the performance of the PSE method is unsatisfactory in high-mobility environments. To solve the problem, we seek for a more flexible and powerful iterative method for matrix inversion. Specifically, we find Newton's iteration is useful. Newton's iteration is well-known for its fast convergence [61], [62], and it has been investigated extensively [63], [64], [65], [66]. Let \mathbf{W}_k be the estimated matrix inverse of $\widetilde{\mathbf{M}}$ at the k th iteration. The $(k + 1)$ th Newton's iteration can be described as follows:

$$\mathbf{W}_{k+1} = (2\mathbf{I}_N - \mathbf{W}_k\widetilde{\mathbf{M}})\mathbf{W}_k, \quad k = 0, 1, 2, \dots, \infty. \quad (2.5)$$

Let $\widetilde{\mathbf{R}}_k = \mathbf{I}_N - \mathbf{W}_k\widetilde{\mathbf{M}}$ represent the estimation residual. Equation (2.5) implies that $\|\mathbf{I}_N - \mathbf{W}_k\widetilde{\mathbf{M}}\| \leq \|\mathbf{I}_N - \mathbf{W}_0\widetilde{\mathbf{M}}\|^{2^k}$ for all k . If $\|\mathbf{I}_N - \mathbf{W}_0\widetilde{\mathbf{M}}\| < 1$, we then have a quadratic

convergence [67]. From (2.5), we can clearly see that Newton's iteration requires matrix-to-matrix multiplications whose computational complexity is $\mathcal{O}(N^3)$. Thus, the computational complexity is high. As a matter of fact, its complexity is even higher than that of the direct ZF method when k is large. Thus, direct application of Newton's iteration for matrix inversion is not feasible. In what follows, we propose a method to solve the problem.

Iterating (2.5), we obtain a sequence of matrices $\{\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_k\}$. The relationship between \mathbf{W}_0 and \mathbf{W}_k can be found straightforwardly in

$$\mathbf{W}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} (\mathbf{W}_0 \widetilde{\mathbf{M}})^m \mathbf{W}_0, \quad (2.6)$$

where $\bar{c}_{k,m}$ is the coefficient of the m th summation term in (2.6). The expression in (2.6) can be seen as an expansion form of Newton's iteration, while that in (2.5) an iterative form. It turns out that to obtain a low-complexity algorithm, we have to use the expansion form. Assign $\bar{c}_{k,m}$'s as coefficients of a polynomial function of z , i.e., $f_k(z) = \bar{c}_{k,0}z^0 + \bar{c}_{k,1}z^1 + \dots + \bar{c}_{k,2^k-1}z^{2^k-1}$. Then, the polynomial $f_{k+1}(z)$ can be derived from $f_k(z)$ as $f_{k+1}(z) = 2f_k(z) - z[f_k(z)]^2$, where $f_0(z) = 1$. This is to say that $\bar{c}_{k,m}$ can be recursively calculated. Note that our objective is to obtain the equalized result $\mathbf{W}_k \widetilde{\mathbf{y}}$, not the matrix inverse \mathbf{W}_k itself. Multiplying both sides of (2.6) by $\widetilde{\mathbf{y}}$ and letting $\bar{\mathbf{x}}_k = \mathbf{W}_k \widetilde{\mathbf{y}}$ and $\bar{\mathbf{u}}_m = (\mathbf{W}_0 \widetilde{\mathbf{M}})^m \mathbf{W}_0 \widetilde{\mathbf{y}}$, we have the equalized result as

$$\bar{\mathbf{x}}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \bar{\mathbf{u}}_m. \quad (2.7)$$

From the definition of $\bar{\mathbf{u}}_m$, we can then have the following relationship:

$$\bar{\mathbf{u}}_{m+1} = (\mathbf{W}_0 \widetilde{\mathbf{M}}) \bar{\mathbf{u}}_m. \quad (2.8)$$

As a result, $\bar{\mathbf{u}}_m$ can be recursively calculated as well. Using this approach, we have transformed matrix-to-matrix multiplications in (2.6) into matrix-to-vector multiplications in (2.7) and (2.8).

To complete our low-complexity algorithm, we make use of the special structure inherent in the ICI matrix. From the foregoing derivation, we know that $\widetilde{\mathbf{M}} = \widetilde{\mathbf{H}}_0 + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\widetilde{\mathbf{H}}_1$. Using

this structure, we can then rewrite (2.8) as

$$\begin{aligned}\bar{\mathbf{u}}_{m+1} &= \left[\mathbf{W}_0 \left(\tilde{\mathbf{H}}_0 + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{H}}_1 \right) \right] \bar{\mathbf{u}}_m \\ &= \mathbf{W}_0 \left[\tilde{\mathbf{H}}_0\bar{\mathbf{u}}_m + \mathbf{G}\mathbf{V}_1\mathbf{G}^H \left(\tilde{\mathbf{H}}_1\bar{\mathbf{u}}_m \right) \right].\end{aligned}\quad (2.9)$$

Note that $\tilde{\mathbf{H}}_0$, $\tilde{\mathbf{H}}_1$, and \mathbf{V}_1 are all diagonal matrices. If we further pose a constraint that \mathbf{W}_0 is a diagonal matrix, we can transform matrix-to-matrix operations into vector-to-vector and DFT/IDFT operations as shown in (2.9). As we know, DFTs/IDFTs can be efficiently implemented with FFTs/IFFTs whose complexity is $\mathcal{O}(N \log_2 N)$. As a result, the computational complexity of the proposed algorithm is $\mathcal{O}(N \log_2 N)$. The constraint on \mathbf{W}_0 may not always yield satisfactory performance in all scenarios. Instead of a diagonal matrix, we may let \mathbf{W}_0 be a low-bandwidth banded matrix. Let the (i, j) th entry of a matrix \mathbf{B} be denoted as $\mathbf{B}(i, j)$. The banded matrix is defined as follows. $\mathbf{B}(i, j) \neq 0$, if $|i - j| \leq D$, and $\mathbf{B}(i, j) = 0$, otherwise. Here, D is the bandwidth of the banded matrix. If $D = 0$, the banded matrix is reduced to a diagonal matrix. If $D = 1$, the banded matrix will have three non-zero diagonal vectors. With this type of \mathbf{W}_0 , the computational complexity in (2.9) will only be increased slightly. For later simulations, we will only consider the cases of $D = 0$ and $D = 1$. It turns out that for $D = 1$, the performance of the proposed algorithm is good enough. For easy reference, we denote the proposed low-complexity ZF method as the Newton-ZF (N-ZF) method.

§ 2.2.2 Derivation of the Initial Matrix

In Subsection 2.2.1, we have proposed the N-ZF method to reduce the complexity of the direct ZF method using FFTs. However, we still have to determine the initial matrix \mathbf{W}_0 for the N-ZF method. A good initial matrix can reduce the number of iterations significantly and provide good mitigation performance. As known, the main function of ZF is to invert the ICI matrix, and in the ideal case, $\mathbf{I}_N - \mathbf{W}_k\tilde{\mathbf{M}} = \mathbf{0}_N$, where $\mathbf{0}_N$ is an $N \times N$ zero matrix. As a result, if $\mathbf{I}_N - \mathbf{W}_0\tilde{\mathbf{M}}$ can be made as close to $\mathbf{0}_N$ as possible, fast convergence in Newton's algorithm can be obtained. Based on this idea, we propose to minimize the Frobenius norm of $\mathbf{I}_N - \mathbf{W}_0\tilde{\mathbf{M}}$,

i.e.,

$$\mathbf{W}_0 = \arg \min_{\mathbf{w}} \|\mathbf{I}_N - \mathbf{W}\tilde{\mathbf{M}}\|_F^2, \quad (2.10)$$

where $\|\tilde{\mathbf{R}}\|_F$ means the Frobenius norm of $\tilde{\mathbf{R}}$ and \mathbf{W} is a banded matrix with bandwidth D .

Before the derivation of the optimal solution in (2.10), we first observe a property in a banded matrix. Fig. 2.1 shows an example of a banded initial matrix for $N = 8$ and $D = 1$. In the figure, only the data in the shaded area are non-zeros. Note that the number of the non-zero elements in each row may not be the same. For the 0th and the 7th rows, the number of the non-zero elements is 2. For the rest of rows, the number of the non-zero elements is 3. For a general case, the number of the non-zero elements in the i th row first increases, remains the same, and finally decreases (as i increases). Due to this property, we need to consider the three cases when solving (2.10). Define $\tilde{m}_{i,j} = \tilde{\mathbf{M}}(i,j)$, $w_{i,j} = \mathbf{W}_0(i,j)$, and $a_{i,j} = \sum_{n=0}^{N-1} \tilde{m}_{i,n}^* \tilde{m}_{j,n}$. Differentiating (2.10) with respect to $w_{i,j}^*$ and setting the result to zero, we can

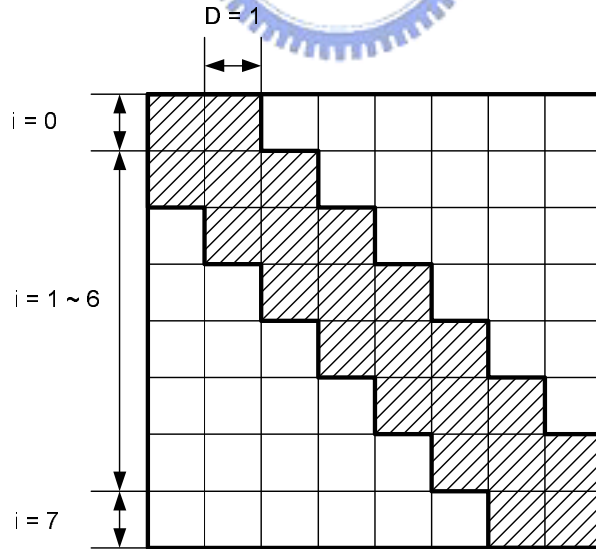


Figure 2.1: An example of the structure of a banded initial matrix for $N = 8$ and $D = 1$. The elements in the shaded area are non-zeros, while the others are zeros.

obtain the following equation:

$$\mathbf{A}_i \mathbf{w}_i = \mathbf{m}_i, \quad i = 0, 1, \dots, N - 1, \quad (2.11)$$

where \mathbf{w}_i consists of the non-zero elements in the i th row vector of the optimum \mathbf{W}_0 . \mathbf{A}_i , \mathbf{w}_i , and \mathbf{m}_i for the above-mentioned three cases are defined as follows:

1. For $i = D, D + 1, \dots, N - 1 - D$,

$$\mathbf{A}_i = \begin{bmatrix} a_{i-D, i-D} & \cdots & a_{i-D, i+D} \\ \vdots & \ddots & \vdots \\ a_{i+D, i-D} & \cdots & a_{i+D, i+D} \end{bmatrix}, \quad (2.12)$$

$$\mathbf{w}_i = [w_{i, i-D}, w_{i, i-D+1}, \dots, w_{i, i+D}]^T, \quad (2.13)$$

$$\mathbf{m}_i = [\tilde{m}_{i-D, i}^*, \tilde{m}_{i-D+1, i}^*, \dots, \tilde{m}_{i+D, i}^*]^T. \quad (2.14)$$

2. For $i = 0, 1, \dots, D - 1$,

$$\mathbf{A}_i = \mathbf{A}_D(0 : D + i, 0 : D + i), \quad (2.15)$$

$$\mathbf{w}_i = [w_{i, 0}, w_{i, 1}, \dots, w_{i, i+D}]^T, \quad (2.16)$$

$$\mathbf{m}_i = [\tilde{m}_{0, i}^*, \tilde{m}_{1, i}^*, \dots, \tilde{m}_{i+D, i}^*]^T, \quad (2.17)$$

where $\mathbf{C}(i_1 : i_2, j_1 : j_2)$ indicates a submatrix of \mathbf{C} , obtained from the i_1 th row to the i_2 th row and from the j_1 th column to the j_2 th column of \mathbf{C} .

3. For $i = N - D, N - D + 1, \dots, N - 1$,

$$\mathbf{A}_i = \mathbf{A}_{N-1-D}(i - N + 1 + D : 2D, i - N + 1 + D : 2D), \quad (2.18)$$

$$\mathbf{w}_i = [w_{i, i-D}, w_{i, i-D+1}, \dots, w_{i, N-1}]^T, \quad (2.19)$$

$$\mathbf{m}_i = [\tilde{m}_{i-D, i}^*, \tilde{m}_{i-D+1, i}^*, \dots, \tilde{m}_{N-1, i}^*]^T. \quad (2.20)$$

Note that \mathbf{A}_i in the second case is an upper left submatrix of \mathbf{A}_D in (2.12), while that in the third case is a lower right submatrix of \mathbf{A}_{N-1-D} in (2.12). Now, we can obtain the optimum

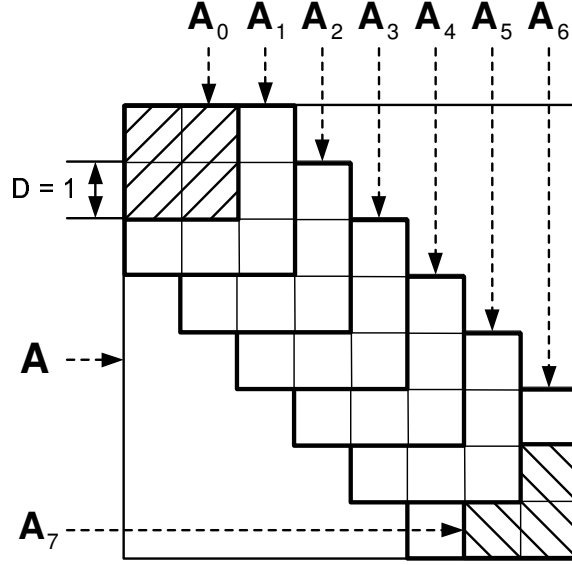


Figure 2.2: An example of \mathbf{A}_i for $N = 8$ and $D = 1$. Note that \mathbf{A}_i overlaps with \mathbf{A}_{i-1} and \mathbf{A}_{i+1} .

solution for (2.10) by $w_i = \mathbf{A}_i^{-1} \mathbf{m}_i$. For clearly understanding the structure of \mathbf{A}_i , we show an example in Fig. 2.2 for $N = 8$ and $D = 1$, where $\mathbf{A}(i, j) = a_{i,j}$. From the figure, we can see that \mathbf{A}_0 is the upper left 2×2 submatrix of \mathbf{A}_1 . For $i = 1, 2, \dots, 5$, the lower right 2×2 submatrix of \mathbf{A}_i is exactly the same as the upper left 2×2 submatrix of \mathbf{A}_{i+1} . The lower right 2×2 submatrix of \mathbf{A}_6 is \mathbf{A}_7 . Using this property, we can obtain a recursive algorithm for fast computation of \mathbf{A}_i^{-1} .

Since $a_{j,i} = a_{i,j}^*$, \mathbf{A}_i is a Hermitian matrix. For $i = D, D + 1, \dots, N - 2 - D$, we can further partition \mathbf{A}_i into the following form

$$\mathbf{A}_i = \begin{bmatrix} s_i & \mathbf{s}_i^H \\ \mathbf{s}_i & \mathbf{U}_i \end{bmatrix}, \quad (2.21)$$

and \mathbf{A}_{i+1} into the following form

$$\mathbf{A}_{i+1} = \begin{bmatrix} \mathbf{U}_i & \tilde{\mathbf{u}}_{i+1} \\ \tilde{\mathbf{u}}_{i+1}^H & \tilde{\mathbf{u}}_{i+1} \end{bmatrix}, \quad (2.22)$$

where s_i and \tilde{u}_{i+1} are scalars, \mathbf{s}_i and $\tilde{\mathbf{u}}_{i+1}$ are column vectors, and \mathbf{U}_i is a square matrix whose dimension is smaller than that of \mathbf{A}_i by one. Since \mathbf{A}_i is a Hermitian matrix, we can write its inverse as

$$\mathbf{A}_i^{-1} = \begin{bmatrix} v_i & \mathbf{v}_i^H \\ \mathbf{v}_i & \mathbf{V}_i \end{bmatrix}, \quad (2.23)$$

where v_i is a scalar, \mathbf{v}_i is a column vector, and \mathbf{V}_i is a square matrix with dimension smaller than that of \mathbf{A}_i^{-1} by one. From the block matrix inversion formula [68], we can obtain \mathbf{A}_{i+1}^{-1} from \mathbf{A}_i^{-1} via the following formula

$$\mathbf{A}_{i+1}^{-1} = \begin{bmatrix} \mathbf{U}_i^{-1} + \tilde{\mathbf{b}}_{i+1} \tilde{\mathbf{b}}_{i+1}^H \tilde{\beta}_{i+1} & \tilde{\mathbf{b}}_{i+1} \tilde{\beta}_{i+1} \\ \tilde{\mathbf{b}}_{i+1}^H \tilde{\beta}_{i+1} & \tilde{\beta}_{i+1} \end{bmatrix}, \quad (2.24)$$

where $\tilde{\beta}_{i+1} = (\tilde{u}_{i+1} - \tilde{\mathbf{u}}_{i+1}^H \mathbf{U}_i^{-1} \tilde{\mathbf{u}}_{i+1})^{-1}$, $\tilde{\mathbf{b}}_{i+1} = -\mathbf{U}_i^{-1} \tilde{\mathbf{u}}_{i+1}$, and

$$\mathbf{U}_i^{-1} = \mathbf{V}_i - \frac{\mathbf{v}_i \mathbf{v}_i^H}{v_i}. \quad (2.25)$$

For $i = 0, 1, \dots, D-1$, \mathbf{A}_{i+1} includes \mathbf{A}_i as its submatrix. We then have

$$\mathbf{A}_{i+1} = \begin{bmatrix} \mathbf{A}_i & \tilde{\mathbf{u}}_{i+1} \\ \tilde{\mathbf{u}}_{i+1}^H & \tilde{u}_{i+1} \end{bmatrix}. \quad (2.26)$$

Consequently, \mathbf{A}_{i+1}^{-1} can be obtained by (2.24), where $\mathbf{U}_i^{-1} = \mathbf{A}_i^{-1}$. For $i = N-D, \dots, N-1$, \mathbf{A}_i becomes a submatrix of \mathbf{A}_{i-1} given by

$$\mathbf{A}_{i-1} = \begin{bmatrix} s_i & \mathbf{s}_i^H \\ \mathbf{s}_i & \mathbf{A}_i \end{bmatrix}. \quad (2.27)$$

Thus, \mathbf{A}_i^{-1} can be obtained with (2.25) as follows,

$$\mathbf{A}_i^{-1} = \mathbf{V}_{i-1} - \frac{\mathbf{v}_{i-1} \mathbf{v}_{i-1}^H}{v_{i-1}}. \quad (2.28)$$

Thus, we only have to conduct one matrix inversion, explicitly, i.e., \mathbf{A}_0^{-1} , and its dimension is $(D+1) \times (D+1)$.

To further reduce the complexity, we can make an approximation when calculating $a_{i,j}$. From the definition, we have $a_{i,j} = \sum_{n=0}^{N-1} \tilde{m}_{i,n}^* \tilde{m}_{j,n}$. We can reduce the number of terms included in the summation. We let $a_{i,j} \approx \sum_{n \in \Omega} \tilde{m}_{i,n}^* \tilde{m}_{j,n}$, where $\Omega = \langle i - S : i + S, N \rangle \cap \langle j - S : j + S, N \rangle$, and S is the number of one-sided ICI terms taken into consideration ($0 \leq S \leq N/2 - 1$). The notation $\langle i : j, N \rangle$ denotes a sequence of $\{i - N \lfloor \frac{i}{N} \rfloor, i + 1 - N \lfloor \frac{i+1}{N} \rfloor, \dots, j - N \lfloor \frac{j}{N} \rfloor\}$ (i and j are integers and $i \leq j$). With this approach, $a_{i,j}$ is approximately evaluated, so is \mathbf{A}_i in (2.11). The value of S then determines the accuracy of the solution in (2.11). A small S can greatly reduce the complexity, but results in low accuracy of the solution. Recall that ICI on a subcarrier mainly comes from a few neighboring subcarriers. As a result, we can always find a small S only affecting the final result slightly. For the determination of S , it depends on the value of ICI; the larger the ICI, the larger S we should use. In our simulations, the largest S we use is two.

As mentioned, if $D = 0$, \mathbf{W}_0 will become a diagonal matrix. In this case, the initial values can be approximated as

$$w_{i,i} \approx \frac{\tilde{m}_{i,i}^*}{\sum_{n \in \Omega'} |\tilde{m}_{i,n}|^2}, \quad (2.29)$$

where $\Omega' = \langle i - S : i + S, N \rangle$. There is an interesting property in (2.29). If we only take the diagonal terms of the ICI matrix into account (i.e., $S = 0$), the initial values will degenerate into the coefficients of the conventional one-tap frequency-domain equalizer (FEQ). If there is no ICI, Newton's iteration with (2.29) will stop after initialization ($k = 0$).

§ 2.2.3 Complexity Analysis

In Subsections 2.2.1 and 2.2.2, we have completed the derivation of the N-ZF method in SISO-OFDM systems. In this subsection, we will analyze the required computational complexity of the N-ZF method, and compare it with that of the PSE and direct ZF methods.

From (2.7) and (2.8), it is clear that the computational complexity of the N-ZF method mainly consists of the following three parts:

1. $\bar{\mathbf{u}}_m$ iteration, where $\bar{\mathbf{u}}_m = (\mathbf{W}_0 \widetilde{\mathbf{M}}) \bar{\mathbf{u}}_{m-1}$ and $\bar{\mathbf{u}}_0 = \mathbf{W}_0 \tilde{\mathbf{y}}$,
2. Banded $\widetilde{\mathbf{M}}$ construction, where $\widetilde{\mathbf{M}} = \mathbf{H}_0 + \mathbf{G} \mathbf{V}_1 \mathbf{G}^H \mathbf{H}_1$,
3. Banded \mathbf{W}_0 calculation.

Since the diagonal and DFT/IDFT structures in $\widetilde{\mathbf{M}}$, $\bar{\mathbf{u}}_m$ can be obtained using (2.9). As a result, we require $[N \log_2 N + (2 + 2D + 3/2)N - D(D + 1)]$ complex multiplications (CMs) and $[2N \log_2(N) + (2D + 1/2)N - D(D + 1)]$ complex additions (CAs). In addition, we need $[(1 + 2D)N - D(D + 1)]$ CMs and $[2DN - D(D + 1)]$ CAs for $\bar{\mathbf{u}}_0 = \mathbf{W}_0 \tilde{\mathbf{y}}$ and $2N$ real additions (RAs) for each $\bar{c}_{k,m} \bar{\mathbf{u}}_m$ in (2.7). As to the construction of the banded $\widetilde{\mathbf{M}}$, we require $(S + 1/2)N$ CMs and $(S + 1/2)N$ CAs. For calculating \mathbf{W}_0 , we need to construct matrices \mathbf{A}_i for $i = 0, 1, \dots, N - 1$ and they require $[(-2D^2 + D + 4SD + S + 1/2)N + (8/3)D^3 - (2/3)D - 4SD^2 - 2SD]$ CMs and $[(S + 4SD - 2D^2 - D)N + (8/3)D^3 + 2D^2 + (1/3)D - 4SD^2 - 2SD]$ CAs. For solving $\mathbf{w}_i = \mathbf{A}_i^{-1} \mathbf{m}_i$, it requires $[(37/2)N - 97/4]$ CMs, $[(7/2)N - 21/4]$ CAs, and $(2N - 2)$ real divisions (RDs) for the case of $D = 1$. For the PSE method, it can be obtained in the same way. Finally, we summarize the required computational complexity for the N-ZF method, the PSE method, and the direct ZF method [69] for a SISO-OFDM system in Table 2.1. We also summarize the computational complexity for calculating the initial matrix in the N-ZF method in Table 2.2.

Table 2.1: Complexity comparison among N-ZF, PSE, and direct ZF methods in a SISO-OFDM system.

Methods	Real multiplications	Real divisions	Real additions
Direct ZF	$\frac{4}{3}N^3 + 7N^2 - \frac{1}{3}N$	$N^2 + N$	$\frac{4}{3}N^3 + \frac{11}{2}N^2 - \frac{23}{6}N$
PSE	$4UN \log_2(N) + (6U + 12)N$	$4N$	$6UN \log_2(N) + (4U + 6)N$
N-ZF ($D = 0$)	$(2^{k+2} - 4)N \log_2(N) + [2^{k+4} + 8S - 8]N$	$2N$	$3(2^{k+1} - 2)N \log_2(N) + [5 \times 2^{k+1} + 8S - 5]N$
N-ZF ($D = 1$)	$(2^{k+2} - 4)N \log_2(N) + [24S + 3 \times 2^{k+3} + 62]N - 24S - 2^{k+3} - 89$	$2N - 2$	$3(2^{k+1} - 2)N \log_2(N) + (24S + 9 \times 2^{k+1} + 31)N - 24S - 2^{k+3} - 45$

Table 2.2: Complexity of the initial matrix calculation for the N-ZF method in a SISO-OFDM system.

Methods	Real multiplications	Real divisions	Real additions
N-ZF ($D = 1$)	$(24S + 74)N - 24S - 89$	$(24S + 39)N - 24S - 45$	$2N - 2$

§ 2.2.4 Performance Analysis

For the proposed N-ZF algorithm, the iteration number is usually preset. Unlike other iterative algorithms, the convergence is not a concern here. The reason we can use a preset iteration number is due to the fast convergence property of Newton's iteration and our good initial values. If the proposed algorithm converges, only a small number of iterations is necessary. On the

other hand, if the proposed algorithm diverges, the preset number of iterations will limit the performance degradation. As a matter of fact, even for divergence cases, we can still have improved SINRs if the iteration number is set properly. We will provide intuitive statements to explain why this is true. It turns out that the determination of the iteration number is simple and straightforward.

Now we start with the analysis of convergence behavior. After that, we will derive theoretical SINRs the proposed algorithm can provide. We first perform the eigenvalue decomposition for $\tilde{\mathbf{R}}_0$ as follows:

$$\tilde{\mathbf{R}}_0 = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}, \quad (2.30)$$

where $\mathbf{U} = [\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1}]$ is a matrix composed of eigenvectors of $\tilde{\mathbf{R}}_0$, and $\mathbf{D} = \text{diag}([\lambda_0, \lambda_1, \dots, \lambda_{N-1}]^T)$ consists of eigenvalues, λ_i 's. We assume that $|\lambda_i| \geq |\lambda_j|$ for $i \leq j$. Since $\tilde{\mathbf{R}}_k = \tilde{\mathbf{R}}_{k-1}^2$, then we can decompose $\tilde{\mathbf{R}}_k$ as

$$\tilde{\mathbf{R}}_k = \mathbf{U}\mathbf{D}^{2^k}\mathbf{U}^{-1}. \quad (2.31)$$

If $|\lambda_0| < 1$, then $\tilde{\mathbf{R}}_k \rightarrow \mathbf{0}_N$ as $k \rightarrow \infty$. Thus, we can have the convergence condition for Newton's iteration as $\rho(\tilde{\mathbf{R}}_0) < 1$, where $\rho(\tilde{\mathbf{R}}_0)$ denotes the spectral radius of $\tilde{\mathbf{R}}_0$; the spectral radius indicates the largest absolute value of all eigenvalues [62]. That is to say, for Newton's iteration to converge, the amplitudes of all eigenvalues of $\tilde{\mathbf{R}}_0$ have to be smaller than one. For a moderate mobile speed, this condition holds for most cases. If not, the number of eigenvalues with amplitudes greater than one is small and their amplitudes does not deviate from one too much. These results can be easily observed from simulations though difficult to be proved theoretically. In what follows, we will first show that even for divergence cases, we may still benefit from Newton's iteration. Let $\mathbf{U}^{-1} = [\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{N-1}]^T$, $|\lambda_i| > 1$ for $i = 0, 1, \dots, P-1$ and $|\lambda_i| < 1$ for $i = P, P+1, \dots, N-1$. By definition, $\tilde{\mathbf{R}}_k = \mathbf{I}_N - \mathbf{W}_k\tilde{\mathbf{M}}$. Then, we can

represent the ICI matrix as

$$\begin{aligned}\mathbf{W}_k \widetilde{\mathbf{M}} &= \mathbf{I}_N - \sum_{i=0}^{N-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T \\ &= \mathbf{I}_N - \sum_{i=0}^{P-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T - \sum_{j=P}^{N-1} \lambda_j^{2^k} \mathbf{u}_j \mathbf{p}_j^T.\end{aligned}\quad (2.32)$$

As for \mathbf{W}_k , we can reformulate it as

$$\begin{aligned}\mathbf{W}_k &= (\mathbf{I}_N + \mathbf{I}_N - \mathbf{W}_{k-1} \widetilde{\mathbf{M}}) \mathbf{W}_{k-1} \\ &= (\mathbf{I}_N + \widetilde{\mathbf{R}}_{k-1}) (\mathbf{I}_N + \widetilde{\mathbf{R}}_{k-2}) \dots (\mathbf{I}_N + \widetilde{\mathbf{R}}_0) \mathbf{W}_0.\end{aligned}\quad (2.33)$$

Using (2.31), we can further express \mathbf{W}_k as

$$\begin{aligned}\mathbf{W}_k &= \mathbf{U} (\mathbf{I}_N + \mathbf{D}^{2^{k-1}}) \mathbf{U}^{-1} \mathbf{U} (\mathbf{I}_N + \mathbf{D}^{2^{k-2}}) \mathbf{U}^{-1} \dots \mathbf{U} (\mathbf{I}_N + \mathbf{D}) \mathbf{U}^{-1} \mathbf{W}_0 \\ &= \sum_{j=0}^{N-1} \left\{ \prod_{i=0}^{k-1} (1 + \lambda_j^{2^i}) \right\} \mathbf{u}_j \mathbf{p}_j^T \mathbf{W}_0 \\ &= \sum_{j=0}^{N-1} \phi_{j,k} \mathbf{u}_j \mathbf{p}_j^T \mathbf{W}_0,\end{aligned}\quad (2.34)$$

where $\phi_{j,k} = \prod_{i=0}^{k-1} (1 + \lambda_j^{2^i})$. With (2.32) and (2.34), the ZF-equalized signal can be expressed as

$$\begin{aligned}\bar{\mathbf{x}}_k &= \mathbf{W}_k \widetilde{\mathbf{M}} \tilde{\mathbf{x}} + \mathbf{W}_k \tilde{\mathbf{z}} \\ &= \tilde{\mathbf{x}} - \sum_{i=0}^{M-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T \tilde{\mathbf{x}} - \sum_{j=M}^{N-1} \lambda_j^{2^k} \mathbf{u}_j \mathbf{p}_j^T \tilde{\mathbf{x}} + \tilde{\mathbf{z}}_k \\ &= \tilde{\mathbf{x}} - \sum_{i=0}^{P-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T \tilde{\mathbf{x}} - \sum_{j=P}^{N-1} \lambda_j^{2^k} \mathbf{u}_j \mathbf{p}_j^T \tilde{\mathbf{x}} + \sum_{j=0}^{N-1} \phi_{j,k} \mathbf{u}_j \mathbf{p}_j^T \tilde{\mathbf{z}}_0,\end{aligned}\quad (2.35)$$

where $\tilde{\mathbf{z}}_k = \mathbf{W}_k \tilde{\mathbf{z}}$. Since the eigenvectors $\{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1}\}$ span the N -dimensional space, we can decompose $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{z}}_0$ using these eigenvectors. Let $\tilde{\mathbf{x}} = \sum_{l=0}^{N-1} \beta_l \mathbf{u}_l$ and $\tilde{\mathbf{z}}_0 = \sum_{l=0}^{N-1} \gamma_l \mathbf{u}_l$, where $[\beta_0, \beta_1, \dots, \beta_{N-1}]^T = \mathbf{U}^{-1} \tilde{\mathbf{y}}$ and $[\gamma_0, \gamma_1, \dots, \gamma_{N-1}]^T = \mathbf{U}^{-1} \tilde{\mathbf{z}}_0$ are the corresponding co-

efficients in the decomposition of $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{z}}_0$, respectively. Then, we can rewrite (2.35) as

$$\begin{aligned}\bar{\mathbf{x}}_k &= \tilde{\mathbf{x}} - \sum_{i=0}^{P-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T \left(\sum_{l=0}^{N-1} \beta_l \mathbf{u}_l \right) - \sum_{j=P}^{N-1} \lambda_j^{2^k} \mathbf{u}_j \mathbf{p}_j^T \left(\sum_{l=0}^{N-1} \beta_l \mathbf{u}_l \right) + \sum_{j=0}^{N-1} \phi_{j,k} \mathbf{u}_j \mathbf{p}_j^T \left(\sum_{l=0}^{N-1} \gamma_l \mathbf{u}_l \right) \\ &= \tilde{\mathbf{x}} - \sum_{i=0}^{P-1} \lambda_i^{2^k} \beta_i \mathbf{u}_i - \sum_{j=P}^{N-1} \lambda_j^{2^k} \beta_j \mathbf{u}_j + \sum_{j=0}^{N-1} \phi_{j,k} \gamma_j \mathbf{u}_j.\end{aligned}\quad (2.36)$$

Let $\bar{\mathbf{x}}_k = [\bar{x}_{k,0}, \bar{x}_{k,1}, \dots, \bar{x}_{k,N-1}]^T$ and $\tilde{\mathbf{x}} = [\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{N-1}]^T$. Thus, the m th subcarrier signal after equalization can be expressed as

$$\bar{x}_{k,m} = \tilde{x}_m + \tilde{f}_{1,k,m} + \tilde{f}_{2,k,m} + \tilde{f}_{3,k,m}, \quad (2.37)$$

where $\tilde{f}_{1,k,m} = -\sum_{i=0}^{P-1} \lambda_i^{2^k} \beta_i \mathbf{u}_i(m)$, $\tilde{f}_{2,k,m} = -\sum_{j=P}^{N-1} \lambda_j^{2^k} \beta_j \mathbf{u}_j(m)$, and $\tilde{f}_{3,k,m} = \sum_{j=0}^{N-1} \phi_{j,k} \gamma_j \mathbf{u}_j(m)$. From (2.37), we can see that the equalized signal suffers from three interference terms. For $\tilde{f}_{1,k,m}$, it will become large when k increases; however, for $\tilde{f}_{2,k,m}$, it will become small when k increases. As for the noise term, $\tilde{f}_{3,k,m}$, its dependence on k is not strong. As mentioned, only a few eigenvalues' amplitudes will be larger than one (i.e., P is small) and their amplitudes often do not deviate from one too much. Then, it is easy to see that the decreasing amount of $\tilde{f}_{2,k,m}$ will be larger than the increasing amount of $\tilde{f}_{1,k,m}$ in the early iteration. Thus, for divergence cases, the interference will decrease first and then increase as the iteration proceeds. If we can stop the iteration before the overall interference increases, we can still have the performance gain even though the iteration diverges eventually. Additionally, we can increase D to make the initial matrix closer to the exact matrix inverse. By this way, P may be minimized and $\tilde{f}_{1,k,m}$ will decrease, which makes the proposed method work better. Because of the fast convergence property of Newton's method, the number of iterations required is small. For example, it can be as small as one or two when $D = 1$. For divergent cases, the overall interference is still decreasing in the first one or two iterations.

Since the performance of an OFDM system depends on each subcarrier SINR, we will analyze the subcarrier SINR of the proposed algorithm in the sequel. From (2.3), we can express the equalized signal as $\mathbf{W}_k \tilde{\mathbf{y}} = \mathbf{T}_k \tilde{\mathbf{x}} + \mathbf{W}_k \tilde{\mathbf{z}}$, where $\mathbf{T}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} (\mathbf{W}_0 \tilde{\mathbf{M}})^{m+1}$ is the

equalized ICI matrix. Ideally, \mathbf{T}_k will be an identity matrix. Let $\tilde{\mathbf{z}} = [\tilde{z}_0, \tilde{z}_1, \dots, \tilde{z}_{N-1}]^T$, $\sigma_{\tilde{x}}^2 = E\{|\tilde{x}_i|^2\}$, $\sigma_{\tilde{z}}^2 = E\{|\tilde{z}_i|^2\}$ ($0 \leq i \leq N-1$), and $\alpha = \sigma_{\tilde{z}}^2/\sigma_{\tilde{x}}^2$. The subcarrier SINR for the proposed method with k iterations in the i th subcarrier, denoted as $\text{SINR}_{k,i}$, can be shown as

$$\begin{aligned} \text{SINR}_{k,i} &= \frac{E\{|t_{i,i}^k \tilde{x}_i|^2\}}{E\{|\sum_{\substack{j=0 \\ j \neq i}}^{N-1} t_{i,j}^k \tilde{x}_j|^2\} + E\{|\sum_{j=0}^{N-1} w_{i,j}^k \tilde{z}_j|^2\}} \\ &= \frac{\sigma_{\tilde{x}}^2 |t_{i,i}^k|^2}{\sigma_{\tilde{x}}^2 \sum_{\substack{j=0 \\ j \neq i}}^{N-1} |t_{i,j}^k|^2 + \sigma_{\tilde{z}}^2 \sum_{j=0}^{N-1} |w_{i,j}^k|^2} \\ &= \frac{|t_{i,i}^k|^2}{\sum_{\substack{j=0 \\ j \neq i}}^{N-1} |t_{i,j}^k|^2 + \alpha \sum_{j=0}^{N-1} |w_{i,j}^k|^2}, \end{aligned} \quad (2.38)$$

where $t_{i,j}^k = \mathbf{T}_k(i, j)$ and $w_{i,j}^k = \mathbf{W}_k(i, j)$. For comparison, we also calculate the SINR in the i th subcarrier before equalization, denoted as SINR_i , as follows:

$$\begin{aligned} \text{SINR}_i &= \frac{E\{|\tilde{m}_{i,i} \tilde{x}_i|^2\}}{E\{|\sum_{\substack{j=0 \\ j \neq i}}^{N-1} \tilde{m}_{i,j} \tilde{x}_j|^2\} + E\{|\tilde{z}_i|^2\}} \\ &= \frac{\sigma_{\tilde{x}}^2 |\tilde{m}_{i,i}|^2}{\sigma_{\tilde{x}}^2 \sum_{\substack{j=0 \\ j \neq i}}^{N-1} |\tilde{m}_{i,j}|^2 + \sigma_{\tilde{z}}^2} \\ &= \frac{|\tilde{m}_{i,i}|^2}{\sum_{\substack{j=0 \\ j \neq i}}^{N-1} |\tilde{m}_{i,j}|^2 + \alpha}. \end{aligned} \quad (2.39)$$

As a result, we must calculate each element in \mathbf{T}_k and \mathbf{W}_k for $\text{SINR}_{k,i}$. To make the following derivation more compact, we rewrite the ICI matrix as $\tilde{\mathbf{M}} = \sum_{r=0}^R \tilde{\mathbf{V}}_r \tilde{\mathbf{H}}_r$, where $\tilde{\mathbf{V}}_0 = \mathbf{I}_N$ and $R = 1$. Since $\tilde{\mathbf{M}} = \sum_{r=0}^R \tilde{\mathbf{V}}_r \tilde{\mathbf{H}}_r$, the equalized ICI matrix \mathbf{T}_k can be expanded as

$$\begin{aligned} \mathbf{T}_k &= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{r_{m,0}=1}^R \mathbf{W}_0 \tilde{\mathbf{V}}_{r_{m,0}} \tilde{\mathbf{H}}_{r_{m,0}} \sum_{r_{m,1}=1}^R \mathbf{W}_0 \tilde{\mathbf{V}}_{r_{m,1}} \tilde{\mathbf{H}}_{r_{m,1}} \dots \sum_{r_{m,m}=1}^R \mathbf{W}_0 \tilde{\mathbf{V}}_{r_{m,m}} \tilde{\mathbf{H}}_{r_{m,m}} \right\} \\ &= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{r_{m,0}=0}^R \sum_{r_{m,1}=0}^R \dots \sum_{r_{m,m}=0}^R \prod_{f=0}^m \mathbf{W}_0 \tilde{\mathbf{V}}_{r_{m,f}} \tilde{\mathbf{H}}_{r_{m,f}} \right\} \\ &= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{r_{m,0}=0}^R \sum_{r_{m,1}=0}^R \dots \sum_{r_{m,m}=0}^R \bar{\mathbf{A}}_m \right\}, \end{aligned} \quad (2.40)$$

where $\bar{\mathbf{A}}_m = \prod_{f=0}^m \mathbf{W}_0 \tilde{\mathbf{V}}_{r_{m,f}} \tilde{\mathbf{H}}_{r_{m,f}}$ and $\bar{\mathbf{A}}_{-1} = \mathbf{I}_N$. Using the same way, we also expand \mathbf{W}_k as

$$\begin{aligned}
\mathbf{W}_k &= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} (\mathbf{W}_0 \tilde{\mathbf{M}})^m \mathbf{W}_0 \\
&= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{r_{m,0}=0}^R \sum_{r_{m,1}=0}^R \cdots \sum_{r_{m,m-1}=0}^R \mathbf{A}_{m-1} \mathbf{W}_0 \right\} \\
&= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{r_{m,0}=0}^R \sum_{r_{m,1}=0}^R \cdots \sum_{r_{m,m-1}=0}^R \bar{\mathbf{B}}_m \right\}, \tag{2.41}
\end{aligned}$$

where $\bar{\mathbf{B}}_m = \bar{\mathbf{A}}_{m-1} \mathbf{W}_0$. We then calculate each element $\bar{\mathbf{A}}_m(i_m, j_m)$. For brevity of presentation, we consider the case of $D = 0$. To compact and simplify the notation, we redefine $\tilde{\mathbf{h}}_r = [\tilde{h}_0^r, \tilde{h}_1^r, \dots, \tilde{h}_{N-1}^r]^T$, and $\tilde{\mathbf{v}}_r = [\tilde{v}_0^r, \tilde{v}_1^r, \dots, \tilde{v}_{N-1}^r]^T$, and let $w_i = w_{i,i}$. Since $\tilde{\mathbf{V}}_{r_{m,f}}$ is a circulant matrix, it can be expressed as $\tilde{\mathbf{V}}_{r_{m,f}} = [\text{cir}(\tilde{\mathbf{v}}_{r_{m,f}})]^T$. Also, from the definition, we have $\tilde{\mathbf{H}}_{r_{m,f}} = \text{diag}(\tilde{\mathbf{h}}_{r_{m,f}})$. For $m = 0$, we have $\bar{\mathbf{A}}_0(i_0, j_0)$ as

$$\begin{aligned}
\bar{\mathbf{A}}_0(i_0, j_0) &\triangleq [\mathbf{W}_0 \tilde{\mathbf{V}}_{r_{0,0}} \tilde{\mathbf{H}}_{r_{0,0}}]_{i_0, j_0} \\
&= w_{i_0} \tilde{v}_{\langle j_0 - i_0, N \rangle}^{r_{0,0}} \tilde{h}_{j_0}^{r_{0,0}}. \tag{2.42}
\end{aligned}$$

Then, we can obtain $\bar{\mathbf{A}}_1 = (\mathbf{W}_0 \tilde{\mathbf{V}}_{r_{1,1}} \tilde{\mathbf{H}}_{r_{1,1}}) \bar{\mathbf{A}}_0$, and $\bar{\mathbf{A}}_2 = (\mathbf{W}_0 \tilde{\mathbf{V}}_{r_{2,2}} \tilde{\mathbf{H}}_{r_{2,2}}) \bar{\mathbf{A}}_1$, accordingly. Repeating this process, we can obtain a formula for $m \geq 1$. For $m \geq 1$, we formulate $\bar{\mathbf{A}}_m(i_m, j_m)$ as

$$\begin{aligned}
\bar{\mathbf{A}}_m(i_m, j_m) &= \sum_{k_{m-1}=0}^{N-1} \sum_{k_{m-2}=0}^{N-1} \cdots \sum_{k_0=0}^{N-1} w_{k_0} w_{k_1} \cdots w_{k_{m-1}} w_{i_m} \tilde{v}_{\langle j_m - k_{m-1}, N \rangle}^{r_{m,m}} \\
&\quad \tilde{v}_{\langle k_{m-1} - k_{m-2}, N \rangle}^{r_{m,m-1}} \cdots \tilde{v}_{\langle k_1 - k_0, N \rangle}^{r_{m,1}} \tilde{v}_{\langle k_0 - i_m, N \rangle}^{r_{m,0}} \tilde{h}_{j_m}^{r_{m,m}} \tilde{h}_{k_{m-1}}^{r_{m,m-1}} \cdots \tilde{h}_{k_0}^{r_{m,0}}. \tag{2.43}
\end{aligned}$$

With $\tilde{\mathbf{M}} = \sum_{r=0}^R \tilde{\mathbf{V}}_r \tilde{\mathbf{H}}_r$, we can expand $\tilde{m}_{i,j}$ as

$$\tilde{m}_{i,j} = \sum_{r=0}^R \tilde{v}_{\langle j-i, N \rangle}^r \tilde{h}_j^r. \tag{2.44}$$

From (2.29) and (2.44), we can express w_i as

$$w_i \approx \frac{(\sum_{r=0}^R \tilde{v}_0^r \tilde{h}_i^r)^*}{\sum_{j \in \Omega'} |\sum_{r=0}^R \tilde{v}_{<j-i, N>}^r \tilde{h}_j^r|^2}. \quad (2.45)$$

Since we have derived \tilde{v}_i^j and \tilde{h}_i^j , we can then calculate w_i . Thus, we have $\bar{\mathbf{A}}_m(i_m, j_m)$. Using (2.40) and $\bar{\mathbf{A}}_m(i_m, j_m)$, we can obtain \mathbf{T}_k . Using the relation of $\bar{\mathbf{B}}_m = \bar{\mathbf{A}}_{m-1} \mathbf{W}_0$, (2.43), and (2.45), we can compute $\bar{\mathbf{B}}_m(i_m, j_m)$ as

$$\bar{\mathbf{B}}_m(i_m, j_m) = \frac{\bar{\mathbf{A}}_{m-1}(i_m, j_m) (\sum_{r=0}^R \tilde{v}_0^r \tilde{h}_{j_m}^r)^*}{\sum_{j \in \Omega''} |\sum_{r=0}^R \tilde{v}_{<j-j_m, N>}^r \tilde{h}_j^r|^2}. \quad (2.46)$$

where $\Omega'' = \langle j_m - S : j_m + S, N \rangle$. From (2.41) and (2.46), we can then calculate \mathbf{W}_k . Having \mathbf{T}_k and \mathbf{W}_k ready, we can finally evaluate $\text{SINR}_{k,i}$ in (2.38). With (2.44), we can further express SINR_i in (2.39) as

$$\text{SINR}_i = \frac{|\sum_{r=0}^R \tilde{v}_0^r \tilde{h}_i^r|^2}{\sum_{\substack{j=0 \\ j \neq i}}^{N-1} |\sum_{r=0}^R \tilde{v}_{<j-i, N>}^r \tilde{h}_j^r|^2 + \alpha}. \quad (2.47)$$

As for the case of $D \neq 0$, it can be derived by the same way.

§ 2.2.5 Simulation Results

In this subsection, we report simulation results to demonstrate the effectiveness of the proposed method. We consider an OFDM system with $N = 128$ and the CP size of 16. The modulation scheme is 16-QAM. The channel length is set to $L = 15$ and the power delay profile is characterized by an exponential function, i.e., $\sigma_l^2 = e^{-l/L} / \sum_{i=0}^{L-1} e^{-i/L}$, where l is the tap index. Each channel tap is generated by Jakes' model [70]. Here, we assume that the channel response is exactly known for the direct ZF method. For the proposed method, the parameters of the LTV channel model are obtained by least-squares (LS) fittings. Define the normalized Doppler frequency as $f_d = f_D N T_s$, where f_D is the maximum Doppler frequency and T_s is the sampling period. Since the N-ZF method with $S = 2$ have the similar performance to that with $S = N/2 - 1$, we set $S = 2$ for all simulations.

First, we evaluate the validity of the analytic output SINRs for the proposed method. Two cases are considered; case 1 meets the convergence condition derived in Subsection 2.2.4, whereas case 2 doesn't. We let $f_d = 0.1$, SNR = 35 dB, $S = 2$, and $D = 0$. Figure 2.3 shows the analytic subcarrier SINRs for case 1. Since the simulated SINRs are identical to the analytic SINRs, they are not shown in the figure. From this figure, we find that each subcarrier exhibits a different SINR due to the characteristic of the frequency-selective channel. Also, subcarrier SINRs are all increased as the number of iterations is increased. The performance of the proposed method with two iterations is similar to that with three iterations. We also can find that the output SINRs of the proposed method are very close to those of the direct ZF method. Fig. 2.4 shows analytic subcarrier SINRs for case 2. We see that even in this divergence case, SINRs are still increased for the first two or three iterations. For the fourth iteration, subcarrier SINRs start to fall. The result for $D = 1$ is similar to that for $D = 0$ except that the required number of iterations is reduced to one or two.

Next, we consider the performance comparison among the proposed and conventional methods. Here, f_d is set to 0.1. Specifically, the bit-error-rate (BER) is used as the performance measure. For the purpose of benchmark, we also show the result of the direct MMSE method, and that of the direct ZF method with $f_d = 0$ (ICI-free). From extensive simulations, we also find that the performance of the PSE method cannot be further enhanced when $U > 2$. For this reason, we only show the result for $U = 2$. Fig. 2.5 shows the BER performance comparison for $D = 0$. As we can see, the performance of the PSE method is limited and has an error floor phenomenon. The N-ZF method outperforms the PSE method even with one iteration only. As mentioned, there is a convergence condition for the PSE method. This condition is totally dependent on the channel. The N-ZF method also has its convergence condition. However, it depends on the initial matrix as well as the channel. It is then possible to reduce the probability of divergence through the determination of \mathbf{W}_0 . This is the main reason why the N-ZF method can outperform the PSE method. The required complexity of the N-ZF method is lower than that of the PSE method (this can be seen later). Moreover, the performance of the N-ZF method

with three iterations can approach to that of the direct ZF method. Here, the performance of the direct ZF method is only slightly worse than that of the direct MMSE method. Figure 2.6 shows the BER performance comparison for $D = 1$. It is obvious that the N-ZF method can quickly approach to the direct ZF method with one or two iterations. Note that the N-ZF method with two iterations can even perform slightly better than the direct ZF method. This behavior is interesting and it needs further discussions. Since the LTV model instead of the Jakes' model is used, one may be curious if the result is due to the modeling error. We then conduct a simulation, in which the Jake's model is used, to answer the question. Fig. 2.7 shows the result. From this figure, we see that the performance of the ZF and MMSE methods using the exact Jakes' channel is almost the same as that of the ZF and MMSE methods using the LTV channel. This indicates the behavior observed in Fig. 2.6 is not due to the modeling error. The reason for the behavior is explained as follows. The N-ZF method only iterates Newton's method two or three times, and it may not converge completely in all cases. As known, the direct ZF method has a noise enhancement problem. It is then possible that the noise enhancement caused by the N-ZF method is smaller. As a result, the performance of the N-ZF method can be better than that of the direct ZF method. If the convergence condition is met and the number of iterations is large enough, the performance of the N-ZF method and the direct ZF method will be the same. This phenomenon has been verified by simulations, but the result is not shown here. Compared to Fig. 2.5, we see that the N-ZF method with $D = 1$ has the better performance than that with $D = 0$, and it can approach to the direct ZF method more quickly.

To test the limitation of all algorithms, we consider a severer case in which $f_d = 0.2$. Figure 2.8 shows the simulation result. From this figure, we see that the N-ZF method can still work, but its performance is degraded since ICI is much larger than that in the previous cases. Also, we can see that the degradation of the MMSE method is smaller, and the performance gap between the ZF and MMSE methods become larger. We also conduct simulations to evaluate the robustness of all algorithms to the variation of the normalized Doppler frequency. Figure 2.9 shows the results for f_d varying from 0 to 0.2. Here, the SNR is set to 30 dB. From this

figure, we can see that the performance of all methods is degraded as the normalized Doppler frequency is increased. Also, the MMSE method is the most robust method while the N-ZF method is the second.

Table 2.3 summarizes the required computational complexity of the direct ZF method, the PSE method, and the N-ZF method for the simulation setting shown above. In this table, the ratios in the parenthesis indicate the number of operations required for the N-ZF method divided by those for the direct ZF and PSE methods. The first ratio is for the direct ZF method and the second one is for the PSE method. From the above simulations, we can say that for $D = 0$, the required iteration number for the N-ZF method is two or three, whereas for $D = 1$, it is one or two. From Table 2.3, we can see that the N-ZF method with $D = 0$ can save tremendous computations compared to the direct ZF method. With two (three) iterations, its multiplication complexity is only 0.007 (0.015) times that of the direct ZF method. As for the case of $D = 1$, it also saves a lot of computations compared to the direct ZF method. We find that the multiplication complexity of the N-ZF method with one (two) iterations is only 0.008 (0.013) times that of the direct ZF method. As to additions, the complexity ratios are similar to those of multiplications. As to divisions, the ratios are 0.016 and 0.015 for $D = 0$ and $D = 1$, respectively. Compared to the PSE method ($U = 2$), the N-ZF method ($k = 1$ and $D = 0$) also has lower complexity and better performance. Its required multiplications (additions, divisions) is 0.85 (0.745, 0.5) times those of the PSE method. For various N , we show the required computational complexity for the direct ZF method and the N-ZF method ($k = 2, 3$) in Figs. 2.10 and 2.11, respectively. In these two figures, RM, RD, and RA denote real multiplication, real division, and real addition, respectively.

Another important property of the proposed N-ZF method is that it can trade the desired performance for the required complexity. However, the direct ZF method doesn't have such a choice. This property will make the N-ZF method a more efficient method since it can adapt itself to various SNR environments. If the operated SNR is not high, the iteration number can be reduced. For example, in Fig. 2.5, it only requires one iteration to approach the direct ZF

method when SNR is less than 25 dB.

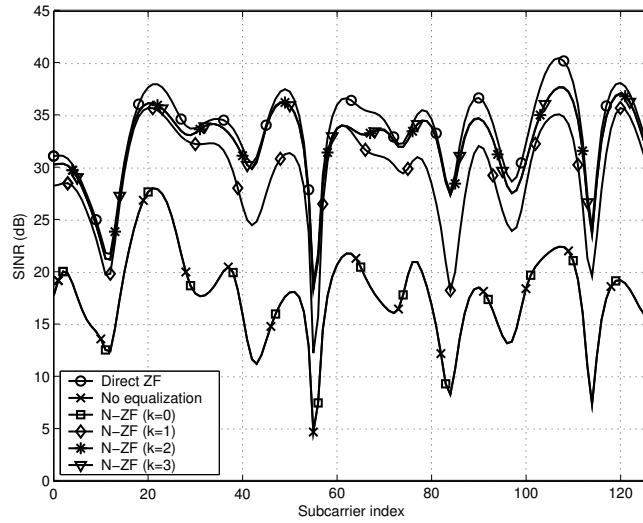


Figure 2.3: SINR analysis of N-ZF method ($D = 0$ and $S = 2$) for case 1, where $f_d = 0.1$ and SNR = 35 dB.

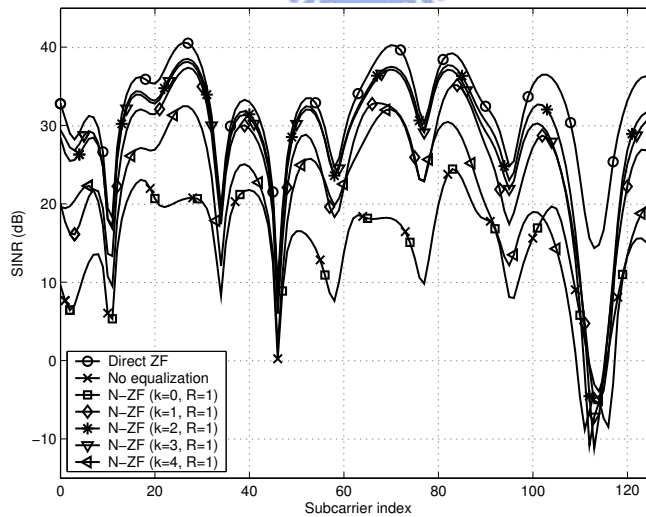
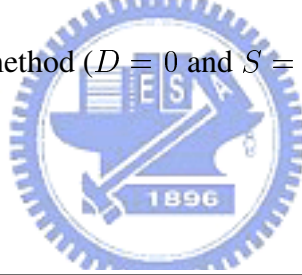


Figure 2.4: SINR analysis of N-ZF method ($D = 0$ and $S = 2$) for case 2, where $f_d = 0.1$ and SNR = 35 dB.

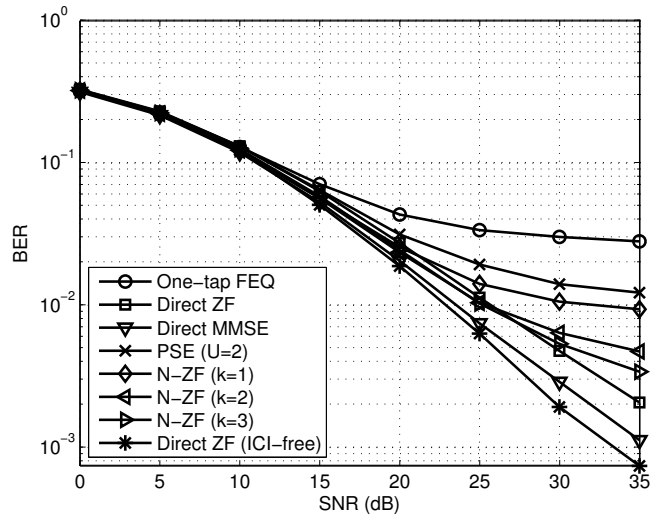


Figure 2.5: BER comparison among one-tap FEQ, PSE, N-ZF ($D = 0$ and $S = 2$), direct ZF, and direct MMSE methods in a SISO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.

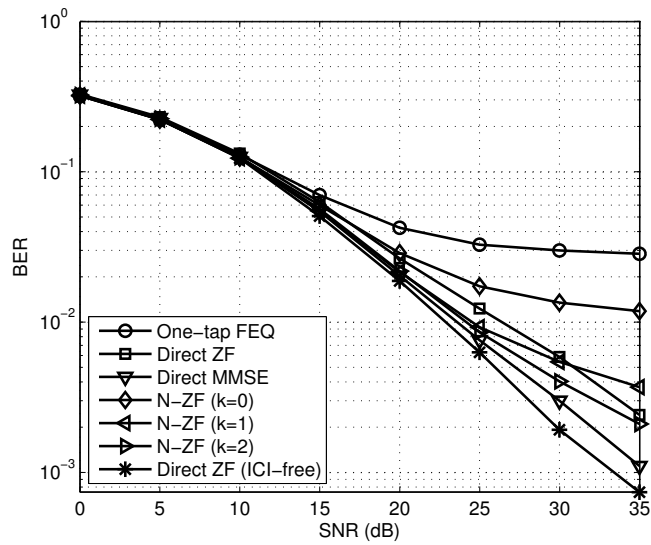


Figure 2.6: BER comparison among one-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a SISO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.

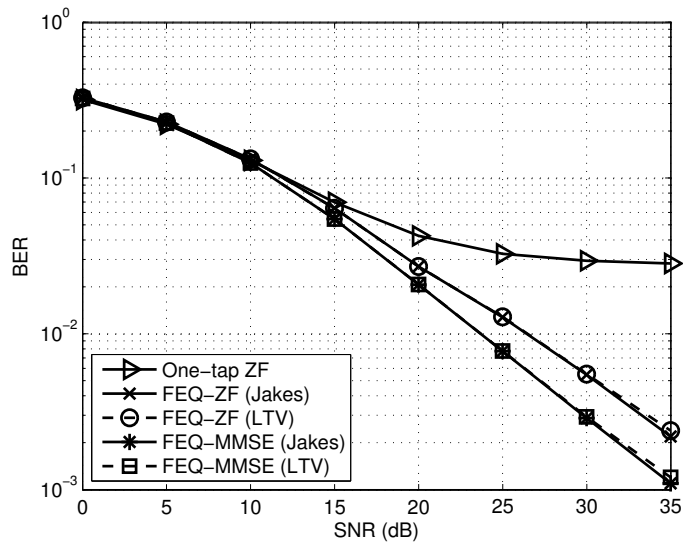


Figure 2.7: BER comparison between the direct ZF and MMSE methods using the exact Jakes and LTV channels in a SISO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.

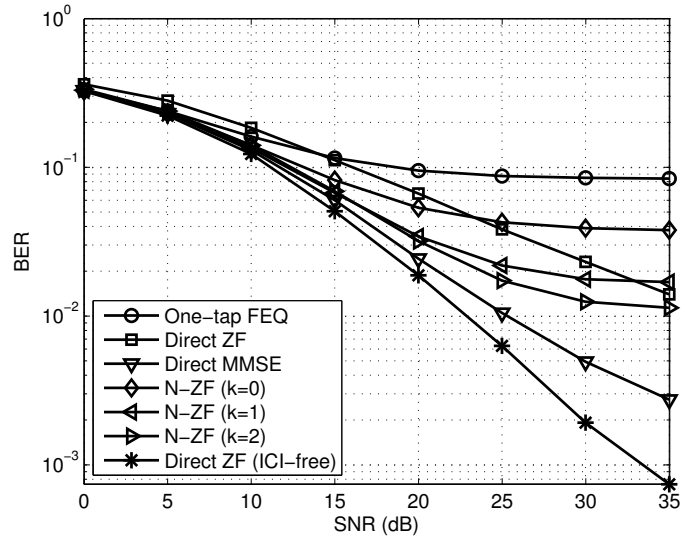


Figure 2.8: BER comparison among one-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a SISO-OFDM system; $f_d = 0.2$ and 16-QAM modulation.

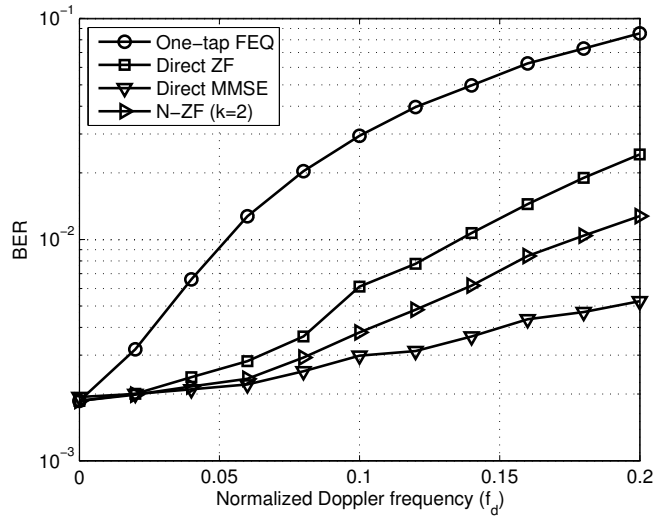


Figure 2.9: BER comparison among one-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a SISO-OFDM system; $f_d = 0 \sim 0.2$, 16 QAM modulation, and SNR = 30 dB.

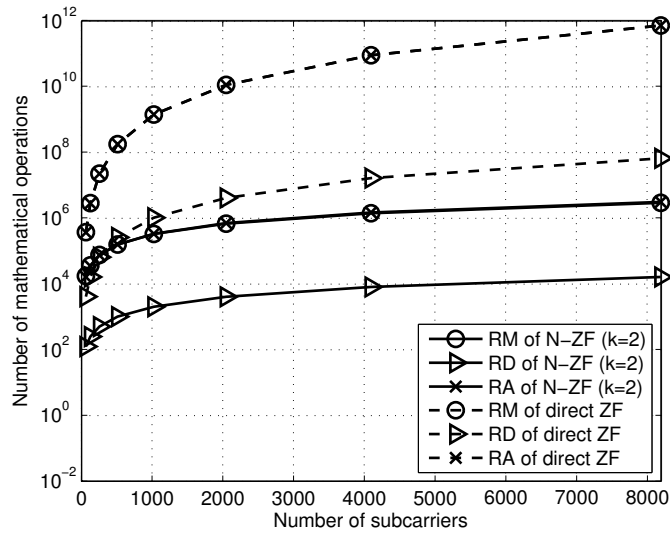


Figure 2.10: Complexity comparison between N-ZF ($D = 1$, $S = 2$, and $k = 2$) and direct ZF methods in a SISO-OFDM system for various N .

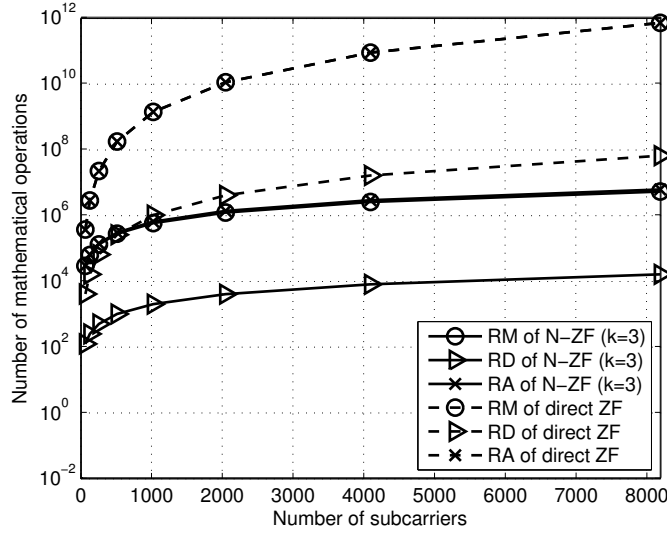


Figure 2.11: Complexity comparison between N-ZF ($D = 1$, $S = 2$, and $k = 3$) and direct ZF methods in a SISO-OFDM system for various N .

§ 2.3 MMSE Method

Except for the ZF method, the MMSE method is another effective ICI mitigation method. The MMSE method minimizes the mean square error (MSE) between the equalized and desired signal vectors. The cost function can be expressed as

$$\min_{\mathbf{F}} E \{ \|\tilde{\mathbf{x}} - \mathbf{F}^H \tilde{\mathbf{y}}\|^2 \}, \quad (2.48)$$

where \mathbf{F}^H is an $N \times N$ matrix. Using the optimum \mathbf{F}^H , we then obtain the optimum estimated signal, denoted as $\bar{\mathbf{x}}_{MMSE}$, shown as

$$\bar{\mathbf{x}}_{MMSE} = \left(\tilde{\mathbf{M}}^H \tilde{\mathbf{M}} + \alpha \mathbf{I}_N \right)^{-1} \tilde{\mathbf{M}}^H \tilde{\mathbf{y}}, \quad (2.49)$$

where $\alpha = \sigma_z^2 / \sigma_x^2$ [17]. From (2.49), it is obvious that the direct MMSE method requires the matrix multiplication and inverse operations. When N is large, the required computational complexity can be very high.

Table 2.3: Complexity comparison among N-ZF, PSE, and direct ZF methods in a SISO-OFDM system ($N = 128$ and $S = 2$).

Methods	Real multiplications (ratio)	Real divisions (ratio)	Real additions (ratio)
Direct ZF	2910848	16512	2885824
PSE ($U = 2$)	10240	512	12544
N-ZF ($D = 0$, $k = 1$)	8704 (0.003, 0.85)	256 (0.016, 0.5)	9344 (0.003, 0.745)
N-ZF ($D = 0$, $k = 2$)	19968 (0.007, 1.95)	256 (0.016, 0.5)	22656 (0.008, 1.806)
N-ZF ($D = 0$, $k = 3$)	42496 (0.015, 4.15)	256 (0.016, 0.5)	49280 (0.017, 3.929)
N-ZF ($D = 1$, $k = 0$)	17007 (0.006, 1.661)	254 (0.015, 0.496)	12315 (0.004, 0.982)
N-ZF ($D = 1$, $k = 1$)	23655 (0.008, 2.310)	254 (0.015, 0.496)	19987 (0.007, 1.593)
N-ZF ($D = 1$, $k = 2$)	36951 (0.013, 3.609)	254 (0.015, 0.496)	35331 (0.012, 2.817)

§ 2.3.1 Proposed Newton-MMSE Method

Now, we know that the MMSE method has a problem of high-complexity. Although existing iterative methods [58] can reduce the complexity of the matrix inversion, the matrix multiplication operation remains. Motivated by this issue, we develop a method to solve the problem.

First, we rewrite the direct MMSE solution with a new form as

$$\begin{aligned}
\bar{\mathbf{x}}_{MMSE} &= \left[\widetilde{\mathbf{M}}^{-H} \left(\widetilde{\mathbf{M}}^H \widetilde{\mathbf{M}} + \alpha \mathbf{I}_N \right) \right]^{-1} \tilde{\mathbf{y}} \\
&= \left(\widetilde{\mathbf{M}} + \alpha \widetilde{\mathbf{M}}^{-H} \right)^{-1} \tilde{\mathbf{y}} \\
&= \widetilde{\mathbf{Q}}^{-1} \tilde{\mathbf{y}},
\end{aligned} \tag{2.50}$$

where $\widetilde{\mathbf{Q}} = \widetilde{\mathbf{M}} + \alpha \widetilde{\mathbf{M}}^{-H}$. As we can see, (2.50) has to conduct the matrix inversion twice. The key idea to avoid the matrix multiplication and inversion is to apply an efficient iterative matrix inversion method twice in (2.50). In Chapter 2, we have shown that the ZF method with Newton's iteration has good performance and its performance is almost as good as that of the direct ZF method. Here, we extend the idea to reduce the computational complexity of the direct MMSE method.

Let the estimated $\widetilde{\mathbf{Q}}^{-1}$ at the k th iteration be \mathbf{W}_k . Then from Newton's iteration, \mathbf{W}_{k+1} can be described as follows [61]– [66]:

$$\mathbf{W}_{k+1} = (2\mathbf{I}_N - \mathbf{W}_k \widetilde{\mathbf{Q}}) \mathbf{W}_k, \quad k = 0, 1, \dots, \infty. \tag{2.51}$$

From (2.51), it is obvious that Newton's iteration requires matrix-to-matrix multiplications. Thus, direct application of Newton's iteration for matrix inversion is not feasible.

As we did in the N-ZF method, we can iterate (2.51) to obtain a sequence of matrices $\{\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_k\}$ and then \mathbf{W}_k can be derived from \mathbf{W}_0 and $\widetilde{\mathbf{Q}}$ in the following form

$$\mathbf{W}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} (\mathbf{W}_0 \widetilde{\mathbf{Q}})^m \mathbf{W}_0, \tag{2.52}$$

where $\bar{c}_{k,m}$ is the coefficient of the m th summation term in (2.52). It turns out that to obtain a low-complexity algorithm, we have to use the expanded form.

Note that the matrix inverse \mathbf{W}_k is not the desired result, whereas $\mathbf{W}_k \tilde{\mathbf{y}}$ is the desired result. Let $\bar{\mathbf{x}}_k = \mathbf{W}_k \tilde{\mathbf{y}}$ and $\bar{\mathbf{v}}_m = (\mathbf{W}_0 \widetilde{\mathbf{Q}})^m \mathbf{W}_0 \tilde{\mathbf{y}}$. Then, multiplying (2.52) by $\tilde{\mathbf{y}}$, we have

$$\bar{\mathbf{x}}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \bar{\mathbf{v}}_m. \tag{2.53}$$

From the definition of $\bar{\mathbf{v}}_m$, it is simple to see that $\bar{\mathbf{v}}_m$ can be recursively calculated as

$$\bar{\mathbf{v}}_{m+1} = \mathbf{W}_0 \tilde{\mathbf{Q}} \bar{\mathbf{v}}_m. \quad (2.54)$$

Recall that $\tilde{\mathbf{Q}} = \tilde{\mathbf{M}} + \alpha \tilde{\mathbf{M}}^{-H}$. Thus, to obtain $\bar{\mathbf{v}}_m$, $\tilde{\mathbf{M}}^{-H}$ must be calculated. Once again, we apply Newton's method described above to expand $\tilde{\mathbf{M}}^{-H}$. Let the estimated $\tilde{\mathbf{M}}^{-H}$ at the p th iteration be \mathbf{B}_p . We then have

$$\mathbf{B}_p = \sum_{j=0}^{2^p-1} \bar{d}_{p,j} \left(\mathbf{W}_0 \tilde{\mathbf{M}}^H \right)^j \mathbf{B}_0, \quad (2.55)$$

where $\bar{d}_{p,j}$ and \mathbf{B}_0 are defined the same as $\bar{c}_{k,m}$ and \mathbf{W}_0 in (2.52), respectively. From (2.54), we then have

$$\bar{\mathbf{v}}_{m+1} = \mathbf{W}_0 \left[\tilde{\mathbf{M}} + \alpha \sum_{j=0}^{2^p-1} \bar{d}_{p,j} \left(\mathbf{W}_0 \tilde{\mathbf{M}}^H \right)^j \mathbf{B}_0 \right] \bar{\mathbf{v}}_m. \quad (2.56)$$

As we can see, the number of the expansion terms in Newton's iteration grows exponentially. Now, we have two iterative processes, i.e., (2.53) and the summation terms in (2.56), the former for the approximation of $\tilde{\mathbf{Q}}^{-1}$ and the latter for the approximation of $\tilde{\mathbf{M}}^{-H}$. The required computational complexity will be high if we fully iterate these two processes. From (2.50), we can see that $\tilde{\mathbf{M}}^{-H}$ is weighted by α which will be much less than one in high SNR scenarios. As a result, the approximation of $\tilde{\mathbf{M}}^{-H}$ is less critical. For simplicity, we only use the first-order expansion for the approximation of $\tilde{\mathbf{M}}^{-H}$ in (2.56), i.e., $p = 1$. From (2.56), we then have

$$\begin{aligned} \bar{\mathbf{v}}_{m+1} &= \mathbf{W}_0 \left[\tilde{\mathbf{M}} + \alpha \left(\bar{d}_{1,0} \mathbf{I}_N + \bar{d}_{1,1} \mathbf{B}_0 \tilde{\mathbf{M}}^H \right) \mathbf{B}_0 \right] \bar{\mathbf{v}}_m \\ &= \mathbf{W}_0 \left[\tilde{\mathbf{M}} + \bar{e}_{1,0} \mathbf{B}_0 + \bar{e}_{1,1} \mathbf{B}_0 \tilde{\mathbf{M}}^H \mathbf{B}_0 \right] \bar{\mathbf{v}}_m, \end{aligned} \quad (2.57)$$

where $\bar{e}_{i,j} = \alpha \times \bar{d}_{i,j}$. Let $\bar{\mathbf{t}}_m = \mathbf{B}_0 \bar{\mathbf{v}}_m$. Exploring the structure of the ICI matrix, we can further rewrite (2.57) as follows:

$$\begin{aligned} \bar{\mathbf{v}}_{m+1} &= \mathbf{W}_0 \left\{ \left(\tilde{\mathbf{H}}_0 + \mathbf{G} \mathbf{V}_1 \mathbf{G}^H \tilde{\mathbf{H}}_1 \right) + \left[\bar{e}_{1,0} \mathbf{I}_N + \bar{e}_{1,1} \mathbf{B}_0 \left(\tilde{\mathbf{H}}_0^H + \tilde{\mathbf{H}}_1^H \mathbf{G} \mathbf{V}_1 \mathbf{G}^H \right) \right] \mathbf{B}_0 \right\} \bar{\mathbf{v}}_m \\ &= \mathbf{W}_0 \left\{ \tilde{\mathbf{H}}_0 \bar{\mathbf{v}}_m + \mathbf{G} \mathbf{V}_1 \left(\mathbf{G}^H \tilde{\mathbf{H}}_1 \bar{\mathbf{v}}_m \right) + \bar{e}_{1,0} \bar{\mathbf{t}}_m + \bar{e}_{1,1} \mathbf{B}_0 \left[\tilde{\mathbf{H}}_0^H \bar{\mathbf{t}}_m + \tilde{\mathbf{H}}_1^H \mathbf{G} \mathbf{V}_1 \left(\mathbf{G}^H \bar{\mathbf{t}}_m \right) \right] \right\}. \end{aligned} \quad (2.58)$$

Note that $\tilde{\mathbf{H}}_0$, $\tilde{\mathbf{H}}_1$, and \mathbf{V}_1 are all diagonal matrices. If we further let \mathbf{W}_0 and \mathbf{B}_0 be diagonal matrices, (2.58) will only involve vector-to-vector and DFT/IDFT operations. It is well-known that DFTs/IDFTs can be efficiently implemented with FFTs/IFFTs, whose complexity is $\mathcal{O}(N \log_2 N)$. Thus, the computational complexity of the proposed MMSE algorithm is $\mathcal{O}(2N \log_2 N)$. The diagonal constraint on \mathbf{W}_0 and \mathbf{B}_0 may not always yield satisfactory performance in all scenarios. To obtain higher performance and at the same time to retain the low-complexity property, we can relax the constraint slightly. We may let \mathbf{W}_0 and \mathbf{B}_0 be low-bandwidth circular band matrices. The structure of a circular band matrix is depicted in Fig. 2.12. Let the (i, j) th entry of a circular band matrix \mathbf{B} be denoted as $b_{i,j} = \mathbf{B}(i, j)$. Given a fixed index i , $b_{i,j}$ is non-zero only for $j \in \{i - D, i - D + 1, \dots, i + D\}$. Here, D is the bandwidth of a circular band matrix. Note that the index, j , is calculated with modulo- N arithmetic. Thus, from Fig. 2.12, we can see that there are non-zero elements in the upper right and lower left corners of a circular band matrix. If $D = 0$, the circular band matrix is reduced to a diagonal matrix. If $D = 1$, the circular band matrix will have three non-zero diagonal vectors. With this type of \mathbf{W}_0 and \mathbf{B}_0 , the computational complexity in (2.58) will only be increased slightly. Thus, we only consider the case where $D = 1$. For easy reference, we denote the proposed low-complexity MMSE method as the Newton-MMSE (N-MMSE) method.

§ 2.3.2 Derivation of the Initial Matrix

In the previous subsection, we have proposed the N-MMSE method to solve the problems of the matrix multiplication and inversion. However, we have to determine the initial matrices \mathbf{W}_0 and \mathbf{B}_0 . Good initial matrices can reduce the number of iterations significantly and provide good mitigation performance.

First, we discuss the determination of \mathbf{W}_0 . As we did in the N-ZF method, we also adopt the minimum-Frobenius-norm criterion to obtain optimal initial matrices. Let \mathbf{W} be a circular

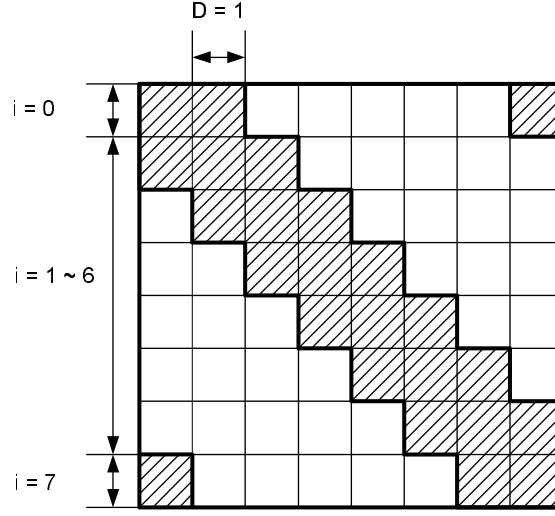


Figure 2.12: The structure of a circular band initial matrix is depicted for $N = 8$ and $D = 1$. The elements in the shaded area are non-zeros, while the others are zeros.

band matrix with bandwidth D_2 . Then \mathbf{W}_0 can be obtained by the following criterion

$$\mathbf{W}_0 = \arg \min_{\mathbf{w}} \|\mathbf{I}_N - \mathbf{W}\tilde{\mathbf{Q}}\|_F^2, \quad (2.59)$$

where $\tilde{\mathbf{Q}} = \tilde{\mathbf{M}} + \alpha\tilde{\mathbf{M}}^{-H}$. Obviously, we have to evaluate $\tilde{\mathbf{M}}^{-H}$ in order to obtain $\tilde{\mathbf{Q}}$. However, $\tilde{\mathbf{M}}^{-H}$ itself is the target we want to find out in (2.55). Since \mathbf{W}_0 is only an initial matrix, it turns out that we only have to calculate $\tilde{\mathbf{M}}^{-H}$ roughly. We will discuss this problem later. For the time being, we can simply assume that $\tilde{\mathbf{Q}}$ is known as *a priori*.

Define a vector consisting of non-zero elements in the i th row of \mathbf{W}_0 as \mathbf{w}_i , i.e.,

$$\mathbf{w}_i = [w_{i,i-D_2}, \dots, w_{i,i}, \dots, w_{i,i+D_2}]^T, \quad (2.60)$$

where $w_{i,j} = \mathbf{W}_0(i, j)$. Furthermore, define $\mathbf{A} = \tilde{\mathbf{Q}}^* \tilde{\mathbf{Q}}^T$ and $\tilde{q}_{i,j} = \tilde{\mathbf{Q}}(i, j)$. Differentiating (2.59) with respect to $w_{i,j}^*$, and setting the corresponding result to zero, we can obtain the following equations:

$$\mathbf{A}_i \mathbf{w}_i = \mathbf{q}_i, \quad i = 0, 1, \dots, N-1, \quad (2.61)$$

where

$$\mathbf{A}_i = \mathbf{A}(i - D_2 : i + D_2, i - D_2 : i + D_2), \quad (2.62)$$

$$\mathbf{q}_i = [\tilde{q}_{i-D_2,i}^*, \dots, \tilde{q}_{i,i}^*, \dots, \tilde{q}_{i+D_2,i}^*]^T. \quad (2.63)$$

Note that the indices in \mathbf{A}_i , \mathbf{w}_i , and \mathbf{q}_i are calculated with modulo- N arithmetic. Now, we can

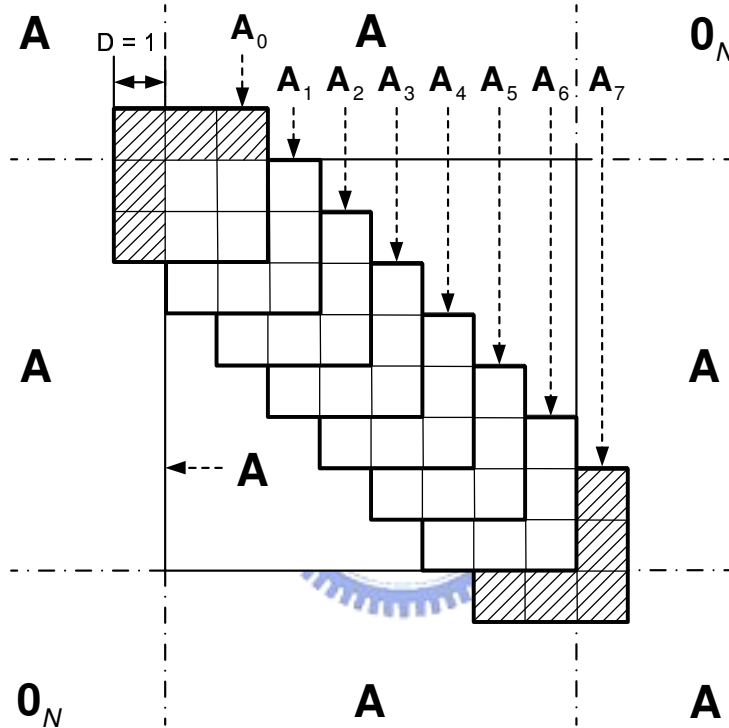


Figure 2.13: The structure of \mathbf{A}_i is presented for $N = 8$ and $D = 1$. Note that \mathbf{A}_i overlaps \mathbf{A}_{i+1} for modulo-8 i (i.e., the relationship is circular).

obtain the optimum solution of (2.59) by $\mathbf{w}_i = \mathbf{A}_i^{-1} \mathbf{q}_i$. For easily understanding the structure of \mathbf{A}_i , we show an example in Fig. 2.13 for $N = 8$ and $D = 1$. From Fig. 2.13, we can see that the lower right 2×2 submatrix of \mathbf{A}_i is exactly the same as the upper left 2×2 submatrix of \mathbf{A}_{i+1} . Note that this relationship is circular. The circular relationship means that the lower right 2×2 submatrix of \mathbf{A}_7 is exactly the same as the upper left 2×2 submatrix of \mathbf{A}_0 . Thus, \mathbf{A}_i overlaps \mathbf{A}_{i+1} for all modulo-8 i . Using this property, we can use the recursive algorithm

mentioned in Subsection 2.2.2 for fast computation of \mathbf{A}_i^{-1} . Consequently, we can obtain \mathbf{A}_i^{-1} recursively. Using this approach, we only have to actually calculate one matrix inversion, i.e., \mathbf{A}_0^{-1} , and its matrix dimension is $(2D_2 + 1) \times (2D_2 + 1)$. Note that as mentioned, \mathbf{A}_i has the circular relationship, we can start out from any i .

To further reduce the computational complexity in the calculation of the initial matrix, we can use a circular band matrix derived from $\tilde{\mathbf{Q}}$ to calculate \mathbf{A} . Define a matrix operation $\text{cirb}(\cdot, \cdot)$ as follows. If $\mathbf{C} = \text{cirb}(\tilde{\mathbf{C}}, D)$, \mathbf{C} is a circular band matrix such that $c_{i,j} = \tilde{c}_{i,j}$ for $j \in \{i - D, i - D + 1, \dots, i + D\}$ and $c_{i,j} = 0$, otherwise. Here, $\tilde{c}_{i,j}$ and $c_{i,j}$ denote the (i, j) th element of $\tilde{\mathbf{C}}(i, j)$ and $\mathbf{C}(i, j)$, respectively. The index, j , is calculated with modulo- N arithmetic. Using the operation, we define $\hat{\mathbf{Q}} = \text{cirb}(\tilde{\mathbf{Q}}, S_2)$ and $\hat{\mathbf{A}} = \hat{\mathbf{Q}}^* \hat{\mathbf{Q}}^T$. Moreover, let $a_{i,j} = \mathbf{A}(i, j)$ and then $a_{i,j} \approx \hat{\mathbf{A}}(i, j) = \sum_{n \in \Gamma} \tilde{q}_{i,n}^* \tilde{q}_{j,n}$, where $\Gamma = \langle i - S_2 : i + S_2, N \rangle \cap \langle j - S_2 : j + S_2, N \rangle$. Here, S_2 is the number of one-sided ICI terms that we want to take into calculation ($0 \leq S_2 \leq N/2 - 1$). Since ICI on a subcarrier mainly comes from a few neighboring subcarriers, we can make such an approximation safely. Note that this approximation is used only for the calculation of the initial matrix.

To obtain \mathbf{B}_0 , we can use the same approach, namely, $\mathbf{B}_0 = \arg \min_{\mathbf{B}} \|\mathbf{I}_N - \mathbf{B}\tilde{\mathbf{M}}^H\|_F^2$. Recall that $\tilde{\mathbf{Q}}$ is needed in (2.59). A simple way to approximate $\tilde{\mathbf{Q}}$ is to use a zeroth-order expansion for $\tilde{\mathbf{Q}}$, i.e., $\tilde{\mathbf{Q}} \approx \tilde{\mathbf{M}} + \alpha \mathbf{B}_0$. The reason why we can use the zeroth-order expansion for $\tilde{\mathbf{Q}}$ is that the precision for initial values can be lower. Thus, its computational complexity can be reduced with limited performance loss. Note that \mathbf{B}_0 is used twice; one is for the iterative step (2.57) and the other is for the initial matrix calculation.

§ 2.3.3 Complexity Analysis

In Subsections 2.3.1 and 2.3.2, we reformulate the MMSE solution and then obtain a low-complexity N-MMSE method. In this subsection, we analyze the required computational complexity of the proposed N-MMSE method and make a comparison between the N-MMSE and direct MMSE methods.

From Subsection 2.3.1, we know that the major computational load results from Eqs. (2.53), (2.54), and (2.56) shown as follows:

1. $\bar{\mathbf{v}}_m$ iteration, where $\bar{\mathbf{v}}_m = (\mathbf{W}_0 \tilde{\mathbf{Q}}) \bar{\mathbf{v}}_{m-1}$ and $\bar{\mathbf{v}}_0 = \mathbf{W}_0 \tilde{\mathbf{y}}$,
2. $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{Q}}$ construction, where $\tilde{\mathbf{Q}} = \left[\tilde{\mathbf{M}} + \alpha \sum_{j=0}^{2^p-1} \bar{d}_{p,j} \left(\mathbf{W}_0 \tilde{\mathbf{M}}^H \right)^j \mathbf{B}_0 \right]$,
3. \mathbf{B}_0 and \mathbf{W}_0 calculation.

Due to the diagonal and DFT/IDFT structures in $\tilde{\mathbf{Q}}$, $\bar{\mathbf{v}}_m$ can be obtained using (2.58). As a result, we require $[2^p N \log_2 N + (2^{p+1} D_1 + 2D_2 + 2^{p+2} + 1/2)N]$ CMs and $[2^p N \log_2 N + (2^{p+1} D_1 + 2D_2 + 2^p - 1/2)N]$ CAs for $m \geq 1$. In addition, we need $(1 + 2D_2)N$ CMs and $2D_2 N$ CAs for $\bar{\mathbf{v}}_0 = \mathbf{W}_0 \tilde{\mathbf{y}}$ and $2N$ RAs for each $\bar{c}_{k,m} \bar{\mathbf{v}}_m$. As to the construction of $\tilde{\mathbf{M}}^H$ for \mathbf{B}_0 , we require $(S_1 + 1/2)N$ CMs and $(S_1 + 1/2)N$ CAs. For calculating \mathbf{B}_0 , we need to construct matrices \mathbf{A}_i for $i = 0, 1, \dots, N-1$ and they require $(S_1 + 4S_1 D_1 - 2D_1^2 + D_1 + 1/2)N$ CMs and $(S_1 + 4S_1 D_1 - 2D_1^2 - D_1)$ CAs. For solving $\mathbf{w}_i = \mathbf{A}_i^{-1} \mathbf{m}_i$, it requires $[(37/2)N + 1/2]$ CMs, $[(7/2)N + 3/2]$ CAs, and $(2N - 1)$ RDs for the case of $D_1 = 1$. To obtain $\tilde{\mathbf{Q}}$, we need $(3/2)N$ CMs and $(3/2)N$ CAs for $D_2 = 1$. For calculating \mathbf{W}_0 , we require $(S_2 + 4S_2 D_2 - 2D_2^2 + D_2 + 1/2)N$ CMs and $(S_2 + 4S_2 D_2 - 2D_2^2 - D_2)$ CAs to construct \mathbf{A}_i for $i = 0, 1, \dots, N-1$. To solve $\mathbf{w}_i = \mathbf{A}_i^{-1} \mathbf{m}_i$, it requires $[(37/2)N + 1/2]$ CMs, $[(7/2)N + 3/2]$ CAs, and $(2N - 1)$ RDs for the case of $D_2 = 1$. As for the direct MMSE method, the matrix inversion can be implemented by Gaussian elimination [69]. Finally, we summarize the required complexity of the N-MMSE method and the direct MMSE method for a SISO-OFDM system in Table 2.4. We also summarize the complexity of calculating the initial matrix in the N-MMSE method in Table 2.5.

Table 2.4: Complexity comparison between the N-MMSE and direct MMSE methods in a SISO-OFDM system.

Methods	Real multiplications	Real divisions	Real additions
Direct MMSE	$\frac{8}{3}N^3 + 5N^2 + 4N \log_2 N + \frac{37}{3}N$	$N^2 + N$	$\frac{8}{3}N^3 + 4N^2 + 6N \log_2 N + \frac{19}{3}N$
N-MMSE ($D_2 = D_1 = 1$)	$(2^{k+2} - 4)2^p N \log_2 N + [24S_1 + 20S_2 + 3 \times 2^{k+p+3} + 3 \times 2^{k+2} - 3 \times 2^{p+3} + 152]N + 2^p + 3$	$4N - 2$	$3(2^{k+1} - 2)2^p N \log_2 N + [24S_1 + 20S_2 + 9 \times 2^{k+p+1} + 5 \times 2^{k+1} - 9 \times 2^{p+1} + 82]N + 8$

Table 2.5: Complexity of the initial matrices calculation for the N-MMSE method in a SISO-OFDM system.

Methods	Real multiplications	Real divisions	Real additions
N-MMSE ($D_1 = D_2 = 1$)	$(24 * S_1 + 20 * S_2 + 152)N + 4$	$(24 * S_1 + 20 * S_2 + 82)N + 8$	$4N - 2$

§ 2.3.4 Simulations

In this subsection, we report simulation results to demonstrate the effectiveness of the proposed N-MMSE method. The simulated system parameters are the same as those in Subsection 2.2.5. We compare the performance of the N-MMSE, one-tap FEQ, and direct MMSE methods. Here, we set $f_d = 0.05$, and use the BER as the performance measure. Fig. 2.14 shows the simulation results. From this figure, we see that the performance of the one-tap FEQ method suffers from an error floor phenomenon. However, the N-MMSE method effectively avoids this phenomenon

and successfully mitigates ICI. With one or two iterations ($k = 1$ or 2), the performance of the N-MMSE method can closely approach to that of the direct MMSE method. Figure 2.15 further shows the BER comparison for $f_d = 0.1$. Due to the larger f_d , the performance of the one-tap FEQ method is further degraded. The N-MMSE method can still work very well. Compared to the direct MMSE method, the N-MMSE method has a little performance loss. However, we can have a significant complexity reduction. We further investigate the impact of f_d on the BER performance. Figure 2.16 shows the results for f_d varying from 0 to 0.2 when the SNR is set to 30 dB. From this figure, we see that the performance of all methods is degraded as the normalized Doppler frequency is increased. Also, the N-MMSE method has larger performance loss (compared to the direct MMSE method) when f_d gets larger.

Table 2.6 summarizes the required computational complexity for the direct MMSE method and the N-MMSE method using the same simulation settings. In this table, the number in a parenthesis shows the ratio of the number of required operations for the N-MMSE method divided by that for the direct MMSE method. As we can see, in the case of $f_d = 0.05$, $k = 1$ or 2 is sufficient for the N-MMSE method. When one (two) iteration is used, the complexity ratio for multiplication, division, and addition is only 0.008 (0.014), 0.031 (0.031) and 0.007 (0.013), respectively. In the case of $f_d = 0.1$, the required iteration number is two or three. When two (three) iterations are used, the ratio for multiplication, division, and addition turns out to be 0.016 (0.027), 0.031 (0.031), and 0.016 (0.028), respectively. From the comparison, we clearly see that the N-MMSE method dose save a lot of computations while its performance is comparable to that of the direct MMSE method. For other values of N , we show the required computational complexity for the direct MMSE method and the N-MMSE method ($k = 2, 3$) in Figs. 2.17 and 2.18, respectively.

Another advantage of the N-MMSE method is that its computational complexity can be easily controlled with the iteration number. In other words, we can always trade the performance for the lower complexity. Also, for different SNR environments, the required number of iterations may be different. If the received SNR is not high, the iteration number can be lower. For

example, in Fig. 2.15, the N-MMSE method only requires one iteration to approach the direct MMSE method when SNR is less than 25 dB. The direct MMSE method, however, does not have this option.

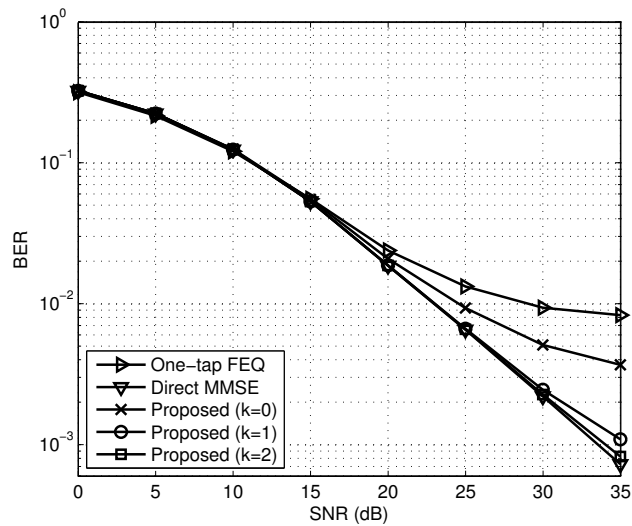


Figure 2.14: BER comparison among the one-tap FEQ, N-MMSE ($\{D_1, D_2\} = \{1, 1\}$ and $\{S_1, S_2\} = \{2, 2\}$), and direct MMSE methods in a SISO-OFDM system; $f_d = 0.05$ and 16-QAM modulation.

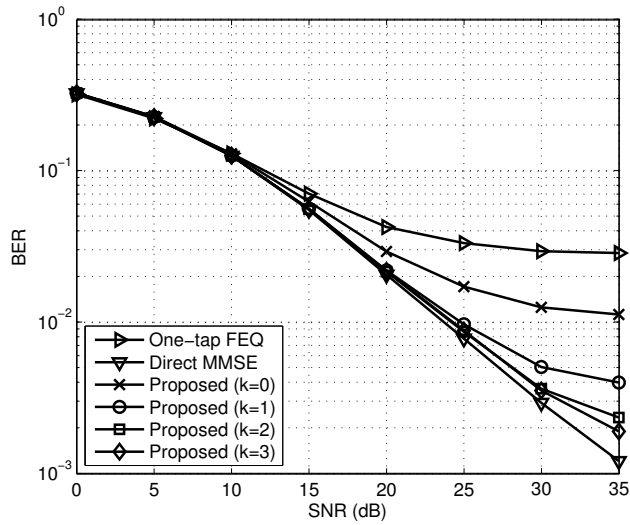


Figure 2.15: BER comparison among the one-tap FEQ, N-MMSE ($\{D_1, D_2\} = \{1, 1\}$ and $\{S_1, S_2\} = \{5, 5\}$), and direct MMSE methods in a SISO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.

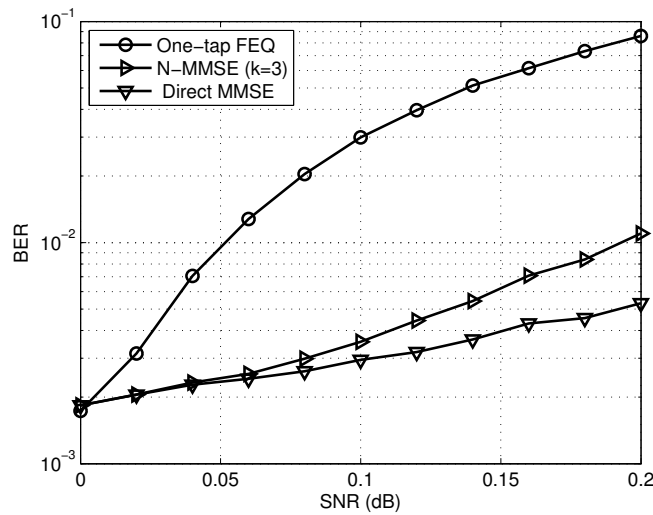


Figure 2.16: BER comparison among one-tap FEQ, N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 5$, and $k = 3$), and direct MMSE methods in a SISO-OFDM system; $f_d = 0 \sim 0.2$, 16 QAM modulation, and SNR = 30 dB.

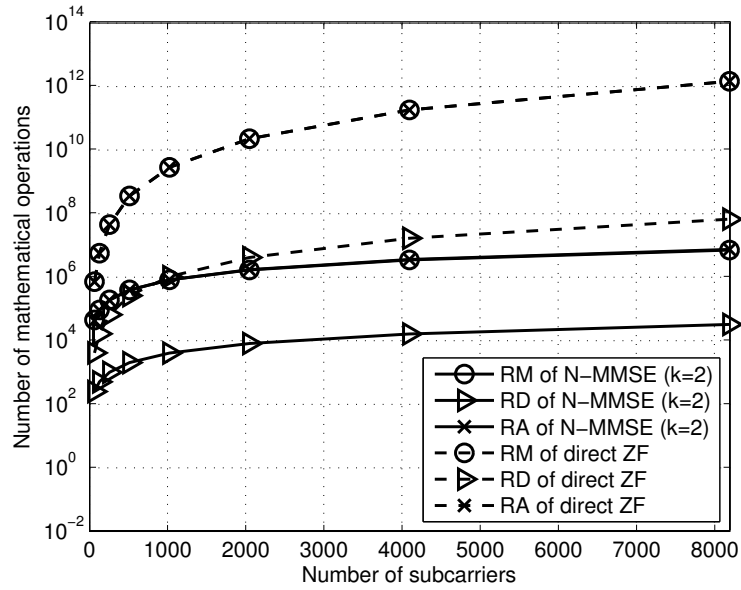


Figure 2.17: Complexity comparison between N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 5$, and $k = 2$) and direct MMSE methods in a SISO-OFDM system for various N .

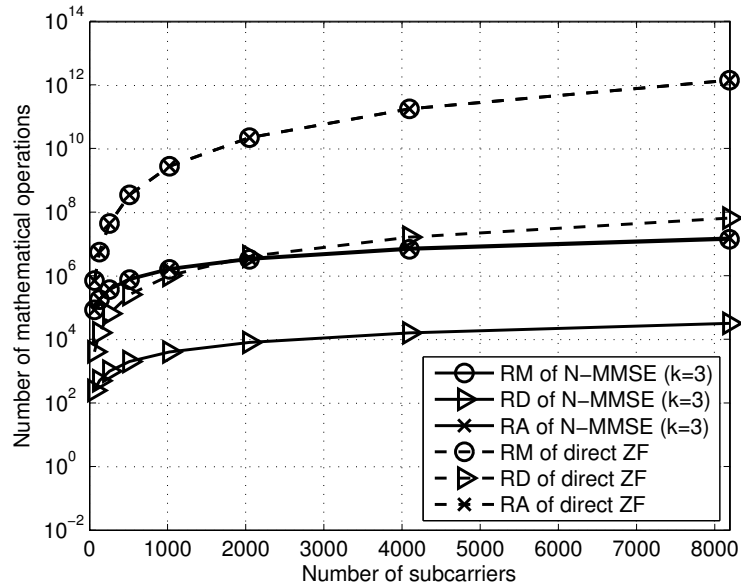


Figure 2.18: Complexity comparison between N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 10$, and $k = 3$) and direct MMSE methods in a SISO-OFDM system for various N .

Table 2.6: Complexity comparison between the N-MMSE and direct MMSE methods in a SISO-OFDM system ($N = 128$, and $\{D_1, D_2\} = \{1, 1\}$).

Methods	Real multiplications (ratio)	Real divisions (ratio)	Real additions (ratio)
Direct MMSE	5679488	16512	5664128
N-MMSE ($k = 1$, $\{S_1, S_2\} = \{2, 2\}$)	47109 (0.008)	510 (0.031)	39688 (0.007)
N-MMSE ($k = 2$, $\{S_1, S_2\} = \{2, 2\}$)	76805 (0.014)	510 (0.031)	72968 (0.013)
N-MMSE ($k = 2$, $\{S_1, S_2\} = \{5, 5\}$)	93701 (0.016)	510 (0.031)	89864 (0.016)
N-MMSE ($k = 3$, $\{S_1, S_2\} = \{5, 5\}$)	153093 (0.027)	510 (0.031)	156424 (0.028)

Chapter 3

Mobility-induced ICI Mitigation for MIMO-OFDM Systems

§ 3.1 Signal Model

The MIMO technique is a promising method to further improve the transmission data rate. Combined with OFDM, a MIMO-OFDM system has been adopted in many systems. In the application of spatial multiplexing, inter-antenna interference is further introduced. Operating in a high-mobility environment, MIMO-OFDM systems will be subject to both ICI and inter-antenna interference. Thus, ICI mitigation is more challenging in MIMO-OFDM systems. It is possible to formulate the whole system with a linear model and to apply the ZF or MMSE equalizer to suppress all interference. However, the dimension of a MIMO-OFDM ICI matrix becomes much larger than that in a SISO-OFDM system and then the required complexity is even more intractable. In this chapter, we extend the method proposed in Chapter 2 to solve the problem. Considering an $M \times M$ system and using the signal model in SISO-OFDM systems, we can express the frequency-domain signal for the i th receive antenna as

$$\tilde{\mathbf{y}}_i = \sum_{j=1}^M \tilde{\mathbf{M}}_{i,j} \tilde{\mathbf{x}}_j + \tilde{\mathbf{z}}_i, \quad (3.1)$$

where $\tilde{\mathbf{y}}_i$ is the frequency-domain signal in the i th receive antenna, $\tilde{\mathbf{x}}_j$ is the frequency-domain signal in the j th transmit antenna, $\tilde{\mathbf{z}}_i$ is the frequency-domain noise in the i th receive antenna, and $\tilde{\mathbf{M}}_{i,j}$ is the ICI matrix for the j th transmit antenna and the i th receive antenna. By stacking all the receive frequency-domain signals from all antennas in a column vector, we have the following signal model

$$\tilde{\mathbf{y}} = \tilde{\mathbf{M}}\tilde{\mathbf{x}} + \tilde{\mathbf{z}}, \quad (3.2)$$

where $\tilde{\mathbf{y}} = [\tilde{\mathbf{y}}_1^T, \tilde{\mathbf{y}}_2^T, \dots, \tilde{\mathbf{y}}_M^T]^T$ is the receive frequency-domain signal, $\tilde{\mathbf{x}} = [\tilde{\mathbf{x}}_1^T, \tilde{\mathbf{x}}_2^T, \dots, \tilde{\mathbf{x}}_M^T]^T$ is the transmit frequency-domain signal, $\tilde{\mathbf{z}} = [\tilde{\mathbf{z}}_1^T, \tilde{\mathbf{z}}_2^T, \dots, \tilde{\mathbf{z}}_M^T]^T$ is the frequency-domain noise, and the frequency-domain ICI channel matrix can be expressed as follows,

$$\tilde{\mathbf{M}} = \begin{bmatrix} \tilde{\mathbf{M}}_{1,1} & \tilde{\mathbf{M}}_{1,2} & \cdots & \tilde{\mathbf{M}}_{1,M} \\ \tilde{\mathbf{M}}_{2,1} & \tilde{\mathbf{M}}_{2,2} & \cdots & \tilde{\mathbf{M}}_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{M}}_{M,1} & \tilde{\mathbf{M}}_{M,2} & \cdots & \tilde{\mathbf{M}}_{M,M} \end{bmatrix}. \quad (3.3)$$

For the ease of description, we only consider a 2×2 MIMO-OFDM system in the following derivation. However, the proposed algorithm (presented later) can be extended to a general $M \times M$ MIMO-OFDM system without any difficulties. Similar to the adopted channel model in SISO-OFDM systems, we also use the LTV model for MIMO-OFDM channels. Thus, we can obtain the MIMO-OFDM ICI matrix given by

$$\begin{aligned} \tilde{\mathbf{M}} &= \begin{bmatrix} \tilde{\mathbf{M}}_{1,1} & \tilde{\mathbf{M}}_{1,2} \\ \tilde{\mathbf{M}}_{2,1} & \tilde{\mathbf{M}}_{2,2} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\mathbf{A}}_0 + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{A}}_1 & \tilde{\mathbf{B}}_0 + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{C}}_0 + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{C}}_1 & \tilde{\mathbf{D}}_0 + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{D}}_1 \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\mathbf{A}}_0 & \tilde{\mathbf{B}}_0 \\ \tilde{\mathbf{C}}_0 & \tilde{\mathbf{D}}_0 \end{bmatrix} + \begin{bmatrix} \mathbf{G}\mathbf{V}_1\mathbf{G}^H & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{G}\mathbf{V}_1\mathbf{G}^H \end{bmatrix} \times \begin{bmatrix} \tilde{\mathbf{A}}_1 & \tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{C}}_1 & \tilde{\mathbf{D}}_1 \end{bmatrix}, \end{aligned} \quad (3.4)$$

where $\tilde{\mathbf{A}}_i$, $\tilde{\mathbf{B}}_i$, $\tilde{\mathbf{C}}_i$, and $\tilde{\mathbf{D}}_i$ play the same role as $\tilde{\mathbf{H}}_i$ in a SISO-OFDM system. Note that the derived signal model is obtained by grouping together subcarrier signals in the same antenna.

§ 3.2 ZF Method

As mentioned, the ZF equalized signal, $\bar{\mathbf{x}}_{ZF}$, can be obtained as $\bar{\mathbf{x}}_{ZF} = \widetilde{\mathbf{M}}^{-1}\tilde{\mathbf{y}}$. Clearly, the direct implementation of the ZF method will require higher computational complexity in MIMO-OFDM systems. For the purpose of comparison, we consider a simple ZF equalizer ignoring the ICI effect in (3.4). In this case, the ICI channel matrix turns out to be

$$\widetilde{\mathbf{M}}_s = \begin{bmatrix} \widetilde{\mathbf{A}}_0 & \widetilde{\mathbf{B}}_0 \\ \widetilde{\mathbf{C}}_0 & \widetilde{\mathbf{D}}_0 \end{bmatrix}. \quad (3.5)$$

Thus, the equalized signal can be obtained with $\bar{\mathbf{x}}_s = \widetilde{\mathbf{M}}_s^{-1}\tilde{\mathbf{y}}$. Using the block matrix inversion formula, we know that to obtain the i th subcarrier equalized signal for antenna one or two, the method will require two weights. For the ease of reference, we denote this equalization method as a two-tap FEQ method.

§ 3.2.1 Proposed Newton-ZF Method

Since the direct ZF method will require higher computational complexity in MIMO-OFDM systems, in this subsection, we will propose a low-complexity ZF method to solve this problem. Using the derived MIMO-OFDM ICI matrix in (3.4), we now can apply Newton's iteration to implement the ZF equalizer. With (2.6) and (3.2), we can obtain the equalized result as

$$\bar{\mathbf{x}}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \bar{\mathbf{v}}_m, \quad (3.6)$$

where $\bar{\mathbf{x}}_k$, $\bar{c}_{k,m}$, and $\bar{\mathbf{v}}_m$ are defined as those in (2.7). Let the initial matrix \mathbf{W}_0 be composed of four $N \times N$ matrices expressed as

$$\mathbf{W}_0 = \begin{bmatrix} \mathbf{W}_\alpha & \mathbf{W}_\beta \\ \mathbf{W}_\gamma & \mathbf{W}_\omega \end{bmatrix}, \quad (3.7)$$

and $\bar{\mathbf{v}}_m = [\bar{\mathbf{v}}_{m,1}^T, \bar{\mathbf{v}}_{m,2}^T]^T$. Recall that $\bar{\mathbf{v}}_{m+1} = \mathbf{W}_0 \widetilde{\mathbf{M}} \bar{\mathbf{v}}_m$. According to this definition of $\bar{\mathbf{v}}_m$, we can obtain $\bar{\mathbf{v}}_{m+1}$ as

$$\begin{aligned} \bar{\mathbf{v}}_{m+1} &= \begin{bmatrix} \mathbf{W}_\alpha & \mathbf{W}_\beta \\ \mathbf{W}_\gamma & \mathbf{W}_\omega \end{bmatrix} \left\{ \begin{bmatrix} \tilde{\mathbf{A}}_0 & \tilde{\mathbf{B}}_0 \\ \tilde{\mathbf{C}}_0 & \tilde{\mathbf{D}}_0 \end{bmatrix} + \begin{bmatrix} \mathbf{G}\mathbf{V}_1\mathbf{G}^H & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{G}\mathbf{V}_1\mathbf{G}^H \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{A}}_1 & \tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{C}}_1 & \tilde{\mathbf{D}}_1 \end{bmatrix} \right\} \begin{bmatrix} \bar{\mathbf{v}}_{m,1} \\ \bar{\mathbf{v}}_{m,2} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{W}_\alpha & \mathbf{W}_\beta \\ \mathbf{W}_\gamma & \mathbf{W}_\omega \end{bmatrix} \left\{ \begin{bmatrix} \tilde{\mathbf{A}}_0 \bar{\mathbf{v}}_{m,1} + \tilde{\mathbf{B}}_0 \bar{\mathbf{v}}_{m,2} \\ \tilde{\mathbf{C}}_0 \bar{\mathbf{v}}_{m,1} + \tilde{\mathbf{D}}_0 \bar{\mathbf{v}}_{m,2} \end{bmatrix} + \begin{bmatrix} \mathbf{G}\mathbf{V}_1\mathbf{G}^H (\tilde{\mathbf{A}}_1 \bar{\mathbf{v}}_{m,1} + \tilde{\mathbf{B}}_1 \bar{\mathbf{v}}_{m,2}) \\ \mathbf{G}\mathbf{V}_1\mathbf{G}^H (\tilde{\mathbf{C}}_1 \bar{\mathbf{v}}_{m,1} + \tilde{\mathbf{D}}_1 \bar{\mathbf{v}}_{m,2}) \end{bmatrix} \right\}. \end{aligned} \quad (3.8)$$

Note that $\tilde{\mathbf{A}}_i$, $\tilde{\mathbf{B}}_i$, $\tilde{\mathbf{C}}_i$, $\tilde{\mathbf{D}}_i$, and \mathbf{V}_1 are all diagonal matrices. It is obvious that if we let \mathbf{W}_α , \mathbf{W}_β , \mathbf{W}_γ , and \mathbf{W}_ω be low-bandwidth banded matrices, (3.8) can be implemented by vector and FFTs/IFFTs operations. Note that the FFT size is N instead of $2N$. Thus, the required computational complexity is $\mathcal{O}(MN \log_2 N)$. It is straightforward to see that the computational complexity of the direct ZF method is $\mathcal{O}(M^3 N^3)$. The complexity reduction achieved by the proposed method in MIMO-OFDM systems can be greater compared to that in SISO-OFDM systems.

Note that the proposed method can be extended to a general $M_T \times M_R$ MIMO-OFDM system, where M_T is the number of transmit antennas, M_R is the number of receive antennas, and $M_T < M_R$. In such a system, we have to use the LS method to mitigate ICI instead of ZF. Using the LS method, we have the equalized signal vector formulated as

$$\begin{aligned} \bar{\mathbf{x}} &= (\widetilde{\mathbf{M}}^H \widetilde{\mathbf{M}})^{-1} \widetilde{\mathbf{M}}^H \tilde{\mathbf{y}} \\ &= \tilde{\mathbf{Q}}^{-1} \bar{\mathbf{y}}, \end{aligned} \quad (3.9)$$

where $\tilde{\mathbf{Q}} = \widetilde{\mathbf{M}}^H \widetilde{\mathbf{M}}$ and $\bar{\mathbf{y}} = \widetilde{\mathbf{M}}^H \tilde{\mathbf{y}}$. The matrix $\tilde{\mathbf{Q}}$, inheriting the properties of $\widetilde{\mathbf{M}}$, consists of diagonal and DFT/IDFT matrices too. As a result, the proposed method discussed above can be applied. Since the derivation is simple and straightforward, it is omitted here. Due to the application of the LS method, the required complexity in this scenario will be somewhat higher.

§ 3.2.2 Derivation of the Initial Matrix

To complete the proposed N-ZF method, we have to determine the $2N \times 2N$ initial matrix, optimally. As we did in SISO-OFDM systems, we adopt the minimum-Frobenius-norm criterion to obtain the initial matrix. Note that the initial matrix for the MIMO-OFDM scenario is no longer a banded matrix. Instead, it is a matrix composed of four banded submatrices. Let $\alpha_{i,j} = \mathbf{W}_\alpha(i,j)$, $\beta_{i,j} = \mathbf{W}_\beta(i,j)$, $\gamma_{i,j} = \mathbf{W}_\gamma(i,j)$, and $\omega_{i,j} = \mathbf{W}_\omega(i,j)$. Also define $q_{i,j} = \sum_{n=0}^{N-1} (\tilde{a}_{i,n}^* \tilde{a}_{j,n} + \tilde{b}_{i,n}^* \tilde{b}_{j,n})$, $r_{i,j} = \sum_{n=0}^{N-1} (\tilde{c}_{i,n}^* \tilde{c}_{j,n} + \tilde{d}_{i,n}^* \tilde{d}_{j,n})$, and $s_{i,j} = \sum_{n=0}^{N-1} (\tilde{a}_{i,n}^* \tilde{c}_{j,n} + \tilde{b}_{i,n}^* \tilde{d}_{j,n})$, where $\tilde{a}_{i,j} = \tilde{\mathbf{M}}_{1,1}(i,j)$, $\tilde{b}_{i,j} = \tilde{\mathbf{M}}_{1,2}(i,j)$, $\tilde{c}_{i,j} = \tilde{\mathbf{M}}_{2,1}(i,j)$, and $\tilde{d}_{i,j} = \tilde{\mathbf{M}}_{2,2}(i,j)$. It turns out that we can obtain the optimum initial values by solving the following equations

$$\mathbf{T}_i \mathbf{b}_i = \mathbf{c}_i, \quad i = 0, 1, \dots, 2N - 1, \quad (3.10)$$

where \mathbf{b}_i consists of the non-zero elements in the i th row of the optimum \mathbf{W}_0 . Definitions of \mathbf{T}_i , \mathbf{b}_i , and \mathbf{c}_i may be different for different i . Note that we have two sets of banded matrices to deal with; one is for $i = 0, 1, \dots, N - 1$ and the other is for $i = N, N + 1, \dots, 2N - 1$. Moreover, we need to consider three cases for each set. Fortunately, they are similar. For the set of $i = 0, 1, \dots, N - 1$, we have the following definitions:

1. For $i = D, D + 1, \dots, N - 1 - D$,

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{S}_i \\ \mathbf{S}_i^H & \mathbf{R}_i \end{bmatrix}, \quad (3.11)$$

$$\mathbf{Q}_i = \begin{bmatrix} q_{i-D, i-D} & \cdots & q_{i-D, i+D} \\ \vdots & \ddots & \vdots \\ q_{i+D, i-D} & \cdots & q_{i+D, i+D} \end{bmatrix}, \quad (3.12)$$

$$\mathbf{b}_i = [\alpha_{i, i-D}, \dots, \alpha_{i, i+D}, \beta_{i, i-D}, \dots, \beta_{i, i+D}]^T, \quad (3.13)$$

$$\mathbf{c}_i = [\tilde{a}_{i-D, i}^*, \dots, \tilde{a}_{i+D, i}^*, \tilde{c}_{i-D, i}^*, \dots, \tilde{c}_{i+D, i}^*]^T. \quad (3.14)$$

Matrices \mathbf{R}_i and \mathbf{S}_i can be obtained by replacing $q_{m,n}$ in \mathbf{Q}_i with $r_{m,n}$ and $s_{m,n}$, respectively.

2. For $i = 0, 1, \dots, D - 1$,

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{S}_i \\ \mathbf{S}_i^H & \mathbf{R}_i \end{bmatrix}, \quad (3.15)$$

$$\mathbf{Q}_i = \mathbf{Q}_D(0 : D + i, 0 : D + i), \quad (3.16)$$

$$\mathbf{R}_i = \mathbf{R}_D(0 : D + i, 0 : D + i), \quad (3.17)$$

$$\mathbf{S}_i = \mathbf{S}_D(0 : D + i, 0 : D + i), \quad (3.18)$$

$$\mathbf{b}_i = [\alpha_{i,0}, \dots, \alpha_{i,i+D}, \beta_{i,0}, \dots, \beta_{i,i+D}]^T, \quad (3.19)$$

$$\mathbf{c}_i = [\tilde{a}_{0,i}^*, \dots, \tilde{a}_{i+D,i}^*, \tilde{c}_{0,i}^*, \dots, \tilde{c}_{i+D,i}^*]^T. \quad (3.20)$$

3. For $i = N - D, N - D + 1, \dots, N - 1$,

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{S}_i \\ \mathbf{S}_i^H & \mathbf{R}_i \end{bmatrix}, \quad (3.21)$$

$$\mathbf{Q}_i = \mathbf{Q}_{N-1-D}(i - N + 1 + D : 2D, i - N + 1 + D : 2D), \quad (3.22)$$

$$\mathbf{R}_i = \mathbf{R}_{N-1-D}(i - N + 1 + D : 2D, i - N + 1 + D : 2D), \quad (3.23)$$

$$\mathbf{S}_i = \mathbf{S}_{N-1-D}(i - N + 1 + D : 2D, i - N + 1 + D : 2D), \quad (3.24)$$

$$\mathbf{b}_i = [\alpha_{i,i-D}, \dots, \alpha_{i,N-1}, \beta_{i,i-D}, \dots, \beta_{i,N-1}]^T, \quad (3.25)$$

$$\mathbf{c}_i = [\tilde{a}_{i-D,i}^*, \dots, \tilde{a}_{N-1,i}^*, \tilde{c}_{i-D,i}^*, \dots, \tilde{c}_{N-1,i}^*]^T. \quad (3.26)$$

Note that \mathbf{Q}_i , \mathbf{R}_i , and \mathbf{S}_i in the second case correspond to upper left submatrices of \mathbf{Q}_D , \mathbf{R}_D , and \mathbf{S}_D , respectively. Also, \mathbf{Q}_i , \mathbf{R}_i , and \mathbf{S}_i in the third case correspond to lower right submatrices of \mathbf{Q}_{N-1-D} , \mathbf{R}_{N-1-D} , and \mathbf{S}_{N-1-D} , respectively.

For $i = N, N + 1, \dots, 2N - 1$, we can use the same equations shown in (3.11)–(3.26). However, we have to let $\mathbf{T}_i = \mathbf{T}_{i-N}$, replace $\alpha_{i,j}$ and $\beta_{i,j}$ in \mathbf{b}_i with $\gamma_{i,j}$ and $\omega_{i,j}$, respectively,

and $\tilde{a}_{i,j}^*$ and $\tilde{c}_{i,j}^*$ in \mathbf{c}_i with $\tilde{b}_{i,j}^*$ and $\tilde{d}_{i,j}^*$, respectively. Since the initial matrix is no longer a banded matrix, we are not able to obtain a recursive relationship in solving (3.10). Gaussian elimination may be a good choice for this problem. As mentioned, $\mathbf{T}_i = \mathbf{T}_{i-N}$; we need only to construct \mathbf{T}_i and perform Gaussian elimination of \mathbf{T}_i for $i = 0, 1, \dots, N - 1$. Also, \mathbf{T}_i is a Hermitian matrix, making the complexity of Gaussian elimination even lower.

To further reduce the computational complexity, we can make some approximations in the calculation of $q_{i,j}$, $r_{i,j}$, and $s_{i,j}$. We can let $q_{i,j} \approx \sum_{n \in \Omega} (\tilde{a}_{i,n}^* \tilde{a}_{j,n} + \tilde{b}_{i,n}^* \tilde{b}_{j,n})$, $r_{i,j} \approx \sum_{n \in \Omega} (\tilde{c}_{i,n}^* \tilde{c}_{j,n} + \tilde{d}_{i,n}^* \tilde{d}_{j,n})$, and $s_{i,j} \approx \sum_{n \in \Omega} (\tilde{a}_{i,n}^* \tilde{c}_{j,n} + \tilde{b}_{i,n}^* \tilde{d}_{j,n})$, where $\Omega = \langle i - S : i + S, N \rangle \cap \langle j - S : j + S, N \rangle$. Again, this approximation makes use of the property that elements in $\tilde{\mathbf{M}}_{i,j}$ close to the main diagonal has larger values than the others.

§ 3.2.3 Complexity Analysis

Up to now, we have extended the N-ZF method from SISO-OFDM systems to MIMO-OFDM systems. In this subsection, we further discuss the required computational complexity of the N-ZF method and compare it with that of the direct ZF method in a 2×2 MIMO-OFDM system.

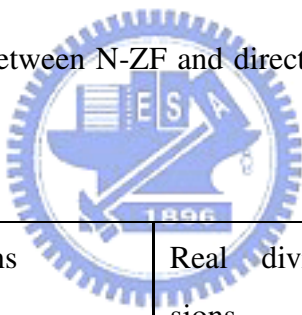
From (3.6) and (3.8), we find that the computational complexity of the N-ZF method mainly consists of the following three parts:

1. $\bar{\mathbf{v}}_m$ iteration, where $\bar{\mathbf{v}}_m = (\mathbf{W}_0 \tilde{\mathbf{M}}) \bar{\mathbf{v}}_{m-1}$ and $\bar{\mathbf{v}}_0 = \mathbf{W}_0 \tilde{\mathbf{y}}$,
2. $\tilde{\mathbf{M}}$ construction,
3. \mathbf{W}_0 calculation.

Since we adopt the LTV channel model, the MIMO-OFDM ICI matrix $\tilde{\mathbf{M}}$ has the special structure of diagonal and DFT/IDFT matrices. Consequently, $\bar{\mathbf{v}}_m$ can be obtained using (3.8). As a result, we require $[2N \log_2 N + (8D + 13)N - 4D^2 - 4D]$ CMs and $[4N \log_2 N + (8D + 7)N - 4D^2 - 4D]$ CAs. In addition, we need $[(8D + 4)N - 4D^2 - 4D]$ CMs and $[(8D + 2)N - 4D^2 - 4D]$ CAs for $\bar{\mathbf{v}}_0 = \mathbf{W}_0 \tilde{\mathbf{y}}$ and $2N$ RAs for each $\bar{\mathbf{c}}_{k,m} \bar{\mathbf{v}}_m$ in (3.6). As to the construction of $\tilde{\mathbf{M}}$,

we require $(4S + 2)N$ CMs and $(4S + 2)N$ CAs. For calculating \mathbf{W}_0 , we need to construct matrices \mathbf{T}_i for $i = 0, 1, \dots, N - 1$ and they require $[(8S + 32SD - 16D^2 + 8D + 4)N - 32SD^2 - 16SD + (64/3)D^3 - (16/3)D]$ CMs and $[(8S + 32SD - 16D^2 - 8D)N - 32SD^2 - 16SD + (64/3)D^3 + 16D^2 + (8/3)D]$ CAs. For solving $\mathbf{T}_i \mathbf{b}_i = \mathbf{c}_i$, it requires $\{[(32/3)D^3 + 44D^2 + (85/3)D + 9/2]N - (34/3)D^4 - (138/3)D^3 - (271/6)D^2 - (21/2)D\}$ CMs, $\{[(32/3)D^3 + 44D^2 + (85/3)D + 9/2]N - (34/3)D^4 - (138/3)D^3 - (271/6)D^2 - (21/2)D\}$ CAs, $[(8D + 4)N + (56/3)D^3 - 4D^2 - (20/3)D]$ CDs, and $[(16D^2 + 12D + 2)N - (40/3)D^3 - 18D^2 - (14/3)D]$ RDs. Finally, we summarize the required complexity for the N-ZF method and the direct ZF method [69] operating in a 2×2 MIMO-OFDM system in Table 3.1. Furthermore, we summarize the complexity for calculating the initial matrices in the N-ZF method in Table 3.2.

Table 3.1: Complexity comparison between N-ZF and direct ZF methods in a 2×2 MIMO-OFDM system.



Methods	Real multiplications	Real divisions	Real additions
Direct ZF	$\frac{32}{3}N^3 + 28N^2 - \frac{2}{3}N$	$4N^2 + 2N$	$\frac{32}{3}N^3 + 22N^2 - \frac{35}{3}N$
N-ZF ($D = 1$)	$(2^{k+3} - 8)N \log_2(N) + (176S + 11 \times 2^{k+3} + \frac{1122}{3})N - 192S - 2^{k+5} - \frac{1020}{3}$	$54N - 20$	$3(2^{k+2} - 4)N \log_2(N) + (176S + 17 \times 2^{k+2} + 322)N - 192S - 2^{k+5} - 316$

Table 3.2: Complexity of the initial matrix calculation for the N-ZF method in a 2×2 MIMO-OFDM system.

Methods	Real multiplications	Real divisions	Real additions
N-ZF ($D = 1$)	$(176S + 414)N - 192S - 340$	$(176S + 338)N - 192S - 316$	$54N - 20$

§ 3.2.4 Simulations

In this subsection, we report simulations to demonstrate the effectiveness of the proposed method. We consider a 2×2 MIMO-OFDM system with $N = 128$ and the CP size of 16. The modulation scheme is 16-QAM. The channel length is set to $L = 15$ and the power delay profile is characterized by an exponential function, i.e., $\sigma_l^2 = e^{-l/L} / \sum_{i=0}^{L-1} e^{-i/L}$, where l is the tap index. Each channel tap is generated by Jakes' model [70]. We assume that the channel response is exactly known for the direct ZF method. For the proposed method, the parameters of the LTV channel model are obtained by LS fittings. Since the N-ZF method with $S = 2$ have the similar performance to that with $S = N/2 - 1$, $S = 2$ is set for all the following simulations.

From the previous simulations in a SISO-OFDM system, we see that the computational complexity of the N-ZF method with $D = 0$ and $D = 1$ is similar when the required number of iterations is taken into account. However, the N-ZF method with $D = 1$ tends to have better results. Thus, we will only consider the N-ZF method with $D = 1$. Here, we consider two environments, i.e., $f_d = 0.05$ and 0.1 . Figure 3.1 depicts the BER performance comparison among the proposed method, the direct ZF method, and the direct MMSE method for $f_d = 0.05$. From this figure, we find that the two-tap FEQ method has an irreducible error floor due to ICI. It is obvious that the N-ZF method can effectively mitigate the ICI and its performance can quickly approach to that of the direct ZF method. As we can see, the iteration number can be as small as one. Also note that the direct MMSE method does not give too much performance

enhancement compared to the direct ZF method. Figure 3.2 illustrates the BER performance for the case of $f_d = 0.1$. The N-ZF method still effectively mitigates the interference and its performance can approach to that of the direct ZF method quickly. In this case, one or two iterations are sufficient for the N-ZF method. Also note that with two iterations, the N-ZF method can outperform the direct ZF method. We further show the BER performance for various f_d . Figure 3.3 shows the results when f_d varies from 0 to 0.2 and the SNR is set to 30 dB. From this figure, we see that all the methods are degraded as f_d becomes larger. Again, the N-ZF method ($k = 2$) has better performance than the direct ZF method.

Table 3.3 summarizes the required computational complexity of the direct ZF method and the N-ZF method. With one iteration, the multiplication (addition) complexity of the N-ZF method ($k = 1$) is only 0.0053 (0.005) times that of the direct ZF method. With two iterations, the complexity ratios of multiplications and additions is 0.0069 and 0.0067. For various N , we show the the required computational complexity of the direct ZF method and the N-ZF method ($k = 2, 3$) in Figs. 3.4 and 3.5, respectively.

Comparing SISO-OFDM in Chapter 2 and MIMO-OFDM systems, we find that the reduction in computational complexity is greater in a MIMO-OFDM system than that in a SISO-OFDM system. In a SISO-OFDM system, the ratio of multiplications is 0.013, while that in a MIMO-OFDM system is 0.0069 ($D = 1$, and $k = 2$). This is because the complexity of the direct ZF method is proportional to $\mathcal{O}(M^3 N^3)$, whereas that of the N-ZF method is in proportion to $\mathcal{O}(MN \log_2 N)$. As a result, we can save more computations as M increases. Also when N gets larger, the reduction in computations becomes more apparent too.

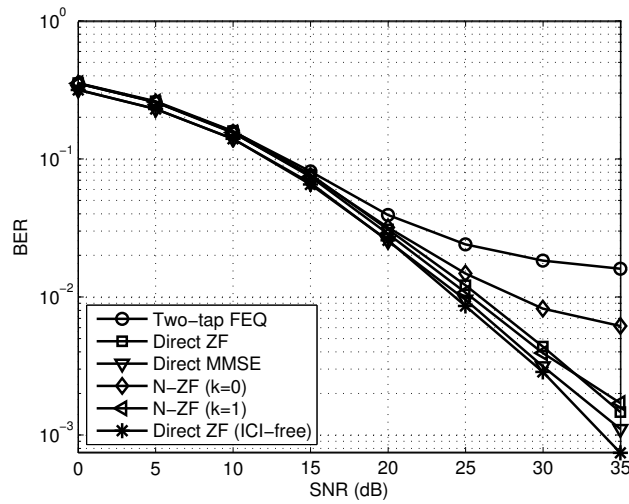


Figure 3.1: BER comparison among two-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0.05$ and 16-QAM modulation.

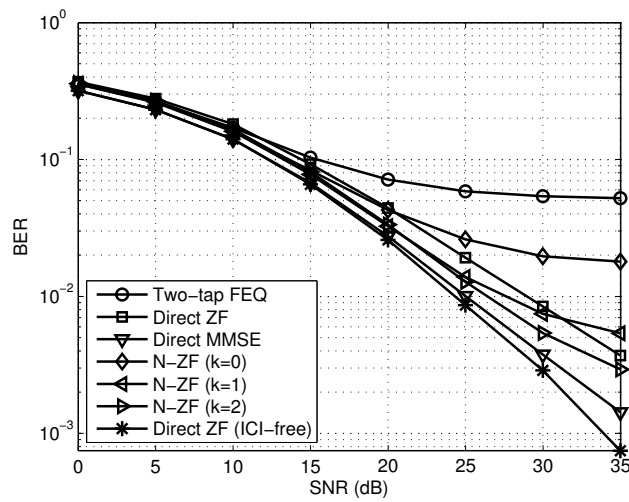


Figure 3.2: BER comparison among two-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.

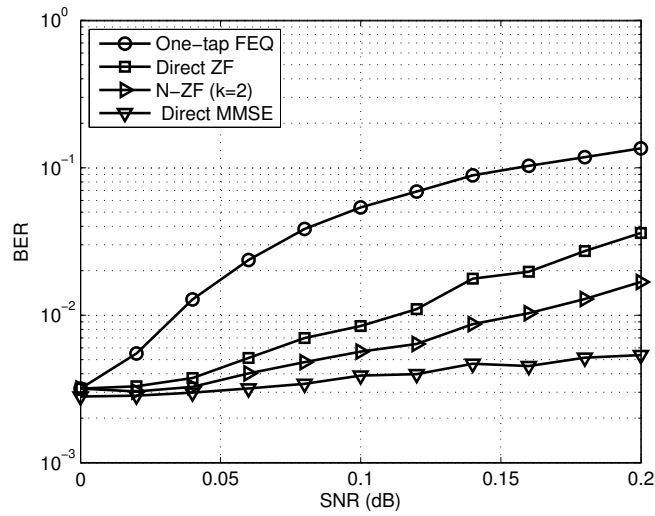


Figure 3.3: BER comparison among one-tap FEQ, N-ZF ($D = 1$ and $S = 2$), direct ZF, and direct MMSE methods in a 2×2 SISO-OFDM system; $f_d = 0 \sim 0.2$, 16 QAM modulation, and SNR = 30 dB.

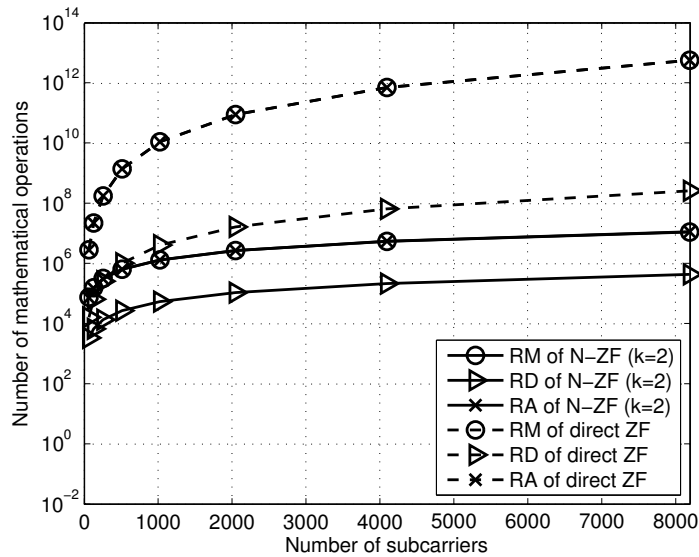


Figure 3.4: Complexity comparison between N-ZF ($D = 1$, $S = 2$, and $k = 2$) and direct ZF methods in a 2×2 MIMO-OFDM system for various N .

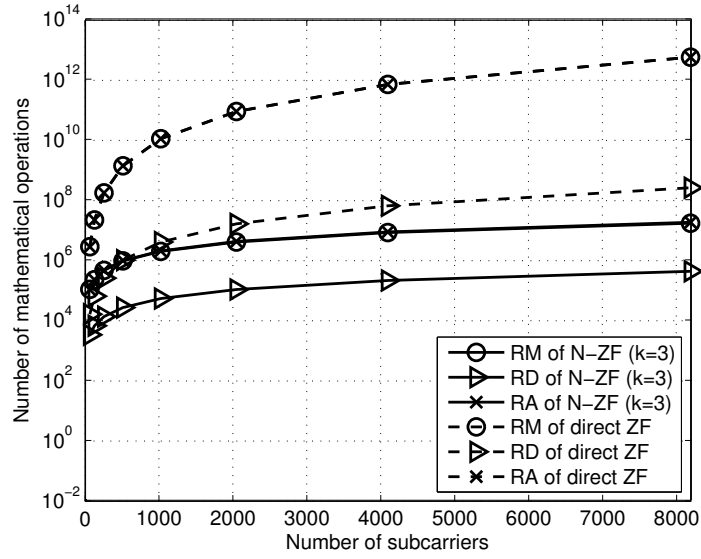


Figure 3.5: Complexity comparison between N-ZF ($D = 1$, $S = 2$, and $k = 3$) and direct ZF methods in a 2×2 MIMO-OFDM system for various N .

Table 3.3: Complexity comparison between N-ZF and direct ZF methods in a 2×2 MIMO-OFDM system ($N = 128$ and $S = 2$).

Methods	Real multiplications (ratio)	Real divisions (ratio)	Real additions (ratio)
Direct ZF	22828288	65792	22728576
N-ZF ($D = 1$, $k = 0$)	103436 (0.0045)	6892 (0.1048)	94244 (0.0041)
N-ZF ($D = 1$, $k = 1$)	121836 (0.0053)	6892 (0.1048)	113668 (0.0050)
N-ZF ($D = 1$, $k = 2$)	158636 (0.0069)	6892 (0.1048)	152516 (0.0067)

§ 3.3 MMSE Method

§ 3.3.1 Proposed Newton-MMSE Method

As mentioned, the MMSE solution can be expressed as $\bar{\mathbf{x}}_{MMSE} = \left(\widetilde{\mathbf{M}}^H \widetilde{\mathbf{M}} + \alpha \mathbf{I}_N \right)^{-1} \widetilde{\mathbf{M}}^H \tilde{\mathbf{y}}$. From this formula, it is clear that the computational complexity is even more intractable in MIMO-OFDM systems since the system dimension becomes larger. This situation may limit its application. In view of this problem, we are devoted to proposing a method which will make the MMSE method applicable in real-world implementation.

As we did in the SISO-OFDM system, the MMSE solution can be expressed in another form as $\bar{\mathbf{x}}_{MMSE} = \left(\widetilde{\mathbf{M}} + \alpha \widetilde{\mathbf{M}}^{-H} \right)^{-1} \tilde{\mathbf{y}}$. Using the derived MIMO-OFDM ICI matrix in (3.4), we now apply Newton's iteration to facilitate the MMSE method in a 2×2 MIMO-OFDM system. With (2.52) and (3.2), we can obtain the equalized signal as

$$\bar{\mathbf{x}}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \bar{\mathbf{v}}_m, \quad (3.27)$$

where $\bar{\mathbf{x}}_k$, $\bar{c}_{k,m}$, and $\bar{\mathbf{v}}_m$ are defined the same as those in (2.53). Note that in this case, $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{v}}_m$ are $2N \times 1$ column vectors. Let the initial matrix \mathbf{W}_0 be composed of four $N \times N$ circular band matrices formulated as

$$\mathbf{W}_0 = \begin{bmatrix} \mathbf{W}_\alpha & \mathbf{W}_\beta \\ \mathbf{W}_\gamma & \mathbf{W}_\omega \end{bmatrix}. \quad (3.28)$$

We also let \mathbf{B}_0 have the same structure as \mathbf{W}_0 . Thus, \mathbf{B}_0 is expressed as

$$\mathbf{B}_0 = \begin{bmatrix} \mathbf{B}_\alpha & \mathbf{B}_\beta \\ \mathbf{B}_\gamma & \mathbf{B}_\omega \end{bmatrix}. \quad (3.29)$$

Moreover, we partition $\bar{\mathbf{v}}_m$ into $\bar{\mathbf{v}}_m = [\bar{\mathbf{v}}_{m,1}^T, \bar{\mathbf{v}}_{m,2}^T]^T$, where $\bar{\mathbf{v}}_{m,1}$ and $\bar{\mathbf{v}}_{m,2}$ are $N \times 1$ column vectors. Define $\bar{\mathbf{t}}_{m,1} = \mathbf{B}_\alpha \bar{\mathbf{v}}_{m,1} + \mathbf{B}_\beta \bar{\mathbf{v}}_{m,2}$, $\bar{\mathbf{t}}_{m,2} = \mathbf{B}_\gamma \bar{\mathbf{v}}_{m,1} + \mathbf{B}_\omega \bar{\mathbf{v}}_{m,2}$, $\bar{\mathbf{u}}_{m,1} = \mathbf{G} \mathbf{V}_1 \mathbf{G}^H \bar{\mathbf{t}}_{m,1}$,

and $\bar{\mathbf{u}}_{m,2} = \mathbf{G}\mathbf{V}_1\mathbf{G}^H\bar{\mathbf{t}}_{m,2}$. With the definition of $\bar{\mathbf{v}}_m, \bar{\mathbf{v}}_{m+1}$ can be decomposed as follows:

$$\begin{aligned}
\bar{\mathbf{v}}_{m+1} &= \begin{bmatrix} \mathbf{W}_\alpha & \mathbf{W}_\beta \\ \mathbf{W}_\gamma & \mathbf{W}_\omega \end{bmatrix} \left\{ \begin{bmatrix} \tilde{\mathbf{A}}_0 & \tilde{\mathbf{B}}_0 \\ \tilde{\mathbf{C}}_0 & \tilde{\mathbf{D}}_0 \end{bmatrix} + \begin{bmatrix} \mathbf{G}\mathbf{V}_1\mathbf{G}^H & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{G}\mathbf{V}_1\mathbf{G}^H \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{A}}_1 & \tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{C}}_1 & \tilde{\mathbf{D}}_1 \end{bmatrix} \right. \\
&+ \begin{bmatrix} \bar{e}_{1,0} & \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{I}_N \end{bmatrix} \\ \bar{e}_{1,1} & \begin{bmatrix} \mathbf{B}_\alpha & \mathbf{B}_\beta \\ \mathbf{B}_\gamma & \mathbf{B}_\omega \end{bmatrix} \end{bmatrix} \left(\begin{bmatrix} \tilde{\mathbf{A}}_0^H & \tilde{\mathbf{C}}_0^H \\ \tilde{\mathbf{B}}_0^H & \tilde{\mathbf{D}}_0^H \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{A}}_1^H & \tilde{\mathbf{C}}_1^H \\ \tilde{\mathbf{B}}_1^H & \tilde{\mathbf{D}}_1^H \end{bmatrix} \right. \\
&\left. \left. \begin{bmatrix} \mathbf{G}\mathbf{V}_1\mathbf{G}^H & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{G}\mathbf{V}_1\mathbf{G}^H \end{bmatrix} \right) \begin{bmatrix} \mathbf{B}_\alpha & \mathbf{B}_\beta \\ \mathbf{B}_\gamma & \mathbf{B}_\omega \end{bmatrix} \right\} \begin{bmatrix} \bar{\mathbf{v}}_{m,1} \\ \bar{\mathbf{v}}_{m,2} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{W}_\alpha & \mathbf{W}_\beta \\ \mathbf{W}_\gamma & \mathbf{W}_\omega \end{bmatrix} \left\{ \begin{bmatrix} \tilde{\mathbf{A}}_0\bar{\mathbf{v}}_{m,1} + \tilde{\mathbf{B}}_0\bar{\mathbf{v}}_{m,2} \\ \tilde{\mathbf{C}}_0\bar{\mathbf{v}}_{m,1} + \tilde{\mathbf{D}}_0\bar{\mathbf{v}}_{m,2} \end{bmatrix} + \begin{bmatrix} \mathbf{G}\mathbf{V}_1\mathbf{G}^H(\tilde{\mathbf{A}}_1\bar{\mathbf{v}}_{m,1} + \tilde{\mathbf{B}}_1\bar{\mathbf{v}}_{m,2}) \\ \mathbf{G}\mathbf{V}_1\mathbf{G}^H(\tilde{\mathbf{C}}_1\bar{\mathbf{v}}_{m,1} + \tilde{\mathbf{D}}_1\bar{\mathbf{v}}_{m,2}) \end{bmatrix} + \bar{e}_{1,0} \right. \\
&\left. \begin{bmatrix} \bar{\mathbf{t}}_{m,1} \\ \bar{\mathbf{t}}_{m,2} \end{bmatrix} + \bar{e}_{1,1} \begin{bmatrix} \mathbf{B}_\alpha & \mathbf{B}_\beta \\ \mathbf{B}_\gamma & \mathbf{B}_\omega \end{bmatrix} \left(\begin{bmatrix} \tilde{\mathbf{A}}_0^H\bar{\mathbf{t}}_{m,1} + \tilde{\mathbf{C}}_0^H\bar{\mathbf{t}}_{m,2} \\ \tilde{\mathbf{B}}_0^H\bar{\mathbf{t}}_{m,1} + \tilde{\mathbf{D}}_0^H\bar{\mathbf{t}}_{m,2} \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{A}}_1^H\bar{\mathbf{u}}_{m,1} + \tilde{\mathbf{C}}_1^H\bar{\mathbf{u}}_{m,2} \\ \tilde{\mathbf{B}}_1^H\bar{\mathbf{u}}_{m,1} + \tilde{\mathbf{D}}_1^H\bar{\mathbf{u}}_{m,2} \end{bmatrix} \right) \right\}. \tag{3.30}
\end{aligned}$$

Note that $\tilde{\mathbf{A}}_i, \tilde{\mathbf{B}}_i, \tilde{\mathbf{C}}_i, \tilde{\mathbf{D}}_i$, and \mathbf{V}_1 are all diagonal matrices. It is obvious that if $\mathbf{W}_\alpha, \mathbf{W}_\beta, \mathbf{W}_\gamma, \mathbf{W}_\omega, \mathbf{B}_\alpha, \mathbf{B}_\beta, \mathbf{B}_\gamma$, and \mathbf{B}_ω are assumed to be low-bandwidth circular band matrices, (3.30) can be implemented with vector and FFT/IFFT operations. Note that the FFT size is N instead of $2N$. Thus, the required computational complexity is $\mathcal{O}(2MN \log_2 N)$. It is straightforward to see that the computational complexity of the direct MMSE method in this case is $\mathcal{O}(M^3N^3)$. As a result, the complexity reduction in MIMO-OFDM systems can be greater than that in SISO-OFDM systems. This will be verified with simulations.

§ 3.3.2 Derivation of the Initial Matrix

To complete the proposed algorithm, we have to determine the two $2N \times 2N$ initial matrices, \mathbf{W}_0 and \mathbf{B}_0 , optimally. As we did before, we again use the minimum-Frobenius-norm criterion. Note that the initial matrices in the MIMO-OFDM scenario are not circular band matrices anymore. Instead, they are matrices composed of four circular band submatrices. We first discuss \mathbf{W}_0 . As for \mathbf{B}_0 , it can be obtained in the same way. Let each circular band submatrix of

\mathbf{W}_0 have bandwidth D_2 . Then define $\alpha_{i,j} = \mathbf{W}_\alpha(i,j)$, $\beta_{i,j} = \mathbf{W}_\beta(i,j)$, $\gamma_{i,j} = \mathbf{W}_\gamma(i,j)$, and $\omega_{i,j} = \mathbf{W}_\omega(i,j)$. We partition $\tilde{\mathbf{Q}}$ into the following format

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \tilde{\mathbf{Q}}_{1,1} & \tilde{\mathbf{Q}}_{1,2} \\ \tilde{\mathbf{Q}}_{2,1} & \tilde{\mathbf{Q}}_{2,2} \end{bmatrix}, \quad (3.31)$$

where $\tilde{\mathbf{Q}}_{i,j}$ is an $N \times N$ submatrix of $\tilde{\mathbf{Q}}$. Then we define three matrices as $\mathbf{Q} = \tilde{\mathbf{Q}}_{1,1}^* \tilde{\mathbf{Q}}_{1,1}^T + \tilde{\mathbf{Q}}_{1,2}^* \tilde{\mathbf{Q}}_{1,2}^T$, $\mathbf{R} = \tilde{\mathbf{Q}}_{2,1}^* \tilde{\mathbf{Q}}_{2,1}^T + \tilde{\mathbf{Q}}_{2,2}^* \tilde{\mathbf{Q}}_{2,2}^T$, and $\mathbf{S} = \tilde{\mathbf{Q}}_{1,1}^* \tilde{\mathbf{Q}}_{2,1}^T + \tilde{\mathbf{Q}}_{1,2}^* \tilde{\mathbf{Q}}_{2,2}^T$. Note that since $\mathbf{Q}^H = \mathbf{Q}$ and $\mathbf{R}^H = \mathbf{R}$, then \mathbf{Q} and \mathbf{R} are Hermitian matrices. Following the procedure in the SISO-OFDM system, we can obtain the optimum initial values by solving the following equations:

$$\mathbf{T}_i \mathbf{b}_i = \mathbf{c}_i, \quad i = 0, 1, \dots, 2N - 1, \quad (3.32)$$

where \mathbf{b}_i^T is composed of the non-zero elements in the i th non-zero row vector of \mathbf{W}_0 , as defined in (2.61). Note that we have two sets of circular band matrices to deal with; one is for $i = 0, 1, \dots, N - 1$ and the other is for $i = N, N + 1, \dots, 2N - 1$. Define $\tilde{a}_{i,j} = \tilde{\mathbf{Q}}_{1,1}(i,j)$, $\tilde{b}_{i,j} = \tilde{\mathbf{Q}}_{1,2}(i,j)$, $\tilde{c}_{i,j} = \tilde{\mathbf{Q}}_{2,1}(i,j)$, and $\tilde{d}_{i,j} = \tilde{\mathbf{Q}}_{2,2}(i,j)$. For the set of $i \in \{0, 1, \dots, N - 1\}$, we have \mathbf{T}_i , \mathbf{b}_i , and \mathbf{c}_i defined as:

$$\mathbf{T}_i = \begin{bmatrix} \mathbf{Q}_i & \mathbf{S}_i \\ \mathbf{S}_i^H & \mathbf{R}_i \end{bmatrix}, \quad (3.33)$$

$$\mathbf{Q}_i = \mathbf{Q}(i - D_2 : i + D_2, i - D_2 : i + D_2), \quad (3.34)$$

$$\mathbf{R}_i = \mathbf{R}(i - D_2 : i + D_2, i - D_2 : i + D_2), \quad (3.35)$$

$$\mathbf{S}_i = \mathbf{S}(i - D_2 : i + D_2, i - D_2 : i + D_2), \quad (3.36)$$

$$\mathbf{b}_i = [\mathbf{b}_{1,i}^T, \mathbf{b}_{2,i}^T]^T, \quad (3.37)$$

$$\mathbf{c}_i = [\mathbf{c}_{1,i}^T, \mathbf{c}_{2,i}^T]^T, \quad (3.38)$$

$$\mathbf{b}_{1,i} = [\alpha_{i,i-D_2}, \dots, \alpha_{i,i}, \dots, \alpha_{i,i+D_2}]^T, \quad (3.39)$$

$$\mathbf{b}_{2,i} = [\beta_{i,i-D_2}, \dots, \beta_{i,i}, \dots, \beta_{i,i+D_2}]^T, \quad (3.40)$$

$$\mathbf{c}_{1,i} = [\tilde{a}_{i-D_2,i}^*, \dots, \tilde{a}_{i,i}^*, \dots, \tilde{a}_{i+D_2,i}^*]^T, \quad (3.41)$$

$$\mathbf{c}_{2,i} = [\tilde{c}_{i-D_2,i}^*, \dots, \tilde{c}_{i,i}^*, \dots, \tilde{c}_{i+D_2,i}^*]^T. \quad (3.42)$$

For $i \in \{N, N + 1, \dots, 2N - 1\}$, Eqs. (3.33)–(3.42) are still valid but we need some modifications. We have to let $\mathbf{T}_i = \mathbf{T}_{i-N}$, and replace $\alpha_{i,j}$, $\beta_{i,j}$, $\tilde{a}_{i,j}^*$, and $\tilde{c}_{i,j}^*$ in $\mathbf{b}_{1,i}$, $\mathbf{b}_{2,i}$, $\mathbf{c}_{1,i}$, and $\mathbf{c}_{2,i}$ with $\gamma_{i,j}$, $\omega_{i,j}$, $\tilde{b}_{i,j}^*$, and $\tilde{d}_{i,j}^*$ respectively. Since the initial matrix is not a circular band matrix anymore, we are not able to obtain a recursive formula solving (3.32). In this case, Gaussian elimination can be chosen to solve this problem. As mentioned, $\mathbf{T}_i = \mathbf{T}_{i-N}$; we only need to construct \mathbf{T}_i and conduct Gaussian elimination for $i \in \{0, 1, \dots, N - 1\}$. Note that \mathbf{T}_i is a Hermitian matrix, making the complexity of Gaussian elimination lower.

Again, we can further reduce the computational complexity when evaluating \mathbf{Q} , \mathbf{R} , and \mathbf{S} . As mentioned, the elements close to the main diagonal of $\tilde{\mathbf{Q}}_{i,j}$ have dominative values. Thus, the terms far from the main diagonal can be ignored. Let $\hat{\mathbf{Q}}_{i,j} = \text{cirb}(\tilde{\mathbf{Q}}_{i,j}, S_2)$, $\hat{\mathbf{Q}} = \hat{\mathbf{Q}}_{1,1}^* \hat{\mathbf{Q}}_{1,1}^T + \hat{\mathbf{Q}}_{1,2}^* \hat{\mathbf{Q}}_{1,2}^T$, $\hat{\mathbf{R}} = \hat{\mathbf{Q}}_{2,1}^* \hat{\mathbf{Q}}_{2,1}^T + \hat{\mathbf{Q}}_{2,2}^* \hat{\mathbf{Q}}_{2,2}^T$, and $\hat{\mathbf{S}} = \hat{\mathbf{Q}}_{1,1}^* \hat{\mathbf{Q}}_{2,1}^T + \hat{\mathbf{Q}}_{1,2}^* \hat{\mathbf{Q}}_{2,2}^T$. Moreover, define $q_{i,j} = \mathbf{Q}(i, j)$, $r_{i,j} = \mathbf{R}(i, j)$, and $s_{i,j} = \mathbf{S}(i, j)$. Now, we have the approximation as follows: $q_{i,j} \approx \hat{\mathbf{Q}}(i, j) = \sum_{n \in \Gamma} (\tilde{a}_{i,n}^* \tilde{a}_{j,n} + \tilde{b}_{i,n}^* \tilde{b}_{j,n})$, $r_{i,j} \approx \hat{\mathbf{R}}(i, j) = \sum_{n \in \Gamma} (\tilde{c}_{i,n}^* \tilde{c}_{j,n} + \tilde{d}_{i,n}^* \tilde{d}_{j,n})$, and $s_{i,j} \approx \hat{\mathbf{S}}(i, j) = \sum_{n \in \Gamma} (\tilde{a}_{i,n}^* \tilde{c}_{j,n} + \tilde{b}_{i,n}^* \tilde{d}_{j,n})$, where $\Gamma = \langle i - S_2 : i + S_2, N \rangle \cap \langle j - S_2 : j + S_2, N \rangle$.

§ 3.3.3 Complexity Analysis

So far, we have derived a low-complexity algorithm, namely, N-MMSE that is suitable for a MIMO-OFDM system. In this subsection, we focus on analyzing the required computational complexity of the proposed N-MMSE method and then make a comparison between the N-MMSE and direct MMSE methods.

From Subsections 3.3.1, we know that the major computational load results from Eqs. (3.27) and (3.30) and it can be summarized as follows:

1. $\bar{\mathbf{v}}_m$ iteration, where $\bar{\mathbf{v}}_m = (\mathbf{W}_0 \tilde{\mathbf{Q}}) \bar{\mathbf{v}}_{m-1}$ and $\bar{\mathbf{v}}_0 = \mathbf{W}_0 \tilde{\mathbf{y}}$,
2. $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{Q}}$ construction, where $\tilde{\mathbf{Q}} = \left[\tilde{\mathbf{M}} + \alpha \sum_{j=0}^{2^p-1} \bar{d}_{p,j} \left(\mathbf{W}_0 \tilde{\mathbf{M}}^H \right)^j \mathbf{B}_0 \right]$,
3. \mathbf{B}_0 and \mathbf{W}_0 calculation.

The iterative step (3.30) involves $[2^{p+1}N \log_2 N + (2^{p+3}D_1 + 8D_2 + 7 \times 2^{p+1} + 3)N]$ CMs and $[2^{p+2}N \log_2 N + (2^{p+3}D_1 + 8D_2 + 9 \times 2^p + 3)N]$ CAs for $m \geq 1$. Furthermore, we need $(8D_2 + 4)N$ CMs and $(8D_2 + 2)N$ CAs for $\bar{\mathbf{v}}_0 = \mathbf{W}_0 \tilde{\mathbf{y}}$ and $4N$ RAs for each $\bar{c}_{k,m} \bar{\mathbf{v}}_m$. As to the construction of $\tilde{\mathbf{M}}_{j,i}^H$ for \mathbf{B}_0 , we require $(4S_1 + 2)N$ CMs and $(4S_1 + 2)N$ CAs. For calculating \mathbf{B}_0 , we need to construct matrices \mathbf{T}_i for $i = 0, 1, \dots, N - 1$ and they require $(8S_1 + 32S_1D_1 - 16D_1^2 + 8D_1 + 4)N$ CMs and $(8S_1 + 32S_1D_1 - 16D_1^2 - 8D_1)N$ CAs. To solve $\mathbf{T}_i \mathbf{b}_i = \mathbf{c}_i$ for \mathbf{B}_0 , it requires $[(32/3)D_1^3 + 44D_1^2 + (85/3)D_1 + 9/2]$ CMs, $[(32/3)D_1^3 + 44D_1^2 + (85/3)D_1 + 9/2]$ CAs, $(8D_1 + 4)$ CDs, and $(16D_1^2 + 12D_1 + 2)N$ RDs. To obtain $\tilde{\mathbf{Q}}$, we need $(4D_1 + 2)N$ CMs and $(4D_1 + 2)N$ CAs. Before calculating \mathbf{W}_0 , we require $(8S_2 + 32S_2D_2 - 16D_2^2 + 8D_2 + 4)N$ CMs and $(8S_2 + 32S_2D_2 - 16D_2^2 - 8D_2)N$ CAs to construct \mathbf{T}_i for $i = 0, 1, \dots, N - 1$. To obtain \mathbf{W}_0 , we have to solve $\mathbf{T}_i \mathbf{b}_i = \mathbf{c}_i$ which requires $[(32/3)D_2^3 + 44D_2^2 + (85/3)D_2 + 9/2]$ CMs, $[(32/3)D_2^3 + 44D_2^2 + (85/3)D_2 + 9/2]$ CAs, $(8D_2 + 4)$ CDs, and $(16D_2^2 + 12D_2 + 2)N$ RDs. As for the direct MMSE method, the matrix inversion can be implemented by Gaussian elimination [69]. Finally, we summarize the required complexity of the N-MMSE method and the direct MMSE method for a 2×2 MIMO-OFDM system in Table 3.4. We also summarize the complexity for calculating the initial matrices in the N-MMSE method in Table 3.5.

Table 3.4: Complexity comparison between the N-MMSE and direct MMSE methods in a 2×2 MIMO-OFDM system.

Methods	Real multiplications	Real divisions	Real additions
Direct MMSE	$\frac{64}{3}N^3 + 20N^2 + 16N \log_2 N + \frac{134}{3}N$	$4N^2 + 2N$	$\frac{64}{3}N^3 + 16N^2 + 24N \log_2 N + \frac{98}{3}N$
N-MMSE	$(2^k - 1)2^{p+3}N \log_2 N + \{48S_1 + 128S_1D_1 + 32S_2 + 128S_2D_2 + (128/3)D_1^3 + 112D_1^2 + [(2^k - 1)2^{p+5} + (592/3)]D_1 + (128/3)D_2^3 + 112D_2^2 + (2^{k+5} + 580/3)D_2 + 7 \times 2^{k+p+3} + 2^{k+4} - 7 \times 2^{p+3} + 126\}N + (2^p - 1)$	$(16D_1^2 + 28D_1 + 16D_2^2 + 28D_2 + 20)N$	$3(2^k - 1)2^{p+2}N \log_2 N + \{48S_1 + 128S_1D_1 + 32S_2 + 128S_2D_2 + (128/3)D_1^3 + 112D_1^2 + [(2^k - 1)2^{p+5} + (424/3)]D_1 + (128/3)D_2^3 + 112D_2^2 + (2^{k+5} + 412/3)D_2 + 23 \times 2^{k+p+1} + 2^{k+4} - 23 \times 2^{p+1} + 82\}N$

Table 3.5: Complexity of the initial matrices calculation for the N-MMSE method in a 2×2 MIMO-OFDM system.

Methods	Real multiplications	Real divisions	Real additions
N-MMSE ($D_1 = D_2 = 1$)	$(176 * S_1 + 160 * S_2 + 826)N$	$(176 * S_1 + 160 * S_2 + 674)N$	$108N$

§ 3.3.4 Simulations

We report the simulation results in a MIMO-OFDM system whose settings are the same as those in a SISO-OFDM system. Specifically, we consider a 2×2 MIMO-OFDM system. We

also consider two cases, $f_d = 0.05$ and 0.1 . Figure 3.6 shows the corresponding BER result for $f_d = 0.05$. From this figure, we observe the similar result to that in a SISO-OFDM system. The N-MMSE method can still approach to the direct MMSE method within one or two iterations. Figure 3.7 shows the result for $f_d = 0.1$. It is apparent that the performance of the two-tap FEQ method becomes even poorer than that in a SISO-OFDM system. This is because the inter-antenna interference is introduced in addition to ICI. By contrast, the performance of the N-MMSE method remains similar to that in a SISO-OFDM system. Both ICI and inter-antenna interference are mitigated. To understand the impact of f_d , we show the BER performance for various f_d . Figure 3.8 shows the results when f_d varies from 0 to 0.2 and the SNR is set to 30 dB. From this figure, we see that the N-MMSE method ($k = 3$) has larger performance loss when f_d becomes larger.

Table 3.6 shows the complexity comparison for the N-MMSE and direct MMSE methods. For the case of $f_d = 0.05$, the required number of iterations is one or two. When one (two) iteration is used, the complexity ratio for multiplication, division, and addition is 0.005 (0.007), 0.210 (0.210), and 0.005 (0.007), respectively. For $f_d = 0.1$, the required iteration number is two or three. When two (three) iterations are used, the complexity ratio for multiplication, division, and addition turns out to be 0.015 (0.019), 0.210 (0.210), and 0.015 (0.019), respectively. Although the ratio of division is increased in MIMO-OFDM systems, the number of divisions is minor compared to that of multiplications. From Tables 2.6 and 3.6, we can see that the major computational load is the multiplication. Thus, in MIMO-OFDM systems, the N-MMSE method can achieve more complexity reduction. For example, in the SISO-OFDM system, the complexity ratio for multiplication is 0.014, while that in the MIMO-OFDM system is 0.007 ($f_d = 0.05$ and $k = 2$). This is because the computational complexity of the direct MMSE method is $\mathcal{O}(M^3 N^3)$, whereas that of the N-MMSE method is $\mathcal{O}(2MN \log_2 N)$. Consequently, the complexity of the direct MMSE method grows faster as M increases. From the complexity analysis, we can infer that when N becomes larger, the computational complexity reduction achieved with the N-MMSE method will be more apparent. For various N , we show

the required computational complexity for the direct MMSE method and the N-MMSE method ($k = 2, 3$) in Figs. 3.9 and 3.10, respectively.

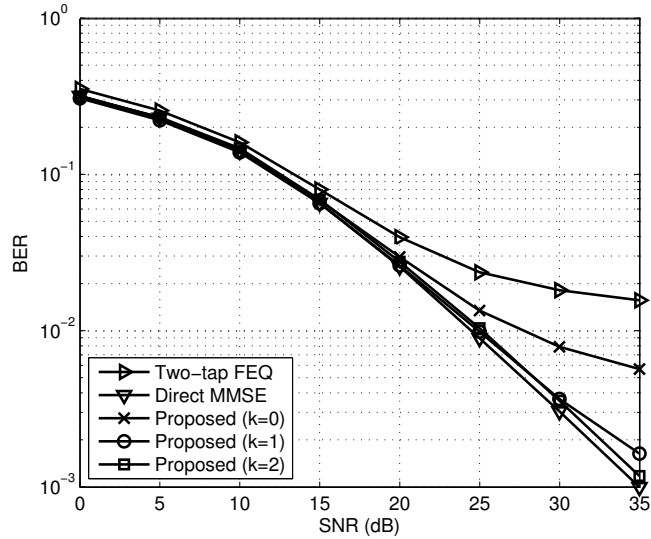


Figure 3.6: BER comparison among the two-tap FEQ, N-MMSE ($\{D_1, D_2\} = \{1, 1\}$ and $\{S_1, S_2\} = \{2, 2\}$), and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0.05$ and 16-QAM modulation.

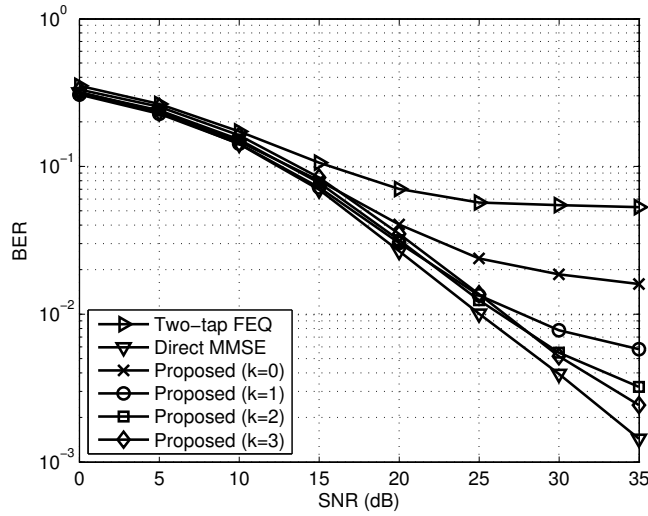


Figure 3.7: BER comparison among the two-tap FEQ, N-MMSE ($\{D_1, D_2\} = \{1, 1\}$ and $\{S_1, S_2\} = \{10, 10\}$), and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0.1$ and 16-QAM modulation.

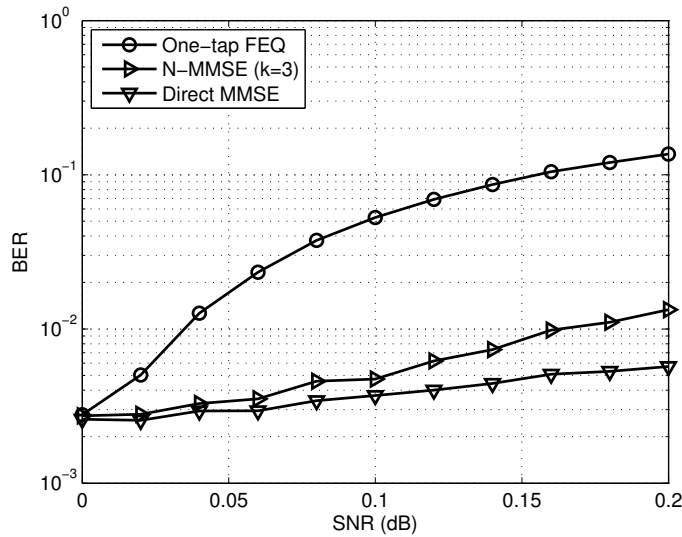


Figure 3.8: BER comparison among one-tap FEQ, N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 10$, and $k = 3$), and direct MMSE methods in a 2×2 MIMO-OFDM system; $f_d = 0 \sim 0.2$, 16 QAM modulation, and SNR = 30 dB.

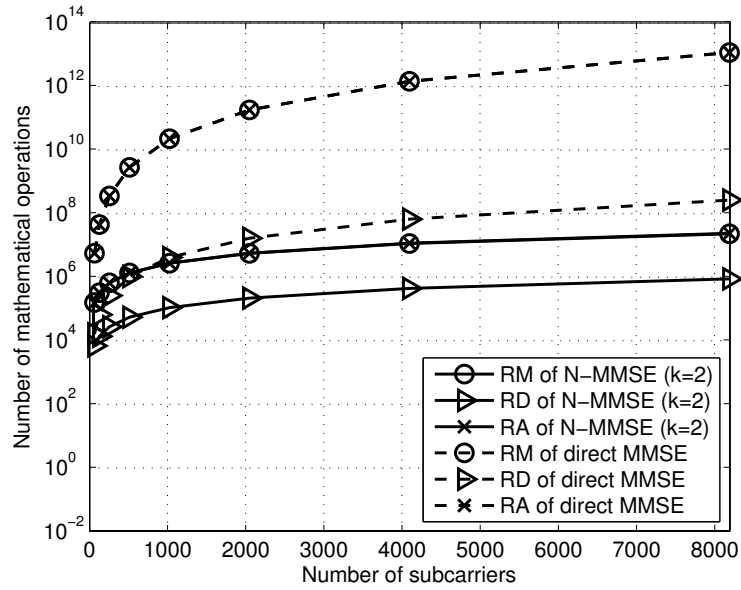


Figure 3.9: Complexity comparison between N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 5$, and $k = 2$) and direct MMSE methods in a 2×2 MIMO-OFDM system for various N .

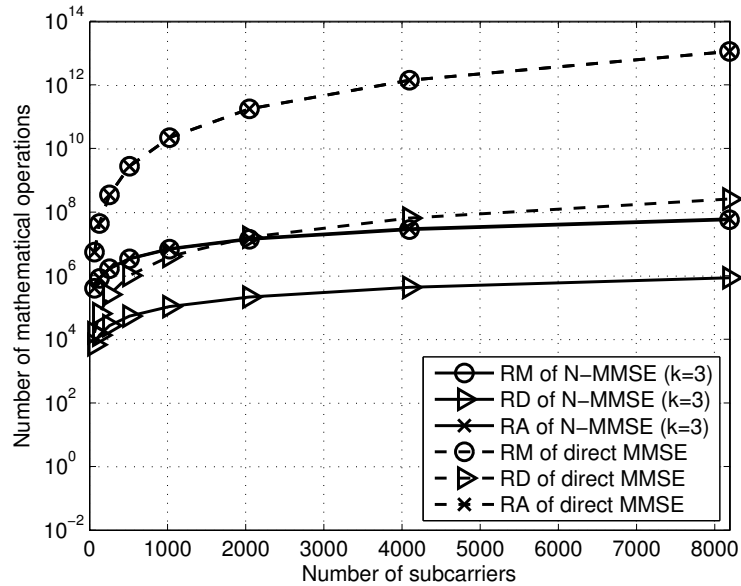


Figure 3.10: Complexity comparison between N-MMSE ($D_1 = D_2 = 1$, $S_1 = S_2 = 10$, and $k = 3$) and direct MMSE methods in a 2×2 MIMO-OFDM system for various N .

Table 3.6: Complexity comparison between the N-MMSE and direct MMSE methods in a 2×2 MIMO-OFDM system ($N = 128$, and $\{D_1, D_2\} = \{1, 1\}$).

Methods	Real multiplications (ratio)	Real divisions (ratio)	Real additions (ratio)
Direct MMSE	45086976	65792	45027072
N-MMSE ($k = 1$, $\{S_1, S_2\} = \{2, 2\}$)	240897 (0.005)	13824 (0.210)	225536 (0.005)
N-MMSE ($k = 2$, $\{S_1, S_2\} = \{2, 2\}$)	326913 (0.007)	13824 (0.210)	320768 (0.007)
N-MMSE ($k = 2$, $\{S_1, S_2\} = \{10, 10\}$)	670977 (0.015)	13824 (0.210)	664832 (0.015)
N-MMSE ($k = 3$, $\{S_1, S_2\} = \{10, 10\}$)	843009 (0.019)	13824 (0.210)	855296 (0.019)

Chapter 4

Mobility-induced ICI Mitigation for SISO/MIMO-OFDMA Systems

§ 4.1 SISO-OFDMA Signal Model

In Chapters 2 and 3, we have discussed ICI mitigation in SISO/MIMO-OFDM systems. In recent years, OFDMA have been a promising multiple access scheme and has been extensively investigated. For now on, we will focus on ICI mitigation in a SISO/MIMO-OFDMA system. Based on the previous SISO/MIMO-OFDM signal model, we will derive the SISO/MIMO-OFDMA signal model for high-mobility environments. Using the model, we can then apply the low-complexity ICI mitigation methods developed in Chapter 2.

A SISO-OFDMA system is a multicarrier system and the available bandwidth is divided into N equally spaced subbands. Each subcarrier uses a subband with a bandwidth of $1/(NT_s)$. Furthermore, a SISO-OFDMA system is also a multiuser system and all active users share the N subcarriers. Without loss of generality, we assume that each user is assigned the same number of subcarriers, N_s , and $N_s = N/Q$, where Q is the number of active users. Let Υ_q be the set of indices indicating subcarriers assigned to user q and \tilde{x}_k^q be the user's transmit signal at the k th subcarrier, where $k \in \Upsilon_q$. To avoid multiple-access interference, it is usually assumed that

$\Upsilon_q \cap \Upsilon_{q'} = \emptyset$ if $q \neq q'$ and $\bigcup_{q=1}^Q \Upsilon_q = \{0, 1, \dots, N-1\}$. An OFDMA system usually uses an interleaved structure, i.e., $\Upsilon_q = \{q-1, q-1+Q, \dots, q-1+(N/Q-1) \times Q\}$. Since the subcarriers assigned to a user are evenly distributed in the transmission band, this scheme can achieve the maximum frequency diversity.

For all users, we assume that the CP length N_g is long enough to prevent the ISI effect. Consider a specific OFDMA symbol of user q . After CP removal, the receive time-domain signal can be expressed as

$$\mathbf{y}^q = \mathbf{H}^q \mathbf{x}^q, \quad (4.1)$$

where \mathbf{y}^q is the q th user's receive time-domain $N \times 1$ signal vector and \mathbf{x}^q is the q th user's transmit time-domain $N \times 1$ signal vector, i.e., $\mathbf{x}^q = (1/\sqrt{N})\mathbf{G}^H \tilde{\mathbf{x}}^q$, where $\tilde{\mathbf{x}}^q$ is the q th user's frequency-domain signal vector. The matrix, \mathbf{H}^q , is the corresponding time-domain time-variant $N \times N$ channel matrix for user q . Note that since this is a multiuser system, elements of $\tilde{\mathbf{x}}^q$ are nonzero only in the designated subcarrier positions. Then the overall receive time-domain signal from Q users can be expressed as

$$\mathbf{y} = \sum_{q=1}^Q \mathbf{H}^q \mathbf{x}^q + \mathbf{z}, \quad (4.2)$$

where \mathbf{y} is the overall receive time-domain signal vector and \mathbf{z} denotes the noise vector. After the FFT operation, we obtain the corresponding frequency-domain signal as

$$\begin{aligned} \tilde{\mathbf{y}} &= \sqrt{N}\mathbf{G}\mathbf{y} \\ &= \sum_{q=1}^Q \tilde{\mathbf{H}}^q \tilde{\mathbf{x}}^q + \tilde{\mathbf{z}}, \end{aligned} \quad (4.3)$$

where $\tilde{\mathbf{H}}^q = \mathbf{G}\mathbf{H}^q\mathbf{G}^H$ is the frequency-domain ICI channel matrix of user q and $\tilde{\mathbf{z}} = \sqrt{N}\mathbf{G}\mathbf{z}$ is the frequency-domain noise vector. Define a diagonal selection matrix \mathbf{S}^q such that $\mathbf{S}^q(i, i) = 1, i \in \Upsilon_q$ and $\mathbf{S}^q(i, i) = 0$, otherwise and a composite transmit frequency-domain signal vector as $\tilde{\mathbf{x}} = [\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{N-1}]^T$, where $\tilde{x}_k = \tilde{x}_k^q$ if $k \in \Upsilon_q$. Based on these definitions, we then have

$\tilde{\mathbf{x}}^q = \mathbf{S}^q \tilde{\mathbf{x}}$. Also, we can further express the receive signal in (4.3) as

$$\begin{aligned}\tilde{\mathbf{y}} &= \left(\sum_{q=1}^Q \tilde{\mathbf{H}}^q \mathbf{S}^q \right) \tilde{\mathbf{x}} + \tilde{\mathbf{z}} \\ &= \tilde{\mathbf{M}} \tilde{\mathbf{x}} + \tilde{\mathbf{z}},\end{aligned}\quad (4.4)$$

where $\tilde{\mathbf{M}} = \sum_{q=1}^Q \tilde{\mathbf{H}}^q \mathbf{S}^q$ is the mobility-induced multiuser ICI channel matrix. If all users are static, $\tilde{\mathbf{M}}$ will become a diagonal matrix. For each user, the channel matrix \mathbf{H}^q is time-variant. As we did in a SISO/MIMO-OFDM system, we also use the LTV channel model to approximate a time-variant channel [59]. According to the LTV channel model, the time-domain time-variant channel for user q in a specific OFDMA symbol period is then approximated as

$$h_l^q(n) = h_{0,l}^q + n \times h_{1,l}^q, \quad (4.5)$$

where n is the time index, $h_l^q(n)$ is the l th-tap channel response at time instant n for user q , $h_{0,l}^q$ is its constant term, and $h_{1,l}^q$ is its variation rate. We assume that n is 0 for the midpoint of an OFDMA symbol. Let $\mathbf{h}_0^q = [h_{0,0}^q, h_{0,1}^q, \dots, h_{0,N-1}^q]^T$, $\mathbf{h}_1^q = [h_{1,0}^q, h_{1,1}^q, \dots, h_{1,N-1}^q]^T$, $\mathbf{H}_0^q = \text{cir}(\mathbf{h}_0^q)$, and $\mathbf{H}_1^q = \text{cir}(\mathbf{h}_1^q)$. Using the above definitions and (4.5), we can express the time-domain channel matrix of the q th user as

$$\mathbf{H}^q = \mathbf{H}_0^q + \mathbf{V}_1 \mathbf{H}_1^q. \quad (4.6)$$

From (4.6), we see that \mathbf{H}^q consists of two circulant matrices and a diagonal matrix. The corresponding frequency-domain ICI channel matrix of the q th user is then expressed as

$$\begin{aligned}\tilde{\mathbf{H}}^q &= \mathbf{G} \mathbf{H}^q \mathbf{G}^H \\ &= \tilde{\mathbf{H}}_0^q + \mathbf{G} \mathbf{V}_1 \mathbf{G}^H \tilde{\mathbf{H}}_1^q,\end{aligned}\quad (4.7)$$

where $\tilde{\mathbf{H}}_0^q = \mathbf{G} \mathbf{H}_0^q \mathbf{G}^H$ and $\tilde{\mathbf{H}}_1^q = \mathbf{G} \mathbf{H}_1^q \mathbf{G}^H$. Since \mathbf{H}_i^q is a circulant matrix, $\tilde{\mathbf{H}}_i^q$ can be further expressed as $\tilde{\mathbf{H}}_i^q = \text{diag}(\tilde{\mathbf{h}}_i^q)$ for $i = 0, 1$. Thus, the ICI matrix can be further rewritten as

$$\tilde{\mathbf{M}} = \sum_{q=1}^Q \tilde{\mathbf{H}}_0^q \mathbf{S}^q + \mathbf{G} \mathbf{V}_1 \mathbf{G}^H \sum_{q=1}^Q \tilde{\mathbf{H}}_1^q \mathbf{S}^q. \quad (4.8)$$

Let $\tilde{\mathbf{h}}_0^q = \sqrt{N}\mathbf{G}\mathbf{h}_0^q$ and $\tilde{\mathbf{h}}_1^q = \sqrt{N}\mathbf{G}\mathbf{h}_1^q$. Also, let the k th component of $\tilde{\mathbf{h}}_0^q$ and that of $\tilde{\mathbf{h}}_1^q$ be denoted as $\tilde{h}_{0,k}^q$ and $\tilde{h}_{1,k}^q$, respectively. Now, we can define two composite channel vectors $\tilde{\mathbf{h}}_0$ and $\tilde{\mathbf{h}}_1$ such that their k th components, denoted as $\tilde{h}_{0,k}$ and $\tilde{h}_{1,k}$, have the following property: $\tilde{h}_{0,k} = \tilde{h}_{0,k}^q$ and $\tilde{h}_{1,k} = \tilde{h}_{1,k}^q$ if $k \in \Upsilon_q$. With these definitions, we then have $\tilde{\mathbf{H}}_0 = \text{diag}(\tilde{\mathbf{h}}_0) = \sum_{q=1}^Q \tilde{\mathbf{H}}_0^q \mathbf{S}^q$ and $\tilde{\mathbf{H}}_1 = \text{diag}(\tilde{\mathbf{h}}_1) = \sum_{q=1}^Q \tilde{\mathbf{H}}_1^q \mathbf{S}^q$. Finally, we can rewrite (4.8) as

$$\tilde{\mathbf{M}} = \tilde{\mathbf{H}}_0 + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{H}}_1. \quad (4.9)$$

From (4.9), we see that the ICI matrix is decomposed into a combination of diagonal and DFT/IDFT matrices. Note that the ICI matrix is a composite matrix; it describes interference generated from all active users. If $Q = 1$, the signal model in (4.9) will be degenerated into the SISO-OFDM signal model. Moreover, the ICI matrix can also be expressed as $\tilde{\mathbf{M}} = \tilde{\mathbf{H}}_0 + \tilde{\mathbf{V}}_1\tilde{\mathbf{H}}_1$, where $\tilde{\mathbf{V}}_1 = \mathbf{G}\mathbf{V}_1\mathbf{G}^H = [\text{cir}(\tilde{\mathbf{v}}_1)]^T$ and $\tilde{\mathbf{v}}_1 = (1/\sqrt{N})\mathbf{G}^H\mathbf{v}_1$.

We find that the derived signal model for a SISO-OFDMA system in (4.9) has the same structure as that in a SISO-OFDM system. That is to say, they both consist of diagonal and DFT/IDFT matrices. As a result, the proposed N-ZF and N-MMSE methods in Chapter 2 can be applied to ICI mitigation in a SISO-OFDMA system. Thus, the required computational complexity is the same as that in SISO-OFDM systems.

§ 4.1.1 Simulations

In Section 4.1, we have derived a SISO-OFDMA signal model to facilitate ICI mitigation. In this subsection, we report simulation results to demonstrate the effectiveness of the proposed low-complexity ZF and MMSE methods. We consider a SISO-OFDMA system with $N = 128$, $N_g = 32$, and $Q = 4$. A fast-fading channel is used in simulations and the power delay profile for the q th user is characterized by an exponential function, i.e., $\sigma_{l,q}^2 = e^{-l \times \alpha_q/L} / \sum_{i=0}^{L-1} e^{-i \times \alpha_q/L}$, where l is the tap index, q is the user index, and L is the number of channel taps. For the following simulations, we let $\alpha_1 = 1$, $\alpha_2 = 3$, $\alpha_3 = 2$, $\alpha_4 = 1.5$, and $L = 15$. Each channel tap is generated by Jakes' model [70]. For the direct ZF and MMSE

methods, we assume that the channel response is exactly known. For the N-ZF and N-MMSE methods, the parameters of the LTV channel model are obtained by LS fittings. The modulation scheme is chosen as 16-QAM.

First, we will present the simulation results of the ZF method and then we report the results for the MMSE method. We assume that f_d of each user is different. For a SISO-OFDMA system, f_d 's are set to $\{0.05, 0.02, 0.04, 0.03\}$. Figure 4.1 shows the BER comparison among the one-tap FEQ method, the direct ZF method, and the N-ZF method. Again, we see that the one-tap FEQ method has an error floor phenomenon which is due to the ignored ICI effect. However, the N-ZF method effectively mitigates this phenomenon and outperforms the one-tap FEQ method. Moreover, with two iterations, the N-ZF method can have a similar performance to that of the direct ZF method. Figure 4.2 shows the BER performance comparison for another case, where f_d 's are set to $\{0.05, 0.1, 0.04, 0.03\}$. In this case, f_d of user 2 is increased up to 0.1, implying more ICI generated from user 2. From this figure, it is obvious that the one-tap FEQ method has an even severer error floor phenomenon. Again, the N-ZF method with two iterations has a similar performance to that of the direct ZF method.

Table 4.1 summarizes the computational complexity comparison for the direct ZF and N-ZF methods under the same settings shown above. In this table, the ratio in a parenthesis indicates the number of additions/multiplications/divisions required for the N-ZF method divided by that for the direct ZF method. From the above simulation results, we can say that the required iteration number for the N-ZF method is two. From Table 4.1, we see that the multiplication/division/addition complexity of the N-ZF method is 0.007/0.016/0.008 times that of the direct ZF method. From Figs. 4.1 and 4.2, we can then conclude that while the performance of the N-ZF method is comparable to that of the direct ZF method, the computational complexity is much lower.

Next we report the BER performance of the MMSE method. Figure 4.3 shows the result when f_d 's are set to $\{0.05, 0.02, 0.04, 0.03\}$. In this case, we clearly see that the N-MMSE method with two iterations has the same performance with the direct MMSE method. Increasing

the f_d of user 2 up to 0.1, we have the result shown in Fig. 4.4. Owing to the increasing amount of ICI from user 2, the N-MMSE method need three iterations to approach the performance of the direct MMSE method.

Table 4.2 compares the computational complexity of the direct MMSE and N-MMSE methods under the same settings shown above. In this table, the ratio in a parenthesis indicates the number of additions/multiplications/divisions required for the N-MMSE method divided by that for the direct MMSE method. The simulation results imply that the required iteration number for the N-MMSE method is two. From Table 4.2, we see that the multiplication/division/addition complexity of the N-MMSE method ($k = 2$) is 0.0135/0.0309/0.0129 times that of the direct MMSE method. Consequently, we can conclude that the N-MMSE method can approach the direct MMSE method while at the same time the N-MMSE method saves a lot of computations.



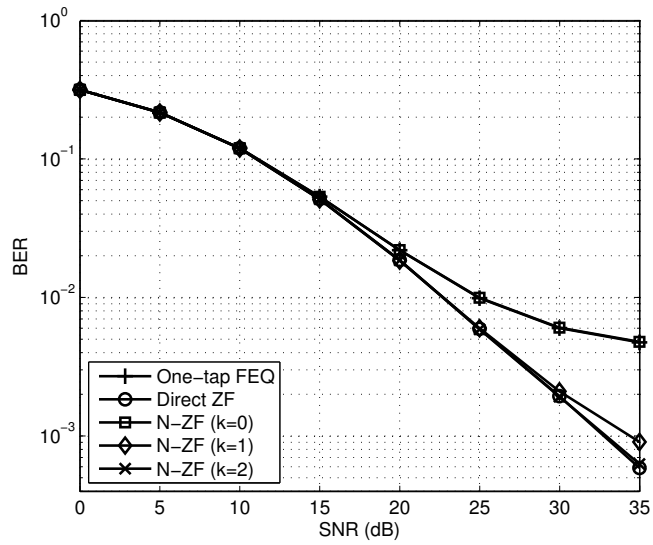


Figure 4.1: BER comparison among one-tap FEQ, direct ZF, and N-ZF ($D = 0$, $S = 2$) methods in a SISO-OFDMA system; $f_d = \{0.05, 0.02, 0.04, 0.03\}$ and 16-QAM modulation.

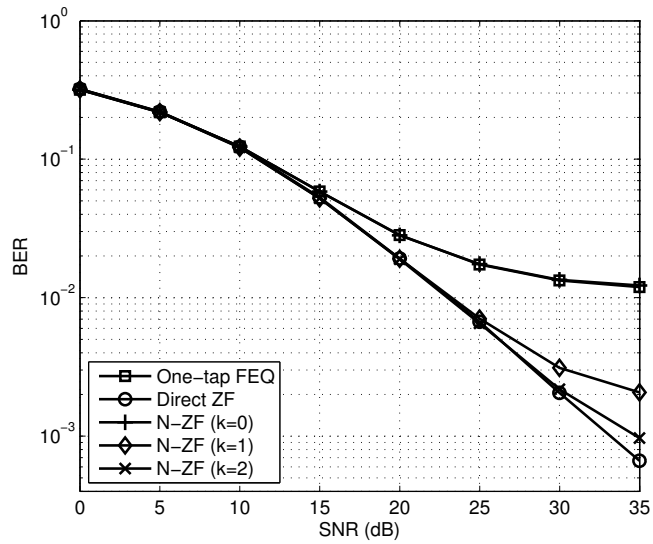


Figure 4.2: BER comparison among one-tap FEQ, direct ZF, and N-ZF ($D = 0$, $S = 2$) methods in a SISO-OFDMA system; $f_d = \{0.05, 0.1, 0.04, 0.03\}$ and 16-QAM modulation.

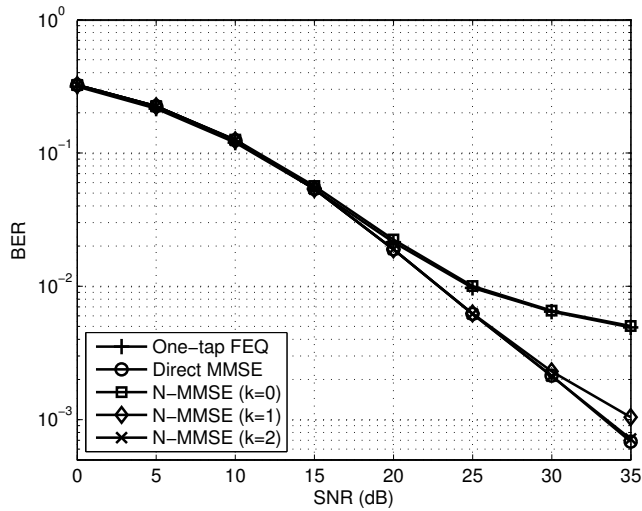


Figure 4.3: BER comparison among one-tap FEQ, direct MMSE, and N-MMSE ($D_1 = D_2 = 0$, $S_1 = S_2 = 2$) methods in a SISO-OFDMA system; $f_d = \{0.05, 0.02, 0.04, 0.03\}$ and 16-QAM modulation.

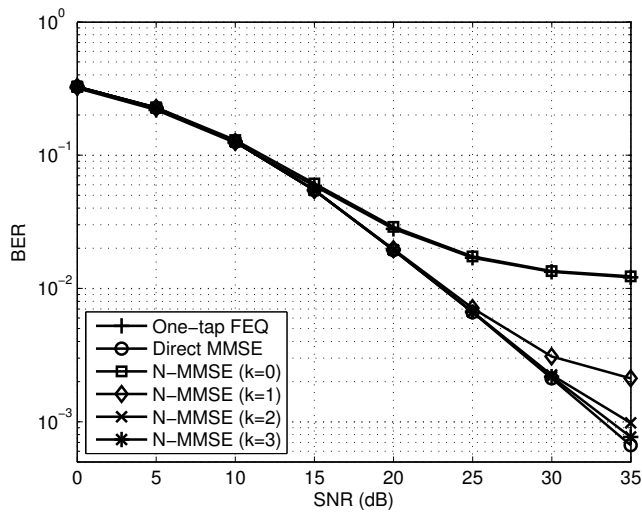


Figure 4.4: BER comparison among one-tap FEQ, direct MMSE, and N-MMSE ($D_1 = D_2 = 0$, $S_1 = S_2 = 2$) methods in a SISO-OFDMA system; $f_d = \{0.05, 0.1, 0.04, 0.03\}$ and 16-QAM modulation.

Table 4.1: Complexity comparison between direct ZF and N-ZF methods in a SISO-OFDMA system ($N = 128$, $D = 0$, and $S = 2$).

Methods	Real multiplications (ratio)	Real divisions (ratio)	Real additions (ratio)
Direct ZF	2910848	16512	2885824
N-ZF ($k = 1$)	8704 (0.003)	256 (0.016)	9344 (0.003)
N-ZF ($k = 2$)	19968 (0.007)	256 (0.016)	22656 (0.008)

Table 4.2: Complexity comparison between direct MMSE and N-MMSE methods in a SISO-OFDMA system ($N = 128$, $D_1 = D_2 = 0$, and $S_1 = S_2 = 2$).

Methods	Real multiplications (ratio)	Real divisions (ratio)	Real additions (ratio)
Direct MMSE	5679488	16512	5664128
N-MMSE ($k = 1$)	47109 (0.0083)	510 (0.0309)	39688 (0.0070)
N-MMSE ($k = 2$)	76805 (0.0135)	510 (0.0309)	72968 (0.0129)
N-MMSE ($k = 3$)	136197 (0.0240)	510 (0.0309)	139528 (0.0246)

§ 4.2 MIMO-OFDMA Signal Model

In the previous section, we have derived the SISO-OFDMA signal model and applied the low-complexity algorithms developed in Chapter 2 for ICI mitigation. Next we will further tackle the ICI problem in MIMO-OFDMA systems. In this section, we will develop a MIMO-OFDMA signal model facilitating the application of the proposed low-complexity algorithms. First, we will generalize the signal model derived in Section 4.1 to the scenario of MIMO-OFDMA systems. Let \mathbf{x}_j^q be the transmit symbol vector of user q from its j th antenna and $\mathbf{y}_{i,j}^q$ be the cor-

responding receive OFDMA symbol vector in the i th receive antenna (without CP and noise). Based on these definitions, we have the signal model as follows:

$$\mathbf{y}_{i,j}^q = \mathbf{H}_{i,j}^q \mathbf{x}_j^q. \quad (4.10)$$

Note that both $\mathbf{y}_{i,j}^q$ and \mathbf{x}_j^q are time-domain signals of dimension $N \times 1$. Moreover, $\mathbf{x}_j^q = (1/\sqrt{N})\mathbf{G}^H \tilde{\mathbf{x}}_j^q$, where $\tilde{\mathbf{x}}_j^q$ is the corresponding frequency-domain symbol vector of \mathbf{x}_j^q . The matrix, $\mathbf{H}_{i,j}^q$, is the time-domain channel matrix for user q consisting of the time-variant channel response between the j th transmit antenna and the i th receive antenna. Note that only in the designated subcarrier positions are the elements of $\tilde{\mathbf{x}}_j^q$ non-zeros. We then have the receive time-domain OFDMA signal, for the channel between the j th transmit antenna and the i th receive antenna, as

$$\mathbf{y}_{i,j} = \sum_{q=1}^Q \mathbf{H}_{i,j}^q \mathbf{x}_j^q, \quad (4.11)$$

where $\mathbf{y}_{i,j}$ is the receive time-domain signal vector in the i th receive antenna contributed from the j th transmit antenna of all users. Transforming (4.11) with DFT, we have the corresponding frequency-domain signal as

$$\tilde{\mathbf{y}}_{i,j} = \sum_{q=1}^Q \tilde{\mathbf{H}}_{i,j}^q \tilde{\mathbf{x}}_j^q, \quad (4.12)$$

where $\tilde{\mathbf{y}}_{i,j}$ is the corresponding frequency-domain signal vector of $\mathbf{y}_{i,j}$ and $\tilde{\mathbf{H}}_{i,j}^q = \mathbf{G}\mathbf{H}_{i,j}^q\mathbf{G}^H$.

For user q , we define $\tilde{x}_{j,k}^q$ as the transmit signal at the k th subcarrier from its j th transmit antenna. Then we let $\tilde{\mathbf{x}}_j = [\tilde{x}_{j,0}, \tilde{x}_{j,1}, \dots, \tilde{x}_{j,N-1}]^T$, where $\tilde{x}_{j,k} = \tilde{x}_{j,k}^q$ if $k \in \Upsilon_q$. Thus, we have $\tilde{\mathbf{x}}_j^q = \mathbf{S}^q \tilde{\mathbf{x}}_j$. Now we can express the receive signal in (4.12) as

$$\begin{aligned} \tilde{\mathbf{y}}_{i,j} &= \left(\sum_{q=1}^Q \tilde{\mathbf{H}}_{i,j}^q \mathbf{S}^q \right) \tilde{\mathbf{x}}_j \\ &= \tilde{\mathbf{M}}_{i,j} \tilde{\mathbf{x}}_j, \end{aligned} \quad (4.13)$$

where $\tilde{\mathbf{M}}_{i,j} = \sum_{q=1}^Q \tilde{\mathbf{H}}_{i,j}^q \mathbf{S}^q$. The matrix, $\tilde{\mathbf{M}}_{i,j}$, represents the mobility-induced composite ICI channel matrix between the i th receive antenna and the j th transmit antenna (for all users). With

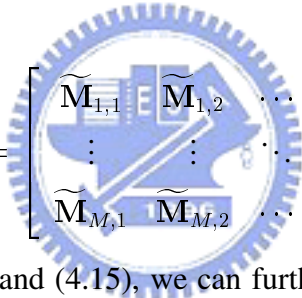
(4.13), we can further express the receive frequency-domain signal in the i th receive antenna (from all transmit antennas of all users) as

$$\tilde{\mathbf{y}}_i = \sum_{j=1}^M \tilde{\mathbf{M}}_{i,j} \tilde{\mathbf{x}}_j + \tilde{\mathbf{z}}_i, \quad (4.14)$$

where $\tilde{\mathbf{z}}_i$ is the receive frequency-domain noise vector in the i th receive antenna. By stacking all the receive signal vectors into a column vector, we finally obtain the following signal model

$$\tilde{\mathbf{y}} = \tilde{\mathbf{M}} \tilde{\mathbf{x}} + \tilde{\mathbf{z}}, \quad (4.15)$$

where $\tilde{\mathbf{y}} = [\tilde{\mathbf{y}}_1^T, \tilde{\mathbf{y}}_2^T, \dots, \tilde{\mathbf{y}}_M^T]^T$ is the overall receive frequency-domain signal vector, $\tilde{\mathbf{x}} = [\tilde{\mathbf{x}}_1^T, \tilde{\mathbf{x}}_2^T, \dots, \tilde{\mathbf{x}}_M^T]^T$ is the transmit frequency-domain signal vector, $\tilde{\mathbf{z}} = [\tilde{\mathbf{z}}_1^T, \tilde{\mathbf{z}}_2^T, \dots, \tilde{\mathbf{z}}_M^T]^T$ is the frequency-domain noise vector, and $\tilde{\mathbf{M}}$ is the frequency-domain ICI channel matrix expressed as



$$\tilde{\mathbf{M}} = \begin{bmatrix} \tilde{\mathbf{M}}_{1,1} & \tilde{\mathbf{M}}_{1,2} & \cdots & \tilde{\mathbf{M}}_{1,M} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{M}}_{M,1} & \tilde{\mathbf{M}}_{M,2} & \cdots & \tilde{\mathbf{M}}_{M,M} \end{bmatrix}. \quad (4.16)$$

Using the LTV channel model and (4.15), we can further derive a model facilitating ICI mitigation in MIMO-OFDMA systems. With the LTV model, we again express the channel in a specific OFDMA symbol as

$$h_{l,i,j}^q(n) = h_{0,l,i,j}^q + n \times h_{1,l,i,j}^q, \quad (4.17)$$

where $h_{l,i,j}^q(n)$ is the l th-tap channel response at time instant n between the i th receive antenna and the j th transmit antenna for user q , $h_{0,l,i,j}^q$ is its constant term, and $h_{1,l,i,j}^q$ is its variation rate. Let $\mathbf{h}_{0,i,j}^q = [h_{0,0,i,j}^q, h_{0,1,i,j}^q, \dots, h_{0,N-1,i,j}^q]^T$, $\mathbf{h}_{1,i,j}^q = [h_{1,0,i,j}^q, h_{1,1,i,j}^q, \dots, h_{1,N-1,i,j}^q]^T$, $\mathbf{H}_{0,i,j}^q = \text{cir}(\mathbf{h}_{0,i,j}^q)$, and $\mathbf{H}_{1,i,j}^q = \text{cir}(\mathbf{h}_{1,i,j}^q)$. Using these definitions and (4.17), we can express the q th user's time-domain channel matrix representing the channel between the i th receive antenna and the j th transmit antenna as

$$\mathbf{H}_{i,j}^q = \mathbf{H}_{0,i,j}^q + \mathbf{V}_1 \mathbf{H}_{1,i,j}^q. \quad (4.18)$$

Transforming $\mathbf{H}_{i,j}^q$ from the time domain to the frequency domain and denoting the result as $\tilde{\mathbf{H}}_{i,j}^q$, we have

$$\tilde{\mathbf{H}}_{i,j}^q = \tilde{\mathbf{H}}_{0,i,j}^q + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{H}}_{1,i,j}^q, \quad (4.19)$$

where $\tilde{\mathbf{H}}_{0,i,j}^q = \mathbf{G}\mathbf{H}_{0,i,j}^q\mathbf{G}^H$ and $\tilde{\mathbf{H}}_{1,i,j}^q = \mathbf{G}\mathbf{H}_{1,i,j}^q\mathbf{G}^H$. As a result, we then come to the following composite channel matrix

$$\tilde{\mathbf{M}}_{i,j} = \sum_{q=1}^Q \tilde{\mathbf{H}}_{0,i,j}^q \mathbf{S}^q + \mathbf{G}\mathbf{V}_1\mathbf{G}^H \sum_{q=1}^Q \tilde{\mathbf{H}}_{1,i,j}^q \mathbf{S}^q. \quad (4.20)$$

Let $\tilde{\mathbf{h}}_{0,i,j}^q = \sqrt{N}\mathbf{G}\mathbf{h}_{0,i,j}^q$ and $\tilde{\mathbf{h}}_{1,i,j}^q = \sqrt{N}\mathbf{G}\mathbf{h}_{1,i,j}^q$. Also denote the k th component of $\tilde{\mathbf{h}}_{0,i,j}^q$ as $\tilde{h}_{0,k,i,j}^q$ and that of $\tilde{\mathbf{h}}_{1,i,j}^q$ as $\tilde{h}_{1,k,i,j}^q$. As a result, we then define two composite vectors $\tilde{\mathbf{h}}_{0,i,j}$ and $\tilde{\mathbf{h}}_{1,i,j}$ such that their k th components, denoted as $\tilde{h}_{0,k,i,j}$ and $\tilde{h}_{1,k,i,j}$, have the following property: $\tilde{h}_{0,k,i,j} = \tilde{h}_{0,k,i,j}^q$ and $\tilde{h}_{1,k,i,j} = \tilde{h}_{1,k,i,j}^q$ if $k \in \Upsilon_q$. With the above definitions, we then have $\tilde{\mathbf{H}}_{0,i,j} = \text{diag}(\tilde{\mathbf{h}}_{0,i,j})$ and $\tilde{\mathbf{H}}_{1,i,j} = \text{diag}(\tilde{\mathbf{h}}_{1,i,j})$. Finally, we can rewrite (4.20) as

$$\tilde{\mathbf{M}}_{i,j} = \tilde{\mathbf{H}}_{0,i,j} + \mathbf{G}\mathbf{V}_1\mathbf{G}^H\tilde{\mathbf{H}}_{1,i,j}, \quad (4.21)$$

where $\tilde{\mathbf{H}}_{s,i,j} = \sum_{q=1}^Q \tilde{\mathbf{H}}_{s,i,j}^q \mathbf{S}^q$ for $s = 0, 1$. Note that $\tilde{\mathbf{H}}_{s,i,j}^q$ can be further rearranged as $\tilde{\mathbf{H}}_{s,i,j}^q = \text{diag}(\tilde{\mathbf{h}}_{s,i,j}^q)$ for $s = 0, 1$. More importantly, we can see that each submatrix $\tilde{\mathbf{M}}_{i,j}$ is a combination of diagonal and DFT/IDFT matrices.

So far, we have derived the signal model for MIMO-OFDMA systems in (4.15), (4.16), and (4.21). Its signal structure is the same as that in MIMO-OFDM systems. Again, we can exploit this special structure to reduce the required computational complexity as we did in Chapter 3. Thus, we can have low-complexity algorithms for MIMO-OFDMA systems. As for the required computational complexity, it is the same as that in MIMO-OFDM systems.

§ 4.2.1 Simulations

In this subsection, we provide simulation results to demonstrate the effectiveness of the proposed method. We consider a 2×2 MIMO-OFDMA system with $N = 128$, $N_g = 32$,

and $Q = 4$. The simulated fading channel is generated by Jakes' model [70]. In addition, the power delay profile of the q th user is characterized by an exponential function, i.e., $\sigma_{l,q}^2 = e^{-l \times \alpha_q / L} / \sum_{i=0}^{L-1} e^{-i \times \alpha_q / L}$, where l is the tap index, q is the user index, and L is the number of channel taps. For the setting of channel parameters, we let $\alpha_1 = 1$, $\alpha_2 = 3$, $\alpha_3 = 2$, $\alpha_4 = 1.5$, and $L = 15$. For the direct ZF and MMSE methods, we assume that the channel response is exactly known. For the N-ZF and N-MMSE methods, the parameters of the LTV channel model are obtained by LS fittings. The chosen modulation scheme is 16-QAM.

In the beginning, we discuss the ZF method. We consider two cases here; in the first case, f_d 's are set to $\{0.02, 0.05, 0.03, 0.04\}$ and in the second case, they are set to $\{0.04, 0.1, 0.08, 0.07\}$. Figure 4.5 shows the BER performance of case 1. In this figure, the performance of a two-tap FEQ method is also compared. The f_d 's in Fig. 4.1 and Fig. 4.5 are the same. From both figures, we find that the behavior of the BER curves is similar except that the BER is higher in Fig. 4.5. This is because the inter-antenna interference is introduced in a MIMO-OFDMA system. Figure 4.6 shows the BER performance in case 2. In this case, f_d 's are larger. As a result, we find that the N-ZF method needs three iterations to approach the direct ZF method.

Table 4.3 summarizes the required computational complexity for the direct ZF method and the N-ZF method in a 2×2 MIMO-OFDMA system. For case 1, two iterations are sufficient. The complexity ratio for multiplication/division/addition is 0.003/0.002/0.003. For case 2, three iterations are needed. The complexity ratio for multiplication/division/addition becomes 0.005/0.002/0.006. As we can see from these figures, significant complexity reduction can still be obtained even though the iteration number is three.

We now compare the required computational complexity for ICI mitigation in MIMO-OFDMA systems and in SISO-OFDMA systems. From Tables 4.1 and 4.3, we see that the multiplication complexity of the direct ZF method is increased up to almost eight times from SISO-OFDMA to 2×2 MIMO-OFDMA. For the N-ZF method, it is only increased about three times. Thus, the complexity reduction achieved by the N-ZF method is greater in a 2×2 MIMO-OFDMA system. In a SISO-OFDMA system, the ratio of multiplication is 0.007 ($k = 2$ and

$f'_d s = \{0.05, 0.02, 0.04, 0.03\}$), while that in a 2×2 MIMO-OFDMA system is 0.003 ($k = 2$ and $f'_d s = \{0.02, 0.05, 0.03, 0.04\}$). This is because the complexity of the direct ZF method is proportional to $\mathcal{O}(M^3 N^3)$, whereas that of the N-ZF method is proportional to $\mathcal{O}(MN \log_2 N)$. Even with three iterations, the complexity reduction achieved by the N-ZF method is still larger in MIMO-OFDMA systems. It can be inferred that the N-ZF method can save more computations when M or N becomes larger.

Now, we report simulation results for the MMSE method. We also consider the above two cases; case 1 is that f'_d 's are set to $\{0.02, 0.05, 0.03, 0.04\}$ and in case 2, they are set to $\{0.04, 0.1, 0.08, 0.07\}$. Figure 4.7 shows the BER performance of case 1. We find that the N-MMSE method with two iterations can avoid the error floor phenomenon of the two-tap FEQ method and in the meanwhile the N-MMSE method approaches the direct MMSE method. Figure 4.8 shows the result of case 2. In this figure, the behavior of the N-MMSE method is similar to that in Fig. 4.7 except the required number of iterations is three.

Table 4.4 summarizes the required computational complexity for the direct MMSE method and the N-MMSE method in a 2×2 MIMO-OFDMA system. For case 1, two iterations are enough. The complexity ratio of multiplication/division/addition is 0.0029/0.0389/0.0031. For case 2, three iterations are needed. The complexity ratio for multiplication/division/addition becomes 0.0056/0.0389/0.0062. From the above reports, it is clear that significant complexity reduction can be obtained by the N-MMSE method.

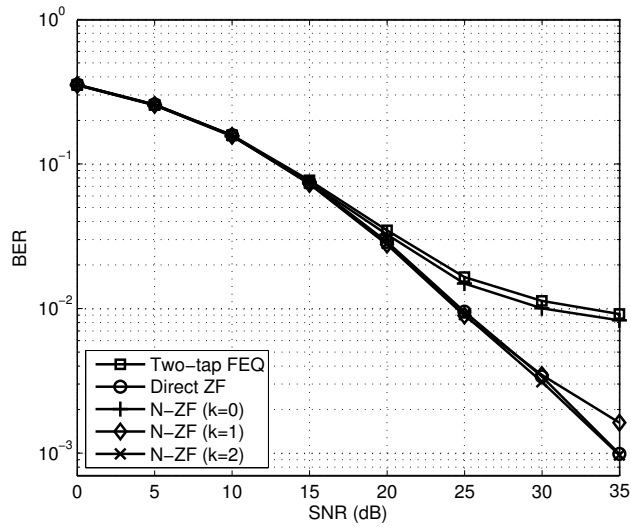


Figure 4.5: BER comparison among one-tap FEQ, direct ZF, and N-ZF ($D = 0, S = 2$) methods in a 2×2 MIMO-OFDMA system; $f_d = \{0.02, 0.05, 0.03, 0.04\}$ and 16-QAM modulation.

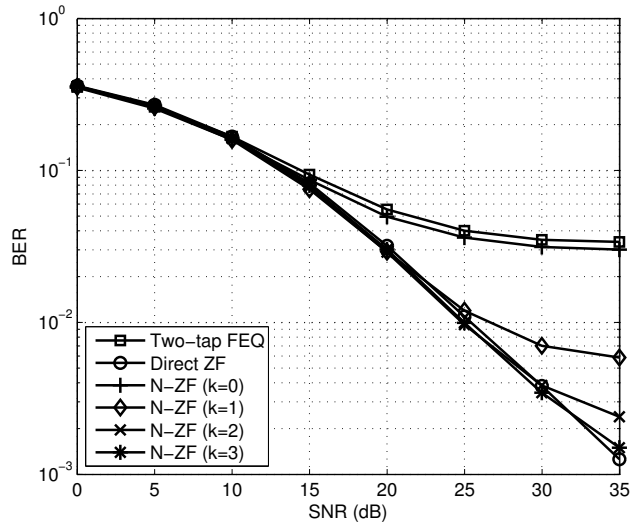


Figure 4.6: BER comparison among one-tap FEQ, direct ZF, and N-ZF ($D = 0, S = 2$) methods in a 2×2 MIMO-OFDMA system; $f_d = \{0.04, 0.1, 0.08, 0.07\}$ and 16-QAM modulation.

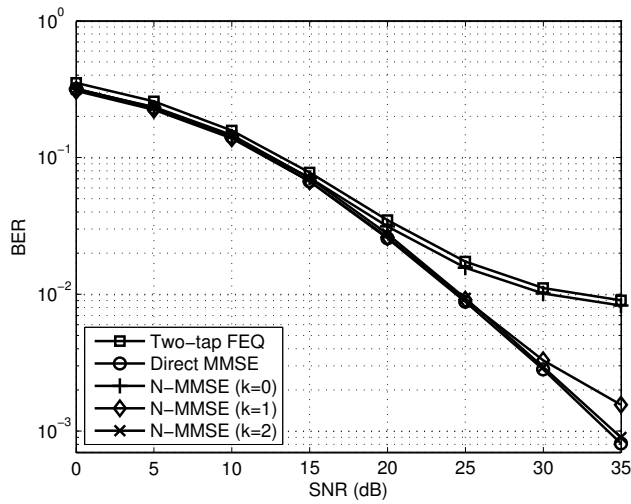


Figure 4.7: BER comparison among two-tap FEQ, direct MMSE, and N-MMSE ($D_1 = D_2 = 0$, $S_1 = S_2 = 2$) methods in a 2×2 MIMO-OFDMA system; $f_d = \{0.02, 0.05, 0.03, 0.04\}$ and 16-QAM modulation.

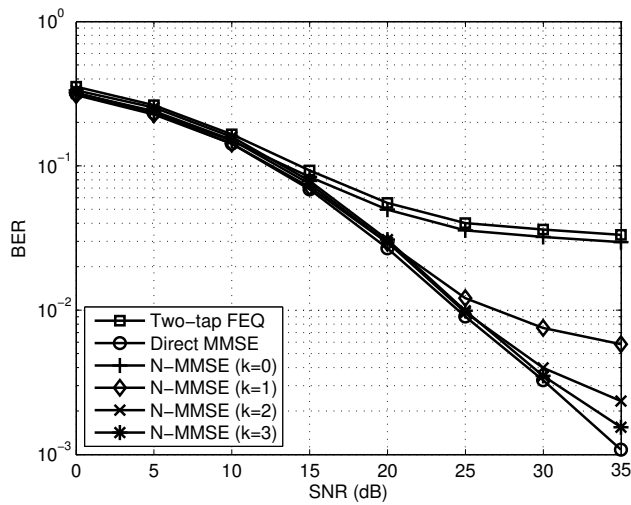


Figure 4.8: BER comparison among two-tap FEQ, direct MMSE, and N-MMSE ($D_1 = D_2 = 0$, $S_1 = S_2 = 2$) methods in a 2×2 MIMO-OFDMA system; $f_d = \{0.04, 0.1, 0.08, 0.07\}$ and 16-QAM modulation.

Table 4.3: Complexity comparison between direct ZF and N-ZF methods in a 2×2 MIMO-OFDMA system ($N = 128$, $D = 0$, and $S = 2$).

Methods	Real multiplications (ratio)	Real divisions (ratio)	Real additions (ratio)
Direct ZF	22828288	65792	22728576
N-ZF ($k = 1$)	35712(0.002)	128(0.002)	34560(0.002)
N-ZF ($k = 2$)	64384(0.003)	128(0.002)	65280(0.003)
N-ZF ($k = 3$)	121728(0.005)	128(0.002)	126720(0.006)



Table 4.4: Complexity comparison between the N-MMSE and direct MMSE methods in a 2×2 MIMO-OFDMA system ($N = 128$, $\{D_1, D_2\} = \{0, 0\}$, and $\{S_1, S_2\} = \{2, 2\}$).

Methods	Real multiplications (ratio)	Real divisions (ratio)	Real additions (ratio)
Direct MMSE	45086976	65792	45027072
N-MMSE ($k = 1$)	69377 (0.0015)	2560 (0.0389)	68352 (0.0015)
N-MMSE ($k = 2$)	130817 (0.0029)	2560 (0.0389)	139008 (0.0031)
N-MMSE ($k = 3$)	253697 (0.0056)	2560 (0.0389)	280320 (0.0062)



Chapter 5

CFO-induced ICI Mitigation for OFDMA Uplink Systems

§ 5.1 Signal Model



In the previous chapters, we discuss the mobility-induced ICI in SISO/MIMO-OFDM(A). Except for mobility, CFO also induces ICI in an OFDM-based system. In this chapter, we will focus on the CFO-induced ICI in an OFDMA uplink system. In an OFDMA uplink system with Q active users, the available bandwidth is divided into N equally spaced subbands. Each subcarrier uses a subband with bandwidth $1/(NT_s)$, where T_s is the sampling period. In such a system, Q users share the N subcarriers. Without loss of generality, we assume that each user uses $N_s = N/Q$ subcarriers. For the q th user, the transmitted frequency-domain signal at the k th subcarrier is denoted by \tilde{x}_k^q , where $k \in \Upsilon_q$ and Υ_q is the set of the subcarrier indices for the q th user. It is assumed that $\Upsilon_i \cap \Upsilon_j = \emptyset$ for $i \neq j$ and $\bigcup_{q=1}^Q \Upsilon_q = \{0, 1, 2, \dots, N-1\}$. Usually OFDMA adopts the interleaved subcarrier allocation scheme. In other words, $\Upsilon_q = \{q-1, q-1+Q, \dots, q-1+(N/Q-1) \times Q\}$. Since the subcarriers assigned to the different users are interleaved in the whole bandwidth, this scheme can achieve the maximum frequency diversity. For each user, we assume that the CP length is long enough to prevent the ISI effect.

Note that the channel we consider here is quasi-static, i.e., it is time-invariant in one OFDMA symbol period.

Consider a specific OFDMA symbol for the q th user. The channel output signal, after CP removal, can be expressed as

$$\mathbf{y}^q = \mathbf{H}^q \mathbf{x}^q, \quad (5.1)$$

where \mathbf{x}^q is the vector representation of the q th user's time-domain OFDMA symbol, i.e., $\mathbf{x}^q = (1/\sqrt{N})\mathbf{G}^H \tilde{\mathbf{x}}^q$. Here $\tilde{\mathbf{x}}^q$ is the corresponding frequency-domain signal vector. The matrix, \mathbf{H}^q , is a circulant channel matrix with the first column vector being \mathbf{h}^q which is the channel response \mathbf{x}^q experiences. Zeros are padded in \mathbf{h}^q since the channel length is assumed to be smaller than the CP length. Note that \mathbf{H}^q can be decoupled as $\mathbf{G}^H \tilde{\mathbf{H}}^q \mathbf{G}$, where $\tilde{\mathbf{H}}^q$ is a diagonal matrix with the diagonal vector of $\tilde{\mathbf{h}}^q = \sqrt{N}\mathbf{G}\mathbf{h}^q$ and elements of $\tilde{\mathbf{x}}^q$ are nonzeros only in the designated subcarrier positions. The receive time-domain OFDMA symbol from Q active users can be expressed as

$$\begin{aligned} \mathbf{r} &= \sum_{q=1}^Q \mathbf{E}^q \mathbf{y}^q + \mathbf{v} \\ &= \frac{1}{\sqrt{N}} \sum_{q=1}^Q \mathbf{E}^q \mathbf{H}^q \mathbf{G}^H \tilde{\mathbf{x}}^q + \mathbf{v}. \end{aligned} \quad (5.2)$$

In (5.2), \mathbf{E}^q denotes a diagonal matrix and its diagonal elements consist of U_q^k , $0 \leq k \leq N-1$, where $U_q^k = e^{\frac{j2\pi\epsilon_q k}{N}}$ and ϵ_q is the normalized CFO (with respect to the subcarrier spacing) for user q . Also, \mathbf{v} denotes the noise vector. After the FFT operation, we have the corresponding frequency-domain signal as

$$\begin{aligned} \tilde{\mathbf{r}} &= \sqrt{N}\mathbf{G}\mathbf{r} \\ &= \sum_{q=1}^Q \tilde{\mathbf{E}}^q \tilde{\mathbf{y}}^q + \tilde{\mathbf{v}}, \end{aligned} \quad (5.3)$$

where $\tilde{\mathbf{v}} = \sqrt{N}\mathbf{G}\mathbf{v}$, $\tilde{\mathbf{E}}^q = \mathbf{G}\mathbf{E}^q\mathbf{G}^H$, and $\tilde{\mathbf{y}}^q = \tilde{\mathbf{H}}^q \tilde{\mathbf{x}}^q$. Note that $\tilde{\mathbf{E}}^q$ is a circulant matrix. Denote its first column as $\tilde{\mathbf{e}}^q$. Then, $\tilde{\mathbf{e}}^q = (1/\sqrt{N})\mathbf{G}\mathbf{e}^q$, where $\mathbf{e}^q = [U_q^0, U_q^1, \dots, U_q^{N-1}]^T$. Let

$\tilde{\mathbf{x}} = [\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_{N-1}]^T$ be the composite transmit data (for all users), $\tilde{\mathbf{h}} = [\tilde{h}_0, \tilde{h}_1, \dots, \tilde{h}_{N-1}]^T$ be the composite channel frequency response (for all users), and $\tilde{\mathbf{h}}^q = [\tilde{h}_0^q, \tilde{h}_1^q, \dots, \tilde{h}_{N-1}^q]^T$. Then, $\tilde{x}_k = \tilde{x}_k^q$ and $\tilde{h}_k = \tilde{h}_k^q$ when $k \in \Upsilon_q$. We also define a diagonal selection matrix \mathbf{S}^q such that

$$\mathbf{S}^q(j, j) = \begin{cases} 1, & j \in \Upsilon_q, \\ 0, & \text{otherwise.} \end{cases} \quad (5.4)$$

Thus we can express the receive signal in (5.3) as [50], [51]

$$\begin{aligned} \tilde{\mathbf{r}} &= \left(\sum_{q=1}^Q \tilde{\mathbf{E}}^q \mathbf{S}^q \right) \tilde{\mathbf{H}} \tilde{\mathbf{x}} + \tilde{\mathbf{v}} \\ &= \tilde{\mathbf{M}} \tilde{\mathbf{y}} + \tilde{\mathbf{v}}, \end{aligned} \quad (5.5)$$

where $\tilde{\mathbf{H}}$ is a diagonal matrix with the diagonal vector being $\tilde{\mathbf{h}}$, $\tilde{\mathbf{y}} = \tilde{\mathbf{H}} \tilde{\mathbf{x}}$, and $\tilde{\mathbf{M}} = \sum_{q=1}^Q \tilde{\mathbf{E}}^q \mathbf{S}^q = \sum_{q=1}^Q \mathbf{G} \mathbf{E}^q \mathbf{G}^H \mathbf{S}^q$ is the CFO-induced ICI matrix. From the above formulation, we can see that the ICI matrix is composed of diagonal and DFT/IDFT matrices. This ICI matrix structure will be exploited in the developed low-complexity method.

§ 5.2 Previous Methods

§ 5.2.1 Conventional Method

Since CFO induces the self-ICI and MUI, the system performance can be seriously affected. Thus it is crucial either to correct CFOs or to mitigate the ICI effect. Several ICI mitigation methods have been proposed in the literature [45], [46], [47]. In this subsection, we will first review the work in [45].

In a single-user OFDM system, CFO is an important factor affecting the system performance. Since CFO produces a phase rotation on the receive time-domain signal, the conventional method compensates for the effect by a phase de-rotation operation. In an OFDMA uplink

system, this method can also be used to compensate for the CFO effect [45]. From (5.2), the compensated signal for user i by the conventional method can be written as

$$\begin{aligned}
\bar{\mathbf{y}}_C^i &= (\mathbf{E}^i)^H \mathbf{r} \\
&= \sum_{q=1}^Q (\mathbf{E}^i)^H \mathbf{E}^q \mathbf{y}^q + (\mathbf{E}^i)^H \mathbf{v} \\
&= \mathbf{y}^i + \underbrace{\sum_{\substack{q=1 \\ q \neq i}}^Q (\mathbf{E}^i)^H \mathbf{E}^q \mathbf{y}^q}_{\text{MUI}} + (\mathbf{E}^i)^H \mathbf{v}.
\end{aligned} \tag{5.6}$$

From (5.6), we can see that the first term is the desired signal and the second term is the MUI. Note that the MUI after CFO compensation can become smaller or larger. If the CFO compensation for the i th user makes the resultant CFO of the q th user, $\epsilon_q - \epsilon_i$, become larger, a larger MUI from the q th user will be induced. We then conclude that the conventional method can compensate for the self-induced ICI, but not MUI.

§ 5.2.2 CLJL Method

Since the time-domain conventional method in [45] may induce a larger MUI, the method in [46], called the CLJL method, compensates for the CFO-induced ICI in the frequency domain. From (5.3), the CFO-compensated signal for the i th user can be written as follows:

$$\begin{aligned}
\bar{\mathbf{y}}_{CLJL}^i &= \mathbf{S}^i (\tilde{\mathbf{E}}^i)^H \mathbf{S}^i \tilde{\mathbf{r}} \\
&= \mathbf{S}^i (\tilde{\mathbf{E}}^i)^H \mathbf{S}^i \sum_{q=1}^Q \tilde{\mathbf{E}}^q \tilde{\mathbf{y}}^q + \mathbf{S}^i (\tilde{\mathbf{E}}^i)^H \mathbf{S}^i \tilde{\mathbf{v}} \\
&= \mathbf{S}^i (\tilde{\mathbf{E}}^i)^H \mathbf{S}^i \tilde{\mathbf{E}}^i \tilde{\mathbf{y}}^i + \underbrace{\sum_{\substack{q=1 \\ q \neq i}}^Q \mathbf{S}^i (\tilde{\mathbf{E}}^i)^H \mathbf{S}^i \tilde{\mathbf{E}}^q \tilde{\mathbf{y}}^q}_{\text{MUI}} + \mathbf{S}^i (\tilde{\mathbf{E}}^i)^H \mathbf{S}^i \tilde{\mathbf{v}},
\end{aligned} \tag{5.7}$$

From (5.7), we can see that the first term is the desired signal after CFO compensation, and the second term is the MUI. It is apparent that the desired signal suffers from an amplitude

reduction, and the MUI still exists. When the CFO is large, the performance of the CLJL method will be unsatisfactory.

§ 5.2.3 CLJL-PIC Method

Since the methods in [45], [46] suffer from the MUI, the method in [47], called the CLJL-PIC method, further suppress the MUI using the PIC technique. The CLJL-PIC method can be summarized in the following steps:

1. Using the CLJL method to obtain the initial estimate $\bar{\mathbf{y}}_0^i$ for all active users

$$\bar{\mathbf{y}}_0^i = \mathbf{S}^i (\tilde{\mathbf{E}}^i)^H \mathbf{S}^i \tilde{\mathbf{r}}, \quad i = 1, 2, \dots, Q, \quad (5.8)$$

where $\bar{\mathbf{y}}_p^i$ is the estimated data of user i at the p th iteration.

2. Regenerate the MUI by $\tilde{\mathbf{E}}^i$ and $\bar{\mathbf{y}}_{p-1}^i$ obtained at the $(p-1)$ th stage. Then cancel the regenerated MUI from the received signal

$$\tilde{\mathbf{r}}_p^i = \tilde{\mathbf{r}} - \sum_{\substack{q=1 \\ q \neq i}}^Q \tilde{\mathbf{E}}^q \bar{\mathbf{y}}_{p-1}^q, \quad i = 1, 2, \dots, Q, \quad (5.9)$$

where $\tilde{\mathbf{r}}_p^i$ is the output signal after the PIC processing at the p th stage.

3. Using the CLJL method to compensate for the CFO for each active user

$$\bar{\mathbf{y}}_p^i = \mathbf{S}^i (\tilde{\mathbf{E}}^i)^H \mathbf{S}^i \tilde{\mathbf{r}}_p^i, \quad i = 1, 2, \dots, Q. \quad (5.10)$$

4. Repeat Step 2 and Step 3.

The main drawback of this method is that its performance is affected by the initial estimate $\bar{\mathbf{y}}_0^i$ used for the PIC processing. When a large CFO occurs, the performance improvement by the PIC technique with the poor initial estimates (owing to the large MUI) will be limited. If

the modulation scheme has a large QAM-size such as 64-QAM, the performance will become sensitive to the residual MUI. Note that Step 2 can be implemented more efficiently as follows:

$$\tilde{\mathbf{r}}_p^i = \left(\tilde{\mathbf{r}} - \sum_{q=1}^Q \tilde{\mathbf{E}}^q \tilde{\mathbf{y}}_{p-1}^q \right) + \tilde{\mathbf{E}}^i \tilde{\mathbf{y}}_{p-1}^i, \quad i = 1, 2, \dots, Q. \quad (5.11)$$

This formula will be used to evaluate its computational complexity in a comparison described later.

§ 5.3 ZF Method

§ 5.3.1 Proposed Newton-ZF Method

From (5.5), we can see that a straightforward method to compensate for the CFO effect is the ZF method given by [51]

$$\tilde{\mathbf{y}}_{ZF} = \tilde{\mathbf{M}}^{-1} \tilde{\mathbf{r}}. \quad (5.12)$$

Although the direct ZF method can completely suppress the CFO effect, it needs to invert the ICI matrix with dimension N , the FFT size. When the size is large, the required computational complexity can become prohibitively high. Unfortunately, in real-world applications, the symbol size is usually large. For example, for IEEE 802.16e, the size can be as large as 2048. Here, we propose a low-complexity ZF method to solve the problem. The main idea is to use an iterative procedure such that the direct matrix inversion can be avoided. Specifically, we use Newton's method as we did in the case of mobility-induced ICI.

Base on the previous discussion on the mobility-induced ICI, we also approximate the matrix inversion of $\tilde{\mathbf{M}}$ via the expanded form of Newton's iteration as

$$\mathbf{W}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} (\mathbf{W}_0 \tilde{\mathbf{M}})^m \mathbf{W}_0. \quad (5.13)$$

Moreover, our final objective is to obtain the CFO-compensated result $\mathbf{W}_k \tilde{\mathbf{r}}$ not the matrix inversion \mathbf{W}_k itself. Multiplying both sides of (5.13) by the receive signal $\tilde{\mathbf{r}}$, we then have

$$\mathbf{W}_k \tilde{\mathbf{r}} = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} (\mathbf{W}_0 \tilde{\mathbf{M}})^m \mathbf{W}_0 \tilde{\mathbf{r}}. \quad (5.14)$$

Let $\bar{\mathbf{y}}_k = \mathbf{W}_k \tilde{\mathbf{r}}$ and $\bar{\mathbf{s}}_m = (\mathbf{W}_0 \tilde{\mathbf{M}})^m \mathbf{W}_0 \tilde{\mathbf{r}}$. According to these definitions, we can rewrite (5.14) as

$$\bar{\mathbf{y}}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \bar{\mathbf{s}}_m. \quad (5.15)$$

Based on the definition of $\bar{\mathbf{s}}_m$, it is simple to see that

$$\bar{\mathbf{s}}_{m+1} = (\mathbf{W}_0 \tilde{\mathbf{M}}) \bar{\mathbf{s}}_m. \quad (5.16)$$

As a result, $\bar{\mathbf{s}}_m$ can be recursively calculated. With this approach, we have transformed the matrix-to-matrix multiplications in (5.13) into the matrix-to-vector multiplications in (5.15) and (5.16).

To complete our low-complexity algorithm, we further let \mathbf{W}_0 be a diagonal matrix and explore the special structure inherent in the CFO-induced ICI matrix, $\tilde{\mathbf{M}}$. Recall that $\tilde{\mathbf{M}} = \sum_{q=1}^Q \mathbf{G} \mathbf{E}^q \mathbf{G}^H \mathbf{S}^q$. Thus, we can then rewrite (5.16) as

$$\bar{\mathbf{s}}_{m+1} = \mathbf{W}_0 \mathbf{G} \sum_{q=1}^Q \mathbf{E}^q (\mathbf{G}^H \mathbf{S}^q \bar{\mathbf{s}}_m). \quad (5.17)$$

Note that operations in (5.17) only involve vector multiplications, IDFTs, and a DFT. It is well-known that DFT/IDFT can be implemented with FFT/IFFT and then the required computational complexity can be greatly reduced. Thus, evaluation of (5.15) only involves vector multiplications, FFTs, and IFFTs. The required computational complexity is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}((Q+1)N \log_2 N)$.

With the interleaved-OFDMA structure, the computational complexity can be reduced further. Let $\bar{\mathbf{s}}_m = [\bar{s}_m(0), \bar{s}_m(1), \dots, \bar{s}_m(N-1)]^T$ and $\mathbf{u}_m^q = \mathbf{S}^q \bar{\mathbf{s}}_m = [u_m^q(0), u_m^q(1), \dots, u_m^q(N-1)]^T$.

1)]^T. From the definition of \mathbf{S}^q , we have

$$u_m^q(i) = \begin{cases} \bar{s}_m(i), & i \in \Upsilon_q, \\ 0 & , \text{ otherwise.} \end{cases} \quad (5.18)$$

This is to say that \mathbf{u}_m^q corresponds to an upsampled sequence of the desired elements in \bar{s}_m . The nonzero elements in \mathbf{u}_m^q , denoted by $\mathbf{d}_m^q = [\bar{s}_m(q-1), \bar{s}_m(q-1+Q), \dots, \bar{s}_m(q-1+(N/Q-1) \times Q)]^T$, can be obtained by circularly shifting \mathbf{u}_m^q with $q-1$ elements and downsampling the result with a factor of Q . Let $\bar{\mathbf{d}}_m^q = (1/\sqrt{Q})\mathbf{G}_{N_s}^H \mathbf{d}_m^q$, where \mathbf{G}_{N_s} is an $N_s \times N_s$ DFT matrix, and construct an $N \times 1$ vector by duplicating $\bar{\mathbf{d}}_m^q$, Q times shown as

$$\mathbf{a}_m^q = [(\bar{\mathbf{d}}_m^q)^T, \dots, (\bar{\mathbf{d}}_m^q)^T]^T. \quad (5.19)$$

Consequently, we can obtain $\mathbf{G}^H \mathbf{S}^q \bar{s}_m$ by the following method

$$\mathbf{G}^H \mathbf{S}^q \bar{s}_m = \mathbf{C}^q \mathbf{a}_m^q, \quad (5.20)$$

where \mathbf{C}^q is a diagonal matrix with the diagonal vector of $\mathbf{c}^q = [Z_q^0, Z_q^1, \dots, Z_q^{N-1}]^T$ and $Z_q^k = e^{j2\pi(q-1)k/N}$. Note that the operation \mathbf{C}^q results from the circular shift of \mathbf{u}_m^q . Equation (5.20) implies that we can implement $\mathbf{G}^H \mathbf{S}^q \bar{s}_m$ by an IDFT with dimension N/Q instead of N . Using this approach, we can reduce the computational complexity further by rewriting (5.16) as

$$\bar{s}_{m+1} = \mathbf{W}_0 \mathbf{G} \sum_{q=1}^Q \mathbf{E}^q \mathbf{C}^q \mathbf{a}_m^q. \quad (5.21)$$

Note that \mathbf{a}_m^q is a column vector and both \mathbf{C}^q and \mathbf{E}^q are complex diagonal matrices. As assumed, \mathbf{W}_0 is a diagonal matrix. Equation (5.21) only involves one DFT with size N , Q IDFTs with size N/Q , and some vector operations. As mentioned, DFT/IDFT can be efficiently implemented with FFT/IFFT. Finally, the required computational complexity is reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log_2(N^2/Q))$.

The final thing we have to deal with is how to determine the initial matrix \mathbf{W}_0 . A well-designed initial values can reduce the number of iterations significantly and provide good mit-

igation performance. Let $\mathbf{W}_0 = \text{diag}([w_0, w_1, \dots, w_{N-1}]^T)$. Again, we adopt the minimum-Frobenius-norm criterion to obtain optimum initial values. The criterion is given by

$$\mathbf{W}_{opt,0} = \arg \min_{\mathbf{w}_0} \|\mathbf{I}_N - \mathbf{W}_0 \widetilde{\mathbf{M}}\|_F^2. \quad (5.22)$$

We can expand $\|\mathbf{I}_N - \mathbf{W}_0 \widetilde{\mathbf{M}}\|_F^2$ as follows:

$$\|\mathbf{I}_N - \mathbf{W}_0 \widetilde{\mathbf{M}}\|_F^2 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\delta(i-j) - w_i \widetilde{m}_{i,j}|^2, \quad (5.23)$$

where $\widetilde{m}_{i,j} = \widetilde{\mathbf{M}}(i, j)$. Then, the optimum initial values for minimizing (5.23) can be obtained by setting the first derivative of $\|\mathbf{I}_N - \mathbf{W}_0 \widetilde{\mathbf{M}}\|_F^2$ with respect to w_k^* to zero. The first derivative of $\|\mathbf{I}_N - \mathbf{W}_0 \widetilde{\mathbf{M}}\|_F^2$ with respect to w_k^* can be found in

$$\frac{\partial}{\partial w_k^*} \left\{ \|\mathbf{I}_N - \mathbf{W}_0 \widetilde{\mathbf{M}}\|_F^2 \right\} = -\widetilde{m}_{k,k}^* + w_{opt,k} \sum_{j=0}^{N-1} |\widetilde{m}_{k,j}|^2. \quad (5.24)$$

As a result, we can have the optimal initial value w_k , denoted as $w_{opt,k}$, as

$$w_{opt,k} = \frac{\widetilde{m}_{k,k}^*}{\sum_{j=0}^{N-1} |\widetilde{m}_{k,j}|^2}. \quad (5.25)$$

For further complexity reduction, we can make an approximation to (5.25) as

$$w_{opt,k} \approx \frac{\widetilde{m}_{k,k}^*}{\sum_{j=\langle k-S, k+S, N \rangle} |\widetilde{m}_{k,j}|^2}, \quad (5.26)$$

where S is a parameter controlling the number of ICI terms considered ($0 \leq S \leq N/2 - 1$). The approximation is based on the fact that the CFO-induced ICI on a subcarrier mainly comes from neighboring subcarriers. Moreover, this approximation is only for the initial matrix calculation. The final result will be updated by Newton's method. For easy reference, we denote this method as the N-ZF method.

For the direct ZF method, the matrix inversion is obtained by solving a set of linear equations as $\widetilde{\mathbf{M}} \overline{\mathbf{y}}_{ZF} = \widetilde{\mathbf{r}}$, where $\overline{\mathbf{y}}_{ZF}$ is the ZF-compensated $\widetilde{\mathbf{y}}$. This can be implemented by triangular factorization (Gaussian elimination), and forward and backward substitution [69]. Finally, for signal detection, a conventional one-tap FEQ is applied to each subcarrier (compensating for the channel effect). We can express the result as $\overline{\mathbf{x}} = \widetilde{\mathbf{H}}^{-1} \overline{\mathbf{y}}$, where $\overline{\mathbf{x}}$ is the estimate of $\widetilde{\mathbf{x}}$ while $\overline{\mathbf{y}}$ is the CFO-compensated $\widetilde{\mathbf{y}}$.

§ 5.3.2 Pre-compensation Method

If CFOs are large, the performance of the proposed method may be affected. The larger the CFO, the worse the performance it will result. In OFDM systems, we can compensate for the CFO effect with a phase de-rotation operation. Although the same method cannot be used to compensate for all CFOs here, it can be used to reduce their magnitudes in some cases. Note that this is equivalent to a pre-compensation (PC) approach. Let the receive signal be pre-compensated by a normalized CFO value ϵ . Thus, the resultant CFO for the q th user, denoted as φ_q , is now changed to

$$\varphi_q = \epsilon_q - \epsilon. \quad (5.27)$$

Here, we propose a minimum square error criterion, shown below, to obtain optimum ϵ , i.e.,

$$\epsilon_{opt} = \min_{\epsilon} \sum_{q=1}^Q \varphi_q^2. \quad (5.28)$$

Setting the first derivative of $\sum_{q=1}^Q \varphi_q^2$ with respect to ϵ to zero, we can obtain the optimum ϵ for (5.28) as

$$\epsilon_{opt} = \frac{1}{Q} \sum_{q=1}^Q \epsilon_q. \quad (5.29)$$

From this result, it is simple to see that the optimum ϵ is just the mean of all CFOs. Compensating for a pre-determined CFO can be implemented by windowing the receive time-domain signal with the windowing vector $\mathbf{w} = [U_o^0, U_o^1, \dots, U_o^{N-1}]^T$, where $U_o^k = e^{-j \frac{2\pi k \epsilon_{opt} t}{N}}$. Thus, the PC method will need extra N complex multiplications which is minor compared to other operations. In simulations, we will show that the PC method can greatly enhance the performance of the proposed algorithm in some cases.

§ 5.3.3 Complexity Analysis

In the previous subsections, we have proposed the N-ZF method for an interleaved-OFDMA uplink system. In this subsection, we will analyze the computational complexity of the proposed

method, and compare it with that of existing methods. From (5.15) and (5.16), we can clearly see that the computational complexity of the proposed method mainly consists of the following three parts:

1. $\bar{\mathbf{s}}_m$ iteration, where $\bar{\mathbf{s}}_{m+1} = (\mathbf{W}_0 \widetilde{\mathbf{M}}) \bar{\mathbf{s}}_m$ and $\bar{\mathbf{s}}_0 = \mathbf{W}_0 \tilde{\mathbf{r}}$,
2. $\widetilde{\mathbf{M}}$ construction with $\widetilde{\mathbf{M}} = \sum_{q=1}^Q \widetilde{\mathbf{E}}^q \mathbf{S}^q$,
3. \mathbf{W}_0 calculation with (5.26).

Since the operations in $\bar{\mathbf{s}}_m$ iteration can be implemented by (5.21), we only need one FFT operation with size N , Q IFFT operations with size N/Q , and a couple of vector operations. As a result, we require $(N/2)[\log_2 N + \log_2(N/Q) + 4Q]$ CMs and $N[\log_2 N + \log_2(N/Q) + (Q-1)]$ CAs. In addition, we need N CMs for $\bar{\mathbf{s}}_0 = \mathbf{W}_0 \tilde{\mathbf{r}}$, and $2N$ RMs for each $c_m^k \bar{\mathbf{s}}_m$ in (5.15). As to the construction of $\widetilde{\mathbf{M}}$, we first have to obtain $\widetilde{\mathbf{E}}^q$ by $\tilde{\mathbf{e}}^q = (1/\sqrt{N})\mathbf{G}\mathbf{e}^q$ for each user. This will require $(1/2)QN \log_2(N)$ CMs, and $QN \log_2(N)$ CAs. Despite the diagonal property of \mathbf{W}_0 , we can use the special structure of $\widetilde{\mathbf{M}}$ to reduce the complexity further. To see this, we can rewrite $\widetilde{\mathbf{M}}$ as

$$\widetilde{\mathbf{M}}(:, j) = \mathcal{C}_j(\tilde{\mathbf{e}}^{\langle j, Q \rangle + 1}), \quad j = 0, 1, \dots, N-1, \quad (5.30)$$

where $\mathcal{C}_j(\mathbf{a})$ denotes the circular shift of a column vector \mathbf{a} downwards or a row vector \mathbf{a} leftwards by j elements. From (5.30), it is straightforward to see that

$$\widetilde{\mathbf{M}}(i, :) = \mathcal{C}_{jQ}(\widetilde{\mathbf{M}}(i + jQ, :)), \quad i = 0, 1, \dots, Q-1, \quad j = 0, 1, \dots, N_s - 1. \quad (5.31)$$

As a result, we only have to calculate $\mathbf{W}_0(i, i)$ for $i = 0, \dots, Q-1$. This is because

$$\mathbf{W}_0(i, i) = \mathbf{W}_0(i + jQ, i + jQ), \quad i = 0, 1, \dots, Q-1, \quad j = 0, 1, \dots, N_s - 1. \quad (5.32)$$

As mentioned, we can only take some neighboring ICI terms into account for \mathbf{W}_0 calculation. Let $\tilde{\mathbf{e}}^q = [\tilde{e}_0^q, \tilde{e}_1^q, \dots, \tilde{e}_{N-1}^q]^T$. Thus, we only need the following values: $\{\tilde{e}_{(N-S)}^q, \dots, \tilde{e}_{(N-1)}^q, \tilde{e}_0^q, \tilde{e}_1^q, \dots, \tilde{e}_S^q\}$ for $q = 1, 2, \dots, Q$. As a result, we require $Q(2S+1)$ CMs, QS CAs, and $2Q$ RDs

for \mathbf{W}_0 calculation. The required computational complexity is summarized in Table 5.1. For the proposed method with PC, only extra N complex multiplications are required. Table 5.1 also shows the required computational complexity of the CLJL-PIC method and the direct ZF method [69].

Table 5.1: Complexity comparison among proposed method, CLJL-PIC method, and direct ZF method.

Complexity	Proposed method	CLJL-PIC method	Direct ZF method
Real multi- plications	$2(2^k - 1 + Q)N \log_2(N) +$ $2(2^k - 1)N \log_2(N/Q) +$ $[8(2^k - 1)Q + 2(2^k + 1)]N +$ $4Q(2S + 1)$	$4[p + (p+1)/Q]N^2 +$ $2QN \log_2(N)$	$\frac{4}{3}N^3 + 5N^2 +$ $2QN \log_2(N) - \frac{1}{3}N$
Real divi- sions	$2Q$	0	$N^2 + N$
Real addi- tions	$3(2^k - 1 + Q)N \log_2(N) +$ $3(2^k - 1)N \log_2(N/Q) +$ $[6(2^k - 1)Q + 2]N +$ $2Q(3S + 1)$	$4[p + (p+1)/Q]N^2 +$ $3QN \log_2(N) - 2N$	$\frac{4}{3}N^3 + \frac{7}{2}N^2 +$ $3QN \log_2(N) - \frac{11}{6}N$

§ 5.3.4 Performance Analysis

For an iterative algorithm, the convergence problem is usually a main concern. The proposed algorithm uses Newton's iterative algorithm; however, the convergence is less critical here. This is due to two facts described below. The first one is that for the application considered here, the proposed algorithm converges for most of the cases; only in few cases, will it diverge. The second is that the required number of iterations is pre-determined. Thanks to the fast

convergence property of Newton's iteration and good initial values we developed, only two or three iterations are necessary if it converges. Since the iteration number is finite, the proposed algorithm will never diverge. As shown below, even for divergence cases, SINR can still be increased for the designated iteration number. We will provide two analysis approaches mainly based on the analysis in Chapter 2. The first one is simpler, but the result is only approximated; the other one is complicated, but the result is exact. The first approach can provide an intuitive understanding of the convergence behavior of Newton's method. Here, we start with the simpler one. We first perform the eigenvalue decomposition for $\tilde{\mathbf{R}}_0$ as follows:

$$\tilde{\mathbf{R}}_0 = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}, \quad (5.33)$$

where $\mathbf{U} = [\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1}]$ is a matrix composed of eigenvectors of \mathbf{R}_0 , and $\mathbf{D} = \text{diag}([\lambda_0, \lambda_1, \dots, \lambda_{N-1}]^T)$ is a diagonal matrix having the i th eigenvalue, λ_i , as its i th diagonal element. We assume that $|\lambda_i| \geq |\lambda_j|$ for $i \leq j$. Since $\tilde{\mathbf{R}}_k = \tilde{\mathbf{R}}_{k-1}^2$, we can then decompose $\tilde{\mathbf{R}}_k$ as

$$\begin{aligned} \tilde{\mathbf{R}}_k &= \tilde{\mathbf{R}}_0^{2^k} \\ &= \mathbf{U}\mathbf{D}^{2^k}\mathbf{U}^{-1}. \end{aligned} \quad (5.34)$$

If $|\lambda_0| < 1$, then $\tilde{\mathbf{R}}_k \rightarrow \mathbf{0}_N$ as $k \rightarrow \infty$. Thus, we can have the convergence condition for Newton's iteration as $\rho(\tilde{\mathbf{R}}_0) < 1$, where $\rho(\tilde{\mathbf{R}}_0)$ denotes the spectral radius of $\tilde{\mathbf{R}}_0$. This is to say, for Newton's iteration to converge, the amplitudes of all eigenvalues of $\tilde{\mathbf{R}}_0$ have to be smaller than one. As mentioned, this condition holds for most of the cases. In few cases, the condition does not hold; however, the number of eigenvalues with amplitudes greater than one is small and their amplitudes do not deviate from one much. These results can be easily observed from simulations though difficult to be proved theoretically. In what follows, we will show that even for divergence cases, we may still benefit from Newton's iteration.

Let $\mathbf{U}^{-1} = [\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{N-1}]^T$. By definition, $\tilde{\mathbf{R}}_k = \mathbf{I}_N - \mathbf{W}_k\tilde{\mathbf{M}}$. We can represent the

CFO-compensated ICI matrix as

$$\begin{aligned}\mathbf{W}_k \widetilde{\mathbf{M}} &= \mathbf{I}_N - \widetilde{\mathbf{R}}_k \\ &= \mathbf{I}_N - \sum_{i=0}^{N-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T.\end{aligned}\quad (5.35)$$

Let $|\lambda_i| > 1$ for $i = 0, 1, \dots, P-1$ and $|\lambda_i| < 1$ for $i = P, P+1, \dots, N-1$. Thus, we can rewrite (5.35) as

$$\mathbf{W}_k \widetilde{\mathbf{M}} = \mathbf{I}_N - \sum_{i=0}^{P-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T - \sum_{j=P}^{N-1} \lambda_j^{2^k} \mathbf{u}_j \mathbf{p}_j^T. \quad (5.36)$$

As for \mathbf{W}_k , we can reformulate it as

$$\begin{aligned}\mathbf{W}_k &= (2\mathbf{I}_N - \mathbf{W}_{k-1} \widetilde{\mathbf{M}}) \mathbf{W}_{k-1} \\ &= (\mathbf{I}_N + \widetilde{\mathbf{R}}_{k-1})(\mathbf{I}_N + \widetilde{\mathbf{R}}_{k-2}) \dots (\mathbf{I}_N + \widetilde{\mathbf{R}}_0) \mathbf{W}_0.\end{aligned}\quad (5.37)$$

Using (5.34), we can further express \mathbf{W}_k as

$$\begin{aligned}\mathbf{W}_k &= \mathbf{U}(\mathbf{I}_N + \mathbf{D}^{2^{k-1}}) \mathbf{U}^{-1} \mathbf{U}(\mathbf{I}_N + \mathbf{D}^{2^{k-2}}) \mathbf{U}^{-1} \dots \mathbf{U}(\mathbf{I}_N + \mathbf{D}) \mathbf{U}^{-1} \mathbf{W}_0 \\ &= \sum_{j=0}^{N-1} \left\{ \prod_{i=0}^{k-1} (1 + \lambda_j^{2^i}) \right\} \mathbf{u}_j \mathbf{p}_j^T \mathbf{W}_0 \\ &= \sum_{j=0}^{N-1} \phi_{j,k} \mathbf{u}_j \mathbf{p}_j^T \mathbf{W}_0,\end{aligned}\quad (5.38)$$

where $\phi_{j,k} = \prod_{i=0}^{k-1} (1 + \lambda_j^{2^i})$. With (5.36) and (5.38), the CFO-compensated signal can be expressed as

$$\begin{aligned}\bar{\mathbf{y}}_k &= \mathbf{W}_k \widetilde{\mathbf{M}} \tilde{\mathbf{y}} + \mathbf{W}_k \tilde{\mathbf{v}} \\ &= \tilde{\mathbf{y}} - \sum_{i=0}^{P-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T \tilde{\mathbf{y}} - \sum_{j=P}^{N-1} \lambda_j^{2^k} \mathbf{u}_j \mathbf{p}_j^T \tilde{\mathbf{y}} + \tilde{\mathbf{v}}_k \\ &= \tilde{\mathbf{y}} - \sum_{i=0}^{P-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T \tilde{\mathbf{y}} - \sum_{j=P}^{N-1} \lambda_j^{2^k} \mathbf{u}_j \mathbf{p}_j^T \tilde{\mathbf{y}} + \sum_{j=0}^{N-1} \phi_{j,k} \mathbf{u}_j \mathbf{p}_j^T \tilde{\mathbf{v}}_0,\end{aligned}\quad (5.39)$$

where $\tilde{\mathbf{v}}_k = \mathbf{W}_k \tilde{\mathbf{v}}$. Since the eigenvectors $\{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1}\}$ span the N -dimensional space, we can decompose $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{v}}_0$ using these vectors. Let $\tilde{\mathbf{y}} = \sum_{l=0}^{N-1} \beta_l \mathbf{u}_l$ and $\tilde{\mathbf{v}}_0 = \sum_{l=0}^{N-1} \gamma_l \mathbf{u}_l$, respectively. Then, we can rewrite (5.39) as

$$\begin{aligned}
\bar{\mathbf{y}}_k &= \tilde{\mathbf{y}} - \sum_{i=0}^{P-1} \lambda_i^{2^k} \mathbf{u}_i \mathbf{p}_i^T \left(\sum_{l=0}^{N-1} \beta_l \mathbf{u}_l \right) - \sum_{j=P}^{N-1} \lambda_j^{2^k} \mathbf{u}_j \mathbf{p}_j^T \left(\sum_{l=0}^{N-1} \beta_l \mathbf{u}_l \right) + \sum_{j=0}^{N-1} \phi_{j,k} \mathbf{u}_j \mathbf{p}_j^T \left(\sum_{l=0}^{N-1} \gamma_l \mathbf{u}_l \right) \\
&= \tilde{\mathbf{y}} - \sum_{i=0}^{P-1} \lambda_i^{2^k} \beta_i \mathbf{u}_i - \sum_{j=P}^{N-1} \lambda_j^{2^k} \beta_j \mathbf{u}_j + \sum_{j=0}^{N-1} \phi_{j,k} \gamma_j \mathbf{u}_j \\
&= \tilde{\mathbf{y}} + \bar{\mathbf{y}}_{k,i} + \bar{\mathbf{y}}_{k,n},
\end{aligned} \tag{5.40}$$

where $\bar{\mathbf{y}}_{k,i} = -\sum_{i=0}^{P-1} \lambda_i^{2^k} \beta_i \mathbf{u}_i - \sum_{j=P}^{N-1} \lambda_j^{2^k} \beta_j \mathbf{u}_j$, and $\bar{\mathbf{y}}_{k,n} = \sum_{j=0}^{N-1} \phi_{j,k} \gamma_j \mathbf{u}_j$. Note that $\bar{\mathbf{y}}_{k,i}$ is the residual interference term and $\bar{\mathbf{y}}_{k,n}$ is the noise term. The average SINR for the k th iteration, denoted as SINR_k^a , can be expressed as

$$\text{SINR}_k^a = \frac{E\{\tilde{\mathbf{y}}^H \tilde{\mathbf{y}}\}}{E\{\bar{\mathbf{y}}_{k,i}^H \bar{\mathbf{y}}_{k,i}\} + E\{\bar{\mathbf{y}}_{k,n}^H \bar{\mathbf{y}}_{k,n}\}}. \tag{5.41}$$

Assume that cross-correlations of \mathbf{u}_l 's are small, and can be ignored. Then, the powers of the desired signal, interference, noise can be approximated as

$$\begin{aligned}
E\{\tilde{\mathbf{y}}^H \tilde{\mathbf{y}}\} &= \sum_{l=0}^{N-1} \sum_{l'=0}^{N-1} E\{\beta_l^* \beta_{l'}\} \mathbf{u}_l^H \mathbf{u}_{l'} \\
&\approx \sum_{l=0}^{N-1} E\{|\beta_l|^2\},
\end{aligned} \tag{5.42}$$

$$\begin{aligned}
E\{\bar{\mathbf{y}}_{k,i}^H \bar{\mathbf{y}}_{k,i}\} &= E \left\{ \left(\sum_{i=0}^{P-1} \lambda_i^{2^k} \beta_i \mathbf{u}_i + \sum_{j=P}^{N-1} \lambda_j^{2^k} \beta_j \mathbf{u}_j \right)^H \left(\sum_{i'=0}^{P-1} \lambda_{i'}^{2^k} \beta_{i'} \mathbf{u}_{i'} + \sum_{j'=P}^{N-1} \lambda_{j'}^{2^k} \beta_{j'} \mathbf{u}_{j'} \right) \right\} \\
&\approx \sum_{i=0}^{P-1} |\lambda_i|^{2^{k+1}} E\{|\beta_i|^2\} + \sum_{l=P}^{N-1} |\lambda_l|^{2^{k+1}} E\{|\beta_l|^2\},
\end{aligned} \tag{5.43}$$

and

$$\begin{aligned}
E\{\bar{\mathbf{y}}_{k,n}^H \bar{\mathbf{y}}_{k,n}\} &= E\left\{\left(\sum_{j=0}^{N-1} \phi_{j,k} \gamma_j \mathbf{u}_j\right)^H \left(\sum_{j'=0}^{N-1} \phi_{j',k} \gamma_{j'} \mathbf{u}_{j'}\right)\right\} \\
&\approx \sum_{j=0}^{N-1} |\phi_{j,k}|^2 E\{|\gamma_j|^2\}.
\end{aligned} \tag{5.44}$$

Finally, the average SINR can be approximated as

$$\begin{aligned}
\text{SINR}_k^a &\approx \frac{\sum_{l=0}^{N-1} E\{|\beta_l|^2\}}{\sum_{i=0}^{P-1} |\lambda_i|^{2k+1} E\{|\beta_i|^2\} + \sum_{l=P}^{N-1} |\lambda_l|^{2k+1} E\{|\beta_l|^2\} + \sum_{j=0}^{N-1} |\phi_{j,k}|^2 E\{|\gamma_j|^2\}} \\
&= \frac{\text{SP}}{\text{IP}_1 + \text{IP}_2 + \text{NP}},
\end{aligned} \tag{5.45}$$

where $\text{SP} = \sum_{l=0}^{N-1} E\{|\beta_l|^2\}$, $\text{IP}_1 = \sum_{i=0}^{P-1} |\lambda_i|^{2k+1} E\{|\beta_i|^2\}$, $\text{IP}_2 = \sum_{l=P}^{N-1} |\lambda_l|^{2k+1} E\{|\beta_l|^2\}$, and $\text{NP} = \sum_{j=0}^{N-1} |\phi_{j,k}|^2 E\{|\gamma_j|^2\}$. Now, we can examine the three terms in the denominator of (5.45). The first term IP_1 , involving eigenvalues with amplitudes greater than one, is monotonically increased, and the second term, involving eigenvalues with amplitudes less than one, is monotonically decreased as k is increased. Recall that only few eigenvalues' amplitudes will be greater than one (i.e., P is small), and their amplitudes often do not deviate from one too much. Also, from the definition of NP , it can be shown that the third term tends to be increased when k is increased and its variation is not large (see the definition of $\phi_{j,k}$). Then, it is simple to see that in the first several iterations, the amount of decreasing in IP_2 will be larger than that of increasing in IP_1 . We then conclude that for typical divergence cases, SINR will be increased and then decreased as the iteration is proceeded. Thus, if we can stop the iteration before SINR is degraded, we can still have the performance gain even though the iteration diverges eventually. Due to the fast convergence property of Newton's method, the number of iterations can be as small as two or three for convergent cases. For divergent cases, SINR is still increasing in the first two or three iterations. Note that the magnitudes of eigenvalues are related to those of CFOs. The proposed method with CFO pre-compensation can reduce the magnitudes of CFOs. Thus, it can improve the performance of the proposed method. As an example, we let $N = 64$, $Q = 4$, $S = 2$, and the normalized CFO (for each user) be randomly sampled from the interval

$[-\rho, \rho]$, where $\rho \leq 0.5$. After exhaustive simulations, we found that without PC, the largest ρ rendering the amplitudes of all eigenvalues smaller than one is 0.36, and with PC, it can be as large as 0.5.

We now develop the second method to analyze the convergence behavior of the proposed algorithm. Recalling (5.5) and (5.14), we can rewrite the receive signal after CFO compensation as

$$\begin{aligned}\mathbf{W}_k \tilde{\mathbf{r}} &= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} (\mathbf{W}_0 \tilde{\mathbf{M}})^{m+1} \tilde{\mathbf{y}} + \sum_{m=0}^{2^k-1} \bar{c}_{k,m} (\mathbf{W}_0 \tilde{\mathbf{M}})^m \mathbf{W}_0 \tilde{\mathbf{v}} \\ &= \mathbf{T}_k \tilde{\mathbf{y}} + \mathbf{W}_k \tilde{\mathbf{v}},\end{aligned}\quad (5.46)$$

where $\mathbf{T}_k = \sum_{m=0}^{2^k-1} \bar{c}_{k,m} (\mathbf{W}_0 \tilde{\mathbf{M}})^{m+1}$ is the CFO-compensated ICI matrix. Define $\mathbf{B} = \mathcal{B}_b(\mathbf{Z})$ as a banded matrix with upper bandwidth b and lower bandwidth b , i.e., $\mathbf{B}(i, j) = 0$ whenever $|j - i| > b$, and $\mathbf{B}(i, j) = \mathbf{Z}(i, j)$ otherwise. In addition, $\bar{\mathbf{B}} = \bar{\mathcal{B}}_b(\mathbf{Z})$ is the complement of $\mathcal{B}_b(\mathbf{Z})$, i.e., $\bar{\mathbf{B}}(i, j) = \mathbf{Z}(i, j)$ whenever $|j - i| > b$, and $\bar{\mathbf{B}}(i, j) = 0$ otherwise. Then, we can rewrite (5.46) as

$$\mathbf{W}_k \tilde{\mathbf{r}} = \mathcal{B}_0(\mathbf{T}_k) \tilde{\mathbf{y}} + \bar{\mathcal{B}}_0(\mathbf{T}_k) \tilde{\mathbf{y}} + \mathbf{W}_k \tilde{\mathbf{v}}. \quad (5.47)$$

The first term, $\mathcal{B}_0(\mathbf{T}_k) \tilde{\mathbf{y}}$, is the desired signal, the second term, $\bar{\mathcal{B}}_0(\mathbf{T}_k) \tilde{\mathbf{y}}$, is the interference, and the last term, $\mathbf{W}_k \tilde{\mathbf{v}}$, is the noise. Let $\tilde{\mathbf{y}} = [\tilde{y}_0, \tilde{y}_1, \dots, \tilde{y}_{N-1}]^T$, and $\tilde{\mathbf{v}} = [\tilde{v}_0, \tilde{v}_1, \dots, \tilde{v}_{N-1}]^T$. Then we can define the average SINR for the proposed method with k iterations as follows:

$$\begin{aligned}\text{SINR}_k^a &= \frac{E\{\|\mathcal{B}_0(\mathbf{T}_k) \tilde{\mathbf{y}}\|_2^2\}}{E\{\|\bar{\mathcal{B}}_0(\mathbf{T}_k) \tilde{\mathbf{y}}\|_2^2\} + E\{\|\mathbf{W}_k \tilde{\mathbf{v}}\|_2^2\}} \\ &= \frac{\sigma_{\tilde{y}}^2 \|\mathcal{B}_0(\mathbf{T}_k)\|_F^2}{\sigma_{\tilde{y}}^2 \|\bar{\mathcal{B}}_0(\mathbf{T}_k)\|_F^2 + \sigma_{\tilde{v}}^2 \|\mathbf{W}_k\|_F^2} \\ &= \frac{\|\mathcal{B}_0(\mathbf{T}_k)\|_F^2}{\|\bar{\mathcal{B}}_0(\mathbf{T}_k)\|_F^2 + \psi \|\mathbf{W}_k\|_F^2},\end{aligned}\quad (5.48)$$

where $\sigma_{\tilde{y}}^2 = E\{|\tilde{y}_i|^2\} (0 \leq i \leq N-1)$, $\sigma_{\tilde{v}}^2 = E\{|\tilde{v}_i|^2\} (0 \leq i \leq N-1)$, and $\psi = \sigma_{\tilde{v}}^2 / \sigma_{\tilde{y}}^2 = \|\tilde{\mathbf{M}}\|_F^2 / (N \times \text{SNR}) = 1/\text{SNR}$. For comparison, the average SINR of the receive signal without

CFO-compensation is also calculated as

$$\begin{aligned}
\text{SINR}^a &= \frac{E\{\|\mathcal{B}_0(\widetilde{\mathbf{M}})\widetilde{\mathbf{y}}\|_2^2\}}{E\{\|\overline{\mathcal{B}}_0(\widetilde{\mathbf{M}})\widetilde{\mathbf{y}}\|_2^2\} + E\{\|\widetilde{\mathbf{v}}\|_2^2\}} \\
&= \frac{\sigma_{\widetilde{\mathbf{y}}}^2\|\mathcal{B}_0(\widetilde{\mathbf{M}})\|_F^2}{\sigma_{\widetilde{\mathbf{y}}}^2\|\overline{\mathcal{B}}_0(\widetilde{\mathbf{M}})\|_F^2 + N\sigma_{\widetilde{\mathbf{v}}}^2} \\
&= \frac{\|\mathcal{B}_0(\widetilde{\mathbf{M}})\|_F^2}{\|\overline{\mathcal{B}}_0(\widetilde{\mathbf{M}})\|_F^2 + N\psi}.
\end{aligned} \tag{5.49}$$

To obtain the SINR_k^a in (5.48), we must calculate each element in \mathbf{T}_k and \mathbf{W}_k . Since $\widetilde{\mathbf{M}} = \sum_{q=1}^Q \widetilde{\mathbf{E}}^q \mathbf{S}^q$, the CFO-compensated ICI matrix \mathbf{T}_k can be expanded as

$$\begin{aligned}
\mathbf{T}_k &= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left(\sum_{q=1}^Q \mathbf{W}_0 \widetilde{\mathbf{E}}^q \mathbf{S}^q \right)^{m+1} \\
&= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{q_{m,0}=1}^Q \sum_{q_{m,1}=1}^Q \cdots \sum_{q_{m,m}=1}^Q \prod_{f=0}^m \mathbf{W}_0 \widetilde{\mathbf{E}}^{q_{m,f}} \mathbf{S}^{q_{m,f}} \right\} \\
&= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{q_{m,0}=1}^Q \sum_{q_{m,1}=1}^Q \cdots \sum_{q_{m,m}=1}^Q \mathbf{A}_m \right\},
\end{aligned} \tag{5.50}$$

where $\mathbf{A}_m = \prod_{f=0}^m \mathbf{W}_0 \widetilde{\mathbf{E}}^{q_{m,f}} \mathbf{S}^{q_{m,f}}$, and $\mathbf{A}_{-1} = \mathbf{I}_N$. In the same way, we can also expand \mathbf{W}_k as

$$\begin{aligned}
\mathbf{W}_k &= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left(\sum_{q=1}^Q \mathbf{W}_0 \widetilde{\mathbf{E}}^q \mathbf{S}^q \right)^m \mathbf{W}_0 \\
&= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{q_{m,0}=1}^Q \sum_{q_{m,1}=1}^Q \cdots \sum_{q_{m,m-1}=1}^Q \mathbf{A}_{m-1} \mathbf{W}_0 \right\} \\
&= \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{q_{m,0}=1}^Q \sum_{q_{m,1}=1}^Q \cdots \sum_{q_{m,m-1}=1}^Q \mathbf{B}_m \right\},
\end{aligned} \tag{5.51}$$

where $\mathbf{B}_m = \mathbf{A}_{m-1} \mathbf{W}_0$. In what follows, we calculate each element of $\mathbf{A}_m(i_m, j_m)$. To make the expression simpler, we denote $w_{opt,i}$ as w_i . First, we compute $\mathbf{A}_0(i_0, j_0)$ which can be

expressed as

$$\begin{aligned}\mathbf{A}_0(i_0, j_0) &= [\mathbf{W}_0 \tilde{\mathbf{E}}^{q_0,0} \mathbf{S}^{q_0,0}]_{i_0, j_0} \\ &= w_{\langle i_0, Q \rangle} \tilde{e}_{\langle i_0 - j_0, N \rangle}^{q_0,0} \delta(q_0,0 - 1 - \langle j_0, Q \rangle).\end{aligned}\quad (5.52)$$

Next, we express $\mathbf{A}_1(i_1, j_1)$ as

$$\begin{aligned}\mathbf{A}_1(i_1, j_1) &= [(\mathbf{W}_0 \tilde{\mathbf{E}}^{q_1,0} \mathbf{S}^{q_1,0})(\mathbf{W}_0 \tilde{\mathbf{E}}^{q_1,1} \mathbf{S}^{q_1,1})]_{i_1, j_1} \\ &= \sum_{\substack{k_0=0 \\ k_0 \in \Upsilon_{q_1,0}}}^{N-1} w_{\langle i_1, Q \rangle} w_{\langle k_0, Q \rangle} \tilde{e}_{\langle i_1 - k_0, N \rangle}^{q_1,0} \tilde{e}_{\langle k_0 - j_1, N \rangle}^{q_1,1} \delta(q_1,1 - 1 - \langle j_1, Q \rangle).\end{aligned}\quad (5.53)$$

Furthermore, $\mathbf{A}_2(i_2, j_2)$ can be presented by

$$\begin{aligned}\mathbf{A}_2(i_2, j_2) &= [(\mathbf{W}_0 \tilde{\mathbf{E}}^{q_2,0} \mathbf{S}^{q_2,0})(\mathbf{W}_0 \tilde{\mathbf{E}}^{q_2,1} \mathbf{S}^{q_2,1})(\mathbf{W}_0 \tilde{\mathbf{E}}^{q_2,2} \mathbf{S}^{q_2,2})]_{i_2, j_2} \\ &= \sum_{\substack{k_1=0 \\ k_1 \in \Upsilon_{q_2,1}}}^{N-1} \sum_{\substack{k_0=0 \\ k_0 \in \Upsilon_{q_2,0}}}^{N-1} w_{\langle i_2, Q \rangle} w_{\langle k_0, Q \rangle} w_{\langle k_1, Q \rangle} \tilde{e}_{\langle i_2 - k_0, N \rangle}^{q_2,0} \tilde{e}_{\langle k_0 - k_1, N \rangle}^{q_2,1} \tilde{e}_{\langle k_1 - j_2, N \rangle}^{q_2,2} \\ &\quad \delta(q_2,1 - 1 - \langle k_1, Q \rangle) \delta(q_2,2 - 1 - \langle j_2, Q \rangle).\end{aligned}\quad (5.54)$$

Finally, we can formulate $\mathbf{A}_m(i_m, j_m)$ for $m \geq 1$ as

$$\begin{aligned}\mathbf{A}_m(i_m, j_m) &= [(\mathbf{W}_0 \tilde{\mathbf{E}}^{q_m,0} \mathbf{S}^{q_m,0})(\mathbf{W}_0 \tilde{\mathbf{E}}^{q_m,1} \mathbf{S}^{q_m,1}) \dots (\mathbf{W}_0 \tilde{\mathbf{E}}^{q_m,m} \mathbf{S}^{q_m,m})]_{i_m, j_m} \\ &= \sum_{\substack{k_{m-1}=0 \\ k_{m-1} \in \Upsilon_{q_m, m-1}}}^{N-1} \sum_{\substack{k_{m-2}=0 \\ k_{m-2} \in \Upsilon_{q_m, m-2}}}^{N-1} \dots \sum_{\substack{k_0=0 \\ k_0 \in \Upsilon_{q_m, 0}}}^{N-1} w_{\langle i_m, Q \rangle} w_{\langle k_0, Q \rangle} w_{\langle k_1, Q \rangle} \dots w_{\langle k_{m-1}, Q \rangle} \\ &\quad \tilde{e}_{\langle i_m - k_0, N \rangle}^{q_m,0} \tilde{e}_{\langle k_0 - k_1, N \rangle}^{q_m,1} \tilde{e}_{\langle k_1 - k_2, N \rangle}^{q_m,2} \dots \tilde{e}_{\langle k_{m-1} - j_m, N \rangle}^{q_m,m} \delta(q_m,1 - 1 - \langle k_1, Q \rangle) \\ &\quad \delta(q_m,2 - 1 - \langle k_2, Q \rangle) \dots \delta(q_m, m-1 - 1 - \langle k_{m-1}, Q \rangle) \\ &\quad \delta(q_m, m - 1 - \langle j_m, Q \rangle).\end{aligned}\quad (5.55)$$

Since $\tilde{\mathbf{e}}^q = (1/\sqrt{N}) \mathbf{G} \mathbf{e}^q$, we can further express \tilde{e}_k^q as

$$\begin{aligned}\tilde{e}_k^q &= \frac{1}{N} \sum_{n=0}^{N-1} e^{j \frac{2\pi n \epsilon_q}{N}} e^{-j \frac{2\pi n k}{N}} \\ &= \frac{\sin[\pi(k - \epsilon_q)]}{N \sin\left[\frac{\pi(k - \epsilon_q)}{N}\right]} e^{-j\pi(k - \epsilon_q)\left(\frac{N-1}{N}\right)}.\end{aligned}\quad (5.56)$$

By (5.56), we can expand $\tilde{m}_{i,j}$ as

$$\begin{aligned}\tilde{m}_{i,j} &= \tilde{e}_{\langle i-j, N \rangle}^{\langle j, Q \rangle+1} \\ &= \frac{\sin\{\pi[\langle i-j, N \rangle - \epsilon_{\langle j, Q \rangle+1}]\}}{N \sin\left\{\frac{\pi[\langle i-j, N \rangle - \epsilon_{\langle j, Q \rangle+1}]}{N}\right\}} e^{-j\pi[\langle i-j, N \rangle - \epsilon_{\langle j, Q \rangle+1}]\left(\frac{N-1}{N}\right)}.\end{aligned}\quad (5.57)$$

With $\tilde{m}_{i,j}$, we can further formulate $w_{\langle k, Q \rangle}$ as

$$\begin{aligned}w_{\langle k, Q \rangle} &\approx \frac{\tilde{m}_{\langle k, Q \rangle, \langle k, Q \rangle}^*}{\sum_{j=\langle\langle k, Q \rangle - S, \langle k, Q \rangle + S, N \rangle} |\tilde{m}_{\langle k, Q \rangle, j}|^2} \\ &= \frac{(\tilde{e}_0^{\langle k, Q \rangle+1})^*}{\sum_{j=\langle\langle k, Q \rangle - S, \langle k, Q \rangle + S, N \rangle} |\tilde{e}_{\langle\langle k, Q \rangle - j, N \rangle}^{\langle j, Q \rangle+1}|^2} \\ &= \frac{N \sin[\pi\epsilon_{\langle k, Q \rangle+1}] e^{-j\pi\epsilon_{\langle k, Q \rangle+1}\left(\frac{N-1}{N}\right)}}{\sin\left[\frac{\pi\epsilon_{\langle k, Q \rangle+1}}{N}\right] \sum_{j=\langle\langle k, Q \rangle - S, \langle k, Q \rangle + S, N \rangle} \frac{\sin^2\{\pi[\langle\langle k, Q \rangle - j, N \rangle - \epsilon_{\langle j, Q \rangle+1}]\}}{\sin^2\left\{\frac{\pi[\langle\langle k, Q \rangle - j, N \rangle - \epsilon_{\langle j, Q \rangle+1}]}{N}\right\}}}.\end{aligned}\quad (5.58)$$

By (5.56) and (5.58), we can completely express $\mathbf{A}_m(i_m, j_m)$ in (5.55). From (5.55) and (5.58), we can formulate $\mathbf{B}_m(i_m, j_m)$ as

$$\begin{aligned}\mathbf{B}_m(i_m, j_m) &= \mathbf{A}_{m-1}(i_m, j_m) w_{\langle j_m, Q \rangle} \\ &= \frac{\mathbf{A}_{m-1}(i_m, j_m) N \sin\{\pi\epsilon_{\langle j_m, Q \rangle+1}\} e^{-j\pi\epsilon_{\langle j_m, Q \rangle+1}\left(\frac{N-1}{N}\right)}}{\sin\left[\frac{\pi\epsilon_{\langle j_m, Q \rangle+1}}{N}\right] \sum_{j=\langle\langle j_m, Q \rangle - S, \langle j_m, Q \rangle + S, N \rangle} \frac{\sin^2\{\pi[\langle\langle j_m, Q \rangle - j, N \rangle - \epsilon_{\langle j_m, Q \rangle+1}]\}}{\sin^2\left\{\frac{\pi[\langle\langle j_m, Q \rangle - j, N \rangle - \epsilon_{\langle j_m, Q \rangle+1}]}{N}\right\}}}.\end{aligned}\quad (5.59)$$

Now we can calculate each term in (5.48) by (5.55) and (5.59). The term $\|\mathcal{B}_0(\mathbf{T}_k)\|_F^2$ related to the desired signal can be formulated as

$$\|\mathcal{B}_0(\mathbf{T}_k)\|_F^2 = \sum_{i_m=0}^{N-1} \left| \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{q_{m,0}=1}^Q \sum_{q_{m,1}=1}^Q \cdots \sum_{q_{m,m}=1}^Q \mathbf{A}_m(i_m, i_m) \right\} \right|^2, \quad (5.60)$$

the interference-related $\|\bar{\mathcal{B}}_0(\mathbf{T}_k)\|_F^2$ is expressed as

$$\|\bar{\mathcal{B}}_0(\mathbf{T}_k)\|_F^2 = \sum_{j_m=0}^{N-1} \sum_{\substack{i_m=0 \\ i_m \neq j_m}}^{N-1} \left| \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{q_{m,0}=1}^Q \cdots \sum_{q_{m,m}=1}^Q \mathbf{A}_m(i_m, j_m) \right\} \right|^2, \quad (5.61)$$

and $\|\mathbf{W}_k\|_F^2$ can be calculated by

$$\|\mathbf{W}_k\|_F^2 = \sum_{j_m=0}^{N-1} \sum_{i_m=0}^{N-1} \left| \sum_{m=0}^{2^k-1} \bar{c}_{k,m} \left\{ \sum_{q_{m,0}=1}^Q \cdots \sum_{q_{m,m-1}=1}^Q \mathbf{B}_m(i_m, j_m) \right\} \right|^2. \quad (5.62)$$

Thus, the average SINR of the proposed method with k iterations can be explicitly calculated by (5.55), (5.59), (5.60), (5.61), and (5.62). As for (5.49), we can further express the result by $\tilde{m}_{i,j}$ in (5.57) as

$$\text{SINR}^a = \frac{\sum_{j=0}^{N-1} \frac{\sin^2\{\pi\epsilon\langle j, Q \rangle + 1\}}{N^2 \sin^2\left\{\frac{\pi\epsilon\langle j, Q \rangle + 1}{N}\right\}}}{\sum_{j=0}^{N-1} \sum_{\substack{i=0 \\ i \neq j}}^{N-1} \frac{\sin^2\{\pi[\langle i-j, N \rangle - \epsilon\langle j, Q \rangle + 1]\}}{N^2 \sin^2\left\{\frac{\pi[\langle i-j, N \rangle - \epsilon\langle j, Q \rangle + 1]}{N}\right\}} + N\psi}. \quad (5.63)$$

We also evaluate the SINR for each subcarrier. The SINR for the proposed algorithm with k iterations in the i th subcarrier, denoted as $\text{SINR}_{k,i}^s$, is shown to be

$$\begin{aligned} \text{SINR}_{k,i}^s &= \frac{E\{|t_{i,i}^k \tilde{y}_i|^2\}}{E\{|\sum_{\substack{j=0 \\ j \neq i}}^{N-1} t_{i,j}^k \tilde{y}_j|^2\} + E\{|\sum_{j=0}^{N-1} w_{i,j}^k \tilde{v}_j|^2\}} \\ &= \frac{|t_{i,i}^k|^2}{\sum_{\substack{j=0 \\ j \neq i}}^{N-1} |t_{i,j}^k|^2 + \frac{\sigma_v^2}{\sigma_y^2} \sum_{j=0}^{N-1} |w_{i,j}^k|^2} \\ &= \frac{|t_{i,i}^k|^2}{\sum_{\substack{j=0 \\ j \neq i}}^{N-1} |t_{i,j}^k|^2 + \psi \sum_{j=0}^{N-1} |w_{i,j}^k|^2}, \end{aligned} \quad (5.64)$$

where $t_{i,j}^k = \mathbf{T}_k(i, j)$ and $w_{i,j}^k = \mathbf{W}_k(i, j)$. Here, $\mathbf{T}_k(i, j)$ and $\mathbf{W}_k(i, j)$ are defined in (5.50) and (5.51), respectively. The i th subcarrier SINR of the received signal without CFO-compensation, denoted as SINR_i^s , can be described as

$$\begin{aligned} \text{SINR}_i^s &= \frac{E\{|\tilde{m}_i \tilde{y}_i|^2\}}{E\{|\sum_{\substack{j=0 \\ j \neq i}}^{N-1} \tilde{m}_i \tilde{y}_j|^2\} + E\{|\tilde{v}_i|^2\}} \\ &= \frac{\frac{\sin^2\{\pi\epsilon\langle i, Q \rangle + 1\}}{N^2 \sin^2\left\{\frac{\pi\epsilon\langle i, Q \rangle + 1}{N}\right\}}}{\sum_{\substack{j=0 \\ j \neq i}}^{N-1} \frac{\sin^2\{\pi[\langle i-j, N \rangle - \epsilon\langle j, Q \rangle + 1]\}}{N^2 \sin^2\left\{\frac{\pi[\langle i-j, N \rangle - \epsilon\langle j, Q \rangle + 1]}{N}\right\}} + \psi}. \end{aligned} \quad (5.65)$$

§ 5.3.5 Simulations

In this subsection, we report simulation results to demonstrate the effectiveness of the proposed method. We consider an interleaved-OFDMA uplink system with $N = 64$, $Q = 4$, and the CP length $N_g = 16$. The adopted modulation scheme is 16-QAM or 64-QAM. The length of the channel response, L , is set to 15 for all users, and the power delay profile of user q is described with an exponential function, i.e., $\sigma_{q,l}^2 = e^{-\alpha_q l} / \sum_{i=0}^{L-1} e^{-\alpha_q i}$, where l is the tap index, and α_q is a parameter of the function. For later simulations, we let $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\} = \{0.1, 0.2, 0.3, 0.4\}$ for each user. Each channel tap is generated according to Rayleigh distribution. Also, we have found that the performance of the proposed algorithm with $S = 2$ is similar to that with $S = 31$. Thus, in the following experiments, we will only consider the setting of $S = 2$.

First, we evaluate the validity of our output SINR analytic results. Two cases are considered. Case 1 corresponds to the case that the amplitudes of all eigenvalues of \mathbf{R}_0 are smaller than one, and in case 2 some eigenvalues' amplitudes are larger than one. In case 1, CFOs = $\{0.1, -0.2, -0.05, 0.2\}$ and SNR = 30 dB, while in case 2, CFOs = $\{0.49, 0.49, 0.15, 0.4\}$ and SNR = 15 dB. Note that most of the CFO values in case 2 are quite large and positive. Figure 5.1 shows the average SINRs calculated for the proposed method with the approaches in Subsection 5.3.4. From this figure, we find that the simulated output SINRs are identical to the results of the exact analysis, which verifies the correctness of the exact analysis. We also see that in both cases, the average SINR obtained by the approximated analysis is close to that by the exact analysis, especially when the iteration number gets larger. Just as mentioned in Subsection 5.3.4, even in the divergent case, the SINR increases for first two iterations. In case 1, the SINR is saturated at the second iteration.

Next, we investigate the effect of input SNR on output SINR. We assume that CFOs are set to $\{0.1, -0.2, -0.05, 0.2\}$, and use the result of exact analysis. Figure 5.2 shows the average SINR for the proposed method. In this figure, we also plot the theoretical average SINR without CFO compensation using (5.63). From this figure, we can see that the average output SINR is

improved when the number of iterations increases. For the proposed method with two iterations, the theoretical average output SINR is almost the same as the input SNR. This result indicates that the proposed method almost cancels the CFO-induced ICI thoroughly. The theoretical subcarrier SINR analysis for the proposed method is also shown in Fig. 5.3. Here, the input SNR is set to 30 dB. From this figure, we can see that different subcarriers have different SINRs and the SINRs are improved when the number of iterations is increased. When the number of iterations is two, all subcarriers almost have the same output SINR which is close to the input SNR (30 dB). We have also tried other scenarios and obtained the similar result. We then conclude that when amplitudes of CFOs are moderate, a suitable choice for the iteration number is two.

Now, we present simulation results to evaluate the BER performance of the proposed method. We consider a 16-QAM modulation scheme and set CFOs to $\{0.1, -0.2, -0.05, 0.2\}$. Figure 5.4 shows the simulation result. From the figure, it is obvious that the conventional method and the CLJL method both have a serious error floor phenomenon. This is because when the q th user's CFO is compensated, other users' CFOs may be enlarged, magnifying MUI. We can also see that the CLJL method performs slightly better than the conventional method. Since the CLJL-PIC method further processes the MUI, it improves the performance of the CLJL method. The CLJL-PIC method with a 2-stage PIC can perform similarly as that with a 3-stage PIC. The performance bound, in which no CFOs are added, is also shown in the figure. Even with a 3-stage PIC, the CLJL-PIC method still cannot approach the performance bound. In higher SNR regions, the performance loss is larger. With only two iterations, the proposed method performs as well as the direct ZF method. Also, its performance closely approaches the performance bound.

We also consider a 64-QAM modulation scheme for the same CFO setting in Fig. 5.4 (CFOs = $\{0.1, -0.2, -0.05, 0.2\}$). Figure 5.5 shows the BER performance comparison. We can see that the performance of the CLJL-PIC method degrades. This is because in high QAM-size modulation, the performance of an OFDMA system is more sensitive to the residual MUI. The

performance of the proposed method can still approach that of the direct ZF method.

We further consider a worse scenario, where the CFO of some user is large. Specifically, we set CFOs to $\{0.1, -0.4, -0.05, 0.2\}$ and use a 16-QAM modulation scheme. Figure 5.6 shows the BER performance comparison. In addition to the conventional method and the CLJL method, the CLJL-PIC method (even with a 3-stage PIC) performs poorly. In this case, the PIC method fails to cancel MUI. This may be due to an error propagation effect inherent in the PIC scheme. The performance of the proposed method is slightly affected. This performance loss results from the insufficient iteration number. We have shown the result for the proposed method with three iterations in the figure. It can be seen that the performance can be further enhanced at the expense of the increased computational complexity.

To clearly see the impact of the CFO magnitude, we consider a scenario that the fourth user's CFO is increased from 0 to 0.5. If the CFO of the fourth user is increased, the MUI from the fourth user will be increased. The CFOs of other three users are set to 0.1, -0.2 , and -0.05 , respectively. The adopted modulation scheme is 16-QAM and the simulated SNRs are 25 dB and 35 dB. We simulate the average BER of the first three users. Figure 5.7 shows the simulation result. From this figure, we can see that the CLJL-PIC method (with $p = 3$) is sensitive to the CFO variation. For SNR = 35 dB, we find that the BER for the CLJL-PIC method begins to increase when the fourth user's CFO is 0.1. When the CFO is increased further, its performance is degraded rapidly. For the proposed method, only little performance loss is observed. If the iteration number is three, the proposed method almost does not have performance degradation compared to the direct ZF method. As the results observed above, the proposed method can have the same performance as the direct ZF method.

Since OFDMA is a multiuser system, the near-far phenomenon may occur. In such a case, some users may have stronger receive power than others. To realize the impact of the near-far effect, we report simulations with a scenario that the powers of the first three users are equal and fixed and that of the fourth user is varied. The power ratio of the fourth user to anyone of the first three users is defined as the near-far power ratio κ ranging from -15 dB to 15

dB. Let CFOs be $\{0.1, -0.2, -0.05, 0.2\}$ and modulation be 16-QAM. Similar to the previous simulation setting, we calculate the average BER of the first three users. Figure 5.8 shows the result. From the figure, we find that both the CJLJ-PIC and proposed algorithms are affected by the near-far problem. However, the CJLJ-PIC is more sensitive in the near-far environment.

We consider an extreme case, in which most CFOs are very large and positive. Here, we set CFOs to $\{0.49, 0.49, 0.1, 0.4\}$ and use a 16-QAM modulation scheme. In this case, ϵ_{opt} is calculated as 0.37, which is large. We compare the performance of all methods mentioned above. For the proposed method, we also try the PC technique. Figure 5.9 shows the simulation results. In this case, the conventional method, the CLJL method, and the CLJL-PIC method all have bad performance. Note that the conventional method can even have better performance than the CLJL-PIC method. The proposed method without PC does not perform well either. Only does the proposed method with PC perform well. Its performance is almost identical to that of the direct ZF method. The result shows the effectiveness of the proposed PC method. We also consider another case that CFOs are not all positive. Specifically, CFOs are set to $\{-0.1, 0.3, 0.4, -0.2\}$. Figure 5.10 shows the BER result. We again see that the proposed method with PC has similar performance with the direct ZF method. These results show that the PC method can always be applied to improve the performance of the proposed method.

Finally, we present simulation results in an interleaved-OFDMA uplink system with the large number of subcarriers and more users. Specifically, $N = 2048$, $Q = 16$, and $N_g = 128$. The modulation scheme is 16-QAM. The channel length is set to 127 for all users and let $\{\alpha_1, \alpha_2, \dots, \alpha_Q\} = \{0, 0.2, 0.4, \dots, 3\}$. CFOs for all users are set to $\{0.1, -0.2, -0.05, 0.2, -0.3, 0, -0.1, 0.4, -0.3, 0.05, 0, -0.1, 0.05, -0.1, 0.3, 0.15\}$. The performances of five methods, namely, the conventional, CLJL, direct ZF, banded ZF, and proposed methods, are compared in our simulations. The banded ZF method indicates that it modifies the ICI matrix into a banded matrix with bandwidth B . Figure 5.11 shows the simulation results. From this figure, we find that the conventional and CLJL methods both have a serious error floor phenomenon. The performance of the proposed method with three iterations can approach that of the direct ZF

method. The complexity of the banded ZF method depends greatly on its matrix bandwidth. For a fair comparison, we let B be 16 for the banded ZF method. In this case, the complexity of the banded ZF method and the proposed method ($k = 3$) are roughly equal. Figure 5.11 shows that the proposed method performs much better than the banded ZF method ($B = 16$).

From the above simulations, we can see that the proposed method is more robust to the large modulation QAM-size and CFOs compared to the CLJL-PIC method. In what follows, we will compare the computational complexity of the direct ZF method, the CLJL-PIC method, and the proposed method in an OFDMA system shown above ($N = 64, Q = 4$). Substituting the required parameters into Table 5.1, we then derive the computational complexity of each method, and show the result in Table 5.2. The number of iterations for the proposed method is set to two here. Note that the number inside the parenthesis of the forth row indicate the ratio of the required number of mathematic operations for the proposed algorithm divided by that of other methods. From the table, we can see that the real multiplications/additions for the proposed method is 0.286/0.306 times those for the CLJL-PIC method (with a 2-stage PIC), and is 0.037/0.041 times those for the direct ZF method. The required number of divisions is small (for three methods) compared with that of multiplications/additions. For example, the proposed method only needs 8 real divisions.

Table 5.3 shows the computational complexities of the direct ZF, banded ZF, and proposed algorithms in an OFDMA uplink system with 2048 subcarriers and 16 users. In this table, the two numbers inside each set of parentheses (in the forth row) are the ratios of the number of operations (indicated by each column) required for the proposed method to those of the direct ZF and banded ZF methods, respectively. From this table, we can see that the real multiplications/additions/divisions required for the proposed method are 0.000271/0.000282/0.000008 times those for the direct ZF method. It is apparent that the proposed method requires a much lower complexity. Although the banded ZF method can have low complexity, its performance is not satisfactory. At a similar complexity, the proposed method outperforms the banded ZF method.

From the results shown above, we can have following conclusions. Compared to the CLJL-PIC method, the proposed method not only performs better, but also requires lower computational complexity. Compared to the direct ZF method, the proposed method has similar performance, but requires much lower computational complexity.



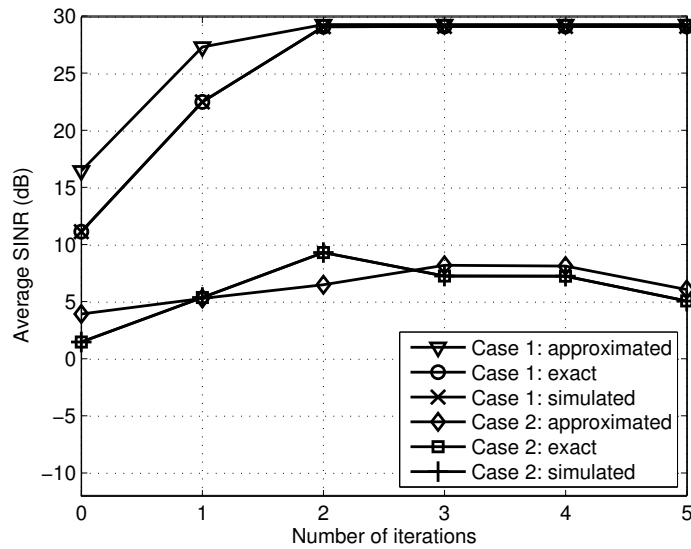


Figure 5.1: Theoretical average SINR for the proposed method, exact and approximated (Case 1: CFOs = {0.1, -0.2, -0.05, 0.2} and SNR = 30 dB; Case 2: CFOs = {0.49, 0.49, 0.15, 0.4} and SNR = 15 dB).

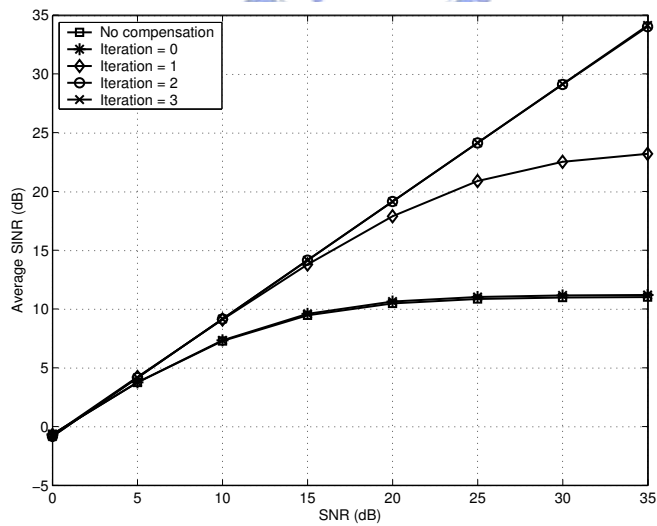


Figure 5.2: Theoretical average SINR for the proposed method (exact).

Table 5.2: Complexity comparison among direct ZF method, CLJL-PIC method, and proposed method when $N = 64$ and $Q = 4$.

Methods	Real multi- cations	Real divisions	Real additions
Direct ZF	373056	4160	368352
CLJL-PIC ($p = 2$)	48128	0	49536
Proposed ($S = 2$, $k = 2$)	13776 (0.037, 0.286)	8 (0.002, -)	15160 (0.041, 0.306)



Table 5.3: Complexity comparison of the direct ZF method, the banded ZF method, and the proposed method when $N = 2048$ and $Q = 16$.

Methods	Real multiplications	Real divisions	Real additions
Direct ZF	11474937856	4196352	11469003776
Banded ZF ($B = 16$)	3275760	69360	3532168
Proposed ($S = 2$, $k = 3$)	3109184 (0.000271, 0.949150)	32 (0.000008, 0.000461)	3236064 (0.000282, 0.916170)

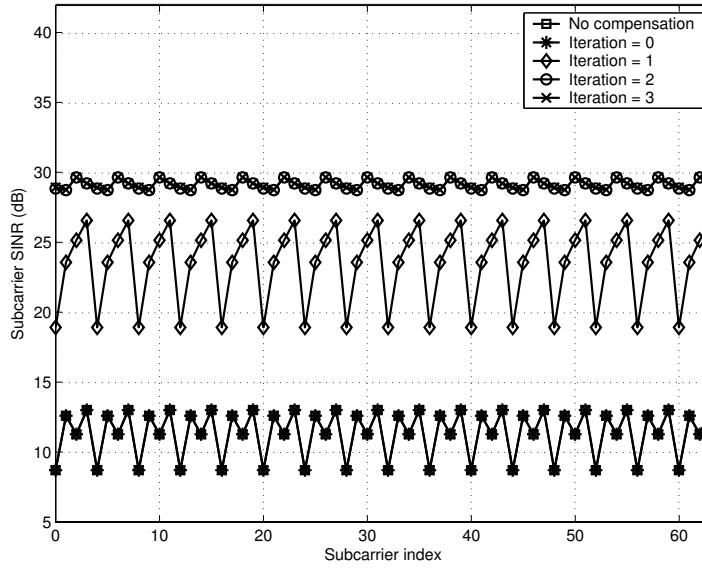


Figure 5.3: Theoretical subcarrier SINR for the proposed method (exact; SNR = 30 dB).

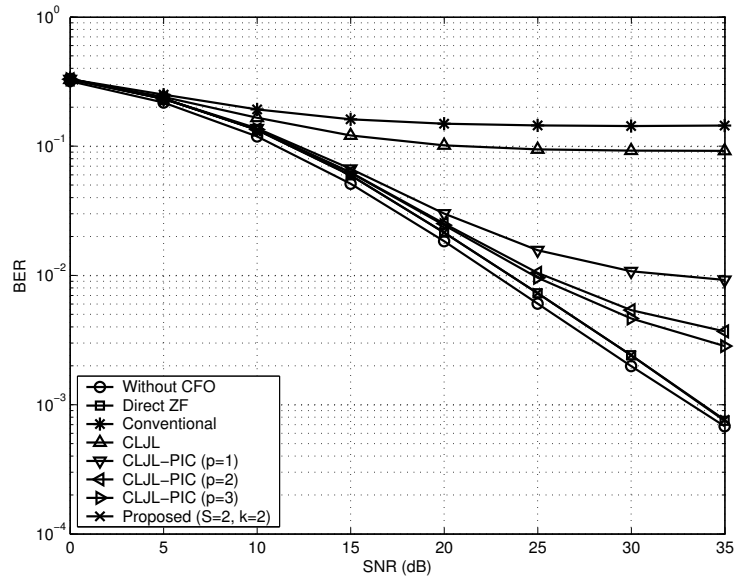


Figure 5.4: BER performance comparison for conventional, CLJL, CLJL-PIC, proposed, and direct ZF methods (16-QAM modulation, and CFOs = {0.1, -0.2, -0.05, 0.2}).

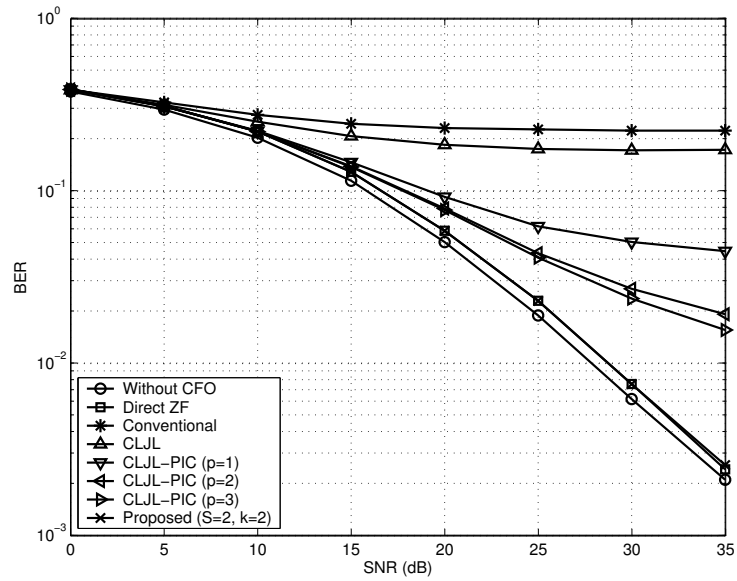


Figure 5.5: BER performance comparison for conventional, CLJL, CLJL-PIC, proposed, and direct ZF methods (64-QAM modulation, and CFOs = {0.1, -0.2, -0.05, 0.2}).

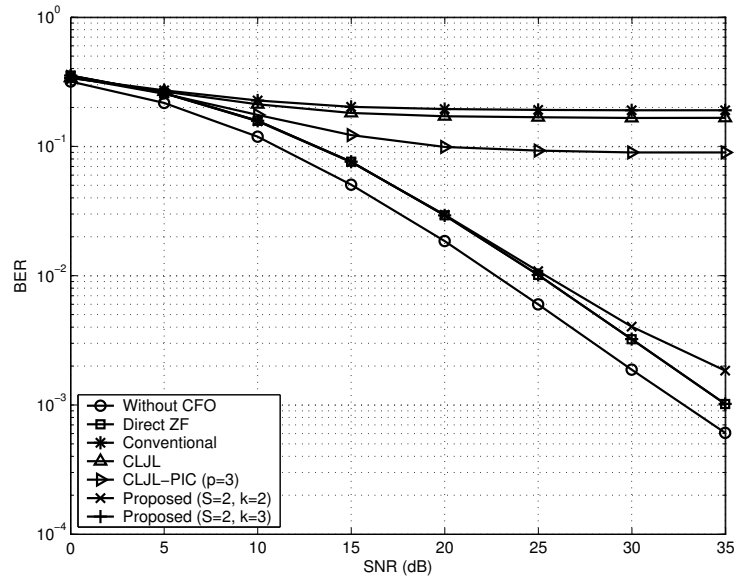


Figure 5.6: BER performance comparison for conventional, CLJL, CLJL-PIC, proposed, and direct ZF methods (16-QAM modulation, and CFOs = {0.1, -0.4, -0.05, 0.2}).

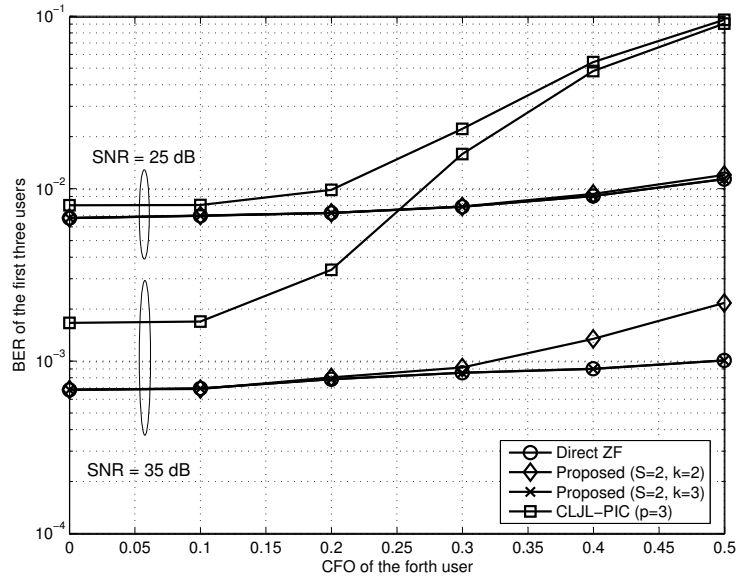


Figure 5.7: BER performance comparison for CLJL-PIC ($p = 3$), proposed, and direct ZF methods (16-QAM modulation, and CFO of the fourth user increases from 0 to 0.5).

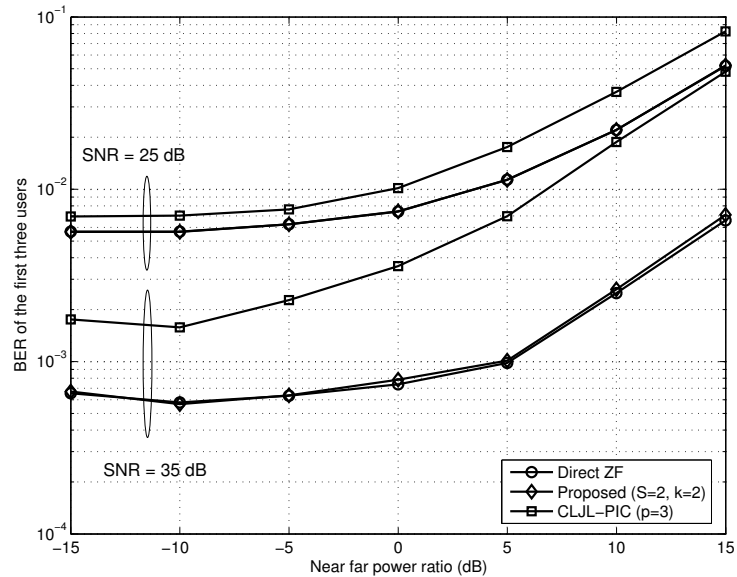


Figure 5.8: BER performance comparison for CLJL-PIC ($p = 3$), proposed ($k = 2$), and direct ZF methods (16-QAM modulation, CFOs = $\{0.1, -0.2, -0.05, 0.2\}$, and near-far scenario).

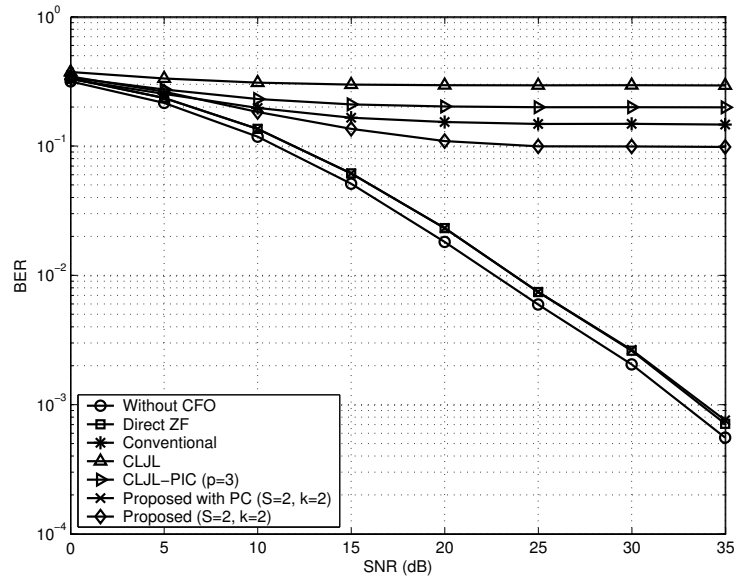


Figure 5.9: BER performance comparison for the proposed method with and without PC (16-QAM modulation, and CFOs = $\{0.49, 0.49, 0.1, 0.4\}$).

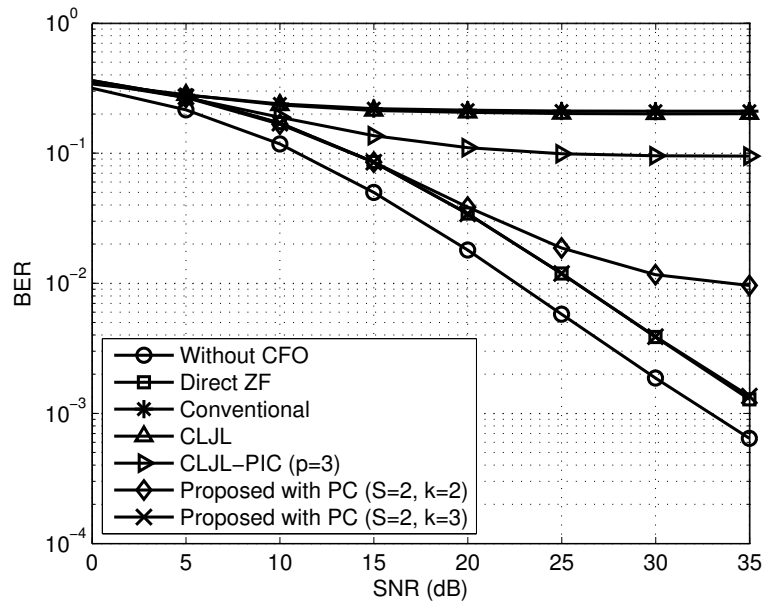


Figure 5.10: BER performance comparison for the proposed method with PC (16-QAM modulation and CFOs = $\{-0.1, 0.3, 0.4, -0.2\}$).

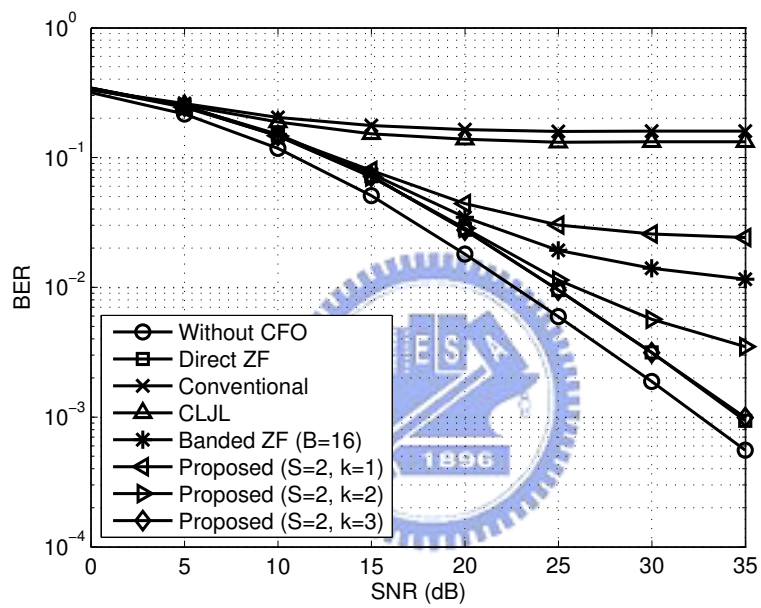


Figure 5.11: BER performance comparison for the conventional, CLJL, banded ZF, and direct ZF and proposed methods (16-QAM modulation, and CFOs = {0.1, -0.2, -0.05, 0.2, -0.3, 0, -0.1, 0.4, -0.3, 0.05, 0, -0.1, 0.05, -0.1, 0.3, 0.15}).

Chapter 6

Conclusions

In this dissertation, we investigate ICI mitigation methods in high-mobility SISO/MIMO-OFDM(A) and CFO-presented OFDMA uplink systems. Conventional ZF and MMSE methods require the inversion of a large ICI matrix, and the computational complexity is usually very high. This precludes the real-world application of the ICI mitigation methods. This is the motivation of our development of low-complexity ZF and MMSE algorithms. The main idea is to explore the special structure inherent in the ICI matrix resulting from mobility or CFO and to apply Newton's iteration for matrix inversion. Using our formulation, we can use FFTs/IFFTs in the iteration, and this can reduce the computational complexity dramatically. Note that the FFT/IFFT module is embedded in OFDM/OFDMA transceiver, and the required circuitries to implement the proposed algorithms are limited. We also analyze the convergence behavior of the proposed methods and derive theoretical output SINRs for some scenarios. Simulations show that the performances of the proposed ZF and MMSE methods are almost as good as those of the direct ZF and MMSE methods. In concluding the dissertation, we suggest some possible topics for further research.

1. In the dissertation, we only consider the ZF and MMSE methods for ICI mitigation. As described in Chapter 5, other nonlinear methods can be combined with the proposed methods to obtain further performance improvement. For example, we can use the pro-

posed ZF or MMSE method to obtain initial estimates of the desired data, and apply SIC for further processing. Note that in the scenario, we will have two iterative processes. How to allocate the iteration numbers to obtain the optimum results will be the main issue.

2. For MIMO-OFDM systems, we only consider the application of spatial multiplexing. As we know, an important class of the MIMO technology is the space-time coding. The assumption for the application of an orthogonal space-time code is that the channel response must be static during the transmission of a codeword. However, in high-mobility MIMO-OFDM environments, this assumption no longer holds. As a result, the orthogonality is destroyed, and the performance is degraded. The proposed ZF and MMSE methods can then be used to solve the problem.
3. Except for OFDMA, there exist other more efficient methods for multiple access. For example, for MIMO systems we can conduct the ZF method in the transmitter side (other than the receiver side) to cancel the MUI. This is usually referred to as the multiuser downlink technology. The proposed low-complexity methods may find applications in the area.
4. In this dissertation, we do not consider channel coding. In real-world applications, channel coding is an essential operation conducted in the transmitter side. If we treat the ICI as the result of an inner code, and the channel encoder as an outer code, we can then apply turbo equalization at the receiver. Turbo equalization involves an iteration between equalization and decoding. The equalization can be realized with the MMSE method. Note that the input of the MMSE method includes priori signal information. How to extend the proposed low-complexity ZF and MMSE methods to include the priori information will be the main issue. Note that we have two iterative processes, and how to allocate the iteration number will be another issue.

Bibliography

- [1] S. B. Weinstein and P. B. Ebert, "Data transmission by frequency-division multiplexing using the discrete Fourier transform," *IEEE Trans. Commun. Technology*, vol. 19, no. 5, pp. 628–634, Oct. 1971.
- [2] A. Peled and A. Ruiz, "Frequency domain data transmission using reduced computational complexity algorithms," in *Proc. IEEE ICASSP'80*, vol. 5, Apr. 1980, pp. 964–967.
- [3] J. A. C. Bingham, "Multicarrier modulation for data transmission: an idea whose time has come," *IEEE Commun. Mag.*, vol. 28, no. 5, pp. 5–14, May 1990.
- [4] I. Koffman and V. Roman, "Broadband wireless access solutions based on OFDM access in IEEE 802.16," *IEEE Commun. Mag.*, vol. 40, no. 4, pp. 96–103, Apr. 2002.
- [5] T. Kwon, H. Lee, S. Choi, J. Kim, D.-H. Cho, S. Cho, S. Yun, W.-H. Park, and K. Kim, "Design and implementation of a simulator based on a cross-layer protocol between MAC and PHY layers in a WiBro compatible IEEE 802.16e OFDMA system," *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 136–146, Dec. 2005.
- [6] *Air Interface for Fixed and Mobile Broadband Wireless Access Systems—Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1*, IEEE Std. 802.16e-2005, 2005.
- [7] M. Russell and G. L. Stüber, "Interchannel interference analysis of OFDM in a mobile environment," in *Proc. IEEE VTC'95*, vol. 2, July 25–28, 1995, pp. 820–824.

- [8] P. Robertson and S. Kaiser, "The effects of Doppler spreads in OFDM(A) mobile radio systems," in *Proc. IEEE VTC'99-Fall*, vol. 1, Sept. 19–22, 1999, pp. 329–333.
- [9] Y. Li and L. J. Cimini, Jr., "Bounds on the interchannel interference of OFDM in time-varying impairments," *IEEE Trans. Commun.*, vol. 49, no. 3, pp. 401–404, Mar. 2001.
- [10] J. Li and M. Kavehrad, "Effects of time selective multipath fading on OFDM systems for broadband mobile applications," *IEEE Commun. Lett.*, vol. 3, no. 12, pp. 332–334, Dec. 1999.
- [11] H. Steendam and M. Moeneclaey, "Analysis and optimization of the performance of OFDM on frequency-selective time-selective fading channels," *IEEE Trans. Commun.*, vol. 47, no. 12, pp. 1811–1819, Dec. 1999.
- [12] Y. Zhang and H. Liu, "MIMO-OFDM Systems in the Presence of Phase Noise and Doubly Selective Fading," *IEEE Trans. Vehicular Technology*, vol. 56, no. 4, pp. 2277–2285, July 2007.
- [13] T. Pollet, M. Van Bladel, and M. Moeneclaey, "BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise," *IEEE Trans. Commun.*, vol. 43, no. 7, pp. 191–193, Feb./Mar./Apr. 1995.
- [14] L. Wei and C. Schlegel, "Synchronization requirements for multi-user OFDM on satellite mobile and two-path Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 43, no. 2–4, pp. 887–895, Feb.–Apr. 1995.
- [15] M. S. El-Tanany, Y. Wu, and L. Hazy, "OFDM uplink for interactive broadband wireless: analysis and simulation in the presence of carrier, clock and timing errors," *IEEE Trans. Broadcast.*, vol. 47, no. 1, pp. 3–19, Mar. 2001.

- [16] W. G. Jeon, K. H. Chang, and Y. S. Cho, "An equalization technique for orthogonal frequency-division multiplexing systems in time-variant multipath channels," *IEEE Trans. Commun.*, vol. 47, no. 1, pp. 27–32, Jan. 1999.
- [17] Y.-S. Choi, P. J. Voltz, and F. A. Cassara, "On channel estimation and detection for multi-carrier signals in fast and selective Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1375–1387, Aug. 2001.
- [18] X. Cai and G. B. Giannakis, "Bounding performance and suppressing intercarrier interference in wireless mobile OFDM," *IEEE Trans. Commun.*, vol. 51, no. 12, pp. 2047–2056, Dec. 2003.
- [19] W.-S. Hou and B.-S. Chen, "ICI cancellation for OFDM communication systems in time-varying multipath fading channels," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2100–2110, Sept. 2005.
- [20] A. Gorokhov and J.-P. Linnartz, "Robust OFDM receivers for dispersive time-varying channels: equalization and channel acquisition," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 572–583, Apr. 2004.
- [21] S. Tomasin, A. Gorokhov, H. Yang, and J.-P. Linnartz, "Iterative interference cancellation and channel estimation for mobile OFDM," *IEEE Trans. Wireless Commun.*, vol. 4, no. 1, pp. 238–245, Jan. 2005.
- [22] X. Huang and H.-C. Wu, "Robust and Efficient Intercarrier Interference Mitigation for OFDM Systems in Time-Varying Fading Channels," *IEEE Trans. Vehicular Technology*, vol. 56, no. 5, pp. 2517–2528, Sept. 2007.
- [23] W.-G. Song and J.-T. Lim, "Channel estimation and signal detection for MIMO-OFDM with time varying channels," *IEEE Commun. Lett.*, vol. 10, no. 7, pp. 540–542, July 2006.

- [24] Y. Zhao and S.-G. Häggman, "Intercarrier interference self-cancellation scheme for OFDM mobile communication systems," *IEEE Trans. Commun.*, vol. 49, no. 7, pp. 1185–1191, July 2001.
- [25] Y. Zhang and H. Liu, "Frequency-domain correlative coding for MIMO-OFDM systems over fast fading channels," *IEEE Commun. Lett.*, vol. 10, no. 5, pp. 347–349, May 2006.
- [26] K. W. Park and Y. S. Cho, "An MIMO-OFDM technique for high-speed mobile channels," *IEEE Commun. Lett.*, vol. 9, no. 7, pp. 604–606, July 2005.
- [27] S. Das and P. Schniter, "Beamforming and combining strategies for MIMO-OFDM over doubly selective channels," in *Proc. IEEE ACSSC'06*, Oct.-Nov. 2006, pp. 804–808.
- [28] A. Stamoulis, S. N. Diggavi, and N. Al-Dhahir, "Intercarrier interference in MIMO OFDM," *IEEE Trans. Signal Processing*, vol. 50, no. 10, pp. 2451–2464, Oct. 2002.
- [29] I. Barhumi, G. Leus, and M. Moonen, "Equalization for OFDM over doubly selective channels," *IEEE Trans. Signal Processing*, vol. 54, no. 4, pp. 1445–1458, Apr. 2006.
- [30] I. Barhumi, G. Leus, and M. Moonen, "Time-varying FIR equalization for doubly selective channels," *IEEE Trans. Wireless Commun.*, vol. 4, no. 1, pp. 202–214, Jan. 2005.
- [31] I. Barhumi, G. Leus, and M. Moonen, "Time-domain and frequency-domain per-tone equalization for OFDM over doubly selective channels," *Signal Process.*, vol. 44/11, pp. 2055–2066, Nov. 2004.
- [32] K. van Acker, G. Leus, M. Moonen, O. van de Wiel, and T. Pollet, "Per tone equalization for DMT-based systems," *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 109–119, Jan. 2001.
- [33] G. Leus and M. Moonen, "Per-tone equalization for MIMO OFDM systems," *IEEE Trans. Signal Processing*, vol. 51, no. 11, pp. 2965–2975, Nov. 2003.

- [34] P. Schniter, “Low-complexity equalization of OFDM in doubly selective channels,” *IEEE Trans. Signal Processing*, vol. 52, no. 4, pp. 1002–1011, Apr. 2004.
- [35] R. Koetter, A. C. Singer, and M. Tuchler, “Turbo equalization,” *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 67–80, Jan. 2004.
- [36] H. Chae, Y. Kim, J. K. Lee, and D. K. Kim, “Complexity and Performance of Turbo Equalization for OFDM Systems in Doubly Selective Channels,” in *Proc. IEEE 65th VTC2007-Spring*, 22-25 Apr. 2007, pp. 1608–1611.
- [37] K. Fang, L. Rugini, and G. Leus, “Low-complexity block turbo equalization for OFDM system in time-varying channels,” *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5555–5566, Nov. 2008.
- [38] J.-J. van de Beek, M. Sandell, and P. O. Börjesson, “ML estimation of time and frequency offset in OFDM systems,” *IEEE Trans. Signal Process.*, vol. 45, no. 7, pp. 1800–1805, July 1997.
- [39] P. H. Moose, “A technique for orthogonal frequency division multiplexing frequency offset correction,” *IEEE Trans. Commun.*, vol. 42, no. 10, pp. 2908–2914, Oct. 1994.
- [40] J. Armstrong, “Analysis of new and existing methods of reducing intercarrier interference due to carrier frequency offset in OFDM,” *IEEE Trans. Commun.*, vol. 47, no. 3, pp. 365–369, Mar. 1999.
- [41] K. Sathanathan, R. M. A. P. Rajatheva, and S. B. Slimane, “Cancellation technique to reduce intercarrier interference in OFDM,” *IEE Electron. Lett.*, vol. 36, no. 25, pp. 2078–2079, Dec. 2000.
- [42] H. Yoo and D. Hong, “Edge sidelobe suppressor scheme for OFDMA uplink systems,” *IEEE Commun. Lett.*, vol. 7, no. 11, pp. 534–536, Nov. 2003.

- [43] K. Sathananthan and C. Tellambura, "Partial transmit sequence and selected mapping schemes to reduce ICI in OFDM systems," *IEEE Commun. Lett.*, vol. 6, no. 8, pp. 313–315, Aug. 2002.
- [44] S.-H. Tsai, Y.-P. Lin, and C.-C. J. Kuo, "An approximately MAI-free multiaccess OFDM system in carrier frequency offset environment," *IEEE Trans. Signal Processing*, vol. 53, no. 11, pp. 4339–4353, Nov. 2005.
- [45] A. M. Tonello and S. Pupolin, "Performance of single user detectors in multitone multiple access asynchronous communications," in *Proc. IEEE 55th VTC*, vol. 1, May 6–9, 2002, pp. 199–203.
- [46] J. Choi, C. Lee, H. W. Jung, and Y. H. Lee, "Carrier frequency offset compensation for uplink of OFDM-FDMA systems," *IEEE Commun. Lett.*, vol. 4, no. 12, pp. 414–416, Dec. 2000.
- [47] D. Huang and K. B. Letaief, "An interference-cancellation scheme for carrier frequency offsets correction in OFDMA systems," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1155–1165, July 2005.
- [48] M. Huang, X. Chen, S. Zhou, and J. Wang, "Iterative ICI cancellation algorithm for uplink OFDMA system with carrier-frequency offset," in *Proc. IEEE 62nd VTC-2005-Fall*, vol. 3, Sept. 25–28, 2005, pp. 1613–1617.
- [49] R. Fantacci, D. Marabissi, and S. Papini, "Multiuser interference cancellation receivers for OFDMA uplink communications with carrier frequency offset," in *Proc. IEEE GLOBECOM '04*, vol. 5, Nov. 29–Dec. 3, 2004, pp. 2808–2812.
- [50] Z. Cao, U. Tureli, Y.-D. Yao, and P. Honan, "Frequency synchronization for generalized OFDMA uplink," in *Proc. IEEE GLOBECOM '04*, vol. 2, Nov. 29–Dec. 3, 2004, pp. 1071–1075.

- [51] Z. Cao, U. Tureli, and Y.-D. Yao, "Analysis of two receiver schemes for interleaved OFDMA uplink," in *Proc. Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, vol. 2, Nov. 3–6, 2002, pp. 1818–1821.
- [52] Z. Cao, U. Tureli, and Y.-D. Yao, "Deterministic multiuser carrier-frequency offset estimation for interleaved OFDMA uplink," *IEEE Trans. Commun.*, vol. 52, no. 9, pp. 1585–1594, Sept. 2004.
- [53] M.-O. Pun, M. Morelli, and C.-C. J. Kuo, "Maximum-likelihood synchronization and channel estimation for OFDMA uplink transmissions," *IEEE Trans. Commun.*, vol. 54, no. 4, pp. 726–736, Apr. 2006.
- [54] J. J. van de Beek, P. O. Börjesson, M.-L. Boucheret, D. Landström, J. M. Arenas, P. Ödling, C. Östberg, M. Wahlqvist, and S. K. Wilson, "A time and frequency synchronization scheme for multiuser OFDM," *IEEE J. Select. Areas Commun.*, vol. 17, no. 11, pp. 1900–1914, Nov. 1999.
- [55] M. Moreli, "Timing and frequency synchronization for the uplink of an OFDMA system," *IEEE Trans. Commun.*, vol. 52, no. 2, pp. 296–306, Feb. 2004.
- [56] X. Qian and L. Zhang, "Interchannel interference cancellation in wireless OFDM systems via Gauss-Seidel method," in *Proc. IEEE ICCT2003*, Apr. 9–11, 2003, pp. 1051–1055.
- [57] A. F. Molisch, M. Toeltsch, and S. Vermani, "Iterative methods for cancellation of inter-carrier interference in OFDM Systems," *IEEE Trans. Vehicular Technology*, vol. 56, no. 4, pp. 2158–2167, July 2007.
- [58] J. Fu, C.-Y. Pan, Z.-X. Yang, and L. Yang, "Low-complexity equalization for TDS-OFDM systems over doubly selective channels," *IEEE Trans. Broadcast.*, vol. 51, no. 3, pp. 401–407, Sept. 2005.

- [59] Y. Mostofi and D. C. Cox, "ICI mitigation for pilot-aided OFDM mobile systems," *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 765–774, Mar. 2005.
- [60] G. H. Golub and C. F. Van Loan, *Matrix Computations*. London, The Johns Hopkins Press Ltd., 1996.
- [61] G. Schulz, "Iterative Berechnung der reziproken Matrix," *Z. Angew. Math. Mech.*, v. 13, pp. 57–59, 1933.
- [62] A. S. Householder, *The Theory of Matrix in Numerical Analysis*. New York, Dover Publications, Inc., 1964.
- [63] A. Ben-Israel, "A note on iterative method for generalized inversion of matrices," *Math. Comp.*, vol. 20, no. 95, pp. 439–440, July 1966.
- [64] A. Ben-Israel and D. Cohen, "On iterative computation of generalized inverses and associated projections," *SIAM J. Numer. Anal.*, vol. 3, no. 3, pp. 410–419, Sept. 1966.
- [65] V. Y. Pan and R. Schreiber, "An improved Newton iteration for the generalized inverse of a matrix, with applications," *SIAM J. Sci. Stat. Comput.*, vol. 12, no. 5, pp. 1109–1130, Sept. 1991.
- [66] V. Y. Pan and J. Reif, "Fast and efficient parallel solution of dense linear systems," *Computers and Mathematics with Applications*, vol. 17, no. 11, pp. 1481–1491, 1989.
- [67] V. Y. Pan, Y. Rami, and X. Wang, "Structured matrices and Newton's iteration: unified approach," *Linear Algebra Appl.*, vol. 343–344, pp. 233–265, Mar. 2002.
- [68] D. A. Harville, *Matrix Algebra From a Statistician's Perspective*. New York, Springer-Verlag New York, Inc., 1997.
- [69] S. J. Leon, *Linear Algebra With Applications*. New Jersey, Prentice-Hall International, Inc., 1998.

- [70] M. F. Pop and N. C. Beaulieu, "Limitations of sum-of-sinusoids fading channel simulators," *IEEE Trans. Commun.*, vol. 49, no. 4, pp. 699–708, Apr. 2001.



自傳

姓 名：許兆元

性 別：男

出生年月日：民國 68 年 06 月 22 日

籍 貫：高雄縣

學 歷：國立中山大學電機系學士(民國 86 年 9 月~民國 90 年 6 月)

國立交通大學電信所碩士(民國 90 年 9 月~民國 92 年 6 月)

國立交通大學電信所博士(民國 92 年 9 月~民國 98 年 2 月)

獲 獎：國立中山大學電機系第一名畢業

中華民國斐陶斐榮譽學會會員

國立交通大學碩士班書卷獎

