FAST EQUALIZATION ALGORITHMS FOR WIDE BAND WIRELESS COMMUNICATION SYSTEMS

BY

NAVEED IQBAL

A Dissertation Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In

ELECTRICAL ENGINEERING

APRIL 2014

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **NAVEED IQBAL** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING**.

Dissertation Committee

azzdine zerguine

Dr. Azzedine Zerguine (Adviser)

Dr. Tareq Y. Al-Naffouri (Member)

Dr. Mohamed Deriche (Member)

Dr. Abdelmalek Zidouri (Member)

Dr. Ali H. Muqaibel (Member)

Dr. Ali A. Al-shaikhi
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

21/5/14

Date

© Naveed Iqbal
April 2014

Dedicated to my loving
parents and family

ACKNOWLEDGMENTS

In the name of Allah, the Most Beneficent, the Most Merciful

All praise to Allah for His countless blessings. May Allah bestow blessings on His last prophet, Muhammad (sallallahu alaihi wasallam).

Firstly, I want to express my appreciation to King Fahd University of Petroleum and Minerals (KFUPM), and to its brilliant and knowledgeable faculty members who have been a motivation for me during my whole PhD.

It is giving my great pleasure to acknowledge the efforts of my advisor, Dr. Azzedine Zerguine, and key member of my committee, Dr. Naofal Al-Dhahir, whose expert opinion and enthusiastic encouragement helped in achieving my goals during my stay at KFUPM. Their friendly working environment and beneficial guidance made the PhD dissertation not only possible but interesting.

I am grateful to my dissertation committee members Dr. Tareq Y. Al-Naffouri, Dr.Mohamed Deriche, Dr. Ali H. Muqaibel and Dr. Abdelmalek Zidouri for their useful critique and feedback during my research work.

I would also like to thank all my friends at KFUPM for their moral support. It is only because of their friendship that made my stay here at KFUPM likable and delightful.

Specially, I want to thank those who hated me because it helped me to get closer to perfection.

Finally, I would like to thank my parents and siblings from the core of my heart for their prayers and moral support that they provided.

May Allah guide us to follow the righteous paths according to Quran and Sunna!(Ameen)

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

LIST OF TABLES ix

LIST OF FIGURES x

NOMENCLATURE xiv

ABSTRACT (ENGLISH) xviii

ABSTRACT (ARABIC) xxii

CHAPTER 1. INTRODUCTION 1

1.1 Background 1

1.1.1 Classification of Equalization techniques 3

1.2 Dissertation Contributions and Organization 7

CHAPTER 2. ADAPTIVE FREQUENCY-DOMAIN RLS DFE FOR UPLINK MIMO SC-FDMA 10

2.1 Introduction 11

2.2 System Description 16

2.2.1 Carrier Frequency Offset (CFO) 19

2.2.2 Fading and Doppler Effect 20

2.3 AFD-DFE FOR SISO SC-FDMA 21

2.3.1 RLS Update 22

2.3.2 Reduced-Complexity RLS Update 24
2.4 AFD-DFE FOR MIMO SC-FDMA 27
2.4.1 AFD-DFE for SFBC SC-FDMA 27
2.4.2 AFD-DFE for Spatially-Multiplexed (SM) SC-FDMA . . . 34
2.4.3 AFD-DFE for Hybrid SM-SFBC SC-FDMA 39
2.5 Performance and Complexity Analysis 46
2.5.1 Transient Analysis 46
2.5.2 Steady-State Analysis 50
2.5.3 Tracking Analysis 53
2.5.4 Computational Complexity 55
2.6 Simulation Results 57
2.7 Conclusion 64

CHAPTER 3. ADAPTIVE FREQUENCY-DOMAIN DECISION FEEDBACK EQUALIZATION USING CONSTRAINT-BASED RLS FOR UPLINK SC-FDMA 66

3.1 Introduction 68
3.2 Recursive Least Squares with constraint (CRLS) 71
3.2.1 Case 1 73
3.2.2 Case 2 79
3.2.3 Reduced-Complexity CRLS Update 82
3.2.4 Reduced-training AFD-DFE 83
3.3 Carrier Frequency Offset (CFO) in SC-FDMA 83
3.4 Space-Frequency Block Coded (SFBC) SC-FDMA 90
3.5 Carrier Frequency Offset (CFO) in SFBC SC-FDMA 95
3.6 Performance and Complexity Analysis 98
3.6.1 Transient Analysis 99
3.6.2 Steady-State Analysis 103
3.6.3 Tracking Analysis 105
3.7 Simulation Results 107

3.8 Conclusion 116

CHAPTER 4. DECISION FEEDBACK EQUALIZATION USING PARTICLE SWARM OPTIMIZATION 117

4.1 Introduction 118
4.2 The PSO Algorithm 122
    4.2.1 BASIC PSO 122
    4.2.2 PSO Variants 124
4.3 Problem Formulation 127
    4.3.1 SISO Case 127
    4.3.2 MIMO Case 129
4.4 Particle Swarm Optimization-Least Mean Squares (PSO-LMS) Algorithm 132
4.5 PSO-LMS-DFE For Complex-Valued Data 137
4.6 Simulation Results 139
    4.6.1 Parameter adjustment analysis 140
    4.6.2 Performance of PSO-DFE 143
4.7 Complexity Analysis 147
4.8 Conclusion 150

CHAPTER 5. ADAPTIVE EQUALIZATION USING PARTICLE SWARM OPTIMIZATION FOR UPLINK SC-FDMA 152

5.1 Introduction 153
5.2 PSO-based adaptive equalization 154
    5.2.1 Fitness Function 156
    5.2.2 Re-randomization 158
5.3 Complexity Analysis 159
5.4 Simulation Results 160
5.5 Conclusion 162

CHAPTER 6. SPARSE LEAST MEAN SQUARE (LMS) ALGORITHM 163

6.1 Introduction 163
6.2 Symbol by Symbol equalization 166
6.3 Block equalization 169
6.4 Sparse LMS 171
6.5 Computational Complexity 174
6.6 Simulation Results 175

CHAPTER 7. CONCLUSIONS AND FUTURE RECOMMENDATIONS 181

7.1 Conclusion 181
7.2 Future recommendations 183

REFERENCES 185

VITAE 197

LIST OF TABLES

2.1 Summary of the adaptation algorithm for AFD-DFE 26

2.2 Computational complexity of the AFD-DFE and MMSE DFE . . 56

3.1 Summary of the adaptation algorithm for AFD-DFE 84

3.2 Summary of the reduced-training adaptation algorithm for AFD-DFE 85

4.1 Computational complexity of different PSO algorithms 150

5.1 Computational complexity of PSO algorithms 160

6.1 Delay Spread of various channels for 100% active taps 178

LIST OF FIGURES

1.1 Model of a digital communication system. 2
1.2 Symbol spaced Channel. 5
1.3 Fractionally spaced Channel. 6
1.4 Comparison of symbol spaced equalizers 6
1.5 Comparison of fractionally spaced equalizers. 7

2.1 Block diagram of a transceiver of SC-FDMA system. 16
2.2 Block diagram of AFD-DFE. 25
2.3 Block diagram of a transceiver of SFBC SC-FDMA system. 27
2.4 Block diagram of AFD-DFE for two-transmit one-receive antenna. 33
2.5 Block diagram of a transceiver of a SM SC-FDMA system. 34
2.6 Block diagram of transceiver of Hybrid SM-SFBC SC-FDMA system. 39
2.7 Computational complexity of MMSE DFE [1] and proposed AFD-DFE versus the number of antennas for $M = 16$. 56
2.8 Learning curves of LE and DFE in SCFDMA system. 58
2.9 Comparison of RLS AFD-DFE, LMS AFD-DFE, LE and MMSE DFE [1] 59
2.10 System's performance under CFO's effect with Single user for two mapping techniques. 60
2.11 System's performance under CFO's effect with three users. 61
2.12 Effect of user's velocity on system's performance with single user. 62
2.13 Effect of user's velocity on system's performance with three users. 62

2.14 Theoretical and simulated MSE of RLS AFD-DFE as a function of user's velocity. 63
2.15 Effect of user's velocity on SFBC SC-FDMA system. 63
2.16 Effect of user's velocity on hybrid SM-SFBC SC-FDMA with two receive antennas and six users. 64

3.1 Structure of $\boldsymbol{\mathcal{C}}^{(m)}\hat{\boldsymbol{\Lambda}}^{(m)}$(normalized) matrix. 86
3.2 Learning curves of AFD-LE and AFD-DFE in SCFDMA system with user velocity $v = 3$ Km/h and SNR of 20 dB. 110
3.3 b_0 versus iteration. 110
3.4 Comparison of various equalizers with user velocity $v = 3$ Km/h . 110
3.5 Effect of velocity on system's performance 111
3.6 Theoretical and simulated MSE of RLS AFD-DFE as a function of user's velocity. 111
3.7 SC-FDMA system's performance under CFO's effect with user's velocity $v = 300$ Km/h 111
3.8 Comparison of E-SFBC and C-SFBC with user's velocity $v = 3$ Km/h 112
3.9 E-SFBC SC-FDMA System's performance under CFO's effect with user velocity $v = 300$ Km/h 113
3.10 Number of training symbols in SC-FDMA block vs MSE ($v = 3$ Km/h) 113
3.11 Constellation diagram, SNR= $20dB$ 114
3.12 MSE convergence curves, SNR= $20dB$ 115
3.13 Residual ISI convergence curves,SNR= $20dB$ 115

4.1 Flow chart of the particle swarm optimization algorithm. 124
4.2 Block diagram of the decision feedback equalizer. 127
4.3 2×2 MIMO system. 130
4.4 Flow chart of the PSO-LMS. 137
4.5 Flow chart of the PSO-LMS for complex-valued data. 139

4.6 Effect of the acceleration constants (c_1 and c_2). 140

4.7 Effect of the k_{max} and k_{min}. 140

4.8 Effect of the block size (N). 142

4.9 Effect of the number of particles (n). 142

4.10 Effect of the maximum velocity (v_{max}) and maximum position (p_{max}). 142

4.11 Learning Curves of different algorithms for channel $C1$. 143

4.12 Learning Curves of different algorithms for channel $C2$. 143

4.13 Learning Curves of different algorithms for MIMO channel. 144

4.14 MSE curves for blind equalization. 144

4.15 Symbol error Rate for $C1$. 145

4.16 Symbol error Rate for $C2$. 146

4.17 Symbol error Rate for MIMO channel. 146

4.18 Number of multiplications of various PSO algorithms versus block size (N). 150

4.19 Number of additions of various PSO algorithms versus block size (N). 150

5.1 Comparison of PSO algorithm with RLS and LMS for $f_d T_s = 0.0001$ 161

5.2 Effect of Doppler on PSO algorithm 162

6.1 Block diagram of the decision feedback equalizer. 166

6.2 Impulse response of the equalizer using LMS and SLMS with $L = 200$ and SNR=20 dB. 172

6.3 Impulse response of the equalizer using LMS and SLMS with dominant tap approach, $L = 200$ and SNR=20 dB. 174

6.4 Computational Ccmparison of LMS and SLMS. 175

6.5 Comparison of LMS and SLMS for SISO case with $L = 200$, SNR=20 dB. 176

6.6 MSE versus SNR for SISO case with $L = 200$. 176

6.7 Comparison of LMS and SLMS for MIMO case with $L = 200$, SNR=20 dB. 176

6.8 Effect of block size (MIMO case), SNR=20 dB. 177
6.9 MSE versus active tap percentage for various channels, SNR=20 dB. 179
6.10 Time constant τ versus active tap percentage, SNR=20 dB. . . . 179
6.11 Theatrical maladjustment, SNR=20 dB. 179
6.12 MSE versus step size μ, SNR=20 dB. 180

Nomenclature

Abbreviations

AFD-DFE	:	Adaptive Frequency Domain Decision Feedback Equalizer
BER	:	Bit Error Rate
BAFD-DFE	:	Blind AFD-DFE
CFO	:	Carrier Frequency Offset
CSI	:	Channel State Information
CRLS	:	Constraint-based block Recursive Least Squares
DFE	:	Decision Feedback Equalizer
EMSE	:	Excess Mean Square Error
FD-PSO	:	Frequency-domain Particle Swarm Optimization
ICI	:	Inter-Carrier Interference
LMS	:	Least Mean Square algorithm
LE	:	Linear Equalizer
LTE	:	Long Term Evolution
ML	:	Maximum Likelihood
MSE	:	Mean Square Error
MMSE	:	Minimum Mean Square Error
MIMO	:	Multiple Input Multiple Output
OFDMA	:	Orthogonal Frequency Division Multiple Access

PSO : Particle Swarm Optimization

SFBC : Space-Frequency Block code

STBC : Space-Time Block Codes

SM : Spatially-Multiplexed

SC-FDE : Single Carrier Frequency Domain Equalization

SC-FDMA : Single Carrier Frequency Division Multiple Access

SNR : Signal-to-Noise Ratio

TD-PSO : Time-domain Particle Swarm Optimization

Notations

k : Iteration number

$E[.]$: Expectation operator

$(.)^T$: Matrix transposition

$(.)^*$: Complex conjugation

$(.)^H$: Hermitian transposition for matrices

$Re[x]$: Real part of x

μ : Step-size

M : Size of SC-FDMA block

α : Lagrange multiplier

$||.||^2$: Squared Euclidean norm

$Tr\{.\}$: trace of a matrix

$||\boldsymbol{x}||_W^2$: $\boldsymbol{x}^* W \boldsymbol{x}$ for a column vector $\boldsymbol{x}$ and positive-definite matrix W

$diag(\boldsymbol{x})$: Diagonal matrix with entries read from $\boldsymbol{x}$

N : Number of sub-carriers in SC-FDMA system

$\sigma_{\mathcal{N}}$: Noise variance

$\boldsymbol{R}^{(m)}$: $N \times M$ resource allocation matrix for m^{th} user

$\boldsymbol{x}^{(m)}$: m^{th} user's time-domain data symbol

$\boldsymbol{\mathcal{X}}^{(m)}$: m^{th} user's frequency-domain data symbol

$\boldsymbol{F}_N$: $N \times N$ DFT matrix

$\boldsymbol{\Lambda}^{(m)}$: m^{th} user's frequency-domain channel matrix

$\boldsymbol{\mathcal{N}}$: Frequency-domain noise vector

$\boldsymbol{h}^{(m)}$: m^{th} user's time-domain channel matrix

Ω_m : m^{th} user's CFO normalized by the sub-carrier spacing

f_d : Doppler frequency

$\boldsymbol{\mathcal{F}}$: Vector of the frequency-domain Feedforward filter coefficients

$\boldsymbol{\mathcal{B}}$: Vector of the frequency-domain Feedback filter coefficients

$\boldsymbol{\mathcal{D}}$: Decision matrix

$\boldsymbol{\mathcal{W}}$: Frequency domain weight vector of the equalizer

$\widetilde{\boldsymbol{\mathcal{W}}}$: Frequency domain weight error vector

$MSE(i)$: MSE of the i^{th} frequency bin

J_L : MMSE for the LE

J_D : MMSE for the DFE

J_{k+1}^r : MSE of the RLS AFD-DFE at instant $k+1$

J_{k+1}^l : MSE of the LMS AFD-DFE at instant $k+1$

ξ_{k+1}^a : a priori estimation output error

ξ_{k+1}^p : a posteriori estimation error

$\boldsymbol{R}_{\mathcal{A}}$: Input correlation matrix of the AFD-DFE

$\boldsymbol{p}_{i,k}$: i^{th} particle position at instant k

$\boldsymbol{v}_{i,k}$: i^{th} particle velocity at instant k

THESIS ABSTRACT

NAME:	Naveed Iqbal
TITLE OF STUDY:	Fast Equalization Algorithms for Wide Band Wireless Communication Systems
MAJOR FIELD:	Electrical Engineering
DATE OF DEGREE:	April 2014

It is well known that in the case of highly frequency-selective fading channels, the Linear Equalizer (LE) can suffer significant performance degradation compared to the Decision Feedback Equalizer (DFE). For time varying channels adaptive equalizers are used and one of the famous adaptive algorithm is the Least Mean Square (LMS) algorithm but it has the limitation of slow convergence. However, there are other algorithms like Recursive Least Squares (RLS) algorithm that have fast convergence but their high complexity limit their use. Beside these algorithms there are heuristic approaches like Particle Swarm Optimization (PSO). Unlike stochastic gradient algorithms, PSO is known to have fast convergence which does not depend on the underlying structure.This dissertation proposes new ideas to improve performance and reduce complexity of the above mentioned algorithms.

In particular, the contributions are as follows.

First, we develop a low-complexity Adaptive Frequency Domain Decision Feedback Equalizer (AFD-DFE) for Single Carrier Frequency Division Multiple Access (SC-FDMA) systems, where both the feedforward and feedback filters operate in the frequency-domain and are adapted using the well-known block RLS algorithm. SC-FDMA has been adopted as a multiple access technique for uplink in Long Term Evolution (LTE) standard. Since this DFE design operation is performed entirely in the frequency-domain, the complexity of the block RLS algorithm can be reduced substantially when compared to its time-domain counterpart by exploiting matrix structure in the frequency-domain. We extend our formulation to Multiple Input Multiple Output (MIMO) SC-FDMA systems where we show that the AFD-DFE enjoys a significant reduction in computational complexity when compared to the frequency-domain non-adaptive DFE. Extensive simulations are carried out to demonstrate the robustness of our proposed AFD-DFE to high Doppler and Carrier Frequency Offset (CFO).

Second, we develop a constraint-based block Recursive Least Squares (CRLS) for an AFD-DFE when used in an uplink SC-FDMA systems. The performance of the CRLS algorithm is better than that of the RLS with no significant increase in the computational complexity. Moreover, we extend our design to the Space-Frequency Block code (SFBC) SC-FDMA system. We also show that the AFD-DFE with CRLS not only enjoys a significant reduction in computational complexity when compared to the frequency-domain non-adaptive optimum MMSE-DFE

but its performance is also better than the practical MMSE DFE (the one with error decisions) and closed to the ideal MMSE DFE (the one with correct decisions). Simulation results are carried out to demonstrate the robustness of our proposed algorithm to high Doppler. To mitigate Inter-Carrier Interference (ICI) due to large CFO, we have designed a 3*-tap AFD-DFE for Single-Input Single-Output (SISO) and SFBC SC-FDMA systems by exploiting the banded and sparse structure of the channel. We show that the* 3*-tap AFD-DFE has an excellent performance as compared to the* 1*-tap AFD-DFE with a low computational complexity. Further, it is shown that we can reduce the training symbols in each SC-FDMA block that are transmitted in the training phase with no significant performance degradation. To further reduce the overhead blind AFD-DFE is also introduced.*

Third, PSO is used for adaptive equalization. For time-varying channels, adaptive equalizers are commonly designed based on the LMS algorithm which, unfortunately, has the limitation of slow convergence specially in channels having large eigenvalue spread. The eigenvalue problem becomes even more pronounced in MIMO channels. PSO enjoys fast convergence and, therefore, its application to the DFE merits investigation. In this work, we show that a PSO-DFE with a variable constriction factor is superior to the LMS/RLS-based DFE (LMS/RLS-DFE) and PSO-based LE (PSO-LE), especially on channels with large eigenvalue spread. We also propose a hybrid PSO-LMS-DFE algorithm, and modify it to deal with complex-valued data. The PSO-LMS-DFE not only outperforms the PSO-DFE in terms of performance but its complexity is also low. To further reduce its *complexity, a fast PSO-LMS-DFE algorithm is introduced. The system overhead is reduced by devising a blind PSO algorithm.*

Fourth, an adaptive frequency-domain equalizer for SC-FDMA system using PSO technique is proposed. The cost function used in a PSO is formulated based on the respective structure of the equalizer, whether it is LE or a DFE. The robustness of our proposed PSO algorithm is demonstrated on a high Doppler scenario. Furthermore it is shown that the performance improves more when using re-randomization. It is shown that the PSO based frequency domain equalizer is more computationally efficient than its time domain counterpart.

Lastly, to increase convergence speed of the LMS algorithm,adaptive equalizer based on Sparse LMS (SLMS) is devised.

مُلخّص الرسالة

الاسـم: نفيد إقبـال

عنوان الرسالة: خوارزمية سريعة لمعادلة انظمة الاتصال اللاسلكية ذات النطاق الترددي الواسع

التخصص: الهندسة الكهربائية

تاريخ الدرجة العلمية: نيسان 2014

معروف جيدا أنه في حالة اختيار الترددات المرتفعة في القناوات المتلاشية. المعادل الخطي يمكن أن يعاني من إضمحلال في الأداء بالمقارنة مع المعادل الذي يعتمد على ردود الفعل الخلفية المختارة. بالنسبة للقنوات المتغيرة مع الزمن, يستخدم المعادل المتكيف و من أشهر الخوارزميات المتكييفة هي خوارزميةمعدل التربيع الأقل, لكن يوجد فيها قصور نتيجة بطئ الألتقاء فيها. على الرغم من ذلك هناك خوارزميات مثل خوارزمية أقلتربيع متكرر لديها سرعة التقاء لكن التعقيد فيها يحد من استخدامها. بجانب هذه الخوارزميات هناك نهج قريبيساعد على الكشف مثل سرب الجسيمات الأمثل. على عكس خوارزميات الميل الأحصائية, سرب الجسيمات الأمثل يعرف بمتلاكه سرعة في الاتقاء و الذي لا يعتمد على البنية الأساسية. هذه الأطروحة تقدم فكرة جديدة لتطوير الأداء و تقليل التعقيد للخوارزميات سابقة الذكر. على وجه الخصوص المساهمات هي كالاتي.

أولا, لقد طورنا معادل متكيف يعتمد على ردود الفعل المختارة يعمل في المجال الترددي و قليل التعقيد لنظام أحادي الناقل تقسيم متعدد الوصول, حيث أن مرشح ردود الفعل الخلفية و مرشح ردود الفعل الأمامية يعملان في المجال الترددي و متكيفان نتيجة لأستخدام صندوق خوارزمية التربيع الأقلالمتكرر. المعادل المتكيف الذي يعتمد على ردود الفعل المختارة استند على تقنية تعدد الوصول في الأرسال حسب معيار المدى الطويل المتطور. حيث أن عملية تصميم معادل ردود الفعل المختارة تحصل في نطاق المجال الترددي, أما التعقيد في صندوق خوارزمية معدل التربيع الأقل المتكرر يمكن تقليله جوهريا عندما نقارنه بنظيره في الأمتداد في المجال الزمني من خلال استغلال البنية المصفوفية في المجال الترددي . لقد وسعنا صيغاتنا لنظام متعدد الوصول متعدد الخروج حيث وضحنا أن المعادل المتكيف الذي يعتمد على ردود الفعل الخلفية المختارة يتمتعبالتقليل في التعقيدات المحوسبة عندما نقارنه مع المجال الترددي الغير متكيف لمعادل ردود الفعل الخلفية.بناءا على ذلكفإن محاكاة موسعة تم تنفيذها لملاحظة قوة مقترحنا في المجال الترددي المتكيف لمعادل ردود الفعل الخلفية في الدوبلر العالي وموازن ناقل التردد.

ثانيا, لقد طورنا قيود بالأعتماد على التربيع الأقل المتكرر للمجال الترددي المتكيف لمعادل ردود الفعل المختارة عند استخدامه في الأرسال في نظام الناقل الأحادي في المجال الترددي متعدد الوصول. الأداء في خوارزمية قيود التربيع الأقل المتكرر أفضل من التربيع الأقل المتكرر مع زيادة في التعقيدات المحوسبة. علاوة على ذلك لقد وسعنا تصميمنا لصندوق الرمزالفراغي الترددي لنظام الناقل الأحادي في المجال الترددي متعدد الوصول. و أيضا وضحنا أن المجال الترددي المتكيف لمعادل ردود الفعل المختارة مع قيود التربيع الأقل المتكرر لا تتمتع فقط بتقليل في التعقديات المحوسبة فقط و انما في أداء أفضل من أقل معدل تربيع الخطأ في المعادل لردود الفعل المختارة العملي الذي يحتوي على القرارت

الخاطئة. و قريبه من أقل معدل تربيع الخطأ في المعادل لردود الفعل المختارة المثالي الذي يحتوي على القرارات الصحيحة. نتائج المحاكاة تم ايجادها لتوضح قوة الخوارزميات التي اقترحناها لدوبلر العالي. لنقلل تشويش دخول الناقل نتيجة لموازن الناقل الترددي, فلقد صممنا ثلاث صنابير للمتكيف في المجال الترددي لمعادل ردود الفعل الخلفية لنظام احادي المدخل احادي المخرج و نظام صندوق الترميز الفراغي الترددي لناقل الأحادي في المجال الترددي متعدد الوصول من خلال استغلال النطاقات و البنية المتناثرة في القناة. و وضحنا أن نظام ثلاثي الصنابير المتكيف في المجال الترددي لمعادل ردود الفعل الخلفية يمتلك أداء ممتاز عند مقارنته بنفس النظام لكن أحادي الصنبور ولكن بزيادة في التعقيدات المحوسبة. أيضا, وضحنا أنه بإمكاننا أن نقلل رموز التدريب في كل صندوق من الناقل الأحادي في المجال الترددي متعدد الوصول التي ترسل في مرحلة التدريب بدون وجود انخفاض في الأداء . الذي أيضا يقودنا الى تقليل الأعمى فوقية في النظام المتكيف في المجال الترددي لمعادل ردود الفعل الخلفية.

ثالثا, سرب الجسيمات الأمثل تستخدم في المعادل المتكيف. للقنوات المتغيرة مع الزمن, المعادلات المتكيفة تصمم عادة بالأعتماد على خوارزمية أقل معدل تربيع, و التي للأسف لديها قصور في بطئ الإلتقاء خصوصا في القنوات التي تكون فيها القيمة الذاتية متناثرة. مشكلة القيمة الذاتية أصبحت أكثر وضوحا في القنوات متعدد الوصول متعدد الخروج. سرب الجسيمات الأمثل يتمتع بسرعة في الاتقاء و لذلك تطبيقاته في معادل ردود الفعل الخلفية المختارة ذو تحقيقات مميزة. في هذا العمل أظهرنا أن سرب الجسيمات الأمثل في معادل ردود الفعل الخلفية المختارة مع معامل الأنقباض المتغير يتفوق على أقل معدل تربيع و أقل تربيع متكرر بالأعتماد على معادل ردود الفعل الخلفية المتكررة و سرب الجسيمات الأمثل بالاعتماد على أقل خطأ. خصوصا في القنوات ذات القيمة الذاتية كثيرة التشتت. و اقترحنا أيضا خوارزمية التهجين من سرب الجسيمات الأمثل وأقل معدل تربيع و معادل ردود الفعل الخلفية المتكرر و حسنها لتتعامل مع البيانات ذات القيم المعقدة. سرب الجسيمات الأمثل لأقل معدل تربيع مع معادل ردود الفعل الخلفية المتكررة لا تتفوق فقط على سرب الجسيمات الأمثل مع معادل ردود الفعل الخلفية المتكررة في مجال الأداء ولكن تعقيدها أيضا أقل. و للمزيد من التقليل في التعقيد , قدمنا خوارزمية سرب الجسيمات الأمثل لأقل معدل تربيع لمعادل ردود الفعل الخلفية المتكرر السريعة. الفوقية في النظام تم تقليلها من خلال استنباط خوارزمية سرب الجسيمات الأمثل العمياء.

رابعا, اقتراح نظام معادل النطاق الترددي المتكيف لناقل الأحادي للمجال الترددي متعدد الوصول بأستخدام تقنية سرب الجسيمات الامثل. اقتران التكلفة المستخدم في سرب الجسيمات الأمثل تمت صياغته بالاستناد على بنية مختصة للمعادل , سواءا كان المعادل الخطي و معادل ردود الفعل الخلفية المتكررة. قوة مقترحنا في خوارزمية سرب الجسيمات تتمثل في سيناريو الدوبلر العالي. علاوة على ذلك, انه من الواضح أن الأداء تحسنأكثر عند استخدام اعادة التوزيع العشوائي. و انه من الواضح أن سرب الجسيمات الأمثل بالأعتماد على معادل المجال الترددي هو ذي كفاءة محوسبة من نظيرة في المجال الزمني . أخيرا لزيادة سرعة الألتقاء في خوارزمية أقل معدل تربيع , ينصح بستخدام معادل متكيف يستند على أقل معدل تربيع متناثر.

CHAPTER 1

INTRODUCTION

1.1 Background

In communication system, band limited channel causes Inter Symbol Interference (ISI). Due to this non ideal characteristic of the channel, signal spread in time causing interference with the neighboring signals which limits the data rate. Another cause of ISI is multipath propagation in which the transmitted signal reaches the receiver through different paths having different delays. This happens due to reflection (from buildings), refraction (through trees) and atmospheric effects. Therefore, equalization is employed at the receiver to mitigate ISI and for perfect equalization there is a need for high performance and computationally efficient equalizers.

Consider a digital communication system in Fig.1.1 where $C(z)$ and $E(z)$ represent the channel and equalizer. Furthermore, $x(k)$,$y(k)$ and $n(k)$ denote the input to the channel, input to the equalizer and additive noise, respectively, and

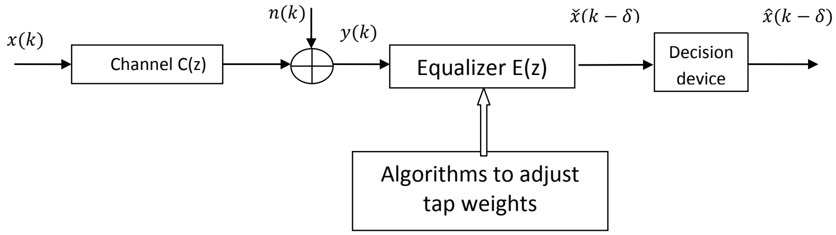


Figure 1.1: Model of a digital communication system.

$\check{x}(k)$ and $\hat{x}(k)$ are the output of the equalizer and the decision device, respectively. The decision delay δ determines which symbol is detected at current instant.

Using the linear, causal, and FIR model of the channel and the equalizer, we have

$$C(z) = \sum_{m=0}^{N_c-1} c_m z^{-m} \tag{1.1}$$

$$E(z) = \sum_{m=0}^{N_e-1} e_m z^{-m} \tag{1.2}$$

$$G(z) = C(z)E(z) = \sum_{m=0}^{N_g-1} g_m z^{-m} \tag{1.3}$$

c_m, e_m, g_m are impulse responses of the channel, equalizer, and combined channel/equalizer, respectively, whereas N_g, N_c, $N_e - 1$ are their respective lengths and $N_g = N_c + N_e - 1$. For perfect equalization $E(z) = e^{\phi j} Z^{-\delta}$, i.e., $G(z)$ is an impulse. This is the simplest criteria, called as zero forcing criteria, but suffers from noise enhancement problem. ϕ denotes the phase introduce by an equalizer.

Next, a brief overview of the equalization techniques commonly used in communication systems are presented.

1.1.1 Classification of Equalization techniques

Equalization can be classified according to the following criteria.

Coefficient adjustment

Equalizer tap weights can be fixed or adjusted adaptively. In the former case, channel state information is needed, which is estimated by using the pilot symbols whereas in the later case, training symbols are needed. There are several algorithms to be used for adaptation, e.g., Least Mean Square (LMS), Recursive Least Squares (RLS) and heuristic approaches like Particle Swarm Optimization (PSO) etc. Among these algorithms, RLS and PSO are known to have fast convergence and high complexity in contrast to LMS, which has slow convergence and less complexity. To efficiently utilize the bandwidth, pilot symbols or training sequence can be avoided by performing equalization blindly. There are different configurations/algorithms to implement blind equalization e.g. Sato, Godard, Benveniste-Goursat-Ruget (BGR), and Stop-and-go [2].

Structure

Structure of equalizer can be transversal, systolic, or Lattice. Mostly transversal equalizers, also called tapped delay line equalizers, are used for their simplicity.

Criteria for error minimization

There are mainly three criteria for minimization of error, these are, peak distortion or Zero Forcing (ZF), Mean Square Error (MSE) and Maximum Likelihood

Sequence Estimation (MLSE) criteria. MLSE is implemented using Viterbi algorithm and it is optimum but computationally complex, therefore it is used only for reference whereas ZF and MSE are suboptimum. ZF has problem of noise enhancement whereas MSE is mostly used as it takes noise into account.

Sampling

Equalizers can be symbol spaced or fractionally spaced. In symbol spaced, channel is sampled at symbol rate whereas in latter higher sampling rate is used which depends on the pulse shaping [3]. Fractionally spaced equalizer has several advantages over symbol spaced equalizer. First, sampling at higher rates allows timing recovery. Second, its performance is satisfactory in channels with deep nulls. Third, perfect equalization is possible for Finite Impulse Response (FIR) channels. Despite of these advantages, there are several disadvantages, e.g., not all channels can be equalized [4], higher power consumption due to sampling at higher rates, increased computational complexity.

Linearity

If a non-linear device is used within the equalizer then it is said to be a non-linear equalizer and vice versa. A well known non-linear equalizer is Decision Feedback Equalizer (DFE) which uses two filters, namely, Feedforward (FF) and feedback (FB). In this equalizer the previous detected symbols are fed back to cancel out the ISI caused by these symbols to the present symbol. The problem with DFE is the error propagation i.e. if the decisions on past symbol are incorrect then it will reflect in the several symbols to follow. Longer the feedback filter length, the longer these errors will retain in the equalizer and performance of equalizer lowers, particularly, at low SNR [5]. Despite of the error propagation, the DFE is used in most of the practical systems since it out performs the linear equalizer.

Time/frequency domain

Equalization can be performed in the frequency domain. The frequency domain equalizer is computationally more efficient as compared to the time-domain equalizer. The frequency-domain equalizer finds its application in Orthogonal Frequency Domain Multiple Access (OFDMA) and Single Carrier Frequency Division Multiple Access (SC-FDMA) systems, which are extensively used in the uplink for multiuser access scenarios such as the Long Term Evolution (LTE) standard [6].

Next we compared different equalizers. The symbol spaced and fractionally space channel for comparison are shown in Fig.1.2 and Fig.1.3 respectively.

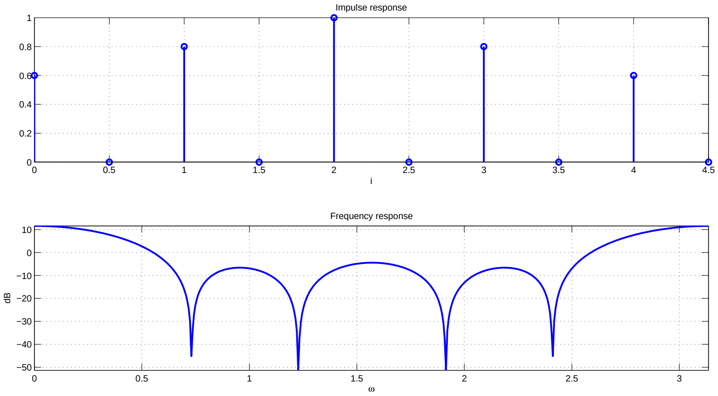


Figure 1.2: Symbol spaced Channel.

As shown in Fig.1.4 and Fig.1.5, MLSE gives the optimum performance but

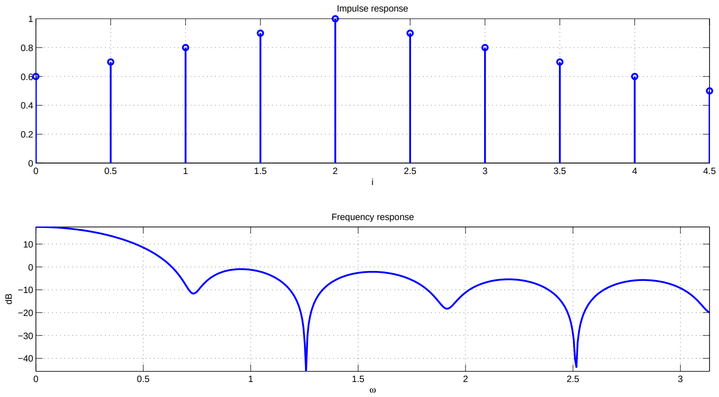


Figure 1.3: Fractionally spaced Channel.

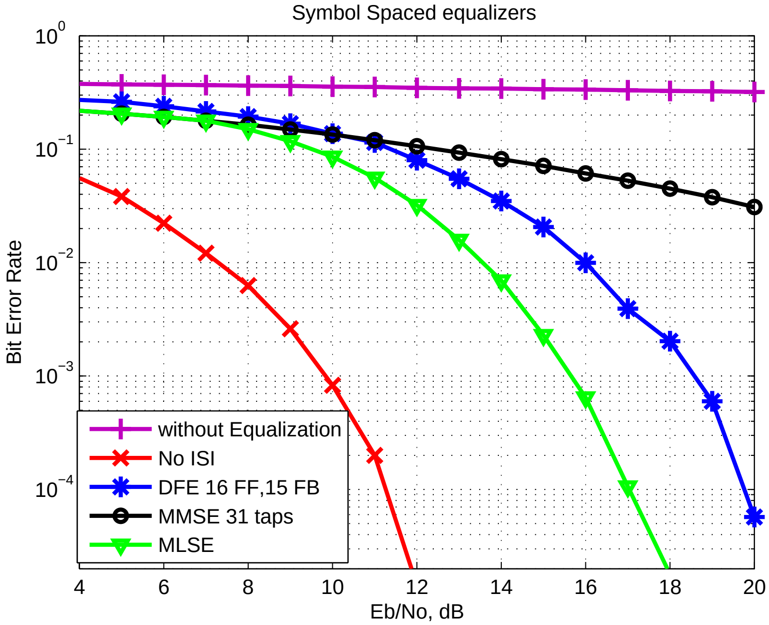


Figure 1.4: Comparison of symbol spaced equalizers .

among the sub-optimal equalizers, the DFE outperforms the others. Furthermore, performance of the fractionally space equalizer is better than its symbol spaced counter part.

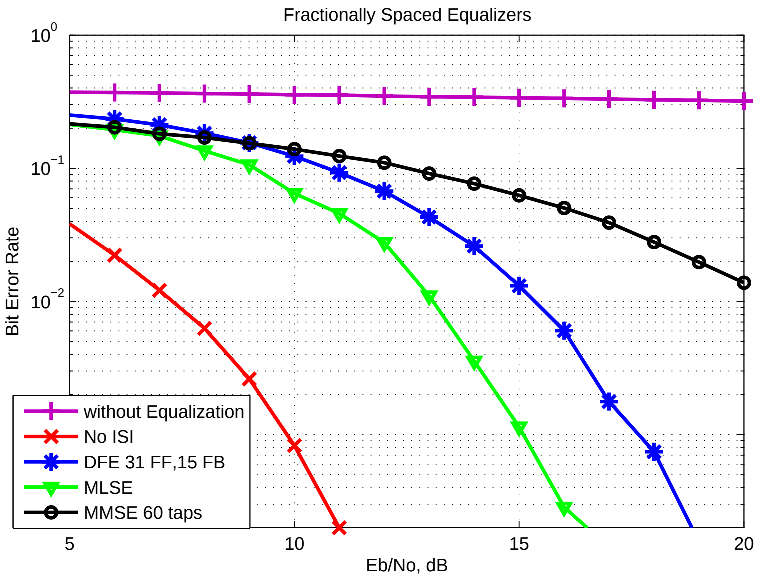


Figure 1.5: Comparison of fractionally spaced equalizers.

1.2 Dissertation Contributions and Organization

This dissertation is about proposing new ideas for fast algorithms which are to be used for adaptive equalization. By "Fast algorithms", we mean that the algorithms have fast convergence and processing (less number of multiplications/additions).

It is worth highlighting that the contributions of this thesis are somewhat scattered, tackling various issues related to adaptive equalization in time and frequency-domain. In particular, we have tried to eliminate the weaknesses of various adaptive equalization algorithms, namely, LMS, RLS and PSO, while keeping their strengths. Therefore, instead of endeavoring to put forth a common data model and a comprehensive literature review covering all the algorithms at once, each chapter of this thesis is made self contained in a sense that it includes its own motivation of the problem, literature review and contributions. The author is well aware of the fact that some material is repeated in different chapters, which was necessary in order to eliminate the need of shuffling the chapters over and over again. By doing all this, not only flow of the thoughts are maintained but it also facilitates the individual reading. The only exception is that the chapters 2, 3, and 5 uses a common data model, therefore, to reduce the redundancy, the data model is presented only once in chapter 2.

In the ensuring, the main contributions of this work are summarized:

- An Adaptive Frequency Domain Decision Feedback Equalization (AFD-DFE) is designed for a SC-FDMA system with frequency-domain feedforward and feedback filters in Chapter 2. The block RLS algorithm is used to update both the feedforward and feedback filters as it is known for its fast tracking/convergence properties. Here, we have shown that due to the special structure in the matrices involved, the proposed algorithm has a low computational complexity. Moreover, the design is extended to Multiple Input Multiple Output (MIMO) SC-FDMA systems.

- An AFD-DFE is formulated using a constraint-based RLS for SISO and MIMO SC-FDMA systems in Chapter 3. In frequency-domain, the channel matrix is diagonal and we use 1-tap equalizer per subcarrier. However, due to the carrier frequency offset, the diagonal structure of the channel matrix is lost, and it contains most of the energy along its three main diagonals. Therefore, by assuming banded and sparse structure of the channel matrix, we have extended our design to 3 taps per subcarrier AFD-DFE. We have also shown that the reduction of the overhead can be achieved by using less training symbols during the training phase and introduced a Blind AFD-DFE.

- A PSO algorithm is proposed for a DFE structure in Chapter 4. To improve the performance of the PSO algorithm, a hybrid PSO-LMS-DFE algorithm is proposed and modified to deal with complex-valued data. The complexity of the PSO algorithm is reduce by proposing a fast PSO-LMS algorithm. To further reduce the system overhead a Blind PSO algorithm is also presented.

- A frequency domain PSO algorithm is proposed and formulated based on the respective structure of the equalizer, whether it is a LE or a DFE in Chapter 5.

- The convergence speed of the LMS depends on the number of the equalizer taps. Lesser the taps, fast is the convergence. Since the impulse response of the equalizer is sparse. Utilizing this property , a sparse LMS algorithm is proposed in Chapter 6 to increase the convergence speed of the LMS algorithm.

- Finally, Chapter 7 concludes the findings of this research and considers new directions and future recommendations for the topic under observation.

CHAPTER 2

ADAPTIVE FREQUENCY-DOMAIN RLS DFE FOR UPLINK MIMO SC-FDMA

It is well known that in the case of highly frequency-selective fading channels, the Linear Equalizer (LE) can suffer significant performance degradation compared to the Decision Feedback Equalizer (DFE). In this work, we develop a low-complexity Adaptive Frequency Domain Decision Feedback Equalizer (AFD-DFE) for Single Carrier Frequency Division Multiple Access (SC-FDMA) systems, where both the feedforward and feedback filters operate in the frequency-domain and are adapted using the well-known block Recursive Least Squares (RLS) algorithm. Since this DFE design operation is performed entirely in the frequency-domain, the complexity of the block RLS algorithm can be reduced substantially when compared to its time-domain counterpart by exploiting matrix structure in the frequency-domain. We extend our formulation to Multiple Input Multiple Output (MIMO) SC-FDMA systems where we show that the AFD-DFE enjoys a significant reduction in computational complexity when compared to the frequency-domain non-adaptive DFE. Finally, extensive simulations are carried out to demonstrate the robustness of our proposed AFD-DFE to high Doppler and Carrier Frequency Offset (CFO).

2.1 Introduction

Orthogonal Frequency Domain Multiple Access (OFDMA) is a multiple access technique which has been adopted in many standards like European Telecommunications Standards Institute Digital Video Broadcast– Return Channel Terrestrial (ETSI DVB-RCT) [7], [8], WiFi [9] and WiMAX [10]. On the other hand, SC-FDMA is extensively used in the uplink for multiuser access scenarios such as the Long Term Evolution (LTE) standard [6]. SC-FDMA has an advantage over OFDMA of having a low Peak Average Power Ratio (PAPR), which helps in reducing cost and power consumption in mobile terminals. SC-FDMA is the multiple access form of Single Carrier Frequency Domain Equalization (SC-FDE) having comparable complexity and performance to that of OFDMA [11]. The sensitivity comparison of OFDMA and SC-FDMA to Carrier Frequency Offset (CFO) and Doppler effect has been reported in [12] where it was shown that for large CFO, the performance of SC-FDMA can get worse than OFDMA.

In SC-FDMA, equalization is performed after transforming the received signal from the time-domain to frequency-domain using the Discrete Fourier Transform (DFT). A linear equalizer can be adopted in SC-FDMA but in case of severe frequency-selective fading channels, in which spectral nulls are present, this will not be effective as the noise in these spectral nulls will be amplified causing significant performance degradation. Therefore, a Decision Feedback Equalizer (DFE) is a more attractive solution in these scenarios.

There are different diversity techniques to combat multi-path fading effectively. The most popular transmit diversity scheme with two antennas was proposed by Alamouti [13]. Although the overall throughput is not increased but this simple technique has certain interesting features which makes it attractive for implementation. It is an open loop technique, i.e., it does not require Channel State Information (CSI) at the transmitter, encoding and decoding is simple and due to linear processing at the terminals, complexity is reduced. It achieves full spatial diversity gain at rate 1 (i.e., full rate, as two symbol are transmitted in two time slots) and failure of one antenna chain does not result in data loss, so it is compatible with single antenna systems.

The scheme proposed by Alamouti is a special case of Space Time Block Codes (STBC) [14]. Alamouti's STBC can be applied to the 3GPP LTE uplink over two SC-FDMA symbols and two transmit antennas. Due to single carrier transmission from both antennas, the low PAPR property of SC-FDMA is not affected by signs and/or complex-conjugate operation. Unfortunately, we cannot use STBC in SC-FDMA due to the following reason. For STBC, the coding is done in pairs, therefore all the frames in SC-FDMA should contain an even number of SC-FDMA symbols but this is not guaranteed in LTE system (this is called the orphan symbol problem). In many cases, the frames contain an odd number of SC-FDMA symbols. Moreover, in STBC it is assumed that channel remains constant for two SC-FDMA symbols. This is not valid in case of fast varying channels and performance degradation will result.

An alternative scheme to STBC is Space-Frequency Block Codes (SFBC) [15]. SFBC is applied within each M-point DFT block. Therefore, it does not suffer from the problem faced by STBC when used in SC-FDMA and the number of symbols in each frame is not required to be even. As SFBC is applied to each SC-FDMA block, the size of the block should be even which is guaranteed in LTE systems. One major drawback of SFBC is that the low PAPR property is affected due to frequency inversions between successive sub-carriers [16]. Therefore, a scheme is proposed in [17] in order to tackle the orphan symbol problem. It uses spatial repetition on odd symbols and applies STBC on even symbols. This scheme cannot achieve full diversity and its performance degrades at high Signal to Noise Ratios (SNR). In [18], another approach is presented to avoid the orphan symbol problem by applying the STBC before the M-point DFT but in some channels its performance is not satisfactory especially at high SNR. In this work we will use conventional SFBC.

Several works studied the frequency-domain DFE. In [19–21], a hybrid DFE structure is used for SC-FDE systems to perform equalization in the frequency-domain. In hybrid DFE, the feedforward filter is implemented in the frequency-domain while the feedback filter is realized in the time-domain. Due to frequency-domain implementation of the feedforward filter there is a considerable reduction in complexity in a hybrid DFE as compared to its time-domain counterpart. In [22], a hybrid DFE is developed for SC-FDMA to be applied for each user separately. In [23], both the feedforward and feedback filters are implemented in the frequency-domain and an iterative procedure is used to solve the causality problem. The complexity of DFE with both filters implemented in the frequency-domain is greatly reduced as compared to the hybrid DFE. In [1] a framework is proposed to find the optimum weights of the frequency-domain DFE for SC-FDMA systems and it was shown that its performance is better than Frequency Domain LE (FD-LE) in frequency selective channels. All of the above mentioned DFE structures are non-adaptive and require CSI at the receiver.

As equalization in SC-FDMA requires Channel State Information (CSI), which is estimated from pilot symbols inserted in each block of data and optimum equalization is used to remove channel effects. Alternatively, an adaptive equalizer can be used which does not rely on CSI estimation and, hence, the channel estimation overhead will be reduced. Moreover, an adaptive equalizer can also help in tracking time varying channels. An adaptive LE for Space Time Block Coded (STBC) SC-FDE system is developed and extended for multiuser scenarios in [24].

In this work, an Adaptive Frequency Domain Decision Feedback Equalization (AFD-DFE) is designed for a SC-FDMA system. Designing a adaptive hybrid DFE will not be an easy task and, moreover, it will be computationally prohibitive due to the involvement of both time and frequency-domain signals. Therefore, we propose an adaptive DFE algorithm with frequency-domain feedforward and feedback filters. The block RLS algorithm [5] is used to update both the feedforward and feedback filters as it is known for its fast tracking/convergence properties, however, it is computationally complex due to the required matrix inversion operation. Interestingly, we will show that due to the special structure in the matrices involved, this algorithm has a low computational complexity. Moreover, the design is extended to SFBC SC-FDMA, Spatially-Multiplexed (SM) SC-FDMA and hybrid SM-SFBC SC-FDMA systems. The AFD-DFE in SC-FDMA is complicated when compared to the AFD-LE due to the iterative procedure but since SC-FDMA transmission is confined to the uplink, where processing is done at the Base Station (BS), the computationally heavy equalizer will be used at the BS (where the power limitations are more relaxed than at the user terminals). Our AFD-DFE formulation is general and does not depend on the sub-carrier mapping techniques in SC-FDMA. Furthermore, we show that AFD-DFE is also computationally efficient when compared to the channel-estimate-based DFE [1]. Finally, we demonstrate that AFD-DFE has superior performance to the LE and LMS-based approaches and it is robust to Inter-Carrier Interference (ICI) caused by high Doppler and Carrier Frequency Offset (CFO).

In summary, the main contributions of the work are

- Adaptive RLS-based implementation of the AFD-DFE for SISO SC-FDMA.
- Extension to the MIMO (both SFBC and SM) and multi-user scenarios.
- Reduced-Complexity implementation by exploiting matrix structure.
- Demonstration of performance superiority to LE and LMS-based approaches.
- Reduced Complexity compared to the channel-estimate-based approach.
- Demonstrated robustness to CFO and Doppler.

2.2 System Description

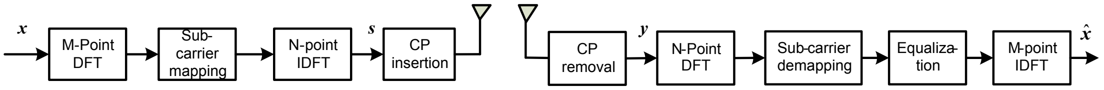


Figure 2.1: Block diagram of a transceiver of SC-FDMA system.

In this section, the SC-FDMA transceiver (see Fig. 2.1) is described for two different sub-carrier mapping methods, namely, localized and interleaved. We assume K users and a total of N sub-carriers with M sub-carriers for each user, i.e., $N = KM$. For the m^{th} user, M data symbols are grouped to form a block $\boldsymbol{x}^{(m)}$, i.e., $\boldsymbol{x}^{(m)} = [x(0)^{(m)}, x(1)^{(m)}, ..., x(M-1)^{(m)}]^T$, where T denotes the transpose operation. An M-point DFT is applied to transform $\boldsymbol{x}^{(m)}$ to the frequency-domain symbol, $\boldsymbol{\mathcal{X}}^{(m)} = [X(0)^{(m)}, X(1)^{(m)}, ..., X(M-1)^{(m)}]^T$. Next, $\boldsymbol{\mathcal{X}}^{(m)}$ is mapped to

N sub-carriers according to different mapping techniques, i.e.,

$$\boldsymbol{S}^{(m)} = \boldsymbol{R}^{(m)}\boldsymbol{\mathcal{X}}^{(m)} \quad m = 1, 2, ..., K \tag{2.1}$$

where $\boldsymbol{R}^{(m)}$ is the $N \times M$ resource allocation matrix. For example, consider the following two choices for $\boldsymbol{R}^{(m)}$

$$\boldsymbol{R}_1^{(m)} = [\mathbf{0}_{M\times(m-1)M}\ \boldsymbol{I}_M\ \mathbf{0}_{M\times(K-m)M}]^T \tag{2.2}$$

$$\boldsymbol{R}_2^{(m)} = \begin{bmatrix} \mathbf{0}^T_{M\times(m-1)} \\ \left[I_1\ \mathbf{0}_{M\times(K-1)}\ I_2\ \mathbf{0}_{M\times(K-1)}\ \ldots\quad I_{M-1}\ \mathbf{0}_{M\times(K-1)}\ I_M\right]^T \\ \mathbf{0}^T_{M\times(K-m)} \end{bmatrix} \tag{2.3}$$

which correspond to localized and interleaved allocations for the m^{th} user, respectively. $\boldsymbol{I}_M$ is $M \times M$ identity matrix with columns $I_1, I_2, ..., I_M$ and $\mathbf{0}_{M\times M}$ is the $M \times M$ all-zero matrix. Note that $\boldsymbol{R}^{(m)}$ is orthogonal for different users, that is

$$\boldsymbol{R}^{(m)T}\boldsymbol{R}^{(l)} = \begin{cases} \boldsymbol{I}_M, & m = l \\ \mathbf{0}_{M\times M}, & m \neq l \end{cases}$$

Then, the block $\boldsymbol{S}^{(m)}$ is transformed to the time-domain, $\boldsymbol{s}^{(m)}$, by applying an N-point inverse DFT (IDFT)

$$\boldsymbol{s}^{(m)} \;=\; \boldsymbol{F}_N^H\boldsymbol{R}^{(m)}\boldsymbol{F}_M\boldsymbol{x}^{(m)} = \boldsymbol{F}_N^H\boldsymbol{R}^{(m)}\boldsymbol{\mathcal{X}}^{(m)} \tag{2.4}$$

where $\boldsymbol{F}_N$ is an $N \times N$ DFT matrix and H denotes the Hermitian (i.e. complex-conjugate transpose) operation. Then, a cyclic prefix of length L_{cp} is inserted where cyclic prefix matrix is defined as

$$\boldsymbol{C}_1 = \begin{bmatrix} \boldsymbol{0}_{L_{cp}\times(N-L_{cp})} & \boldsymbol{I}_{L_{cp}} \\ \multicolumn{2}{c}{\boldsymbol{I}_N} \end{bmatrix} \tag{2.5}$$

The transmitted signal is formed after converting the signal from parallel to serial format, i.e., $\boldsymbol{s}_{cp}^{(m)} = \boldsymbol{C}_1\boldsymbol{s}^{(m)}$. We denote the impulse response of the channel for the m^{th} user by $\boldsymbol{h}^{(m)} = [h_0^{(m)}, h_1^{(m)}, ..., h_{L(m)}^{(m)}]$. For elimination of ISI using cyclic prefix, $L(m)$ should be shorter than L_{cp}. At the receiver, first the cyclic prefix is removed, i.e., $\boldsymbol{y} = \boldsymbol{C}_2\boldsymbol{y}_{cp}$ where $\boldsymbol{C}_2 = [\boldsymbol{0}_{N\times L_{cp}} \quad \boldsymbol{I}_N]$. The cyclic prefix insertion at the transmitter and removal at the receiver is equivalent to circular convolution between the transmitted signal and the channel vectors. The received signal before applying the N-point DFT can be expressed as

$$\boldsymbol{y} = \sum_{m=1}^{K} \boldsymbol{s}^{(m)} \circledast \boldsymbol{h}^{(m)} + \boldsymbol{n}^{(m)} \tag{2.6}$$

where $\circledast$ denotes the N-point circular convolution operation and $\boldsymbol{n}^{(m)}$ is the noise vector. After applying an N-point DFT the received signal, (2.6) can be expressed as

$$\boldsymbol{\mathcal{Y}} = \sum_{m=1}^{K} \hat{\boldsymbol{\Lambda}}^{(m)} \boldsymbol{R}^{(m)} \boldsymbol{\mathcal{X}}^{(m)} + \boldsymbol{\mathcal{N}} \tag{2.7}$$

where $\hat{\boldsymbol{\Lambda}}^{(m)}$ is a $N \times N$ diagonal matrix containing the DFT of $\boldsymbol{h}^{(m)}$ as diagonal elements, $\boldsymbol{\mathcal{N}}$ is noise vector due to $\sum_{m=1}^{K} \boldsymbol{n}^{(m)}$ with variance $\sigma_{\mathcal{N}}^2 \boldsymbol{I}_N$. After demapping, the m^{th} user's received signal can be expressed as

$$\boldsymbol{\mathcal{Y}}^{(m)} = \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}^{(m)} \boldsymbol{R}^{(m)} \boldsymbol{\mathcal{X}}^{(m)} + \boldsymbol{\mathcal{N}}^{(m)} \tag{2.8}$$

Let $\boldsymbol{\Lambda}^{(m)} = \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}^{(m)} \boldsymbol{R}^{(m)}$, then $\boldsymbol{\Lambda}^{(m)}$ is $M \times M$ diagonal matrix. To simplify the notation, we will ignore the superscript m, without loss of generality, then (2.8) becomes

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\Lambda} \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} \tag{2.9}$$

2.2.1 Carrier Frequency Offset (CFO)

In the above description, perfect frequency synchronization has been assumed between the transmitter and the receiver. However, CFO arises in practical SC-FDMA systems due to transmitter/receiver frequency oscillators' misalignment and causes interference (energy leakage) from neighboring sub-carriers. Therefore, for localized allocation, CFO results in intra-user interference while for interleaved subcarrier allocation, CFO results in inter-user (i.e. multi-user) interference.

Let the m^{th} user's CFO normalized by the sub-carrier spacing, be denoted by Ω_m where $0 \leq \Omega_m \leq 0.5$. Now, define a diagonal matrix to characterize the effect of CFO as $\boldsymbol{C}^{(m)} = diag([e^{j2\pi\Omega_m \times 0/N} \; e^{j2\pi\Omega_m \times 1/N} \; ... \; e^{j2\pi\Omega_m \times (N-1)/N}])$. In this case, the pre-DFT received signal can be expressed as

$$\boldsymbol{y} = \sum_{m=1}^{K} \boldsymbol{C}^{(m)} (\boldsymbol{s}^{(m)} \circledast \boldsymbol{h}^{(m)}) + \boldsymbol{n}^{(m)} \tag{2.10}$$

After applying the N-point DFT, the received signal is given by

$$\acute{\boldsymbol{\mathcal{Y}}} = \sum_{m=1}^{K} \boldsymbol{\mathcal{C}}^{(m)} \hat{\boldsymbol{\Lambda}}^{(m)} \boldsymbol{R}^{(m)} \boldsymbol{\mathcal{X}}^{(m)} + \boldsymbol{\mathcal{N}} \tag{2.11}$$

where $\boldsymbol{\mathcal{C}}^{(m)}$ is a circulant matrix with entries $\boldsymbol{\mathcal{C}}^{(m)}_{p,q} = \frac{1}{N}\sum_{n=0}^{N-1} e^{j2\pi(\Omega^{(m)}+p-q)n/N},\ p,q = 1,...,N.$

2.2.2 Fading and Doppler Effect

Another cause of ICI is the Doppler effect, which occurs due to the motion of the user. This motion causes the channel $\boldsymbol{h}^{(m)}$ to be time variant with each tap $h_i^{(m)},\ i = 1,...,L(m)$ being modeled as Wide Sense Stationary (WSS) narrow-band complex Gaussian process, independent of the other taps. Hence, the amplitude $|h_i^{(m)}|$ is assumed to be Rayleigh distributed, i.e.,

$$P(|h_i^{(m)}|) = |h_i^{(m)}| e^{-|h_i^{(m)}|^2/2}, \quad |h_i^{(m)}| \geq 0 \tag{2.12}$$

and its phase is uniformly distributed, i.e.,

$$P(\angle h_i^{(m)}) = \frac{1}{2\pi}, \quad -\pi \leq \angle h_i^{(m)} \leq \pi \tag{2.13}$$

The autocorrelation function of the $h_i^{(m)}$ is modeled as zeroth-order Bessel function of the first kind, i.e.,

$$\mathcal{R}(n) \triangleq \mathcal{J}_o(2\pi f_d t_s n), \quad n = ..., -1, 0, 1... \tag{2.14}$$

where f_d is the maximum Doppler frequency, t_s is the sampling time and $\mathcal{J}_o$ is defined as

$$\mathcal{J}_o(z) \triangleq \frac{1}{\pi}\int_0^{\pi} \cos(z\sin\theta)d\theta \tag{2.15}$$

The Doppler frequency f_d is related to the user speed v and carrier frequency f_c as $f_d = vf_c/c$, where c is the speed of light. The Rayleigh fading channel is generated using the famous Jake's model.

2.3 AFD-DFE FOR SISO SC-FDMA

Let $\boldsymbol{\mathcal{Z}} = diag(\boldsymbol{\mathcal{Y}})$ and denote the feedforward and feedback filter coefficients in the frequency-domain by $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{B}}$, respectively. The output of the equalizer in frequency-domain at instant k is given by

$$\check{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{Z}}_k\boldsymbol{\mathcal{F}}_{k-1} + \boldsymbol{\mathcal{D}}_k\boldsymbol{\mathcal{B}}_{k-1} \tag{2.16}$$

The explicit knowledge of the filter coefficients is not needed for the development of the adaptive solution. The decision matrix $\boldsymbol{\mathcal{D}}_k$ is defined as follows

$$\boldsymbol{\mathcal{D}}_k = \begin{cases} diag(F_M\boldsymbol{x}_k), & \text{for training mode} \\ \\ diag(F_M\hat{\boldsymbol{x}}_k), & \text{for decision-directed mode} \end{cases}$$

Denoting $\boldsymbol{\mathcal{W}}_k = \left[\begin{smallmatrix}\boldsymbol{\mathcal{F}}_k \\ \boldsymbol{\mathcal{B}}_k\end{smallmatrix}\right]$. We express (2.16) as

$$\check{\boldsymbol{\mathcal{X}}}_k = [\boldsymbol{\mathcal{Z}}_k \quad \boldsymbol{\mathcal{D}}_k]\boldsymbol{\mathcal{W}}_{k-1} \tag{2.17}$$

Hence, the output of the equalizer in the time-domain is $\check{\boldsymbol{x}}_k = F_M^H \check{\boldsymbol{\mathcal{X}}}_k$.

2.3.1 RLS Update

We motivate the RLS algorithm as a stochastic gradient method. Let $\boldsymbol{D} = [D(0)\ D(1)\ \ldots\ D(M-1)]$ be the vector containing the diagonal elements of $\boldsymbol{\mathcal{D}}$. The Mean Square Error (MSE) at the i^{th} frequency bin is given as

$$MSE(i) = E|D(i) - \check{X}(i)|^2 \tag{2.18}$$

where $E[.]$ stands for the expectation operation. Minimizing (2.18) for the feed-forward filter and the feedback filter separately, results in the following updates

$$\begin{aligned} \mathcal{F}_{k+1}(i) = \mathcal{F}_k(i) \quad &+ \quad \left(\frac{\mu_{k+1}}{\epsilon_{k+1} + E[\mathcal{Y}(i)^*\mathcal{Y}(i)]}\right) \mathcal{Y}^*_{k+1}(i)\{D_{k+1}(i) \\ &- \quad [\mathcal{Y}_{k+1}(i)\mathcal{F}_k(i) + D_{k+1}(i)\mathcal{B}_k(i)]\} \end{aligned} \tag{2.19}$$

and

$$\begin{aligned} \mathcal{B}_{k+1}(i) = \mathcal{B}_k(i) \quad &+ \quad \left(\frac{\mu_{k+1}}{\epsilon_{k+1} + E[D^*(i)D(i)]}\right) D^*_{k+1}(i)\{D_{k+1}(i) \\ &- \quad [\mathcal{Y}_{k+1}(i)\mathcal{F}_k(i) + D_{k+1}(i)\mathcal{B}_k(i)]\} \end{aligned} \tag{2.20}$$

Next, we replace $E[\mathcal{Y}(i)^*\mathcal{Y}(i)]$ and $E[D^*(i)D(i)]$ by their estimates, which for the RLS update, are chosen to be the exponentially-weighted sample averages for some scalar $0 \ll \lambda \leq 1$. Choosing the step size as $\mu_{k+1} = 1/(k+2)$, the regularization factor as $\epsilon_{k+1} = \lambda^{k+2}\epsilon/(k+2)$, and collecting all the coefficients in one vector $\boldsymbol{\mathcal{W}}$, Equations (2.19) and (2.20) become

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + \left[\lambda^{k+2}\epsilon\boldsymbol{I}_{2M} + \sum_{j=0}^{k+1}\lambda^{k+1-j}\boldsymbol{\mathcal{A}}_j^H\boldsymbol{\mathcal{A}}_j\right]^{-1}\boldsymbol{\mathcal{A}}_{k+1}^H\boldsymbol{\mathcal{E}}_{k+1} \tag{2.21}$$

where $\boldsymbol{\mathcal{A}}_k$ and $\boldsymbol{\mathcal{E}}_k$ are given as

$$\boldsymbol{\mathcal{A}}_k = \begin{bmatrix} \boldsymbol{\mathcal{Z}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{D}}_k \end{bmatrix} \tag{2.22}$$

and

$$\boldsymbol{\mathcal{E}}_k = \begin{bmatrix} \boldsymbol{D}_k - \check{\boldsymbol{\mathcal{X}}}_k \\ \boldsymbol{D}_k - \check{\boldsymbol{\mathcal{X}}}_k \end{bmatrix} \tag{2.23}$$

It is not convenient to find the inverse of the matrix in (2.21) as it requires us to combine all the previous and present data to form the matrix. Therefore, we define

$$\Theta_{k+1} \triangleq \left(\lambda^{k+2}\epsilon\boldsymbol{I}_{2M} + \sum_{j=0}^{k+1}\lambda^{k+1-j}\boldsymbol{\mathcal{A}}_j^H\boldsymbol{\mathcal{A}}_j\right) \tag{2.24}$$

which satisfies the following recursion

$$\Theta_{k+1} = \lambda\Theta_k + \boldsymbol{\mathcal{A}}_{k+1}^H\boldsymbol{\mathcal{A}}_{k+1}, \quad \Theta_1 = \epsilon\boldsymbol{I}_{2M} \tag{2.25}$$

Let $\boldsymbol{P}_{k+1} = \Theta_{k+1}^{-1}$ and applying the matrix inversion lemma [5] gives

$$\boldsymbol{\mathcal{P}}_{k+1} = \lambda^{-1}[\boldsymbol{\mathcal{P}}_k - \lambda^{-1}\boldsymbol{\mathcal{P}}_k\boldsymbol{\mathcal{A}}_{k+1}^H \times (\boldsymbol{I}_{2M} + \lambda^{-1}\boldsymbol{\mathcal{A}}_{k+1}\boldsymbol{\mathcal{P}}_k\boldsymbol{\mathcal{A}}_{k+1}^H)^{-1}\boldsymbol{\mathcal{A}}_{k+1}\boldsymbol{\mathcal{P}}_k] \tag{2.26}$$

where λ is the forgetting factor chosen close to 1. Finally, the RLS update is given as

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + \boldsymbol{\mathcal{P}}_{k+1}\boldsymbol{\mathcal{A}}_{k+1}^H\boldsymbol{\mathcal{E}}_{k+1} \tag{2.27}$$

with $\boldsymbol{\mathcal{E}}_k$ defined as in (2.23). Initially $\boldsymbol{\mathcal{W}}_0 = \mathbf{0}$ and $\boldsymbol{\mathcal{P}}_0 = diag(\epsilon_{\mathcal{F}}^{-1}\boldsymbol{I}_M \ \epsilon_{\mathcal{B}}^{-1}\boldsymbol{I}_M)$.

2.3.2 Reduced-Complexity RLS Update

Due to the special structure in $\boldsymbol{\mathcal{P}}_{k+1}$, it turns out that no matrix inversion is required for computing $\boldsymbol{\mathcal{P}}_{k+1}$ resulting in a significant reduction in computational complexity.

We start by noting that the matrix $\boldsymbol{\mathcal{P}}_{k+1}$ has a diagonal structure, i.e., $\boldsymbol{\mathcal{P}}_{k+1} = diag([\boldsymbol{P}_{k+1}^1 \quad \boldsymbol{P}_{k+1}^2])$, where $\boldsymbol{P}_{k+1}^1$ and $\boldsymbol{P}_{k+1}^2$ are diagonal as well. We can write

$$\begin{aligned}\boldsymbol{P}_{k+1}^1 &= \lambda^{-1}[\boldsymbol{P}_k^1 - \lambda^{-1}\boldsymbol{P}_k^1\boldsymbol{\mathcal{Z}}_{k+1}^H(\boldsymbol{I}_M + \lambda^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}_k^1\boldsymbol{\mathcal{Z}}_{k+1}^H)^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}_k^1] \\ &= \lambda^{-1}[\boldsymbol{P}_k^1 - \lambda^{-1}\boldsymbol{P}_k^1\boldsymbol{\mathcal{Z}}_{k+1}^H(\boldsymbol{I}_M + \lambda^{-1}|\boldsymbol{\mathcal{Z}}_{k+1}|^2\boldsymbol{P}_k^1)^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}_k^1] \qquad (2.28)\end{aligned}$$

Moreover, the term $(\boldsymbol{I}_M + \lambda^{-1}|\boldsymbol{\mathcal{Z}}_{k+1}|^2\boldsymbol{P}_k^1)^{-1}$ is also diagonal, therefore,

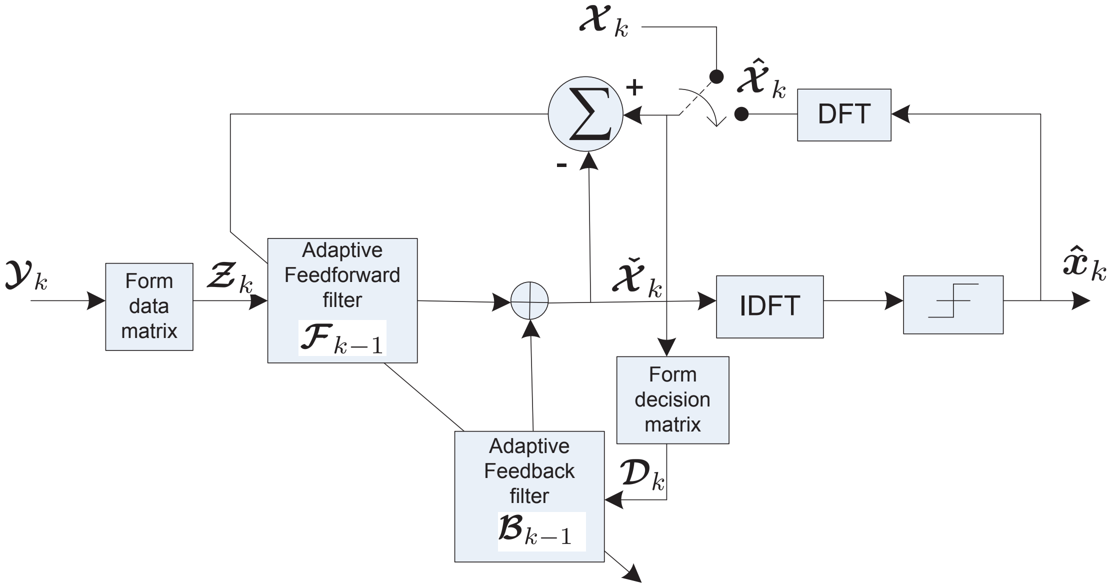
$$\boldsymbol{P}_{k+1}^1 = \lambda^{-1}[\boldsymbol{P}_k^1 - \lambda^{-1}\boldsymbol{P}_k^1(|\boldsymbol{\mathcal{Z}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}_k^1)^{-1}\boldsymbol{P}_k^1] \tag{2.29}$$

and the matrix inversion here is simply M scalar inversions. Similarly, as in the case of $\boldsymbol{P}_{k+1}^1$, it can be argued following the same steps for $\boldsymbol{P}_{k+1}^2$ that

$$\boldsymbol{P}_{k+1}^2 = \lambda^{-1}[\boldsymbol{P}_k^2 - \lambda^{-1}\boldsymbol{P}_k^2(|\boldsymbol{\mathcal{D}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}_k^2)^{-1}\boldsymbol{P}_k^2] \tag{2.30}$$

Therefore, the final reduced-complexity RLS update has the form

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + diag([\boldsymbol{P}^1_{k+1} \quad \boldsymbol{P}^2_{k+1}])\boldsymbol{\mathcal{A}}^H_{k+1}\boldsymbol{\mathcal{E}}_{k+1} \tag{2.31}$$



$\boldsymbol{\mathcal{X}}_k$
$\hat{\boldsymbol{\mathcal{X}}}_k$
$\sum$
+
-
DFT
$\boldsymbol{\mathcal{Y}}_k$
Form data matrix
$\boldsymbol{\mathcal{Z}}_k$
Adaptive Feedforward filter
$\boldsymbol{\mathcal{F}}_{k-1}$
$\check{\boldsymbol{\mathcal{X}}}_k$
IDFT
$\hat{\boldsymbol{x}}_k$
Form decision matrix
$\boldsymbol{\mathcal{D}}_k$
Adaptive Feedback filter
$\boldsymbol{\mathcal{B}}_{k-1}$

Figure 2.2: Block diagram of AFD-DFE.

The block diagram of the resulting AFD-DFE is depicted in Fig. 2.2. For a linear equalizer, the derivation is the same except that the feedback coefficients are set to zero. To summarize, the overall procedure for updating a frequency-domain equalizer is as follows. The received signal is first transformed to the frequency-domain using the DFT and the data matrix $\boldsymbol{\mathcal{A}}_k$ is formed. The equalizer output is calculated using (2.16) and transformed to the time-domain using an IDFT, then decisions are made on this data to produce the equalizer output. The error signal is generated by comparing the equalizer output with the desired response in the frequency-domain, which is used in (2.31). First, the training mode is used to allow the algorithm to converge and then it is switched to the decision-directed mode in which the equalizer uses the previous decisions to update its weights. To prevent the algorithm from diverging, in case of fast fading channels, retraining blocks can be used. The causality problem in (2.16) can be solved using an iterative procedure similar to that described in [25], [26]. Note here that we have reserved the word 'iterative' for (2.16), which is to be solved iteratively whereas the word 'recursive' is used for the RLS update (2.31). The update procedure for AFD-DFE with CRLS is shown in Table 2.1.

Table 2.1: Summary of the adaptation algorithm for AFD-DFE

Initialization:

Initialize the algorithm by setting

$\boldsymbol{\mathcal{W}}_0 = \mathbf{0}$

λ is close to one

and $\boldsymbol{\mathcal{P}}_0 = \begin{bmatrix} \epsilon_{\mathcal{F}}^{-1}\boldsymbol{I}_M & \mathbf{0} \\ \mathbf{0} & \epsilon_{\mathcal{B}}^{-1}\boldsymbol{I}_M \end{bmatrix}$

For each instant of time, k=0,1,2,...

In training mode:

(1) Update $\boldsymbol{P}^1_{k+1}$ and $\boldsymbol{P}^2_{k+1}$ via

$\boldsymbol{P}^1_{k+1} = \lambda^{-1}[\boldsymbol{P}^1_k - \lambda^{-1}\boldsymbol{P}^1_k(|\boldsymbol{\mathcal{Z}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}^1_k)^{-1}\boldsymbol{P}^1_k]$

$\boldsymbol{P}^2_{k+1} = \lambda^{-1}[\boldsymbol{P}^2_k - \lambda^{-1}\boldsymbol{P}^2_k(|\boldsymbol{\mathcal{X}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}^2_k)^{-1}\boldsymbol{P}^2_k]$

(2) Update the equalizer weights $\boldsymbol{\mathcal{W}}_{k+1}$ recursively as

$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + diag([\boldsymbol{P}^1_{k+1} \quad \boldsymbol{P}^2_{k+1}])(\boldsymbol{\mathcal{A}}^H_{k+1}\boldsymbol{\mathcal{E}}_{k+1})$

In decision-directed mode:

(1) Iterate on $\check{\boldsymbol{\mathcal{X}}}_{k+1} = \boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{\mathcal{F}}_k + \boldsymbol{\mathcal{D}}_{k+1}\boldsymbol{\mathcal{B}}_k$

(2) Update $\boldsymbol{P}^1_{k+1}$ and $\boldsymbol{P}^2_{k+1}$ via

$\boldsymbol{P}^1_{k+1} = \lambda^{-1}[\boldsymbol{P}^1_k - \lambda^{-1}\boldsymbol{P}^1_k(|\boldsymbol{\mathcal{Z}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}^1_k)^{-1}\boldsymbol{P}^1_k]$

$\boldsymbol{P}^2_{k+1} = \lambda^{-1}[\boldsymbol{P}^2_k - \lambda^{-1}\boldsymbol{P}^2_k(|\check{\boldsymbol{\mathcal{X}}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}^2_k)^{-1}\boldsymbol{P}^2_k]$

(3) Update the equalizer weights $\boldsymbol{\mathcal{W}}_{k+1}$ recursively as

$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + diag([\boldsymbol{P}^1_{k+1} \quad \boldsymbol{P}^2_{k+1}])(\boldsymbol{\mathcal{A}}^H_{k+1}\boldsymbol{\mathcal{E}}_{k+1})$

2.4 AFD-DFE FOR MIMO SC-FDMA

MIMO systems are used either to enhance the performance (spatial diversity) or to increase the rate (spatial multiplexing) or a hybrid thereof. In the following, we extend our AFD-DFE design to these MIMO systems.

2.4.1 AFD-DFE for SFBC SC-FDMA

An attractive technique for spatial diversity is Space-Frequency Block Code (SFBC). For our AFD-DFE design we implement SFBC at the block level. The block diagram of SFBC SC-FDMA is shown in Fig.(2.3). After applying the M-point DFT, the block $\boldsymbol{x}^{(m)}$ is transformed to the frequency-domain $\boldsymbol{\mathcal{X}}^{(m)}$, where $\boldsymbol{\mathcal{X}}^{(m)} = [X(0)^{(m)}, X(1)^{(m)}, ..., X(M-1)^{(m)}]^T$. After SFBC, we get $\boldsymbol{\mathcal{X}}_1^{(m)} = [X(0)^{(m)}, -X^*(1)^{(m)}, ..., X(M-2)^{(m)}, -X^*(M-1)^{(m)}]^T$ and $\boldsymbol{\mathcal{X}}_2^{(m)} = [X(1)^{(m)}, X^*(0)^{(m)}, ..., X(M-1)^{(m)}, X^*(M-2)^{(m)}]^T$, where $(\cdot)^*$ denotes the complex-conjugate operation. After mapping and applying the N-point IDFT, the transmitted signals from the two antennas are denoted by $\boldsymbol{s}_1^{(m)}$ and $\boldsymbol{s}_2^{(m)}$ corresponding to $\boldsymbol{\mathcal{X}}_1^{(m)}$ and $\boldsymbol{\mathcal{X}}_2^{(m)}$. The transmitted signals are circularly convolved with their respective channels (due to CP insertion) and the received signal be-

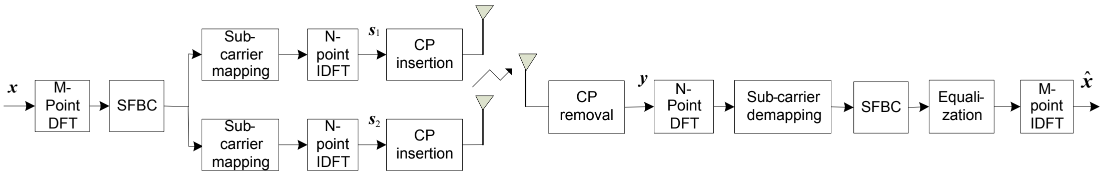


Figure 2.3: Block diagram of a transceiver of SFBC SC-FDMA system.

comes

$$\boldsymbol{y} = \sum_{m=1}^{K} [\boldsymbol{s}_1^{(m)} \circledast \boldsymbol{h}_1^{(m)} + \boldsymbol{s}_2^{(m)} \circledast \boldsymbol{h}_2^{(m)}] + \boldsymbol{n}^{(m)} \tag{2.32}$$

where $\boldsymbol{h}_i^{(m)} = [h_{i0}^{(m)}, h_{i1}^{(m)}, ..., h_{iL(m)}^{(m)}]$ for $i = 1, 2$. After applying the N-DFT of (2.32), the signal becomes

$$\hat{\boldsymbol{\mathcal{Y}}} = \sum_{m=1}^{K} [\hat{\boldsymbol{\Lambda}}_1^{(m)} \boldsymbol{R}^{(m)} \boldsymbol{\mathcal{X}}_1^{(m)} + \hat{\boldsymbol{\Lambda}}_2^{(m)} \boldsymbol{R}^{(m)} \boldsymbol{\mathcal{X}}_2^{(m)}] + \boldsymbol{\mathcal{N}} \tag{2.33}$$

where $\hat{\boldsymbol{\Lambda}}_i^{(m)}$ is a $N \times N$ diagonal matrix, i.e., $\hat{\boldsymbol{\Lambda}}_i^{(m)} = diag(DFT(\boldsymbol{h}_i^{(m)}))$ for $i = 1, 2$ and $\boldsymbol{\mathcal{N}}$ is the noise component with variance $\sigma_{\mathcal{N}} \boldsymbol{I}_N$. The received signal for the m^{th} user, after demapping, can be expressed as

$$\boldsymbol{\mathcal{Y}}^{(m)} = \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}_1^{(m)} \boldsymbol{R}^{(m)} \boldsymbol{\mathcal{X}}_1^{(m)} + \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}_2^{(m)} \boldsymbol{R}^{(m)} \boldsymbol{\mathcal{X}}_2^{(m)} + \boldsymbol{\mathcal{N}}^{(m)} \tag{2.34}$$

Let $\boldsymbol{\Lambda}_i^{(m)} = \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}_i^{(m)} \boldsymbol{R}^{(m)}$ for $i = 1, 2$, then $\boldsymbol{\Lambda}_i^{(m)}$ is $M \times M$ diagonal matrix. To simplify the notation we will drop the superscript m, thus (2.34) becomes

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\Lambda}_1 \boldsymbol{\mathcal{X}}_1 + \boldsymbol{\Lambda}_2 \boldsymbol{\mathcal{X}}_2 + \boldsymbol{\mathcal{N}} \tag{2.35}$$

Using the odd even expansion, the odd component of (2.35) is

$$\boldsymbol{\mathcal{Y}}_o = \boldsymbol{\Lambda}_{1o} \boldsymbol{\mathcal{X}}_{1o} + \boldsymbol{\Lambda}_{2o} \boldsymbol{\mathcal{X}}_{2o} + \boldsymbol{\mathcal{N}}_o \tag{2.36}$$

and the even component is

$$\boldsymbol{\mathcal{Y}}_e = \mathbf{\Lambda}_{1e}\boldsymbol{\mathcal{X}}_{1e} + \mathbf{\Lambda}_{2e}\boldsymbol{\mathcal{X}}_{2e} + \boldsymbol{\mathcal{N}}_e \tag{2.37}$$

where $\boldsymbol{\mathcal{X}}_{io}$ and $\boldsymbol{\mathcal{X}}_{ie}$ represent the odd and even components, respectively, of $\boldsymbol{\mathcal{X}}_i$ for $i = 1, 2$. Denoting the odd and the even components of $\boldsymbol{\mathcal{X}}$ as $\boldsymbol{\mathcal{X}}_o$ and $\boldsymbol{\mathcal{X}}_e$, respectively, we have

$$\boldsymbol{\mathcal{X}}_{1o} = \boldsymbol{\mathcal{X}}_o,\ \boldsymbol{\mathcal{X}}_{2o} = \boldsymbol{\mathcal{X}}_e,\ \boldsymbol{\mathcal{X}}_{1e} = -\boldsymbol{\mathcal{X}}_e^*,\ \boldsymbol{\mathcal{X}}_{2e} = \boldsymbol{\mathcal{X}}_o^* \tag{2.38}$$

Consequently, (2.36) and (2.37) become, respectively, as

$$\boldsymbol{\mathcal{Y}}_o = \mathbf{\Lambda}_{1o}\boldsymbol{\mathcal{X}}_o + \mathbf{\Lambda}_{2o}\boldsymbol{\mathcal{X}}_e + \boldsymbol{\mathcal{N}}_o \tag{2.39}$$

and

$$\boldsymbol{\mathcal{Y}}_e = -\mathbf{\Lambda}_{1e}\boldsymbol{\mathcal{X}}_e^* + \mathbf{\Lambda}_{2e}\boldsymbol{\mathcal{X}}_o^* + \boldsymbol{\mathcal{N}}_e \tag{2.40}$$

Combining (2.39) and (2.40), the linear relationship [15] can be written as

$$\begin{aligned} \boldsymbol{\mathcal{Y}}_{oe} &= \begin{bmatrix} \boldsymbol{\mathcal{Y}}_o \\ \boldsymbol{\mathcal{Y}}_e^* \end{bmatrix} = \begin{bmatrix} \mathbf{\Lambda}_{1o} & \mathbf{\Lambda}_{2o} \\ \mathbf{\Lambda}_{2e}^* & -\mathbf{\Lambda}_{1e}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{X}}_o \\ \boldsymbol{\mathcal{X}}_e \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mathcal{N}}_o \\ \boldsymbol{\mathcal{N}}_e^* \end{bmatrix} \\ &\triangleq \mathbf{\Lambda}\boldsymbol{\mathcal{X}}_{oe} + \boldsymbol{\mathcal{N}}_{oe} \end{aligned} \tag{2.41}$$

Moreover, assuming

$$\boldsymbol{\Lambda}_{ie} \;=\; \boldsymbol{\Lambda}_{io}, \quad i = 1, 2 \tag{2.42}$$

$\boldsymbol{\Lambda}$ becomes an Alamouti-like matrix. A matrix is said to be an Alamouti matrix if it has the form $\left[\begin{smallmatrix} \Delta_1 & \Delta_2 \\ \Delta_2^* & -\Delta_1^* \end{smallmatrix}\right]$. After MMSE equalization, we get

$$\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_o \\ \check{\boldsymbol{\mathcal{X}}}_e \end{bmatrix} = (\boldsymbol{\Lambda}^H \boldsymbol{\Lambda} + \frac{1}{SNR} \boldsymbol{I}_{2M})^{-1} \boldsymbol{\Lambda}^H \boldsymbol{\mathcal{Y}}_{oe} \tag{2.43}$$

where SNR is the signal-to-noise ratio at the receiver. Since $\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}^H$ has an Alamouti-like structure, therefore

$$\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_o \\ \check{\boldsymbol{\mathcal{X}}}_e \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 \\ \boldsymbol{\Phi}_2^* & -\boldsymbol{\Phi}_1^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{Y}}_o \\ \boldsymbol{\mathcal{Y}}_e^* \end{bmatrix} \tag{2.44}$$

where $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are diagonal matrices. Alternatively, (2.44) can be written as

$$\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_o \\ \check{\boldsymbol{\mathcal{X}}}_e^* \end{bmatrix} = \begin{bmatrix} diag(\boldsymbol{\mathcal{Y}}_o) & diag(\boldsymbol{\mathcal{Y}}_{\boldsymbol{e}}^*) \\ -diag(\boldsymbol{\mathcal{Y}}_e) & diag(\boldsymbol{\mathcal{Y}}_{\boldsymbol{o}}^*) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Upsilon}_1 \\ \boldsymbol{\Upsilon}_2 \end{bmatrix}$$

where $\boldsymbol{\Upsilon}_1$ and $\boldsymbol{\Upsilon}_2$ are the vectors containing the diagonal elements of $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$. For a DFE, we have

$$\begin{aligned}\underbrace{\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_o \\ \check{\boldsymbol{\mathcal{X}}}_e^* \end{bmatrix}}_{\check{\boldsymbol{\mathcal{X}}}_{oe}} &= \begin{bmatrix} diag(\boldsymbol{\mathcal{Y}}_o) & diag(\boldsymbol{\mathcal{Y}}_e^*) \\ -diag(\boldsymbol{\mathcal{Y}}_e) & diag(\boldsymbol{\mathcal{Y}}_o^*) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Upsilon}_1 \\ \boldsymbol{\Upsilon}_2 \end{bmatrix} + \begin{bmatrix} diag(\boldsymbol{D}_o) & \mathbf{0} \\ \mathbf{0} & diag(\boldsymbol{D}_e^*) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_1 \\ \boldsymbol{\Psi}_2 \end{bmatrix} \\ &\triangleq \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{F}} + \boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{B}} \end{aligned} \tag{2.45}$$

where $\boldsymbol{D}_o$ and $\boldsymbol{D}_e$ are $\boldsymbol{\mathcal{X}}_o$ and $\boldsymbol{\mathcal{X}}_e$, respectively, for the training mode or frequency-domain decisions on $\check{\boldsymbol{\mathcal{X}}}_o$ and $\check{\boldsymbol{\mathcal{X}}}_e$, respectively, for the decision-directed mode. The feedforward and feedback filter coefficients in the frequency-domain are $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{B}}$ containing the elements $\{\boldsymbol{\Upsilon}_1, \boldsymbol{\Upsilon}_2\}$ and $\{\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2\}$, respectively. Moreover, $\boldsymbol{\mathcal{Z}}$ is an $M \times M$ Alamouti-like matrix containing the received symbols and $\boldsymbol{\mathcal{D}}$ is a diagonal matrix containing the decisions. However, these coefficients will be computed adaptively; hence, an exact solution is not required. At the k^{th} instant, the output of the equalizer is given

$$\check{\boldsymbol{\mathcal{X}}}_{oe,k} = \boldsymbol{\mathcal{Z}}_k \boldsymbol{\mathcal{F}}_{k-1} + \boldsymbol{\mathcal{D}}_k \boldsymbol{\mathcal{B}}_{k-1} \tag{2.46}$$

The RLS AFD-DFE recursion is given as in (2.27) with error vector as

$$\boldsymbol{\mathcal{E}}_k = \begin{bmatrix} \boldsymbol{D}_k - \check{\boldsymbol{\mathcal{X}}}_{oe,k} \\ \boldsymbol{D}_k - \check{\boldsymbol{\mathcal{X}}}_{oe,k} \end{bmatrix} \tag{2.47}$$

where $\boldsymbol{D}_k$ denotes the decisions at the k^{th} instant, i.e., $\boldsymbol{D}_k = \begin{bmatrix} \boldsymbol{D}_{o,k} \\ \boldsymbol{D}^*_{e,k} \end{bmatrix}$ and $\boldsymbol{\mathcal{A}}_k$ is given as in(2.22).

Reduced-Complexity RLS AFD-DFE: It might seem that (2.27) requires matrix inversion. However, due to the special structure of SFBC, no inversion is required resulting in significant complexity reductions as shown below.

The matrix $\boldsymbol{\mathcal{P}}_{k+1}$ has a diagonal structure, i.e., $\boldsymbol{\mathcal{P}}_{\boldsymbol{k+1}} = diag([\boldsymbol{P}^1_{k+1} \quad \boldsymbol{P}^2_{k+1}])$, where $\boldsymbol{P}^1_{k+1}$ and $\boldsymbol{P}^2_{k+1}$ are diagonal as well and $\boldsymbol{P}^1_{k+1}$ is given by.

$$\boldsymbol{P}^1_{k+1} \;=\; \lambda^{-1}[\boldsymbol{P}^1_k - \lambda^{-1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}^H_{k+1}(\boldsymbol{I}_M + \lambda^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}^H_{k+1})^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}^1_k] \quad (2.48)$$

Now, simplifying the term $(\lambda^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}^H_{k+1})$, we get

$$\begin{aligned}
\lambda^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}^H_{k+1} &= \lambda^{-1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{\mathcal{Z}}^H_{k+1} \\
&= \lambda^{-1}\boldsymbol{P}^1_k \begin{bmatrix} diag(\boldsymbol{\mathcal{Y}}_{o,k+1}) & diag(\boldsymbol{\mathcal{Y}}^*_{e,k+1}) \\ -diag(\boldsymbol{\mathcal{Y}}_{e,k+1}) & diag(\boldsymbol{\mathcal{Y}}^*_{o,k+1}) \end{bmatrix} \\
&\quad \times \begin{bmatrix} diag(\boldsymbol{\mathcal{Y}}^*_{o,k+1}) & -diag(\boldsymbol{\mathcal{Y}}^*_{e,k+1}) \\ diag(\boldsymbol{\mathcal{Y}}_{e,k+1}) & diag(\boldsymbol{\mathcal{Y}}_{o,k+1}) \end{bmatrix} \\
&= \lambda^{-1}\boldsymbol{P}^1_k diag\left[diag(|\boldsymbol{\mathcal{Y}}_{o,k+1}|^2) + diag(|\boldsymbol{\mathcal{Y}}_{e,k+1}|^2)\right] \quad (2.49)
\end{aligned}$$

Now $\boldsymbol{\mathcal{Z}}^H_{k+1}(\boldsymbol{I_M} + \lambda^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}^H_{k+1})^{-1}\boldsymbol{\mathcal{Z}}_{k+1} = diag([\o \quad \o]) \triangleq \boldsymbol{\psi}^1_{k+1}$, where $\o$ is diagonal given as $\o = (diag(|\boldsymbol{\mathcal{Y}}_{\boldsymbol{o,k+1}}|^2) + diag(|\boldsymbol{\mathcal{Y}}_{\boldsymbol{e,k+1}}|^2))^{-1} + \lambda^{-1}\boldsymbol{P}^1_k]^{-1}$. It follows that $\boldsymbol{P}^1_{k+1}$ will have the form

$$\boldsymbol{P}^1_{k+1} \quad = \quad \lambda^{-1}[\boldsymbol{P}^1_k - \lambda^{-1}\boldsymbol{P}^1_k\boldsymbol{\psi}^1_{k+1}\boldsymbol{P}^1_k] \tag{2.50}$$

Using the same approach as in the SISO case, $\boldsymbol{P}^2_{k+1}$ can be expressed as follows

$$\boldsymbol{P}^2_{k+1} \quad = \quad \lambda^{-1}[\boldsymbol{P}^2_k - \lambda^{-1}\boldsymbol{P}^2_k\boldsymbol{\psi}^2_{k+1}\boldsymbol{P}^2_k] \tag{2.51}$$

where $\boldsymbol{\psi}^2_{k+1} = (|\boldsymbol{\mathcal{D}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}^2_k)^{-1}$. Finally, the RLS AFD-DFE recursion has the form

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + diag([\boldsymbol{P}^1_{k+1} \quad \boldsymbol{P}^2_{k+1}])\boldsymbol{\mathcal{A}}^H_{k+1}\boldsymbol{\mathcal{E}}_{k+1} \tag{2.52}$$

where $\boldsymbol{P}^2_{k+1}$ and $\boldsymbol{P}^2_{k+1}$ are defined by (2.50) and (2.51), respectively. The block diagram of the resulting AFD-DFE for SFBC SC-FDMA is depicted in Fig. 2.4.

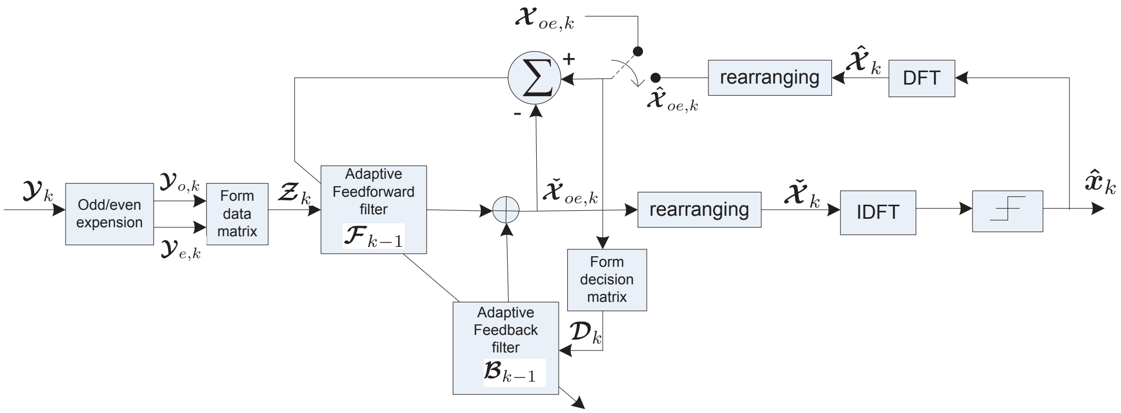

$\boldsymbol{\mathcal{X}}_{oe,k}$
+
-
rearranging
$\hat{\boldsymbol{\mathcal{X}}}_k$
DFT
$\hat{\boldsymbol{\mathcal{X}}}_{oe,k}$
$\boldsymbol{\mathcal{Y}}_k$
Odd/even expension
$\boldsymbol{\mathcal{Y}}_{o,k}$
$\boldsymbol{\mathcal{Y}}_{e,k}$
Form data matrix
$\boldsymbol{\mathcal{Z}}_k$
Adaptive Feedforward filter
$\boldsymbol{\mathcal{F}}_{k-1}$
$\check{\boldsymbol{\mathcal{X}}}_{oe,k}$
rearranging
$\check{\boldsymbol{\mathcal{X}}}_k$
IDFT
$\hat{\boldsymbol{x}}_k$
Form decision matrix
$\boldsymbol{\mathcal{D}}_k$
Adaptive Feedback filter
$\boldsymbol{\mathcal{B}}_{k-1}$

Figure 2.4: Block diagram of AFD-DFE for two-transmit one-receive antenna.

2.4.2 AFD-DFE for Spatially-Multiplexed (SM) SC-FDMA

In this section, we will use MIMO to increase the number of users supported by the system, i.e., spatial multiplexing. In a SM SC-FDMA scenario, multiple users use the same frequency and time slot to transmit data. The number of antennas at the Base Station (BS) is equal to the number of users using the same frequency and time slots. AFD-DFE is used for joint interference cancelation and equalization at the receiver. The system model for SM SC-FDMA is depicted in Fig. 2.5. Let N_t and N_r be the number of transmit and receive antennas, respectively. The transmission from each user can be viewed as a SISO SC-FDMA transmission and the received signal at the r^{th} antenna after cyclic prefix removal and before applying the N-point DFT can be expressed as

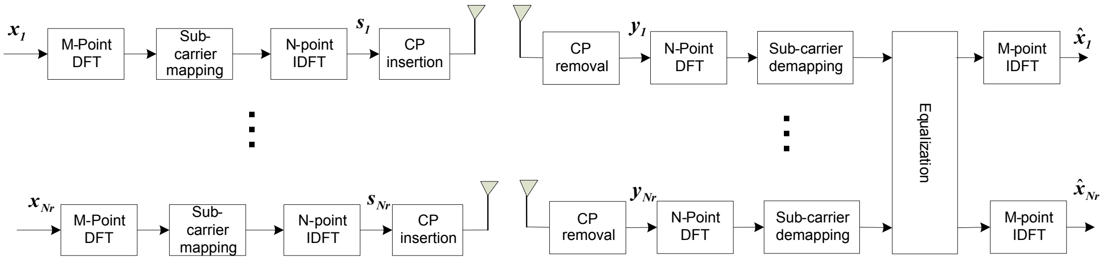


Figure 2.5: Block diagram of a transceiver of a SM SC-FDMA system.

$$\boldsymbol{y}_r = \sum_{t=1}^{N_r} \sum_{m=1}^{K} \boldsymbol{s}_t^{(m)} \circledast \boldsymbol{h}_{tr}^{(m)} + \boldsymbol{n}_r^{(m)}, \quad r = 1, 2, ..., N_r \tag{2.53}$$

where $\boldsymbol{n}_r^{(m)}$ is the noise vector at r^{th} receive antenna. After applying an N-point DFT, the received signal can be expressed as

$$\acute{\boldsymbol{\mathcal{Y}}}_r = \sum_{t=1}^{N_r} \sum_{m=1}^{K} \hat{\boldsymbol{\Lambda}}_{tr}^{(m)} \boldsymbol{R}^{(m)} \boldsymbol{\mathcal{X}}_t^{(m)} + \boldsymbol{\mathcal{N}}_r \tag{2.54}$$

where $\hat{\boldsymbol{\Lambda}}_{tr}^{(m)}$ is a $N \times N$ diagonal matrix, i.e., $\hat{\boldsymbol{\Lambda}}_{tr}^{(m)} = diag(DFT(\boldsymbol{h}_{tr}^{(m)}))$ and $\boldsymbol{\mathcal{N}}_r$ is the noise vector with variance $\sigma_{\mathcal{N}_r} \boldsymbol{I}_N$ corresponding to $\sum_{m=1}^{K} \boldsymbol{n}_r^{(m)}$. After demapping, the m^{th} user's received signal at the r^{th} antenna can be expressed as

$$\boldsymbol{\mathcal{Y}}_r^{(m)} = \sum_{t=1}^{N_r} \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}_{tr}^{(m)} \boldsymbol{R}^{(m)} \boldsymbol{\mathcal{X}}_t^{(m)} + \boldsymbol{\mathcal{N}}_r^{(m)} \tag{2.55}$$

Let $\boldsymbol{\Lambda}_{tr}^{(m)} = \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}_{tr}^{(m)} \boldsymbol{R}^{(m)}$, then $\boldsymbol{\Lambda}_{tr}^{(m)}$ is $M \times M$ diagonal matrix. To simplify the notation, we will ignore the superscript m, then (2.55) becomes

$$\boldsymbol{\mathcal{Y}}_r = \sum_{t=1}^{N_r} \boldsymbol{\Lambda}_{tr} \boldsymbol{\mathcal{X}}_t + \boldsymbol{\mathcal{N}}_r \tag{2.56}$$

Since the number of receive antennas is assumed to be equal to the number of users transmitting using the same frequency band and time slot, hence, $N_t = N_r$.

Collecting all the received SC-FDMA symbols in one vector [27], we get

$$\begin{bmatrix} \boldsymbol{\mathcal{Y}}_1 \\ \boldsymbol{\mathcal{Y}}_2 \\ \vdots \\ \boldsymbol{\mathcal{Y}}_{N_r} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{21} & \dots & \boldsymbol{\Lambda}_{N_r1} \\ \boldsymbol{\Lambda}_{12} & \boldsymbol{\Lambda}_{22} & \dots & \boldsymbol{\Lambda}_{N_r2} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Lambda}_{1N_r} & \boldsymbol{\Lambda}_{2N_r} & \dots & \boldsymbol{\Lambda}_{N_rN_r} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{X}}_1 \\ \boldsymbol{\mathcal{X}}_2 \\ \vdots \\ \boldsymbol{\mathcal{X}}_{N_r} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mathcal{N}}_1 \\ \boldsymbol{\mathcal{N}}_2 \\ \vdots \\ \boldsymbol{\mathcal{N}}_{N_r} \end{bmatrix}$$

$$\triangleq \boldsymbol{\Lambda}\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} \tag{2.57}$$

where $\boldsymbol{\Lambda}_{ij}$ is the frequency-domain channel matrix from the i^{th} user's transmit antenna to the j^{th} receive antenna having diagonal structure. Assuming that the feedforward and feedback taps matrices have similar structure as the channel matrix $\boldsymbol{\Lambda}$. The output of the AFD-DFE is given as

$$\begin{aligned} \begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_1 \\ \check{\boldsymbol{\mathcal{X}}}_2 \\ \vdots \\ \check{\boldsymbol{\mathcal{X}}}_{N_r} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\mathcal{F}}_{1,1} & \boldsymbol{\mathcal{F}}_{1,2} & \dots & \boldsymbol{\mathcal{F}}_{1,N_r} \\ \boldsymbol{\mathcal{F}}_{2,1} & \boldsymbol{\mathcal{F}}_{2,2} & \dots & \boldsymbol{\mathcal{F}}_{2,N_r} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{F}}_{N_r,1} & \boldsymbol{\mathcal{F}}_{N_r,2} & \dots & \boldsymbol{\mathcal{F}}_{N_r,N_r} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{Y}}_1 \\ \boldsymbol{\mathcal{Y}}_2 \\ \vdots \\ \boldsymbol{\mathcal{Y}}_{N_r} \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{\mathcal{B}}_{1,1} & \boldsymbol{\mathcal{B}}_{1,2} & \dots & \boldsymbol{\mathcal{B}}_{1,N_r} \\ \boldsymbol{\mathcal{B}}_{2,1} & \boldsymbol{\mathcal{B}}_{2,2} & \dots & \boldsymbol{\mathcal{B}}_{2,N_r} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\mathcal{B}}_{N_r,1} & \boldsymbol{\mathcal{B}}_{N_r,2} & \dots & \boldsymbol{\mathcal{B}}_{N_r,N_r} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{D}}_1 \\ \boldsymbol{\mathcal{D}}_2 \\ \vdots \\ \boldsymbol{\mathcal{D}}_{N_r} \end{bmatrix} \end{aligned} \tag{2.58}$$

where the elements $\boldsymbol{\mathcal{F}}_{i,j}$ and $\boldsymbol{\mathcal{B}}_{i,j}$ are diagonal matrices containing the feedforward and the feedback tap weights as diagonal elements, respectively. Next, we will design an adaptive solution for entries of these matrices to avoid the need for the high cost and rate overhead associated with accurate MIMO multi-user channel estimation. The desired response $\boldsymbol{D}_i$ is defined as

$$\boldsymbol{D}_i = \begin{cases} F_M \boldsymbol{x}_i, & \text{for training mode} \\ \\ F_M \hat{\boldsymbol{x}}_i, & \text{for decision-directed mode} \end{cases}$$

Defining $\boldsymbol{\mathcal{Z}}_i = diag(\boldsymbol{\mathcal{Y}}_i)$ and the decision matrix $\boldsymbol{\mathcal{D}}_i = diag(\boldsymbol{D}_i)$, (2.58) becomes

$$\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_1 \\ \check{\boldsymbol{\mathcal{X}}}_2 \\ \vdots \\ \check{\boldsymbol{\mathcal{X}}}_{N_r} \end{bmatrix} = \sum_{i=1}^{N_r} \left(\begin{bmatrix} \boldsymbol{\mathcal{Z}}_i & & \\ & \ddots & \\ & & \boldsymbol{\mathcal{Z}}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}}_i^1 \\ \vdots \\ \boldsymbol{\mathcal{F}}_i^{N_r} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mathcal{D}}_i & & \\ & \ddots & \\ & & \boldsymbol{\mathcal{D}}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{B}}_i^1 \\ \vdots \\ \boldsymbol{\mathcal{B}}_i^{N_r} \end{bmatrix} \right) \tag{2.59}$$

where $\boldsymbol{\mathcal{F}}_i^j$ and $\boldsymbol{\mathcal{B}}_i^j$ are the vectors contains the diagonal elements of $\boldsymbol{\mathcal{F}}_{j,i}$ and $\boldsymbol{\mathcal{B}}_{j,i}$, respectively. Let $\check{\boldsymbol{\mathcal{X}}} = [\check{\boldsymbol{\mathcal{X}}}_1^T \ \ ... \ \ \check{\boldsymbol{\mathcal{X}}}_{N_r}^T]^T$, $\boldsymbol{\mathcal{F}}_i = [\boldsymbol{\mathcal{F}}_1^{iT} \ \ ... \ \ \boldsymbol{\mathcal{F}}_{N_r}^{iT}]^T$,$\boldsymbol{\mathcal{B}}_i = [\boldsymbol{\mathcal{B}}_1^{iT} \ \ ... \ \ \boldsymbol{\mathcal{B}}_{N_r}^{iT}]^T$ and $\acute{\boldsymbol{\mathcal{Z}}}_i$ ($\acute{\boldsymbol{\mathcal{D}}}_i$) is the diagonal matrix containing the elements $\boldsymbol{\mathcal{Z}}_i(\boldsymbol{\mathcal{D}}_i)$ on its diagonal, (2.59) can be expressed as follows

$$\check{\boldsymbol{\mathcal{X}}} \ = \ \sum_{i=1}^{N_r} [\acute{\boldsymbol{\mathcal{Z}}}_i \boldsymbol{\mathcal{F}}_i + \acute{\boldsymbol{\mathcal{D}}}_i \boldsymbol{\mathcal{B}}_i] \triangleq \boldsymbol{\mathcal{Z}} \boldsymbol{\mathcal{F}} + \boldsymbol{\mathcal{D}} \boldsymbol{\mathcal{B}} \tag{2.60}$$

where $\boldsymbol{\mathcal{Z}} = [\acute{\boldsymbol{\mathcal{Z}}}_1 \quad ... \quad \acute{\boldsymbol{\mathcal{Z}}}_{N_r}]$, $\boldsymbol{\mathcal{F}} = [\boldsymbol{\mathcal{F}}_1^T \quad ... \quad \boldsymbol{\mathcal{F}}_{N_r}^T]^T$,$\boldsymbol{\mathcal{B}} = [\boldsymbol{\mathcal{B}}_1^T \quad ... \quad \boldsymbol{\mathcal{B}}_{N_r}^T]^T$ and $\boldsymbol{\mathcal{D}} = [\acute{\boldsymbol{\mathcal{D}}}_1 \quad ... \quad \acute{\boldsymbol{\mathcal{D}}}_{N_r}]$. Now the output of the equalizer at instant k is given as

$$\check{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{Z}}_k \boldsymbol{\mathcal{F}}_{k-1} + \boldsymbol{\mathcal{D}}_k \boldsymbol{\mathcal{B}}_{k-1} \tag{2.61}$$

Defining the desired response vector as $\boldsymbol{D}_k = [\boldsymbol{D}_{1,k}^T \quad ... \quad \boldsymbol{D}_{N_r,k}^T]^T$, then $\boldsymbol{\mathcal{W}}_k$, $\boldsymbol{\mathcal{A}}_k$ and $\boldsymbol{\mathcal{E}}_k$ are given as in (2.31), (2.22) and (2.23), respectively.

<u>Reduced-Complexity RLS AFD-DFE:</u> Starting with $k = 0$ and using $\boldsymbol{P}_0^1 = \epsilon^{-1}\boldsymbol{I}_{(N_r)^2M}$, $\boldsymbol{P}_1^1$ is given by

$$\begin{aligned} \boldsymbol{P}_1^1 &= \lambda^{-1}[\boldsymbol{P}_0^1 - \lambda^{-1}\boldsymbol{P}_0^1\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_{N_rM} + \lambda^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{P}_0^1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{P}_0^1] \\ &= \lambda^{-1}[\epsilon^{-1}\boldsymbol{I}_{(N_r)^2M} - \lambda^{-1}\epsilon^{-1}\boldsymbol{I}_{(N_r)^2M}\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_{N_rM} + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1\epsilon^{-1}\boldsymbol{I}_{(N_r)^2M}] \end{aligned}$$

It can easily be seen that $\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H = [|\acute{\boldsymbol{\mathcal{Z}}}_{1,1}|^2 + |\acute{\boldsymbol{\mathcal{Z}}}_{2,1}|^2 + \quad ... \quad |\acute{\boldsymbol{\mathcal{Z}}}_{N_r,1}|^2]$. As $|\acute{\boldsymbol{\mathcal{Z}}}_{i,1}|^2$ has a diagonal structure therefore, $|\boldsymbol{\mathcal{Z}}_1|^2$ is diagonal matrix and (3.50) does not require matrix inversion. Let $\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_{N_rM} + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1 = \emptyset$ be the $(N_r^2M \times N_r^2M)$ block matrix, where each sub-matrix $\emptyset_{i,j}$ is a diagonal matrix given by $\emptyset_{i,j} = \acute{\boldsymbol{\mathcal{Z}}}_{i,1}^H(\boldsymbol{I}_{NrM} + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\acute{\boldsymbol{\mathcal{Z}}}_{j,1}$. Therefore, $\boldsymbol{P}_1^1$ has a similar structure to that of $\emptyset$ with each sub-matrix $\boldsymbol{P}_{1,(i,j)} = \lambda^{-1}[\epsilon^{-1}\boldsymbol{I}_{N_rM} - \lambda^{-1}\epsilon^{-2}\boldsymbol{I}_{N_rM}\emptyset_{i,j}]$ is also diagonal matrix. For $k = 1$, we have

$$\boldsymbol{P}_2^1 = \lambda^{-1}[\boldsymbol{P}_1^1 - \lambda^{-1}\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H(\boldsymbol{I}_{N_rM} + \lambda^{-1}\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H)^{-1}\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1] \tag{2.62}$$

where $\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H = \sum_{j=1}^{N_r} \acute{\boldsymbol{\mathcal{Z}}}_{j,2}(\sum_{i=1}^{N_r} \acute{\boldsymbol{\mathcal{Z}}}_{i,2}\boldsymbol{P}_{1,(i,j)}^1)$ is a diagonal matrix. Therefore, matrix inversion in $\boldsymbol{P}_2^1$ is simply N_rM scalar inversions. For $k > 1$, $\boldsymbol{P}_k^1$ has similar structure and, therefore, avoids complex matrix inversion and the same conclusion applies to $\boldsymbol{P}_k^2$.

2.4.3 AFD-DFE for Hybrid SM-SFBC SC-FDMA

To increase the capacity and performance, i.e. combine spatial diversity with spatial multiplexing we integrate SFBC and SM SC-FDMA. The block diagram is depicted in Fig. 2.6.

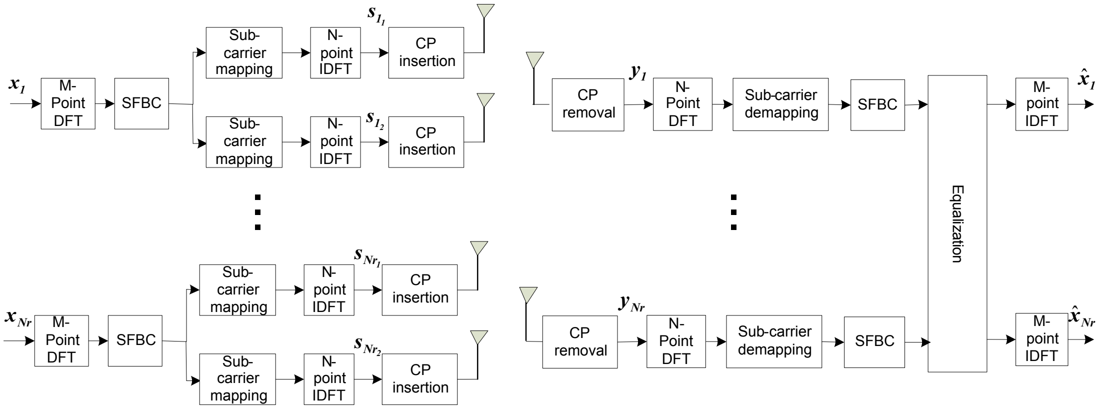


Figure 2.6: Block diagram of transceiver of Hybrid SM-SFBC SC-FDMA system.

Here we have $2N_t$ transmit antennas for N_r receive antennas. The received signal at the r^{th} antenna after cyclic prefix removal becomes

$$\boldsymbol{y}_r = \sum_{t=1}^{N_r}\sum_{m=1}^{K}[\boldsymbol{s}_{t_1}^{(m)} \circledast \boldsymbol{h}_{tr_1}^{(m)} + \boldsymbol{s}_{t_2}^{(m)} \circledast \boldsymbol{h}_{tr_2}^{(m)}] + \boldsymbol{n}_r^{(m)} \tag{2.63}$$

where $\boldsymbol{h}_{tr_i}^{(m)} = [h_{i0}^{(m)}, h_{i1}^{(m)}, ..., h_{iL(m)}^{(m)}]_{tr}$ for $i = 1, 2$. After applying an N-DFT, the signal becomes

$$\acute{\boldsymbol{\mathcal{Y}}}_r = \sum_{t=1}^{N_r}\sum_{m=1}^{K}[\hat{\boldsymbol{\Lambda}}_{tr_1}^{(m)}\boldsymbol{R}^{(m)}\boldsymbol{\mathcal{X}}_{t_1}^{(m)} + \hat{\boldsymbol{\Lambda}}_{tr_2}^{(m)}\boldsymbol{R}^{(m)}\boldsymbol{\mathcal{X}}_{t_2}^{(m)}] + \boldsymbol{\mathcal{N}}_r \tag{2.64}$$

where $\hat{\boldsymbol{\Lambda}}_{tr_i}^{(m)}$ is a $N \times N$ diagonal matrix, i.e., $\hat{\boldsymbol{\Lambda}}_{tr_i}^{(m)} = diag(DFT(\boldsymbol{h}_{tr_i}^{(m)}))$ for $i = 1, 2$ and $\boldsymbol{\mathcal{N}}_r$ is noise with variance $\sigma^2_{\mathcal{N}_r}\boldsymbol{I}_N$. After demapping, the received signal for the m^{th} user at the r^{th} antenna can be expressed as

$$\boldsymbol{\mathcal{Y}}_r^{(m)} = \sum_{t=1}^{N_r}[\boldsymbol{R}^{(m)T}\hat{\boldsymbol{\Lambda}}_{tr_1}^{(m)}\boldsymbol{R}^{(m)}\boldsymbol{\mathcal{X}}_{t_1}^{(m)} + \boldsymbol{R}^{(m)T}\hat{\boldsymbol{\Lambda}}_{tr_2}^{(m)}\boldsymbol{R}^{(m)}\boldsymbol{\mathcal{X}}_{t_2}^{(m)}] + \boldsymbol{\mathcal{N}}_r^{(m)} \tag{2.65}$$

Let $\boldsymbol{\Lambda}_{tr_i}^{(m)} = \boldsymbol{R}^{(m)T}\hat{\boldsymbol{\Lambda}}_{tr_i}^{(m)}\boldsymbol{R}^{(m)}$ for $i = 1, 2$, then $\boldsymbol{\Lambda}_{tr_i}^{(m)}$ is $M \times M$ diagonal matrix. To simplify the notation we will ignore the superscript m, then (2.65)

$$\boldsymbol{\mathcal{Y}}_r = \sum_{t=1}^{N_r}[\boldsymbol{\Lambda}_{tr_1}\boldsymbol{\mathcal{X}}_{t_1} + \boldsymbol{\Lambda}_{tr_2}\boldsymbol{\mathcal{X}}_{t_2}] + \boldsymbol{\mathcal{N}}_r \tag{2.66}$$

Using the odd even expansion, the odd component of (2.66) is

$$\boldsymbol{\mathcal{Y}}_{r_o} = \sum_{t=1}^{N_r}[\boldsymbol{\Lambda}_{tr_{1o}}\boldsymbol{\mathcal{X}}_{t_{1o}} + \boldsymbol{\Lambda}_{tr_{2o}}\boldsymbol{\mathcal{X}}_{t_{2o}}] + \boldsymbol{\mathcal{N}}_{r_o} \tag{2.67}$$

and the even component is

$$\boldsymbol{\mathcal{Y}}_{r_e} = \sum_{t=1}^{N_r}[\boldsymbol{\Lambda}_{tr_{1e}}\boldsymbol{\mathcal{X}}_{t_{1e}} + \boldsymbol{\Lambda}_{tr_{2e}}\boldsymbol{\mathcal{X}}_{t_{2e}}] + \boldsymbol{\mathcal{N}}_{r_e} \tag{2.68}$$

where $\boldsymbol{\mathcal{X}}_{t_{io}}$ and $\boldsymbol{\mathcal{X}}_{t_{ie}}$ represent the odd and even components of $\boldsymbol{\mathcal{X}}_{t_i}$ for $i = 1, 2$.

Denoting odd and even component of $\boldsymbol{\mathcal{X}}_t$ as $\boldsymbol{\mathcal{X}}_{t_o}$ and $\boldsymbol{\mathcal{X}}_{t_e}$, respectively, we have

$$\boldsymbol{\mathcal{X}}_{t_{1o}} = \boldsymbol{\mathcal{X}}_{t_o}, \quad \boldsymbol{\mathcal{X}}_{t_{2o}} = \boldsymbol{\mathcal{X}}_{t_e}, \quad \boldsymbol{\mathcal{X}}_{t_{1e}} = -\boldsymbol{\mathcal{X}}_{t_e}^*, \quad \boldsymbol{\mathcal{X}}_{t_{2e}} = \boldsymbol{\mathcal{X}}_{t_o}^* \tag{2.69}$$

Now, (2.67) and (2.68) become,respectively, as

$$\boldsymbol{\mathcal{Y}}_{r_o} = \sum_{t=1}^{N_r} [\boldsymbol{\Lambda}_{tr_{1o}} \boldsymbol{\mathcal{X}}_{t_o} + \boldsymbol{\Lambda}_{tr_{2o}} \boldsymbol{\mathcal{X}}_{t_e}] + \boldsymbol{\mathcal{N}}_{r_o} \tag{2.70}$$

and

$$\boldsymbol{\mathcal{Y}}_{r_e} = \sum_{t=1}^{N_r} [-\boldsymbol{\Lambda}_{tr_{1e}} \boldsymbol{\mathcal{X}}_{t_e}^* + \boldsymbol{\Lambda}_{tr_{2e}} \boldsymbol{\mathcal{X}}_{t_o}^*] + \boldsymbol{\mathcal{N}}_{t_e} \tag{2.71}$$

Combining (2.70) and (2.71), the linear relationship can be written as

$$\begin{aligned} \boldsymbol{\mathcal{Y}}_{r_{oe}} &= \begin{bmatrix} \boldsymbol{\mathcal{Y}}_{r_o} \\ \boldsymbol{\mathcal{Y}}_{r_e}^* \end{bmatrix} = \sum_{t=1}^{N_r} \left(\begin{bmatrix} \boldsymbol{\Lambda}_{tr_{1o}} & \boldsymbol{\Lambda}_{tr_{2o}} \\ \boldsymbol{\Lambda}_{tr_{2e}}^* & -\boldsymbol{\Lambda}_{tr_{1e}}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{X}}_{t_o} \\ \boldsymbol{\mathcal{X}}_{t_e} \end{bmatrix} \right) + \begin{bmatrix} \boldsymbol{\mathcal{N}}_{r_o} \\ \boldsymbol{\mathcal{N}}_{r_e}^* \end{bmatrix} \\ &\triangleq \sum_{t=1}^{N_r} (\boldsymbol{\Lambda}_{tr} \boldsymbol{\mathcal{X}}_{t_{oe}}) + \boldsymbol{\mathcal{N}}_{r_{oe}} \end{aligned} \tag{2.72}$$

Assuming $\boldsymbol{\Lambda}_{tr_{ie}} = \boldsymbol{\Lambda}_{tr_{io}}, \quad i = 1, 2$. $\boldsymbol{\Lambda}_{tr}$ becomes the Alamouti's matrix. Collecting all the received SC-FDMA symbols in one vector, we get

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\mathcal{Y}}_{1_{oe}} \\ \boldsymbol{\mathcal{Y}}_{2_{oe}} \\ \vdots \\ \boldsymbol{\mathcal{Y}}_{N_{r oe}} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{21} & \dots & \boldsymbol{\Lambda}_{N_r 1} \\ \boldsymbol{\Lambda}_{12} & \boldsymbol{\Lambda}_{22} & \dots & \boldsymbol{\Lambda}_{N_r 2} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Lambda}_{1 N_r} & \boldsymbol{\Lambda}_{2 N_r} & \dots & \boldsymbol{\Lambda}_{N_r N_r} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{X}}_{1_{oe}} \\ \boldsymbol{\mathcal{X}}_{2_{oe}} \\ \vdots \\ \boldsymbol{\mathcal{X}}_{N_{r oe}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mathcal{N}}_{1_{oe}} \\ \boldsymbol{\mathcal{N}}_{2_{oe}} \\ \vdots \\ \boldsymbol{\mathcal{N}}_{N_{r oe}} \end{bmatrix} \\ &\triangleq \boldsymbol{\Lambda} \boldsymbol{\mathcal{X}}_{oe} + \boldsymbol{\mathcal{N}}_{oe} \end{aligned} \tag{2.73}$$

where $\boldsymbol{\Lambda}_{ij}$ is the frequency-domain channel matrix from the i^{th} user to the j^{th} receive antenna having an Alamouti like structure. Since $N_t = 2{*}N_r$ and assuming that the feedforward taps matrices have similar structure as channel matrix $\boldsymbol{\Lambda}$, the output of the equalizer is given as

$$\begin{aligned}\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_{1,oe} \\ \check{\boldsymbol{\mathcal{X}}}_{2,oe} \\ \vdots \\ \check{\boldsymbol{\mathcal{X}}}_{N_{r_{oe}}} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\Upsilon}_{1,1} & \boldsymbol{\Upsilon}_{1,2} & \dots & \boldsymbol{\Upsilon}_{1,N_r} \\ \boldsymbol{\Upsilon}_{2,1} & \boldsymbol{\Upsilon}_{2,1} & \dots & \boldsymbol{\Upsilon}_{2,N_r} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Upsilon}_{N_r,1} & \boldsymbol{\Upsilon}_{N_r,2} & \dots & \boldsymbol{\Upsilon}_{N_r,N_r} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{Y}}_{1_{oe}} \\ \boldsymbol{\mathcal{Y}}_{2_{oe}} \\ \vdots \\ \boldsymbol{\mathcal{Y}}_{N_{r_{oe}}} \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{\Psi}_{1,1} & \boldsymbol{\Psi}_{1,2} & \dots & \boldsymbol{\Psi}_{1,N_r} \\ \boldsymbol{\Psi}_{2,1} & \boldsymbol{\Psi}_{2,1} & \dots & \boldsymbol{\Psi}_{2,N_r} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Psi}_{N_r,1} & \boldsymbol{\Psi}_{N_r,2} & \dots & \boldsymbol{\Psi}_{N_r,N_r} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{D}}_{1_{oe}} \\ \boldsymbol{\mathcal{D}}_{2_{oe}} \\ \vdots \\ \boldsymbol{\mathcal{D}}_{N_{r_{oe}}} \end{bmatrix} \end{aligned} \tag{2.74}$$

where $\boldsymbol{\mathcal{D}}_{i_{oe}}$ is equal to $\boldsymbol{\mathcal{X}}_{i_{oe}}$ in the case of the training mode and the frequency-domain decisions on $\check{\boldsymbol{\mathcal{X}}}_{i_{oe}}$ in the case of the decision-directed mode, respectively. $\boldsymbol{\Upsilon}_{i,j}$ and $\boldsymbol{\Psi}_{i,j}$ are the Alamouti like matrices of the feedforward and diagonal matrices of the feedback weights of the equalizer, respectively, i.e.,

$$\boldsymbol{\Upsilon}_{i,j} = \begin{bmatrix} \boldsymbol{\Upsilon}^{1}_{i,j} & \boldsymbol{\Upsilon}^{2}_{i,j} \\ \boldsymbol{\Upsilon}^{2*}_{i,j} & -\boldsymbol{\Upsilon}^{1*}_{i,j} \end{bmatrix}, \quad \boldsymbol{\Psi}_{i,j} = \begin{bmatrix} \boldsymbol{\Psi}^{1}_{i,j} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\Psi}^{2*}_{i,j} \end{bmatrix} \tag{2.75}$$

$\boldsymbol{\Upsilon}^{1}_{i,j}$, $\boldsymbol{\Upsilon}^{2}_{i,j}$, $\boldsymbol{\Psi}^{1}_{i,j}$ and $\boldsymbol{\Psi}^{2}_{i,j}$ are diagonal matrices. However, the explicit solution is not needed for these matrices for the adaptive solution. For the i^{th} user, we have

$$\check{\boldsymbol{\mathcal{X}}}_{i_{oe}} = \begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_{i_o} \\ \check{\boldsymbol{\mathcal{X}}}_{i_e} \end{bmatrix} = \sum_{j=1}^{N_r} (\boldsymbol{\Upsilon}_{i,j} \boldsymbol{\mathcal{Y}}_{j_{oe}} + \boldsymbol{\Psi}_{i,j} \boldsymbol{\mathcal{D}}_{j_{oe}}) \tag{2.76}$$

(2.76) can be rewritten as

$$\begin{aligned} \begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_{i_o} \\ \check{\boldsymbol{\mathcal{X}}}^*_{i,e} \end{bmatrix} &= \sum_{j=1}^{N_r} \begin{bmatrix} diag(\boldsymbol{\mathcal{Y}}_{j_o}) & diag(\boldsymbol{\mathcal{Y}}^*_{j_e}) \\ -diag(\boldsymbol{\mathcal{Y}}_{j_e}) & diag(\boldsymbol{\mathcal{Y}}^*_{j_o}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}}^1_{i,j} \\ \boldsymbol{\mathcal{F}}^2_{i,j} \end{bmatrix} \\ &+ \begin{bmatrix} diag(\boldsymbol{\mathcal{D}}_{j_o}) & \mathbf{0} \\ \mathbf{0} & diag(\boldsymbol{\mathcal{D}}^*_{j_e}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{B}}^1_{i,j} \\ \boldsymbol{\mathcal{B}}^2_{i,j} \end{bmatrix} \\ &\triangleq \sum_{j=1}^{N_r} \boldsymbol{\mathcal{Y}}^j_{oe} \boldsymbol{\mathcal{F}}_{i,j} + \boldsymbol{\mathcal{D}}^j_{oe} \boldsymbol{\mathcal{B}}_{i,j} \end{aligned} \tag{2.77}$$

where $\boldsymbol{\mathcal{F}}^1_{i,j}$ and $\boldsymbol{\mathcal{F}}^2_{i,j}$ are vectors containing the diagonal elements of $\boldsymbol{\Upsilon}^1_{i,j}$ and $\boldsymbol{\Upsilon}^2_{i,j}$, respectively. Similarly, $\boldsymbol{\mathcal{B}}^1_{i,j}$ and $\boldsymbol{\mathcal{B}}^2_{i,j}$ are vectors containing the diagonal elements of $\boldsymbol{\Psi}^1_{i,j}$ and $\boldsymbol{\Psi}^2_{i,j}$, respectively and $\boldsymbol{\mathcal{F}}_{i,j} = [\boldsymbol{\mathcal{F}}^{1T}_{i,j}\ \boldsymbol{\mathcal{F}}^{2T}_{i,j}]^T$ and $\boldsymbol{\mathcal{B}}_{i,j} = [\boldsymbol{\mathcal{B}}^{1T}_{i,j}\ \boldsymbol{\mathcal{B}}^{2T}_{i,j}]^T$. For all users, we can write

$$\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_{1_{oe}} \\ \vdots \\ \check{\boldsymbol{\mathcal{X}}}_{N_{r\,oe}} \end{bmatrix} = \sum_{j=1}^{N_r} \left(\begin{bmatrix} \boldsymbol{\mathcal{Y}}^j_{oe} & & \\ & \ddots & \\ & & \boldsymbol{\mathcal{Y}}^j_{oe} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{F}}_{1,j} \\ \vdots \\ \boldsymbol{\mathcal{F}}_{N_r,j} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mathcal{D}}^j_{oe} & & \\ & \ddots & \\ & & \boldsymbol{\mathcal{D}}^j_{oe} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{B}}_{1,j} \\ \vdots \\ \boldsymbol{\mathcal{B}}_{N_r,j} \end{bmatrix} \right) \tag{2.78}$$

(2.78) can be rewritten in compact form as

$$\check{\boldsymbol{\mathcal{X}}}_{oe} = \sum_{j=1}^{N_r} [\acute{\boldsymbol{\mathcal{Z}}}_j \boldsymbol{\mathcal{F}}_j + \acute{\boldsymbol{\mathcal{D}}}_j \boldsymbol{\mathcal{B}}_j] \tag{2.79}$$

At the k^{th} instant, we have

$$\check{\boldsymbol{\mathcal{X}}}_{oe,k} \triangleq \boldsymbol{\mathcal{Z}}_k \boldsymbol{\mathcal{F}}_k + \boldsymbol{\mathcal{D}}_k \boldsymbol{\mathcal{B}}_k \tag{2.80}$$

where $\boldsymbol{\mathcal{Z}}_k = [\acute{\boldsymbol{\mathcal{Z}}}_{1,k} \; ... \; \acute{\boldsymbol{\mathcal{Z}}}_{N_r,k}]$, $\boldsymbol{\mathcal{D}}_k = [\acute{\boldsymbol{\mathcal{D}}}_{1,k} \; ... \; \acute{\boldsymbol{\mathcal{D}}}_{N_r,k}]$, $\boldsymbol{\mathcal{F}}_k = [\boldsymbol{\mathcal{F}}_{1,k}^T \; ... \; \boldsymbol{\mathcal{F}}_{N_r,k}^T]^T$ and $\boldsymbol{\mathcal{B}}_k = [\boldsymbol{\mathcal{B}}_{1,k}^T \; ... \; \boldsymbol{\mathcal{B}}_{N_r,k}^T]^T$ Defining the desired response as $\boldsymbol{D}_k = [\boldsymbol{D}_{1,k}^T, \quad ... \quad \boldsymbol{D}_{N_r,k}^T]^T$ and

$$\boldsymbol{D}_{i,k} = \begin{cases} \begin{bmatrix} \boldsymbol{\mathcal{X}}_{i_o,k} \\ \\ \boldsymbol{\mathcal{X}}_{i_e,k}^* \end{bmatrix}, & \text{for training mode} \\ \\ \hat{\boldsymbol{\mathcal{X}}}_{i_{oe},k}, & \text{for decision-directed mode} \end{cases}$$

Now $\boldsymbol{\mathcal{W}}_k$, $\boldsymbol{\mathcal{A}}_k$ and $\boldsymbol{\mathcal{E}}_k$ are given as in (2.31), (2.22) and (2.47), respectively.

Reduced-Complexity RLS AFD-DFE: Now exploiting the special structure of SFBC matrix, it can be seen that there is no matrix inversion involved altogether and hence complexity is significantly reduced. Starting with $k = 0$ and using $\boldsymbol{P}_0^1 = \epsilon^{-1}\boldsymbol{I}_{(N_r)^2M}$, $\boldsymbol{P}_1^1$ is given by

$$\boldsymbol{P}_1^1 = \lambda^{-1}[\epsilon^{-1}\boldsymbol{I}_{(N_r)^2M} - \lambda^{-1}\epsilon^{-1}\boldsymbol{I}_{(N_r)^2M}\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_{N_rM} + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1\epsilon^{-1}\boldsymbol{I}_{(N_r)^2M}] \tag{2.81}$$

It can easily be seen that $\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H = [|\acute{\boldsymbol{\mathcal{Z}}}_{1,1}|^2 + \quad ... + \quad |\acute{\boldsymbol{\mathcal{Z}}}_{N_r,1}|^2]$, where $\acute{\boldsymbol{\mathcal{Z}}}_{i,k}\acute{\boldsymbol{\mathcal{Z}}}_{i,k}^H =$

$diag[|\boldsymbol{\mathcal{Y}}^i_{oe}|^2 \quad ,..., \quad |\boldsymbol{\mathcal{Y}}^i_{oe}|^2]$. Therefore, $|\boldsymbol{\mathcal{Z}}_k|^2$ has diagonal structure and the RLS update equation (2.81) does not require any matrix inversion. Now $\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_{N_rM} + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1 = \boldsymbol{\emptyset}$ is $(N_r)^2M \times (N_r)^2M$ matrix, with each entry $\boldsymbol{\emptyset}_{i,j} = \acute{\boldsymbol{\mathcal{Z}}}^H_{i,1}(\boldsymbol{I}_{N_rM} + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\acute{\boldsymbol{\mathcal{Z}}}_{j,1}$ is $N_rM \times N_rM$ block diagonal matrix. Each sub-matrix in $\boldsymbol{\emptyset}_{i,j}$ is $M \times M$ block matrix, which further consists of 4 $\frac{M}{2} \times \frac{M}{2}$ diagonal matrices. Therefore, $\boldsymbol{P}_1^1$ has a similar structure to that of $\boldsymbol{\emptyset}$ with sub-matrix $\boldsymbol{P}_{1,(i,j)} = \lambda^{-1}[\epsilon^{-1}\boldsymbol{I}_{N_rM} - \lambda^{-1}\epsilon^{-2}\boldsymbol{I}_{N_rM}\boldsymbol{\emptyset}_{i,j}]$ is $N_rM \times N_rM$ block diagonal matrix. For $k = 1$, we have

$$\boldsymbol{P}_2^1 = \lambda^{-1}[\boldsymbol{P}_1^1 - \lambda^{-1}\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H(\boldsymbol{I}_{N_rM} + \lambda^{-1}\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H)^{-1}\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1] \tag{2.82}$$

Using simple algebra, it can be verified that

$$(\boldsymbol{I}_{N_rM} + \lambda^{-1}\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H)^{-1} = \begin{bmatrix} \boldsymbol{A}_1^{-1} & & \\ & \ddots & \\ & & \boldsymbol{A}_{N_r}^{-1} \end{bmatrix} \tag{2.83}$$

where $\boldsymbol{A}_i = \begin{bmatrix} A_{i,1} & A_{i,2} \\ A_{i,3} & A_{i,4} \end{bmatrix}$ and each entry $A_{i,j}$ is a diagonal matrix. Therefore, inverse of $\boldsymbol{A}_i$ can be found using block matrix inversion [28] as follows

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix} \tag{2.84}$$

where all the sub-matrices A, B, C, D are diagonal, therefore, their inversions are scalar inversions. For $k > 1$, $\boldsymbol{P}_k^1$ has similar structure. Moreover, for $\boldsymbol{P}_k^2$ same reasons are valid as for the SM case.

2.5 Performance and Complexity Analysis

In this section, the Minimum Mean Square Error (MMSE), transient, steady-state, tracking and computational complexity analyses are carried out. In the derivations, we assume that the data sequences (both transmitted data and detected data) are independent and identically distributed (i.i.d) with zero mean, and independent of the noise. Using the optimal MMSE equalizer weights $\boldsymbol{\mathcal{W}}^o$ found in [23] and [1], the corresponding i^{th} frequency bin MMSE for the LE ($J_L(i)$) and the DFE ($J_D(i)$) are derived, respectively, and expressed as

$$J_L(i) = \frac{N\sigma_{\mathcal{N}}^2\sigma_x^2(M|\Lambda(i)|^2\sigma_x^2 + N\sigma_{\mathcal{N}}^2)}{(\sigma_x^2|\Lambda(i)|^2M + N\sigma_{\mathcal{N}}^2)^2} \tag{2.85}$$

and

$$J_D(i) = \frac{N\sigma_{\mathcal{N}}^2\sigma_x^2(M|\Lambda(i)|^2\sigma_x^2 + N\sigma_{\mathcal{N}}^2)}{(\sigma_x^2\sum_{j=0}^{M-1}|\Lambda(j)|^2 + N\sigma_{\mathcal{N}}^2)^2} \tag{2.86}$$

2.5.1 Transient Analysis

We start by deriving the transient behavior of the RLS AFD-DFE in a stationary environment, assuming that the forgetting factor λ is unity. The MSE of the RLS

AFD-DFE at instant $k+1$ is given by

$$J_{k+1}^{r}(i) = E\left|\xi_{k+1}(i)\right|^{2} \tag{2.87}$$

The a priori estimation output error $\xi_{k+1}(i)$ can be expressed as

$$\xi_{k+1}(i) = \xi_{k+1}^{o}(i) + \boldsymbol{a}_{k+1}(i)\underbrace{[\boldsymbol{\mathcal{W}}^{o}(i) - \boldsymbol{\mathcal{W}}_{k}(i)]}_{\widetilde{\boldsymbol{\mathcal{W}}}_{k}(i)} \tag{2.88}$$

where $X_k(i) = \boldsymbol{a}_k(i)\boldsymbol{\mathcal{W}}^{o}(i) + \xi_k^{o}(i)$, $\boldsymbol{a}_k(i) = [\mathcal{Y}_k(i) \;\; D_k(i)]$ and $\boldsymbol{\mathcal{W}}_k(i) = [\mathcal{F}_k(i) \;\; \mathcal{B}_k(i)]^{T}$. Substituting (2.88) into (2.87) and then expending terms, we get

$$\begin{aligned} J_{k+1}^{r}(i) &= \underbrace{E\left|\xi_{k+1}^{o}(i)\right|^{2}}_{J_D(i)} + E\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}^{H}(i)\boldsymbol{a}_{k+1}^{H}(i) \\ &\quad + E\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}(i)\xi_{k+1}^{o*}(i) + E\xi_{k+1}^{o}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}^{H}(i)\boldsymbol{a}_{k+1}^{H}(i) \end{aligned} \tag{2.89}$$

We can express the second expectation in (2.89) as

$$\begin{aligned} E\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}^{H}(i)\boldsymbol{a}_{k+1}^{H}(i) &= E\;Tr\{\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}^{H}(i)\boldsymbol{a}_{k+1}^{H}(i)\} \\ &= E\;Tr\{\widetilde{\boldsymbol{\mathcal{W}}}_{k}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}^{H}(i)\boldsymbol{a}_{k+1}^{H}(i)\boldsymbol{a}_{k+1}(i)\} \\ &= Tr\;E\{\widetilde{\boldsymbol{\mathcal{W}}}_{k}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k}^{H}(i)\boldsymbol{a}_{k+1}^{H}(i)\boldsymbol{a}_{k+1}(i)\} \end{aligned} \tag{2.90}$$

where $Tr\{.\}$ is the trace of a matrix. Using the assumption that the product $\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)$ varies at a slower rate than the product $\boldsymbol{a}_{k+1}^H(i)\boldsymbol{a}_{k+1}(i)$, we can write

$$E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)\boldsymbol{a}_{k+1}^H(i)\boldsymbol{a}_{k+1}(i) \approx Tr\{E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)E\boldsymbol{a}_{k+1}^H(i)\boldsymbol{a}_{k+1}(i)\} \quad (2.91)$$

where $E\boldsymbol{a}_k^H(i)\boldsymbol{a}_k(i) = \boldsymbol{R}_{\mathcal{A}}(i)$ is the input correlation matrix having diagonal structure as can be seen from (2.24), i.e., $\boldsymbol{R}_{\mathcal{A}}(i) = diag([|\Lambda(i)|^2\sigma_x^2 + \frac{N}{M}\sigma_{\mathcal{N}}^2 \quad \sigma_x^2])$. Now, to find $E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)$, we proceed by writing the normal equations

$$\boldsymbol{\mathcal{W}}_k(i) = \Theta_k^{-1}(i)\boldsymbol{R}_k(i) \quad (2.92)$$

where $\boldsymbol{R}_k(i) = \sum_{j=0}^{k}\boldsymbol{a}_j^H(i)X_j(i)$. Using the values of $\Theta_k(i)$, $\boldsymbol{R}_k(i)$ and $X_k(i)$ in (2.92) and ignoring the initial conditions, $\widetilde{\boldsymbol{\mathcal{W}}}_k(i)$ is given as

$$\widetilde{\boldsymbol{\mathcal{W}}}_k(i) = -\Theta_k^{-1}(i)\sum_{j=0}^{k+1}\boldsymbol{a}_j^H(i)\xi_j^o(i) \quad (2.93)$$

Therefore, the weight-error correlation matrix is given as

$$E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i) = E\Theta_k^{-1}(i)\sum_{j=0}^{k+1}\boldsymbol{a}_j^H(i)\boldsymbol{a}_j(i)\Theta_k^{-1}(i)\xi_j^o(i)\xi_j^{o*}(i) \quad (2.94)$$

Using the assumption that $\boldsymbol{a}_k(i)$, and therefore $\Theta_k^{-1}(i)$, is independent of the noise $\xi_k^o(i)$, (2.94) can be expressed as a product of two expectations as follows

$$\begin{aligned} E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i) &= E\Theta_k^{-1}(i)\sum_{j=0}^{k+1}\boldsymbol{a}_j^H(i)\boldsymbol{a}_j(i)\Theta_k^{-1}(i)\ E\xi_j^o(i)\xi_j^{o*}(i) \\ &= J_D(i)\ E\Theta_k^{-1}(i) \end{aligned} \tag{2.95}$$

Assuming ergodicity, we may express the ensemble-average correlation matrix of the input of the AFD-DFE as $\boldsymbol{R}_{\mathcal{A}}(i) = \frac{1}{k}\Theta_k(i)$. Hence, the weight-error correlation matrix reduces to $E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i) = \frac{1}{k}J_D(i)\boldsymbol{R}_{\mathcal{A}}^{-1}(i)$. Therefore, (2.90) becomes

$$E\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)\boldsymbol{a}_{k+1}^H(i) \approx \frac{1}{k}J_D(i)Tr\{\boldsymbol{R}_{\mathcal{A}}(i)\boldsymbol{R}_{\mathcal{A}}^{-1}(i)\} = \frac{2}{k}J_D(i) \tag{2.96}$$

The third and fourth expectations in (2.89) are zero because $\widetilde{\boldsymbol{\mathcal{W}}}_k(i)$ depends on past values of $\boldsymbol{a}_{k+1}(i)$ and $\xi_{k+1}^o(i)$. Also $\boldsymbol{a}_{k+1}(i)$ and $\xi_{k+1}^o(i)$ are statistically independent and $\xi_{k+1}^o(i)$ has zero mean. Therefore,

$$J_{k+1}^r(i) = J_D(i)\left[1 + \frac{2}{k}\right] \tag{2.97}$$

To compare the RLS AFD-DFE and LMS AFD-DFE, the MSE of LMS AFE-DFE can be shown to be [29]

$$J_{k+1}^l(i) = J_D(i)\left[1 + \mu\sum_{j=0}^{1}\frac{r_j}{2-\mu r_j}\right] + \sum_{j=0}^{1} r_j\left(|\nu_j(0)|^2 - \frac{\mu J_D(i)}{2-\mu r_j}\right)(1-\mu r_j)^{2k} \tag{2.98}$$

where r_j is the j^{th} eigenvalue of the correlation matrix $\boldsymbol{R}_{\mathcal{A}}(i)$ and $\boldsymbol{\nu}(k+1) = [\nu_0(k+1), \nu_1(k+1)]^T = (\boldsymbol{I}_2 - \mu \boldsymbol{R}_{\mathcal{A}}(i))\boldsymbol{\nu}(k) - \mu \boldsymbol{\mathcal{A}}_{k+1}^H(i)\boldsymbol{\mathcal{E}}_{k+1}(i)$. The evolution of $J_{k+1}^l(i)$ with step size μ is governed by the exponential quantity $(1-\mu r_j)^{2k}$. This clearly shows that the RLS AFD-DFE converges faster than the LMS AFD-DFE. The simulation results support this claim.

2.5.2 Steady-State Analysis

To begin with, the update recursion (2.27) for the i^{th} frequency bin can be written as

$$\boldsymbol{\mathcal{W}}_{k+1}(i) = \boldsymbol{\mathcal{W}}_k(i) + \boldsymbol{\mathcal{P}}_{k+1}(i)\boldsymbol{a}_{k+1}^H(i)\xi_{k+1}(i) \tag{2.99}$$

In terms of the weight-error vector $\widetilde{\boldsymbol{\mathcal{W}}}_k(i)$, we can write (2.99) as

$$\widetilde{\boldsymbol{\mathcal{W}}}_{k+1}(i) = \widetilde{\boldsymbol{\mathcal{W}}}_k(i) - \boldsymbol{\mathcal{P}}_{k+1}(i)\boldsymbol{a}_{k+1}^H(i)\xi_{k+1}(i) \tag{2.100}$$

Multiplying (2.100) by $\boldsymbol{a}_{k+1}(i)$ from the left, we may write it in terms of a priori estimation error $\xi_{k+1}^a(i)$ and a posteriori estimation error $\xi_{k+1}^p(i)$ as follows

$$\underbrace{\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_{k+1}(i)}_{\xi_{k+1}^p(i)} = \underbrace{\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_k(i)}_{\xi_{k+1}^a(i)} - \underbrace{\boldsymbol{a}_{k+1}(i)\boldsymbol{\mathcal{P}}_{k+1}(i)\boldsymbol{a}_{k+1}^H(i)}_{\|\boldsymbol{a}_{k+1}(i)\|_{\boldsymbol{\mathcal{P}}}}\xi_{k+1}(i) \tag{2.101}$$

where $\| \, . \, \|_{\mathcal{P}}$ stands for the squared-weighted Euclidean norm of a vector. Combining (2.100) and (2.101) to eliminate ξ_{k+1}, we get

$$\begin{aligned} \widetilde{\boldsymbol{\mathcal{W}}}_{k+1}(i) + \boldsymbol{\mathcal{P}}_{k+1}(i)\boldsymbol{a}^H_{k+1}(i)\underbrace{(\| \boldsymbol{a}_{k+1}(i) \|_{\mathcal{P}})^{\dagger}}_{\bar{a}_{k+1}(i)}\xi^a_{k+1}(i) \quad &= \quad \widetilde{\boldsymbol{\mathcal{W}}}_k(i) + \boldsymbol{\mathcal{P}}_{k+1}(i)\boldsymbol{a}^H_{k+1}(i) \\ &\quad \times(\| \boldsymbol{a}_{k+1}(i) \|_{\mathcal{P}})^{\dagger}\xi^p_{k+1}(i) \end{aligned} \tag{2.102}$$

where $(.)^{\dagger}$ represents the pseudo-inverse. Now, equating the energies (squared Euclidean norms) of both sides of (2.102) with $[\boldsymbol{\mathcal{P}}_{k+1}(i)]^{-1}$ as a weighting matrix, the energy conservation relation becomes

$$\| \widetilde{\boldsymbol{\mathcal{W}}}_{k+1}(i) \|^2_{\mathcal{P}^{-1}} + \bar{a}_{k+1}(i)|\xi^a_{k+1}(i)|^2 = \| \widetilde{\boldsymbol{\mathcal{W}}}_k(i) \|^2_{\mathcal{P}^{-1}} + \bar{a}_{k+1}(i)|\xi^p_{k+1}(i)|^2 \tag{2.103}$$

Taking the expectation of (2.103) and using the steady-state approximations, $E\boldsymbol{\mathcal{P}}_{k+1}(i) \approx (1-\lambda)\boldsymbol{\mathcal{R}}^{-1}_{\mathcal{A}} = \mathcal{P}$, $E\widetilde{\boldsymbol{\mathcal{W}}}_{k+1}(i) = E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)$ and $E \| \widetilde{\boldsymbol{\mathcal{W}}}_{k+1}(i) \|^2_{\mathcal{P}^{-1}} = E \| \widetilde{\boldsymbol{\mathcal{W}}}_k(i) \|^2_{\mathcal{P}^{-1}}$, we arrive at

$$E\bar{a}_{k+1}(i)|\xi^a_{k+1}(i)|^2 = E\bar{a}_{k+1}(i)|\xi^p_{k+1}(i)|^2, \; k \to \infty \tag{2.104}$$

Substituting $\xi^p_{k+1}(i)$ from (2.101) into (2.104), we get

$$E\bar{a}_{k+1}(i)|\xi^a_{k+1}(i)|^2 = E\bar{a}_{k+1}(i)|\xi^a_{k+1}(i) - \| \boldsymbol{a}_{k+1}(i) \|_{\mathcal{P}} \, \xi_{k+1}(i)|^2, \; k \to \infty \tag{2.105}$$

which upon expansion and simplification reduces to

$$E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} |\xi_{k+1}(i)|^2 = 2Re(E\xi^{a*}_{k+1}(i)\xi_{k+1}(i)), \ k \to \infty \tag{2.106}$$

As $\xi_{k+1}(i) = \xi^o_{k+1}(i) + \xi^a_{k+1}(i)$, (2.106) becomes

$$J_D(i)E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} + E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} |\xi^a_{k+1}(i)|^2 = 2\underbrace{E|\xi^a_{k+1}(i)|^2}_{J_{exss}(i)}, \ k \to \infty \tag{2.107}$$

where $J_{exss}(i)$ is the Excess Mean Square Error (EMSE). Assume that at steady-state, $\parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}}$ is independent of $\xi^a_{k+1}(i)$. This condition allows us to separate the expectation $E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} |\xi^a_{k+1}(i)|^2$ into a product of two expectations as follows

$$E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} |\xi^a_{k+1}(i)|^2 = E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} E|\xi^a_{k+1}(i)|^2 \tag{2.108}$$

If we replace $\boldsymbol{\mathcal{P}}_{k+1}(i)$ by its assumed mean value, we obtain the approximation

$$E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} \approx Tr\{\boldsymbol{\mathcal{R}}\mathcal{P}\} = 2(1-\lambda) \tag{2.109}$$

Substituting into (2.107), we get

$$J_{exss}(i) = \frac{J_D(i)(1-\lambda)}{\lambda} \tag{2.110}$$

Therefore, the MSE at the steady-state $J_{ss}(i)$ is given as

$$J_{ss}(i) = J_D(i) + J_{exss}(i) = \frac{J_D(i)}{\lambda} \tag{2.111}$$

2.5.3 Tracking Analysis

For time-varying channels, we will adopt a first-order random walk model for the variation in the tap weight vector $\boldsymbol{\mathcal{W}}^o_k$. The model assumes that $\boldsymbol{\mathcal{W}}^o_k$ undergoes random variations of the form

$$\boldsymbol{\mathcal{W}}^o_{k+1}(i) = \boldsymbol{\mathcal{W}}^o_k(i) + \boldsymbol{q}_{k+1}(i) \tag{2.112}$$

where $\boldsymbol{q}_k(i) = [\boldsymbol{f}_L(i)\boldsymbol{q}^F_k(i), \boldsymbol{f}_L(i)\boldsymbol{q}^B_k(i)]^T$ and $\boldsymbol{f}_L(i)$ is the i^{th} row of partial $(M \times L)$ DFT matrix. $\boldsymbol{q}^F_k(i)$ and $\boldsymbol{q}^B_k(i)$ are the time-domain random column vectors of length L with zero mean and correlation matrix $\boldsymbol{Q}_t = (1 - \mathcal{R}^2(1))\boldsymbol{I}_L$ [5]. Assuming $\boldsymbol{q}^F_k(i)$ and $\boldsymbol{q}^B_k(i)$ are independent and note that $\boldsymbol{f}_L(i)\boldsymbol{f}_L(i)^H = L/M$, the covariance matrix of $\boldsymbol{q}_k(i), i = 1, ..., M$, is $\boldsymbol{Q} = \frac{L}{M}(1 - \mathcal{R}^2(1))\boldsymbol{I}_{2L}$.

Now, defining $\widetilde{\boldsymbol{\mathcal{W}}}_k(i) = \boldsymbol{\mathcal{W}}^o_k(i) - \boldsymbol{\mathcal{W}}_k(i)$ and invoking the energy-conservation relation leads to

$$\begin{aligned} \| \boldsymbol{\mathcal{W}}^o_{k+1}(i) - \boldsymbol{\mathcal{W}}_{k+1}(i) \|^2_{\mathcal{P}^{-1}} + \bar{a}_{k+1}(i)|\xi^a_{k+1}(i)|^2 &= \| \boldsymbol{\mathcal{W}}^o_{k+1}(i) - \boldsymbol{\mathcal{W}}_k(i) \|^2_{\mathcal{P}^{-1}} \\ &\quad + \bar{a}_{k+1}(i)|\xi^p_{k+1}(i)|^2 \end{aligned} \tag{2.113}$$

where $\xi^p_{k+1}(i) = \boldsymbol{a}_{k+1}(i)[\boldsymbol{\mathcal{W}}^o_{k+1}(i) - \boldsymbol{\mathcal{W}}_{k+1}(i)]$ and $\xi^a_{k+1}(i) = \boldsymbol{a}_{k+1}(i)[\boldsymbol{\mathcal{W}}^o_{k+1}(i) -$

$\boldsymbol{\mathcal{W}}_k(i)$]. Moreover, the random walk model (2.112) allows us to relate $E \parallel \boldsymbol{\mathcal{W}}^o_{k+1}(i) - \boldsymbol{\mathcal{W}}_k(i) \parallel^2_{\mathcal{P}^{-1}}$ to $E \parallel \widetilde{\boldsymbol{\mathcal{W}}}_k(i) \parallel^2_{\mathcal{P}^{-1}}$ as follows

$$\begin{aligned} E \parallel \boldsymbol{\mathcal{W}}^o_{k+1}(i) - \boldsymbol{\mathcal{W}}_k(i) \parallel^2_{\mathcal{P}^{-1}} &= E \parallel \boldsymbol{\mathcal{W}}^o_k(i) + \boldsymbol{q}_{k+1}(i) - \boldsymbol{\mathcal{W}}_k(i) \parallel^2_{\mathcal{P}^{-1}} \\ &= E \parallel \widetilde{\boldsymbol{\mathcal{W}}}_k(i) + \boldsymbol{q}_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} \\ &= E \parallel \widetilde{\boldsymbol{\mathcal{W}}}_k(i) \parallel^2_{\mathcal{P}^{-1}} + E \parallel \boldsymbol{q}_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} \end{aligned} \quad (2.114)$$

where the last step follows from the fact that $\boldsymbol{\mathcal{W}}_k(i)$ is independent of $\boldsymbol{q}_{k+1}(i)$ and uses steady-state assumption. Next, taking expectation of (2.113), we get

$$\begin{aligned} E \parallel \widetilde{\boldsymbol{\mathcal{W}}}_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} + E\bar{a}_{k+1}(i)|\xi^a_{k+1}(i)|^2 &= E \parallel \widetilde{\boldsymbol{\mathcal{W}}}_k(i) \parallel^2_{\mathcal{P}^{-1}} + E \parallel \boldsymbol{q}_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} \\ &\quad + E\bar{a}_{k+1}(i)|\xi^p_{k+1}(i)|^2 \end{aligned} \quad (2.115)$$

Moreover, $\boldsymbol{q}_{k+1}(i)$ is independent of $\boldsymbol{\mathcal{P}}_{k+1}(i)$, so that

$$E \parallel \boldsymbol{q}_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} = Tr\ E\{\boldsymbol{q}_{k+1}(i)^H \mathcal{P}^{-1} \boldsymbol{q}_{k+1}(i)\} = Tr\{\boldsymbol{Q}\mathcal{P}^{-1}\} = \frac{1}{(1-\lambda)} Tr\{\boldsymbol{Q}\boldsymbol{R}_{\mathcal{A}}\} \quad (2.116)$$

Solving (2.115) as done in steady-state analysis and using (2.116), it can be shown that in time-varying environment, the MSE of the RLS AFD-DFE is given as

$$\begin{aligned} J_{ss}(i) &= J_D(i) + \frac{2(1-\lambda)J_{DFE}(i) + \frac{1}{(1-\lambda)} Tr\{\boldsymbol{Q}\boldsymbol{R}\}}{2 - 2(1-\lambda)} \\ &= \frac{J_D(i)}{\lambda} + \frac{Tr\{\boldsymbol{Q}\boldsymbol{R}\}}{2\lambda(1-\lambda)} \end{aligned} \quad (2.117)$$

2.5.4 Computational Complexity

In this section, the computational complexity of the AFD-DFE for SISO SC-FDMA system is compared with that of the MMSE DFE [1] with perfect channel knowledge. The computational complexity will be evaluated in terms of the total number of real multiplications required to compute the feedforward and feedback filter coefficients per block (one SC-FDMA block) for complex-valued data.

- First, the matrices $\boldsymbol{P}_{k+1}^{1}$ and $\boldsymbol{P}_{k+1}^{2}$ as given in (2.29) and (2.30), respectively, require $13M$ real multiplications each, therefore, $\boldsymbol{\mathcal{P}}_{k+1}$ amounts to $26M$ real multiplications.

- Second, the term $\boldsymbol{\mathcal{P}}_{k+1}\boldsymbol{\mathcal{A}}_{k+1}^{H}\boldsymbol{\mathcal{E}}_{k+1}$ in (2.27) requires $16M$ real multiplications.

- Finally, $8M$ real multiplications are required for computing the error term $\boldsymbol{\mathcal{E}}_{k+1}$ in (2.23).

Therefore, the total real multiplications count for the AFD-DFE is $50M$. For the MMSE DFE $20(M+1)N_l + 2N_l + 12M$ real multiplications are needed, where N_l denotes the number of iterations for each block and $N_l > 1$ (these iterations are needed to solve the causality problem in (2.16) [23]). Compared with the AFD-LE, the number of computations required to calculate the coefficients of the AFD-DFE is doubled due to the feedback filter. Moreover, the AFD-LE does not require iterative procedure.

Our AFD-DFE is computationally efficient in the MIMO case as well, since the MIMO MMSE DFE requires matrix inversion [30]. The required number of

Table 2.2: Computational complexity of the AFD-DFE and MMSE DFE

	Structure	Real Multiplications
SISO	AFD-DFE	$50M$
	MMSE DFE [1] (with known channel)	$20(M+1)N_l + 2N_l + 12M$
MIMO	AFD-DFE	$26N_t^3M + 26N_t^2M + 2N_tM$
	MMSE DFE [30](with known channel)	$8MN_t^3 - (20N_l)/3 - 6N_lN_t^2$ $+(62N_lN_t^3)/3 + (4N_lN_t^4)/3$ $+2N_t^2 + 2N_t^3 + (20N_lN_t)/3$ $+(20MN_lN_t)/3 - 6MN_lN_t^2+$ $(86MN_lN_t^3)/3 + (4MN_lN_t^4)/3$

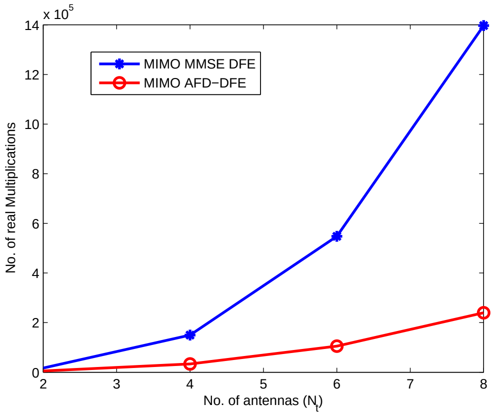


Figure 2.7: Computational complexity of MMSE DFE [1] and proposed AFD-DFE versus the number of antennas for $M = 16$.

real multiplications for the MIMO AFD-DFE (assuming equal number of transmit and receive antennas) is detailed as follows.

- $\boldsymbol{P}_{k+1}^1$ and $\boldsymbol{P}_{k+1}^2$ requires $(9N_t^3M + 5N_t^2M + N_tM)$ real multiplications each, therefore $\boldsymbol{\mathcal{P}}_{k+1}$ requires $(18N_t^3M + 10N_t^2M + 2N_tM)$ real multiplications.
- The error $\boldsymbol{\mathcal{E}}_{k+1}$ requires $8N_t^2M$ real multiplications.
- Additional $(8N_t^3M + 8N_t^2M)$ real multiplications are needed to calculate the equalizer weights $\boldsymbol{\mathcal{W}}_{k+1}$.

In summary, the MIMO AFE-DFE requires $(26N_t^3M + 26N_t^2M + 2N_tM)$ real multiplications. In case of the MIMO MMSE DFE, we estimate the number of multiplications required for matrix inversion, using the approach given in [31]. Finally, Table 2.2 summarizes the real multiplications of the AFD-DFE and MMSE DFE for complex valued data. It is clear from Fig. 2.7 that the computational complexity of the AFD-DFE is less than that of the MMSE DFE for $N_l = 4$ ($N_l > 4$ does not give any significant improvement in the performance [23]). Note here that we are assuming that the channel is known in case of the MMSE DFE but in reality it needs to be estimated which will increase the complexity. Furthermore, pilots will be needed to estimate the channel which will increase the overhead.

2.6 Simulation Results

In this section, the theoretical findings are validated. Similar to an LTE system, the carrier frequency and bandwidth are set to 2 GHz and 5 MHz, respectively. Other simulation parameters used are $M = 16$ and $N = 512$, therefore, the maximum number of users that the system can support is $K = 32$. The modulation scheme used is Quadrature Phase shift Keying (QPSK) and the channel is frequency selective with 6-paths and each path fades independently, according to the Rayleigh distribution. For Figs. 2.8 to 2.11, the user velocity v is taken as 3km/h.

Fig. 2.8 assumes interleaved mapping with no CFO and the signal-to-noise ratio (SNR) is set to 20dB. As can be seen in this figure, the AFD-DFE outperforms the AFD-LE in this scenario where more than 18dB gain in MSE is achieved. More importantly, both equalizers have the same convergence speed. The computational complexity of the AFD-DFE is slightly higher than that of the AFD-LE but when compared to the performance obtained through the use of the AFD-DFE, this additional complexity at the base station is well justified. Also, the figure shows that the performance of the RLS-based AFD-DFE is better as compared to that of the LMS-based AFD-DFE in terms of convergence speed and MSE. Only 15 iterations are needed for the RLS to converge; hence, the resulting complexity and latency increase due to adaptation are not significant. This figure also depicts the theoretical curves for the three algorithms. Close agreement between the theoretical findings and simulations is observed. Note that the theoretical curve of AFD-DFE assumes perfect decisions.

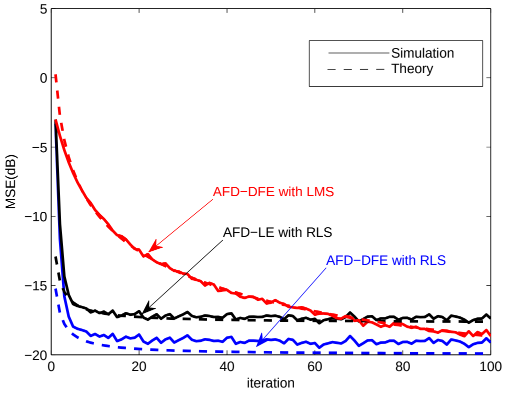


Figure 2.8: Learning curves of LE and DFE in SCFDMA system.

The Bit Error Rate (BER) performance of the AFD-DFE with RLS is shown in Fig. 2.9. It is clear that the AFD-DFE with RLS outperforms the LMS-based AFD-DFE and LE with known channel in terms of BER. Note here that minimal error propagation was observed in the AFD-DFE at low SNR unlike the MMSE-DFE of [1] and [23] with known channel, which was due to the poor estimation of the correlation between the transmitted data and the decisions. Furthermore, the reliability of the proposed algorithm is increased by a judicious choice of the initial value of the autocorrelation matrix for both the feedforward and feedback sections. This gives the proposed algorithm similar performance to that of the linear equalizer at low SNR's. For the rest of this chapter we have used the RLS-based AFD-DFE.

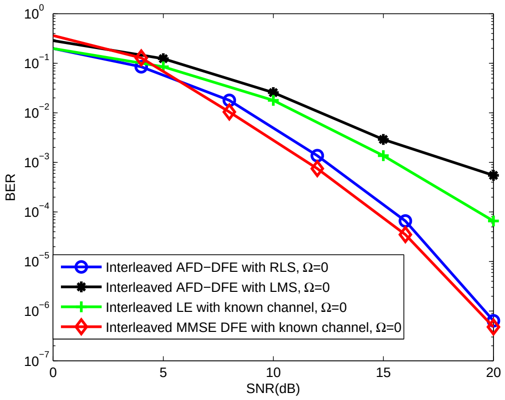


Figure 2.9: Comparison of RLS AFD-DFE, LMS AFD-DFE, LE and MMSE DFE [1]

.

Next, two mapping techniques, as defined by (2.1), are compared and the results are reported in Fig. 2.10 which shows that the performance of interleaved mapping is better than the localized mapping. The reason is that the performance of the localized mapping is sub-band dependent. If a user is assigned a sub-band which is near a deep null of the channel then the performance degrades.

CFO, Ω, is very harmful if there is a slip in frequency and, therefore, can de-

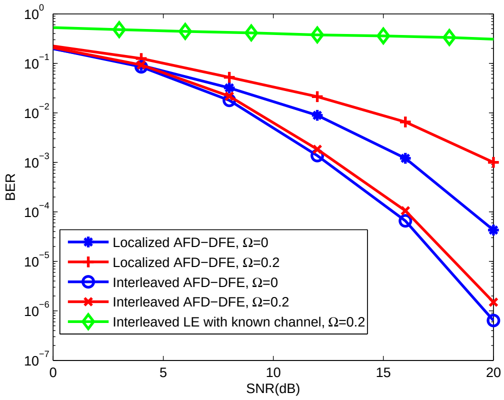


Figure 2.10: System's performance under CFO's effect with Single user for two mapping techniques.

grade the system's performance significantly. Fig. 2.10 depicts the performance of the system under two different CFO's, $\Omega = 0$ and $\Omega = 0.2$ for a single user. Localized mapping is more sensitive to CFO in the single user case due to interference from adjacent carriers, while there is no significant performance loss in case of interleaved mapping in worst case ($\Omega = 0.2$). Moreover, clear gain in the performance of AFD-DFE over LE (with perfectly known channel) in case of $\Omega = 0.2$, can be seen. In the ensuing simulations, the interleaved mapping technique is used as it achieves acceptable performance which does not depend on sub-band assigned to the user.

The effect of the CFO on the performance of the SC-FDMA system is also investigated for three users. The same parameters used in the aforementioned scenario are used here. For this case, the CFO's of user 1, user 2 and user 3 is denoted by Ω_1, Ω_2 and Ω_3, respectively. As can be seen from Fig. 2.11, there is no significant performance loss in the worst case i.e., $\Omega_1 = 0.1, \Omega_2 = 0.2, \Omega_3 = 0.3$ as compared to the best case i.e., $\Omega_1 = \Omega_2 = \Omega_3 = 0$, which shows the robustness of our AFD-DFE to CFO. The reason of good performance of the AFD-DFE is explained next. Under CFO, the frequency-domain channel matrix is no longer diagonal. However, the three main diagonals contain most of the channel energy. Therefore, most of the interference will occur only between the adjacent sub-carriers. Since, there are maximum of three users only in our simulations, these users are not assigned adjacent carriers in the case of distributed sub-carriers allocation. Hence, the effect of CFO is minimal. However, as the number of users becomes large, system operating under full load condition, CFO will be unavoidable as there will be no freedom in assigning non-adjacent sub-carriers to the users.

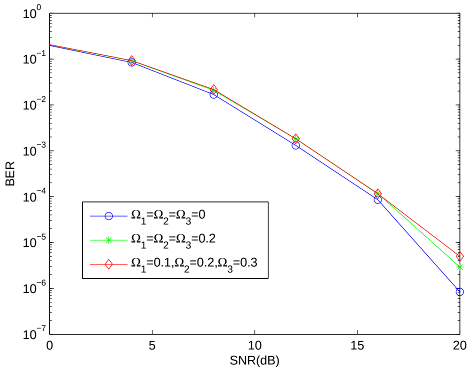


Figure 2.11: System's performance under CFO's effect with three users.

In addition, the effect of the user velocity (v) on the system's performance is shown in Fig. 2.12 and 2.13 for a single user and three users, respectively. Perfect transmitter/ receiver oscillator's synchronization is assumed when evaluating the performance under Doppler effect. Three velocities are chosen for comparison i.e. low (3 km/h), medium (30 km/h) and high (300 km/h). For the case of three users, different velocities are assigned to the users and these are v_1, v_2 and v_3

for user 1, user 2 and user 3, respectively. As can be observed from these two figures, our proposed RLS AFD-DFE is robust to the Doppler Effect; however, at very high SNR the system's performance deteriorates as the Doppler frequency increases due to interference from adjacent sub-carriers. Fig. 2.14 depicts the theoretical and simulated MSE for different user's velocities. Slight degradation is observed for large values of velocity. This performance degradation due to Doppler can be mitigated using SFBC a shown next.

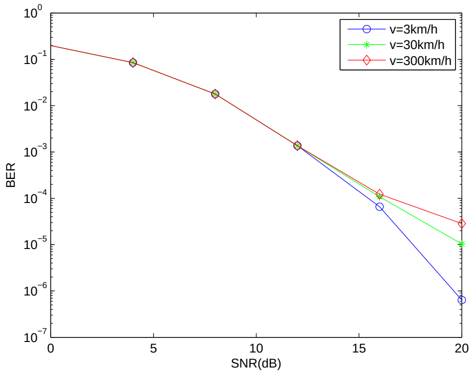


Figure 2.12: Effect of user's velocity on system's performance with single user.

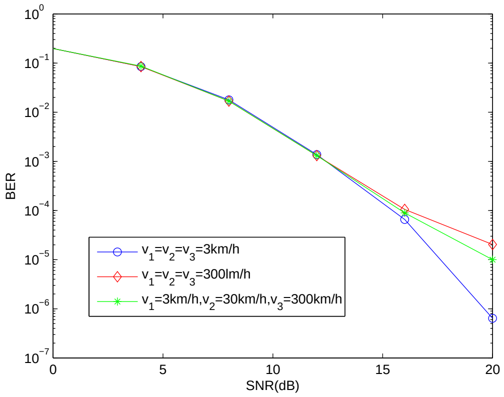


Figure 2.13: Effect of user's velocity on system's performance with three users.

During the simulations of the SFBC SC-FDMA system, a 2-slot interleave mapping is used where two consecutive sub-carriers are assigned to one user and

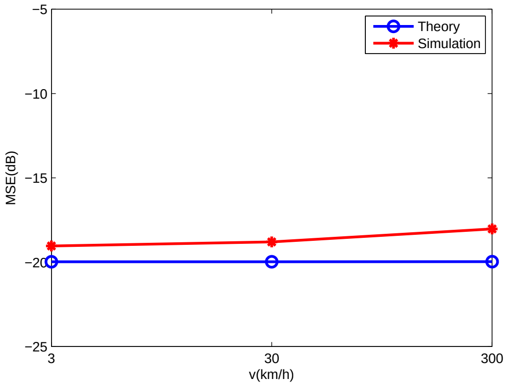


Figure 2.14: Theoretical and simulated MSE of RLS AFD-DFE as a function of user's velocity.

this is to satisfy condition (2.42) with the assumption that the channel does not change over two consecutive sub-carriers. SFBC is used with single user and the effect of the users's velocity on the system's performance is depicted in Fig. 2.15. In MIMO scenarios, independent 6-path Rayleigh fading channels are used for each transmit/receive antenna pair. Fig. 2.15 shows that great improvement in performance is obtained through the use of SFBC with 2-slot interleaving.

Finally, the performance of a two-receive antenna scenario when three users share the same frequency band and time slot with three other different users, is

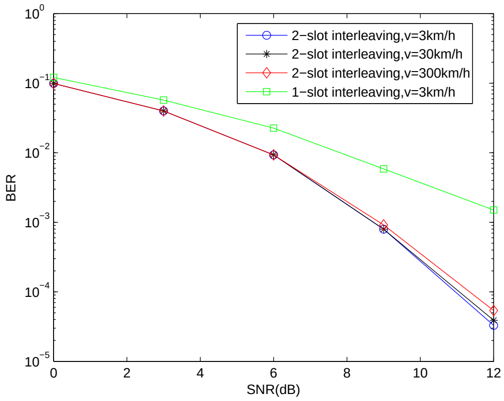


Figure 2.15: Effect of user's velocity on SFBC SC-FDMA system.

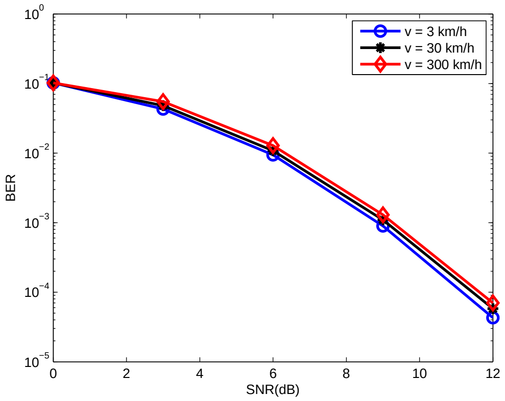


Figure 2.16: Effect of user's velocity on hybrid SM-SFBC SC-FDMA with two receive antennas and six users.

evaluated and this is reported in Fig. 2.16. All users are assumed to be moving with same velocities v. From the figure, it is clear that the AFD-DFE can separate co-channel users without sacrificing performance.

2.7 Conclusion

In this work, an adaptive frequency-domain DFE is proposed for the first time for SC-FDMA, SFBC SC-FDMA, SM SC-FDMA and hybrid SM-SFBC SFBC SC-FDMA systems with both feedforward and feedback filters operating in the frequency-domain. The equalizer operates without channel estimation at the receiver. The updating scheme used for the frequency-domain DFE is the RLS algorithm. The proposed algorithm is shown to have a low complexity and this is due to the special structure of the matrices involved in computing the weights of the feedforward ad feedback filters in the frequency-domain. The AFD-DFE is also more computationally efficient than non-adaptive frequency-domain DFE.

Simulation results for a time varying frequency-selective fading channel under the effect of high Doppler frequency and CFO on the system's performance are conducted and demonstrate the significant performance gain and robustness of the proposed algorithm.

CHAPTER 3

ADAPTIVE FREQUENCY-DOMAIN DECISION FEEDBACK EQUALIZATION USING CONSTRAINT-BASED RLS FOR UPLINK SC-FDMA

The Decision Feedback equalizer is well known to outperform a Linear Equalizer (LE) in highly frequency-selective fading channels. In this work, we develop a constraint-based block Recursive Least Squares (CRLS) for an Adaptive Frequency Domain Decision Feedback Equalizer (AFD-DFE) when used in an uplink Single Carrier Frequency Division Multiple Access (SC-FDMA) systems. For the AFD-DFE, both the feedforward and feedback filters are implemented in the frequency-domain, therefore, the complexity of the constraint-based block RLS can be reduced substantially when compared to its time-domain counterpart by exploiting the matrix structure in the frequency-domain. The performance of the CRLS algorithm is better than that of the RLS with no significant increase in the computational complexity. Moreover, we extend our design to the Space-Frequency Block code (SFBC) SC-FDMA system. We also show that the AFD-DFE not only enjoys a significant reduction in computational complexity when compared to the frequency-domain non-adaptive optimum MMSE DFE but its performance is also better than practical MMSE DFE (the one with error decisions) and close to ideal MMSE DFE (the one with correct decisions). Simulation results are carried out to demonstrate the robustness of our proposed algorithm to high Doppler. To mitigate Inter-Carrier Interference (ICI) due to large Carrier Frequency Offset (CFO), we have designed 3-tap per sub-carrier AFD-DFE by exploiting the banded and sparse structure of the channel, for Single-Input Single-Output (SISO) and SFBC SC-FDMA systems and show that it has an excellent performance as compared to 1-tap AFD-DFE and has a low computational complexity . Finally, it is shown that we can reduce the training symbols in each SC-FDMA block that are transmitted in the training phase with no significant performance degradation. To further reduce the overhead blind AFD-DFE is also introduced.

3.1 Introduction

Constraint-based algorithms help in achieving the better performance by putting some condition on the behavior of the algorithm. There are several examples of the constraint-based algorithms like normalized LMS [32], noise-constraint LMS [33], l_0 norm constraint LMS [34] and noise-constraint diffusion LMS [35], just to name a few. In time-domain DFE, the previous decisions are fed back but not the present one. To ensure that the present symbol is not canceled out, the feedback filter coefficient corresponding to that symbol is set to zero. Here, the frequency-domain version of DFE is used, therefore, to make sure that the present symbol is not canceled out in the feedback loop, the equalizer is constrained to follow this action. Thus achieving performance improvement as compared to the unconstraint AFD-DFE.

To improve reliability at the user terminal, a transmit diversity is employed in LTE-Advanced (LTE-A) [36]. A popular two transmit diversity technique was proposed by Alamouti [13], which has linear Maximum Likelihood (ML) decoding complexity with full diversity gain. The Alamouti's scheme is a special case of Space Time Block Codes (STBC) [14]. Alamouti's STBC can not be applied to SC-FDMA system, since in LTE the frames contain odd number of SC-FDMA symbols but for STBC this number should be even. Moreover, in STBC it is assumed that the channel remains constant for two SC-FDMA symbols which is not valid in the case of fast varying channels and consequently, performance degradation will result. An alternative to STBC is Space-Frequency Block Codes

(SFBC) [15]. In SFBC, the number of symbols in each frame is not necessary to be even but when applied to a SC-FDMA system it effects the low PAPR property. In [17] and [18], new schemes to deal with the aforementioned issues are proposed. However, their performances degrade at high Signal-to-Noise Ratio (SNR). In [37], an embedded SFBC technique is proposed which preserves low PAPR property of SC-FDMA as well as Alamouti's structure in case of Inter-Carrier Interference (ICI).

In this work, we propose an AFD-DFE (1-tap per subcarrier) where both the feedforward and feedback filters operate in the frequency-domain and use a constraint-based RLS for adaptation. The block RLS algorithm [5] is used as it is known to provide fast convergence/ tracking when compared to Least Mean Square (LMS) algorithm. Generally the complexity of the block RLS is high due to matrix inversion operation involved but when used in the frequency-domain, the inversion operation is simplified due to special structure of the matrices and, hence, a reduction in the complexity. The design is then extended to SFBC SC-FDMA and 3-tap AFD-DFE. In previous works, 1-tap per subcarrier frequency domain equalizer is used, which becomes highly suboptimal in presence of ICI. Therefore, we design 3-tap per subcarrier AFD-DFE by assuming banded and sparse structure of the channel matrix. To further improve the performance of the AFD-DFE, a 3-tap AFD-DFE is combined with the SFBC technique. It is also shown that the 3-tap AFD-DFE also exhibits low complexity, thanks again to the structure of the matrices. The AFD-DFE is complex when compared to the

AFD-LE but as we are working in uplink so the complex equalizer will be required at the Base Station (BS), where the power limitations are more relaxed than at the user terminals. Our designed constraint-based AFD-DFE is general and does not depend on mapping technique used. The complexity of the constraint-based AFD-DFE is almost similar to that of the AFD-DFE with standard RLS (without constraint) and performance wise the former is much better. We also demonstrated the robustness of our designed AFD-DFE to ICI due to high Doppler and showed that in case of large Carrier Frequency Offset (CFO), the 3-tap AFD-DFE performance is better. Furthermore, it is shown that in order to reduce the overhead in the training phase, we can use some symbols in SC-FDMA block for training and the rest for data without compromise of performance.

Adaptive equalization algorithms utilize a initial training sequence. However to save resources or when sending the training symbols are not possible, it is desirable to use an equalizer without the aid of a training sequence. Such an equalizer which does not utilize a training mode is known as blind equalizer. Designing an efficient and fully blind DFE remains a challenging task due to the decision errors in the feedback loop. Note that here we are using block DFE, which has certain advantages to be used as blind equalizer. First, error propagation is limited to one block. Therefore, by keeping the block size small, error propagation can be minimized. Second, in block DFE we have a problem of causality, which is solved by using an iterative procedure discussed in chapter 2. This iterative procedure gives best decisions as compared to symbol-by-symbol DFE thus minimizing the error propagation. These two facts are utilized to devise a Blind AFD-DFE (BAFD-DFE).

In summary, the main contributions of this chapter are

- Adaptive constraint RLS-based implementation of the AFD-DFE for SISO SC-FDMA.
- Extension to the SFBC scenarios.
- Reduced-Complexity implementation by exploiting matrix structure.
- Demonstration of performance superiority to LE, practical MMSE-DFE and LMS-based approaches.
- Reduced Complexity compared to the channel-estimate-based approach.
- Demonstrated robustness to Doppler.
- Extension to 3-tap AFD-DFE for SISO and SFBC cases to combat ICI.
- Reduction of overhead by using less training symbols in training phase and introduction of BAFD-DFE.

3.2 Recursive Least Squares with constraint (CRLS)

Lets denote the feedforward and feedback filter coefficients in the frequency-domain as $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{B}}$, respectively. Using the system model described in Section

(2.2), the output of the equalizer in the frequency-domain at instant k is given by

$$\check{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{Z}}_k \boldsymbol{\mathcal{F}}_{k-1} + \boldsymbol{\mathcal{D}}_k \boldsymbol{\mathcal{B}}_{k-1} \tag{3.1}$$

The coefficients of the feedforward and feedback filter are $\mathcal{F}(0), \mathcal{F}(1),\ ...\ \mathcal{F}(M-1)$ and $\mathcal{B}(0), \mathcal{B}(1),\ ...\ \mathcal{B}(M-1)$, respectively. The explicit knowledge of the filter coefficients is not needed for the development of the adaptive solution. The decision matrix $\boldsymbol{\mathcal{D}}$ is defined as

$$\boldsymbol{\mathcal{D}}_k = \begin{cases} diag(F_M \boldsymbol{x}_k), & \text{for training} \\ \\ diag(F_M \hat{\boldsymbol{x}}_k), & \text{for decision-directed} \end{cases}$$

Denoting $\boldsymbol{\mathcal{W}}_k = \left[\begin{smallmatrix} \boldsymbol{\mathcal{F}}_k \\ \boldsymbol{\mathcal{B}}_k \end{smallmatrix}\right]$. We express (3.1) as

$$\check{\boldsymbol{\mathcal{X}}}_k = [\boldsymbol{\mathcal{Z}}_k \quad \boldsymbol{\mathcal{D}}_k] \boldsymbol{\mathcal{W}}_{k-1} \tag{3.2}$$

Hence, the output of the equalizer in the time-domain is $\check{\boldsymbol{x}}_k = F_M^H \check{\boldsymbol{\mathcal{X}}}_k$. Here we will use a constraint to formulate a constraint-based least squares solution for DFE taps to be used in SC-FDMA system. This constraint is used to cancel out the pre and post cursers but not the desired component. To explain this, suppose the feedback filter weights in time-domain are $b_0, b_1, ..., b_L$. Since b_0 corresponds to the present symbol, which is not fed back to avoid self cancelation of the present symbol, therefore, $b_0 = 0$. In frequency domain, this can be translated to $\sum_{i=0}^{M-1} \mathcal{B}(i) = 0$. In the ensuing two methods are used to formulate the CRLS algorithm.

3.2.1 Case 1

In this case, the frequency-domain CRLS is developed from least squares problem. Assume we have available $k+1$ realizations of the matrices $\{\boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{D}}\}$. We collect the desired data and received data matrices as

$$\mathbf{\Pi} = \begin{bmatrix} \boldsymbol{\mathcal{Z}}_0 & \boldsymbol{\mathcal{D}}_0 \\ \boldsymbol{\mathcal{Z}}_1 & \boldsymbol{\mathcal{D}}_1 \\ \vdots & \vdots \\ \boldsymbol{\mathcal{Z}}_k & \boldsymbol{\mathcal{D}}_k \end{bmatrix} \tag{3.3}$$

and

$$\acute{\boldsymbol{\mathcal{D}}} = \begin{bmatrix} \boldsymbol{D}_0 \\ \boldsymbol{D}_1 \\ \vdots \\ \boldsymbol{D}_k \end{bmatrix} \tag{3.4}$$

where $\boldsymbol{D}_k = [D_k(0), D_k(1), ..., D_k(M-1)]^T$ is a vector containing the diagonal elements of $\boldsymbol{\mathcal{D}}_k$. First we will consider a combined cost function of the feedforward and feedback filters to be minimized. The constraint-based least squares problem for this case is given as

$$\min_{\boldsymbol{\mathcal{W}}} ||\acute{\boldsymbol{\mathcal{D}}} - \mathbf{\Pi}\boldsymbol{\mathcal{W}}||^2 \quad \text{subject to} \quad \sum_{i=0}^{M-1} \mathcal{B}(i) = 0 \tag{3.5}$$

where the notation $||.||^2$ denotes the squared Euclidean norm of its argument. To solve this optimization problem, we use the Lagrange multiplier method for the general case of complex valued data. According to this method, the objective function for the problem at hand consists of two terms, given on the right side of the equation

$$J = ||\acute{\boldsymbol{\mathcal{D}}} - \boldsymbol{\Pi}\boldsymbol{\mathcal{W}}||^2 + 2Re\{\alpha^* \sum_{i=0}^{M-1} \mathcal{B}(i)\} \tag{3.6}$$

where α is complex valued Lagrange multiplier and asterisk denotes complex conjugation. As the square of Euclidean norm $||\boldsymbol{\mathcal{D}} - \check{\boldsymbol{\mathcal{X}}}||^2$ is real-valued function. Therefore, real part operator, $Re\{.\}$, is applied to the second term to ensure that the contribution to the objective function is likewise real-valued. The objective function, J, is quadratic, as shown by expanding (3.6) into

$$J = \acute{\boldsymbol{\mathcal{D}}}^H \acute{\boldsymbol{\mathcal{D}}} - \acute{\boldsymbol{\mathcal{D}}}^H \boldsymbol{\Pi}\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{W}}^H \boldsymbol{\Pi}^H \acute{\boldsymbol{\mathcal{D}}} + \boldsymbol{\mathcal{W}}^H \boldsymbol{\Pi}^H \boldsymbol{\Pi}\boldsymbol{\mathcal{W}} + 2Re\{\alpha^* \boldsymbol{G}\boldsymbol{\mathcal{W}}\} \tag{3.7}$$

To find the least squares solution differentiate the objective function with respect to weight vector $\boldsymbol{\mathcal{W}}$. Then follow the rule of differentiating a real-valued function with respect to complex-valued vector [29], we get

$$\frac{\partial J}{\partial \boldsymbol{\mathcal{W}}} = -\acute{\boldsymbol{\mathcal{D}}}^H \boldsymbol{\Pi} + \boldsymbol{\mathcal{W}}^H \boldsymbol{\Pi}^H \boldsymbol{\Pi} + \alpha^* \boldsymbol{G} \tag{3.8}$$

where $\boldsymbol{G} = [\mathbf{0}_{1\times M} \quad \mathbf{1}_{1\times M}]$ and $\mathbf{0}_{1\times M}$ and $\mathbf{1}_{1\times M}$ are all zero and all ones row vector of size M, respectively. The final solution of (3.8) becomes

$$\boldsymbol{\mathcal{W}} \;=\; (\mathbf{\Pi}^H\mathbf{\Pi})^{-1}(\mathbf{\Pi}^H\acute{\boldsymbol{\mathcal{D}}} - \boldsymbol{\mathcal{G}}) \tag{3.9}$$

where $\boldsymbol{\mathcal{G}} = \alpha\boldsymbol{G}^T$. To update the least square solution (3.9) recursively, we proceed as follows: The time updated least squares problem is given as

$$\min_{\boldsymbol{\mathcal{W}}} ||\acute{\boldsymbol{\mathcal{D}}}_{k+1} - \mathbf{\Pi}_{k+1}\boldsymbol{\mathcal{W}}||^2 \quad \text{subject to} \quad \sum_{i=0}^{M-1}\mathcal{B}(i) = 0 \tag{3.10}$$

which has the following solution

$$\boldsymbol{\mathcal{W}}_{k+1} = (\mathbf{\Pi}_{k+1}^H\mathbf{\Pi}_{k+1})^{-1}(\mathbf{\Pi}_{k+1}^H\acute{\boldsymbol{\mathcal{D}}}_{k+1} - \boldsymbol{\mathcal{G}}_{k+1}) \tag{3.11}$$

To develop update scheme for the least squares solution (3.11) define $\mathbf{\Pi}_{k+1}$ and $\acute{\boldsymbol{\mathcal{D}}}_{k+1}$, respectively, as

$$\mathbf{\Pi}_{k+1} = \begin{bmatrix} \mathbf{\Pi}_k \\ \boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1} \end{bmatrix} \tag{3.12}$$

and

$$\acute{\boldsymbol{\mathcal{D}}}_{k+1} = \begin{bmatrix} \acute{\boldsymbol{\mathcal{D}}}_k \\ \boldsymbol{D}_{k+1} \end{bmatrix} \tag{3.13}$$

Note that

$$\mathbf{\Pi}_{k+1}^H\mathbf{\Pi}_{k+1} = \mathbf{\Pi}_k^H\mathbf{\Pi}_k + [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}] \tag{3.14}$$

and

$$\boldsymbol{\Pi}_{k+1}^H\acute{\boldsymbol{\mathcal{D}}}_{k+1} - \boldsymbol{\mathcal{G}}_{k+1} = \boldsymbol{\Pi}_k^H\acute{\boldsymbol{\mathcal{D}}}_k - \boldsymbol{\mathcal{G}}_k + [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H\boldsymbol{D}_{k+1} - \alpha_{k+1}\boldsymbol{G}^T \tag{3.15}$$

Lets introduce the matrix

$$\boldsymbol{P}_{k+1} = (\boldsymbol{\Pi}_{k+1}^H\boldsymbol{\Pi}_{k+1})^{-1} \tag{3.16}$$

Then (3.11) can be written in more compact form as

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{P}_{k+1}(\boldsymbol{\Pi}_{k+1}^H\acute{\boldsymbol{\mathcal{D}}}_{k+1} - \boldsymbol{\mathcal{G}}_{k+1}) \tag{3.17}$$

Putting (3.14) in (3.16), we get

$$\begin{aligned}\boldsymbol{P}_{k+1} &= (\boldsymbol{\Pi}_k^H\boldsymbol{\Pi}_k + [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}])^{-1} \\ &= (\boldsymbol{P}_k^{-1} + [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}])^{-1} \end{aligned} \tag{3.18}$$

and

$$\boldsymbol{P}_{k+1}^{-1} = \boldsymbol{P}_k^{-1} + [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}] \tag{3.19}$$

Using the matrix inversion identity [5], it can easily be shown that

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{P}_k[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H\boldsymbol{\beta}[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]\boldsymbol{P}_k \tag{3.20}$$

where $\boldsymbol{\beta} = (\boldsymbol{I}_M + [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]\boldsymbol{P}_k[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H)^{-1}$ Putting (3.20) and (3.15) in (3.17), we get

$$\begin{aligned}
\boldsymbol{\mathcal{W}}_{k+1} &= \left(\boldsymbol{P}_k - \boldsymbol{P}_k[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H \boldsymbol{\beta} [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]\boldsymbol{P}_k\right)(\boldsymbol{\Pi}_k^H \acute{\boldsymbol{\mathcal{D}}}_k - \boldsymbol{\mathcal{G}}_k \\
&\quad + [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H \boldsymbol{D}_{k+1} - \alpha_k \boldsymbol{G}^T) \\
&= \underbrace{\boldsymbol{P}_k \boldsymbol{\Pi}_k^H \acute{\boldsymbol{\mathcal{D}}}_k - \boldsymbol{P}_k \boldsymbol{\mathcal{G}}_k}_{\boldsymbol{\mathcal{W}}_k} - \left(\boldsymbol{P}_k[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H \boldsymbol{\beta} [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]\right) \\
&\quad \times \underbrace{\boldsymbol{P}_k(\boldsymbol{\Pi}_k^H \acute{\boldsymbol{\mathcal{D}}}_k - \boldsymbol{\mathcal{G}}_k)}_{\boldsymbol{\mathcal{W}}_k} + \boldsymbol{P}_k[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H \\
&\quad \left(\boldsymbol{I}_M - \boldsymbol{\beta}[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]\boldsymbol{P}_k[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H\right) \boldsymbol{D}_{k+1} - \boldsymbol{P}_{k+1}(\alpha_{k+1} \boldsymbol{G}^T)
\end{aligned} \tag{3.21}$$

After rearranging, we get

$$\begin{aligned}
\boldsymbol{\mathcal{W}}_{k+1} &= \boldsymbol{\mathcal{W}}_k + \left(\boldsymbol{P}_k[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H \boldsymbol{\beta}\right)(\boldsymbol{D}_{k+1} - [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]\boldsymbol{\mathcal{W}}_k) \\
&\quad - \boldsymbol{P}_{k+1}(\alpha_{k+1} \boldsymbol{G}^T) \\
&= \boldsymbol{\mathcal{W}}_k + \boldsymbol{P}_{k+1}[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H(\boldsymbol{D}_{k+1} - [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]\boldsymbol{\mathcal{W}}_k) \\
&\quad - \boldsymbol{P}_{k+1}(\alpha_{k+1} \boldsymbol{G}^T)
\end{aligned} \tag{3.22}$$

it can easily be prove that

$$\boldsymbol{P}_{k+1}[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H = \left(\boldsymbol{P}_k[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H \boldsymbol{\beta}\right) \tag{3.23}$$

Finally,

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + \boldsymbol{P}_{k+1}([\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H(\boldsymbol{D}_{k+1} - [\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]\boldsymbol{\mathcal{W}}_k) - \alpha_{k+1}\boldsymbol{G}^T) \tag{3.24}$$

If we use the exponentially-weighted RLS, then (3.20) becomes

$$\boldsymbol{P}_{k+1} = \lambda^{-1}\left[\boldsymbol{P}_k - \lambda^{-1}\boldsymbol{P}_k[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]^H \boldsymbol{\beta}[\boldsymbol{\mathcal{Z}}_{k+1} \quad \boldsymbol{\mathcal{D}}_{k+1}]\boldsymbol{P}_k\right] \tag{3.25}$$

where $0 \ll \lambda \leq 1$. The multiplier α_{k+1} is updated according to stochastic gradient method as

$$\alpha_{k+1} = \alpha_k + \mu \left(\frac{\partial J}{\partial \alpha}\right)_k^* \tag{3.26}$$

Differentiating (3.6) with respect to α, we get

$$\frac{\partial J}{\partial \alpha} = \left(\sum_{i=0}^{M-1} \mathcal{B}(i)\right)^* \tag{3.27}$$

Therefore, (3.26), becomes

$$\alpha_{k+1} = \alpha_k + \mu \sum_{i=0}^{M-1} \mathcal{B}_k(i) \tag{3.28}$$

3.2.2 Case 2

In this case, we motivate the RLS algorithm as simply a stochastic gradient method. The Mean Square Error (MSE) at the i^{th} frequency bin is given as

$$MSE(i) = E[|D(i) - \check{X}(i)|^2] \tag{3.29}$$

where $E[.]$ stands for the expectation operation. Using the constraint the cost function becomes

$$J = E[|D(i) - \check{X}(i|^2] + 2Re[\alpha^* \sum_{i=0}^{M-1} \mathcal{B}(i)] \tag{3.30}$$

Expanding the cost function and ignoring the expectation as we are using stochastic gradient method, we get

$$\begin{aligned} J &= D^*(i)D(i) - D^*(i)\mathcal{Y}(i)\mathcal{F}(i) - D^*(i)D(i)\mathcal{B}(i) - \mathcal{F}^*(i)\mathcal{Y}^*(i)D(i) \\ &- \mathcal{B}^*(i)D^*(i)D(i) + \mathcal{F}^*(i)\mathcal{Y}^*(i)\mathcal{Y}(i)\mathcal{F}(i) + \mathcal{F}^*(i)\mathcal{Y}(i)D(i)\mathcal{B}(i) \\ &+ \mathcal{B}^*(i)D^*(i)\mathcal{Y}(i)\mathcal{F}(i) + \mathcal{B}^*(i)D^*(i)D(i)\mathcal{B}(i) + 2Re[\alpha^* \sum_{i=0}^{M-1} \mathcal{B}(i)] \end{aligned} \tag{3.31}$$

Minimizing (3.31) for the feedforward filter and the feedback filter separately, results in the following updates at instant $(k+1)$

$$\begin{aligned}
\mathcal{F}_{k+1}(i) &= \mathcal{F}_k(i) - \mu^1_{k+1}(i)\left(\frac{\partial J}{\partial \mathcal{F}(i)}\right)^*_k \\
&= \mathcal{F}_k(i) + \mu^1_{k+1}(i)\mathcal{Y}^*_{k+1}(i)\{D_{k+1}(i) - [\mathcal{Y}_{K+1}(i)\mathcal{F}_k(i) + D_{k+1}(i)\mathcal{B}_k(i)]\}
\end{aligned} \tag{3.32}$$

and

$$\begin{aligned}
\mathcal{B}_{k+1}(i) &= \mathcal{B}_k(i) - \mu^2_{k+1}(i)\left(\frac{\partial J}{\partial \mathcal{B}(i)}\right)^*_k \\
&= \mathcal{B}_k(i) + \mu^2_{k+1}(i)D^*_{k+1}(i)\{D_{k+1}(i) - [\mathcal{Y}_{k+1}(i)\mathcal{F}_k(i) + D_{k+1}(i)\mathcal{B}_k(i)]\} \\
&\quad -\mu^2_{k+1}(i)\alpha_{k+1}
\end{aligned} \tag{3.33}$$

where $\mu(i)^1_{k+1}$ and $\mu(i)^2_{k+1}$ are the time varying step sizes, given as

$$\mu(i)^1_{k+1} = \frac{\mu_{k+1}}{\epsilon_{k+1} + E[\mathcal{Y}(i)^*\mathcal{Y}(i)]}, \quad \mu(i)^2_{k+1} = \frac{\mu_{k+1}}{\epsilon_{k+1} + E[D^*(i)D(i)]}$$

Next, we replace $E[\mathcal{Y}(i)^*\mathcal{Y}(i)]$ and $E[D^*(i)D(i)]$ by their estimates, which for RLS update, are choosen as the exponentially-weighted sample average

$$E[\mathcal{Y}(i)^*\mathcal{Y}(i)] = \frac{1}{(k+2)}\sum_{j=0}^{k+1}\lambda^{k+1-j}\mathcal{Y}_j(i)^*\mathcal{Y}_j(i)) \tag{3.34}$$

$$E[D(i)^*D(i)] = \frac{1}{(k+2)}\sum_{j=0}^{k+1}\lambda^{k+1-j}D_j(i)^*D_j(i)) \tag{3.35}$$

for some scalar $0 \ll \lambda \leq 1$. Equations (3.34) and (3.35) amount to averaging all past regressors up to time $k+1$. Now choosing the step size as $\mu_{k+1} = 1/(k+2)$ and the regularization factor as $\epsilon_{k+1} = \lambda^{k+2}\epsilon/(k+2)$, and collecting all the coefficients in one vector $\boldsymbol{\mathcal{W}}$, (3.32) and (3.33) become

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + \left[\lambda^{k+2}\epsilon \boldsymbol{I}_{2M} + \sum_{j=0}^{k+1} \lambda^{k+1-j} \boldsymbol{\mathcal{A}}_j^H \boldsymbol{\mathcal{A}}_j\right]^{-1} (\boldsymbol{\mathcal{A}}_{k+1}^H \boldsymbol{\mathcal{E}}_{k+1} - \alpha_{k+1}\boldsymbol{G}^T) \quad (3.36)$$

where $\boldsymbol{\mathcal{A}}_k$ and $\boldsymbol{\mathcal{E}}_k$ are given as

$$\boldsymbol{\mathcal{A}}_k = \begin{bmatrix} \boldsymbol{\mathcal{Z}}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{D}}_k \end{bmatrix} \quad (3.37)$$

and

$$\boldsymbol{\mathcal{E}}_k = \begin{bmatrix} \boldsymbol{D}_k - \check{\boldsymbol{\mathcal{X}}}_k \\ \boldsymbol{D}_k - \check{\boldsymbol{\mathcal{X}}}_k \end{bmatrix} \quad (3.38)$$

It is not convenient to find the inverse of the matrix in (3.36) as it requires to combine all the previous and present data to form the matrix. Therefore, we define

$$\Theta_{k+1} \triangleq \left(\lambda^{k+2}\epsilon \boldsymbol{I}_{2M} + \sum_{j=0}^{k+1} \lambda^{k+1-j} \boldsymbol{\mathcal{A}}_j^H \boldsymbol{\mathcal{A}}_j\right) \quad (3.39)$$

which satisfies the following recursion

$$\Theta_{k+1} = \lambda\Theta_k + \boldsymbol{\mathcal{A}}_{k+1}^H \boldsymbol{\mathcal{A}}_{k+1}, \quad \Theta_0 = \epsilon \boldsymbol{I}_{2M} \quad (3.40)$$

Let $\boldsymbol{\mathcal{P}}_{k+1} = \Theta_{k+1}^{-1}$ and applying the matrix inversion lemma [5] gives

$$\boldsymbol{\mathcal{P}}_{k+1} = \lambda^{-1}[\boldsymbol{\mathcal{P}}_k - \lambda^{-1}\boldsymbol{\mathcal{P}}_k\boldsymbol{\mathcal{A}}_{k+1}^H \times (\boldsymbol{I}_{2M} + \lambda^{-1}\boldsymbol{\mathcal{A}}_{k+1}\boldsymbol{\mathcal{P}}_k\boldsymbol{\mathcal{A}}_{k+1}^H)^{-1}\boldsymbol{\mathcal{A}}_{k+1}\boldsymbol{\mathcal{P}}_k] \quad (3.41)$$

where λ is the forgetting factor chosen close to 1. Finally the RLS update is given as

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + \boldsymbol{\mathcal{P}}_{k+1}(\boldsymbol{\mathcal{A}}_{k+1}^H\boldsymbol{\mathcal{E}}_{k+1} - \alpha_{k+1}\boldsymbol{G}^T) \quad (3.42)$$

with $\boldsymbol{\mathcal{E}}_k$ define as in (3.38). Initially $\boldsymbol{\mathcal{W}}_0 = \boldsymbol{0}$ and $\boldsymbol{\mathcal{P}}_0 = \epsilon^{-1}\boldsymbol{I}_{2M}$. α_k is updated according to (3.28).

As can be seen that $\boldsymbol{\mathcal{P}}_k$ in case 2 gives a much better estimate of input covariance matrix for each filter as it takes in to account the input of the feedforward filter and feedback filter separately, therefore, its performance is better than case 1. Also the decision error does not effect the feedforward filter in case 2 unlike case 1. This claim is proved later using simulation. The computational complexity of case 2 is also low as it does not require matrix inversion unlike case 1. Due to these two reasons, later case will be only considered in the study.

3.2.3 Reduced-Complexity CRLS Update

Due to the special structure in $\boldsymbol{\mathcal{P}}_{k+1}$, it turns out that no matrix inversion is required for computing $\boldsymbol{\mathcal{P}}_{k+1}$ resulting in a significant reduction in computational complexity. Following the same procedure as done in Section (2.3.2), it can be shown that the final reduced-complexity CRLS update has the form

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + diag([\boldsymbol{P}^1_{k+1} \quad \boldsymbol{P}^2_{k+1}])(\boldsymbol{\mathcal{A}}^H_{k+1}\boldsymbol{\mathcal{E}}_{k+1} - \alpha_{k+1}\boldsymbol{G}^T) \tag{3.43}$$

and constraint is updated using (3.28). The update procedure for AFD-DFE with CRLS is shown in Table 3.1.

3.2.4 Reduced-training AFD-DFE

Instead of using M training symbols in one SC-FDMA block, the training overhead can be reduced by using T training and $M - T$ modulated symbols. Now by introducing the iterative procedure in training mode we can interpolate the rest of the weights. The update procedure in reduced-training mode is shown in Table 3.1. For BAFD-DFE, the algorithm is used in decision-directed mode by utilizing stop and go blind equalization algorithm [38].

3.3 Carrier Frequency Offset (CFO) in SC-FDMA

In the above description, perfect frequency synchronization has been assumed between the transmitter and the receiver. However, CFO arises in practical SC-FDMA systems due to transmitter/receiver frequency oscillators' misalignment and causes interference (energy leakage) from neighboring sub-carriers.

Let the m^{th} user's CFO normalized by the sub-carrier spacing, be denoted by

Table 3.1: Summary of the adaptation algorithm for AFD-DFE

Initialization:

Initialize the algorithm by setting

$$\boldsymbol{\mathcal{W}}_0 = \mathbf{0}$$

$$\alpha_0 = 0$$

λ is close to one

and $$\boldsymbol{\mathcal{P}}_0 = \begin{bmatrix} \epsilon_{\mathcal{F}}^{-1}\boldsymbol{I}_M & \mathbf{0} \\ \mathbf{0} & \epsilon_{\mathcal{B}}^{-1}\boldsymbol{I}_M \end{bmatrix}$$

For each instant of time, k=0,1,2,...

In training mode:

(1) Update α_{k+1} as

$$\alpha_{k+1} = \alpha_k + \mu \textstyle\sum_{i=0}^{M-1} \mathcal{B}_k(i)$$

(2) Update $\boldsymbol{P}_{k+1}^1$ and $\boldsymbol{P}_{k+1}^2$ via

$$\boldsymbol{P}_{k+1}^1 = \lambda^{-1}[\boldsymbol{P}_k^1 - \lambda^{-1}\boldsymbol{P}_k^1(|\boldsymbol{\mathcal{Z}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}_k^1)^{-1}\boldsymbol{P}_k^1]$$

$$\boldsymbol{P}_{k+1}^2 = \lambda^{-1}[\boldsymbol{P}_k^2 - \lambda^{-1}\boldsymbol{P}_k^2(|\boldsymbol{\mathcal{X}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}_k^2)^{-1}\boldsymbol{P}_k^2]$$

(3) Update the equalizer weights $\boldsymbol{\mathcal{W}}_{k+1}$ recursively as

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + diag([\boldsymbol{P}_{k+1}^1 \quad \boldsymbol{P}_{k+1}^2])(\boldsymbol{\mathcal{A}}_{k+1}^H\boldsymbol{\mathcal{E}}_{k+1} - \alpha_{k+1}\boldsymbol{G}^T)$$

In decision-directed mode:

(1) Iterate on $\check{\boldsymbol{\mathcal{X}}}_{k+1} = \boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{\mathcal{F}}_k + \boldsymbol{\mathcal{D}}_{k+1}\boldsymbol{\mathcal{B}}_k$

(2) Update α_{k+1} as

$$\alpha_{k+1} = \alpha_k + \mu \textstyle\sum_{i=0}^{M-1} \mathcal{B}_k(i)$$

(3) Update $\boldsymbol{P}_{k+1}^1$ and $\boldsymbol{P}_{k+1}^2$ via

$$\boldsymbol{P}_{k+1}^1 = \lambda^{-1}[\boldsymbol{P}_k^1 - \lambda^{-1}\boldsymbol{P}_k^1(|\boldsymbol{\mathcal{Z}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}_k^1)^{-1}\boldsymbol{P}_k^1]$$

$$\boldsymbol{P}_{k+1}^2 = \lambda^{-1}[\boldsymbol{P}_k^2 - \lambda^{-1}\boldsymbol{P}_k^2(|\check{\boldsymbol{\mathcal{X}}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}_k^2)^{-1}\boldsymbol{P}_k^2]$$

(4) Update the equalizer weights $\boldsymbol{\mathcal{W}}_{k+1}$ recursively as

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + diag([\boldsymbol{P}_{k+1}^1 \quad \boldsymbol{P}_{k+1}^2])(\boldsymbol{\mathcal{A}}_{k+1}^H\boldsymbol{\mathcal{E}}_{k+1} - \alpha_{k+1}\boldsymbol{G}^T)$$

Ω_m where $0 \leq \Omega_m \leq 0.5$. Now, define a diagonal matrix to characterize the effect of CFO as $\boldsymbol{C}^{(m)} = diag([e^{j2\pi\Omega_m\times 0/N}, e^{j2\pi\Omega_m\times 1/N}, ..., e^{j2\pi\Omega_m\times (N-1)/N}])$. In this case, the pre-DFT received signal can be expressed as

Table 3.2: Summary of the reduced-training adaptation algorithm for AFD-DFE

For each instant of time, k=0,1,2,...

In training mode:

(1) Iterate on $\check{\boldsymbol{\mathcal{X}}}_{k+1} = \boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{\mathcal{F}}_k + \boldsymbol{\mathcal{D}}_{k+1}\boldsymbol{\mathcal{B}}_k$ with SC-FDMA block containing T training symbols and $M-T$ modulated data symbols

(2) Update α_{k+1} as

$\alpha_{k+1} = \alpha_k + \mu \sum_{i=0}^{M-1} \mathcal{B}_k(i)$

(3) Update $\boldsymbol{P}^1_{k+1}$ and $\boldsymbol{P}^2_{k+1}$ via

$\boldsymbol{P}^1_{k+1} = \lambda^{-1}[\boldsymbol{P}^1_k - \lambda^{-1}\boldsymbol{P}^1_k(|\boldsymbol{\mathcal{Z}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}^1_k)^{-1}\boldsymbol{P}^1_k]$

$\boldsymbol{P}^2_{k+1} = \lambda^{-1}[\boldsymbol{P}^2_k - \lambda^{-1}\boldsymbol{P}^2_k(|\check{\boldsymbol{\mathcal{X}}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}^2_k)^{-1}\boldsymbol{P}^2_k]$

(4) Update the equalizer weights $\boldsymbol{\mathcal{W}}_{k+1}$ recursively as

$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + diag([\boldsymbol{P}^1_{k+1} \quad \boldsymbol{P}^2_{k+1}])(\boldsymbol{\mathcal{A}}^H_{k+1}\boldsymbol{\mathcal{E}}_{k+1} - \alpha_{k+1}\boldsymbol{G}^T)$

$$\boldsymbol{y} = \sum_{m=1}^{K} \boldsymbol{C}^{(m)}(\boldsymbol{s}^{(m)} \circledast \boldsymbol{h}^{(m)}) + \boldsymbol{n}^{(m)} \tag{3.44}$$

After applying the N-point DFT, the received signal is given by

$$\acute{\boldsymbol{\mathcal{Y}}} = \sum_{m=1}^{K} \boldsymbol{\mathcal{C}}^{(m)}\hat{\boldsymbol{\Lambda}}^{(m)}\boldsymbol{R}^{(m)}\boldsymbol{\mathcal{X}}^{(m)} + \boldsymbol{\mathcal{N}} \tag{3.45}$$

where $\boldsymbol{\mathcal{C}}^{(m)}$ is a circulant matrix with entries $\boldsymbol{\mathcal{C}}^{(m)}_{p,q} = \frac{1}{N}\sum_{n=0}^{N-1} e^{j2\pi(\Omega^{(m)}+p-q)n/N}$, $p, q = 1, ..., N$. It is important to note here that the channel matrix $\boldsymbol{\mathcal{C}}^{(m)}\hat{\boldsymbol{\Lambda}}^{(m)}$ has structure shown in Fig. 3.1 , which shows that most of the energy of this matrix is in its three main diagonals. We assume that except three main diagonals all other entries are zero and based on this structure we formulate our three tap equalizer in the frequency-domain. After demapping, the m^{th} user's received signal is $\boldsymbol{\mathcal{Y}}^{(m)} = \boldsymbol{R}^{(m)T}\acute{\boldsymbol{\mathcal{Y}}}$. To simplify the notation we will

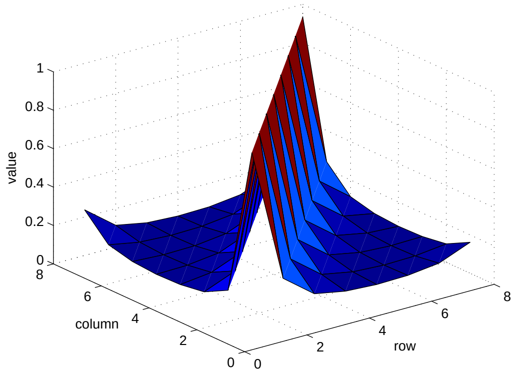


Figure 3.1: Structure of $\boldsymbol{\mathcal{C}}^{(m)}\hat{\boldsymbol{\Lambda}}^{(m)}$(normalized) matrix.

ignore the superscript m and $\boldsymbol{\mathcal{Y}} = [\mathcal{Y}(0), \mathcal{Y}(1), ..., \mathcal{Y}(M-1)]^T$. Now assuming that the equalizer tap matrix has similar structure as channel matrix, i.e. we have only three main diagonals then output of the MMSE 3-tap LE is given as

$$\begin{bmatrix} \check{X}(0) \\ \check{X}(1) \\ \check{X}(2) \\ \vdots \\ \check{X}(M-1) \end{bmatrix} = \begin{bmatrix} \mathcal{F}_1(0) & \mathcal{F}_2(0) & & & \\ \mathcal{F}_1(1) & \mathcal{F}_2(1) & \mathcal{F}_3(1) & & \\ & \mathcal{F}_1(2) & \ddots & \ddots & \\ & & \ddots & \ddots & \mathcal{F}_3(M-2) \\ & & & \mathcal{F}_1(M-1) & \mathcal{F}_2(M-1) \end{bmatrix} \begin{bmatrix} \mathcal{Y}(0) \\ \mathcal{Y}(1) \\ \mathcal{Y}(2) \\ \vdots \\ \mathcal{Y}(M-1) \end{bmatrix} \tag{3.46}$$

where $\mathcal{F}_i(j)$ represents the tap of the LE. and for DFE, we can write

$$\begin{bmatrix} \check{X}(0) \\ \check{X}(1) \\ \check{X}(2) \\ \vdots \\ \check{X}(M-1) \end{bmatrix} = \begin{bmatrix} \mathcal{F}_1(0) & \mathcal{F}_2(0) & & & \\ \mathcal{F}_1(1) & \mathcal{F}_2(1) & \mathcal{F}_3(1) & & \\ & \mathcal{F}_1(2) & \ddots & \ddots & \\ & & \ddots & \ddots & \mathcal{F}_3(M-2) \\ & & & \mathcal{F}_1(M-1) & \mathcal{F}_2(M-1) \end{bmatrix} \begin{bmatrix} \mathcal{Y}(0) \\ \mathcal{Y}(1) \\ \mathcal{Y}(2) \\ \vdots \\ \mathcal{Y}(M-1) \end{bmatrix} + \begin{bmatrix} \mathcal{B}(0) & & & & \\ & \mathcal{B}(1) & & & \\ & & \mathcal{B}(2) & & \\ & & & \ddots & \\ & & & & \mathcal{B}(M-1) \end{bmatrix} \begin{bmatrix} D(0) \\ D(1) \\ D(2) \\ \vdots \\ D(M-1) \end{bmatrix} \tag{3.47}$$

where $\mathcal{F}_i(j)$ and $\mathcal{B}_i(j)$ represent the tap coefficient of the feedforward and feedback filter, respectively. However, to develop an adaptive solution for these taps, we do not need their explicit solution. Denoting $\boldsymbol{\mathcal{U}}_i = [\boldsymbol{\mathcal{Y}}(i-1)\ \boldsymbol{\mathcal{Y}}(i)\ \boldsymbol{\mathcal{Y}}(i+1)]$ for $i = 1, 2, ...M-2$, $\boldsymbol{\mathcal{U}}_0 = [\boldsymbol{\mathcal{Y}}(0)\ \boldsymbol{\mathcal{Y}}(1)]$ and $\boldsymbol{\mathcal{U}}_{M-1} = [\boldsymbol{\mathcal{Y}}(M-2)\ \boldsymbol{\mathcal{Y}}(M-1)]$, (3.47) can be rewritten alternatively as

$$\begin{bmatrix} \check{X}(0) \\ \check{X}(1) \\ \check{X}(2) \\ \vdots \\ \check{X}(M-1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mathcal{U}}_0 & & & & \\ & \boldsymbol{\mathcal{U}}_1 & & & \\ & & \boldsymbol{\mathcal{U}}_2 & & \\ & & & \ddots & \\ & & & & \boldsymbol{\mathcal{U}}_{M-1} \end{bmatrix} \begin{bmatrix} \mathcal{F}_1(0) \\ \mathcal{F}_2(0) \\ \mathcal{F}_1(1) \\ \vdots \\ \mathcal{F}_2(M-1) \end{bmatrix}$$

$$+ \begin{bmatrix} D(0) & & & & \\ & D(1) & & & \\ & & D(2) & & \\ & & & \ddots & \\ & & & & D(M-1) \end{bmatrix} \begin{bmatrix} \mathcal{B}(0) \\ \mathcal{B}(1) \\ \mathcal{B}(2) \\ \vdots \\ \mathcal{B}(M-1) \end{bmatrix} \tag{3.48}$$

which can be written in compact notation, at instant k, as

$$\check{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{Z}}_k \boldsymbol{\mathcal{F}}_{k-1} + \boldsymbol{\mathcal{D}}_k \boldsymbol{\mathcal{B}}_{k-1} \tag{3.49}$$

Defining $\boldsymbol{\mathcal{A}}_k$, $\boldsymbol{\mathcal{E}}_k$ and $\boldsymbol{\mathcal{P}}_k$ as in (3.37), (3.38) and (3.41), respectively, the RLS update is given as in (3.43).

Reduced-Complexity 3-tap CRLS AFD-DFE: Again, we can show that the computational complexity is significantly reduced and no matrix inversion is required. We explain the rationale behind the complexity reduction. Starting with $k = 0$ and using $\boldsymbol{P}_0^1 = \epsilon^{-1}\boldsymbol{I}_{3M-2}$, $\boldsymbol{P}_1^1$ is given by

$$\begin{aligned} \boldsymbol{P}_1^1 &= \lambda^{-1}[\boldsymbol{P}_0^1 - \lambda^{-1}\boldsymbol{P}_0^1\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_M + \lambda^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{P}_0^1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{P}_0^1] \\ &= \lambda^{-1}[\epsilon^{-1}\boldsymbol{I}_{3M-2} - \lambda^{-1}\epsilon^{-1}\boldsymbol{I}_{3M-2}\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_M + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1\epsilon^{-1}\boldsymbol{I}_{3M-2}] \end{aligned} \tag{3.50}$$

It can easily be seen that $\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H = diag[|\boldsymbol{\mathcal{U}}_{0,1}|^2 \ |\boldsymbol{\mathcal{U}}_{1,1}|^2 \ ... \ |\boldsymbol{\mathcal{U}}_{M-1,1}|^2]$ and (3.50) does not require matrix inversion. Now

$$\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_M + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1 = \begin{bmatrix} \varnothing_0 & & & \\ & \varnothing_1 & & \\ & & \ddots & \\ & & & \varnothing_{M-1} \end{bmatrix} \tag{3.51}$$

where as the entries $\varnothing_i, i = 1, ..., M-2$ are 3×3 matrices and $\varnothing_i, i = 0, M-1$ are 2×2 matrices given by $\varnothing_i = \boldsymbol{\mathcal{U}}_{i,1}^H(1 + \lambda^{-1}\epsilon^{-1}|\boldsymbol{\mathcal{U}}_{i,1}|^2)\boldsymbol{\mathcal{U}}_{i,1}, i = 0, ..., M-1$. Now $\boldsymbol{P}_1^1$ has a structure shown below.

$$\boldsymbol{P}_1^1 = \begin{bmatrix} \boldsymbol{P}_{1,0}^1 & & & \\ & \boldsymbol{P}_{1,1}^1 & & \\ & & \ddots & \\ & & & \boldsymbol{P}_{1,M-1}^1 \end{bmatrix} \tag{3.52}$$

where $\boldsymbol{P}_{1,i}^1 = \lambda^{-1}[\epsilon^{-1}\boldsymbol{I}_d - \lambda^{-1}\epsilon^{-2}\varnothing_i]$, $d = 2$ for $i = 0, M-1$ and $d = 3$ for

$i = 1, ..., M-2$. Proceeding for $k = 1$, we have

$$\boldsymbol{P}_2^1 = \lambda^{-1}[\boldsymbol{P}_1^1 - \lambda^{-1}\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H(\boldsymbol{I}_M + \lambda^{-1}\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H)^{-1}\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1]$$

where $\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H = diag[\boldsymbol{\mathcal{U}}_{0,2}\boldsymbol{P}_{1,0}^1\boldsymbol{\mathcal{U}}_{0,2}^H \quad \boldsymbol{\mathcal{U}}_{1,2}\boldsymbol{P}_{1,1}^1\boldsymbol{\mathcal{U}}_{1,2}^H \quad ... \quad \boldsymbol{\mathcal{U}}_{M-1,2}\boldsymbol{P}_{1,M-1}^1\boldsymbol{\mathcal{U}}_{M-1,2}^H]$ and $\boldsymbol{\mathcal{U}}_{i,2}\boldsymbol{P}_{1,i}^1\boldsymbol{\mathcal{U}}_{i,2}^H$ is a scalar quantity, therefore, matrix inversion is just M scalar inversions. For $k > 1$, $\boldsymbol{P}_k^1$ has similar structure and, therefore, avoid matrix inversion. Moreover, for $\boldsymbol{P}_k^2$ same reasons are valid as for the 1-tap AFD-DFE case.

3.4 Space-Frequency Block Coded (SFBC) SC-FDMA

In this section we will discuss Embedded SFBC (E-SFBC) technique. In the presence of CFO and high Doppler, severe ICI from adjacent carriers occurs which destroys the Alamouti structure and results in performance degradation. Therefore, [37] proposes embedded SFBC which preserves the Alamouti structure even when there is ICI and also this technique does not effect the low PAPR property of SC-FDMA unlike Conventional SFBC (C-SFBC). For design of our AFD-DFE we implement the embedded SFBC at block level. Here we will not use the pilots and divide the block in to two. In embedded SFBC, we define $\boldsymbol{\mathcal{X}}_1^{(m)} = [X(0)^{(m)}, X(2)^{(m)}, ..., X(M-2)^{(m)}]^T$ and $\boldsymbol{\mathcal{X}}_2^{(m)} = [X(1)^{(m)}, X(3)^{(m)}, ..., X(M-1)^{(m)}]^T$ i.e. $\boldsymbol{\mathcal{X}}^{(m)}$ is divided into two blocks. Now the sequence to transmit these sub-blocks will be $\acute{\boldsymbol{\mathcal{X}}}_1 = \begin{bmatrix} \boldsymbol{\mathcal{X}}_1^{(m)} \\ -\boldsymbol{\mathcal{X}}_2^{*(m)} \end{bmatrix}$ and $\acute{\boldsymbol{\mathcal{X}}}_2 = \begin{bmatrix} \boldsymbol{\mathcal{X}}_2^{(m)} \\ \boldsymbol{\mathcal{X}}_1^{*(m)} \end{bmatrix}$ for antenna 1 and 2, respectively. After mapping and applying N-point IDFT, the transmitted signals from the two antennas are $\boldsymbol{s}_1^{(m)}$ and $\boldsymbol{s}_2^{(m)}$ corresponding to $\acute{\boldsymbol{\mathcal{X}}}_1^{(m)}$ and $\acute{\boldsymbol{\mathcal{X}}}_2^{(m)}$. The transmitted signals are circularly convolved with their respective channels and the received signal, after applying the N-DFT become

$$\acute{\boldsymbol{\mathcal{Y}}} = \sum_{m=1}^{K} \{\hat{\boldsymbol{\Lambda}}_1^{(m)} \boldsymbol{R}^{(m)} \acute{\boldsymbol{\mathcal{X}}}_1 + \hat{\boldsymbol{\Lambda}}_2^{(m)} \boldsymbol{R}^{(m)} \acute{\boldsymbol{\mathcal{X}}}_2\} + \boldsymbol{\mathcal{N}} \tag{3.53}$$

where $\hat{\boldsymbol{\Lambda}}_i^{(m)}$ is a $N \times N$ diagonal matrix, i.e., $\hat{\boldsymbol{\Lambda}}_i^{(m)} = diag(DFT(\boldsymbol{h}_i^{(m)}))$ for $i = 1, 2$ and $\boldsymbol{\mathcal{N}}$ is noise component with variance $\sigma_{\mathcal{N}} \boldsymbol{I}_N$. The received signal for m^{th} user, after demapping, can be expressed as

$$\boldsymbol{\mathcal{Y}}^{(m)} = \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}_1^{(m)} \boldsymbol{R}^{(m)} [\boldsymbol{\mathcal{X}}_1^{(m)} \; -\boldsymbol{\mathcal{X}}_2^{*(m)}]^T + \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}_2^{(m)} \boldsymbol{R}^{(m)} [\boldsymbol{\mathcal{X}}_2^{(m)} \; \boldsymbol{\mathcal{X}}_1^{*(m)}]^T + \boldsymbol{\mathcal{N}}^{(m)} \tag{3.54}$$

Let $\boldsymbol{\Lambda}_i^{(m)} = \boldsymbol{R}^{(m)T} \hat{\boldsymbol{\Lambda}}_i^{(m)} \boldsymbol{R}^{(m)}$ for $i = 1, 2$, then $\boldsymbol{\Lambda}_i^{(m)}$ is $M \times M$ diagonal matrix. To simplify the notation we will drop the superscript m and write (3.54) as

$$\boldsymbol{\mathcal{Y}}_1 = \boldsymbol{\Lambda}_{11} \boldsymbol{\mathcal{X}}_1 + \boldsymbol{\Lambda}_{12} \boldsymbol{\mathcal{X}}_2 + \boldsymbol{\mathcal{N}}_1 \tag{3.55}$$

$$\boldsymbol{\mathcal{Y}}_2 = \boldsymbol{\Lambda}_{21} \boldsymbol{\mathcal{X}}_1^* - \boldsymbol{\Lambda}_{22} \boldsymbol{\mathcal{X}}_2^* + \boldsymbol{\mathcal{N}}_2 \tag{3.56}$$

where $\boldsymbol{\Lambda}_1 = diag[\boldsymbol{\Lambda}_{11} \; \boldsymbol{\Lambda}_{22}]$ and $\boldsymbol{\Lambda}_2 = diag[\boldsymbol{\Lambda}_{12} \; \boldsymbol{\Lambda}_{21}]$. Combining (3.55) and (3.56) can be written as

$$\begin{aligned} \boldsymbol{\mathcal{Y}} &= \begin{bmatrix} \boldsymbol{\mathcal{Y}}_1 \\ \boldsymbol{\mathcal{Y}}_2^* \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21}^* & -\boldsymbol{\Lambda}_{22}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{X}}_1 \\ \boldsymbol{\mathcal{X}}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mathcal{N}}_1 \\ \boldsymbol{\mathcal{N}}_2^* \end{bmatrix} \\ &\triangleq \boldsymbol{\Lambda}\boldsymbol{\mathcal{X}}_{12} + \boldsymbol{\mathcal{N}}_{12} \end{aligned} \tag{3.57}$$

For Alamouti structure, $\boldsymbol{\Lambda}_{11} = \boldsymbol{\Lambda}_{22}$ and $\boldsymbol{\Lambda}_{12} = \boldsymbol{\Lambda}_{21}$. In order to make this assumption valid, we introduce reordering of the sub-carriers before mapping at the transmitter as $\boldsymbol{O}\acute{\boldsymbol{\mathcal{X}}}_1$ and $\boldsymbol{O}\acute{\boldsymbol{\mathcal{X}}}_2$, where $\boldsymbol{O} = [I_1, I_{M/2+1}, I_2, I_{M/2+2}, ..., I_{M/2}, I_M]$ and assume that the channel does not change over two consecutive sub-carriers. At the receiver side, the reordering is done after demapping by using a matrix $\boldsymbol{O}^T$. After MMSE equalization, we get

$$\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_1 \\ \check{\boldsymbol{\mathcal{X}}}_2 \end{bmatrix} = (\boldsymbol{\Lambda}^H\boldsymbol{\Lambda} + \frac{1}{SNR}\boldsymbol{I}_{2M})^{-1}\boldsymbol{\Lambda}^H\boldsymbol{\mathcal{Y}}_{12} \tag{3.58}$$

where SNR is the signal-to-noise ratio at the receiver. Since $\tilde{\boldsymbol{\Lambda}}\boldsymbol{\Lambda}^H$ has an Alamouti like structure, therefore,

$$\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_1 \\ \check{\boldsymbol{\mathcal{X}}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 \\ \boldsymbol{\Phi}_2^* & -\boldsymbol{\Phi}_1^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{Y}}_1 \\ \boldsymbol{\mathcal{Y}}_2^* \end{bmatrix} \tag{3.59}$$

where $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are diagonal matrices. Alternatively, (3.59) can be setup written as

$$\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_1 \\ \check{\boldsymbol{\mathcal{X}}}_2^* \end{bmatrix} = \begin{bmatrix} diag(\boldsymbol{\mathcal{Y}}_1) & diag(\boldsymbol{\mathcal{Y}}_2^*) \\ -diag(\boldsymbol{\mathcal{Y}}_2) & diag(\boldsymbol{\mathcal{Y}}_1^*) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Upsilon}_1 \\ \boldsymbol{\Upsilon}_2 \end{bmatrix}$$

where $\boldsymbol{\Upsilon}_1$ and $\boldsymbol{\Upsilon}_2$ are the vectors containing the diagonal elements of $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$. For a DFE, we have

$$\underbrace{\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_1 \\ \check{\boldsymbol{\mathcal{X}}}_2^* \end{bmatrix}}_{\check{\boldsymbol{\mathcal{X}}}_{12}} = \begin{bmatrix} diag(\boldsymbol{\mathcal{Y}}_1) & diag(\boldsymbol{\mathcal{Y}}_2^*) \\ -diag(\boldsymbol{\mathcal{Y}}_2) & diag(\boldsymbol{\mathcal{Y}}_1^*) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Upsilon}_1 \\ \boldsymbol{\Upsilon}_2 \end{bmatrix} + \begin{bmatrix} diag(\boldsymbol{\mathcal{D}}_1) & \mathbf{0} \\ \mathbf{0} & diag(\boldsymbol{\mathcal{D}}_2^*) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_1 \\ \boldsymbol{\Psi}_2 \end{bmatrix}$$

$$\triangleq \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{F}} + \boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{B}} \tag{3.60}$$

where $\boldsymbol{\mathcal{D}}_1$ and $\boldsymbol{\mathcal{D}}_2$ are $\boldsymbol{\mathcal{X}}_1$ and $\boldsymbol{\mathcal{X}}_2$, respectively, for the training mode or frequency-domain decisions on $\check{\boldsymbol{\mathcal{X}}}_1$ and $\check{\boldsymbol{\mathcal{X}}}_2$, respectively, for the decision-directed mode. The feedforward and feedback filter coefficients in the frequency-domain are $\boldsymbol{\mathcal{F}} = [\mathcal{F}(0)\ ...\ \mathcal{F}(M-1)]^T$ and $\boldsymbol{\mathcal{B}} = [\mathcal{B}(0)\ ...\ \mathcal{B}(M-1)]^T$ containing the elements $\{\boldsymbol{\Upsilon}_1, \boldsymbol{\Upsilon}_2\}$ and $\{\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2\}$, respectively. Moreover, $\boldsymbol{\mathcal{Z}}$ is an $M \times M$ Alamouti-like matrix containing the received symbols and $\boldsymbol{\mathcal{D}}$ is a diagonal matrix containing decisions. However, these coefficients will be computed adaptively; hence, an exact solution is not required. At the k^{th} instant, the output of the equalizer is given as

$$\check{\boldsymbol{\mathcal{X}}}_{12,k} = \boldsymbol{\mathcal{Z}}_k\boldsymbol{\mathcal{F}}_{k-1} + \boldsymbol{\mathcal{D}}_k\boldsymbol{\mathcal{B}}_{k-1} \tag{3.61}$$

The CRLS AFD-DFE recursion is given as in (3.42) with $\boldsymbol{G} =$

$[\mathbf{0}_{1\times(3M-2)} \quad \mathbf{1}_{1\times M}]$ and error vector as

$$\boldsymbol{\mathcal{E}}_k = \begin{bmatrix} \boldsymbol{D}_k - \check{\boldsymbol{\mathcal{X}}}_{12,k} \\ \boldsymbol{D}_k - \check{\boldsymbol{\mathcal{X}}}_{12,k} \end{bmatrix} \tag{3.62}$$

where $\boldsymbol{D}_k$ denotes the decisions at the k^{th} instant, i.e., $\boldsymbol{D}_k = \begin{bmatrix} \boldsymbol{D}_{1,k} \\ \boldsymbol{D}^*_{2,k} \end{bmatrix}$ and $\boldsymbol{\mathcal{A}}_k$ and $\boldsymbol{\mathcal{P}}_k$ as in (3.37), and (3.41), respectively.

<u>**Reduced-Complexity CRLS AFD-DFE:**</u> It might seem that (3.42) requires matrix inversion. However, due to the special structure of SFBC, no inversion is required resulting in significant complexity reductions as shown below.

The matrix $\boldsymbol{\mathcal{P}}_{k+1}$ has a diagonal structure, i.e., $\boldsymbol{\mathcal{P}}_{\boldsymbol{k+1}} = diag([\boldsymbol{P}^1_{k+1} \quad \boldsymbol{P}^2_{k+1}])$, where $\boldsymbol{P}^1_{k+1}$ and $\boldsymbol{P}^2_{k+1}$ are diagonal as well and $\boldsymbol{P}^1_{k+1}$ is given by.

$$\boldsymbol{P}^1_{k+1} = \lambda^{-1}[\boldsymbol{P}^1_k - \lambda^{-1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}^H_{k+1}(\boldsymbol{I}_M + \lambda^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}^H_{k+1})^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}^1_k] \tag{3.63}$$

Now, simplifying the term $(\lambda^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}^H_{k+1})$, we get

$$\begin{aligned} \lambda^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}^H_{k+1} &= \lambda^{-1}\boldsymbol{P}^1_k\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{\mathcal{Z}}^H_{k+1} \\ &= \lambda^{-1}\boldsymbol{P}^1_k \begin{bmatrix} diag(\boldsymbol{\mathcal{Y}}_{1,k+1}) & diag(\boldsymbol{\mathcal{Y}}^*_{2,k+1}) \\ -diag(\boldsymbol{\mathcal{Y}}_{2,k+1}) & diag(\boldsymbol{\mathcal{Y}}^*_{1,k+1}) \end{bmatrix} \\ &\quad \times \begin{bmatrix} diag(\boldsymbol{\mathcal{Y}}^*_{1,k+1}) & -diag(\boldsymbol{\mathcal{Y}}^*_{2,k+1}) \\ diag(\boldsymbol{\mathcal{Y}}_{2,k+1}) & diag(\boldsymbol{\mathcal{Y}}_{1,k+1}) \end{bmatrix} \\ &= \lambda^{-1}\boldsymbol{P}^1_k diag\left[diag(|\boldsymbol{\mathcal{Y}}_{1,k+1}|^2) + diag(|\boldsymbol{\mathcal{Y}}_{2,k+1}|^2)\right] \end{aligned} \tag{3.64}$$

Now $\boldsymbol{\mathcal{Z}}_{k+1}^{H}(\boldsymbol{I_M} + \lambda^{-1}\boldsymbol{\mathcal{Z}}_{k+1}\boldsymbol{P}_k^1\boldsymbol{\mathcal{Z}}_{k+1}^{H})^{-1}\boldsymbol{\mathcal{Z}}_{k+1} = diag([\boldsymbol{\varnothing} \quad \boldsymbol{\varnothing}]) \triangleq \boldsymbol{\psi}_{k+1}^1$, where $\boldsymbol{\varnothing}$ is diagonal given as $\boldsymbol{\varnothing} = (diag(|\boldsymbol{\mathcal{Y}_{1,k+1}}|^2) + diag(|\boldsymbol{\mathcal{Y}_{2,k+1}}|^2))^{-1} + \lambda^{-1}\boldsymbol{P}_k^1]^{-1}$. It follows that $\boldsymbol{P}_{k+1}^1$ will have the form

$$\boldsymbol{P}_{k+1}^1 = \lambda^{-1}[\boldsymbol{P}_k^1 - \lambda^{-1}\boldsymbol{P}_k^1\boldsymbol{\psi}_{k+1}^1\boldsymbol{P}_k^1] \tag{3.65}$$

Using the same approach as in the SISO case, $\boldsymbol{P}_{k+1}^2$ can be expressed as follows

$$\boldsymbol{P}_{k+1}^2 = \lambda^{-1}[\boldsymbol{P}_k^2 - \lambda^{-1}\boldsymbol{P}_k^2\boldsymbol{\psi}_{k+1}^2\boldsymbol{P}_k^2] \tag{3.66}$$

where $\boldsymbol{\psi}_{k+1}^2 = (|\boldsymbol{\mathcal{D}}_{k+1}|^{-2} + \lambda^{-1}\boldsymbol{P}_k^2)^{-1}$. Finally, the RLS AFD-DFE recursion has the form

$$\boldsymbol{\mathcal{W}}_{k+1} = \boldsymbol{\mathcal{W}}_k + diag([\boldsymbol{P}_{k+1}^1 \quad \boldsymbol{P}_{k+1}^2])(\boldsymbol{\mathcal{A}}_{k+1}^H\boldsymbol{\mathcal{E}}_{k+1} - \alpha_{k+1}\boldsymbol{G}^T) \tag{3.67}$$

where $\boldsymbol{P}_{k+1}^1$ and $\boldsymbol{P}_{k+1}^2$ are defined by (3.65) and (3.66), respectively.

3.5 Carrier Frequency Offset (CFO) in SFBC SC-FDMA

We proceed with embedded SFBC as in C-SFBC, the Alamouti structure is destroyed due to ICI. For CFO, the channel matrices, Λ_{ij}, in (3.57) lost their diagonal structures. We can approximate these matrices as a banded (tridiagonal) structure. Assuming the feedforward taps matrices have similar structure as channel matrices, the equalized signal can be written as

$$\begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_1 \\ \check{\boldsymbol{\mathcal{X}}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 \\ \boldsymbol{\Phi}_2^* & -\boldsymbol{\Phi}_1^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{Y}}_1 \\ \boldsymbol{\mathcal{Y}}_2^* \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Theta}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Theta}_2^* \end{bmatrix} \begin{bmatrix} \boldsymbol{D}_1 \\ \boldsymbol{D}_2 \end{bmatrix} \tag{3.68}$$

where $\boldsymbol{\Phi}_i$ is tri-diagonal matrix and $\boldsymbol{\Theta}_i$ is diagonal matrix. $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are $\boldsymbol{\mathcal{X}}_1$ and $\boldsymbol{\mathcal{X}}_2$, respectively, for the training mode or frequency-domain decisions on $\check{\boldsymbol{\mathcal{X}}}_1$ and $\check{\boldsymbol{\mathcal{X}}}_2$, respectively, for the decision-directed mode. However, the exact knowledge of the equalizer taps matrices is not need for adaptive solution. Now denoting $\boldsymbol{\mathcal{U}}_i = [\boldsymbol{\mathcal{Y}}(i-2)\ \boldsymbol{\mathcal{Y}}(i)\ \boldsymbol{\mathcal{Y}}(i+2)]$ for $i = 2, ...M-3$, $\boldsymbol{\mathcal{U}}_i = [\boldsymbol{\mathcal{Y}}(i)\ \boldsymbol{\mathcal{Y}}(i+2)]$ for $i = 0, 1$ and $\boldsymbol{\mathcal{U}}_i = [\boldsymbol{\mathcal{Y}}(i-2)\ \boldsymbol{\mathcal{Y}}(i)]$ for $i = M-1, M-2$, we can write (3.68) as

$$\begin{aligned} \check{\boldsymbol{\mathcal{X}}} &= \begin{bmatrix} \check{\boldsymbol{\mathcal{X}}}_1 \\ \check{\boldsymbol{\mathcal{X}}}_2^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z}_0 & \boldsymbol{Z}_1^* \\ -\boldsymbol{Z}_1 & \boldsymbol{Z}_0^* \end{bmatrix} \begin{bmatrix} \boldsymbol{\Upsilon}_1 \\ \boldsymbol{\Upsilon}_2 \end{bmatrix} + \begin{bmatrix} diag(\boldsymbol{D}_1) & \mathbf{0} \\ \mathbf{0} & diag(\boldsymbol{D}_2^*) \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_1 \\ \boldsymbol{\Psi}_2 \end{bmatrix} \\ &\triangleq \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{F}} + \boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{B}} \end{aligned} \tag{3.69}$$

where $\boldsymbol{Z}_j = diag[\boldsymbol{\mathcal{U}}_j\, \boldsymbol{\mathcal{U}}_{j+2} \ldots \boldsymbol{\mathcal{U}}_{j+M-2}]$ for $j = 0, 1$. $\boldsymbol{\Upsilon}_1$ and $\boldsymbol{\Upsilon}_2$ ($\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$) are the vectors containing the diagonal elements of $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ ($\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$). Moreover, the feedforward and feedback filter coefficients in the frequency-domain are $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{B}}$ containing the elements $\{\boldsymbol{\Upsilon}_1, \boldsymbol{\Upsilon}_2\}$ and $\{\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2\}$, respectively. Defining $\boldsymbol{\mathcal{A}}_k$, $\boldsymbol{\mathcal{E}}_k$ and $\boldsymbol{\mathcal{P}}_k$ as in (3.37), (3.38) and (3.41), respectively, the RLS update is given as in (3.43) with $\boldsymbol{G} = [\mathbf{0}_{1\times(3M-2)}\quad \mathbf{1}_{1\times M}]$.

Reduced-Complexity 3-tap CRLS AFD-DFE: Now exploiting the special structure of SFBC matrix, it can be seen that there is no matrix inversion involved altogether and, hence, complexity reduction is significantly reduced. Starting with $k = 0$ and using $\boldsymbol{P}_0^1 = \epsilon^{-1}\boldsymbol{I}_{3M-4}$, $\boldsymbol{P}_1^1$ is given by

$$\begin{aligned}
\boldsymbol{P}_1^1 &= \lambda^{-1}[\boldsymbol{P}_0^1 - \lambda^{-1}\boldsymbol{P}_0^1\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_M + \lambda^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{P}_0^1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{P}_0^1] \\
&= \lambda^{-1}[\epsilon^{-1}\boldsymbol{I}_{3M-4} - \lambda^{-1}\epsilon^{-1}\boldsymbol{I}_{3M-4}\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_M + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1\epsilon^{-1}\boldsymbol{I}_{3M-4}]
\end{aligned} \tag{3.70}$$

Now

$$\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H = \begin{bmatrix} \boldsymbol{Z}_0\boldsymbol{Z}_{0,1}^H + \boldsymbol{Z}_{1,1}^*\boldsymbol{Z}_{1,1}^T & -\boldsymbol{Z}_{0,1}\boldsymbol{Z}_{1,1}^H + \boldsymbol{Z}_{1,1}^*\boldsymbol{Z}_{0,1}^T \\ -\boldsymbol{Z}_{1,1}\boldsymbol{Z}_{0,1}^H + \boldsymbol{Z}_{0,1}^*\boldsymbol{Z}_{1,1}^T & -\boldsymbol{Z}_{1,1}\boldsymbol{Z}_{1,1}^H + \boldsymbol{Z}_{0,1}^*\boldsymbol{Z}_{0,1}^T \end{bmatrix} \tag{3.71}$$

It can easily be seen that $\boldsymbol{Z}_0\boldsymbol{Z}_0^H + \boldsymbol{Z}_1^*\boldsymbol{Z}_1^T = diag[|\boldsymbol{\mathcal{U}}_{0,1}|^2 + |\boldsymbol{\mathcal{U}}_{1,1}|^2 \; |\boldsymbol{\mathcal{U}}_{2,1}|^2 + |\boldsymbol{\mathcal{U}}_{3,1}|^2 ... \; |\boldsymbol{\mathcal{U}}_{M-2,1}|^2 + |\boldsymbol{\mathcal{U}}_{M-1,1}|^2]$ is a diagonal matrix and likewise other entries in (3.71). Therefore, $\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H$ is a $M \times M$ matrix containing 4 $\frac{M}{2} \times \frac{M}{2}$ diagonal matrix. This structure allow us to easily find the inverse in (3.70). Now $\boldsymbol{\mathcal{Z}}_1^H(\boldsymbol{I}_M + \lambda^{-1}\epsilon^{-1}\boldsymbol{\mathcal{Z}}_1\boldsymbol{\mathcal{Z}}_1^H)^{-1}\boldsymbol{\mathcal{Z}}_1 = \left[\begin{smallmatrix} \boldsymbol{\varnothing}_0 & \boldsymbol{\varnothing}_1 \\ \boldsymbol{\varnothing}_2 & \boldsymbol{\varnothing}_3 \end{smallmatrix}\right] = \boldsymbol{\varnothing}$ and

$$\boldsymbol{\varnothing}_i = \begin{bmatrix} \boldsymbol{\varnothing}_{i,0} & & & \\ & \boldsymbol{\varnothing}_{i,1} & & \\ & & \ddots & \\ & & & \boldsymbol{\varnothing}_{i,\frac{M}{2}} \end{bmatrix} \tag{3.72}$$

and the entries $\boldsymbol{\phi}_{i,j}, j = 1, ..., \frac{M}{2} - 1$ are 3×3 matrices and $\boldsymbol{\phi}_i, i = 0, \frac{M}{2}$ are 2×2 matrices. Now $\boldsymbol{P}_1^1$ has a similar structure as $\boldsymbol{\phi}$ $\boldsymbol{P}_1^1 = \left[\begin{smallmatrix} \boldsymbol{P}_{1,0}^1 & \boldsymbol{P}_{1,1}^1 \\ \boldsymbol{P}_{1,2}^1 & \boldsymbol{P}_{1,3}^1 \end{smallmatrix}\right]$. Proceeding for $k = 1$, we have

$$\boldsymbol{P}_2^1 \;=\; \lambda^{-1}[\boldsymbol{P}_1^1 - \lambda^{-1}\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H(\boldsymbol{I}_M + \lambda^{-1}\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H)^{-1}\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1] \tag{3.73}$$

Let $\boldsymbol{\mathcal{Z}}_2\boldsymbol{P}_1^1\boldsymbol{\mathcal{Z}}_2^H = \left[\begin{smallmatrix} \boldsymbol{\varphi}_0 & \boldsymbol{\varphi}_1 \\ \boldsymbol{\varphi}_2 & \boldsymbol{\varphi}_3 \end{smallmatrix}\right]$ where $\boldsymbol{\varphi}_0 = (\boldsymbol{Z}_0\boldsymbol{P}_{1,0}^1 + \boldsymbol{Z}_1^*\boldsymbol{P}_{1,2}^1)\boldsymbol{Z}_0^H + (\boldsymbol{Z}_0\boldsymbol{P}_{1,1}^1 + \boldsymbol{Z}_1^*\boldsymbol{P}_{1,3}^1)\boldsymbol{Z}_1^T$, which is a diagonal matrix and similarly for other entries. Therefore, inverse in (3.73) can be found using (2.84), i.e., matrix inversion is just scalar inversions. For $k > 1$, $\boldsymbol{P}_k^1$ has similar structure and for $\boldsymbol{P}_k^2$ same reasons are valid as for the 1-tap AFD-DFE SFBC case.

3.6 Performance and Complexity Analysis

In this section, the Minimum Mean Square Error (MMSE), transient, steady-state, tracking and computational complexity analyses are carried out. In the derivations, we assume that the data sequences (both transmitted data and detected data) are independent and identically distributed (i.i.d) with zero mean, and independent of the noise. The optimal MMSE equalizer weights $\boldsymbol{\mathcal{W}}^o$ for i^{th} frequency bin are given as [23] [1], for LE

$$\mathcal{F}^o(i) = \frac{\sigma_x^2\Lambda(i)^*}{\sigma_x^2|\Lambda(i)|^2 + \frac{N}{M}\sigma_{\mathcal{N}}^2}$$

and for DFE

$$\mathcal{F}^o(i) = \frac{\sigma_x^2 \Lambda(i)^*}{\sigma_x^2 \sum_{j=0}^{M-1} |\Lambda(j)|^2 + \frac{N}{M}\sigma_{\mathcal{N}}^2},\ \mathcal{B}^o(i) = -\mathcal{F}^o(i)\Lambda(i) + \frac{1}{M}\sum_{j=0}^{M-1} \mathcal{F}^o(j)\Lambda(j))$$

Using these weights, the corresponding i^{th} frequency bin MMSE for the LE ($J_L(i)$) and the DFE ($J_D(i)$) are derived, respectively, and expressed as

$$J_L(i) = \frac{N\sigma_{\mathcal{N}}^2 \sigma_x^2 (M|\Lambda(i)|^2 \sigma_x^2 + N\sigma_{\mathcal{N}}^2)}{(\sigma_x^2 |\Lambda(i)|^2 M + N\sigma_{\mathcal{N}}^2)^2} \tag{3.74}$$

and

$$J_D(i) = \frac{N\sigma_{\mathcal{N}}^2 \sigma_x^2 (M|\Lambda(i)|^2 \sigma_x^2 + N\sigma_{\mathcal{N}}^2)}{(\sigma_x^2 \sum_{j=0}^{M-1} |\Lambda(j)|^2 + N\sigma_{\mathcal{N}}^2)^2} \tag{3.75}$$

3.6.1 Transient Analysis

We start by deriving the transient behavior of the RLS AFD-DFE in a stationary environment, assuming that the forgetting factor λ is unity. The MSE of the RLS AFD-DFE at instant $k+1$ is given by

$$J_{k+1}^r(i) = E\left|\xi_{k+1}(i)\right|^2 \tag{3.76}$$

The a priori estimation output error $\xi_{k+1}(i)$ can be expressed as

$$\xi_{k+1}(i) = \xi_{k+1}^o(i) + \boldsymbol{a}_{k+1}(i)\underbrace{[\boldsymbol{\mathcal{W}}^o(i) - \boldsymbol{\mathcal{W}}_k(i)]}_{\widetilde{\boldsymbol{\mathcal{W}}}_k(i)} \tag{3.77}$$

where $X_k(i) = \boldsymbol{a}_k(i)\boldsymbol{\mathcal{W}}^o(i) + \xi_k^o(i)$, $\boldsymbol{a}_k(i) = [\mathcal{Y}_k(i) \;\; D_k(i)]$ and $\boldsymbol{\mathcal{W}}_k(i) = [\mathcal{F}_k(i) \;\; \mathcal{B}_k(i)]^T$. Substituting (3.77) into (3.76) and then expending terms, we get

$$\begin{aligned} J_{k+1}^r(i) &= \underbrace{E\left|\xi_{k+1}^o(i)\right|^2}_{J_D(i)} + E\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)\boldsymbol{a}_{k+1}^H(i) \\ &\quad + E\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\xi_{k+1}^{o*}(i) + E\xi_{k+1}^o(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)\boldsymbol{a}_{k+1}^H(i) \qquad (3.78) \end{aligned}$$

We can express the second expectation in (3.78) as

$$\begin{aligned} E\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)\boldsymbol{a}_{k+1}^H(i) &= E\; Tr\{\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)\boldsymbol{a}_{k+1}^H(i)\} \\ &= E\; Tr\{\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)\boldsymbol{a}_{k+1}^H(i)\boldsymbol{a}_{k+1}(i)\} \\ &= Tr\; E\{\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)\boldsymbol{a}_{k+1}^H(i)\boldsymbol{a}_{k+1}(i)\} \end{aligned} \qquad (3.79)$$

where $Tr\{.\}$ is the trace of a matrix. Using the assumption that the product $\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)$ varies at a slower rate than the product $\boldsymbol{a}_{k+1}^H(i)\boldsymbol{a}_{k+1}(i)$, we can write

$$E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)\boldsymbol{a}_{k+1}^H(i)\boldsymbol{a}_{k+1}(i) \approx Tr\{E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)E\boldsymbol{a}_{k+1}^H(i)\boldsymbol{a}_{k+1}(i)\} \qquad (3.80)$$

where $E\boldsymbol{a}_k^H(i)\boldsymbol{a}_k(i) = \boldsymbol{R}_{\mathcal{A}}(i)$ is the input correlation matrix having diagonal structure as can be seen from (2.24), i.e., $\boldsymbol{R}_{\mathcal{A}}(i) = diag([|\Lambda(i)|^2\sigma_x^2 + \frac{N}{M}\sigma_{\mathcal{N}}^2 \quad \sigma_x^2])$. Now, to find $E\widetilde{\boldsymbol{\mathcal{W}}}_k(i)\widetilde{\boldsymbol{\mathcal{W}}}_k^H(i)$, we proceed by writing the normal equations

$$\boldsymbol{\mathcal{W}}_k(i) = \Theta_k^{-1}(i)\boldsymbol{R}_k(i) \tag{3.81}$$

where $\boldsymbol{R}_k(i) = \sum_{j=0}^{k} \boldsymbol{a}_j^H(i)X_j(i)$. Using the values of $\Theta_k(i)$, $\boldsymbol{R}_k(i)$ and $X_k(i)$ in (3.81) and ignoring the initial conditions, $\widetilde{\boldsymbol{\mathcal{W}}}_k(i)$ is given as

$$\widetilde{\boldsymbol{\mathcal{W}}}_k(i) = -\Theta_k^{-1}(i)\sum_{j=0}^{k+1}\boldsymbol{a}_j^H(i)\xi_j^o(i) + \Theta_k^{-1}(i)\alpha_{k+1}[0 \quad 1]^T \tag{3.82}$$

Now

$$|\widetilde{\boldsymbol{\mathcal{W}}}_k(i)| \leq |\underbrace{-\Theta_k^{-1}(i)\sum_{j=0}^{k+1}\boldsymbol{a}_j^H(i)\xi_j^o(i)}_{\widetilde{\boldsymbol{\mathcal{W}}}'_k(i)}| \tag{3.83}$$

Therefore, the weight-error correlation matrix of $\widetilde{\boldsymbol{\mathcal{W}}}'_k(i)$ is given as

$$E\widetilde{\boldsymbol{\mathcal{W}}}'_k(i)\widetilde{\boldsymbol{\mathcal{W}}}'^H_k(i) = E\Theta_k^{-1}(i)\sum_{j=0}^{k+1}\boldsymbol{a}_j^H(i)\boldsymbol{a}_j(i)\Theta_k^{-1}(i)\xi_j^o(i)\xi_j^{o*}(i) \tag{3.84}$$

Using the assumption that $\boldsymbol{a}_k(i)$, and therefore, $\Theta_k^{-1}(i)$, is independent of the noise $\xi_k^o(i)$, (3.84) can be expressed as a product of two expectations as follows

$$\begin{aligned} E\widetilde{\boldsymbol{\mathcal{W}}}'_k(i)\widetilde{\boldsymbol{\mathcal{W}}}'^H_k(i) &= E\Theta_k^{-1}(i)\sum_{j=0}^{k+1}\boldsymbol{a}_j^H(i)\boldsymbol{a}_j(i)\Theta_k^{-1}(i)\ E\xi_j^o(i)\xi_j^{o*}(i) \\ &= J_D(i)\ E\Theta_k^{-1}(i) \end{aligned} \tag{3.85}$$

Assuming ergodicity, we may express the ensemble-average correlation matrix of the input of the AFD-DFE as $\boldsymbol{R}_{\mathcal{A}}(i) = \frac{1}{k}\Theta_k(i)$. Hence, the weight-error correlation matrix reduces to $E\widetilde{\boldsymbol{\mathcal{W}}}'_k(i)\widetilde{\boldsymbol{\mathcal{W}}}'^H_k(i) = \frac{1}{k}J_D(i)\boldsymbol{R}^{-1}_{\mathcal{A}}(i)$. Therefore,

$$E\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}'_k(i)\widetilde{\boldsymbol{\mathcal{W}}}'^H_k(i)\boldsymbol{a}^H_{k+1}(i) \approx \frac{1}{k}J_D(i)Tr\{\boldsymbol{R}_{\mathcal{A}}(i)\boldsymbol{R}^{-1}_{\mathcal{A}}(i)\} = \frac{2}{k}J_D(i) \tag{3.86}$$

The third and fourth expectations in (3.78) are zero because $\widetilde{\boldsymbol{\mathcal{W}}}_k(i)$ depends on past values of $\boldsymbol{a}_{k+1}(i)$ and $\xi^o_{k+1}(i)$. Also $\boldsymbol{a}_{k+1}(i)$ and $\xi^o_{k+1}(i)$ are statistically independent and $\xi^o_{k+1}(i)$ has zero mean. Therefore,

$$J'^r_{k+1}(i) = J_D(i)\left[1 + \frac{2}{k}\right] \geq J^r_{k+1}(i) \tag{3.87}$$

To compare the CRLS AFD-DFE and LMS AFD-DFE, the MSE of LMS AFE-DFE can be shown to be [29]

$$J'^l_{k+1}(i) = J_D(i)\left[1 + \mu\sum_{j=0}^{1}\frac{r_j}{2-\mu r_j}\right] + \sum_{j=0}^{1} r_j\left(|\nu_j(0)|^2 - \frac{\mu J_D(i)}{2-\mu r_j}\right)(1-\mu r_j)^{2k} \tag{3.88}$$

where r_j is the j^{th} eigenvalue of the correlation matrix $\boldsymbol{R}_{\mathcal{A}}$ and $\boldsymbol{\nu}(k+1) = [v_0(k+1), v_1(k+1)]^T = (\boldsymbol{I}_2 - \mu\boldsymbol{R}(i)_{\mathcal{A}})\boldsymbol{\nu}(k) - \mu\boldsymbol{\mathcal{A}}_{k+1}(i)^H\boldsymbol{\mathcal{E}}_{k+1}(i)$. The evolution of $J'^l_{k+1}(i)$ with step size μ is governed by the exponential quantity $(1-\mu r_j)^{2k}$. This clearly shows that the CRLS AFD-DFE converges faster than the LMS AFD-DFE. The simulation results support this claim.

3.6.2 Steady-State Analysis

To begin with, the update recursion (3.42) for i^{th} frequency bin can be written as

$$\boldsymbol{\mathcal{W}}_{k+1}(i) = \boldsymbol{\mathcal{W}}_{k}(i) + \boldsymbol{\mathcal{P}}_{k+1}(i)(\boldsymbol{a}_{k+1}^{H}(i)\xi_{k+1}(i) - \alpha_{k+1}[01]^{T}) \tag{3.89}$$

In terms of the weight-error vector $\widetilde{\boldsymbol{\mathcal{W}}}'_{k}(i)$, we can write

$$\widetilde{\boldsymbol{\mathcal{W}}}'_{k+1}(i) = \widetilde{\boldsymbol{\mathcal{W}}}'_{k}(i) - \boldsymbol{\mathcal{P}}_{k+1}(i)\boldsymbol{a}_{k+1}^{H}(i)\xi_{k+1}(i) \tag{3.90}$$

Multiplying (3.90) by $\boldsymbol{a}_{k+1}(i)$ from the left, we may write it in terms of a priori estimation error $\xi_{k+1}^{a}(i)$ and a posteriori estimation error $\xi_{k+1}^{p}(i)$ as follows

$$\underbrace{\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}'_{k+1}(i)}_{\xi_{k+1}^{p}(i)} = \underbrace{\boldsymbol{a}_{k+1}(i)\widetilde{\boldsymbol{\mathcal{W}}}'_{k}(i)}_{\xi_{k+1}^{a}(i)} - \underbrace{\boldsymbol{a}_{k+1}(i)\boldsymbol{\mathcal{P}}_{k+1}(i)\boldsymbol{a}_{k+1}^{H}(i)}_{\|\boldsymbol{a}_{k+1}(i)\|_{\mathcal{P}}}\xi_{k+1}(i) \tag{3.91}$$

where $\| \, . \, \|_{\mathcal{P}}$ stands for the squared-weighted Euclidean norm of a vector. Combining (3.90) and (3.91) to eliminate ξ_{k+1}, we get

$$\begin{aligned} \widetilde{\boldsymbol{\mathcal{W}}}'_{k+1}(i) + \boldsymbol{\mathcal{P}}_{k+1}(i)\boldsymbol{a}_{k+1}^{H}(i)\underbrace{(\| \, \boldsymbol{a}_{k+1}(i) \, \|_{\mathcal{P}})^{\dagger}}_{\bar{a}_{k+1}(i)}\xi_{k+1}^{a}(i) \quad &= \quad \widetilde{\boldsymbol{\mathcal{W}}}'_{k}(i) + \boldsymbol{\mathcal{P}}_{k+1}(i)\boldsymbol{a}_{k+1}^{H}(i) \\ &\quad \times(\| \, \boldsymbol{a}_{k+1}(i) \, \|_{\mathcal{P}})^{\dagger}\xi_{k+1}^{p}(i) \end{aligned} \tag{3.92}$$

where $(.)^{\dagger}$ represents the pseudo-inverse. Now, equating the energies (squared Euclidean norms) of both sides of (3.92) with $[\boldsymbol{\mathcal{P}}_{k+1}(i)]^{-1}$ as a weighting matrix, the energy conservation relation becomes

$$\parallel \widetilde{\boldsymbol{\mathcal{W}}}'_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} + \bar{a}_{k+1}(i)|\xi^a_{k+1}(i)|^2 = \parallel \widetilde{\boldsymbol{\mathcal{W}}}'_{k}(i) \parallel^2_{\mathcal{P}^{-1}} + \bar{a}_{k+1}(i)|\xi^p_{k+1}(i)|^2 \qquad (3.93)$$

Taking the expectation of (3.93) and using the steady-state approximations, $E\boldsymbol{\mathcal{P}}_{k+1}(i) \approx (1-\lambda)\boldsymbol{\mathcal{R}}^{-1}_{\mathcal{A}} = \mathcal{P}$, $E\widetilde{\boldsymbol{\mathcal{W}}}'_{k+1}(i) = E\widetilde{\boldsymbol{\mathcal{W}}}'_{k}(i)$ and $E \parallel \widetilde{\boldsymbol{\mathcal{W}}}'_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} = E \parallel \widetilde{\boldsymbol{\mathcal{W}}}'_{k}(i) \parallel^2_{\mathcal{P}^{-1}}$, we arrive at

$$E\bar{a}_{k+1}(i)|\xi^a_{k+1}(i)|^2 = E\bar{a}_{k+1}(i)|\xi^p_{k+1}(i)|^2, \;\; k \to \infty \qquad (3.94)$$

Substituting $\xi^p_{k+1}(i)$ from (3.91) into (3.94), we get

$$E\bar{a}_{k+1}(i)|\xi^a_{k+1}(i)|^2 = E\bar{a}_{k+1}(i)|\xi^a_{k+1}(i) - \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} \xi_{k+1}(i)|^2, \;\; k \to \infty \qquad (3.95)$$

which upon expansion and simplification reduces to

$$E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} |\xi_{k+1}(i)|^2 = 2Re(E\xi^{a*}_{k+1}(i)\xi_{k+1}(i)), \;\; k \to \infty \qquad (3.96)$$

As $\xi_{k+1}(i) = \xi^o_{k+1}(i) + \xi^a_{k+1}(i)$, (3.96) becomes

$$J_D(i)E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} + E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} |\xi^a_{k+1}(i)|^2 = 2\underbrace{E|\xi^a_{k+1}(i)|^2}_{J'_{exss}(i)}, \;\; k \to \infty \qquad (3.97)$$

where $J'_{exss}(i)$ is the Excess Mean Square Error (EMSE). Assume that at steady-state, $\parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}}$ is independent of $\xi^a_{k+1}(i)$. This condition allows us to separate the expectation $E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} |\xi^a_{k+1}(i)|^2$ into a product of two expectations as follows

$$E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} |\xi^a_{k+1}(i)|^2 = E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} E|\xi^a_{k+1}(i)|^2 \tag{3.98}$$

If we replace $\boldsymbol{\mathcal{P}}_{k+1}(i)$ by its assumed mean value, we obtain the approximation

$$E \parallel \boldsymbol{a}_{k+1}(i) \parallel_{\mathcal{P}} \approx Tr\{\boldsymbol{\mathcal{R}}\mathcal{P}\} = 2(1-\lambda) \tag{3.99}$$

Substituting into (3.97), we get

$$J'_{exss}(i) = \frac{J_D(i)(1-\lambda)}{\lambda} \tag{3.100}$$

Therefore, the MSE at the steady-state $J_{ss}(i)$ is given as

$$J'_{ss}(i) = J_D(i) + J'_{exss}(i) = \frac{J_D(i)}{\lambda} \tag{3.101}$$

3.6.3 Tracking Analysis

For time-varying channels, we will adopt a first-order random walk model for the variation in the tap weight vector $\boldsymbol{\mathcal{W}}^o_k$. The model assumes that $\boldsymbol{\mathcal{W}}^o_k$ undergoes random variations of the form

$$\boldsymbol{\mathcal{W}}^o_{k+1}(i) = \boldsymbol{\mathcal{W}}^o_k(i) + \boldsymbol{q}_{k+1}(i) \tag{3.102}$$

where $\boldsymbol{q}_k(i) = [\boldsymbol{f}_L(i)\boldsymbol{q}_k^F(i), \boldsymbol{f}_L(i)\boldsymbol{q}_k^B(i)]^T$ and $\boldsymbol{f}_L(i)$ is the i^{th} row of partial $(M \times L)$ DFT matrix. $\boldsymbol{q}_k^F(i)$ and $\boldsymbol{q}_k^B(i)$ are the time-domain random column vectors of length L with zero mean and correlation matrix $\boldsymbol{Q}_t = (1 - \mathcal{R}^2(1))\boldsymbol{I}_L$ [5]. Assuming $\boldsymbol{q}_k^F(i)$ and $\boldsymbol{q}_k^B(i)$ are independent and note that $\boldsymbol{f}_L(i)\boldsymbol{f}_L(i)^H = L/M$, the covariance matrix of $\boldsymbol{q}_k(i), i = 1, ..., M$, is $\boldsymbol{Q} = \frac{L}{M}(1 - \mathcal{R}^2(1))\boldsymbol{I}_{2L}$.

Now, defining $\boldsymbol{\mathcal{W}}_k^o(i) - \boldsymbol{\mathcal{W}}'_k(i) = \widetilde{\boldsymbol{\mathcal{W}}}'_k(i)$ and writing the energy-conservation relation leads to

$$\begin{aligned} \| \boldsymbol{\mathcal{W}}_{k+1}^o(i) - \boldsymbol{\mathcal{W}}'_{k+1}(i) \|^2_{\mathcal{P}^{-1}} + \bar{a}_{k+1}(i)|\xi_{k+1}^a(i)|^2 &= \| \boldsymbol{\mathcal{W}}_{k+1}^o(i) - \boldsymbol{\mathcal{W}}'_k(i) \|^2_{\mathcal{P}^{-1}} \\ &\quad + \bar{a}_{k+1}(i)|\xi_{k+1}^p(i)|^2 \end{aligned} \tag{3.103}$$

where $\xi_{k+1}^p(i) = \boldsymbol{a}_{k+1}(i)[\boldsymbol{\mathcal{W}}_{k+1}^o(i) - \boldsymbol{\mathcal{W}}'_{k+1}(i)]$ and $\xi_{k+1}^a(i) = \boldsymbol{a}_{k+1}(i)[\boldsymbol{\mathcal{W}}_{k+1}^o(i) - \boldsymbol{\mathcal{W}}'_k(i)]$. Moreover, the random walk model (3.102) allows us to relate $E \| \boldsymbol{\mathcal{W}}_{k+1}^o(i) - \boldsymbol{\mathcal{W}}'_k(i) \|^2_{\mathcal{P}^{-1}}$ to $E \| \widetilde{\boldsymbol{\mathcal{W}}}'_k(i) \|^2_{\mathcal{P}^{-1}}$ as follows

$$\begin{aligned} E \| \boldsymbol{\mathcal{W}}_{k+1}^o(i) - \boldsymbol{\mathcal{W}}'_k(i) \|^2_{\mathcal{P}^{-1}} &= E \| \boldsymbol{\mathcal{W}}_k^o(i) + \boldsymbol{q}_{k+1}(i) - \boldsymbol{\mathcal{W}}'_k(i) \|^2_{\mathcal{P}^{-1}} \\ &= E \| \widetilde{\boldsymbol{\mathcal{W}}}'_k(i) + \boldsymbol{q}_{k+1}(i) \|^2_{\mathcal{P}^{-1}} \\ &= E \| \widetilde{\boldsymbol{\mathcal{W}}}'_k(i) \|^2_{\mathcal{P}^{-1}} + \| \boldsymbol{q}_{k+1}(i) \|^2_{\mathcal{P}^{-1}} \end{aligned} \tag{3.104}$$

where the last step follows from the fact that $\boldsymbol{\mathcal{W}}'_k(i)$ is independent of $\boldsymbol{q}_{k+1}(i)$ and uses steady-state assumption. Next, taking expectation of (3.103), we get

$$\begin{aligned} E \parallel \widetilde{\boldsymbol{\mathcal{W}}}'_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} + E\bar{a}_{k+1}(i)|\xi^a_{k+1}(i)|^2 &= E \parallel \widetilde{\boldsymbol{\mathcal{W}}}'_{k}(i) \parallel^2_{\mathcal{P}^{-1}} + E \parallel \boldsymbol{q}_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} \\ &\quad + E\bar{a}_{k+1}(i)|\xi^p_{k+1}(i)|^2 \end{aligned} \tag{3.105}$$

Moreover, $\boldsymbol{q}_{k+1}(i)$ is independent of $\boldsymbol{\mathcal{P}}_{k+1}(i)$, so that

$$E \parallel \boldsymbol{q}_{k+1}(i) \parallel^2_{\mathcal{P}^{-1}} = Tr\ E\{\boldsymbol{q}_{k+1}(i)^H \mathcal{P}^{-1} \boldsymbol{q}_{k+1}(i)\} = Tr\{\boldsymbol{Q}\mathcal{P}^{-1}\} = \frac{1}{(1-\lambda)} Tr\{\boldsymbol{Q}\boldsymbol{R}_{\mathcal{A}}\} \tag{3.106}$$

Solving (3.105) as done in steady-state analysis and using (3.106), it can be shown that in time-varying environment, the MSE of the RLS AFD-DFE is given as

$$\begin{aligned} J'_{ss}(i) &= J_D(i) + \frac{2(1-\lambda)J_{DFE}(i) + \frac{1}{(1-\lambda)} Tr\{\boldsymbol{Q}\boldsymbol{R}\}}{2 - 2(1-\lambda)} \\ &= \frac{J_D(i)}{\lambda} + \frac{Tr\{\boldsymbol{Q}\boldsymbol{R}\}}{2\lambda(1-\lambda)} \end{aligned} \tag{3.107}$$

By comparing the update equations for RLS and CRLS it can be seen that the computational complexity of CRLS AFD-DFE is same as RLS AFD-DFE.

3.7 Simulation Results

In this section, the theoretical findings are validated. Similar to an LTE system, the carrier frequency and bandwidth are set to 2 GHz and 5 MHz, respectively. Other simulation parameters used are $M = 64$ and $N = 1024$, therefore, the maximum number of users that the system can support is $K = 16$. The modulation scheme used is Quadrature Phase shift Keying (QPSK) and the channel is frequency selective with 12-paths and each path fades independently, according to the Rayleigh distribution.

Fig. 3.2 depicts the performance of the AFD-DFE algorithm. As can be seen from this figure, the best performance is obtained through the use of the AFD-DFE with constraint for case 2, while the worst performance is obtained by the AFD-DFE with constraint for case 1. The reason behind the good performance of case 2 is that the input correlation matrix for the feedforward filter is unaffected by the decision errors. This can be clearly seen from the figure that as AFD-DFE is switched to decision-directed mode, the MSE becomes worst in case 1. The computational complexity of the AFD-DFE (case 1) is slightly higher than that of the AFD-LE but when compared to the performance obtained through the use of the AFD-DFE (case 1), this additional complexity at the base station is well justified. Also, the figure shows that the performance of the RLS-based AFD-DFE is better as compared to that of the Constraint LMS (CLMS) based AFD-DFE in terms of convergence speed and MSE. Only 15 iterations are needed for the RLS to converge; hence, the resulting complexity and latency increase due to adaptation are not significant. This figure also depicts the theoretical curves and close agreement between the theoretical findings and simulations is observed. Note that the theoretical curve of AFD-DFE assumes perfect decisions. For the rest of the figures, we have used AFD-DFE for case 2. The reason of improved performance of the CRLS as compared to the RLS can be seen from Fig. 3.3.

Since in CRLS, b_0 is more close to zero unlike RLS which ensures that the present symbol is not being canceled out.

The BER performance of the AFD-DFE is shown in Fig. 3.4. It is clear that AFD-DFE with CRLS outperforms the RLS-based AFD-DFE, practical MMSE DFE and MMSE LE with known channel in terms of BER. Note here that there in no error propagation in the AFD-DFE unlike the practical MMSE DFE of [1] and [23]with known channel, which was due to poor estimation of the correlation between the transmitted data and the decisions. Moreover, this figure shows that the performance of the constraint-based RLS AFD-DFE is close to that of the ideal MMSE DFE [1] with known channel. For the rest of the figures we have used CRLS AFD-DFE.

In addition, the effect of the Doppler on the SC-FDMA system is reported in figs. 3.5 and 3.6. For this scenario, speed of the user v is varied from 3 km/h to 300 km/h. As can be observed from these two figures, our proposed RLS AFD-DFE is robust to the Doppler Effect. Fig. 3.6 depicts the theoretical and simulated MSE for different user's velocities.

Next, the effect of the CFO on the system performance is investigated, this is because if there is a slip in the frequency (transmitter and receiver not any more synchronized) then the performance degrades. Fig. 3.7 shows that the performance of 3-tap AFD-DFE is better than that of 1-tap AFD-DFE in large CFO, from which we can conclude that 3-tap AFD-DFE is somewhat robust to the ICI.

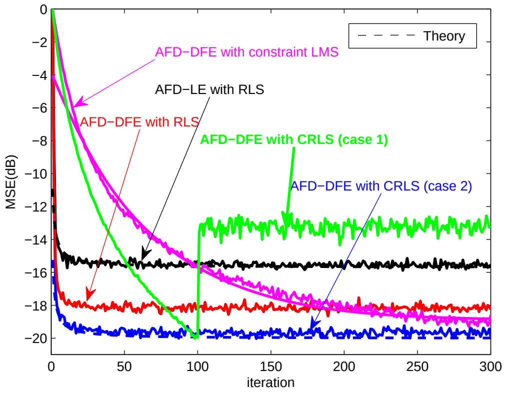


Figure 3.2: Learning curves of AFD-LE and AFD-DFE in SCFDMA system with user velocity $v = 3$ Km/h and SNR of 20 dB.

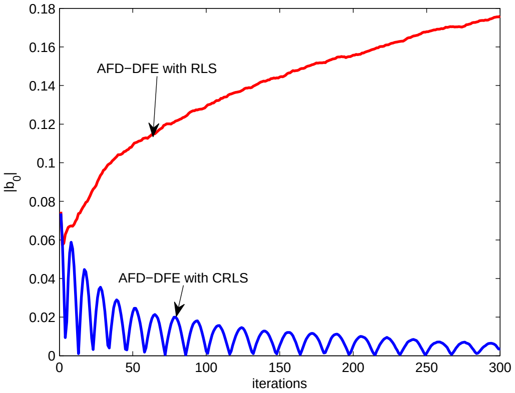


Figure 3.3: b_0 versus iteration.

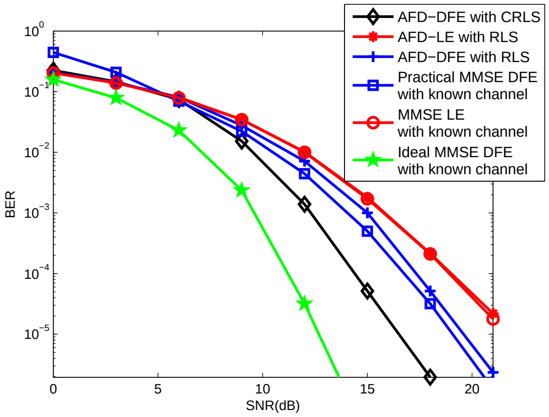


Figure 3.4: Comparison of various equalizers with user velocity $v = 3$ Km/h

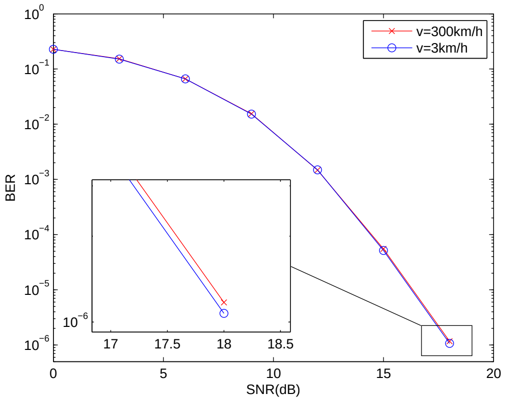


Figure 3.5: Effect of velocity on system's performance

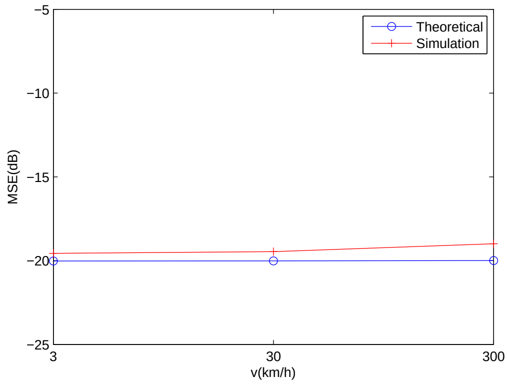


Figure 3.6: Theoretical and simulated MSE of RLS AFD-DFE as a function of user's velocity.

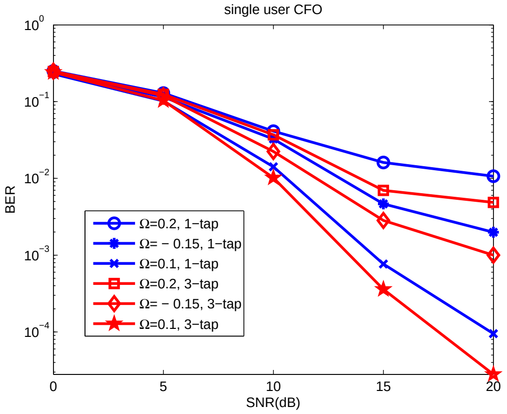


Figure 3.7: SC-FDMA system's performance under CFO's effect with user's velocity $v = 300$ Km/h

For SFBC SC-FDMA system, independent 12-path Rayleigh fading channels are used for each transmit/receive antenna pair. Fig. 3.8 shows that performance of E-SFBC with reordering is better than with reordering. Also E-SFBC outperform C-SFBC at high SNR. With CFO, 3-tap AFD-DFE is better than 1-tap AFD-DFE as depicted in Fig. 3.9. Further, we presented the effect of number of training symbols in one SC-FDMA block on MSE. As can be seen from Fig. 3.10 that if 25 percent of symbols in one SC-FDMA block is used for training and rest for data during the training phase than there is no significant decrease in the MSE. In this way we can reduce the overhead by not sending all the symbols in SC-FDMA block as training during training mode.

Finally, we have evaluated our designed AFD-DFE in blind mode where it is used in decision directed mode by utilizing stop and go blind equalization algorithm [38]. To minimize the error propagation the block size is kept small, i.e., $M = 16$ is used. Figure 3.11 depicts that the constellation at the output of the BAFD-DFE is better than BAFD-LE.

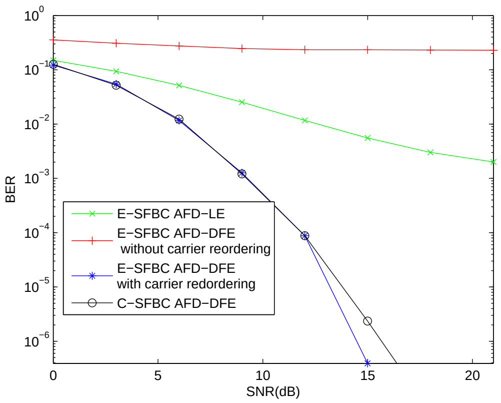


Figure 3.8: Comparison of E-SFBC and C-SFBC with user's velocity $v = 3$ Km/h

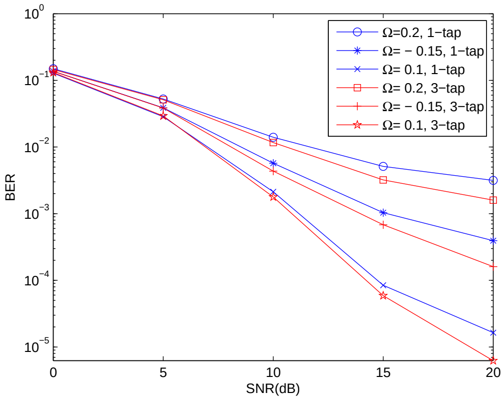


Figure 3.9: E-SFBC SC-FDMA System's performance under CFO's effect with user velocity $v = 300$ Km/h

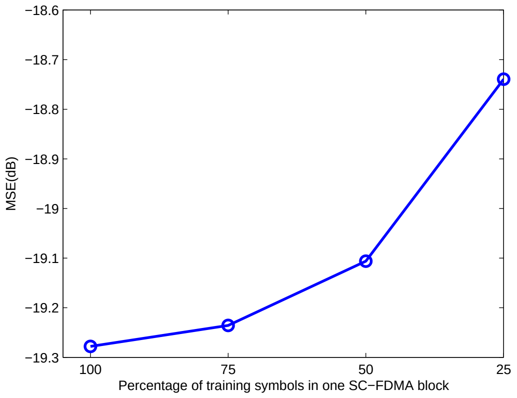


Figure 3.10: Number of training symbols in SC-FDMA block vs MSE ($v = 3$ Km/h)

Figures 3.12 and 3.13 demonstrate the convergence behavior of the BAFD-DFE using LMS and CRLS algorithms. The residual ISI at output of the AFD-LE for m^{th} user is defined as the following

$$ISI = \frac{\sum_{i=0}^{L}[|h_i^{(m)} \star w_i^{(m)}|^2] - |h_i^{(m)} \star w_i^{(m)}|^2_{max}}{|h_i^{(m)} \star w_i^{(m)}|^2_{max}} \tag{3.108}$$

where $\star$ represents the convolution operation , $w_i^{(m)}$ is the corresponding i^{th} time-domain weight of the BAFD-LE and $|h_i^{(m)} \star w_i^{(m)}|^2_{max}$ is the largest value among all the values of $|h_i^{(m)} \star w_i^{(m)}|^2$. For the BAFD-DFE, residual ISI is given as

$$ISI = \frac{\sum_{i=0}^{L}[|h_i^{(m)} \star w_i^{(m)} + b_i^{(m)}|^2] - |h_i^{(m)} \star w_i^{(m)} + b_i^{(m)}|^2_{max}}{|h_i^{(m)} \star w_i^{(m)} + b_i^{(m)}|^2_{max}} \tag{3.109}$$

where $f_i^{(m)}$ and $b_i^{(m)}$, respectively, are the corresponding i^{th} time-domain feedforward and feedback weights of the AFD-DFE. It can be seen from these figures that fast convergence is obtained by the CRLS algorithm whereas lower MSE (or residual ISI) is obtained by the LMS algorithm. The reason being that the input

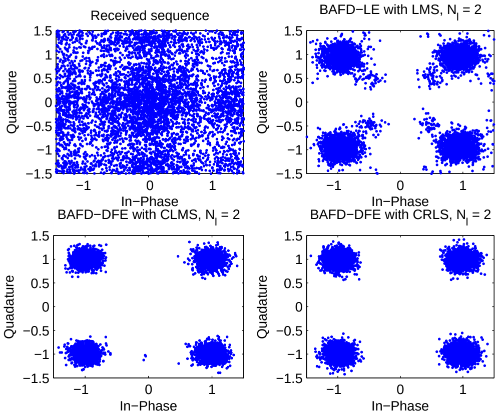


Figure 3.11: Constellation diagram, SNR= $20dB$

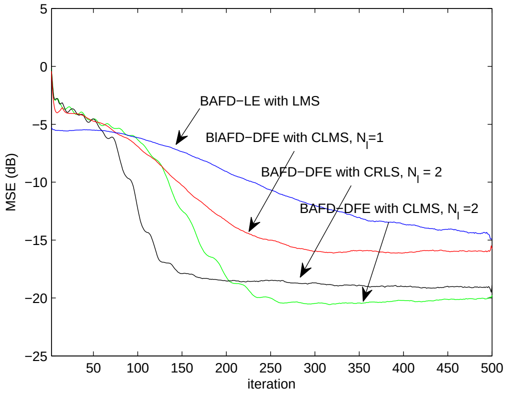


Figure 3.12: MSE convergence curves, SNR= $20dB$

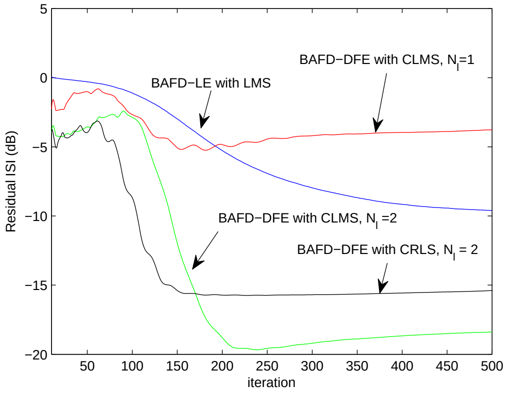


Figure 3.13: Residual ISI convergence curves,SNR= $20dB$

correction matrix in CRLS is effected by the decisions causing MSE (or residual ISI) to be higher than that of the LMS. It is also worth mentioning that no further improvement is obtained for $N_l > 2$ in case of BAFD-DFE.

3.8 Conclusion

In this work, a constraint RLS based adaptive DFE is entirely designed in the frequency-domain for SC-FDMA and extended to a SFBC SCFDMA systems. The CRLS performs better than the RLS algorithm with almost similar complexity. The equalizer operates without channel estimation at the receiver. The proposed algorithm delivers its performance at low complexity due to the special structure of matrices involved in computing the weights of the feedforward and feedback filters in the frequency domain. Our designed CRLS AFD-DFE is more computationally efficient than the non-adaptive frequency-domain DFE. Moreover, CRLS AFD-DFE outperforms the practical MMSE-DFE in terms of BER. We also showed that less training symbols can be used during training phase to reduce the overhead without sacrificing the performance. Simulation results demonstrate the significant performance gain and robustness of the proposed algorithm under the severe Doppler effect. We also extend our design to 3-tap adaptive equalizer in the frequency-domain, which has better performance than 1-tap equalizer when dealing with ICI due to CFO. Finally, blind DFE is also introduced.

CHAPTER 4

DECISION FEEDBACK EQUALIZATION USING PARTICLE SWARM OPTIMIZATION

It is well-known that the Decision Feedback Equalizer (DFE) outperforms the Linear Equalizer (LE) for highly-dispersive channels. For time-varying channels, adaptive equalizers are commonly designed based on the Least Mean Square (LMS) algorithm which, unfortunately, has the limitation of slow convergence specially in channels having large eigenvalue spread. The eigenvalue problem becomes even more pronounced in Multiple-Input Multiple-Output (MIMO) channels. Particle Swarm Optimization (PSO) enjoys fast convergence and, therefore, its application to the DFE merits investigation. In this work, we show that a

PSO-DFE with a variable constriction factor is superior to the LMS/RLS-based DFE (LMS/RLS-DFE) and PSO-based LE (PSO-LE), especially on channels with large eigenvalue spread. We also propose a hybrid PSO-LMS-DFE algorithm, and modify it to deal with complex-valued data. The PSO-LMS-DFE not only outperforms the PSO-DFE in terms of performance but its complexity is also low. To further reduce its complexity, a fast PSO-LMS-DFE algorithm is introduced. Finally, the system overhead is reduced by devising a blind PSO algorithm.

4.1 Introduction

The decision feedback equalizer (DFE) [39] is an effective Inter-symbol Interference (ISI) mitigation technique and can significantly outperform the linear equalizer (LE) on highly-dispersive channels. Adaptive equalization is attractive for time-varying channels, and for this purpose, adaptive algorithms, e.g., the Least Mean Square (LMS) and the Recursive Least Squares (RLS) [29] are widely used. Recently, heuristic techniques applied to equalization/estimation problems, in particular the Particle Swarm Optimization (PSO) technique, showed significant improvement over conventional algorithms [40–43]. It was shown in [43] that the application of PSO to an adaptive linear equalizer provides fast convergence compared to its LMS-based counterpart. To further explore its equalization capabilities, this can be applied to a DFE structure.

PSO, first introduced by Kennedy and Eberhart [44], is a swarm intelligence method and belongs to a heuristic algorithms. In PSO each particle not only act on its own and use its local information but it is also capable of sharing information with other particles to form complex structure for solving multifarious problems.

The PSO is a robust algorithm with fast convergence. It is simple, very easy to code, and has low memory requirements. It can be used to solve multi-modal, non-differential and nonlinear problems [45]. It uses position and velocity update equations to search for the global minimum. Each particle uses its own information and its neighbors information to adjust its position within the search space. In addition, the PSO works based on cooperation among the particles as opposed to the other Evolutionary Algorithms (EA). EA is based on generic operations, that are, selection, mutation and crossover to find the global minimum and unlike PSO at each generation particles are replaced by new ones. In PSO, particles are active and stay alive for the whole run, in contrast to EA. The PSO has demonstrated its distinguished performance in many engineering applications. To mention a few of its recent applications, it has been used in image processing [46], channel prediction [47], and nonlinear active noise control systems [48].

The PSO is used to optimize real and continuous-valued functions in l-dimensional space. The particles constitute the swarm, also known as population, and move in a predefined search space. The position of each particle within the search space represents a possible solution to the problem. Here, in the case of adaptive equalization, the position represents the weights of the equalizer.

Despite its advantages, PSO is vulnerable to local minima, i.e., the particles become stagnant around the global minima and may not be able to reach the global minimum [49]. To deal with this issue, we have introduced a hybrid of PSO and LMS algorithms, which not only solves the problem of particles stagnancy but also reduces the number of computations required in PSO. Another disadvantage of PSO is that it assumes real-valued data. In [43], the authors use a BPSK signal with a real-valued channel impulse response. However, in reality, we have to deal with complex numbers for pass-band transmission then the taps weights of the equalizer become complex. To solve this issue, we have modified the PSO algorithm to handle the complex case without increasing its complexity. To further reduce the complexity, we have introduced the fast PSO-LMS-DFE. Since the major complexity factor in PSO is the convolution operation required to find the equalizer output, the PSO-LMS-DFE performs this operation in the frequency-domain using the FFT to save on computations. Finally, blind PSO is also introduced to avoid the training sequence.

Here, this work is extended to Multiple-Input Multiple-Output (MIMO) scenario. Due to its high complexity, the most challenging task in designing the MIMO receiver is its corresponding MIMO channel equalizer. A MIMO equalizer has to deal with the inter symbol and the inter stream interference. Several works proposed different methods for adaptive MIMO DFE. Among them, the Vertical Bell Labs layered space-time (V-BLAST) architecture [50] is one of the promising method for MIMO equalization. Computationally efficient V-BLAST techniques have been proposed in [51–53] assuming a known channel. Its application to time varying channels is difficult due to frequently estimation of the MIMO channel. An efficient adaptive MIMO equalizer based on V-BLAST and generalized DFE [54] has been presented in [55], where the symbol detection order as well as the equalizer taps are updated recursively; however, this structure suffers from numeric instability. To address this problem, a technique based on square-root factorization of the equalizer input correlation matrix was proposed in [56]. However, unlike the application of MIMO DFE to time-invariant channels, the application of MIMO DFE to time-varying channels still requires excessive computations for the estimation of the parameters. Another challenging problem in these techniques is that substantial training is required when the equalizer length becomes large (as in [57]) and therefore a large number of symbols are needed before the algorithm converges. Algorithms based on reduced rank equalization [58] are less complex and require less training symbols as compared to full rank equalization, while requiring matrix inversion at each iteration. To overcome this problem, in [59] and [60] the covariance matrix is estimated iteratively and hence matrix inversion operation is avoided. These algorithms have a moderate complexity but unfortunately require more than 150 symbols for the training phase which is not suitable for frame-based applications, e.g., IEEE 802.11*p*, where the frame contains less than 150 Orthogonal Frequency Division Multiplexing (OFDM) symbols. All of the above mentioned techniques use the RLS algorithm which is often complex to implement and prone to instability in a real time environment. Therefore, PSO-based algorithms can be a substitute to the RLS-based algorithms with moderate complexity and guaranteed stability as they do not have to calculate the inverse of the autocorrelation matrix of the input signal.

This work reports a detailed analysis for the adjustment of the PSO parameters to ensure the best performance. The superiority of the PSO algorithms is tested on channels with different eigenvalue spreads specifically, in MIMO channels where the performance of the LMS/RLS-DFE can be very bad due to the severe eigenvalue spread problem. Our results demonstrate the performance gain of the proposed algorithms over conventional algorithms.

4.2 The PSO Algorithm

4.2.1 BASIC PSO

Initially, random solutions are assigned to n particles in a d-dimensional search space. The basic PSO algorithm [45] consists of the following elements:

Particle position ($\boldsymbol{p}_{i,k}$): The particle position is represented by a real-valued l-dimensional vector which is the potential solution to the problem at hand. The particle position is the weight vector of the equalizer in our case. The position of the i^{th} particle at instant k is denoted by $\boldsymbol{p}_{i,k} = [p_i(0), p_i(1), p_i(2), ..., p_i(l)]$, where $p_i(l)$ represents the i^{th} particle position in the l^{th} dimension.

Particle Velocity ($\boldsymbol{v}_{i,k}$): The velocity is also represented by a real-valued l-dimensional vector. The velocity of the i^{th} particle at instant k is given as $\boldsymbol{v}_{i,k} = [v_i(0), v_i(1), v_i(2), ..., v_i(l)]$, where $v_i(l)$ represents the i^{th} particle velocity in the l^{th} dimension.

Inertia weight (i_w): This parameter controls the change of velocity between successive iterations. It affects the local and global search capabilities of the particles.

Particle or local best ($\boldsymbol{pbest}_{i,k}$): Each particle remembers its best position $\boldsymbol{pbest}_{i,k}$. The best position is the one which results in the minimum (or maximum depending on the problem at hand) value of the cost function.

Global best ($\boldsymbol{gbest}_k$): The best value of all the $\boldsymbol{pbest}_{i,k}, i = 1, 2, ..., n$ is calculated by comparing the cost function values associated with them. This is the global best $\boldsymbol{gbest}_k$ of the swarm.

Stopping criteria: The algorithm is terminated when the global minimum (or maximum) is attained or after reaching a predefined number of iterations.

We constrain the particle velocity to avoid a possible overflow, i.e., the velocity is restricted as $v_{max} = v_c p_{max}$, where v_c is the velocity constraint factor and p_{max} is the maximum position .

The velocity update equation is given by

$$\begin{aligned} \boldsymbol{v}_{i,k+1} = i_w \boldsymbol{v}_{i,k} &+ c_1 * \boldsymbol{rand}_1 * (\boldsymbol{pbest}_{i,k} - \boldsymbol{p}_{i,k}) \\ &+ c_2 * \boldsymbol{rand}_2 * (\boldsymbol{gbest}_k - \boldsymbol{p}_{i,k}), \end{aligned} \tag{4.1}$$

where c_1 and c_2 are called acceleration constants, i.e., these are the rates at which local and global optima are achieved and $\boldsymbol{rand}_j$, ($j = 1, 2$), is $l \times 1$ dimension vector given as

$$\boldsymbol{rand}_j = [rand_{1,j} \quad rand_{2,j}, \quad \ldots \quad rand_{l,j}]^T \tag{4.2}$$

where $rand_{l,j}$ is a uniformly-distributed random number in $[0, 1]$ for the l^{th} dimension. The position update equation is given by

$$\boldsymbol{p}_{i,k+1} = \boldsymbol{p}_{i,k} + \boldsymbol{v}_{i,k} \tag{4.3}$$

Fig. 4.1 depicts the flow chart of the PSO algorithm.

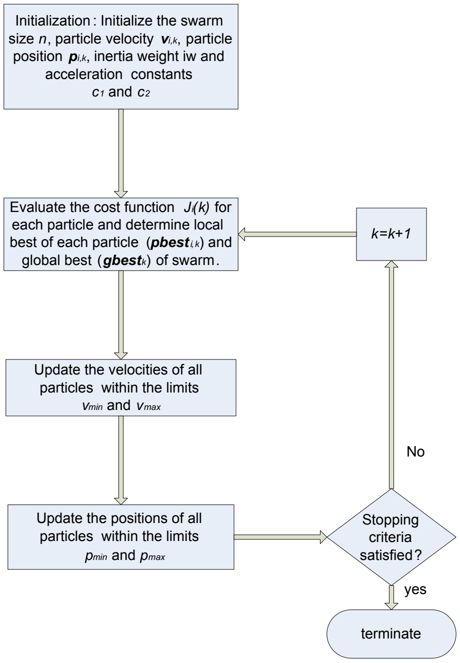

Initialization: Initialize the swarm size n, particle velocity vi,k, particle position pi,k, inertia weight iw and acceleration constants c1 and c2
Evaluate the cost function Ji(k) for each particle and determine local best of each particle (pbesti,k) and global best (gbestk) of swarm.
k=k+1
Update the velocities of all particles within the limits vmin and vmax
No
Update the positions of all particles within the limits pmin and pmax
Stopping criteria satisfied?
yes
terminate

Figure 4.1: Flow chart of the particle swarm optimization algorithm.

4.2.2 PSO Variants

The PSO algorithm has many variants and the most popular ones are:

Variable Inertia Weight: The constant inertia weight, i_w, in (4.1) is decreased linearly at each instnt k according to [61]:

$$i_w(k) = \gamma i_w(k-1), \tag{4.4}$$

where γ is a weighting factor that controls the diminution of $i_w(k)$.

Variable Constriction factor: The constriction factor based-PSO was first proposed in [62] and [63] to update the particle's velocity. Ultimately, this modification will result in a better performance than that of a standard PSO. In order to guarantee convergence, a time-varying constriction factor was recently proposed for the first time in an adaptive equalization scenario [43]. In this approach, a variable constriction factor is introduced in the velocity update equation. The velocity update equation is given as

$$\begin{aligned} \boldsymbol{v}_{i,k+1} = K(k) * [\boldsymbol{v}_{i,k} &+ c_1 * \boldsymbol{rand}_1 * (\boldsymbol{pbest}_{i,k} - p_{i,k}) \\ &+ c_2 * \boldsymbol{rand}_2 * (\boldsymbol{gbest}_k - p_{i,k})] \end{aligned}$$

and $K(k)$ is given as

$$K(k) = \frac{k_c(k)}{|1 - \Phi - \sqrt{\Phi^2 - 4\Phi}|}, \tag{4.5}$$

where Φ should be always greater than four and is defined as

$$\Phi = c_1 + c_2, \tag{4.6}$$

and

$$k_c(k) = k_{min} + (k_{max} - k_{min})\frac{iter - k}{iter - 1}, \tag{4.7}$$

where the variable *iter* is the maximum number of iterations and k is the current iteration. As the particle gets closer to the global minimum, a lower value of the constriction factor is used which helps in stabilizing the algorithm with fast convergence.

Adaptive inertia weight: In this form of PSO, the inertia weight i_w is updated based on the error value and eventually will result in high speed and efficiency. The inertia weight is changed only when a better fit is found in order to move the particle close to the optimum point. The inertia's influence is reduced if it does not attain the lowest estimation error. It maximizes the influence of potentially favorable inertia directions, while minimizing the influence of potentially unfavorable inertia directions. The adaptive inertia weight equation is given by [64]

$$i_{w_i}(k) = \frac{1}{(1 + e^{\frac{-\Delta J_i(k)}{S}})} \tag{4.8}$$

where i_{w_i} is the inertia weight of the i^{th} particle, $\Delta J_i(k)$ is the change in the cost function between the current and the previous iterations, and S is the slope factor. Using this relation will limit the inertia weight to the interval $[0, 1]$ with mean of 0.5 which corresponds to no change in the error. It means that an increase in the error will lead to an inertia weight to be more than 0.5 and vice versa.

In this work, PSO with variable constriction factor is used since its performance is better than other PSO variants [43].

4.3 Problem Formulation

Fig. 6.1 depicts the block diagram of a communication system equalized by the DFE. We will separately formulate the problem for the SISO and MIMO cases as follows.

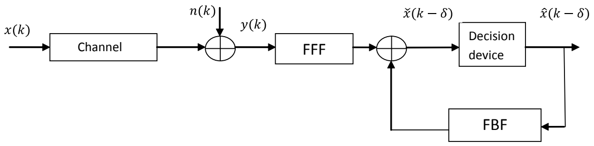


Figure 4.2: Block diagram of the decision feedback equalizer.

4.3.1 SISO Case

In the SISO case, the input data $x(.)$ is transmitted over the channel and an additive white Gaussian Noise ($n(.)$) is added. The noisy version of the channel output is denoted by $y(.)$ which is fed into the feedforward filter of the DFE. The Feedback Filter is fed with the output of the decision device $\hat{x}(.)$. Denote the lengths of feedforward and feedback filters by N_f and N_b, respectively.

At any instant k, the states of Feedforward Filter (FFF) $\boldsymbol{y}_k$ and Feedback Filter (FBF) $\boldsymbol{d}_k$ are, respectively, given as

$$\boldsymbol{y}_k = [y(k) \quad y(k-1) \quad ... \quad y(k-N_f+1)] \tag{4.9}$$

$$\boldsymbol{d}_k = \begin{cases} [x(k-\delta-1) \; ... \; x(k-\delta-N_b)], \text{for training} \\ \\ [\hat{x}(k-\delta-1) \; ... \; \hat{x}(k-\delta-N_b)], \text{for decision-directed} \end{cases} \tag{4.10}$$

where $x(k-\delta)$ is a delayed version of $x(k-\delta)$.The coefficients of the FFF and the FBF are, respectively, defined as

$$\boldsymbol{f}_k = [f_k(0) \quad f_k(1) \quad ... \quad f_k(N_f - 1)]^T, \tag{4.11}$$

$$\boldsymbol{b}_k = [b_k(1) \quad b_k(2) \quad ... \quad b_k(N_b)]^T, \tag{4.12}$$

and are combined in the vector

$$\boldsymbol{w}_k = \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{b}_k \end{bmatrix}. \tag{4.13}$$

The output of the equalizer in a decision-directed mode is given as

$$\check{x}(k) = [\boldsymbol{y}_k \quad -\boldsymbol{d}_k]\boldsymbol{w}_{k-1}, \tag{4.14}$$

Finally, the error signal is given as

$$e(k) = \begin{cases} \hat{x}(k) - \check{x}(k-\delta), & \text{for decision-directed} \\ x(k) - \check{x}(k-\delta), & \text{for training} \end{cases} \tag{4.15}$$

In this work, PSO is used to search for the tap weights that minimize the Mean Square Error (MSE). PSO is more effective when applied on the whole data but due to non-availability of the whole data in our application, we will use a block of data (with size N) and, hence, our objective function in every iteration is the computed MSE, which is defined for the i^{th} particle at the k^{th} iteration as

$$J_i(k) = \frac{1}{N} \sum_{j=1}^{N} |e_{ji}(k)|^2 \tag{4.16}$$

where $e_{ji}(k)$ is the j^{th} error of i^{th} particle and this is obtained from (4.15).

4.3.2 MIMO Case

We will formulate the problem for two transmit and two receive antennas, which can be extended to any number of transmit and receive antennas. In a 2 ×2 MIMO system, independent and identically distributed data symbols $x_1(.)$ and $x_2(.)$ are transmitted from antennas T1 and T2 over the multi-channel environment and received by antennas R1 and R2 after being corrupted by additive white Gaussian noises. $x_1(.)$ travels through channels h_{11} and h_{12} which are the respective impulse responses of the channels between transmit antenna T1 and receive antennas R1 and R2. Likewise, $x_2(.)$ travels through channels h_{22} and h_{21} which are the respective impulse responses of the channels between transmit antenna T2 and receive antennas R1 and R2. Hence, the received signals are not only corrupted by channel and noise but they also interfere with one other. The signals are assumed to be uncorrelated with each other and with the noises. The 2×2 MIMO system is depicted in Fig. 4.3. Let the received signals be $y_1(.)$ and $y_2(.)$, and define the vectors of transmitted samples, received symbols and decisions at instant k as

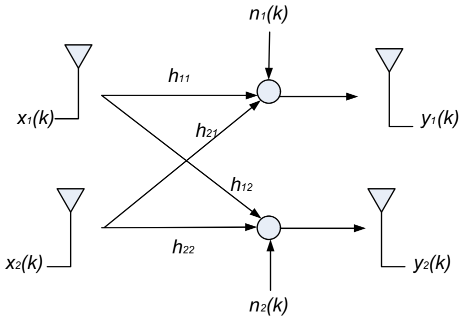


Figure 4.3: 2 × 2 MIMO system.

follows.

$$\begin{aligned}
\boldsymbol{y}(k) &\triangleq [y_1(k) \quad y_2(k)], \\
\boldsymbol{x}(k) &\triangleq [x_1(k) \quad x_2(k)], \\
\hat{\boldsymbol{x}}(k-\delta) &\triangleq [\hat{x}_1(k-\delta) \quad \hat{x}_2(k-\delta)]
\end{aligned}$$

The received vector $\boldsymbol{y}_k$ is fed into the feedforward filter with N_f matrix taps each of dimension 2×2. Similarly, the decisions are fed into a feedback filter with N_b matrix taps each of dimension 2×2. The input to the decision device is given as

$$\begin{aligned}
\check{\boldsymbol{x}}(k-\delta) = \boldsymbol{y}(k)\boldsymbol{F}_0 + \boldsymbol{y}(k-1)\boldsymbol{F}_1 \ ... \\
+ \boldsymbol{y}(k-N_f+1)\boldsymbol{F}_{N_f-1} - \hat{\boldsymbol{x}}(k-\delta)\boldsymbol{B}_0 \\
- \hat{\boldsymbol{x}}(k-\delta-1)\boldsymbol{B}_1 \ ... \ - \hat{\boldsymbol{x}}(k-\delta-N_b+1)\boldsymbol{B}_{N_b-1}
\end{aligned} \tag{4.17}$$

where $\boldsymbol{F}_0, \boldsymbol{F}_1, ..., \boldsymbol{F}_{N_f-1}$ and $\boldsymbol{B}_0, \boldsymbol{B}_1, ..., \boldsymbol{B}_{N_b-1}$ are the matrix coefficients of the feedforward and feedback filters respectively. For the decision-directed mode,

$\hat{\boldsymbol{x}}(k-\delta) = \boldsymbol{x}(k-\delta)$. Let us denote the first and second columns of $\boldsymbol{F}_0$ ($\boldsymbol{B}_0$) by $\boldsymbol{F}_{0,1}$ ($\boldsymbol{B}_{0,1}$) and $\boldsymbol{F}_{0,2}$ ($\boldsymbol{B}_{0,2}$), respectively and likewise for other matrix coefficients of the feedforward and feedback filters. Let $\boldsymbol{W}_{r,k}$,$(r = 1, 2)$ be the vector comprised of the r^{th} column of matrix taps of feedforward and feedback filters at instant k and given by.

$$\boldsymbol{W}_{r,k} \triangleq [\boldsymbol{F}_{0,r}^T \; \boldsymbol{F}_{1,r}^T \; \cdots \; \boldsymbol{F}_{N_f-1,r}^T \; \boldsymbol{B}_{0,r}^T \; \boldsymbol{B}_{1,r}^T \; \cdots \; \boldsymbol{B}_{N_b-1,r}^T]^T, \tag{4.18}$$

where $(.)^T$ represents the transpose operation. We can write (4.17) as follows.

$$\check{x}_r(k) = [\boldsymbol{y}_k \quad -\boldsymbol{d}_k]\boldsymbol{W}_{r,k-1}, \tag{4.19}$$

Now,the error signal is given as

$$e_r(k) = \begin{cases} \hat{x}_r(k) - \check{x}_r(k-\delta), \text{ for decision-directed} \\ x_r(k) - \check{x}_r(k-\delta), \text{ for training} \end{cases} \tag{4.20}$$

For PSO, the computed MSE is given as

$$J_{ri}(k) = \frac{1}{N}\sum_{j=1}^{N} |e_{rji}(k)|^2, \; r = 1, 2 \tag{4.21}$$

where $e_{1ji}(k)$ and $e_{2ji}(k)$ are the j^{th} error samples of i^{th} particle for $x_1(k)$ and $x_2(k)$ respectively.

4.4 Particle Swarm Optimization-Least Mean Squares (PSO-LMS) Algorithm

Two key problems associated with the PSO are

- The particles may become stagnant due to the lack of finer search capabilities in the PSO algorithm. Hence to achieve the global minimum value of cost function might not be possible.
- PSO enjoys fast convergence but its complexity is high and its computations depend on the size of swarm and the convolution operation required to find the equalizer output.

For these reasons, a hybrid PSO, i.e., a combination of PSO and LMS will be attractive in this situation. In [65] a combination of PSO and LMS is proposed to combat the stagnation problem. In this approach, the authors modified the particle position update equation by adding the LMS gradient term which increases the computational cost even more and the influence of LMS on PSO remains constant throughout the iterations. The overhead is also increased due to adding scaling factors for controlling the effect of both algorithms on the particle's position update equation. These difficulties have been overcome in this work by separately using the two algorithms at different times i.e. LMS, with small step size, takes over when the particles become stagnant otherwise PSO is used.

The operation of PSO-LMS algorthm is as follows. Initially, the PSO algorithm with variable constriction factor will be used for global search to explore the search space and fast convergence. As the $\boldsymbol{gbest}_k$, of the particles become stagnant, the LMS recursion is used to update the particle position to find the minimum point locally in the search space. In this way, whenever the particles become stagnant the LMS will give them direction towards the global minimum. If there is any disturbance e.g. channel changes or SNR changes, then PSO will take over the LMS since $\boldsymbol{gbest_k}$ becomes disturbed.

The steps involved in the PSO-LMS algorithm are given below:

1. **Initialization:** At k=0, initialize positions of n particles of l dimensions such that the coefficients $p_i(j), j = 1, 2, ..l$ of the i^{th} particle are uniformly distributed in the range $[p_{min}, p_{max}]$, where $p_{min} = -p_{max}$. In our case, the particle positions are the weights of the equalizer. In the same way, initialize the velocities such that the velocity coefficients $v_i(j), j = 1, 2, ..l$ are also uniformly distributed in the range $[v_{min}, v_{max}]$ where $v_{min} = -v_{max}$

2. **Cost function calculation:** Calculate the Mean Square Error (MSE) for each particle using (4.16) for SISO or (4.21) for MIMO case.

3. **Local best position:** For the first iteration, set the local best position $(\boldsymbol{pbest}_{i,k})$ of i^{th} article to the current particle position $(\boldsymbol{p}_{i,0})$ and its best MSE $J_{i,best}(k)$ to the corresponding MSE value $J_i(0)$. For the remaining iterations, if $J_i(k) < J_{i,best}(k-1)$, then set $J_{i,best}(k) = J_i(k)$ and $\boldsymbol{pbest}_{i,k} = \boldsymbol{p}_{i,k}$ and continue; else set $J_{i,best}(k) = J_{i,best}(k-1)$ and $\boldsymbol{pbest}_{i,k} = \boldsymbol{pbest}_{i,k-1}$ and continue.

4. **Global best position:** The minimum MSE $J_{min}(k)$ among all the particles best MSE $J_{i,best}(k), i = 1, 2, ..., n$ is the global best MSE $J_{min,best}$ and the corresponding position is the global best position $\boldsymbol{gbest}_k$. For any $k > 0$, if $J_{min}(k) < J_{min,best}(k-1)$, then set $J_{min,best}(k) = J_{min}(k)$ and $\boldsymbol{gbest}_k = \boldsymbol{p}_{k,min}$ where $\boldsymbol{p}_{k,min}$ is the position of that particle corresponding to $J_{min,best}(k)$ and continue; else set $J_{min,best}(k) = J_{min,best}(k-1)$ and $\boldsymbol{gbest}_k = \boldsymbol{gbest}_{k-1}$ and continue.

5. **Constriction factor update:** The constriction factor is updated using (4.5).

6. **Test for stagnancy:** Particles become stagnant as they get closer to the global minimum. At this stage, the current value of the global best becomes the same as its prior value i.e. $\boldsymbol{gbest}_k = \boldsymbol{gbest}_{k-1}$. Although the condition to check the stagnancy is to compare the current local best positions of all the particles with their pervious local best positions but this approach will be computationally very heavy. Therefore, only global values are compared. If condition of stagnancy is satisfied then go to step 9 else goto step 7.

7. **Velocity update equation:** Update the velocity of each particle according to (4.5) using the current particle velocity, local best of particle $\boldsymbol{pbest}_{i,k}$ and global best value $\boldsymbol{gbest}_k$. Restrict the coefficients $v_i(l), j = 1, 2, ...d$ in the range $[-v_{max}, v_{max}]$.

8. **Position update using PSO:** Now update the position of each particle

according to (4.3) using the velocity value calculated in the previous step. Restrict the coefficients $p_i(j), j = 1, 2, ...d$ in the range $[-p_{max}, p_{max}]$and goto step 10.

9. **Position update using LMS:** When the stagnancy test is positive then the block LMS recursion is used to update the position of each particle. For block LMS, we define equalizer input matrix as

$$\boldsymbol{Y}_k \triangleq \begin{pmatrix} y(k) & \cdots & y(k-N_f+1) \\ \vdots & \ddots & \vdots \\ y(k+N-1) & \cdots & y(k+N-N_f) \end{pmatrix} \tag{4.22}$$

and the decision matrix as

$$\boldsymbol{D}_k \triangleq \begin{pmatrix} d(k-1) & \cdots & d(k-N_b) \\ \vdots & \ddots & \vdots \\ d(k+N-2) & \cdots & d(k+N-N_b-1) \end{pmatrix} \tag{4.23}$$

where $d(k)$ is the decision on the symbol $\hat{x}(k)$ for the decision-directed mode and $x(k)$ for the training mode. For simplification the delay δ is ignored. The output of the equalizer is given as

$$\check{\boldsymbol{x}}_k = \boldsymbol{Y}_k \boldsymbol{f}_{k-1} + \boldsymbol{D}_k \boldsymbol{b}_{k-1} \tag{4.24}$$

where

$$\check{\boldsymbol{x}}_k = [\check{x}(k) \quad \ldots \quad \check{x}(k+N-1)] \tag{4.25}$$

The error at output of the equalizer is given as

$$\boldsymbol{e}_k = \boldsymbol{d}_k - \check{\boldsymbol{x}}_k \tag{4.26}$$

where

$$\boldsymbol{d}_k = [d(k) \quad \ldots \quad d(k+N-1)] \tag{4.27}$$

Now, the LMS update equation for SISO case (can easily be extended to MIMO case) is given as

$$\boldsymbol{p}_{i,k+1} = \boldsymbol{p}_{i,k} + \mu \boldsymbol{A}_k^T \boldsymbol{e}_{i,k} \tag{4.28}$$

where μ is the step size and and $\boldsymbol{e}_{i,k}$ is calculated using (6.14) for the i^{th} particle and $\boldsymbol{A}_k$ is given as

$$\boldsymbol{A}_k = \begin{bmatrix} \boldsymbol{Y}_k^H \\ \boldsymbol{D}_k^H \end{bmatrix} \tag{4.29}$$

10. **Stopping Criteria:** When the maximum number of iterations is reached, then terminate other wise, continue.
11. **Instant update:** Update the time instant counter as $k = k + 1$ and goto step 2.

Figure 4.4 depicts the flow chart of the PSO-LMS algorithm.

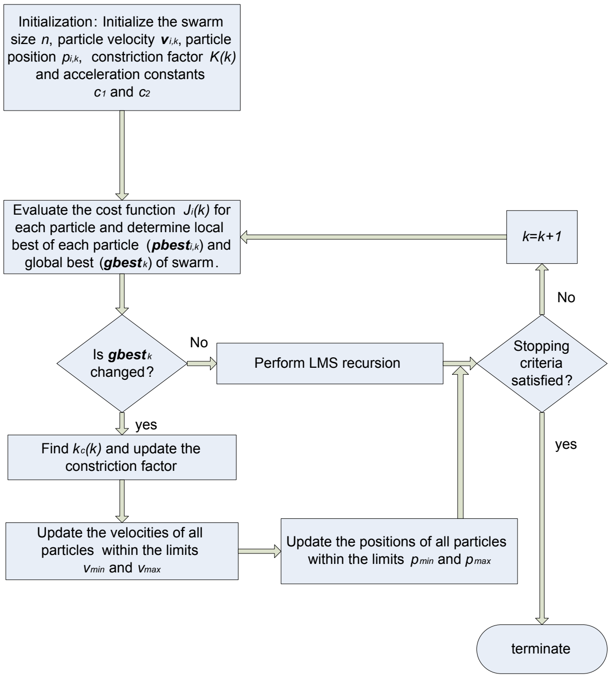

Initialization: Initialize the swarm size n, particle velocity v_{i,k}, particle position p_{i,k}, constriction factor K(k) and acceleration constants c1 and c2
Evaluate the cost function J_i(k) for each particle and determine local best of each particle (pbest_{i,k}) and global best (gbest_k) of swarm.
k=k+1
No
Is gbest_k changed?
No
Perform LMS recursion
Stopping criteria satisfied?
yes
yes
Find k_c(k) and update the constriction factor
Update the velocities of all particles within the limits v_min and v_max
Update the positions of all particles within the limits p_min and p_max
terminate

Figure 4.4: Flow chart of the PSO-LMS.

4.5 PSO-LMS-DFE For Complex-Valued Data

The PSO algorithm works on real numbers. Using PSO to find the optimum weights for complex-valued data, will not be possible without modification. One approach is to use PSO separately on the real and imaginary parts (or on the magnitude and phase) or double the dimensions and use first the d dimensions for the real part and the rest for imaginary part but this will increase the complexity. To avoid this, we have modified the PSO algorithm as follows.

1. Let the position $\boldsymbol{p}_{i,k}$ and the velocity $\boldsymbol{v}_{i,k}$ of the $i$the particle be complex. At k=0, initialize m particles of l complex dimensions, such that the real

and imaginary components of coefficients of position $p_i(l), j = 1, 2, ..l$ of the i^{th} particle are uniformly distributed in the range $[p_{im,min}, p_{im,max}]$ and $[p_{re,min}, p_{re,max}]$ respectively, where $p_{re,max} = p_{im,max} = p_{max}$ and $p_{re,min} = p_{im,min} = -p_{max}$. In the same way, the real and imaginary components of velocity coefficients $v_i(l), j = 1, 2, ..l$ of i^{th} particle are also uniformly distributed in the range $[v_{re,min}, v_{re,max}]$ and $[v_{im,min}, v_{im,max}]$ respectively, where $v_{re,max} = v_{im,max} = v_{max}$ and $v_{re,min} = v_{im,min} = -v_{max}$.

2. Calculate absolute value of the Mean Square Error (MSE) for each particle.

3. For comparison use the absolute value i.e. if $|J_i(k)| < |J_{i,best}(k-1)|$ for $i = 1, 2, 3, ..., m$, then set $J_{i,best}(k) = J_i(k)$ and $\boldsymbol{pbest}_{i,k} = P_{i,k}$ and continue; else set $J_{i,best}(k) = J_{i,best}(k-1)$ and $\boldsymbol{pbest}_{i,k} = \boldsymbol{pbest}_{i,k-1}$ and continue.

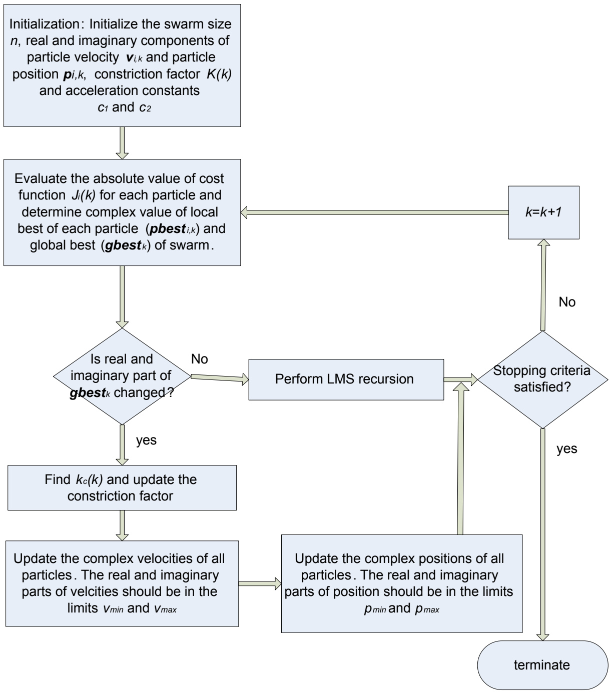
4. Similarly for the global best, if $|J_{min}(k)| < |J_{min,best}(k-1)|$, then set $J_{min,best}(k) = J_{min}(k)$ and $\boldsymbol{gbest}_k = \boldsymbol{p}_{i,k,min}$ and continue; else set $J_{min,best}(k) = J_{min,best}(k-1)$ and $\boldsymbol{gbest}_k = \boldsymbol{gbest}_{k-1}$ and continue.

5. Update the complex velocity and complex positions of each particle by using the velocity and particle update equations. Restrict the real and imaginary components of velocity coefficients $c_i(l), j = 1, 2, ...l$ in the range $[-v_{max}, v_{max}]$ and position coefficients $p_i(l), , j = 1, 2, ..., l$ in the range $[-p_{max}, p_{max}]$.

6. To test the stagnancy, if real and imaginary parts of current value of global best is same as its prior real and imaginary parts, then use LMS; otherwise,

use PSO.

Fig.4.5 depicts the flow chart of the PSO-LMS algorithm for complex-valued data.



Initialization: Initialize the swarm size n, real and imaginary components of particle velocity $v_{i,k}$ and particle position $p_{i,k}$, constriction factor K(k) and acceleration constants c_1 and c_2
Evaluate the absolute value of cost function $J_i(k)$ for each particle and determine complex value of local best of each particle ($pbest_{i,k}$) and global best ($gbest_k$) of swarm.
k=k+1
No
Is real and imaginary part of $gbest_k$ changed?
No
Perform LMS recursion
Stopping criteria satisfied?
yes
yes
Find $k_c(k)$ and update the constriction factor
Update the complex velocities of all particles. The real and imaginary parts of velcities should be in the limits v_{min} and v_{max}
Update the complex positions of all particles. The real and imaginary parts of position should be in the limits p_{min} and p_{max}
terminate

Figure 4.5: Flow chart of the PSO-LMS for complex-valued data.

4.6 Simulation Results

To test the effectiveness of the PSO algorithm when applied to a DFE for the SISO case, we have considered two time invariant channels $C1 = [1, -1.9114, .95]$ having an eigenvalue spread of 635 and $C2 = [0.408, 0.816, 0.408]$ having an eigenvalue spread of 200. The length of the feedforward and feedback filters are 4 and 2, respectively. We assume BPSK modulation. For the MIMO case, the channels h_{11}, h_{12}, h_{21} and h_{22} are taken as three-path Rayleigh i.e. the impulse response is three delta functions which are assumed to fade independently and the length of the feedforward and feedback filters are 8 and 5, respectively. The product of Doppler frequency, f_d, and the sampling time, T_s, is taken as $f_d T_s = 0.0001$ and the modulation scheme in this case is QPSK. The additive white Gaussian noise is selected with signal to noise ratio of 20 dB.

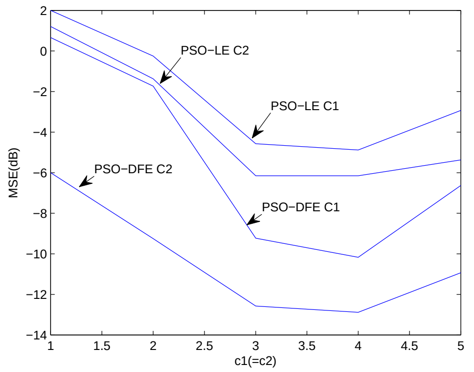


Figure 4.6: Effect of the acceleration constants (c_1 and c_2).

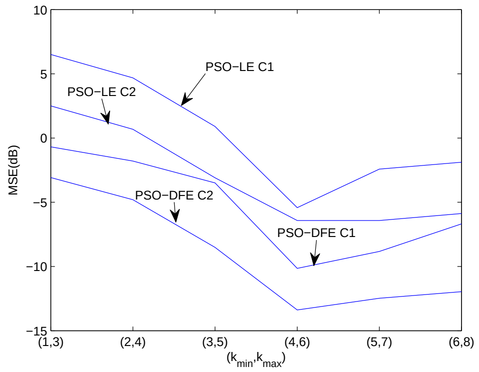


Figure 4.7: Effect of the k_{max} and k_{min}.

4.6.1 Parameter adjustment analysis

Parameter adjustment analysis for only SISO case is presented here. A similar analysis approach follows for the MIMO case. To have near optimum performance, a careful selection of the key parameters is needed. Towards this goal, we have performed thorough experiments to find the values of 5 key parameters that give minimum MSE, namely the data block size N, the population size n, the maximum velocity v_{max}, the acceleration constants c_1 and c_2, and finally, the limits k_{min} and k_{max} of the iteration-dependent factor k_c which directly controls the constriction factor K. Our analysis showed that these parameters do not depend on the modulation type, channel or number of transmit and receive antennas. Hence, once these parameters are adjusted, they are held fixed for any kind of modulation scheme or channel and for any number of transmit and receive antennas.

The outcomes of parameter adjustment analysis are shown in Figs. 4.6-4.10. In our experiments, we have used equal values of c_1 and c_2 to achieve a balance between the local and global searches and the best value achieved is 4 as shown in Fig. 4.6. With respect to the limits k_{min} and k_{max}, the optimum value achieved is $k_{min} = 4$ and $k_{max} = 6$ as shown in Fig. 4.7. Fig. 4.8 shows that no improvement in minimum MSE is obtained beyond $N = 200$.

As a general rule, a large population size will provide fast convergence but there is no significant improvement beyond $n = 40$ as shown in Fig. 4.9. Figure. 4.10 clearly shows that with in the range of $[0.01 * p_{max} \quad 0.3 * p_{max}]$ the maximum velocity of $v_{max} = 0.2 * p_{max}$ is the one that leads to the lowest MSE. For PSO-LMS, all the parameters are the same only $n = 5$ is used since $n > 5$ does not give significant performance gain. In a practical scenario, the different parameters comprising the PSO algorithm have to be selected using a sensitivity analysis, during the training phase similar to the one done here, to be able to select the compromised one.

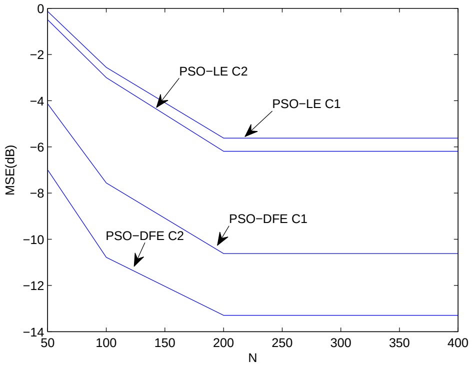


Figure 4.8: Effect of the block size (N).

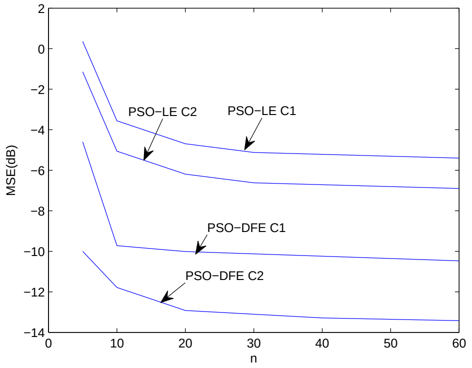


Figure 4.9: Effect of the number of particles (n).

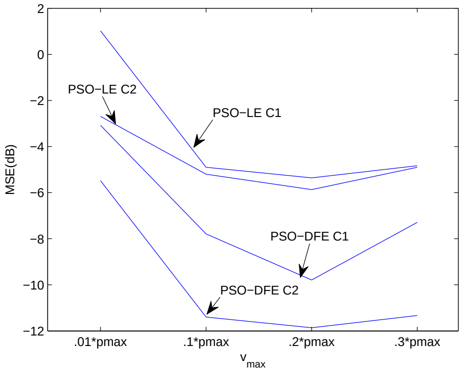


Figure 4.10: Effect of the maximum velocity (v_{max}) and maximum position (p_{max}).

4.6.2 Performance of PSO-DFE

Here, the learning and Symbol Error Rate (SER) curves are obtained for $p_{max} = 2$ and $p_{min} = -2$, and the MSE is averaged over 20 runs. To make the comparison fair, we used the block LMS/RLS. For LMS, the step-size μ is set to 0.025.

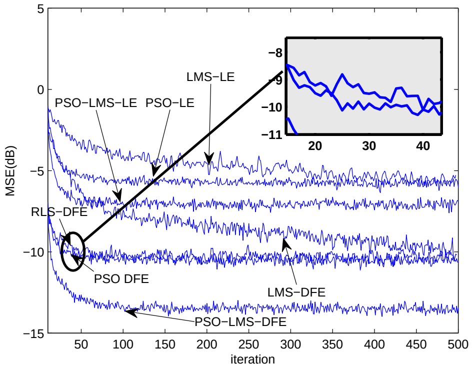


Figure 4.11: Learning Curves of different algorithms for channel $C1$.

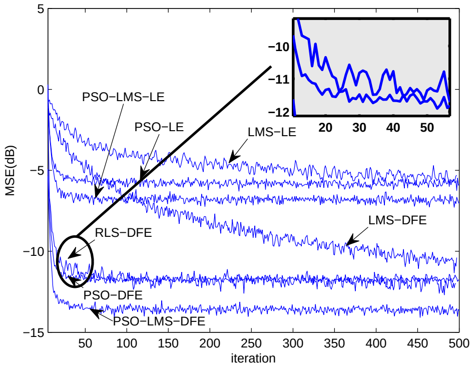


Figure 4.12: Learning Curves of different algorithms for channel $C2$.

Figures 4.11, 4.12 and 4.13 depict the learning curves for $C1$, $C2$ and MIMO channel, respectively. An improvement in convergence time and steady state MSE are achieved by the PSO-DFE over the PSO-LE and LMS/RLS. The improvement in convergence time and MSE is more pronounced in the MIMO channel which

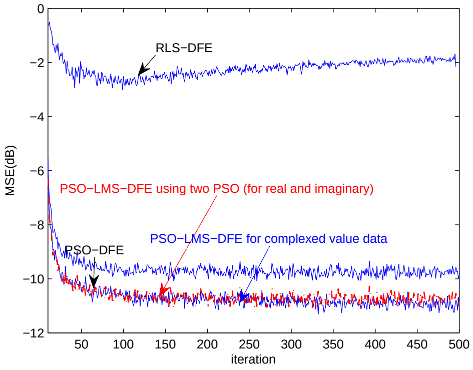


Figure 4.13: Learning Curves of different algorithms for MIMO channel.

has larger eigenvalue spread as shown in Fig. 4.13. The insensitivity of the PSO-DFE to the channel's eigenvalue spread is clear from Figs. 4.11 and 4.12. It is also clear that PSO-LMS-DFE performs even better than PSO-DFE. Figure 4.13 also depicts that the modified PSO-LMS-DFE for complex-valued data and PSO-LMS-DFE using two PSOs for real and imaginary part separately, have equal performance.

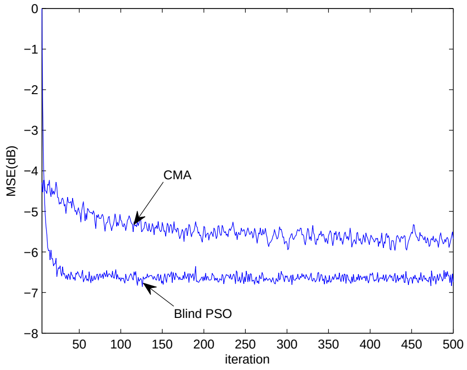


Figure 4.14: MSE curves for blind equalization.

In Fig. 4.14 the comparison of MSE curves for the case of blind equalization is shown. For blind PSO, the error used to calculate the objective function is given as

$$e(k) = \rho - |\check{x}(k)|, \tag{4.30}$$

where ρ is positive scalar quantity equal to $E|\boldsymbol{x}|^4/E|\boldsymbol{x}|^2$. Here we use a center tapping, i.e., the center tap of the equalizer is initialized to one and remaining taps or particle positions to uniform random number between $[p_{min}, p_{max}]$, for each particle. To enhance the performance of PSO, it can be combined with other blind algorithms like Constant-Modulus Algorithm (CMA), Reduced-Constellation Algorithm (RCA), Stop-and-go and Multi-Modulus Algorithm (MMA) [66–69], just to name a few. Hence when particles become stagnant, these algorithms can be used to update the weights of the equalizer.

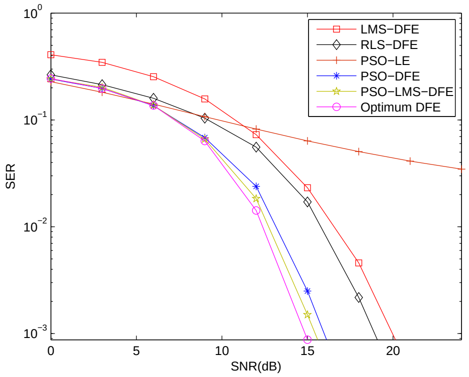


Figure 4.15: Symbol error Rate for $C1$.

Finally, Figs. 4.15, 4.16 and 4.17 show that the PSO-DFE achieves significantly lower SER than PSO-LE and LMS/RLS and its performance is very close to that of the ideal DFE (with perfect channel knowledge) even when the channel eigenvalue spread is large. Due to the fast convergence of PSO, less training symbols are required and hence a great reduction in the throughput overhead.

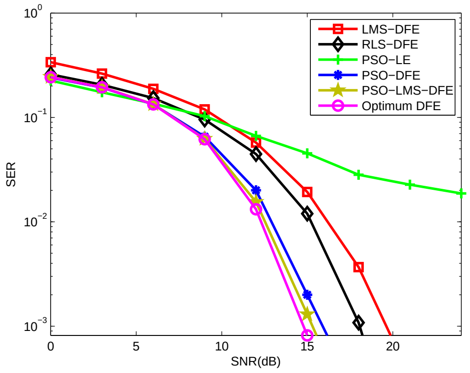


Figure 4.16: Symbol error Rate for $C2$.

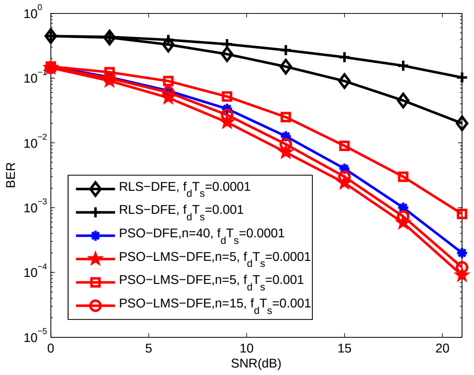


Figure 4.17: Symbol error Rate for MIMO channel.

However, these benefits are achieved at a slight increase in the computational complexity. The PSO-LMS-DFE performance is even closer to the optimum DFE and its complexity is less than that of the PSO-DFE, which is shown in next section. Further, the impact of the number of particles on the performance of the PSO-LMS-DFE algorithm is depicted in Fig. 4.17. As can be seen from this figure, by increasing the swarm size, the performance of the PSO-LMS-DFE improves very well. This is due to the fact that the number of particles have a great impact on the search space. As the number of particles increases, particles will cover all the search space and therefore, PSO will easily find the potential solution in time varying environment.

4.7 Complexity Analysis

The PSO has an advantage over the block RLS of avoiding the matrix inversion operation to update the weights of the equalizer. For complexity analysis of PSO, we will count the total number of complex additions and complex multiplications per iteration. Consider first the PSO with a variable constriction factor operating on complex data.

1. The velocity update requires three complex multiplications and five complex additions per particle per dimension. Multiplication with c_1 and c_2 in the velocity update equation is ignored as $c_1 = c_2 = 4$, which can easily be implemented using shift registers. Hence, for n particles of d dimensions each, $3dn$ complex multiplications and $5dn$ complex additions are required to update the velocity.

2. Computing the constriction factor amounts to three complex multiplications and three complex additions ignoring the calculations due to the terms Φ, $|1 - \Phi - \sqrt{\Phi^2 - 4\Phi}|$, and $(k_{max} - k_{min})$ as these can be calculated off-line. In fact, we can ignore the calculations due to constriction factor as it can be saved in a lookup table.

3. The particle update requires one complex addition.

4. Computing the output vector of the equalizer for each particle, which re-

quires dNn complex multiplication and $(d-1)Nn$ complex additions for block of size N.

5. Further Nn complex additions are required to calculate the error signal.

6. Evaluation of the MSE as given in (4.16) requires $(N-1)n$ complex additions. We can ignore the computations to calculate the square of error and the reason is that PSO compares the MSE due to all the particles and picks the one that achieves the minimum MSE so same results can be obtained by using mean error instead of MSE.

In summary, PSO with variable constriction factor for complex valued signals requires $dNn + 3dn + 3$ complex multiplications and $(d-1)Nn + 5dn + 3 + (N-1)n + Nn = dNn + 5dn + 3 + (N-1)n$ complex additions per iteration. For PSO-LMS, the algorithms switches between PSO and LMS. The LMS requires $2dN$ complex multiplications and $(d-1)N + 2d$ complex additions per particle [5]. Extensive simulations have been performed to calculate the number of updates using the LMS update equation during the whole run and it has been observed that the algorithm uses the LMS update equation for 75 out of 100 iterations on an average. Therefore, the PSO-LMS requires $0.75(2dNn) + .25(dNn + 3dn + 3)$ complex multiplications and $0.75((d-1)N + 2d)n + 0.25(dNn + 5dn + 3 + (N-1)n)$ complex additions per iterations. As it can be seen that complexity depends on the number of particles which are greatly reduced in PSO-LMS. Using $N = 200, d = 6$, and $n = 40$ for PSO ($n = 5$ for PSO-LMS), the PSO-LMS is 5 times faster than PSO in terms of multiplications and 10 times faster in terms of additions.

The convolution operation required to compute the output of the equalizer, can be performed in the frequency-domain using the overlap-save method. This will further reduce the complexity of PSO-LMS . In addition, rather than updating the parameters of PSO-LMS in the time-domain, they can be adapted in the frequency-domain. The PSO-LMS algorithm so implemented is referred to as fast PSO-LMS and it will require three M-point FFT (IFFT) as done for fast LMS in [29], where $M = 2N$ and $N = d$. Each M-point FFT (IFFT) requires $(M/2)\log_2 M$ complex multiplications and $M\log_2 M$ complex additions. Now the computation of frequency-domain output vector of equalizer for each particle requires Mn complex multiplications and $(M-1)n$ complex additions. Hence, the total number of complex multiplications and additions for fast PSO-LMS becomes $0.75(2Mn + 3(M/2)\log_2 M) + .25(Mn + 3(M/2)\log_2 M + 3Mn + 3)$ and $0.75(((M-1)+M)n + 3M\log_2 M) + 0.25((M-1)n + 3M\log_2 M + 5Mn + 3 + (M-1)n + Mn)$ per iteration respectively. For $M = 512$ and $d = N$, fast PSO-LMS is 156 times faster than PSO in terms of multiplications and 92 times faster in terms of additions. Finally, Table 4.1 summarizes the computational complexity of the different algorithms and Figs. 4.18 and 4.19 depict this comparison. Also, from these figures, it can be seen that the size of N has a great impact on the computational complexity of the different algorithms.

Table 4.1: Computational complexity of different PSO algorithms

Algorithm	Multiplications	Additions
PSO	$dNn + 3dn + 3$	$dNn + 5dn + 3$ $+(N-1)n$
PSO-LMS	$3dn/4 +$ $7Ndn/4 + 3/4$	$11dn/4 - Nn/2$ $-n/4 + Ndn + 3/4$
Fast PSO-LMS	$5Mn/2 +$ $3(M/2)\log_2 M$ $+3/4$	$7Mn/2 - 5n/4 +$ $3(M/2)\log_2 M$ $+3/4$

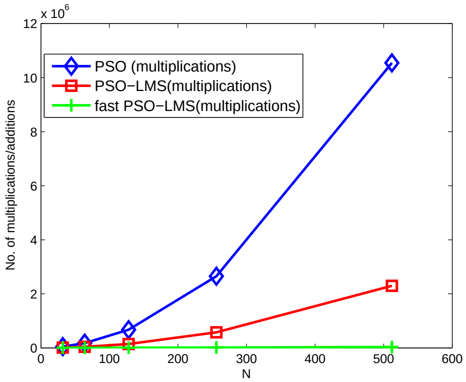


Figure 4.18: Number of multiplications of various PSO algorithms versus block size (N).

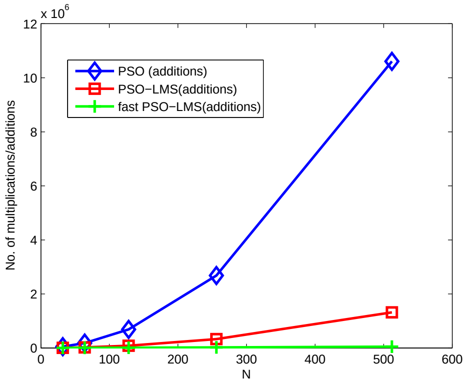


Figure 4.19: Number of additions of various PSO algorithms versus block size (N).

4.8 Conclusion

In this work, we showed how to integrate PSO with the DFE structure and demonstrated the superior performance of the PSO-DFE with respect to the widely-used

LMS/RLS-DFE. This improvement is more pronounced for highly-dispersive channels with large eigenvalue spread and MIMO channels. Moveover, extensive simulations were conducted to optimize the PSO design parameters. The PSO-DFE significantly outperforms the PSO-LE and the LMS/RLS-DFE in terms of convergence time and steady-state MSE. A hybrid algorithm, PSO-LMS-DFE, is also proposed which is not only superior to the PSO-DFE in terms of performance but also enjoys lower complexity. Further complexity reduction is achieved using the FFT in the Fast PSO-LMS-DFE.

CHAPTER 5

ADAPTIVE EQUALIZATION USING PARTICLE SWARM OPTIMIZATION FOR UPLINK SC-FDMA

Single Carrier Frequency Division Multiple Access (SC-FDMA) has been adopted as a multiple access technique for uplink in Long Term Evolution (LTE) standard. In this work, an adaptive frequency-domain equalizer for SC-FDMA system using Particle Swam Optimization (PSO) technique is proposed. Unlike stochastic gradient and Recursive Least Squares (RLS) algorithms, PSO is known to have fast convergence which does not depend on the underlying structure. The cost function used in a PSO is formulated based on the respective structure of the equalizer, whether it is Linear Equalizer (LE) or a Decision Feedback Equalizer (DFE). The robustness of our proposed PSO algorithm is demonstrated on a high Doppler scenario. Furthermore it is shown that the performance improves more when using re-randomization. Finally, it is shown that the PSO based frequency domain equalizer is more computationally efficient than its time domain counterpart.

5.1 Introduction

The most used algorithms for adaptive equalization to deal with time varying channels are Least Mean Square (LMS) and Recursive Least Square (RLS) [29]. However, their performance degrade in channels having large eigenvalue spread. Recently, Particle Swam Optimization (PSO) is used for adaptive estimation/equalization problems and showed its improved performance when compared to other conventional algorithms [40–43]. The PSO algorithm does not assume any underlying model, therefore its performance is independent of the characteristics of the system used. For this reason, PSO is expected to perform well in channel with large eigenvalue spread.

In this work, PSO is used in an adaptive frequency-domain equalizer for uplink SC-FDMA system and it is shown to have less computational complexity as compared to PSO applied to adaptive time-domain equalization. More importantly, In case of a DFE, a separate fitness functions, for feedforward and feedback filters, are developed entirely in the frequency-domain.

5.2 PSO-based adaptive equalization

Let the weight vector of the LE denoted by $\boldsymbol{\mathcal{W}}$. Using the system model described in Section (2.2), the output of the equalizer, $\check{\boldsymbol{\mathcal{X}}}_k$, in the frequency-domain at instant k is given by

$$\check{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{Z}}_k \boldsymbol{\mathcal{W}}_{k-1} \tag{5.1}$$

In case of the DFE, the output of the equalizer is

$$\check{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{Z}}_k \boldsymbol{\mathcal{F}}_{k-1} + \boldsymbol{\mathcal{D}}_k \boldsymbol{\mathcal{B}}_{k-1} \tag{5.2}$$

where $\boldsymbol{\mathcal{F}}_{k-1} = [\mathcal{F}(0)_{k-1}, \mathcal{F}(1)_{k-1}, \ \ldots \ \mathcal{F}(M-1)_{k-1}]^T$ and $\boldsymbol{\mathcal{B}}_{k-1} = [\mathcal{B}(0)_{k-1}, \mathcal{B}(1)_{k-1}, \ \ldots \ \mathcal{B}(M-1)_{k-1}]^T$ are the feedforward and the feedback filters of the DFE, respectively. Note that the exact solution of these filter coefficients is not needed in the case when using an adaptive algorithm. The decision matrix $\boldsymbol{\mathcal{D}}_k$ is defined as

$$\boldsymbol{\mathcal{D}}_k = \begin{cases} diag(F_M(\boldsymbol{x}_k)), & \text{for training} \\ \\ diag(F_M(\hat{\boldsymbol{x}}_k)), & \text{for decision-directed} \end{cases}$$

where $\hat{\boldsymbol{x}}_k$ is the decision on $\check{\boldsymbol{x}}_k$ which is given as

$$\check{\boldsymbol{x}}_k = F_M^H \check{\boldsymbol{\mathcal{X}}}_k \tag{5.3}$$

Finally, the error signal, $\boldsymbol{e}_k = [e_k(0), ..., e_k(M-1)]^T$, is given as

$$\boldsymbol{e}_k = \begin{cases} \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k, & \text{for training} \\ \hat{\boldsymbol{x}}_k - \check{\boldsymbol{x}}_k, & \text{for decision-directed} \end{cases} \tag{5.4}$$

In this section, a PSO-based adaptive frequency domain equalization algorithm is devised for SC-FDMA system. For this, the filter coefficients in (5.1) and (5.2) are calculated adaptively using PSO [43]. We denote the M-dimensional position and velocity vectors of the i^{th} particle at instant k as $\boldsymbol{p}_{i,k} = [p_{i,k}(0), p_{i,k}(1), ..., p_{i,k}(M-1)]$ and $\boldsymbol{v}_{i,k} = [v_{i,k}(0), v_i(1), ..., v_{i,k}(M-1)]$, respectively, where $p_{i,k}(l)$ represents the i^{th} particle position having velocity $v_{i,k}(l)$ in the l^{th}-dimension. Each $p_{i,k}(l)$ and $v_{i,k}(l)$ are clamped in the range $[-p_{max}, +p_{max}]$ and $[-v_{max}, +v_{max}]$, respectively, where $v_{max} = v_c p_{max}$ and v_c is the velocity constraint factor. A fitness function (cost function), discussed later, is minimized to reach the global minimum. The local and global bests in a conventional PSO are found as follows. For the i^{th} particle, among all of the particle's visited positions up to instant k, the one that gives the lowest value of the cost function is the local best of the i^{th} particle denoted by $\boldsymbol{pbest}_{i,k}$. Similarly, for the whole swarm and among all of the swarm's visited positions up to instant k, the one that gives the lowest value of the cost function is the global best of the swarm abbreviated as

$\boldsymbol{gbest}_k$. Now, the velocity update equation is given as

$$\begin{aligned}\boldsymbol{v}_{i,k+1} = K(k)[\boldsymbol{v}_{i,k} &+ c_1 * \boldsymbol{rand}_1 * (\boldsymbol{pbest}_{i,k} - \boldsymbol{p}_{i,k}) \\ &+ c_2 * \boldsymbol{rand}_2 * (\boldsymbol{gbest}_k - \boldsymbol{p}_{i,k})]\end{aligned}$$

where $\boldsymbol{rand}_j = [rand_{0,j} \quad rand_{1,j}, \quad ... \quad rand_{M-1,j}]^T$, $j = 1, 2$ and the l^{th} element $rand_{l,j}$ is uniformly-distributed number in the range $[0, 1]$. $K(k)$ is the time varying constriction factor defined in [43] and c_1 and c_2 are the positive acceleration constants satisfying $c_1 + c_2 > 4$. After updating (5.5), the i^{th} particle's position is changed according to $\boldsymbol{p}_{i,k+1} = \boldsymbol{p}_{i,k} + \boldsymbol{v}_{i,k}$.

5.2.1 Fitness Function

A PSO is more effective in off-line applications where the whole data is available; however, our case is an on-line one therefore, instead of using the whole data, a block of data, i.e., one SC-FDMA block is used. The fitness function (cost function) used in the minimization procedure at the k^{th} iteration is given as

$$J(k) = \sum_{j=0}^{M-1} |e_k(j)| \tag{5.5}$$

where $e_k(j)$ is the j^{th} error at the k^{th} instant and it is obtained from (5.4). Taking the DFT, the frequency-domain version of the error is given as

$$\mathcal{E}_k(l) = \sum_{j=0}^{M-1} e_k(j) exp(-\sqrt{-1}\ 2\pi lj/M),\ l = 0, 1, ...M-1 \tag{5.6}$$

and also

$$\boldsymbol{\mathcal{E}}_k = \boldsymbol{D}_k - \check{\boldsymbol{\mathcal{X}}}_k \tag{5.7}$$

Minimizing (5.5) in the time domain is equivalent to minimizing (5.7) in the frequency domain as (5.5) depends on $\sum_{j=0}^{M-1} |e_k(j)|$ and so does $\boldsymbol{\mathcal{E}}_k$. Therefore, the absolute value of $\boldsymbol{\mathcal{E}}_k$ will be our fitness function. As $\boldsymbol{\mathcal{E}}_k = [\mathcal{E}_k(0), ..., \mathcal{E}_k(M-1)]$ so unlike conventional PSO we define a vector of fitness functions of length M. In other words, the value of the fitness function is different for each dimension. In this way, instead of comparing particle positions to constitute $\boldsymbol{pbest}_{i,k}$ for the i^{th} particle we find best value of each dimension and $\boldsymbol{pbest}_{i,k}$ will be amalgamation of each best dimension up to instant k. Similarly, $\boldsymbol{gbest}_k$ is combinations of each best dimension among all $\boldsymbol{pbest}_{i,k}$, $i = 1, 2, ...n$, where n is the swarm size.

Above fitness function is valid for LE, incase of DFE we have to find the coefficients of both feedforward and feedback filters which is not possible by using same fitness function, Therefore, we put constraint on feedback filter. A constraint to cancel out the pre and post cursers but not the desired component was proposed in [23]. We use this constraint to formulate our fitness function for DFE. The constraint based problem for each frequency bin is given as follows

$$\min_{\mathcal{F}(l), \mathcal{B}(l)} |\mathcal{E}_k(l)|^2 \; subject \; to \; \sum_{j=0}^{M-1} \mathcal{B}_k(j) = 0, l = 0, 1, ...M-1 \tag{5.8}$$

Using Lagrange multiplier, we get

$$f(k) = |\mathcal{E}_k(l)|^2 + \alpha_k \sum_{j=0}^{M-1} \mathcal{B}_k(j) \tag{5.9}$$

Now the gradient of (5.9) with respect to $\mathcal{F}(l)$ and $\mathcal{B}(l)$ is equal to

$$\acute{f}(k)_{\mathcal{F}(l)} = \mathcal{Y}_k(l)|\mathcal{E}_k(l)| \tag{5.10}$$

$$\acute{f}(k)_{\mathcal{B}(l)} = \mathcal{Y}_k(l)|\mathcal{E}_k(l)| + \alpha_k \tag{5.11}$$

PSO compares the fitness value of all the particles and pick the one that gives the lowest value, as the term $\mathcal{Y}_k(l)$ in (5.10) is commonly for all the particles so we ignore this term and the fitness function for feedforward filter is $\acute{f}(k)_{\mathcal{F}(l)} = |\mathcal{E}_k(l)|$, which is same as for the linear equalizer. α_k is updated according to the stochastic gradient method,

$$\alpha_{k+1} = \alpha_k + \mu \sum_{j=0}^{M-1} \mathcal{B}_k(j) \tag{5.12}$$

where μ is the step size.

5.2.2 Re-randomization

One of the problems of the PSO is that once a *gbest* is found, then all particles start to move towards it and hence become stagnant around the global minima leaving empty spaces in the search space. In our scenario, we are using PSO not only to avoid providing channel information at the SC-FDMA receiver but also to track the variation in the channel. Therefore, due to the time- varying nature of the problem, the values of the equalizer taps are not fixed and if the particles become stagnant in one place, the then PSO will not be able to find the plausible solution. To tackle this issue, re-randomization is proposed. In this method, the particles are re-randomized around $gbest_k$ after certain time instants, except during the training phase. In the training phase, the main objective is to enable the particles to search for the global minimum quickly. However re-randomization during this phase, will slow down the speed of convergence of the global minimum search process. The benefit of re-randomization is its capability to allow the particles a higher probability of finding the best solution in time varying environments. As such it also yields a better Bit Error Rate (BER). Re-randomization here can be thought of retraining in RLS/LMS which is used to avoid divergence in these algorithms. Therefore, re-randomization will not only improve the performance but also reduce the overhead that would otherwise be required in retraining blocks needed in RLS/LMS.

5.3 Complexity Analysis

In this section, the computational complexity of the PSO algorithm operating in the frequency-domain (FD-PSO) is compared with PSO operating in the time-domain (TD-PSO) for a LE (for DFE, the complexity roughly becomes twice). The comparison is based on the total number of complex multiplications and additions. For the TD-PSO, the number of complex multiplications and additions required are $M^2n+3Mn+3$ and $M^2n+5Mn+(M-1)n+3$, respectively [43]. In FD-PSO, the convolution operation to find the output of the equalizer is replaced by multiplication, therefore, in this case $Mn+3Mn+3$ complex multiplications and $5Mn+3$ complex additions are required. To illustrate the computational complexity, for $M=512$, FD-PSO is 128 times faster than TD-PSO in terms of multiplication and 103 times faster in terms of addition.

Table 5.1 summarizes the computational complexity of both algorithms. Although the FD-PSO is computationally heavier than the frequency-domain RLS because of the number of particles, it is nevertheless still preferred to use PSO because of its superior performance over that of the RLS. Simulation results will substantiate this fact. Moreover, as processing is carried out in the base station (uplink scenario), the computational complexity is not any more problematic for the FD-PSO algorithm.

Table 5.1: Computational complexity of PSO algorithms

Algorithm	Multiplications	Additions
TD-PSO	$M^2n+3Mn+3$	$M^2n+6Mn-n+3$
FD-PSO	$4Mn+3$	$5Mn+3$

5.4 Simulation Results

In this section, the FD-PSO algorithm is tested on a LTE system with a carrier frequency of 2GHz and a bandwidth of 5MHz. Here, the FD-PSO is implemented using the PSO for complex-value data. Quadrature Phase Shift Keying (QPSK) modulation with interleaved mapping on a 3-path Rayleigh fading channel is used. $M = 512$, $n = 40$ and the rest of the parameters are similar to those used in [43].

Figure 5.1 reports on the performance brought about by the use of the FD-PSO. The product of Doppler frequency, f_d, and the sampling time, T_s, is $f_d T_s = 0.0001$. Almost a 2dB improvement in BER, at BER $= 10^{-3}$, has been achieved by the FD-PSO over the RLS algorithm. Moreover, it can be seen that the worst performance is obtained by the LMS-DFE. Further the impact of number of the particles and re-randomization is depicted in Fig. 5.2. By increasing the swarm size, we can improve the performance of the proposed algorithm and this is due to the fact that the particles will cover all the search space, thus allowing the PSO to easily find the potential solution. Similarly, re-randomization prevents all the particles from converging to a single point and thus improves performance . At a high Doppler, the PSO performs much better than RLS as shown in Fig.5.2.

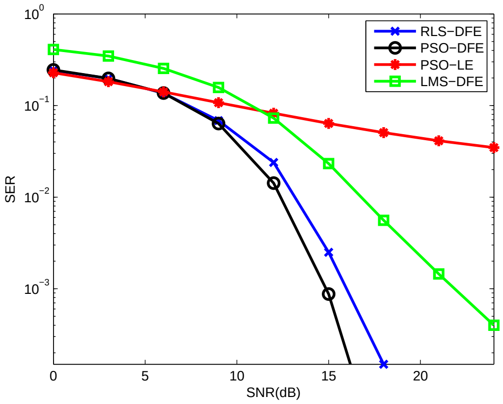


Figure 5.1: Comparison of PSO algorithm with RLS and LMS for $f_d T_s = 0.0001$

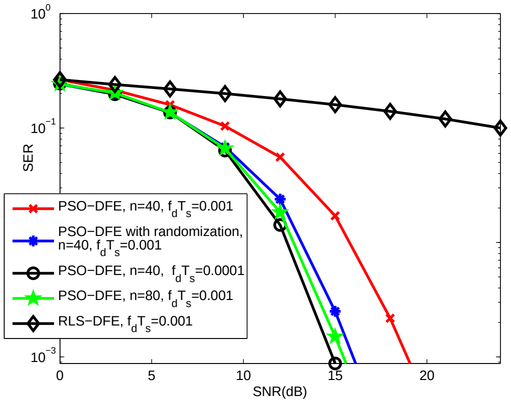


Figure 5.2: Effect of Doppler on PSO algorithm

5.5 Conclusion

A FD-PSO-based adaptive equalization in a SC-FDMA system is proposed in this letter. Simulation results verified that our devised adaptive equalization scheme has a better performance than that of RLS and LMS. Furthermore, in case of a high Doppler, the performance of the FD-PSO algorithm can be improved by increasing the number of particles and the use of re-randomization.

CHAPTER 6

SPARSE LEAST MEAN SQUARE (LMS) ALGORITHM

There are many adaptive algorithms for adaptive equalization, such as Least Mean Squares (LMS) and Recursive Least Squares (RLS). However, these algorithms have no particular advantage in sparse system due to no use of sparse characteristics.

6.1 Introduction

The existing Sparse versions of the LMS are l_0 norm constraint LMS (l_0-LMS), l_1 norm constraint LMS (l_1 -LMS) and l_0 norm constraint Exponentially Forgetting Window LMS (l_0-EFWLMS) [34, 70, 71] , with their normalized and variable step size versions [72]. Generally, these algorithms use the following update equation

$$\begin{aligned} \textit{updated tap weight vector} \quad &= \quad \textit{old tap weight vector} \; + \; \textit{gradient term} \\ &\quad + \; \textit{zero attractor term} \end{aligned} \tag{6.1}$$

As it can be seen that(6.1) contains an LMS update and a zero-attractor term which is used to guarantee the sparsity by attracting the tap coefficients to zero. These algorithms achieve the lower Mean Square Error (MSE) than the convectional LMS in sparse system identification problem but suffer from slow convergence because the gradient and zero-attractor terms in update equation (6.1) are hard to balance. To achieve fast convergence, l_0 Zero Attraction Projection (l_0-ZAP) algorithm is proposed in [71] but it is unable to achieve the lower MSE. Moreover, these algorithms have higher complexity than the conventional LMS because of an additional zero-attractor term.

Our devised Sparse LMS (SLMS) is different from the aforementioned algorithms in a sense that it only updates the significant taps which reduces the computational complexity tremendously with great performance improvement.

In adaptive equalization, one of the issue that badly effects the performance of an equalizer is the number of the taps and decision delay required to perfectly equalize the channel. The decision delay determines which symbol is detected at current instant. Its value can vary from 0 to the one less than the number of total taps. Both number of taps and decision delay significantly affects the performance of the equalizer. Using small number of taps does not serve the purpose, on the other hand, large number of taps will not only increase the complexity but also the EMSE. Hence, there always exit an optimum number of the taps that is best balance between the performance and complexity. Moreover, for a specific length of the equalizer there is an optimum delay which gives lower MMSE. Normally the number of taps and decision delay are predicted based on the length of the channel, which itself is not known at the receiver.

Our devised SLMS not only gives the sparse solution of the problem at hand but it also gives the optimum number of the taps for a specific decision delay, necessary to achieve the best performance. The optimum number of taps varied with the signal-to-noise level for a specific channel.

In SLMS, first the significant taps are calculated using sparse signal recovery algorithm and then only these tap weights are updated using LMS. Before proceeding further, brief overview about the sparse signals recovery is provided.

Given the system of equation $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{w}+\boldsymbol{n}$, where $\boldsymbol{y}$, $\boldsymbol{H}$, $\boldsymbol{w}$ and $\boldsymbol{n}$ are measurement vector, measurement matrix, unknown vector and noise vector, respectively, the sparsest solution can be obtained by solving the following problem

$$\min_{\hat{\boldsymbol{w}}} ||\hat{\boldsymbol{w}}||_0 \text{ subject to } ||\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{w}}||_2^2 \leq \varepsilon \tag{6.2}$$

where ε is chosen to bound noise in measurements. Generally, finding the optimum solution to this problem is computationally not feasible. There are two main approaches to find the suboptimal solution, namely, l_1-norm minimization and greedy algorithms. In l_1-norm minimization, the resulting solution is not exactly sparse because many small entries will exit in $\acute{\boldsymbol{w}}$. On the other hand, greedy algorithms provide more control over number of non-zero entries whose locations and values are determined iteratively. Moreover, finding the solution iteratively matches to our application here, since LMS algorithm is also iterative. Among the greedy algorithms, one of the widely used algorithm is Orthogonal Matching Pursuit (OMP) [73]. It takes $\boldsymbol{y}$, $\boldsymbol{H}$ and a certain stopping criteria as its inputs and computes a sparse solution $\acute{\boldsymbol{w}}$. Hence, OMP is denoted by $\acute{\boldsymbol{w}} = \text{OMP}(\boldsymbol{y},\boldsymbol{H},$ stopping criteria). The stopping criteria may be a predefined sparsity level or bound on residual term, i.e., $||\boldsymbol{y} - \boldsymbol{H}\acute{\boldsymbol{w}}||_2^2$.

Now, to develop SLMS, first the symbol by symbol and block equalization are formulated as follows.

6.2 Symbol by Symbol equalization

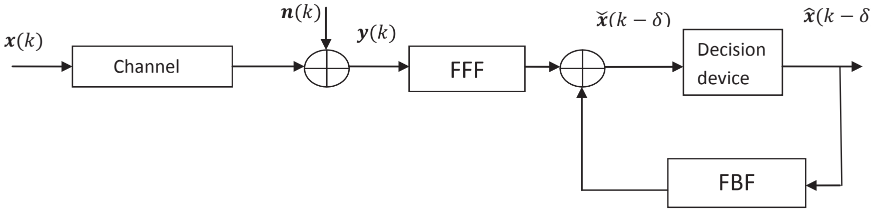


Figure 6.1: Block diagram of the decision feedback equalizer.

Figure 6.1 depicts the block diagram of a communication system equalized by the DFE. We will formulate the problem for N_t transmit and N_r receive antennas as follows.

In a $N_t \times N_r$ MIMO system, independent and identically distributed data symbols $\{x_1(.), x_2(.), ..., x_{N_t}(.)\}$ are transmitted from antennas $T_1, T_2, ..., T_{N_r}$, respectively, over the multi-channel environment and received by antennas $R_1, R_2, ..., R_{N_r}$ after being corrupted by additive white Gaussian noises $\{n_1(.), n_2(.), ..., n_{N_r}(.)\}$. $x_1(.)$ travels through channels $h_{11}, h_{12}, ..., h_{1N_r}$ which are the respective impulse responses of the channels between transmit antenna T_1 and receive antennas $R_1, R_2, ..., R_{N_r}$. Likewise, $x_2(.)$ travels through channels $h_{21}, h_{22}, ..., h_{2N_r}$ which are the respective impulse responses of the channels between transmit antenna T_2 and receive antennas $R_1, R_2, ..., R_{N_r}$ and similarly for other transmit antennas. Hence, the received signals are not only corrupted by channel and noise but they also interfere with one other. The signals $\{x_1(.), x_2(.), ..., x_{N_t}(.)\}$ are assumed to be uncorrelated with each other and with the noises.

Let the received signals be $\{y_1(.), y_2(.), ..., y_{N_r}(.)\}$, and define the vectors of transmitted samples, received symbols and decisions at instant k as follows.

$$\begin{aligned}
\boldsymbol{y}(k) &\triangleq [y_1(k) \quad y_2(k) \quad ... \quad y_{N_r}(k)], \\
\boldsymbol{x}(k) &\triangleq [x_1(k) \quad x_2(k) \quad ... \quad x_{N_t}(k)], \\
\hat{\boldsymbol{x}}(k-\delta) &\triangleq [\hat{x}_1(k-\delta) \quad \hat{x}_2(k-\delta) \quad ... \quad \hat{x}_{N_t}(k-\delta)]
\end{aligned}$$

where $\hat{\boldsymbol{x}}(k-\delta-1)$ is a delayed version of $\hat{\boldsymbol{x}}(k)$. At any instant k, the states of feedforward filter (FFF) $\boldsymbol{y}_k$ and feedback filter (FBF) $\boldsymbol{d}_k$ are, respectively, given as

$$\boldsymbol{y}_k = [\boldsymbol{y}(k) \quad \boldsymbol{y}(k-1) \quad ... \quad \boldsymbol{y}(k-N_f+1)] \tag{6.3}$$

$$\boldsymbol{d}_k = \begin{cases} [\boldsymbol{x}(k-\delta-1)\ ...\ \boldsymbol{x}(k-\delta-N_b)], \text{for training} \\ \\ [\hat{\boldsymbol{x}}(k-\delta-1)\ ...\ \hat{\boldsymbol{x}}(k-\delta-N_b)], \text{for decision directed} \end{cases} \tag{6.4}$$

The received vector $\boldsymbol{y}_k$ is fed into the feedforward filter with N_f matrix taps each of dimension $N_t \times N_r$. Similarly, the decisions are fed into a feedback filter with N_b matrix taps each of dimension $N_t \times N_r$. The input to the decision device is given as

$$\begin{aligned}\check{\boldsymbol{x}}(k) &= \boldsymbol{y}(k)\boldsymbol{F}_0 + \boldsymbol{y}(k-1)\boldsymbol{F}_1\ ... \\ &\quad + \boldsymbol{y}(k-N_f+1)\boldsymbol{F}_{N_f-1} - \hat{\boldsymbol{x}}(k-\delta)\boldsymbol{B}_0 \\ &\quad - \hat{\boldsymbol{x}}(k-\delta-1)\boldsymbol{B}_1\ ...\ - \hat{\boldsymbol{x}}(k-\delta-N_b+1)\boldsymbol{B}_{N_b-1}\end{aligned} \tag{6.5}$$

where $\boldsymbol{F}_0, \boldsymbol{F}_1, ..., \boldsymbol{F}_{N_f-1}$ and $\boldsymbol{B}_0, \boldsymbol{B}_1, ..., \boldsymbol{B}_{N_b-1}$ are the matrix coefficients of the feedforward and feedback filters respectively. For the decision-directed mode, $\hat{\boldsymbol{x}}(k-\delta) = \boldsymbol{x}(k-\delta)$. Let us denote the columns of $\boldsymbol{F}_0$ ($\boldsymbol{B}_0$) by $\boldsymbol{F}_{0,1}$ ($\boldsymbol{B}_{0,1}$), $\boldsymbol{F}_{0,2}$ ($\boldsymbol{B}_{0,2}$),...,$\boldsymbol{F}_{0,N_r}$ ($\boldsymbol{B}_{0,N_r}$) and likewise for other matrix coefficients of the feedforward and feedback filters. Let $\boldsymbol{W}_{r,k-1}$,($r = 1, 2, ..., N_r$) be the vector comprised of the rth column of matrix taps of feedforward and feedback filters at instant $k-1$ and given by.

$$\boldsymbol{W}_{r,k-1} \triangleq [\boldsymbol{F}_{0,r}^T\ \boldsymbol{F}_{1,r}^T\ ...\ \boldsymbol{F}_{N_f-1,r}^T\ \boldsymbol{B}_{0,r}^T\ \boldsymbol{B}_{1,r}^T\ ...\ \boldsymbol{B}_{N_b-1,r}^T]^T, \tag{6.6}$$

where $(.)^T$ represents the transpose operation. We can write (6.5) as follows.

$$\check{x}_r(k) = [\boldsymbol{y}_k \quad -\boldsymbol{d}_k]\boldsymbol{W}_{r,k-1}, \tag{6.7}$$

Now, the error signal is given as

$$e_r(k) = \begin{cases} \hat{x}_r(k-\delta) - \check{x}_r(k), \text{ for decision-directed} \\ x_r(k-\delta) - \check{x}_r(k), \text{ for training} \end{cases} \tag{6.8}$$

Now the weights will be updated according to stochastic gradient algorithm (LMS)

$$\boldsymbol{W}_{r,k} = \boldsymbol{W}_{r,k-1} + \mu[\boldsymbol{y}_k \quad -\boldsymbol{d}_k]^H e(k) \tag{6.9}$$

6.3 Block equalization

For the Block equalizer , let the block length is L. Defining the matrix of received symbols as

$$\boldsymbol{Y_k} = \begin{pmatrix} \boldsymbol{y}(k) & \cdots & \boldsymbol{y}(k-N_f+1) \\ \vdots & \ddots & \vdots \\ \boldsymbol{y}(k+L-1) & \cdots & \boldsymbol{y}(k+L-N_f) \end{pmatrix} \tag{6.10}$$

and the decision matrix,

$$\boldsymbol{D_k} = \begin{pmatrix} \boldsymbol{d}(k-\delta-1) & \cdots & \boldsymbol{d}(k-\delta-N_b) \\ \vdots & \ddots & \vdots \\ \boldsymbol{d}(k-\delta+L-2) & \cdots & \boldsymbol{d}(k-\delta+L-N_b-1) \end{pmatrix} \tag{6.11}$$

where $\boldsymbol{d}(k)$ is the decision on the symbol $\check{\boldsymbol{x}}(k)$ in case of decision-directed mode and $\boldsymbol{x}(k)$ in case of training mode. The output of the equalizer is given as

$$\check{\boldsymbol{x}}_{\boldsymbol{k}} = [\boldsymbol{Y}_k \quad -\boldsymbol{D}_k]\boldsymbol{W}_{r,k-1} \tag{6.12}$$

where

$$\check{\boldsymbol{x}}_k = [\check{x}(k) \quad ... \quad \check{x}(k+L-1)] \tag{6.13}$$

The error at output of the equalizer is

$$\boldsymbol{e}_k = \boldsymbol{d}_k - \check{\boldsymbol{x}}_k \tag{6.14}$$

where

$$\boldsymbol{d_k} = [d(k-\delta) \quad ... \quad d(k-\delta+L-1)] \tag{6.15}$$

Defining $\boldsymbol{A}_k$ as

$$\boldsymbol{A}_k = [\boldsymbol{Y}_k \quad \boldsymbol{D}_k] \tag{6.16}$$

6.4 Sparse LMS

To enhance the performance of LMS, the SLMS is developed as follows.

- Find the locations and values of the non zero taps by using the estimates of the correlation matrices i.e.

$$\hat{\boldsymbol{w}}_1^s = (\underbrace{\boldsymbol{A_1}^H\boldsymbol{A_1}}_{R_{yy}})^{-1}\underbrace{\boldsymbol{A_1}^H\boldsymbol{d_1}^T}_{R_{xy}} \tag{6.17}$$

- Use the same locations to update the weights using LMS algorithm.

we will use the OMP algorithm to solve (6.17). The overall algorithm is explained as following

Step 1: Find the locations of the non zero taps, $LO(\hat{\boldsymbol{w}}_1^s)$, and initial value of the non zero taps, $\hat{\boldsymbol{w}}_1^s$, using the estimates of the correlation matrices. Since matrix inversion is involved in (6.17) so we will use the OMP algorithm to find the sparsest solution of (6.17) as

$$\hat{\boldsymbol{w}}_1^s = OMP(\underbrace{\boldsymbol{A_1}^H\boldsymbol{d_1}^T}_{\triangleq y}, \underbrace{\boldsymbol{A_1}^H\boldsymbol{A_1}}_{\triangleq H}, \underbrace{||\boldsymbol{A_1}^H\boldsymbol{A_1}\hat{\boldsymbol{w}}_1^s - \boldsymbol{A_1}^H\boldsymbol{d_1}^T||_2^2 \leq \varepsilon}_{stopping\ criteria}) \tag{6.18}$$

The working of OMP algorithm in step 1 is as follows.

- INITIALIZATION: Initialize the index set $I_0 = [\ ]$, residual $\boldsymbol{r}_0 = \boldsymbol{y}$, $\hat{\boldsymbol{w}}_1^s = \boldsymbol{0}$ and $t = 1$.

- THE t^{th} ITERATION:

1. Compute $\delta_i = |\boldsymbol{r}_{t-1}^h \boldsymbol{H}(:,i)|$, $\forall i \notin I_{t-1}$
2. Choose $m_t = arg \max_i \delta_i$.
3. Update $I_t = I_{t-1} \cup m_t$. i.e., the indices of the non-zeros elements are augmented by m_t.
4. Compute $\hat{\boldsymbol{w}}_1^s(I_t) = (\boldsymbol{H}(:,I_t))^\dagger \boldsymbol{y}$, where $\hat{\boldsymbol{w}}_1^s(I_t)$ holds the elements of $\hat{\boldsymbol{w}}_1^s$ indexed by I_t.
5. Compute $\boldsymbol{r}_t = \boldsymbol{y} - \boldsymbol{H}(:,I_t)\hat{\boldsymbol{w}}_1^s(I_t)$, where $\boldsymbol{r}_t$ is the residual error at the t^{th} iteration.
6. If $||\boldsymbol{H}\hat{\boldsymbol{w}}_1^s - y||_2^2 \leq \varepsilon$, exit the algorithm, else set $t = t+1$ and go to step 1.

Step 2: Recursions of LMS ($k > 0$): By using the known locations and initial taps obtain from step 1, only update the corresponding weights of known locations using (6.9).

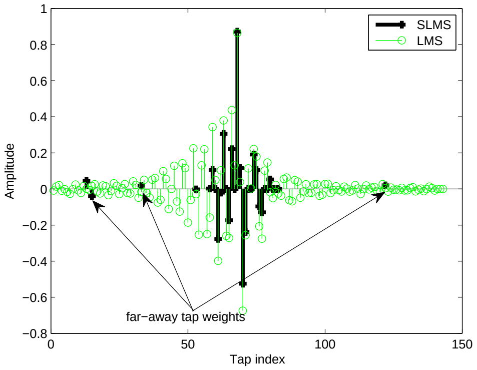


Figure 6.2: Impulse response of the equalizer using LMS and SLMS with $L = 200$ and SNR=20 dB.

Figure 6.2 depicts the impulse response of the equalizer using conventional LMS and SLMS for a ITU vehicular A channel [74], which spans over fourteen symbol durations. Therefore, the 'far-away' weights, indicated in Fig. 6.2, are not needed, in fact these weights create interference inside the equalizer. To eliminate this problem, the OMP algorithm is slightly modified. The OMP algorithm gives the location of the strongest tap in first iteration if the columns of the sensing matrix H have equal norm [75]. Now in our case, as can be seen from Fig. 6.2, that all significant taps are closely packed. Therefore, instead of finding the maximum (step (2) in OMP) in each iteration of the OMP algorithm, the maximum (strongest) tap is determined in the first iteration only and then for the remaining iterations the adjacent taps are used and checked for the stopping criteria. The overall procedure is as follows. In the first iteration, OMP gives the location of the strongest tap, which we call as i^{th} location. In next two iterations, $(i-1)^{th}$ and $(i+1)^{th}$ locations will be selected and tested for stopping criteria and then in 4^{th} and 5^{th} iterations $(i-2)^{th}$ and $(i+2)^{th}$ locations, respectively, will be used and tested, until the stopping criteria is met. In this way, not only the 'far-away' weights are eliminated but the computationally complexity is also reduced due to avoiding the need of finding the maximum among all taps locations in each iteration. We termed this approach as Dominant Tap Approach (DTA). Figure 6.2 depicts the impulse response of the equalizer using conventional LMS and DTA. In summary, we start with large number of taps and a specific decision delay then by using OMP, we get the optimum number of taps with a decision delay which gives the lower MSE. In the simulation , the value of ε is set to be equal to the noise variance, step size μ is 0.005 and $\delta = N_f/2$.

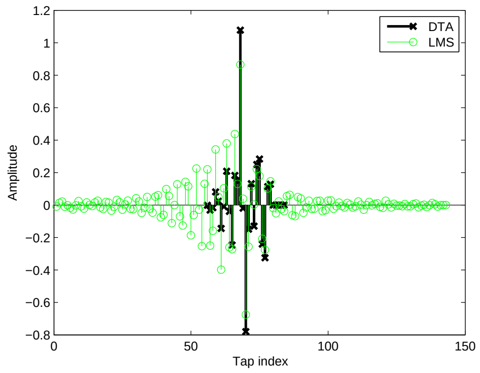


Figure 6.3: Impulse response of the equalizer using LMS and SLMS with dominant tap approach, $L = 200$ and SNR=20 dB.

6.5 Computational Complexity

The complexity of Sparse LMS is less as compared to normal LMS due to updating the significant weights only. On the other hand, using OMP at the start will increase the complexity of the DTA. To compare the compuational complexity, we count the number of multiplications/additons in LMS and OMP algorithms. The LMS algorithm requires $8M$ multiplication/addition [5] whereas, the OMP algorithm requires $(K+1)(N_f+N_b)^2 + K^2(3(N_f+N_b)/2 + K^2/12)$ multiplication/addition [76], where K is the sparsity level. Figure 6.4 depicts the comparison of the computational complexity for different values of active taps percentage (optimum taps by OMP / total number of taps * 100). The curve with 100 percentage of active taps is for conventional LMS. It can be seen that at start the computational complexity of DTA is high but as the iterations progress, DTA takes over the LMS algorithm.

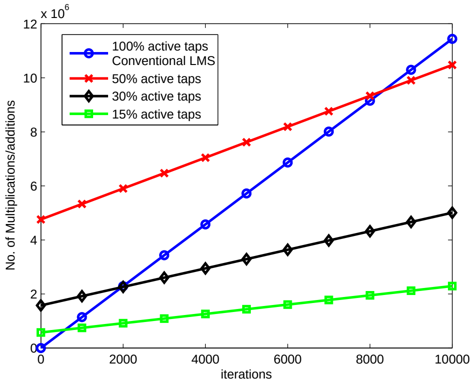


Figure 6.4: Computational Ccmparison of LMS and SLMS.

6.6 Simulation Results

We have used $N_f = 140$, $N_b = 0$ for Linear equalizer and $N_f = 140$, $N_b = 3$ for the DFE, with ITU vehicular A channel. Binary Phase Shift Keying (BPSK) modulation is used in training mode to reduce the number of involved multiplication and additions. Fig. 6.5 depicts that there is a significant improvement in convergence and MSE when using DTA as compared to LMS. The percentage of active taps for DTA is 15.9 percent. l_0-EFWLMS is also plotted as this algorithm achieves the lowest MSE than previous sparse algorithms in the literature [34,70,71]. Moreover, The performance is consistent when using DFE.

Fig. 6.6 depicts the MSE versus SNR and percentage of active taps at each SNR. It can be seen that at low SNR, large number of taps are useless. This not only saves on the computation but gives better performance.

In case of MIMO, the performance gain is even more produced as depicted in Fig. 6.7. Figure also shows that using only location information in DTA also gives improved performance. Next, in Fig. 6.8 the effect of block size L is investigated

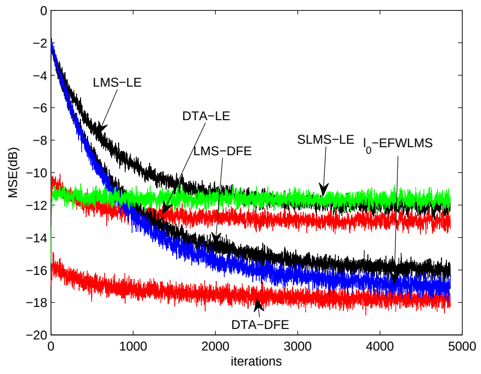


Figure 6.5: Comparison of LMS and SLMS for SISO case with $L = 200$, SNR=20 dB.

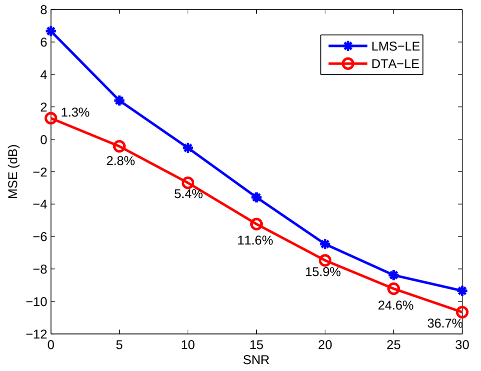


Figure 6.6: MSE versus SNR for SISO case with $L = 200$.

and it has been noted that using large block size ($L = 200$ in this case) only $N_f + L$ training signals are required.

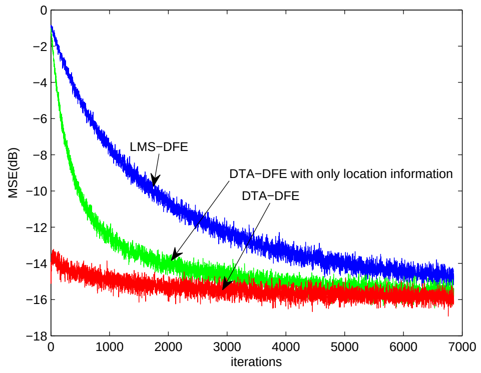


Figure 6.7: Comparison of LMS and SLMS for MIMO case with $L = 200$, SNR=20 dB.

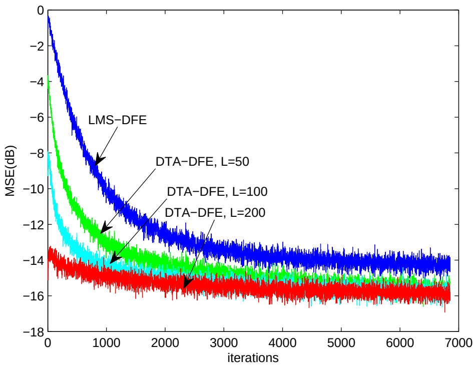


Figure 6.8: Effect of block size (MIMO case), SNR=20 dB.

Next, it is shown theoretically that how large number of taps effect different parameters. In table 6.1, theoretical values of the smallest eigenvalue, largest eigenvalue and eigenvalue spread of the correlation matrix ($\boldsymbol{R}$) at the input of the equalizer, are listed for different channels. The result of this computations are shown in Fig. 6.9 and 6.10. The theoretical MSE shown in Fig. 6.9 is calculated as [5]

$$MSE = \frac{\mu * \ noise\ variance * \ trace(\boldsymbol{R})}{2 - mu * trace(\boldsymbol{R})} + noise\ variance \tag{6.19}$$

and the time constant τ (which gives idea about the rate of convergence) is shown in the Fig. 6.10 , and it is given as [5]

$$\tau_i \approx \frac{-1}{2\ ln(1 - \mu * i^{th}\ eigenvalue)} \tag{6.20}$$

The slower rate of convergence is attained corresponding to the smallest eigenvalue. Therefore, for the Fig. 6.10, the smallest eigenvalue is used to compute the upper bound.

From these figures, it can be seen that large eigenvalue spread has the effect of slowing down the rate of convergence of the adaptive equalizer and also increasing the steady-state value of the average squared error. It is clear from these figures that with small number of active taps, the performance degradation and convergence are almost independent of eigenvalue spread. For remaining of the figures ITU vehicular A channel is used.

The misadjustment $\mathcal{M}$ provides a measure of how close the LMS algorithm is to optimality in the MSE sense and it is given as [29]

$$\mathcal{M} = \frac{\mu}{2} * trace(\boldsymbol{R}) \tag{6.21}$$

Figure 6.11 depicts that the misadjustment increases with the number of active taps. Fig. 6.12 gives the MSE dependency upon μ along with the active taps percentage.

Table 6.1: Delay Spread of various channels for 100% active taps

Channel	Min. eigenvalue	Max. eigenvalue	Eigenvalue spread
Uniform PDP with 11 taps	0.0132	30.1183	2277.9
Exponential PDP with 14 taps	0.0823	0.3485	4.2332
ITu vehicular A channel	0.0218	8.0400	368.8369

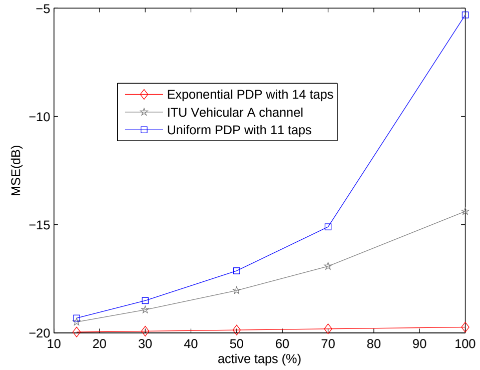


Figure 6.9: MSE versus active tap percentage for various channels, SNR=20 dB.

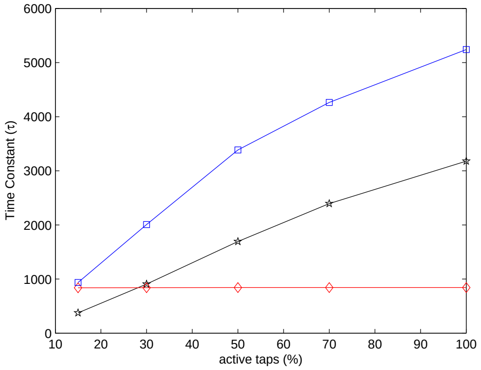


Figure 6.10: Time constant τ versus active tap percentage, SNR=20 dB.

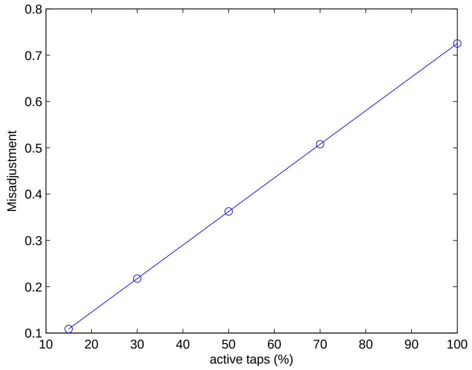


Figure 6.11: Theatrical maladjustment, SNR=20 dB.

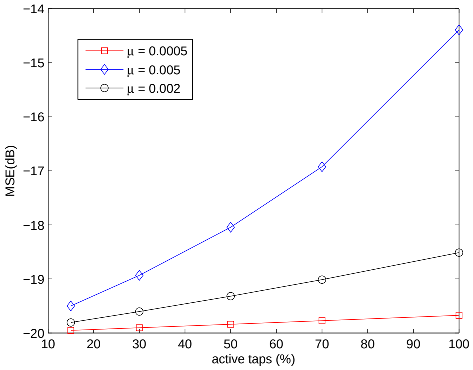


Figure 6.12: MSE versus step size μ, SNR=20 dB.

CHAPTER 7

CONCLUSIONS AND FUTURE RECOMMENDATIONS

7.1 Conclusion

In this dissertation several adaptive algorithms are proposed, in particular to the frequency domain DFE. These algorithms are studied in detail and analysed in various scenarios using either mathematical model or simulations. The main contributions of this dissertation are listed below:

- An adaptive frequency-domain DFE is proposed for the first time for SC-FDMA, SFBC SC-FDMA, SM SC-FDMA and hybrid SM-SFBC SFBC SC-FDMA systems with both feedforward and feedback filters operating in the frequency-domain. The equalizer operates without channel estimation at the receiver. The updating scheme used for the frequency-domain DFE is the RLS algorithm. The proposed algorithm is shown to have a low complexity and this is due to the special structure of the matrices involved in computing the weights of the feedforward and feedback filters in the frequency-domain. The AFD-DFE is also more computationally efficient than non-adaptive frequency-domain DFE. Simulation results for a time varying frequency-selective fading channel under the effect of high Doppler frequency and CFO on the system's performance are conducted and demonstrate the significant performance gain and robustness of the proposed algorithm.

- A constraint RLS based adaptive DFE is entirely designed in the frequency-domain for SC-FDMA and extended to a SFBC SCFDMA systems. The CRLS performs better than the RLS algorithm with almost similar complexity. Moreover, CRLS AFD-DFE outperforms the practical MMSE-DFE in terms of BER. We also showed that less training symbols can be used during training phase to reduce the overhead without sacrificing the performance and introduced a blind AFD-DFE is also introduced. Simulation results demonstrate the significant performance gain and robustness of the proposed algorithm under the severe Doppler effect. We also extend our design to 3-tap adaptive equalizer in the frequency-domain, which has better performance than 1-tap equalizer when dealing with ICI due to CFO.

- We showed how to integrate PSO with the DFE structure and demonstrated the superior performance of the PSO-DFE with respect to the widely-used LMS/RLS-DFE. This improvement is more pronounced for highly-dispersive channels with large eigenvalue spread and MIMO channels. Moveover, extensive simulations were conducted to optimize the PSO design parameters. The PSO-DFE significantly outperforms the PSO-LE and the LMS/RLS-DFE in terms of convergence time and steady-state MSE. A hybrid algorithm, PSO-LMS-DFE, is also proposed which is not only superior to the PSO-DFE in terms of performance but also enjoys lower complexity. Further complexity reduction is achieved using the FFT in the Fast PSO-LMS-DFE.

- A FD-PSO-based adaptive equalization in a SC-FDMA system is proposed. Simulation results verified that our devised adaptive equalization scheme has a better performance than that of RLS and LMS. Furthermore, in case of a high Doppler, it is shown that the performance of the FD-PSO algorithm can be improved by increasing the number of particles and the use of re-randomization.
- Sparse LMS algorithm is developed and shown to have fast convergence and lower MSE as compared to the conventional LMS algorithm.

7.2 Future recommendations

Future work involves the development of substitute algorithms to the proposed ones, e.g., the blind PSO can be combined with other existing blind algorithms like Constant-Modulus Algorithm (CMA), Reduced-Constellation Algorithm (RCA), Stop-and-go and Multi-Modulus Algorithm (MMA), to develop hybrid blind PSO algorithms. Similarly, BAFD-DFE uses only stop-and-go algorithm, above mentioned algorithms can be a good candidate to improve performance. Adaptive turbo DFE equalization in the frequency domain can also be an alternative equalization technique.

REFERENCES

[1] C. Zhang, Z. Wang, Z. Yang, J. Wang, and J. Song, “Frequency Domain Decision Feedback Equalization for Uplink SC-FDMA,” *IEEE Transactions on Broadcasting*, vol. 56, pp. 253–257, June 2010.

[2] O. Tanrikulu, B. Baykal, A. G. Constantinides, and J. A. Chambers, “SOFT CONSTRAINT SATISFACTION (SCS) BLIND CHANNEL EQUALIZATION ALGORITHMS,” *Int. J. Adaptive Control Signal Process*, vol. 12, pp. 117–134, 1998.

[3] S. U. H. Qureshi and S. Member, “Adaptive Equalization,” *Proc. IEEE*, vol. 73, no. 9, pp. 1349–1387, 1985.

[4] T. J. Endres, B. D. Anderson, C. R. Johnson, and L. Tong, “On the Robustness of FIR Channel Identification from Fractionally Spaced Received Signal Second-Order Statistics,” *IEEE Signal Process. Letters*, vol. 3, no. 5, pp. 153–155, 1996.

[5] A. H. Sayed, *Fundamentals of Adaptive Filtering.* New york: Wiley, 2003.

[6] 3GPP, "Physical Channels and Modulation," *ETSI TS 136.211 V10.0.0*, 2011.

[7] European Telecommunications Standards Institute (ETSI), "Digital video broadcating (DVB); Interaction channel for Digital Terrestrial Television (RCT) Incorporating Multiple Access OFDMA," *ETSI EN301 958*, vol. 1.1.1, 2002.

[8] D. Prendergast, B. Caron, and Y. Wu, "The Implementation of a Return Channel for ATSC-DTV," *IEEE Transactions on Broadcasting*, vol. 53, pp. 521–529, June 2007.

[9] IEEE Std 802.11-2007, "IEEE Standard for Information technology-Local and metropolitan area networks–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2007.

[10] IEEE Std 802.16-2009, "IEEE Standard for Local and metropolitan area networks–Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems," 2009.

[11] Q. Wang, C. Yuan, J. Zhang, and Y. Li, "Frequency Domain Soft-Decision Feedback Equalization for SC-FDMA with Insufficient Cyclic Prefix," *International Journal of Computer Science (IJCSI)*, vol. 9, no. 6, pp. 103–108, 2012.

[12] K. Raghunath and A. Chockalingam, "SC-FDMA Versus OFDMA: Sensitivity to Large Carrier Frequency and Timing Offsets on the Uplink," *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6, Nov. 2009.

[13] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.

[14] V. Tarokh, H. Jafarkhani, and A. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Transactions on Information Theroy*, vol. 45, pp. 1456–1467, July 1999.

[15] K. Lee and D. Williams, "A space-frequency transmitter diversity technique for OFDM systems," *IEEE. Global Telecommunications Conference (Globecom)*, vol. 3, pp. 1473–1477, 2000.

[16] C. Ciochina, D. Castelain, D. Mottier, and H. Sari, "A Novel Space-Frequency Coding Scheme for Single Carrier Modulations," *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–5, Sept. 2007.

[17] Alcatel Shanghai Bell Alcatel-Lucent, "STBC-II scheme with non-paired symbols for LTE-Advanced uplink transmit diversity," in *R1-090058, 3GPP TSG RAN WG 1 Meeting #55 bis*, 2008.

[18] W. Y. Lim and Z. Lei, "Space-time block code design for single-carrier frequency division multiple access," *IEEE 20th International Symposium on*

Personal, Indoor and Mobile Radio Communications, pp. 516–520, Sept. 2009.

[19] D. Falconer, S. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *IEEE Communications Magazine*, vol. 40, pp. 58–66, Apr. 2002.

[20] N. Benvenuto and S. Tomasin, "On the comparison between OFDM and single carrier modulation with a DFE using a frequency-domain feedforward filter," *IEEE Transactions on Communications*, vol. 50, pp. 947–955, June 2002.

[21] H. Witschnig, M. Kemptner, R. Weigel, and A. Springer, "Decision feedback equalization for a single carrier system with frequency domain equalization - an overall system approach," *1st International Symposium onWireless Communication Systems*, pp. 26–30, 2004.

[22] G. Huang, A. Nix, and S. Armour, "Decision feedback equalization in SC-FDMA," *19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–5, Sept. 2008.

[23] N. Benvenuto and S. Tomasin, "Iterative Design and Detection of a DFE in the Frequency Domain," *IEEE Transactions on Communications*, vol. 53, pp. 1867–1875, Nov. 2005.

[24] W. Younis, A. Sayed, and N. Al-Dhahir, "Efficient adaptive receivers for joint equalization and interference cancellation in multiuser space-time block-

coded systems," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2849–2862, Nov. 2003.

[25] K. Berberidis and P. Karaivazoglou, "An efficient block adaptive decision feedback equalizer implemented in the frequency domain," *IEEE Transactions on Signal Processing*, vol. 50, pp. 2273–2285, Sept. 2002.

[26] V. Kekatos, K. Berberidis, and A. A. Rontogiannis, "A Block Adaptive Frequency Domain MIMO DFE for Wideband Channels," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. III–197–III–200, 2007.

[27] M. A. Ruder, U. L. Dang, and W. H. Gerstacker, "User Pairing for Multiuser SC-FDMA Transmission over Virtual MIMO ISI Channels," *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–7, Nov. 2009.

[28] T. Kailath, *Linear Systems.* Englewood Cliffs,NJ:Prentice Hall, 1980.

[29] S. Haykin, *Adaptive Filter Theory.* Prentice Hall,Upper-Saddle River, NJ, 4th ed., 2002.

[30] B. Dhivagar, K. Kuchi, and K. Giridhar, "An iterative MIMO-DFE receiver with MLD for uplink SC-FDMA," *National Conference on Communications (NCC)*, pp. 1–4, Feb. 2013.

[31] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Johns Hopkins University Press, 4th ed., 2013.

[32] D. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms," *IEEE Transactions on Signal Processing*, vol. 41, no. 9, pp. 2811–2825, 1993.

[33] S. Gelfand and J. Krogmeier, "Noise-constrained least mean squares algorithm," *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 1961–1970, 2001.

[34] Y. Gu, J. Jin, and S. Mei, "lo Norm Constraint LMS Algorithm for Sparse System Identification," *IEEE Signal Processing Letters*, vol. 16, pp. 774–777, Sept. 2009.

[35] M. O. Bin Saeed, A. Zerguine, and S. A. Zummo, "A noise-constrained algorithm for estimation over distributed networks," *International Journal of Adaptive Control and Signal Processing*, pp. 827–845, Oct. 2012.

[36] "Technical specification group radio access Network," *LTE-A transmit diversity schemes for PUCCH format 1/1a/b, RI-092340*, 2009.

[37] B. Narasimhan, N. Al-Dhahir, and H. Minn, "SFBC design tradeoffs for mobile SC-FDMA with application to LTE-advanced," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3458–3461, 2010.

[38] G. Picchi and G. Prati, "Blind Equalization and Carrier Recovery Using a "Stop-and-Go" Decision-Directed Algorithm," *IEEE Transactions on Communications*, vol. 35, pp. 877–887, Sept. 1987.

[39] N. Al-Dhahir and J. Cioffi, "MMSE decision-feedback equalizers: finite-length results," *IEEE Transactions on Information Theory*, vol. 41, pp. 961–975, July 1995.

[40] D. J. Krusienski and W. K. Jenkins, "The application of particle swarm optimization to adaptive IIR phase equalization," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 693–696, 2004.

[41] H. H. E. Morra, A. U. Sheikh, and A. Zerguine, "Application of Heuristic Algorithms for Multiuser Detection," *International Conference on Communications, ICC-2006*, pp. 3157–3161, 2006.

[42] H. Liu and J. Li, "A Particle Swarm Optimization-Based Multiuser Detection for Receive-Diversity-Aided STBC Systems," *IEEE Signal Processing Letters*, vol. 15, no. 3, pp. 29–32, 2008.

[43] A. T. Al-Awami, A. Zerguine, L. Cheded, A. Zidouri, and W. Saif, "A new modified particle swarm optimization algorithm for adaptive equalization," *Digital Signal Processing*, vol. 21, pp. 195–207, Mar. 2011.

[44] J. Kennedy and R. Eberhart, *Swarm intelligence.* Academic Press, Jan. 2001.

[45] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Intl. Conf. Neural Networks*, pp. 1942–1948, 1995.

[46] T. Ji, Z. Lu, and Q. Wu, "Optimal soft morphological filter for periodic noise removal using a particle swarm optimiser with passive congregation," *Signal Processing*, vol. 87, pp. 2799–2809, Nov. 2007.

[47] C. Potter, G. K. Venayagamoorthy, and K. Kosbar, "RNN based MIMO channel prediction," *Signal Processing*, vol. 90, pp. 440–450, Feb. 2010.

[48] N. V. George and G. Panda, "Advances in active noise control: A survey, with emphasis on recent nonlinear techniques," *Signal Processing*, vol. 93, pp. 363–377, Feb. 2013.

[49] Y. Chi, F. Sun, L. Jiang, C. Yu, and P. Zhang, "Elastic Boundary for Particle Swarm Optimization," *International Conference on Swarm In telligence* ,, pp. 125–132, 2012.

[50] G. Foschini, G. Golden, R. Valenzuela, and P. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 11, pp. 1841–1852, 1999.

[51] B. Hassibi, "An efficient square-root algorithm for BLAST," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. II737–II740, 2000.

[52] S. Blostein, "Modified decorrelating decision-feedback detection of BLAST space-time system," *International Conference on Communications (ICC)*, pp. 335–339, 2002.

[53] H. Zhu, Z. Lei, and F. Chin, "An Improved Square-Root Algorithm for BLAST," *IEEE Signal Processing Letters*, vol. 11, pp. 772–775, Sept. 2004.

[54] G. Ginis and J. Cioffi, "On the relation between V-BLAST and the GDFE," *IEEE Communications Letters*, vol. 5, pp. 364–366, Sept. 2001.

[55] J. Choi, H. Yu, and Y. Lee, "Adaptive MIMO decision feedback equalization for receivers with time-varying channels," *IEEE Transactions on Signal Processing*, vol. 53, pp. 4295–4303, Nov. 2005.

[56] A. Rontogiannis, V. Kekatos, and K. Berberidis, "A Square-Root Adaptive V-BLAST Algorithm for Fast Time-Varying MIMO Channels," *International Conference on Communications (ICC)*, pp. 3135–3139, 2006.

[57] P. Li and R. Murch, "Multiple output selection-LAS algorithm in large MIMO systems," *IEEE Communications Letters*, vol. 14, pp. 399–401, May 2010.

[58] V. Tripathi and M. Honig, "Adaptive turbo reduced-rank equalization for MIMO channels," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 2789–2800, Nov. 2005.

[59] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank Processing Based on Joint and Iterative Interpolation, Decimation, and Filtering," *IEEE Transactions on Signal Processing*, vol. 57, pp. 2503–2514, July 2009.

[60] R. C. de Lamare and R. Sampaio-Neto, "Adaptive Reduced-Rank Equalization Algorithms Based on Alternating Optimization Design Techniques for MIMO Systems," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 2482–2494, July 2011.

[61] Y. Shi and R. Eberhart, “A Modified Particle Swarm Optimizer,” *Int’l conf. on Evolutionary Computation*, pp. 69–73, 1998.

[62] M. Clerc, “The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization,” *Congress on Evolutionary Computation*, pp. 1951–1957, 1999.

[63] R. Eberhart and Y. Shi, “Comparing inertia weights and constriction factors in particle swarm optimization,” *Congress on Evolutionary Computation*, pp. 84–88, 2000.

[64] D. Krusienski and W. Jenkins, “A Modified Particle Swarm Optimization Algorithm for Adaptive Filtering,” *IEEE International Symposium on Circuits and Systems*, pp. 137–140, 2006.

[65] D. Krusienski and W. Jenkins, “A particle swarm optimization - least mean squares algorithm for adaptive filtering,” *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, pp. 241–245, 2004.

[66] J. Yang, J.-J. Werner, and G. Dumont, “The multimodulus blind equalization and its generalized algorithms,” *IEEE Journal on Selected Areas in Communications*, vol. 20, pp. 997–1015, June 2002.

[67] J.-T. Yuan and K.-D. Tsai, “Analysis of the Multimodulus Blind Equalization Algorithm in QAM Communication Systems,” *IEEE Transactions on Communications*, vol. 53, pp. 1427–1431, Sept. 2005.

[68] J.-T. Yuan and T.-C. Lin, “Equalization and Carrier Phase Recovery of CMA and MMA in Blind Adaptive Receivers,” *IEEE Transactions on Signal Processing*, vol. 58, pp. 3206–3217, June 2010.

[69] W. Xue, X. Yang, and Z. Zhang, “A Stop and Go Blind Equalization Algorithm for QAM Signals,” *International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, Sept. 2010.

[70] Y. Chen, Y. Gu, and A. O. Hero, “Sparse LMS for system identification,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3125–3128, Apr. 2009.

[71] J. Jin, Y. Gu, and S. Mei, “A Stochastic Gradient Approach on Compressive Sensing Signal Reconstruction Based on Adaptive Filtering Framework,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 409–420, Apr. 2010.

[72] M. O. Bin Saeed and A. Zerguine, “A variable step size strategy for sparse system identification,” *10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13)*, pp. 1–4, Mar. 2013.

[73] S. Mallat, G. Davis, and Z. Zhang, “Adaptive Time-Frequency Decompositions,” *SPIE J. Optical Eng.*, vol. 33, pp. 2183–2191, 1994.

[74] “Guidelines for The evaluation of radio transmission technologies for IMT-2000,” *Recommendation ITU-R M.1225*, 1997.

[75] J. A. Tropp and A. C. Gilbert, “Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit,” *IEEE Transactions on Information Theory*, vol. 53, pp. 4655–4666, Dec. 2007.

[76] P. Dymarski, N. Moreau, and G. Richard, “Greedy sparse decompositions: a comparative study,” *EURASIP Journal on Advances in Signal Processing*, p. 34, 2011.

Vitae

- Name: Naveed Iqbal
- Nationality: Pakistani
- Date of Birth: Jan 24, 1983
- Present Address: P.O. Box 8652, KFUPM, Dhahran 31261, Saudi Arabia.
- Permanent Address: House No 12/1, Street 1, LANE 7, Opposite westridge, Near Choor Chowk, Rawalpindi, Pakistan.
- Telephone: (+966) 54 256 3510
- Email: *mohammadnaveediqbal@yahoo.com*
- M.Sc. Electrical Engineering: September 2009
- B.Sc. Electrical Engineering: June 2006

List of Publications

- **Naveed Iqbal**, Naofal Al-Dhahir, Azzedine Zerguine, Abdelmalek Zidouri, "Adaptive equalization in SCFDMA uplink", accepted in IEEE Transactions on Vehicular Technology, March 2014.

- **Naveed Iqbal**,Azzedine Zerguine , Naofal Al-Dhahir, "Adaptive equalization using PSO in SCFDMA uplink", IET Electronic Letters, Mar. 2014.

- **Naveed Iqbal**,Azzedine Zerguine, "Adaptive equalization using PSO in LTE uplink", in proceedings of Saudi student conference, Mar. 2014.

- **Naveed Iqbal**, Azzedine Zerguine, Naofal Al-Dhahir, "Decision feedback equalization using Particle swarm optimization", Accepted in Signal Processing journal ELSEVIER, Feb. 2014

- **Naveed Iqbal**, Naofal Al-Dhahir,Azzedine Zerguine, "Constraint Adaptive equalization in SCFDMA uplink", Submitted to IEEE Transactions on Communication, 2014.

- **Naveed Iqbal**, Naofal Al-Dhahir,Azzedine Zerguine, "Adaptive equalization using Sparse LMS", Under preparation.