

CAPITAL UNIVERSITY OF SCIENCE AND
TECHNOLOGY, ISLAMABAD



Applications of Fractional Derivatives in Adaptive Signal Processing Systems

by

Syed Muslim Shah

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the

Faculty of Engineering

Department of Electrical Engineering

2019

Applications of Fractional Derivatives in Adaptive Signal Processing Systems

By

Syed Muslim Shah

(PE111003)

Dr. Manuel Duarte Ortigueira, Professor,
Universidade Nova de Lisboa, Quinta da Torre, Portugal

Dr. Ismail H. Tuncer, Professor,
Middle East Technical University, Ankara, Turkey

Dr. Raza Samar
(Thesis Supervisor)

Dr. Noor Muhammad Khan
(Head, Department of Electrical Engineering)

Dr. Imtiaz Ahmed Taj
(Dean, Faculty of Engineering)

DEPARTMENT OF ELECTRICAL ENGINEERING
CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
ISLAMABAD

2019

Copyright © 2019 by Syed Muslim Shah

All rights reserved. No part of this thesis may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, by any information storage and retrieval system without the prior written permission of the author.

**Dedicated to My Parents, Family, My Grandfather Ganjle Baba, Orya
Maqbool Jan, Hafiz Ghulam Haider Perdesi and to all those who
serves the humanity with love, patience and unconditionally.**



**CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
ISLAMABAD**

Expressway, Kahuta Road, Zone-V, Islamabad
Phone: +92-51-111-555-666 Fax: +92-51-4486705
Email: info@cust.edu.pk Website: <https://www.cust.edu.pk>

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in the thesis, entitled “**Applications of Fractional Derivatives in Adaptive Signal Processing Systems**” was conducted under the supervision of **Dr. Raza Samar**. No part of this thesis has been submitted anywhere else for any other degree. This thesis is submitted to the **Department of Electrical Engineering, Capital University of Science and Technology** in partial fulfillment of the requirements for the degree of Doctor in Philosophy in the field of **Electrical Engineering**. The open defence of the thesis was conducted on **February 13, 2019**.

Student Name: Mr. Syed Muslim Shah (PE111003)

The Examining Committee unanimously agrees to award PhD degree in the mentioned field.

Examination Committee :

(a) External Examiner 1: Dr. Muhammad Tufail,
Professor
PIEAS, Islamabad

(b) External Examiner 2: Dr. Aneela Zameer Jaafery,
Professor
PIEAS, Islamabad

(c) Internal Examiner : Dr. Fazal ur Rehman
Professor
CUST, Islamabad

Supervisor Name : Dr. Raza Samar
Professor
CUST, Islamabad

Name of HoD : Dr. Noor Muhammad Khan
Professor
CUST, Islamabad

Name of Dean : Dr. Imtiaz Ahmad Taj
Professor
CUST, Islamabad

AUTHOR'S DECLARATION

I, **Mr. Syed Muslim Shah (Registration No. PE-113003)**, hereby state that my PhD thesis titled, '**Applications of Fractional Derivatives in Adaptive Signal Processing Systems**' is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/ world.

At any time, if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my PhD Degree.



(**Mr. Syed Muslim Shah**)

Dated: 13 February, 2019

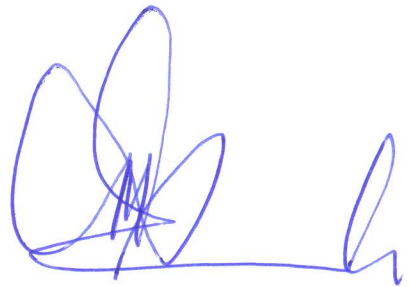
Registration No : PE111003

PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled “**Applications of Fractional Derivatives in Adaptive Signal Processing Systems**” is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/ cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD Degree, the University reserves the right to withdraw/ revoke my PhD degree and that HEC and the University have the right to publish my name on the HEC/ University Website on which names of students are placed who submitted plagiarized thesis.



(Mr. Syed Muslim Shah)
Registration No. PE111003

Dated: 13 February, 2019

List of Publications

It is certified that following publication(s) have been made out of the research work that has been carried out for this thesis:-

Journal Publications

1. **S. M. Shah**, R. Samar, R., Noor. M. Khan, N.M. and M.A.Z Raja, "Design of fractional-order variants of complex LMS and NLMS algorithms for adaptive channel equalization", *Nonlinear Dynamics*, pp.1-20, 2017.
2. **S. M. Shah**, R. Samar, R., Noor. M. Khan, N.M. and M.A.Z Raja, "Fractional-order adaptive signal processing strategies for active noise control systems", *Nonlinear Dynamics*, 85(3), pp.1363-1376, 2016.
3. **S. M. Shah**, "Riemann-Liouville operator-based fractional normalised least mean square algorithm with application to decision feedback equalisation of multipath channels", *IET Signal Processing*, 10(6), pp. 575-582, 2016.
4. **S. M. Shah**, R. Samar, S. M. Naqvi, and J. A. Chambers, "Fractional order constant modulus blind algorithms with application to channel equalisation", *Electronics Letters*, 50(23), pp.1702-1704, 2014.
5. **S. M. Shah**, R. Samar, M.A.Z Raja and J. A. Chambers, "Fractional Normalized Filtered-error Least Mean Squares Algorithm for Applications in Active Noise Control Systems", *Electronics Letters*, vol. (14), pp. 973 - 975, 2014.
6. **S. M. Shah**, R. Samar and M.A.Z Raja, Fractional order algorithms for tracking Rayleigh fading channels, *Nonlinear Dynamics*, pp. 1-17, 2018.

Syed Muslim Shah

(PE111003)

Acknowledgements

I would like to give my great thanks to my research supervisors including Dr. Raza Samar, Dr. Noor Muhammad Khan and Professor Jonathan A. Chambers of New Castle University, UK; without the sincere guidance, encouragement and valuable support of these honorable Professors, this thesis would have not been possible. I am thankful to these Professors to create a positive difference in my technical and academic career through their personnel consultations. I am thankful to the foreign evaluators including Professor Manuel D. Ortigueira, UNINOVA and Faculdade de Ciências e Tecnologia da UNL, Portugal, Dr. Ismail H. Tuncer, Professor, Middle East Technical University, Ankara, Turkey and Professor Matthew C. Turner, University of Leicester, UK. The valuable suggestions and comments of these Professors have greatly improved the presentation and technical strength of the research thesis. I acknowledge the external examiners including Professor Dr. M Tufail and Professor Dr. Aneela Zameer Jaffery of PIEAS, Islamabad for their useful suggestions and recommendations of the thesis for the award of PhD. I also acknowledge the suggestions of the faculty members at the Department of Electrical Engineering including Prof. Aamir Iqbal Bhatti, Professor Fazal-ur-Rehman, Dean/Professor Imtiaz A. Taj and Mr. Khalid Mehmood of the Graduate office. Dr. M. Asif Zahoor Raja of Comsats Institute of IT, Attock is also acknowledged who always motivated me throughout the PhD research processes. I acknowledge the Higher Education Commission, Islamabad Pakistan for sponsoring my research studies. Finally, I acknowledge my mother Hidayat Bibi and father Zahir Syed Badshah for their prayers, mother-in-law Firasat Bibi, father-in-law Dr. Rooh ul Amin, wife Ambreen Shah, daughter Anshrah Syed and sons Syed Ali Hassan Shah, Syed Moiz Hassan Shah and S. M. Hasnain Haider for the sacrifices of their time in which they deserved most of my attentions.

Abstract

Adaptive algorithms are mostly optimized using integer order derivatives for error minimization. The Least Mean Squares (LMS) and Recursive Least Squares (RLS) adaptive filters are among the most commonly employed schemes. The LMS algorithm is simple to implement, has robust tracking performance in nonstationary environments and is less sensitive to floating point precision effects. However, it has the issue of slow convergence especially when the number of weights is large. The RLS achieves faster convergence but is computationally expensive, and problematic in nonstationary environments.

This study introduces fractional calculus techniques in stochastic gradient algorithms. In addition to first order derivative, Fractional Order (FO) derivatives are proposed in the optimization of gradient algorithms. Four configurations have been considered based on whether the fractional derivative is applied to the instantaneous present or posterior error. For evaluation of the FO algorithms, three applications have been considered, that is, (a) adaptive equalization of multipath channels (b) Active Noise Control Systems (ANCS) and (c) tracking of time varying Rayleigh fading sequences. In equalization, both supervised and unsupervised algorithms are considered. For the supervised case, FO variants of LMS and Normalized LMS (NLMS) are applied in both feed-forward and decision feedback configurations. In the unsupervised case, FO variants of Gordan and constant modulus algorithms are developed. In ANCS, FO variants of the NLMS, Filtered-x (input) LMS, Modified FxLMS and Filtered-error LMS algorithms are developed. The noises are modelled as binary, Gaussian and impulsive sources characterized by fractional lower order moments. In tracking, the behavior of FO variants is evaluated for nonstationary environments. A Rayleigh channel has also been considered having Doppler frequency shifts of 0.8KHz to 3KHz. The fractional algorithms are compared with the standard NLMS, RLS and Extended-RLS schemes.

The main performance metrics include (1) mean squared error (2) mean squared deviation (3) relative modelling error (4) model accuracy using both frequency and time domain analysis and (5) symbol error rate. The former three performance

metrics help compare the convergence speed and steady state performance; the latter two are application specific. Simulation results are shown for different step sizes and fractional orders. It is seen that the fractional variants show superior performance in all the three applications and hold great promise for future use.

Contents

Author's Declaration	v
Plagiarism Undertaking	vi
List of Publications	vii
Acknowledgements	viii
Abstract	ix
List of Figures	xiv
List of Tables	xvii
Abbreviations	xviii
Symbols	xx
1 Introduction	1
1.1 Adaptive Filtering Systems	1
1.2 Steepest Descent Algorithm	5
1.3 Least Mean Square Algorithm	6
1.4 Literature Review	6
1.5 Significance of the Research	9
1.6 Research Aims	10
1.7 Contributions of the Thesis	11
1.7.1 Multipath Channel Equalization	11
1.7.2 Active Noise Control Systems	12
1.7.3 Tracking of Rayleigh Fading Sequences	12
1.8 Thesis Organization	13
2 Adaptive Signal Processing Algorithms and Applications	15
2.1 Adaptive Equalization	15
2.1.1 Optimal Filtering	18
2.1.2 The Steepest Descent Algorithm	19

2.1.3	The LMS Algorithm	20
2.1.4	The NLMS Algorithm	20
2.1.5	Blind Algorithms	21
2.2	Algorithms for Active Noise Control Systems	22
2.2.1	Optimal ANCS Controller	24
2.2.2	Gradient Based FxLMS Algorithm and its Variants	25
2.3	Tracking of Rayleigh Fading Channels	28
2.3.1	The RLS Algorithm	29
2.3.2	Rayleigh Fading Channel	31
3	Fractional Order Derivatives and Adaptive Signal Processing	34
3.1	Introduction to Fractional Derivatives	34
3.2	The Riemann-Liouville Fractional Derivatives	35
3.3	Fractional Least Mean Square Algorithm	37
3.4	Summary	39
4	Fractional Normalized Least Mean Square Algorithm for Adaptive Equalization of Multipath Channels	40
4.1	Feedforward and Decision Feedback Equalization	41
4.2	Proposed FNLMS Algorithm	46
4.3	Simulation Results	51
4.3.1	NLMS vs FNLMS Method 1	54
4.3.2	LMS, FLMS, NLMS and FNLMS Method 2	61
4.4	Conclusions	68
5	Fractional Order Constant Modulus Blind Algorithms with Application to Channel Equalization	69
5.1	Fractional Order Blind Algorithms	70
5.2	Simulation Results	75
5.3	Conclusions	78
6	Fractional Order Adaptive Signal Processing Strategies for Active Noise Control Systems	79
6.1	The FN-FeLMS Algorithm	80
6.2	Modified Fractional FeLMS and its Variants	83
6.3	Computational Complexity	88
6.4	Simulation Results and Analysis	89
6.4.1	Simulation Results for FN-FeLMS Algorithm	92
6.4.2	Simulation Results for Modified FO Algorithms	96
6.4.3	Simulation Results for Impulsive Noises	103
6.5	Conclusions	107
7	Tracking of Rayleigh Fading Channels	109
7.1	Introduction	110
7.2	System Model	111

7.3	The Recursive Least Squares Algorithm	113
7.4	Fractional Least Mean Square Algorithm and Its variants	116
7.5	Simulation Results and Analysis	118
7.6	Conclusions	127
8	Summary and Future Work	130
8.1	Summary and Conclusions	130
8.2	Future Work Directions	132
	Bibliography	135

List of Figures

1.1	Block diagram of adaptive filtering system.	2
2.1	Inter-Symbol Interference in a multipath environment.	16
2.2	FIR tapped delay line modelling a multipath channel.	16
2.3	Functional Schematic diagram of the equalization process.	17
2.4	Active noise control system using Feedforward control.	22
2.5	Functional Schematic of Adaptive ANC System.	23
2.6	Schematic of FxLMS Based ANC System.	26
2.7	Linear combiner.	29
2.8	Amplitude and Phase plots of Rayleigh fading with Doppler Frequency of 100 Hz.	32
3.1	The Characteristic curve of the Gamma function.	36
4.1	Communication System with Decision Feedback Equalizer.	42
4.2	Filter structure of FNLMS based DFE.	46
4.3	Symbol-error rate for QAM constellations equalized with NLMS, F-NLMS.	55
4.4	Scatter Plots of 64 and 256 QAM constellations equalized with NLMS, F-NLMS.	56
4.5	SER for 4, 16, 64 and 256 QAM constellations for the NLMS and FNLMS.	57
4.6	Representation of Channel impulse response (right), amplitude response (top left) in dBs vs frequency and Time domain (bottom left) response of the channel (right) with 17-taps.	58
4.7	Combined channel and equalizer frequency response.	61
4.8	Scatter Plots for LMS, F-LMS, with Different Training Symbols.	63
4.9	Scatter Plots with different training symbols and Responses of NLMS, F-NLMS.	63
4.10	Scatter Plots for LMS, FLMS with [100,200,300] Training Symbols.	65
4.11	MSE for 300 Training Symbols of LMS and FLMS.	65
4.12	Scatter Plots of 16 & 64 QAM constellations Equalized with NLMS, F-NLMS.	66
4.13	Scatter Plots of 64 & 256 QAM constellations Equalized with NLMS, F-NLMS.	67
5.1	Effect of varying the fractional order on the term $\frac{w^{(1-\nu)}}{\Gamma(2-\nu)}$	75

5.2	MSE Performance of CMA and FrCMA for channel 1.	76
5.3	MSE Performance of CMA and FrCMA for channel 2 for different step sizes and fractional orders.	77
6.1	Schematic of ANC system for machine noise suppression in a channel using FN-FeLMS in a feedforward configuration.	80
6.2	Normalized frequency responses of the different primary path models.	91
6.3	MSE learning curves for FeLMS & FN-FeLMS for different choices of μ_f and μ_l and for Gaussian noise, $\nu = 0.5$, SNR = 60 dB.	93
6.4	MSD learning curves for FeLMS & FN-FeLMS for different choices of μ_f and μ_l and for Gaussian input, $\nu = 0.1$, SNR = 60 dB.	94
6.5	MSD learning curves for FeLMS & FN-FeLMS for different choices of μ_f and μ_l and for binary Input data, $\nu = 0. - 0.25$	95
6.6	Frequency response of the primary path and secondary path models using FeLMS & FN-FeLMS methods, $\nu = -0.5$	96
6.7	FrFeLMS vs. FN-FeLMS MSD Learning curves for different step sizes.	97
6.8	FxLMS vs. FrFxLMS MSD Learning curves for different step sizes.	99
6.9	MFxLMS vs. FrMFxLMS MSD Learning curves for different step sizes.	99
6.10	MSE performance of NLMS, FxLMS, FeLMS, FNLMS, FrFxLMS and FrFeLMS for Gaussian noise.	100
6.11	FeLMS vs. FrFeLMS MSD Learning curves for different tap-lengths.	101
6.12	FxLMS vs. FrFxLMS MSD Learning curves for different tap-lengths.	102
6.13	MSD Learning performance of FrFeLMS for different fractional orders and Gaussian input noise.	103
6.14	Effects of changing the number of taps of the primary path at different iteration number.	104
6.15	Relative Modelling Error of FeLMS and FrFeLMS algorithms for $S\alpha S$ impulsive input noise.	105
6.16	Relative Modelling Error of FeLMS and FrFeLMS for asymmetric α -stable impulsive input noise.	105
6.17	Relative Modelling Error of FeLMS and FrFeLMS for asymmetric α -stable impulsive input noise.	106
6.18	Effects of fractional order on Gw^{1-v} , the single plot shows the behavior of G for different fractional orders.	107
7.1	Schematic of adaptive tracking problem.	112
7.2	MSE Learning curves of NLMS, F-NLMSv1, RLS and E-RLS for $f_D = 100$ Hz.	120
7.3	MSD Learning curves of NLMS, F-NLMSv1, RLS and E-RLS for $f_D = 100$ Hz.	121
7.4	MSE Learning curves of NLMS, F-NLMSv2, RLS and E-RLS for $f_D = 400$ Hz.	121
7.5	MSD Learning curves of NLMS, F-NLMSv2, RLS and E-RLS for $f_D = 400$ Hz.	122

7.6	MSE Learning curves of NLMS, F-NLMSv3, RLS and E-RLS for $f_D = 900$ Hz.	123
7.7	MSD Learning curves of NLMS, F-NLMSv3, RLS and E-RLS for $f_D = 900$ Hz.	123
7.8	MSE Learning curves of NLMS, F-NLMSv4, RLS and E-RLS for $f_D = 1300$ Hz.	124
7.9	MSD Learning curves of NLMS, F-NLMSv4, RLS and E-RLS for $f_D = 1300$ Hz.	124
7.10	Tracking behavior of different taps of the algorithms.	125
7.11	Tracking behavior of tap-3 for different algorithms.	126
7.12	MSE and MSD Learning curves of NLMS, F-NLMSv4, RLS and E-RLS.	126
7.13	MSE and MSD Learning curves of NLMS, F-NLMSv4, RLS and E-RLS.	127
7.14	MSE and MSD Learning curves of NLMS, F-NLMSv4, RLS and E-RLS.	128
7.15	Tracking behavior of different taps of the algorithms in the hybrid case.	128

List of Tables

4.1	Summary of DFE Algorithm for FNLMS (Method 1).	48
4.2	NLMS and FNLMS Algorithms for FF Equalization (Method 2). . .	52
4.3	DFE Algorithms Based on NLMS and FNLMS (Method 2).	53
4.4	SER of NLMS and FNLMS for different ν values	59
4.5	SER of NLMS and FNLMS algorithms for different μ_2 values	60
4.6	SER of NLMS and Fractional NLMS Algorithms	62
4.7	SER of NLMS and Fractional NLMS Algorithms	64
5.1	FrCMA Based Adaptation Procedure.	74
6.1	Summary of ANC Algorithm for fractional normalized FeLMS. . . .	83
6.2	FrFeLMS Based ANC.	85
6.3	FrNLMS Based ANC.	86
6.4	FrFxLMS Based ANC.	87
6.5	FrMFxLMS Based ANC.	87
6.6	Computational complexity of various algorithms.	88
6.7	Convergence and MSD for FN-FeLMS for $\mu_{l=0.1}, \gamma = 4$	95
6.8	MSD Convergence performance comparison of FrFeLMS vs. FN- FeLMS [67]	98
7.1	Summary of Fractional Normalized LMS v3.	117
7.2	Summary of F-NLMS v4..	118

Abbreviations

ANC	Active Noise Control
ANCS	Active Noise Control Systems
AFA	Affine Projection Algorithm
<i>AαS</i>	Asymmetric Alpha Stable
CDMA	Code Division Multiple Access
CIR	Channel Impulse Response
CMA	Constant Modulus Algorithm
DD	Decision Delay
DFE	Decision Feedback Equalization
E-RLS	Extended Recursive Least Squares
FF	Fractional Calculus
FD	Fractional Derivatives
FB	FeedBack
FDF	Frequency Domain Filters
FF	Feed-Forward
FeLMS	Filtered-error Least Mean Squares
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FLMS	Fractional Least Mean Square
FO	Fractional Order
FOPDT	Fractional Order Plus Dead-Time
FrCMA	Fractional Constant Modulus Algorithm
FrFeLMS	Fractional Filtered-error Least Mean Squares
FrFxLMS	Fractional Filtered-x Least Mean Squares

FrLMS	Fractional Least Mean Squares
FrMFxLMS	Fractional Modified Filtered-error Least Mean Squares
FrNLMS	Fractional Normalized Least Mean Squares
FxAP	Filtered-x Affine Projection
FxLMS	Filtered-x Least Mean Squares
FxRLS	Filtered-x Recursive Least Squares
IOD	Integer Order Derivatives
IS	Interim Standard
ISI	Inter-Symbol Interference
LMF	Least Mean Fourth
LMS	Least Mean Square
LTE	Long Term Evolution
MFxLMS	Modified Filtered-x Least mean Squares
MIMO	Multi-Input Multi-Output
MMA	Multi Modulus Algorithm
MMA	Multi Modulus Algorithm
MMSE	Minimum Mean Square Error
MSD	Mean Square Deviation
MSE	Mean Square Errors
NLMS	Normalized Least Mean Squares
PI	Proportional-Integral
PP	Primary Path
QAM	Quadrature Amplitude Modulation
RLS	Recursive Least Squares
RME	Relative Modelling Error
SB	Sub-Band
SD	Steepest Descent
SER	Symbol Error Rate
SP	Secondary Path
SαS	Symmetric Alpha Stable

Symbols

ν	Fractional order
n	Integer Order
\mathbf{R}	Auto Correlation Matrix
\mathbf{R}^{-1}	Inverse of matrix
$(.)^T$	Transpose Operation
$(.)^H$	Hermitian Transpose Operation
\mathbf{x}	Input Signal Vector
y	Output of Filter
d	Desired output
e	Error (desired output-filter output)
\mathbf{p}	Cross Correlation Vector
k	Sample number or time index
$\mathbf{x} \odot \mathbf{y}$	Element-wise multiplication between vectors \mathbf{x} and \mathbf{y}

Chapter 1

Introduction

Convergence time, tracking and computational cost are among the most important parameters in designing adaptive signal processing systems. Faster convergence speeds up the learning process and helps increase system efficiency in different applications. Fewer training symbols in equalization or channel estimation help increase efficiency of the transmission system by utilizing more time or bandwidth for useful data. Adaptive filters must have the capability to track time based variations in system parameters especially after the learning phase.

This chapter provides an introduction of adaptive signal processing systems and some applications, for which fractional order filtering algorithms will later be presented. Different metrics have been defined which help evaluate the performance of adaptive filters in different applications. Literature review is provided for different algorithms. Finally, significance, aims and contributions of the research study are highlighted.

1.1 Adaptive Filtering Systems

An adaptive filter progressively adjusts its transfer function according to an adaptation algorithm while monitoring the environment. In a stationary environment,

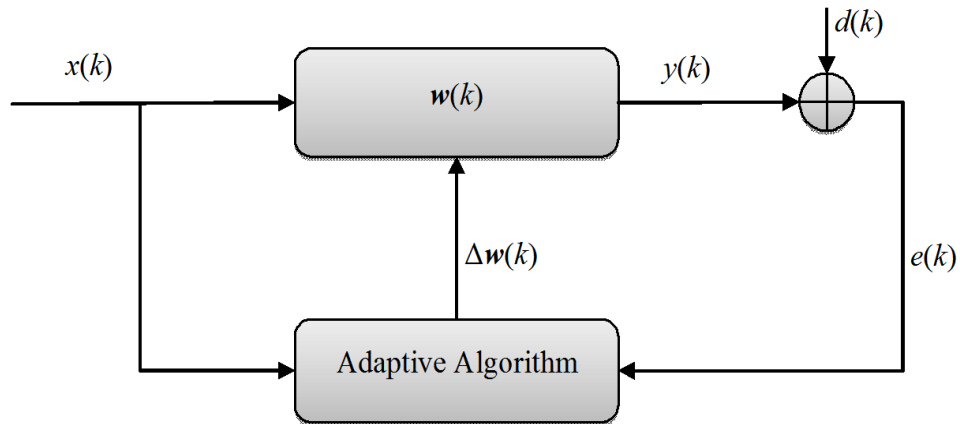


FIGURE 1.1: Block diagram of adaptive filtering system.

the filter is expected to converge to an optimal solution. In non-stationary environments, the filter is expected to track time variations and update its transfer function accordingly. Traditional signal processing techniques utilize integer order calculus for optimization of filter weights to minimize the error. In adaptive strategies, the gradient search method and its variants have been employed. These are among the most celebrated algorithms due to their cost effectiveness, robustness and ease of implementation.

There are two basic operations involved in adaptive filtering process: adaptation and filtering. In adaptation, the filter tries to adjust its coefficients based on some constraints such as error minimization, tap-weight length or maintaining orthogonality. In filtering, a desired output is produced. The mean squared value of the error signal is often used as the optimization criterion. Important parameters also include numerical stability, accuracy and robustness. Adaptive filters are vital to many applications such as instrumentation, communication, controls and bio-medical signal processing.

A block diagram of an adaptive filtering system is shown in Figure 1.1. At time k , the filter has input $x(k)$ and desired output $d(k)$. It is assumed that both the observed signals are corrupted by zero mean additive noise terms. Using vector notations and assuming an N^{th} order filter, the estimated output $y(k)$ of the filter is:

$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k) \quad (1.1)$$

where

$$\mathbf{w}(k) = [w_0(k), w_1(k) \dots w_{N-1}(k)]^T \quad (1.2)$$

is a vector having the filter weights at time k and

$$\mathbf{x}(k) = [x(k), x(k-1) \dots x(k-N+1)]^T. \quad (1.3)$$

Note that the superscripts H , T and $*$ represent Hermitian, transpose and conjugate operations, respectively. The goal of the filter is to produce output $y(k)$ which is best estimate of the desired signal. This is achieved by devising a cost function represented by $J_w(k)$ which is a function of the error $e(k) = d(k) - y(k)$. Most commonly used criterion for optimization is the Mean Squared Error (MSE) which is defined as [1]:

$$J_w(k) = E[e^2(k)]. \quad (1.4)$$

This cost function has some attractive features such as [1–6]:

- Physical relevance to energy
- Smooth performance surface with continuous derivatives
- Single global minimum (convex paraboloid) of the performance which allows unambiguous selection of optimum parameters
- In the absence of noise, the minima provide best unbiased estimation
- For predication, minimum phase is guaranteed

As stated above for the MSE, there is a single global minimum, the optimal solution is called the Wiener filter [1–6], and is given by:

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p} \quad (1.5)$$

In equation (1.5), \mathbf{R} is the autocorrelation matrix of the input and is defined as $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^T(k)]$. The vector \mathbf{p} is obtained from the cross correlation of input and the desired signals, that is, $\mathbf{p} = E[\mathbf{x}(k)d(k)]$. The matrix \mathbf{R} is symmetric and Toeplitz, which facilitates a computationally efficient inverse operation with complexity of $O(N^2)$ [5, 6]. The computation of \mathbf{R} , however, requires all data samples, which is expensive. An iterative algorithm on the other hand is more appropriate for practical use and suits the framework of an adaptive system. The basic philosophy in such cases is to minimize the MSE at each time instant k , a correction term is applied to the filter weights to form a new set of coefficients at time $k+1$, that is:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}(k). \quad (1.6)$$

The design of adaptive filters deals with the formation of this correction term. Using the gradient search method based on first order derivatives, the most commonly applied techniques to approximate the optimal Wiener solution \mathbf{w}_o are the Least Mean Squares (LMS) and Recursive Least Squares (RLS) algorithms. The LMS is based on instantaneous error approximation of the MSE which makes it attractive for practical applications; however it has relatively slow convergence. The RLS approximates the autocorrelation matrix in a recursive manner which gives it faster convergence, but is computationally expensive. The traditional algorithms are based on Integer Order Derivatives (IOD) such as the gradient or Hessian approaches. However, the IODs are local operators [7], and are determined by properties of differentiable functions only in an infinitely small neighborhood of the considered point. The thesis proposes the applications of Fractional Derivative (FD) in adaptive algorithms for the correction term $\Delta\mathbf{w}(k)$. Since, the FD is a generalization of the ordinary IOD, the properties that make FD suitable for

modeling certain complex systems which may have “non-local dynamics”. The processes dynamics may have a certain degree of memory [8] and can better be modelled with non-local fractional operators instead of the ordinary IOD which is a local operator [9].

1.2 Steepest Descent Algorithm

In the steepest descent algorithm, the correction term in equation(1.6) is formed by exploiting a step size μ in the negative gradient direction, that is:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \nabla J_w(k). \quad (1.7)$$

Applying first order derivatives for the correction term in equation (1.7),

$$\nabla J_w(k) = -2E[(d(k) - y(k))\mathbf{x}(k)] = -2E[e(k)\mathbf{x}(k)]. \quad (1.8)$$

which can be rewritten, using equation (1.1) for $y(k)$ as:

$$\nabla J_w(k) = -2E[(d(k) - y(k))\mathbf{x}(k)] = -2E[(d(k) - \mathbf{w}^T(k)\mathbf{x}(k))\mathbf{x}(k)]. \quad (1.9)$$

Simplifying equation (1.9) further and putting the value of gradient in equation (1.7), the update equation becomes:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(k)]. \quad (1.10)$$

Equation (1.10) is the update equation for the steepest descent algorithm. It can be seen that equation (1.10) still requires the knowledge of \mathbf{p} and \mathbf{R} . This algorithm has theoretical significance, rather than practical application as an adaptive filter.

1.3 Least Mean Square Algorithm

An approximation of equation (1.8) is to use the instantaneous value of the error, this results in the Least Mean Square (LMS) algorithm. The correction term is given by:

$$\nabla J_w(k) = -2e(k)\mathbf{x}(k). \quad (1.11)$$

A generalized update equation based on time varying step size for the LMS algorithm is given by:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k). \quad (1.12)$$

The complexity of LMS is $O(N)$ which is attractive for online computations in adaptive filtering applications. The LMS algorithm, however, exhibits slow convergence and excessive misadjustment in the presence of noise since the instantaneous value of the gradient may not be zero [2–6, 10]. The consequence is perturbation of the tap-weight vector around the optimal weight vector even in the steady state.

1.4 Literature Review

Traditional adaptive signal processing techniques utilize integer order calculus for optimization. The gradient search method [5, 6, 11] and its approximations have attracted many researchers to work on applications in many scientific fields. The Steepest Descent (SD) [6] algorithm and its stochastic variants such as the LMS, the Least Mean Fourth (LMF) [12], the Normalized Least Mean Square NLMS [13], and the Constant Modulus Algorithms (CMA) [14] are amongst the most celebrated algorithms. These are mostly applied in system identification problems such as in Active Noise Control Systems (ANCS) [1–4, 15–24], adaptive channel equalization (explicit and implicit)[10, 25, 26] of multipath channels, adaptive

beamforming, line echo cancelation and many more due to their ease of implementation. However, these algorithms have issues of slow convergence [11]. This implies the need for more training data in communication systems [27, 28], or problems in tracking of the parameters of an unknown time varying system [29]. The only user tuneable parameter in these algorithms is the step size. A large step size increases convergence speed [30–35] but may result in degraded performance in the steady state. Conversely, a small step size helps improve performance in the steady state [32] but requires more training data for convergence.

Adaptive step sizes strategies [18, 36, 37] have been developed which keep the step size large in the start to increase convergence rate, and then reduce the step size in the steady state to minimize error. In certain applications, such as ANCS, the step size is also dependent on secondary path effects [30, 31]. This keeps the step size requirement challenging, and usually smaller step sizes have to be used resulting in slower convergence [32, 33]. One of the widely used algorithms in ANCS is the Filtered-input (x) Least Mean Squares (FxLMS) algorithm due to its better noise reduction and lower computational cost. The FxLMS algorithm has been analyzed extensively in the literature [16]. Different variants including the variable tap solutions [15, 38] for feed-forward configuration, Modified FxLMS (MFxLMS), Leaky FxLMS [17], Filtered-error LMS (FeLMS)[33] and Filtered-x Normalized LMS (FxnLMS) [18] have been proposed to improve its performance. However, the problem of slow convergence exists, especially when the number of weights is large, and divergence can result for relatively large step sizes [30, 31]. To improve convergence rate, other complex algorithms such as Filtered-x Recursive Least Squares (FxRLS) [19, 39] or Filtered-x Affine Projection (FxAP) [35] can be used. However, their real-time realization is not as cost effective for most ANCS applications. To attain faster convergence and to reduce computational complexity, Fast Fourier Transform (FFT) based Frequency Domain Filters (FDFs) were developed [20]. However, these have the drawback of a time delay equal to the FFT length between the input and output [21]. To overcome the problem of delay, Sub-Band (SB) ANCS algorithms were proposed which are based on system identification (delay-less SB) and its FxLMS adaptation. Among the popular variants

of SB adaptive filters are those discussed in [22–24]. Parallel implementation with lower order adaptive filters gives faster convergence; the computational complexity is less than FDFs but higher than FxLMS.

To improve the convergence rate, other alternative approaches have been devised such as RLS or Extended RLS (E-RLS) which is equivalent to Kalman filtering [40]. However, in time varying nonstationary environments [11], the LMS, RLS and E-RLS algorithms have unsatisfactory performance after convergence. In mobile applications, nonstationary environments are modelled as time varying Rayleigh sequences. In such cases, vehicular speeds are of the order of 500 km/h for Long Term Evolution (LTE) [38] and 225km/h in Interim Standard 136 (IS-136) [41] which produce Doppler shifts approximately of 0.8KHz to 3KHz depending upon the transmission frequency and relative motion between the transmitter and receiver. The steady state performance of traditional algorithms severely degrades in this case [11, 13, 40]. The RLS achieves faster convergence but has high computational complexity as its update equation involves calculation of the inverse of correlation matrix of higher orders. The steady state behavior of the RLS degrades as the degree of dynamics increases while the LMS has superior performance [11, 40] in steady state, but suffers from poor convergence. To get improved performance in both transient and steady state, the Kalman filter can be applied, but results in high computational cost due to online solution of the Riccati equation [11].

Recently, fractional calculus has been applied in control and signal processing applications [42–45]. For a fractional order system, inputs and outputs of the system are related through differential equations having noninteger orders, the orders can be positive, negative or even complex [46]. Fractional order techniques have been exploited in many applications [47–51] used in optimal tuning of Fractional Order (FO) Proportional-Integral (PI) controllers for FO Plus Dead-Time (FOPDT) processes [44], design of a digital Riesz FO differentiator [52, 53] and ANCS [37]. Among other approaches in fractional calculus include tridiagonal matrices [54], optimal tuning of fractional controllers [55], parameters and differentiation order estimation [56], implementation [57, 58], Robustness, stability and fault-tolerance

[59–61] and transform domain fractional techniques [62, 63]. Most fractional signal processing techniques as discussed above and in [47, 49, 64] are for discrete time fixed filters [65], the development of corresponding adaptive filters is still in its infancy [66–69]. In [68–71], the standard LMS algorithm has been modified to include fractional order derivative of the gradient of the cost function in addition to the first order derivative, the new approach obtained better results in system identification problem [72]. In the area of signal processing, pioneering work is being done by Ortguerira [72, 73]. Adaptive and variable fractional order FIR differentiators have been investigated in [74, 75]. Simulation for time domain systems with fractional order has been studied in [76].

1.5 Significance of the Research

Adaptive filters are required to have faster convergence, better tracking especially in steady state, ease of implementation and numerical stability. These are mostly trade-offs, improving one result in compromising the other. Adaptive signal processing techniques have wide applications in biomedical, controls, communication and defence fields. Applications may be categorized into (1) system identification (2) inverse system modelling (3) spatial filtering (beamforming) and (4) prediction. Examples include channel estimation, noise cancellation, channel equalization and interference cancelation.

In this study, the strength of fractional calculus is exploited to develop fractional order algorithms to improve performance of some existing signal processing algorithms. These fractional algorithms have comparable computational complexity and are easy to implement. The effects of varying the fractional order on the convergence and performance of the algorithm have been studied. It is seen that fractional algorithms provide improvement in convergence as well as steady state error.

Various algorithms such as LMS, NLMS, CMA, FxLMS are modified using fractional signal processing techniques to improve performance in terms of convergence as well as tracking in the steady state. The modified algorithms exhibit

faster convergence and have better tracking performance due to two extra control parameters: the fractional order step size and the fractional order. It is worth mentioning that the step size is not only adapted according to the input signal energy but also according to the fractional order.

1.6 Research Aims

The aim of this research is to apply fractional derivative in the stochastic gradient calculation to improve the performance of adaptive algorithms. The performance is seen in various applications of system identification and tracking.

The first application is equalization of multipath channels with the objectives to nullify efficiently the effects of inter symbol interference. Examples included are linear equalization, decision feedback equalization and blind equalization based on constant modulus algorithm. Typically, linear equalizers are implemented with Finite Impulse Response (FIR) filters (feed-forward configuration), the weights are adjusted using a training symbol set which is known to both the transmitter and receiver. Decision Feedback Equalizers incorporate an FIR structure in the feedback path and operate on the decisions from the detector. The objectives are to improve symbol error rate performance, in addition to the use of smaller number of training symbols to improve frame efficiency. Better equalization techniques also allow using higher order modulation schemes that further enhance the data rate which is highly desirable in modern day communication systems.

The second application is ANCS which has numerous applications such as mitigating noise in personal hearing aids, duct-acoustics, room-acoustics, engine exhausts, head sets, vehicle enclosures, vibrating machines and aircraft cabins [1–4, 32–34]. Fractional order variants are developed for ANCS which have faster convergence and are easily implementable solutions as required in such applications due to miniaturization of electronic modules.

The third application is tracking. Fast convergent algorithms based on integer order derivatives have degraded tracking performance in non-stationary environments such as Rayleigh channels with higher Doppler shifts. In this regard, the

performance of fractional order variants is evaluated in Rayleigh channels and compared with the traditional counterparts.

Fractional signal processing techniques are introduced in most of the variants of the LMS algorithm to improve the performance. This approach takes advantage of using fractional derivatives [16, 17] in addition to the standard positive integer order for minimizing the MSE. These algorithms exhibit faster convergence and have better tracking performance. Fractional (Fr) order variants have been developed such that for the LMS, it is abbreviated as FrLMS, for CMA as FrCMA, for normalized LMS as FrNLMS, for FxLMS as FrFxLMS, for modifiedFxLMS as FrMFxLMS and for FeLMS the corresponding version as FrFeLMS. In these algorithms, the step size is not only adapted according to the input signal energy but also FO which further controls the learning process.

1.7 Contributions of the Thesis

This research study contributes in the following applications.

1.7.1 Multipath Channel Equalization

The main contributions for the channel equalization problem are summarized as under:

- Design of FO variants of the LMS and NLMS algorithms for channel equalization (both Feed-Forward (FF) and Decision Feedback Equalization (DFE) configurations).
- Novel Tapped-delay line filter structures for DFE for the new fractional schemes.
- Verification and validation of the proposed algorithm for different modulation schemes, step sizes and fractional orders showing faster convergence and improved Symbol Error Rate (SER).

- Development of fractional order variants of CMA for fast convergence, and low mean squared error performance (for larger step sizes).

1.7.2 Active Noise Control Systems

In ANCS, the main contributions are as follows:

- Design of Fractional Normalized FeLMS (FN-FeLMS) for feed-forward configuration.
- Development of FO variants of NLMS, FxLMS, MFxLMS and FeLMS.
- Verification of the proposed algorithms using different step sizes and fractional orders.
- Performance analysis for different sources such as binary, Gaussian and impulsive noises

1.7.3 Tracking of Rayleigh Fading Sequences

In tracking, this research contributes in the following:

- Verification and validation of the fractional algorithms using different step sizes and fractional orders in both stationary and nonstationary environments, and their comparative analysis with RLS, E-RLS and NLMS algorithms.
- A hybrid approach where the initial weights are trained with RLS algorithm for fast convergence, and subsequently the fractional algorithm is applied for good mean squared deviation/mean squared error (MSD/MSE) performance

1.8 Thesis Organization

The rest of the thesis is organized as follows: Chapter two provides a comprehensive survey of developing update equations for different algorithms along with their limitations. The LMS and its variants are considered for applications in ANCS and equalization. For ANCS, FxLMS based algorithms for feedforward configurations have been presented. For adaptive equalization problem, LMS and NLMS have been presented. The generalized Godard algorithm and its variants have been also discussed. The tracking problem consisting of Rayleigh fading sequences is discussed.

Chapter three provides mathematical background of fractional derivatives, different definitions of fractional derivatives and the β function which is used in such representation. Firstly, fractional derivatives have been introduced. Definitions of specialized functions have been provided along with examples of fractional derivatives of some functions. The development of fractional LMS has been presented and its limitations have been stated.

Chapter four deals with the application of fractional LMS and its modified version in equalization of multipath fading channels. The mapping function is modified by introducing a nonlinear term based on Riemann-Liouville definition of fractional derivatives. Fractional variants of the LMS algorithm and its normalized version have been derived. The decision feedback equalization configuration for higher order quadrature amplitude modulations has been considered. Comparative results are shown in terms of performance metrics as symbol error rate for various fractional orders and step sizes, combined equalizer and channel responses for different number of training symbols.

Chapter five contributes in applying fractional order signal processing to improve the constant modulus algorithm (CMA-2). The cost function is based on the posterior error. Different fractional orders and step sizes have been considered. In addition, simulation results are shown for blind equalization of quadrature phase shift keying symbols in a multipath fading environment.

In chapter six, a novel fractional normalized filtered-error least mean squares (FN-FeLMS) algorithm is designed for ANCS in the presence of secondary path model. In this regard, the main contribution of the chapter is to investigate the use of fractional calculus concepts in ANCS; both positive and negative fractional orders are employed based on the differintegral operator. The proposed arrangement is evaluated for a number of different scenarios by varying the step size and fractional orders. Furthermore, FO variants of FxLMS algorithm including MFxLMS, FeLMS and FN-FeLMS have been developed. The computational complexity of the proposed algorithms along with comparative analysis is discussed. Simulation results and analysis for various fractional orders, different inputs and step sizes are presented. A number of scenarios have been considered where the input signals modelled as binary, Gaussian or impulsive.

Chapter seven presents the tracking behavior of FO variants of the NLMS algorithm in a nonstationary environment modeled as time varying Rayleigh fading sequence. In evaluation, a high speed mobile environment is considered which results in different Doppler frequency shifts depending upon the transmission frequency, relative velocity of the transmitter and receiver. The proposed algorithms are compared with the NLMS, RLS and E-RLS schemes. A hybrid scheme is also shown where the weights of an FO variant are initially trained with RLS and then perform self-adaptation.

Finally, chapter eight provides a summary of the contributions, discusses the future research directions based on thesis findings, and specify concluding remarks.

Chapter 2

Adaptive Signal Processing Algorithms and Applications

This chapter briefly outlines some important algorithms and applications. For the supervised case, the Least Mean Square (LMS) algorithm and its variants are considered for equalization and active noise control systems. For the unsupervised case, Constant Modulus Algorithm (CMA) is considered for blind equalization. The Recursive Least Squares (RLS) algorithm is also discussed.

The presentation of the chapter is as follows: Section 2.1 is about the equalization problem and algorithms such as LMS, normalized LMS (NLMS). The generalized Godard algorithm and its variants are also discussed. In Section 2.2, algorithms for feedforward ANC systems are presented. Section 2.3 is about tracking and the RLS algorithm. The RLS algorithm is used as the benchmark for tracking Rayleigh fading sequences.

2.1 Adaptive Equalization

In terrestrial radio environment, signals are mainly received via echoes (reflections and diffractions). Multipath propagation delays lead to random superposition of

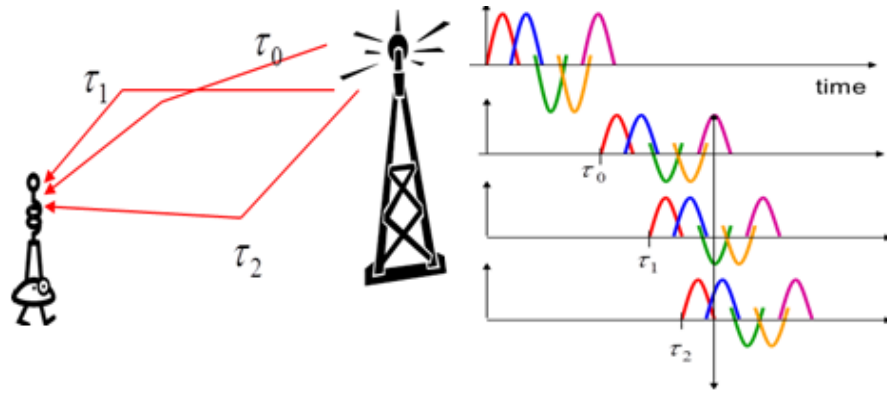


FIGURE 2.1: Inter-Symbol Interference in a multipath environment.

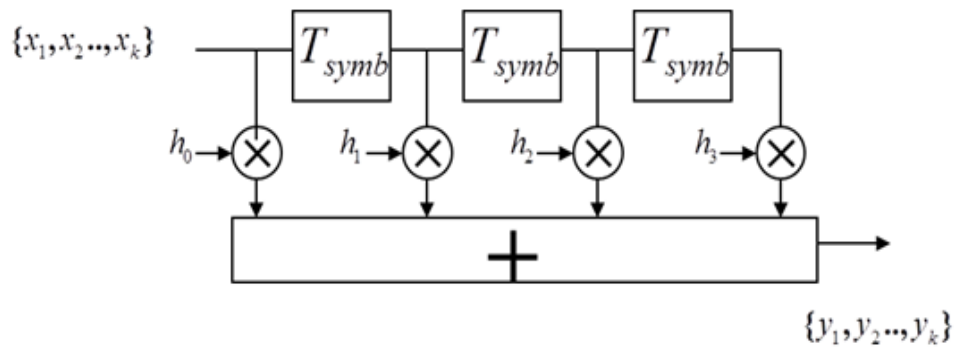


FIGURE 2.2: FIR tapped delay line modelling a multipath channel.

signals at the receiver. This phenomenon is called fading. Depending on multipath propagation delays, a degrading effect known as Inter-Symbol Interference (ISI) may occur. Figure 2.1 represents the transmission scenario of a binary phase shift keying data stream. When the receiver samples the 5th symbol, it will contain interference from the 3rd and 2nd symbols, that is, $y_5 = h_0x_5 + h_1x_3 + h_2x_2$. The ISI phenomenon depends on the transmission rate T_{symb} and the morphology (scatterer distribution) of the environment. In indoor environments, root mean square delay of up to hundreds of nanoseconds can occur for a data rate of 20Mbps ($T_{symb} = 50$ ns). Looking at the output of the receiver's sampler, the ISI phenomenon can be mathematically modelled as a Finite Impulse Response (FIR) filter as shown in Figure 2.2. The channel tap values are random complex numbers with Rician/Rayleigh magnitudes. In the frequency domain, this has a distorting effect on the signal spectrum as the wireless channel is time varying in nature [6, 11, 77–79].

To cancel ISI, both linear and Decision Feedback Equalization (DFE) techniques

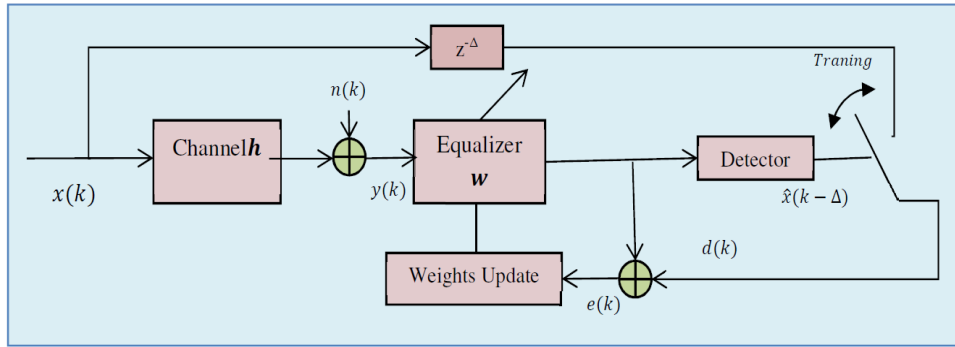


FIGURE 2.3: Functional Schematic diagram of the equalization process.

are applied. For severely faded channels, the effects of ISI cannot be removed effectively with feed-forward linear equalizers. To solve such issues, the DFE uses an additional feedback filter to reduce post decision ISI [5, 6, 10]. There are two modes of an adaptive equalizer: training and decision directed [2]. In training, a known pseudorandom sequence set of symbols is sent by the transmitter. The equalizer filter adjusts its coefficients appropriately until the symbols in the sequence are received without ISI. The training length and frame size are decided in such a way that the equalizer weights remain optimum during data transmission phase. A functional block diagram for the equalization process is shown in Figure 2.3. The approach is inverse system identification where the input signal $x(k)$ passes through the channel \mathbf{h} having M taps. The received sample $y(k)$ (corrupted by additive noise) is given by:

$$y(k) = h_0x(k) + \sum_{i=1}^{M-1} h_i x(k-i) + n(k) \quad (2.1)$$

The first term on the right side of equation (2.1) is the channel output due to the input symbol, the second term represents ISI due to previous symbols [1–3], and $n(k)$ is additive white Gaussian noise. The input to the N -taps equalizer is the distorted output of the channel, further corrupted by measurement noise. Ideally, the dot product of channel vector \mathbf{h} and equalizer weight vector \mathbf{w} should be unity so that the estimated and input symbols are the same. The weights are adjusted to minimize the squared error. The error at iteration or time index k represented as $e(k)$ is the difference between the desired response $d(k)$ and the output $\mathbf{w}^H(k)\mathbf{y}(k)$ of the equalizer, that is, $e(k) = d(k) - \mathbf{w}^H(k)\mathbf{y}(k)$. Adaptive signal processing

systems mostly operate on random processes; these are mostly characterized by statistical averages which are time dependent. Examples of statistical averages of the input and output signals include mean, variance, covariance, autocorrelation function, cross correlation function and so on. The performance metrics utilizes the expectation operator; these metrics are used for evaluating the performance of signal processing algorithms such as mean squared error, mean squared deviation and relative modeling error. The objective in equalization filtering is to minimize the Mean Squared Error (MSE) which is defined as the expectation of the squared error and is written as:

$$\min_{\mathbf{w}} E[e^*(k)e(k)] \quad (2.2)$$

During the training phased $d(k) = x(k - \Delta)$, which is known to the receiver. The parameter $\Delta \in \{0, 1, 2, \dots, M + N - 1\}$ is an integer which corresponds to the decision delay of the equalizer. In the decision directed mode, $d(k)$ is the output of the detector $\hat{x}(k - \Delta)$. In the following sections, various algorithms are described for optimization of equalizer weights. The design objective is to use the least training symbols (and more data symbols) in the frame for higher efficiency and better bandwidth utilization.

2.1.1 Optimal Filtering

Ignoring the decision delay Δ , and applying the expectation, the expanded equation (2.2) can be written as:

$$E[|e(k)|^2] = \sigma_x^2 - \mathbf{w}^H E[\mathbf{y}(k)\mathbf{x}^*(k)] - \mathbf{w}^H E[\mathbf{x}(k)\mathbf{y}^*(k)] + \mathbf{w}^H E[\mathbf{y}(k)\mathbf{y}^H(k)] \mathbf{w} \quad (2.3)$$

Here, $\sigma_x^2 = E[d^2(k)]$ is the average power of the transmitted symbol set. The quantity $E[\mathbf{y}(k)\mathbf{x}^*(k)] = E[\mathbf{x}(k)\mathbf{y}^*(k)]$ is the cross correlation vector \mathbf{p} between the input and output of the channel. The expectation $E[\mathbf{y}(k)\mathbf{y}^H(k)]$ is the channel output autocorrelation matrix \mathbf{R} having size $N \times N$. Defining the mean squared

error as $J_w(k) = E[e^2(k)]$, the cost function can be written in expanded form as:

$$J_w(k) = \sigma_x^2 - 2\mathbf{w}^H \mathbf{p} + \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (2.4)$$

Minimum MSE (MMSE) is the optimum value of MSE with respect to equalizer coefficients, i.e., $\min J_w(k)$ over all possible weights [5, 6, 10, 25, 26, 80]. Differentiating equation (2.4) with respect to \mathbf{w} and equating to zero, the optimal Wiener solution becomes: $\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p}$, and the resulting MMSE is:

$$J_{\min}(\mathbf{w}) = \sigma_x^2 - \mathbf{p}^T \mathbf{R}^{-1} \mathbf{p}. \quad (2.5)$$

The computational complexity of computing \mathbf{R}^{-1} is of order $O(N^3)$ [10, 25, 81], which increases quickly with N (and more so when multiplications are in the complex domain). Statistical information about the channel is also required, which is to be estimated. This information is not known *a priori* and is also time varying. In such a case, \mathbf{R} and \mathbf{p} are seldom calculated in real time as the channel impulse response varies from frame to frame.

2.1.2 The Steepest Descent Algorithm

Many techniques have been developed so that $J_w(k)$ is minimized in equation (2.4) such as in [5, 6, 10, 12, 25, 26, 80–86]. The update equation of the steepest descent algorithm is written as:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - 0.5\mu \nabla J_w(k). \quad (2.6)$$

where the step size μ needs to be suitably chosen to control the speed of convergence. The step size is bounded as $0 < \mu < \frac{2}{\sum_{i=0}^{N-1} \lambda_i}$ where λ denotes eigenvalues of the correlation matrix \mathbf{R} [5, 6, 10, 25, 26]. Substituting for the gradient in equation (2.6), the weight adaptation equation becomes:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(k)]. \quad (2.7)$$

This also requires prior knowledge of \mathbf{p} and \mathbf{R} .

2.1.3 The LMS Algorithm

Taking the first order derivative of equation (2.2), the gradient can be written as $2E[e(k)y^*(k)]$, replacing the expectation by its one point approximation, equation (2.7) can be written as the standard LMS update equation:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e^*(k)\mathbf{y}(k). \quad (2.8)$$

The LMS weight update equation is easy to implement and has a low computational cost. It does not rely on the matrix \mathbf{R} and vector \mathbf{p} for implementation. However, its convergence properties are relatively poor [5, 6, 10, 25, 26]. Various variants of this algorithm have been proposed (at the expense of computational cost), however, fast convergence and stability remain a challenge [82–84]. Step size is the only parameter that can be tuned to improve the rate of convergence of the LMS algorithm.

2.1.4 The NLMS Algorithm

A variable step size may be selected to increase the speed of convergence, minimize MSE and enhance the tracking capability of the LMS algorithm [82, 83]. Normalized LMS is used to adjust its step size automatically and try to achieve the aforementioned objectives; the variable step size in a given iteration is calculated as [5, 6, 10, 25, 26]:

$$\mu_k = \frac{1}{\|\mathbf{y}(k)\|^2}. \quad (2.9)$$

Since this step size is based on instantaneous values, there is a chance of misadjustments. A new step size parameter β with limits $0 < \beta < 2$ is introduced.

Substituting for μ_k , the NLMS update equation becomes:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \beta e(k) \frac{\mathbf{y}^*(k)}{\|\mathbf{y}(k)\|^2}. \quad (2.10)$$

The Normalized LMS (NLMS) has a faster convergence than LMS due to automatic adjustment of step size in the algorithm. Although the problem of equalization has been studied extensively, the approach here is different in the sense that the problem is reformulated using fractional calculus. Fractional order derivatives are applied to minimize the mean squared error, in addition to the standard integer order derivatives, this way the convergence is controlled by three parameters, i.e., two step sizes and the fractional order.

2.1.5 Blind Algorithms

In digital modulation the statistical parameters such as mean and variance (signal power) are usually known. Blind equalization techniques are very popular and especially useful for digital modulation [14, 66, 80, 87]. To lessen the effect of ISI and separate the source symbol from noisy and distorted data, the design objective of the blind equalizer is to minimize the mean p-power generalized Godard cost function [14] which is defined as:

$$\varepsilon = E[(|\mathbf{w}^H(k)\mathbf{x}(k)|^q - \gamma_q)^p] \quad (2.11)$$

The quantities p and q are positive numbers and different values of these constants generate different variants. For $p = q = 2$, the algorithm is called the constant modulus algorithm (CMA). In such a case, the aims minimization of error is achieved by keeping the output or modulus $|\mathbf{w}^H(k)\mathbf{x}(k)|^2$ as closed as possible to the constant value γ_2 . The constant γ_q is the constellation level to be achieved and is defined as:

$$\gamma_q = \frac{E[|x(k)|^{2q}]}{E[|x(k)|^q]} \quad (2.12)$$

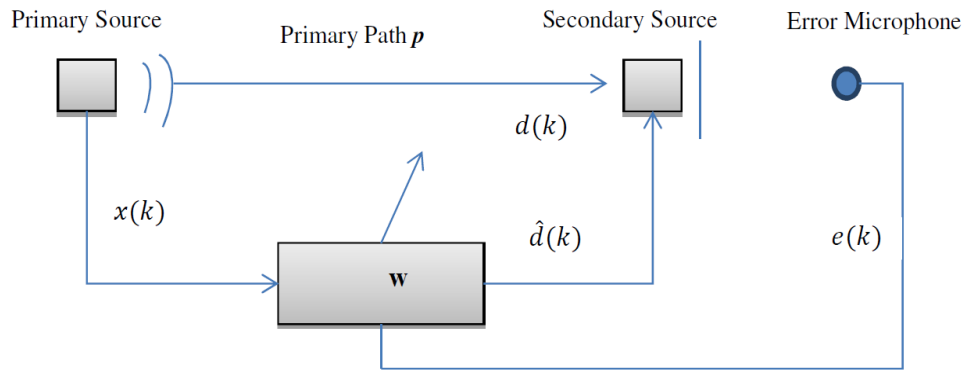


FIGURE 2.4: Active noise control system using Feedforward control.

The update equation is obtained by differentiating equation (2.11) with respect to the filter weights. The adaptation algorithm for the Godard weights [66, 87] is:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu pq(|y(k)|^q - \gamma_q)^{p-1} |y(k)|^{q-2} y^*(k) \mathbf{x}(k) \quad (2.13)$$

where μ is a positive step size.

2.2 Algorithms for Active Noise Control Systems

Active Noise Control Systems (ANCS) involve creating a destructive interference signal through a Secondary Path (SP) to cancel out the effects of noise [1]. ANCS has numerous applications including mitigating noise in personal hearing aids, duct-acoustics, room-acoustics, engine exhausts, vehicle enclosures, vibrating machines and aircraft cabins [2, 3]. Figure 2.4 shows a feed-forward ANCS and Figure 2.5 shows a functional schematic [6, 30–33]. The signal $d(k)$ is the filtered version of the noise signal $x(k)$ through the primary path (PP) \mathbf{p} . The antinoise signal $\hat{d}(k)$ is generated by an FIR filter \mathbf{w} of length N with the intent to nullify $d(k)$. The error signal $e(k)$ is filtered version as it has to travel through an unknown paths further between the secondary source and residual error measurement microphone.

The output $y(k)$ of the control filter is defined as:

$$y(k) = \sum_{m=0}^{N-1} w_m(k-1)x(k-m) = \mathbf{w}^T(k-1)\mathbf{x}(k) \quad (2.14)$$

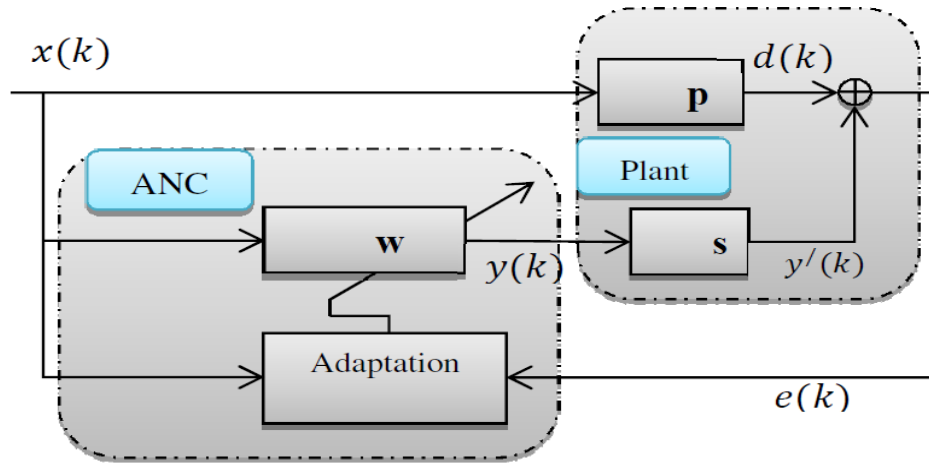


FIGURE 2.5: Functional Schematic of Adaptive ANC System.

The error is a vector of the same size as the SP having length Q :

$$\mathbf{e}(k) = [e(k), e(k-1) \dots e(k-Q+1)]^T \quad (2.15)$$

The noisy measurement $d(k)$ is given by:

$$d(k) = \mathbf{p}^T(k)\mathbf{x}(k) + g(k). \quad (2.16)$$

The term $g(k)$ in equation (2.16) incorporates measurement and modelling errors; it is assumed that this has a zero mean and having Gaussian distribution. The error is calculated as:

$$e(k) = d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k). \quad (2.17)$$

The problem in ANCS is complicated by the existence of SP between the error microphone and the control speaker. If the error microphone is considered to be ideal, that is, $g(k) = 0$, then the error becomes:

$$e(k) = d(k) - y'(k). \quad (2.18)$$

The output of the SP filter $y'(k)$ is a function of the SP filter \mathbf{s} , that is:

$$y'(k) = \mathbf{s}^T(k)\mathbf{y}(k) = \sum_{h=q=0}^{Q-1} s_q \mathbf{x}(k-q)\mathbf{w}(k-q-1) \quad (2.19)$$

The vector $\mathbf{y}(k) = [y(k), y(k-1) \dots y(k-Q+1)]^T$ is input of the SP filter, and is formed from present and past outputs of the ANC adaptive filter. Equation (2.19) shows that the identification of the PP is complicated by the existence of the SP. This prohibits the use of standard algorithms such as LMS, RLS, APA [1-3, 30-34] and others in ANCS.

2.2.1 Optimal ANCS Controller

The residual noise $e(k)$ is used to optimize the weights of the ANCS controller [31-33]. Stating $e(k)$ in terms of the SP term:

$$e(k) = d(k) - \mathbf{s}^T(k)\mathbf{y}(k) \quad (2.20)$$

The cost function $J_w(k)$ is the mean square of the residual error (noise) and is defined as:

$$J_w(k) = E[e^2(k)] \quad (2.21)$$

To have the optimal weights \mathbf{w}_o of the ANC controller, the MSE is minimized, that is, $e(k) = e_o(k)$. In such a case, the required (optimal) output of the filter is:

$$y_o(k) = \mathbf{w}_o^T \mathbf{x}(k) \quad (2.22)$$

The optimal residual noise can be written in convolution form as:

$$e_o(k) = d(k) - \mathbf{w}_o * s(k) * x(k) = d(k) - \mathbf{w}_o^T \mathbf{f}(k). \quad (2.23)$$

Where $\mathbf{f}(k) = s(k) * \mathbf{x}(k)$ represents the filtered input. For optimal weights, the cost function can be stated in expanded form as:

$$J_o(\mathbf{w}) = \sigma_d^2 - 2\mathbf{w}_o \mathbf{p}_f + \mathbf{w}_o^H \mathbf{R}_f \mathbf{w}_o. \quad (2.24)$$

where $\sigma_d^2 = E[d^2(k)]$ is the power of the primary noise signal $d(k)$. The subscript f represents the filtered version as stated above. The vector \mathbf{p}_f is the cross correlation vector between filtered input $\mathbf{f}(k)$ and output $\mathbf{y}'(k)$ while \mathbf{R}_f is the autocorrelation matrix of the filtered input $\mathbf{f}(k)$. Differentiating J_o with respect to the filter weights, setting to zero, the optimal weights are calculated as:

$$\mathbf{w}_o = \mathbf{R}_f^{-1} \mathbf{p}_f. \quad (2.25)$$

The optimal cost function or the MMSE, which is the minimum residual noise, becomes:

$$J_o(\mathbf{w}) = \sigma_d^2 - \mathbf{p}_f \mathbf{R}_f^{-1} \mathbf{p}_f. \quad (2.26)$$

2.2.2 Gradient Based FxLMS Algorithm and its Variants

The calculation of optimal weights and MMSE requires considerable computations such as \mathbf{p}_f and \mathbf{R}_f which further depends on PP and SP models. For gradient based optimization, the weight update is formed as:

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \mu \left. \frac{\partial J_w(k)}{\partial \mathbf{w}} \right|_{\mathbf{w}=\mathbf{w}(k-1)}. \quad (2.27)$$

The cost function $J_w(k)$ is a one point approximation of the expectation operator. It is defined as:

$$J_w(k) = e^2(k). \quad (2.28)$$

Differentiating $J_w(k)$ with respect to the filter weights \mathbf{w} :

$$\nabla J_w(k) = \nabla e^2(k) = 2e(k) \nabla e(k). \quad (2.29)$$

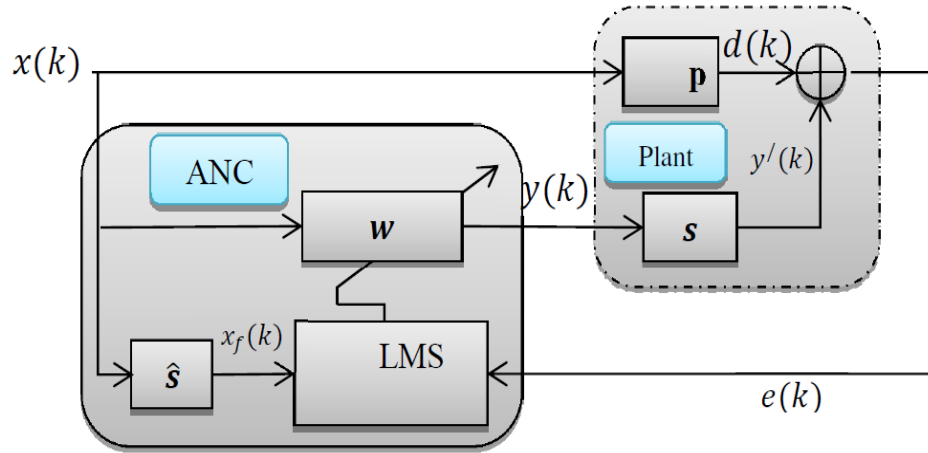


FIGURE 2.6: Schematic of FxLMS Based ANC System.

Using equation (2.20) for the error to include PP and SP, the gradient in equation (2.29) results in:

$$\nabla e(k) = -\mathbf{f}(k). \quad (2.30)$$

Subsequently,

$$\nabla J_w(k) = -2e(k)\mathbf{f}(k). \quad (2.31)$$

Substituting in equation (2.27), the FxLMS update equation becomes:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu(k)e(k)\mathbf{f}(k). \quad (2.32)$$

Practically, $\mathbf{f}(k)$ is not available directly, estimate of $\mathbf{f}(k)$ is used in the implementation of equation (2.32). A modified diagram for the FxLMS algorithm is shown in Figure 2.6. The modeling filter $\hat{\mathbf{s}}$ is used to compensate for the effects of the SP [1–3, 30, 31]. For optimal convergence, the response of $\hat{\mathbf{s}}$ should match with that of \mathbf{s} . Using $\hat{\mathbf{s}} \rightarrow \mathbf{s}$, the update equation of FxLMS [1, 31, 67] is:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu(k)e_f(k)\mathbf{x}_f^T(k). \quad (2.33)$$

where $e_f(k) = \sum_{q=0}^{Q-1} s_q(k)e(k-q)$ is the instantaneous filtered error, the output of the SP filter is $x_f(k) = \sum_{q=0}^{Q-1} s_q(k)x(k-q)$, which forms the input (filtered) of the adaptive filter. The new regression vector is formed from present and past samples of the SP filter outputs which is: $\mathbf{x}_f(k) = [x_f(k), x_f(k-1) \dots x_f(k-Q+1)]^T$. The parameter μ_k is the time varying [5, 6, 33, 34] positive step size; it controls the adaptation speed and its value depends on the input signal energy which is updated in each iteration. The filtered-x algorithm requires knowledge of the SP filter \mathbf{s} for filtering the input regression data, which causes the convergence rate to be slow, and has to employ small step sizes to avoid divergence. The filtered-error variant does not require knowledge of the filter \mathbf{s} for the input, instead it exploits the filtered-error [33, 67]; and has the following update equation:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu(k)e_f(k)\mathbf{x}^T(k). \quad (2.34)$$

Another version of equation (2.34) is the modified FxLMS (MFxLMS) which has improved convergence performance over FxLMS. The update equation of MFxLMS is [1, 21, 33]:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu(k)(e_f(k) + y_f(k) - \tilde{y}_f(k))\mathbf{x}_f^T(k). \quad (2.35)$$

where $y_f(k) = \sum_{q=0}^{Q-1} s_q(k)y(k-q)$ is the filtered output, and $\tilde{y}_f(k) = \mathbf{w}^T(k-1)\mathbf{x}_f(k)$ is the output based on the filtered input. The term $y_f(k) - \tilde{y}_f(k)$ is the estimation error in the MFxLMS algorithm [33] [88–91]. The extra terms in equation (2.35) result in increased computational complexity of the MFxLMS algorithm, but provide improved stability and convergence by better modeling the SP. The standard NLMS version is also considered which has a time varying step size. This provides the performance benchmark of system identification case without considering the effects of the SP in its update equation. It is stated as below:

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mu(k)e(k)\mathbf{x}^T(k). \quad (2.36)$$

Many other adaptive approaches are available [1–3] [20–24, 27]; these are, however, computationally expensive which prohibits their use in real-time applications. Filtered-x Least Mean Square (FxLMS) algorithm is widely applied due to its improved noise reduction performance and small computational complexity. The FxLMS algorithm has different variants including the variable tap solutions [13, 15, 16, 38, 39] for feed-forward configuration, nonlinear solutions [48, 92–94], Modified FxLMS (MFxLMS)[18], Leaky FxLMS [19], Filtered-error LMS (FeLMS) [33, 34] and Filtered-x Normalized LMS (FxNLMS) [35]. These algorithms were proposed to improve the performance of the FxLMS. To attain faster convergence and reduce computational complexity, Fast Fourier Transform (FFT) based Frequency Domain Filters (FDFs) were developed [20], however, these have the shortcoming of a time delay equal to the FFT length between the input and output [21, 22] which is overcome by Sub-Band (SB) ANCS algorithms. In SB adaptive filters as discussed in [20–23]; the convergence rate is fast and the computational complexity is considerably less than that of FDFs but higher than that of FxLMS.

2.3 Tracking of Rayleigh Fading Channels

The autocorrelation matrix \mathbf{R} and the cross-correlation vector \mathbf{p} behave as fixed quantities for stationary processes. This results in a fixed performance surface and the optimal weights are $\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$. This sets a bench mark for calculation of the MMSE and the convergence performance of different algorithms and is used for comparison. However, in many applications, the underlying processes may not be stationary. Consequently, the solution, $\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$, is time varying. In such a situation, adaptive algorithm is expected to track variations of the optimum tap weights. Tracking is a steady-state behavior in which a system is following variations of its environment after it is converged.

Consider the linear combiner as shown in Figure 2.7. The output $d(k)$ is characterized by the equation:

$$d(k) = \mathbf{w}_o^T(k)\mathbf{x}(k) + g_o(k). \quad (2.37)$$

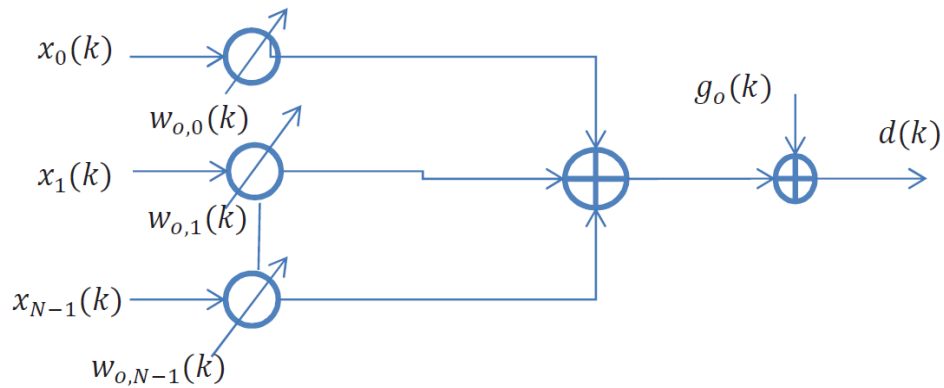


FIGURE 2.7: Linear combiner.

where $g_o(k)$ is the plant noise, $\mathbf{x}(k) = [x_0(k), x_1(k) \dots x_{N-1}(k)]^T$ is the tap-input vector and $\mathbf{w}_o(k) = [w_{o,0}(k), w_{o,1}(k) \dots w_{o,N-1}(k)]^T$ is the tap-weight vector. The time index k in $\mathbf{w}_o(k)$ emphasize that the plant tap-weight vector is time variant. The time-varying $\mathbf{w}_o(k)$ is selected to be a random-walk process defined as [5, 6, 79]:

$$\mathbf{w}_o(k+1) = \mathbf{w}_o(k) + \epsilon_o(k). \quad (2.38)$$

where $\epsilon_o(k)$ is the process noise vector. The LMS and its variants have been described earlier; the RLS algorithm is derived next. In chapter 7, a performance comparison will be made among the NLMS, RLS and fractional order variants of NLMS algorithm.

2.3.1 The RLS Algorithm

In the RLS algorithm, the input covariance matrix \mathbf{R} is iteratively estimated from its past values using the recursive relation [6, 79],

$$\mathbf{R}(k) = \mathbf{R}(k-1) + \mathbf{x}(k)\mathbf{x}^T(k). \quad (2.39)$$

The above equation is a rank-1 update on the input covariance matrix \mathbf{R} . An alternate equation is given below:

$$\mathbf{R}(k) = \alpha \mathbf{R}(k-1) + (1-\alpha) \mathbf{x}(k) \mathbf{x}^T(k) \quad 0 < \alpha < 1 \quad (2.40)$$

Further, the cross correlation vector \mathbf{p} satisfies the following recursion.

$$\mathbf{p}(k) = \mathbf{p}(k-1) + \mathbf{x}(k) d(k). \quad (2.41)$$

Application of the matrix inversion lemma [6, 71, 79] allows to recursively update the inverse of a matrix.

$$\mathbf{R}^{-1}(k) = \mathbf{R}^{-1}(k-1) + \frac{\mathbf{R}^{-1}(k-1) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{R}^{-1}(k-1)}{1 + \mathbf{x}^T(k) \mathbf{R}^{-1}(k-1) \mathbf{x}(k)}. \quad (2.42)$$

It is important to note that the inversion lemma is useful only when the matrix itself can be expressed using reduced rank updates as in equation (2.39). Summary of the RLS algorithm [6, 79] is outlined below. The starting procedure is to set $\mathbf{R}^{-1}(\mathbf{0}) = c \mathbf{I}$ with c being a large positive constant.

$$\mathbf{g}(k) = \frac{\mathbf{R}^{-1}(k-1) \mathbf{x}(k)}{1 + \mathbf{x}^T(k) \mathbf{R}^{-1}(k-1) \mathbf{x}(k)} \quad (2.43)$$

$$e(k) = d(k) - \mathbf{w}^T(k-1) \mathbf{x}(k) \quad (2.44)$$

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \mathbf{g}(k) e(k) \quad (2.45)$$

$$\mathbf{R}^{-1}(k) = \mathbf{R}^{-1}(k-1) + \mathbf{g}(k) \mathbf{x}^T(k) \mathbf{R}^{-1}(k-1) \quad (2.46)$$

2.3.2 Rayleigh Fading Channel

As already discussed in Section 2.1, signals suffer from multiple reflections with different amplitude and phase distortions while traveling from the transmitter to the receiver. Furthermore, if there is relative motion between the transmitter and the receiver, there will be destructive and constructive interferences which vary with time. A single-tap fading channel can be written as [6]:

$$h(k) = \gamma x(k) - \delta(k - k_o). \quad (2.47)$$

where the time-variant complex sequence $\mathbf{x}(k)$ models the time-variations in the channel, and k_o is the channel delay. The sequence $\mathbf{x}(k)$ is assumed to have unit variance, and the scalar γ is used to model the actual path loss that is introduced by the channel. Several mathematical models can be used to characterize the fading properties of $\mathbf{x}(k)$ or the channel. In the case of Rayleigh fading, provided $|x(k)| > 0$ for each k , the amplitude $|x(k)|$ is assumed to have a distribution defined as:

$$f(|x(k)|) = |x(k)| e^{-(|x(k)|^2/2)}. \quad (2.48)$$

while the phase $\angle x(k)$ is assumed to be uniformly distributed within $[-\pi, \pi]$:

$$f(\angle x(k)) = \frac{1}{2\pi}. \quad (2.49)$$

The Doppler frequency f_D is related to the speed of the mobile user v , and to the carrier frequency f_c , as follows:

$$f_D = \frac{vf_c}{c}. \quad (2.50)$$

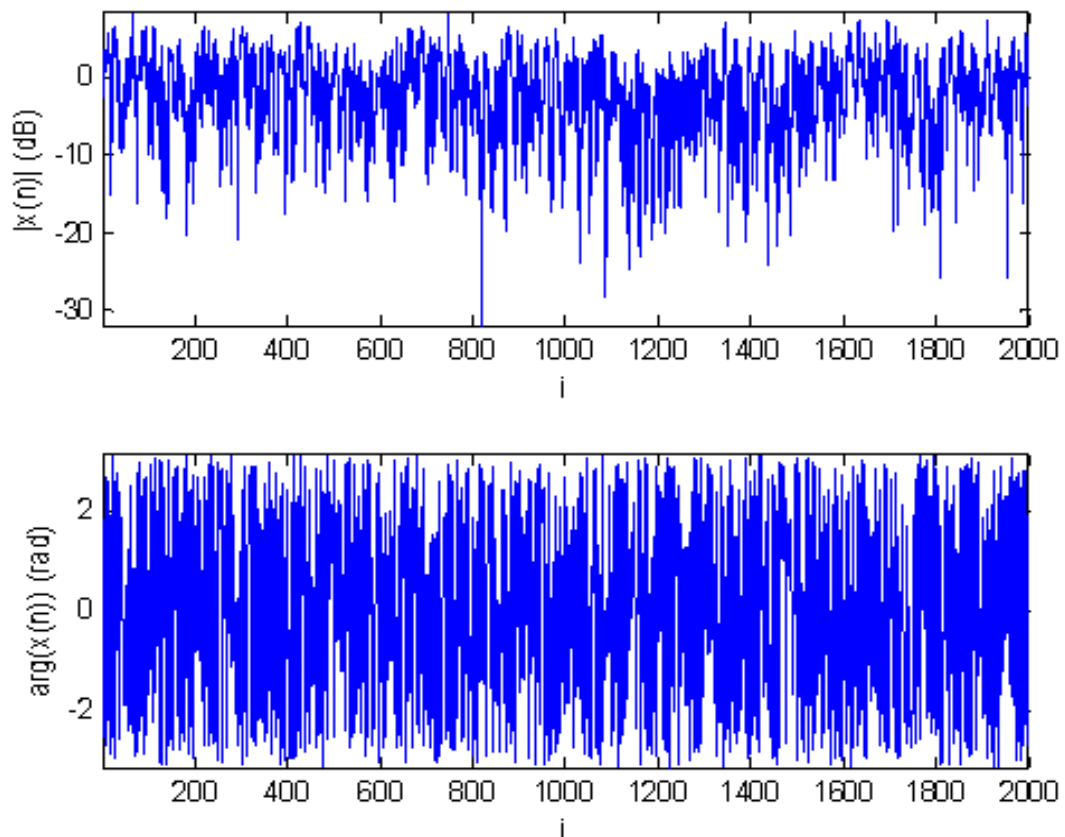


FIGURE 2.8: Amplitude and Phase plots of Rayleigh fading with Doppler Frequency of 100 Hz.

where c denotes the speed of light, $c = 3 \times 10^8$ m/s. Provided $|f| \leq f_D$ the power spectrum of the channel fading gain $x(t)$, in continuous-time, would have the following well-known U-shaped spectrum [6].

$$S(f) = \frac{1}{\pi f_D \sqrt{1 - \frac{f}{f_D}}}. \quad (2.51)$$

It is worth mentioning that (2.51) may represent the output spectrum of a fractional system. Rayleigh fading channels provide a practical example of non-stationary environment. The correlation matrix \mathbf{R} or the vector \mathbf{p} are time varying for which the optimal Wiener filter weight vector is also time varying. The adaptive filter which does not require the cross correlation vector or auto-correlation matrix in its weight update equation is required to track the changes in the optimal weights. Although the convergence performance of the RLS algorithm is

much superior than the LMS or its variants, its tracking performance is degraded for the non-stationary case [40].

Considering the input derived from a Rayleigh distribution, the performance degradation becomes more apparent when the Doppler shift frequency is increased. It is worth mentioning to note that, in time varying non-stationary environments, the LMS, RLS and their variants compromise convergence with steady state performance. At very high vehicular speeds of the order of 500 km/h for LTE, 225km/h in IS-136, or other standards can result in a maximum Doppler shift of roughly 0.8KHz to 3KHz. In such cases, the steady state performance degrades severely or even fails to perform. RLS achieves faster convergence due to its high computational complexity, since its up-date equation involves calculation of the inverse of correlation matrix of higher orders. However, the steady state behavior of RLS degrades as the degree of dynamics increases while the LMS has superior performance in steady state but has the issue of poor convergence.

An example of Rayleigh channel corresponding to a Doppler frequency of 100 Hz is shown in Figure 2.8. In the thesis, the study is provided for the tracking performance of proposed fractional order adaptive filters with NLMS and RLS algorithms for different Doppler shifts and step sizes. Chapter 7 describes the algorithms and their comparative analysis based on simulation results.

Chapter 3

Fractional Order Derivatives and Adaptive Signal Processing

This chapter introduces fractional derivatives, various definitions and important functions that are used in these representations. Fractional calculus (FC) is regarded as generalization of integer order calculus, most research work has been done by mathematicians since its inception in 1695 [42]. It remained a conceptual idea for a long time until the first conference on applications of fractional calculus in applied science and engineering in 1974, and has later on been used extensively by the research community in almost all fields [95–107].

The presentation in the rest of the chapter is as follows: Section 3.1 gives an introduction to fractional derivatives based on Euler Γ function. Section 3.2 uses the Riemann-Liouville definitions to obtain the fractional derivatives of a polynomial function. Section 3.3 gives an introduction of Fractional Least Mean Square (FLMS) algorithm. The final section provides the summary.

3.1 Introduction to Fractional Derivatives

Fractional Derivatives (FD) had remained an abstract mathematical concept for a long time until 1974 [43]. Recently fractional calculus has been applied in the

areas of control, instrumentation and signal processing [45, 46, 52, 53, 65, 95–101]. There are different ways of defining fractional integrals and derivatives, for instance, Grünwald-Letnikov, Riemann-Liouville, Hadamard, Caputo and Riesz definitions [9, 42, 43, 108, 109]. A useful discussion on different definitions and their uses in signal processing are reported in [43, 108, 110–112]. In the following, fractional order (FO) derivatives and related functions are defined. The n^{th} order integer derivative of a polynomial of power m is defined as [44]:

$$\frac{d^n x^m}{dx^n} = m(m-1)(m-2)\dots(m-n+1)x^{m-n} \quad (3.1)$$

The product term in (3.1) can be written in terms of Euler Γ functions as [9, 42, 109]:

$$\frac{\Gamma(m+1)}{\Gamma(m-n+1)} = m(m-1)(m-2)\dots(m-n+1) \quad (3.2)$$

Using (3.2), the derivatives in (3.1) can be written in terms of Γ functions as [42, 43, 109]:

$$\frac{d^n x^m}{dx^n} = \frac{\Gamma(m+1)}{\Gamma(m-n+1)} x^{m-n} \quad (3.3)$$

where n can have non-integer values, $m \geq n$. The Euler Γ function is defined as [9, 42, 43]

$$\Gamma(x) = \int_0^\infty e^{-\tau} \tau^{x-1} d\tau. \quad (3.4)$$

The convergence of Γ function is provided by the exponential term $e^{-\tau}$ for the bounded input $x > 0$ [9, 42, 108, 109]. For a given fractional order ν , the characteristic curves of Γ functions for parameters $(2-\nu)$ and $(3-\nu)$ are shown in Figure 3.1. In this thesis the fractional order is in the range $(0 < \nu < 1)$. Resultantly the input parameters to the Γ functions, that is, $(2-\nu)$ and $(3-\nu)$ are always positive.

3.2 The Riemann-Liouville Fractional Derivatives

For a finite interval $\Omega = [a, b](-\infty < a < b < \infty)$ on the real line \Re , the left side fractional derivatives of order $(\nu > 0)$ are defined using the Riemann-Liouville

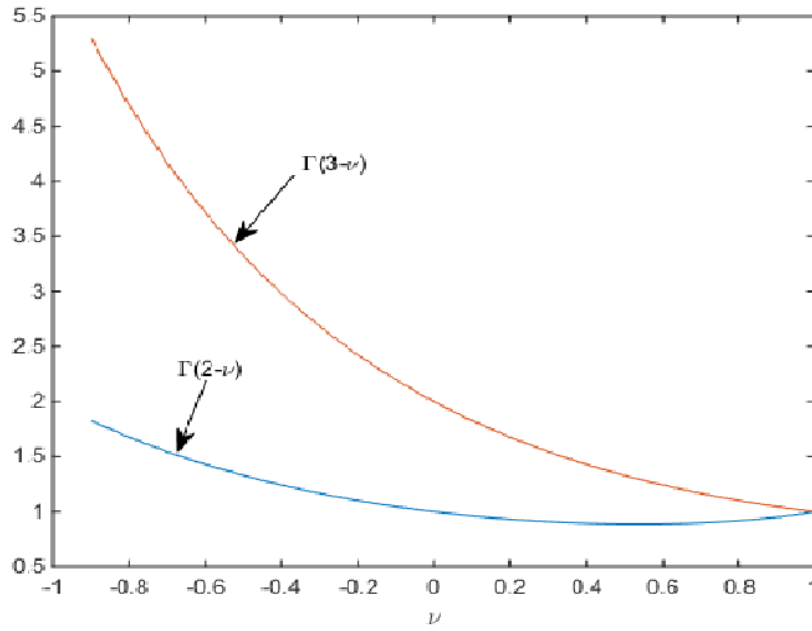


FIGURE 3.1: The Characteristic curve of the Gamma function.

(RL) definition [9, 42, 108, 113] as below:

$${}^{RL}D_{a+}^{\nu}f(x) = \frac{1}{\Gamma(n-\nu)} \left(\frac{d}{dx}\right)^n \int_a^x (x-\tau)^{n-\nu-1} f(\tau) d\tau \quad (x > a) \quad (3.5)$$

The right side ($x < b$) RL fractional derivative is defined as [9, 42, 108, 113]:

$${}^{RL}D_{b-}^{\nu}f(x) = \frac{1}{\Gamma(n-\nu)} \left(-\frac{d}{dx}\right)^n \int_x^b (\tau-x)^{n-\nu-1} f(\tau) d\tau \quad (x < b) \quad (3.6)$$

where $n = [\nu] + 1$ and $[\cdot]$ means the integer part of ν . Generally, for the case of $x > a$, the left side RL based differentiation of fractional order ν of a polynomial function $f(x) = (x-a)^{\beta}$ is defined as [9, 42, 108, 113]:

$${}^{RL}D_{a+}^{\nu}(x-a)^{\beta} = \frac{\Gamma(\beta+1)}{\Gamma(\beta-\nu+1)}(x-a)^{\beta-\nu} \quad (3.7)$$

Similarly, for the case of $x < b$, the right side RL based differentiation of fractional order ν of a function $f(x) = (b-x)^{\beta}$ is defined as [9, 42, 108, 113]:

$${}^{RL}D_{b-}^{\nu}(b-x)^{\beta} = \frac{\Gamma(\beta+1)}{\Gamma(\beta-\nu+1)}(b-x)^{\beta-\nu} \quad (3.8)$$

For $\beta = 2, a = 0$, the left side RL fractional derivative becomes:

$${}^{RL}D^\nu x^2 = \frac{2}{\Gamma(3-\nu)} x^{2-\nu} \quad (3.9)$$

For $\beta = 1, a = 0, x > 0$ (this case is used frequently in the following chapters of the thesis), the left side RL definition becomes:

$${}^{RL}D^\nu x = \frac{1}{\Gamma(2-\nu)} x^{1-\nu} \quad (3.10)$$

Similarly, for $\beta = 1, b = 0, x < 0$, the right side RL definition becomes:

$${}^{RL}D^\nu(-x) = \frac{1}{\Gamma(2-\nu)} (-x)^{1-\nu} \quad (3.11)$$

We will call D^ν as the fractional derivative operator. Moreover, for $0 < \nu < 1$ and $n = 1$, the Γ function always operates on $\Re+$.

3.3 Fractional Least Mean Square Algorithm

FO calculus possesses the potential to achieve better performance than traditional approaches with respect to both accuracy and convergence [88–91, 95–98, 114–118]. Also many theories and methods have been extended using fractional calculus, for instance, controller synthesis [99], differentiation order estimation [56], Fourier transformation [100], filters [101, 102] and system descriptions [73, 103] etc.; FO variants give more design freedom, and can exhibit better performance.

In the field of fractional signals and systems, Ortigueira et al. have done pioneering work [104, 105, 115–118]. The underlying concepts and applications are discussed in [73, 74]. Tseng et. al. have designed 1-D and 2-D FIR filters with constrained fractional derivatives [106, 107]. In [119], Wang et. al. have investigated fractional zero phase filtering using the Riemann-Liouville integral. FO describing functions and generalized power series have been studied in [120–122].

In the LMS algorithm [122], optimum weights are obtained for infinite number of iterations (ideally). The convergence is generally poor [12, 30–35, 85, 86]; although

small step size gives better steady state behavior. To improve convergence performance, fractional LMS (FLMS) algorithms [67–70] were designed by employing FO derivative of the cost function in addition to the integer derivative. Other approaches are given in [123–126]. The basic weight update equation for fractional LMS is given as [37, 68, 69]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu_1 \frac{\partial J_w(k)}{\partial \mathbf{w}} - \mu_2 \frac{\partial J_w^\nu(k)}{\partial \mathbf{w}^\nu}. \quad (3.12)$$

where μ_1 and μ_2 are the step sizes for the standard and fractional update parts, respectively, k is the time index (or iteration number) and $0 < \nu < 1$ denotes the fractional order. The notation includes an N^{th} order filter with weight vector $\mathbf{w}(k) = [w_0(k), w_1(k) \dots w_{N-1}(k)]^T$ and $\mathbf{u}(k) = [u(k), u(k-1) \dots u(k-N+1)]^T$ as the input tap-vector. Equation (3.12) is similar to the augmented error methodology which is based on MSE and minimization of the rate of change of squared error (error penalty) [127–130]. The derivation of fractional update part using fractional derivative operator can be seen in [37, 68, 69] and the final update equation is as given below:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_1 e(k) \mathbf{u}(k) + \mu_2 e(k) \mathbf{u}(k) \odot \frac{\Gamma(\mathbf{3}) \mathbf{w}^{1-\nu}(k)}{\Gamma(\mathbf{2} - \nu)}. \quad (3.13)$$

where \odot shows element-wise multiplication. The relation given in equation (3.13) is the weight update formula for FLMS. It can be seen that FLMS has more design freedom, the update equation has more tunable parameters: an extra step size parameter and a fractional order parameter which can both be adjusted to improve the convergence and performance in the steady state. It has been seen in [66, 69, 71, 88] that this class of algorithms performs better in applications such as equalization, system identification, active noise control systems and tracking of fading channels.

3.4 Summary

This chapter introduced fractional derivatives and some important definitions. The RL definition applied to obtain the fractional derivative of a polynomial. The fractional derivatives based FLMS algorithm has been stated. In the coming chapters, modifications to the FLMS are presented, and other algorithms developed with performance illustrated for various applications.

Chapter 4

Fractional Normalized Least Mean Square Algorithm for Adaptive Equalization of Multipath Channels

In this chapter, a nonlinear term based on fractional derivatives has been introduced to derive the update of Fractional Least Mean Squares (FLMS) algorithm. Different variants of the LMS algorithm and its normalized version are developed for both linear and Decision Feedback Equalization (DFE). The final update term depends on the fractional order and the in-process LMS weights. Large changes are made to the weights which help the equalizer filter to better track the effects of multipath fading channels. The working of the proposed technique in the equalization of multipath fading channels is validated for higher order quadrature amplitude modulations. Comparative results are shown in terms of performance metrics of symbol error rate for various fractional orders and step sizes, combined equalizer and channel responses for different number of training symbols.

The presentation of the chapter is as follows: introduction of the DFE and its problem formulation. In Section 4.2, fractional variants of the LMS algorithm and its normalized version are presented. In section 4.3, simulation results are

shown for higher order quadrature amplitude modulations. Finally, conclusions are derived.

4.1 Feedforward and Decision Feedback Equalization

The effects of multipath propagation include signal fading, delay spread and Doppler spread. Signal fading is caused by the interference between signals propagating through different paths, since the received signal strength is dependent on the locations of the transmitter and receiver. The delay spread is broadening in duration of received signal with respect to the transmitted signal as different delays are associated with the propagation paths. Delay spread introduces Inter-Symbol Interference (ISI) in a digital wireless communication system, which limits the achievable transmission rate [131]. Doppler spread refers to the broadening of the frequency spectrum of the received signal with respect to the transmitted signal, when there is relative motion between the transmitter and the receiver. This is due to the different angles of arrival associated with the propagation paths.

Efficient adaptive signal processing techniques are required to track channel variations in a communication system operating in a multipath propagation environment [4, 5, 131]. Equalizer is an essential part of every receiver to cancel out the effects of transmission path [5, 6, 10, 25, 26]. The complexity of equalizers grows if there are deep nulls in the faded signals [132]. Typically, linear equalizers have poor performance for channels with high amplitude distortion [5, 6, 10, 25, 26]. To compensate for high distortion, the equalizer is required to generate gains with large values. This results in noise amplification as the noise is embedded in the signal [6, 133]. Decision Feedback Equalizers (DFE) are used in such cases. The DFE utilize a Feedback (FB) filter that uses the detected symbols to produce an output which is typically subtracted from the output of the linear equalizer [133–136] or feedforward (FF) filter. In steady-state operation, the DFE contains an estimate of the impulse response (inverse) of the channel.

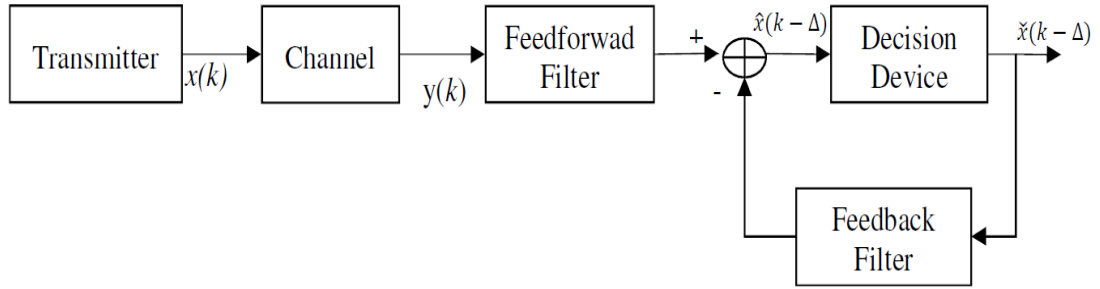


FIGURE 4.1: Communication System with Decision Feedback Equalizer.

A generalized schematic diagram of DFE is shown in Figure 4.1. The DFE process consists of the channel \mathbf{h} with M taps, FF filter \mathbf{w}_{ff} with N_f taps, FB filter \mathbf{w}_{fb} with N_b taps and the detector or decision device. The total taps of equalizer is $N = N_f + N_b$ and are adjusted through an adaptive algorithm. Assuming a direct transmission scenario, the frequency selective multipath channel is modelled as a Finite Impulse Response (FIR) filter. The symbol $y(k)$ at time k received by the receiver is the transmitted symbol $\mathbf{x}(k) = [x(k), x(k-1), \dots, x(k-M+1)]^T$ convolved with the Channel Impulse Response (CIR), that is, $\mathbf{h} = [h_0, h_1, \dots, h_{M-1}]^T$ and disturbed by the receiver internal noise $a(k)$. Defining the channel output vector $\mathbf{y}(k) = [y(k), y(k-1), \dots, y(k-N_f+1)]^T$, each sample is given by [5, 6, 10, 133–136]:

$$y(k) = \sum_{i=0}^{M-1} h_i x(k-i) + a(k) = h_0 x(k) + \sum_{i=1}^{M-1} h_i x(k-i) + a(k) \quad (4.1)$$

In terms of vectors inner product, the equation (4.1) can be written as:

$$y(k) = \mathbf{h}^T \mathbf{x}(k) + a(k) \quad (4.2)$$

In equation (4.1), the ISI is contributed by the second term that corresponds to multipath effect from previous symbols $x(k-i)$. Equalizer design requires to recover $\mathbf{x}(k)$ from $\mathbf{y}(k)$. In the DFE, the measured output $\mathbf{y}(k)$ of the channel is passed through the FF filter with weights vector $\mathbf{w}_{ff} = [w_{f0}, w_{f1}, w_{f2}, \dots, w_{f(N_f-1)}]^T$. Similarly, the decisions from the detector are passed through the FB filter, it's output is $\mathbf{w}_{fb}^T \hat{\mathbf{x}}(k-\Delta)$ where $\mathbf{w}_{fb} = [w_{b0}, w_{b1}, w_{b2}, \dots, w_{b(N_b-1)}]^T$. The error is the difference of the training symbol $x(k-\Delta)$ and the estimated symbol $\hat{x}(k-\Delta)$ and

is defined as:

$$\tilde{\mathbf{x}}(k - \Delta) \cong \mathbf{x}(k - \Delta) - \hat{\mathbf{x}}(k - \Delta). \quad (4.3)$$

The parameter $\Delta \in \{0, 1, 2, \dots, M + N - 1\}$ is an integer which corresponds to the decision delay of the DFE. It depends on the delay spread due to multipath propagation and is characterized by the power delay profile used for extracting the timing parameters of the channel. The performance of the DFE is very sensitive to the choice of Δ [131], especially when the number of taps in the feedforward filter is small. Mostly, the sampling time and decision delays are estimated before the equalization process. Usually, $N > M$ and the optimal delay is almost half of $N + M$ [5, 131] and the number of feedforward filter taps are almost double of the channel taps. A smaller error in (4.3) means that the estimated symbol $\hat{x}(k - \Delta)$ will be sufficiently close to the transmitted symbol $x(k - \Delta)$ and the decision device will map to the correct symbol in constellation. Ideally, the decisions $\tilde{\mathbf{x}}(k - \Delta)$ are correct and equal to $\mathbf{x}(k - \Delta)$. The cost function for the DFE is the Mean Square Error (MSE) which is defined as [5, 6, 10, 131, 133, 134]:

$$J_w(k) = E \left[\left(x(k - \Delta) - \hat{x}(k - \Delta) \right)^2 \right]. \quad (4.4)$$

The objective is to minimize the cost function, that is, $\min_{w_{ff}, w_{fb}} E |x(k - \Delta) - \hat{x}(k - \Delta)|^2$. The estimated output from the equalizer filter, that is, $\hat{x}(k - \Delta) = \mathbf{w}_{ff}^H \mathbf{y}(k) - \mathbf{w}_{fb}^H \hat{\mathbf{x}}(k - \Delta)$ depends on properly adjusted weights \mathbf{w} in the training phase. The design objective is to choose \mathbf{w}_{ff} and \mathbf{w}_{fb} so that the combined impulse response of channel and equalizer is approximately $\delta(k - \Delta)$ where δ is Dirac Delta function. This is difficult to obtain in the case of spectral nulls which cause severe ISI [5, 6, 10, 133, 134]. The FB filter helps in reducing the ISI in such cases. Using a vector notation such that $\mathbf{w}(k) = [\mathbf{w}_{ff}^T(k) \ \mathbf{w}_{fb}^T(k)]^T$ and the input is $\mathbf{u}(k) = [\mathbf{y}^T(k) \ -\hat{\mathbf{x}}^T(k - \Delta)]^T$, the design objective is simplified as:

$$J_w(k) = E \left[\left(x(k - \Delta) - \mathbf{u}^T(k) \mathbf{w}(k - \mathbf{1}) \right)^2 \right]. \quad (4.5)$$

For optimization of equation (4.5) in adaptive algorithms, various solutions exist in the literature. Most are based on integer order derivatives, challenges include convergence, implementation complexity and steady state accuracy. Recent literature on the equalization problem concentrates on improving robustness and convergence [25, 26, 80], decreasing computational complexity for efficient implementation [132–134], blind techniques using Quadrature Amplitude Modulation (QAM) and Amplitude Shift Keying (APSK) in ADSL and Multi Input Multi Output (MIMO) wireless systems [66, 87, 135–137]. Researchers in the recent past have applied various strategies to improve the performance of LMS and its variants such as Leaky least mean fourth algorithm [86], bias compensated NLMS for noisy input [83], improvement of NLMS in impulsive noise [138] and the variable tap-length linear equalizer [84]. Various techniques have been investigated to design training/pilot symbols [27, 139] in order to cope with frequency selective effects.

As stated earlier in chapter one, the basic philosophy of adaptive algorithm is to minimize the MSE at each time instant k , a correction term is applied to the filter weights to form a new set of coefficients at time $k+1$, that is,

$$\mathbf{w}(k) = \mathbf{w}(k - 1) + \Delta\mathbf{w}(k). \quad (4.6)$$

The design of adaptive filters deals with the formation of the correction term $\Delta\mathbf{w}(k)$. The LMS uses the instantaneous error for the MSE and utilize the gradient approach for the calculation of $\Delta\mathbf{w}(k)$. Fractional derivatives are applied to obtain the correction term in (4.6) in addition to the standard first order gradient in different configurations. As mentioned in chapter three, the left fractional derivative D^ν of order $0 < \nu < 1$ of a polynomial function $f(x) = x^\beta$ is written as [42–44]:

$$D^\nu x^\beta = \frac{(\beta + 1)}{\Gamma(\beta - \nu + 1)} x^{\beta-\nu} \quad x > 0 \quad (4.7)$$

The gamma function is defined as: $\Gamma(\alpha) = \int_0^\infty e^{-t} t^{\alpha-1} dt$ [42–44]. For $\beta = 1$, the equation (4.7) becomes:

$$D^\nu x = \frac{\Gamma(2)}{\Gamma(2-\nu)} x^{1-\nu} \quad x > 0 \quad (4.8)$$

Similarly, for $\beta = 1$, the right side fractional derivative becomes:

$$D^\nu(-x) = \frac{\Gamma(2)}{\Gamma(2-\nu)} (-x)^{1-\nu} \quad x < 0 \quad (4.9)$$

In training phase, the desired output $d(k) = x(k-\Delta)$ and filter output $\hat{x}(k-\Delta) = \mathbf{u}^T(k)\mathbf{w}(k-\mathbf{1})$, the error is defined as:

$$e(k) = d(k) - \mathbf{u}^T(k)\mathbf{w}(k-\mathbf{1}). \quad (4.10)$$

For the function $J_w(k) = e^2(k)$, using the chain rule approximation $D^\nu e^2(k) = 2e(k)D^\nu e(k)$ with $0 < \nu < 1$, $w > 0$, the left fractional derivative operator D^ν with respect to the filter weights in (4.10) gives:

$$D^\nu e(k) = D^\nu [d(k) - \mathbf{u}^T(k)\mathbf{w}(k-\mathbf{1})] = -\mathbf{u}(k) \odot \frac{\mathbf{w}^{1-\nu}(k-\mathbf{1})}{\Gamma(2-\nu)}. \quad (4.11)$$

The symbol \odot represents element-wise multiplication between two vectors. Similarly, for $w < 0$, applying the right fractional derivative operator D^ν with respect to the filter weights in (4.10) results in:

$$D^\nu e(k) = D^\nu [d(k) - \mathbf{u}^T(k)\mathbf{w}(k-\mathbf{1})] = -\mathbf{u}(k) \odot \frac{(-\mathbf{w}(k-\mathbf{1}))^{1-\nu}}{\Gamma(2-\nu)}. \quad (4.12)$$

The following identical equation [44, 108, 123, 125] is used for taking care of the sign of the weight:

$$w^{1-\nu}(k) = |w|^{1-\nu}(k) \text{sgn}(w(k)). \quad (4.13)$$

where sgn is the sign function, (4.13) is used to unify equations (4.11) and (4.12). Generally, w can be positive or negative, using equation (4.13), the final form can

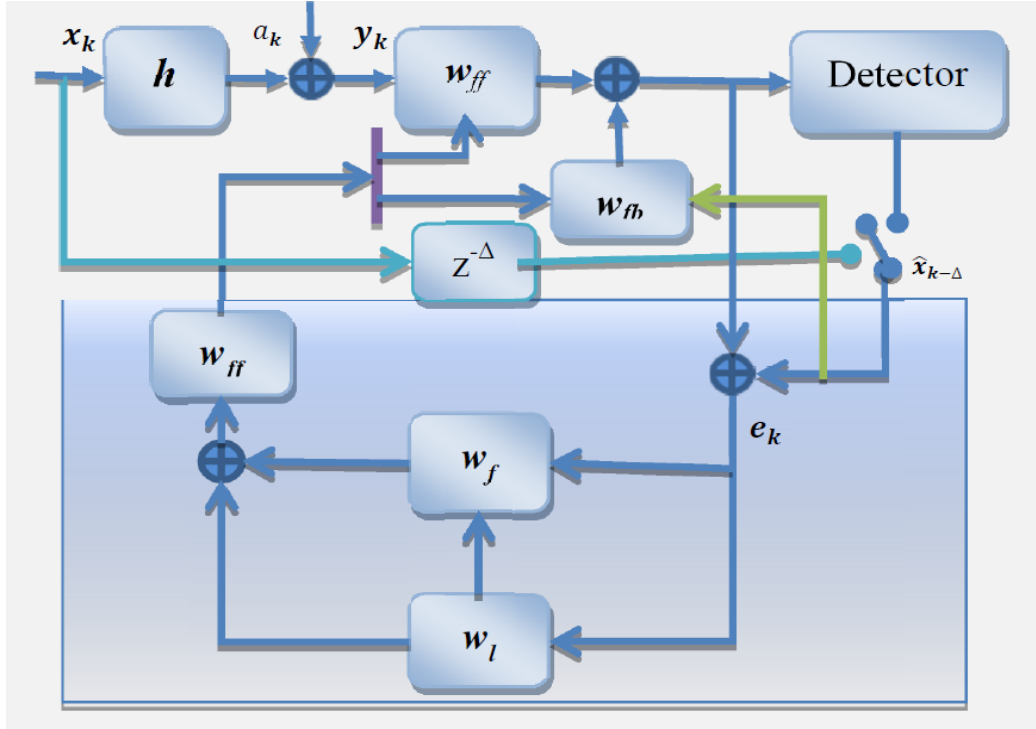


FIGURE 4.2: Filter structure of FNLMS based DFE.

be written as:

$$D^\nu e(k) = -\mathbf{u}(k) \odot \frac{|\mathbf{w}(k-1)|^{1-\nu} \text{sgn}(\mathbf{w}(k-1))}{\Gamma(2-\nu)}. \quad (4.14)$$

In the next section, two approaches are proposed for the equalization filtering based on fractional derivatives [42–44, 112] using equations (4.11) and (4.14).

4.2 Proposed FNLMS Algorithm

Figure 4.2 shows a schematic block diagram of the proposed Fractional Normalized LMS (FNLMS) algorithm based Decision Feedback Equalization (DFE) process. The equalizer has two phases as shown by the switch. One is training phase in which the desired response $d(k)$ is same as the input to the filter $x(k)$. The second is decision directed phase in which the input to the FB filter is the output of the detector. As can be seen, the input signal $x(k)$ which is mapped using square

Quadrature Amplitude Modulated (QAM) [136] constellation is passed through the fading channel \mathbf{h} . The optimization objective is to design the adaptive filter $\mathbf{w}(k) = [\mathbf{w}_{ff}^T(k) \ \mathbf{w}_{fb}^T(k)]^T$ such that the combined output of the filter produces an inverse filtering system so that the error $e(k)$ is kept to the minimum. As stated earlier, the filter tap input column vector is formed from channel output $\mathbf{y}(k)$ and past decisions of the detector $\hat{\mathbf{x}}(k)$ with the assumption that $\Delta = 0$, that is, $\mathbf{u}(k) = [\mathbf{y}^T(k) \ -\hat{\mathbf{x}}^T(k)]^T$.

In gradient based search techniques, the cost function $J_w(k) = E[e^*(k)e(k)] = E[e^*(k) (x(k) - \mathbf{u}^T(k)\mathbf{w}(k-1))]$ is minimized by differentiating with respect to the filter weights \mathbf{w} ; in our proposed scheme the differentiation of the cost function is performed by taking its fractional derivatives using the equations (4.9-4.12) in equation (4.5). The final weights are formed as: The weights are updated as:

$$\mathbf{w}(k) = \mathbf{w}_l(k) + \mathbf{w}_f(k) \quad (4.15)$$

where \mathbf{w}_l is the LMS weight vector and \mathbf{w}_f is correction in the weight vector corresponding to the fractional order derivative. In the proposed design, first of all, the standard LMS update is performed using:

$$\mathbf{w}_l(k) = \mathbf{w}_l(k-1) + \mu_l e^*(k) \mathbf{u}(k) \quad (4.16)$$

where μ_l denotes a small positive step size; it is the only design parameter which controls stability and convergence. To introduce the fractional term, the fractional operators are used in the cost function [42–44] and equations (4.8-4.14). Taking the fractional derivative of order ν of the cost function $J_w(k) = e^*(k)e(k)$ with respect to the filter weights, the update is:

$$\mathbf{w}_f(k) = -\mu_f e^*(k) D^\nu (d(k) - \mathbf{u}(k)\mathbf{w}(k)) \quad (4.17)$$

where μ_f is positive step size for the fractional part; it may be noted that the fractional update equation depends on the current LMS weights. Using the derivative

TABLE 4.1: Summary of DFE Algorithm for FNLMS (Method 1).

Filter Orders N, N_f, N_b , Step Sizes μ_l & μ_f
Input Vectors $\mathbf{u}_f, \mathbf{u}_b$
Channel Output y
Filter Output \hat{x}
Initialize G, ϵ
Initialize $\mathbf{u}_f, \mathbf{u}_b, \mathbf{w}_{ff}, \mathbf{w}_{fb}$ with zeros
Algorithm: for $k = 1, 2, 3 \dots$
$\mathbf{u}(k) = [u_f(k), u_f(k-1) \dots u_f(k-N_f+1), u_b(k), u_b(k-1), \dots u_b(k-N_b+1)]$
$\hat{x}(k) = \mathbf{w}_{ff}^H(k)\mathbf{u}_f(k) - \mathbf{w}_{fb}^H(k)\hat{\mathbf{x}}(k)$
$d(k) = x(k)$ if training phase otherwise detected symbol
$e(k) = d(k) - \hat{x}(k)$
$\mathbf{g}(k) = \frac{e^*(k)\mathbf{u}^H(k)}{\ \mathbf{u}(k)\ ^2 + \epsilon}$
$\mathbf{w}_{lN}(k+1) = \mathbf{w}_N(k) + \mu_l\mathbf{g}(k)$
$\mathbf{w}_{fN}(k+1) = \mu_f\mathbf{g}(k) \odot \frac{\mathbf{w}_{lN}^{1-\nu}(k+1)}{\Gamma(2-\nu)}$
$\mathbf{w}(k+1) = \mathbf{w}_{lN}(k+1) + \mathbf{w}_{fN}(k+1)$
$\mathbf{w}_{ff}(k+1) = \mathbf{w}_N(\mathbf{0} \rightarrow N_f - 1)$ to update Feedforward filter taps
$\mathbf{w}_{fb}(k+1) = \mathbf{w}_N(N_f \rightarrow N_b - 1)$ to update Feedback filter taps
$k = k + 1$

equations (4.11-4.13) in equation (4.17), the fractional update part becomes:

$$\mathbf{w}_f(k) = \mu_f e^*(k) \left(\mathbf{u}(k) \odot \frac{\mathbf{w}_l^{1-\nu}(k)}{\Gamma(2-\nu)} \right) \quad (4.18)$$

To improve the convergence of equation (4.16), automatic adaptive strategies based on variable step size are applied [5, 6, 25, 26, 80, 82, 83, 138], resulting in normalized LMS (NLMS) algorithm, that is,

$$\mathbf{w}_l(k) = \mathbf{w}(k-1) + \mu_l e^*(k) \frac{\mathbf{u}(k)}{\|\mathbf{u}(k)\|^2} \quad (4.19)$$

Similarly, the fractional part can also be normalized by the input signal energy as:

$$\mathbf{w}_f(k) = \mu_f e^*(k) \frac{\mathbf{u}(k)}{\|\mathbf{u}(k)\|^2} \odot \frac{\mathbf{w}_l^{1-\nu}(k)}{\Gamma(2-\nu)} \quad (4.20)$$

The common parts in equations (4.19 & 4.20) can be calculated once, a new variable $\mathbf{g}(k)$ is introduced such that:

$$\mathbf{g}(k) = \frac{e^*(k)\mathbf{u}(k)}{\|\mathbf{u}(k)\|^2} \quad (4.21)$$

This algorithm is summarized in Table 4.1. This is called **Method 1** this chapter in which the step size $\mathbf{g}(k)$ is not dependent on fractional order and is only adapted according to the norm of the input, that is, $\frac{1}{\|\mathbf{u}(k)\|^2}$.

In method 2, the final weight update equation for FNLMS algorithm is obtained with automatic step-size adaptation in the fractional part, this helps to obtain stable and fast convergence. The difference in the squares of the *a-posteriori* and *a-priori* errors [5, 6, 10, 25, 26] is given as:

$$\Delta e^2(k) = -4\mu(k)e^2(k)\mathbf{u}^T(k)\mathbf{u}(k) + 4\mu^2(k)e^2(k)\left(\mathbf{u}^T(k)\mathbf{u}(k)\right)^2. \quad (4.22)$$

Differentiating equation (4.22) with respect to $\mu(k)$ and equating to zero for the optimal value, the step-size for the standard NLMS can be calculated as [5]:

$$\mu_l(k) = \frac{\mu}{\|\mathbf{u}(k)\|^2}. \quad (4.23)$$

Taking fractional derivative D^ν using (4.7) in equation (4.22) with respect to $\mu(k)$, assuming a constant or zero deviation in squared error (which is the case of orthogonality), the following equation is obtained for a given iteration number and fractional order:

$$-\frac{\Gamma(2)}{\Gamma(2-\nu)}\mu^{1-\nu}(k) + \frac{\Gamma(3)}{\Gamma(3-\nu)}\mu^{2-\nu}(k) \left(\mathbf{u}^T(k)\mathbf{u}(k) \right) = 0. \quad (4.24)$$

Simplifying equation (4.24) results in the following relation for step-size adaptation for the fractional part:

$$\mu_f(k) = \frac{\Gamma(3-\nu)}{\Gamma(2-\nu)\Gamma(3)} \frac{\mu_f}{\|\mathbf{u}(k)\|^2}. \quad (4.25)$$

The fractional orders are kept within the bound of $0 < \nu < 1$. For the FNLMS algorithm, the fractional update becomes:

$$\mathbf{w}_f(k) = \frac{\Gamma(3-\nu)}{\Gamma(2-\nu)\Gamma(3)} \mu_f \frac{e^*(k)\mathbf{u}(k)}{\|\mathbf{u}(k)\|^2} \odot \frac{\mathbf{w}_l^{1-\nu}(k)}{\Gamma(2-\nu)}. \quad (4.26)$$

The final FNLMS update is as given below:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_l(k) e^*(k)\mathbf{u}(k) + \mu_f(k) e^*(k)\mathbf{u}(k) \odot \frac{\mathbf{w}^{1-\nu}(k)}{\Gamma(2-\nu)}. \quad (4.27)$$

The computational complexity of the NLMS equation (4.19) as well as its fractional variant (4.27) increases due to calculation of the normalization $\|\mathbf{u}(k)\|^2$ and then division of the update term by the norm; for fractional part it increases still more. However, the norm can be calculated efficiently through recursive techniques; also, the parameters for the function can be calculated beforehand.

To keep the step size within reasonable bounds, it is standard practice to add a small and positive number ϵ to the norm [4–6, 10, 25], that is, $\mu_l(k) = \frac{1}{|\mathbf{u}(k)|^2 + \epsilon}$, the final FNLMS weight update equation becomes:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_l \frac{e^*(k)\mathbf{u}(k)}{\|\mathbf{u}(k)\|^2 + \epsilon} + \mu_f G \frac{e^*(k)\mathbf{u}(k)}{\|\mathbf{u}(k)\|^2 + \epsilon} \odot \frac{\mathbf{w}^{1-\nu}(k)}{\Gamma(2-\nu)}. \quad (4.28)$$

Here, the term $G = \frac{\Gamma(3-\nu)}{\Gamma(2-\nu)\Gamma(3)}$ is a constant, its value depends on the FO. As the fractional order ν increases, the value of G decreases and vice versa. Since, the weights can be positive or negative real quantities, (4.28) can be written in a generalized form as below:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_l \frac{e^*(k)\mathbf{u}(k)}{\|\mathbf{u}(k)\|^2 + \epsilon} + \mu_f G \frac{e^*(k)\mathbf{u}(k)}{\|\mathbf{u}(k)\|^2 + \epsilon} \odot \frac{|\mathbf{w}(k)|^{1-\nu} \text{sgn}(\mathbf{w}(k))}{\Gamma(2-\nu)} \quad (4.29)$$

A simplified configuration without the FB filter is the FF linear equalization. Both the standard NLMS and fractional FNLMS algorithms applied to the feedforward (FF) equalization filtering are summarized in Table 4.2. Both NLMS and FNLMS algorithms applied to the DFE are summarized in Table 4.3. Again the FNLMS gives us more freedom for the selection of adaptation parameters. In the next section, the error performance is shown for these filters and their symbol error rate performance when used in both FF and DFE configurations.

4.3 Simulation Results

In this part, the simulation results are presented for the adaptive equalizer with the newly developed algorithms along with the standard LMS and NLMS. The equalization problem is simulated with different channels including flat fading and frequency selective channels with different Root Mean Square (RMS) Delay Spread (DS) and bandwidth. The fading channels are generated having RMS DS values of the order of 10 Nano-Seconds (nsec) to 5.0 Micro-Seconds (μsec) which covers wireless indoor (in building) to urban macrocellular (1-20 kilometres) environments. The number of taps of the channel depend on the DS, larger values of the RMS DS has more multipaths (power delay profile) which is the average received signal power as a function of delay. The power-delay profile of an environment is the received powers as a function of the delays at different locations. The second central moment of the power-delay profile is referred to as the RMS DS, and can be used as one quantitative measure of the severeness of multipath propagation. The RMS DS is dependent on the environment and the carrier frequency used for

TABLE 4.2: NLMS and FNLMS Algorithms for FF Equalization (Method 2).

Filter Order N , Step Sizes μ_l & μ_f
Input Vectors \mathbf{u}
Channel Output y
Filter Output \hat{x}
Initialize G, ϵ, ν
Initialize \mathbf{u}, \mathbf{w} with zeros
Algorithm: for $k = 1, 2, 3 \dots$
$\mathbf{u}(k) = [u(k), u(k-1) \dots u(k-N+1)]$
$\hat{x}(k) = \mathbf{w}^H(k)\mathbf{u}(k)$
$d(k) = x(k)$ if training phase otherwise detected symbol $\hat{x}(k-\Delta)$
$e(k) = d(k) - \hat{x}(k)$

NLMS Algorithm Based FeedForward Equalization
--

$\mathbf{g}(k) = \frac{e(k)\mathbf{u}^H(k)}{\ \mathbf{u}(k)\ ^2 + \epsilon}$
$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_l \mathbf{g}(k)$
$k = k + 1$

FNLMS Algorithm Based FeedForward Equalization (Method 2)
--

$\mathbf{g}(k) = \frac{e(k)\mathbf{u}^H(k)}{\ \mathbf{u}(k)\ ^2 + \epsilon}$
$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_l \mathbf{g}(k) + \mu_f \mathbf{g}(k) \odot \frac{\mathbf{w}^{1-\nu}(k)}{\Gamma(2-\nu)}$
$k = k + 1$

TABLE 4.3: DFE Algorithms Based on NLMS and FNLMS (Method 2).

Filter Orders N, N_f, N_b , Step Sizes μ_l & μ_f and Constants G, ϵ, ν
Filter Output \hat{x}
Initialize vectors $\mathbf{u}_f, \mathbf{u}_b, \mathbf{w}_{ff}, \mathbf{w}_{fb}$
Algorithm: for $k = 1, 2, 3 \dots$
$\mathbf{u}(k) = [u_f(k), u_f(k-1) \dots u_f(k-N_f+1), u_b(k), u_b(k-1), \dots u_b(k-N_b+1)]$
$\hat{x}(k) = \mathbf{w}_{ff}^H(k)\mathbf{u}_f(k) - \mathbf{w}_{fb}^H(k)\hat{\mathbf{x}}(k)$
$d(k) = x(k)$ if training phase otherwise detected symbol
$e(k) = d(k) - \hat{x}(k)$
$\mathbf{g}(k) = \frac{e(k)\mathbf{u}^H(k)}{\ \mathbf{u}(k)\ ^2 + \epsilon}$

NLMS Based Decision Feedback Equalization

$\mathbf{w}_N(k+1) = \mathbf{w}_N(k) + \mu_l \mathbf{g}(k)$
$\mathbf{w}_{ff}(k+1) = \mathbf{w}_N(\mathbf{0} \rightarrow N_f - 1)$ to update Feedforward filter taps
$\mathbf{w}_{fb}(k+1) = \mathbf{w}_N(N_f \rightarrow N_b - 1)$ to update Feedback filter taps
$k = k + 1$

FNLMS Algorithm Based Decision Feedback Equalization (Method 2)

$\mathbf{w}_{lN}(k+1) = \mathbf{w}_N(k) + \mu_l \mathbf{g}(k)$
$\mathbf{w}_{fN}(k+1) = \mu_f G \mathbf{g}(k) \odot \frac{\mathbf{w}_{lN}^{1-\nu}(k+1)}{\Gamma(2-\nu)}$
$\mathbf{w}(k+1) = \mathbf{w}_{lN}(k+1) + \mathbf{w}_{fN}(k+1)$
$\mathbf{w}_{ff}(k+1) = \mathbf{w}_N(\mathbf{0} \rightarrow N_f - 1)$ to update Feedforward filter taps
$\mathbf{w}_{fb}(k+1) = \mathbf{w}_N(N_f \rightarrow N_b - 1)$ to update Feedback filter taps
$k = k + 1$

transmission [131]. It is, therefore, extremely important for a wireless communication system to be robust against variations in channel parameters. From the shape of the power-delay profile, a channel can be characterized as time-dispersive fading or frequency-selective fading.

A comparative simulation analysis is provided of the four techniques for the performance evaluation of the adaptive linear equalizer. These include Symbol Error Rate (SER) analysis at 20 dB and 30dB Signal to Noise Ratio (SNR) values for the LMS and FLMS algorithms; other ranges of the SNR are also applied in the case of Decision Feedback Equalization (DFE). Various lengths of Training Symbols (TS) have been used in the learning phase. Five to thirty taps filters have been considered with Binary Phase Shift keying (BPSK) modulated symbols for training purpose while Quadrature Phase Shift keying (QPSK) modulation has been used for data symbols. In the case of DFE, higher order Quadrature Amplitude Modulation (QAM) are considered in the decision directed mode. Both TS and data symbols are drawn from a finite symbol set which are equally likely independent and identically distributed. The QPSK and square QAM have been considered, the former has fixed amplitude for all the symbols but different phases, in the latter, each symbol is characterized by an amplitude and phase.

4.3.1 NLMS vs FNLMS Method 1

In this section, simulation illustrations are provided showing the comparative performance of FNLMS with conventional NLMS for DFE with different channels, and with different parameters like number of FF and FB filters taps, step sizes and fractional order for FNLMS. To have a fair comparison of both the algorithms, the step sizes are kept equal in most cases, that is, $\mu_l = \mu_f$. The SER performance is shown which is the number of symbols in errors divided by the total number of symbols transmitted; it is plotted against the SNR in dBs. Other illustrations in the form of scatter plots and combined CIR and DFE performance both in time as well as frequency domain are given.

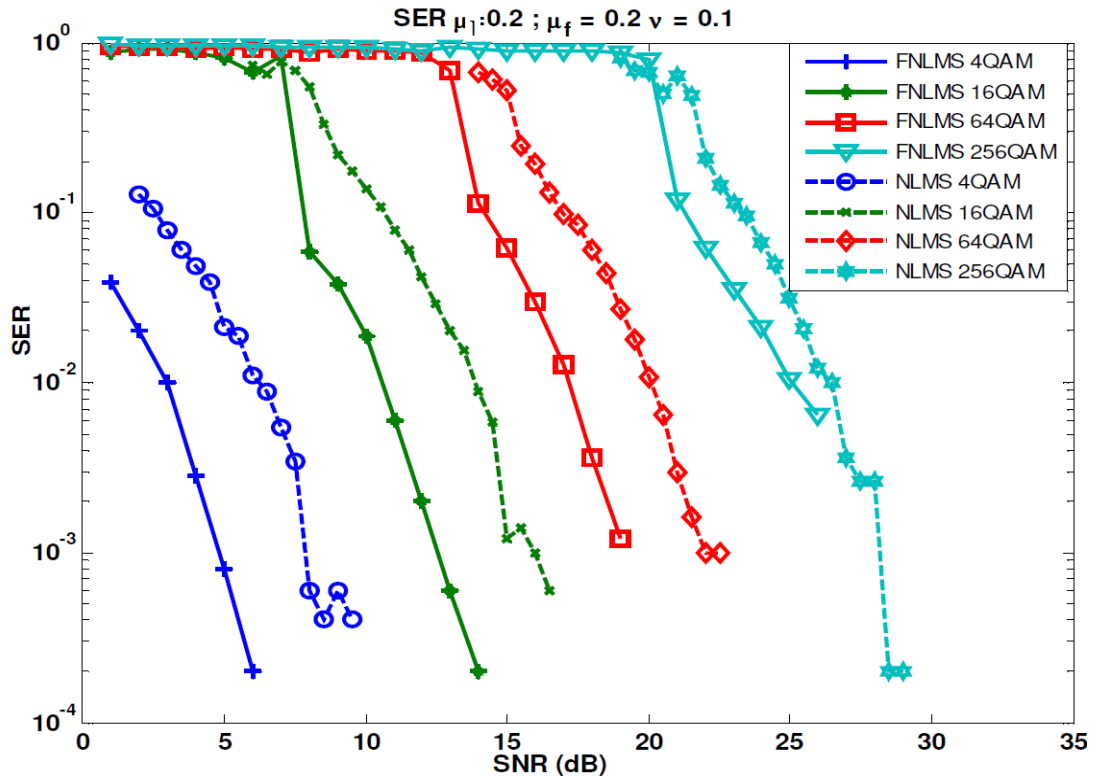


FIGURE 4.3: Symbol-error rate for QAM constellations equalized with NLMS, F-NLMS.

Different number (lengths) of training symbols (TS) are chosen which are modulated using Quadrature Phase Shift Keying (QPSK) technique and the number of data symbols are fixed to 5000, which are Quadrature Amplitude Modulation (QAM) symbols having orders 4, 16, 64 and 256, that is, each symbol carries information of 2, 4, 6 and 8 bits respectively. It is desirable to have low SER at a given SNR, a near-perfect impulse at the delay point in the time domain and close to 0 dB frequency response when the combined channel and equalizer behavior is seen; in this case, the constellation of points will be concentrated towards the mean of the given symbol.

Figure 4.3 shows SER for a channel with $H(z) = 0.5 + 1.2z^{-1} + 1.5z^{-2} - z^{-3}$; the step sizes and fractional order parameters are kept as: $\mu_l = \mu_f = 0.2$ and $\nu = 0.1$ for the equalizer filters based on NLMS and the FNLMS. For both algorithms, TS of lengths 120, 200, 400 and 600 are used for 4, 16, 64 and 256 QAM respectively; the FNLMS is seen to perform better than the NLMS. This is also evidenced by the scatter plots in Figure 4.4 for equalized symbols of 64 and 256 QAM. For a

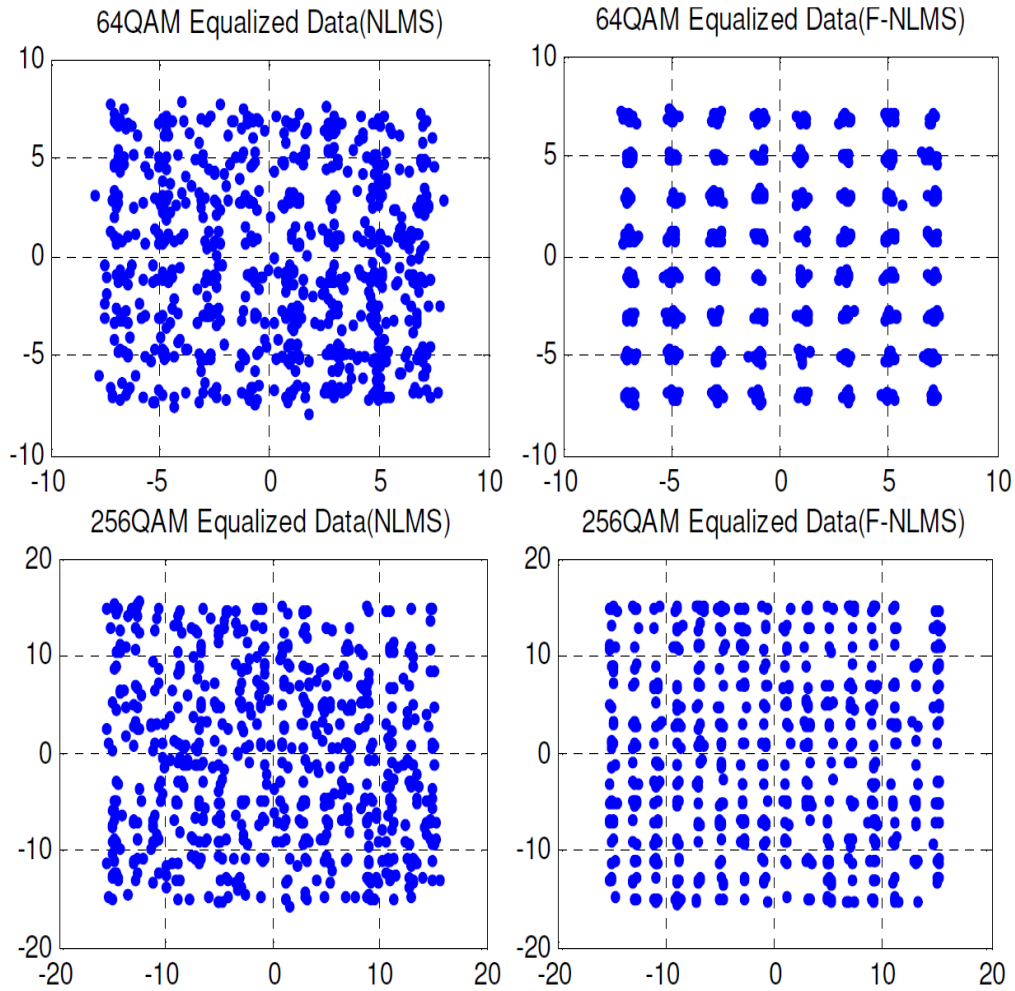


FIGURE 4.4: Scatter Plots of 64 and 256 QAM constellations equalized with NLMS, F-NLMS.

symbol rate of 10^{-3} , almost 3 dB gain is achieved for 4, 16 and 64 QAM, and about 2 dB gain for 256 QAM.

Figure 4.5 shows the SER performance for different SNR values for the channel with RMS delay spread of 50 nanoseconds and 20 MHz bandwidth. The channel model is a 12-tap FIR filter with delay $\Delta = 10$. Simulation parameter settings are: 4 QAM with SNR variation from 5 to 15 dB, 16 QAM with SNR range from 15 to 25 dB, 64 QAM with SNR range from 20 to 30 dB and 256 QAM with SNR range from 25 to 35 dB. The feed-forward filter has $N_f = 36$ taps; the feedback filter has a single tap line. Simulation results are obtained for 50 independent iterations and 1000 data symbols used in decision directed mode. The same number of training symbols has been used i.e. 100, 300, 400 and 600 for 4, 16, 64 and 256

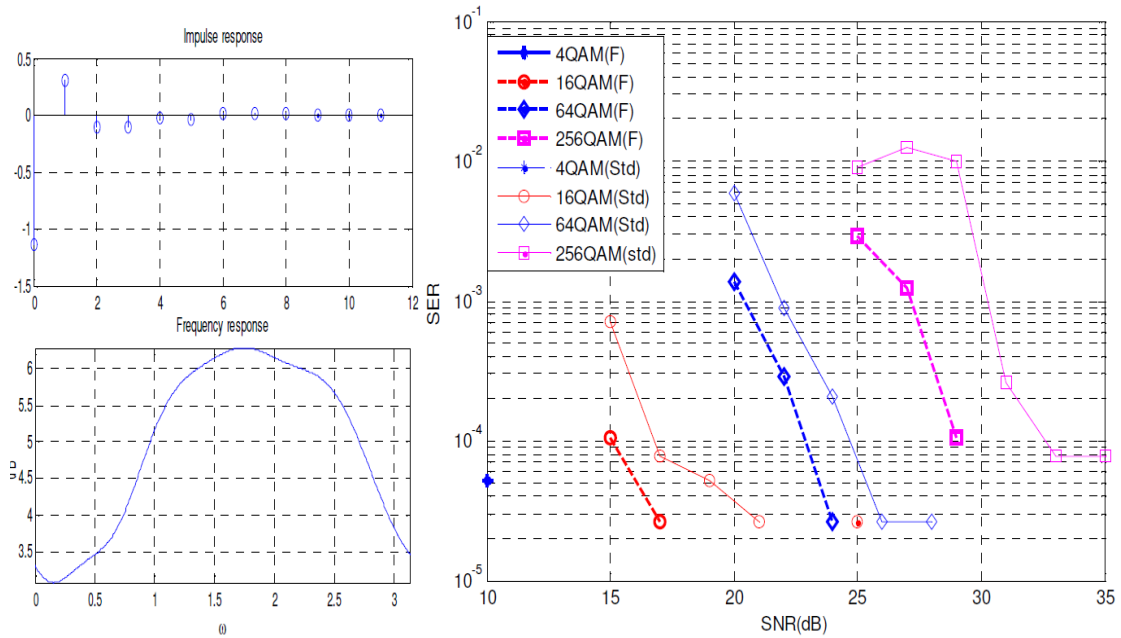
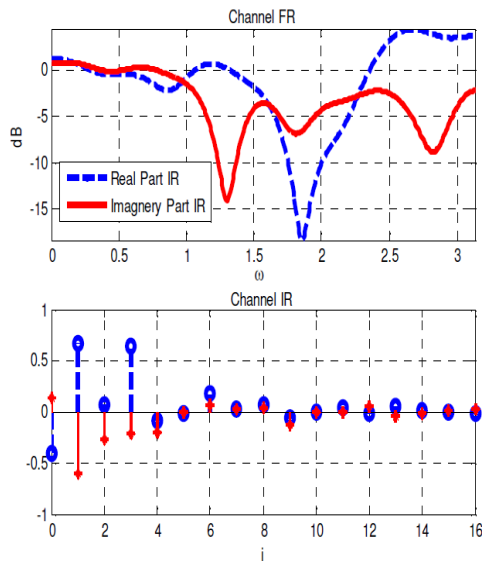


FIGURE 4.5: SER for 4, 16, 64 and 256 QAM constellations for the NLMS and FNLMS.

QAM respectively. The performance in 4 QAM is same (The SER of 8.38×10^{-5} is achieved by both the algorithms but for other higher order modulations, the fractional NLMS outperforms the NLMS; a considerable SNR gain is obtained over the NLMS even for the more stringent SER requirements of 10^{-5} .

Figure 4.6 (left) shows a 17-tap frequency selective channel with its impulse and frequency responses (IR/FR). On the right side of the figure the complex impulse response is given. It can be seen that both the real and imaginary components have deep fades at different frequencies. The real component has about -15 dB and the imaginary component has -20 dB fade; keeping the channel a very selective one.

Figure 4.7(left) shows the combined frequency responses of the channel and the equalizer for different number of training symbols. In the right part of the figure, the corresponding time domain impulse responses have been shown. It is observed that the FNLMS (red) response is almost on the 0 dB line with small peak to peak difference as compared to the standard NLMS. This is true for all cases of 100, 200, 400 and 600 training symbols for training purposes for 4, 16, 64 and 256 QAM respectively. As the number of training symbols increase, the combined response is improved which is desirable for higher order constellations and helps



$$h = [-0.407454 + i0.143711, 0.669106 - i0.601167, \\ 0.073578 - i0.268801, 0.637832 - i0.214996, \\ -0.088216 - i0.198939, -0.016475 + i0.005553, \\ 0.179073 + i0.065765, 0.022043 + i0.026168, \\ 0.076232 + i0.045919, -0.061089 - i0.128842, \\ 0.000139 - i0.003376, 0.035004 + i0.004493, \\ -0.016245 + i0.053553, 0.0544312 - i0.037306, \\ 0.010741 - i0.019895, -0.005978 + i0.014880, \\ -0.011953 + i0.023009]$$

FIGURE 4.6: Representation of Channel impulse response (right), amplitude response (top left) in dBs vs frequency and Time domain (bottom left) response of the channel (right) with 17-taps.

increase the spectral efficiency.

In Table 4.4, SER values are given for QAM symbols using the 17-tap channel (Figure 4.7); the step sizes are fixed at $\mu_l = \mu_f = 0.4$, and the performance checked for different values of fractional orders. SER values in column two correspond to the NLMS algorithm while the rest of SER values are for FNLMS. Different SNR values are selected for different QAM constellations. It can be seen that for all the fractional orders, the FNLMS perform better than NLMS.

In Table 4.5, the fractional order is fixed as 0.4 and simulations are performed for different values of step sizes. The step size for the NLMS is fixed at 0.4. It can be observed that the proposed fractional methodology outperforms the standard NLMS scheme. In most cases, the fractional variant is error free. From the simulation analysis, it is observed that the adaptive equalizers based on fractional LMS and fractional NLMS outperform the standard LMS and NLMS respectively. However, in the low SNR regime, FNLMS has comparable performance.

TABLE 4.4: SER of NLMS and FNLMS for different ν values

QAM	SNR(dB)	NLMS	$\nu = 0.01$	$\nu = 0.1$	$\nu = 0.2$	$\nu = 0.3$	$\nu = 0.4$
4	10	0.0008	2.00E-05	4.00E-05	2.00E-05	2.00E-05	6.00E-05
	12	0.0006	0	2.00E-05	0	0	0
	14	0.0002	2.00E-05	0	0	0	0
	16	0	0	0	0	0	0
	18	0.0003	0	0	0	0	2.00E-05
	20	0.0002	0	0	0	0	0
16	15	0.0009	1.60E-04	6.00E-05	1.00E-04	2.00E-04	1.20E-04
	17	0.0002	0	2.00E-05	4.00E-05	2.00E-05	2.00E-05
	19	0.0002	0	0	0	0	0
	21	0	0	0	0	0	0
	23	0.0001	0	0	0	0	0
	25	0	0	0	0	0	0
64	20	0.0046	6.20E-04	8.40E-04	6.40E-04	7.60E-04	4.80E-04
	22	0.0016	1.40E-04	6.00E-05	4.00E-05	2.80E-04	4.00E-05
	24	0.0026	6.00E-05	4.00E-05	0	2.00E-05	2.00E-05
	26	0.0023	0	2.00E-05	0	1.22E-04	0
	28	0.002	0	0	0	0	0
	30	0	1.00E-05	0	0	0	0
256	25	0.0081	7.40E-04	1.42E-03	1.30E-03	9.00E-04	6.80E-04
	27	0.0056	4.20E-04	1.60E-04	1.20E-04	2.00E-04	5.00E-04
	29	0.0015	3.80E-04	2.00E-05	0	4.00E-05	8.00E-04
	31	0.0006	0	0	6.00E-05	0	2.00E-05
	33	0.0016	0	0	0	2.00E-05	0
	35	0.0003	0	0	0	0	0

TABLE 4.5: SER of NLMS and FNLMS algorithms for different μ_2 values

QAM	SNR	BER(Std.)	BER(F)	BER(F)	BER(F)	BER(F)	BER(F)
	(dB)	$\mu_1 = 0.4$	$\mu_2 = 0.1$	$\mu_2 = 0.2$	$\mu_2 = 0.3$	$\mu_2 = 0.4$	$\mu_2 = 0.5$
4	10	0.0008	0	0.0001	0.0001	0	0
	12	0.0006	0	0.0001	0	0	0.0001
	14	0.0002	0	0	0	0	0
	16	0	0	0	0	0	0
	18	0.0003	0	0	0	0	0
	20	0.0002	0	0	0	0	0
16	15	0.0009	0.0002	0.0002	0.0001	0.0001	0.0002
	17	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001
	19	0.0002	0	0	0	0	0
	21	0	0	0	0	0	0
	23	0.0001	0	0	0	0	0
	25	0	0	0	0	0	0
64	20	0.0046	0.0005	0.0006	0.0004	0.0002	0.0003
	22	0.0016	0.0002	0.0006	0.0001	0.0002	0.0001
	24	0.0026	0.0001	0.0001	0	0	0
	26	0.0023	0	0	0	0	0
	28	0.002	0	0	0	0	0
	30	0	0	0	0	0	0
256	25	0.0081	0.0009	0.0008	0.0008	0.0007	0.0007
	27	0.0056	0.0005	0.0005	0.0006	0.0007	0.0005
	29	0.0015	0.0001	0.0001	0.0001	0	0
	31	0.0006	0	0	0	0	0
	33	0.0016	0	0	0	0	0
	35	0.0003	0	0	0	0	0

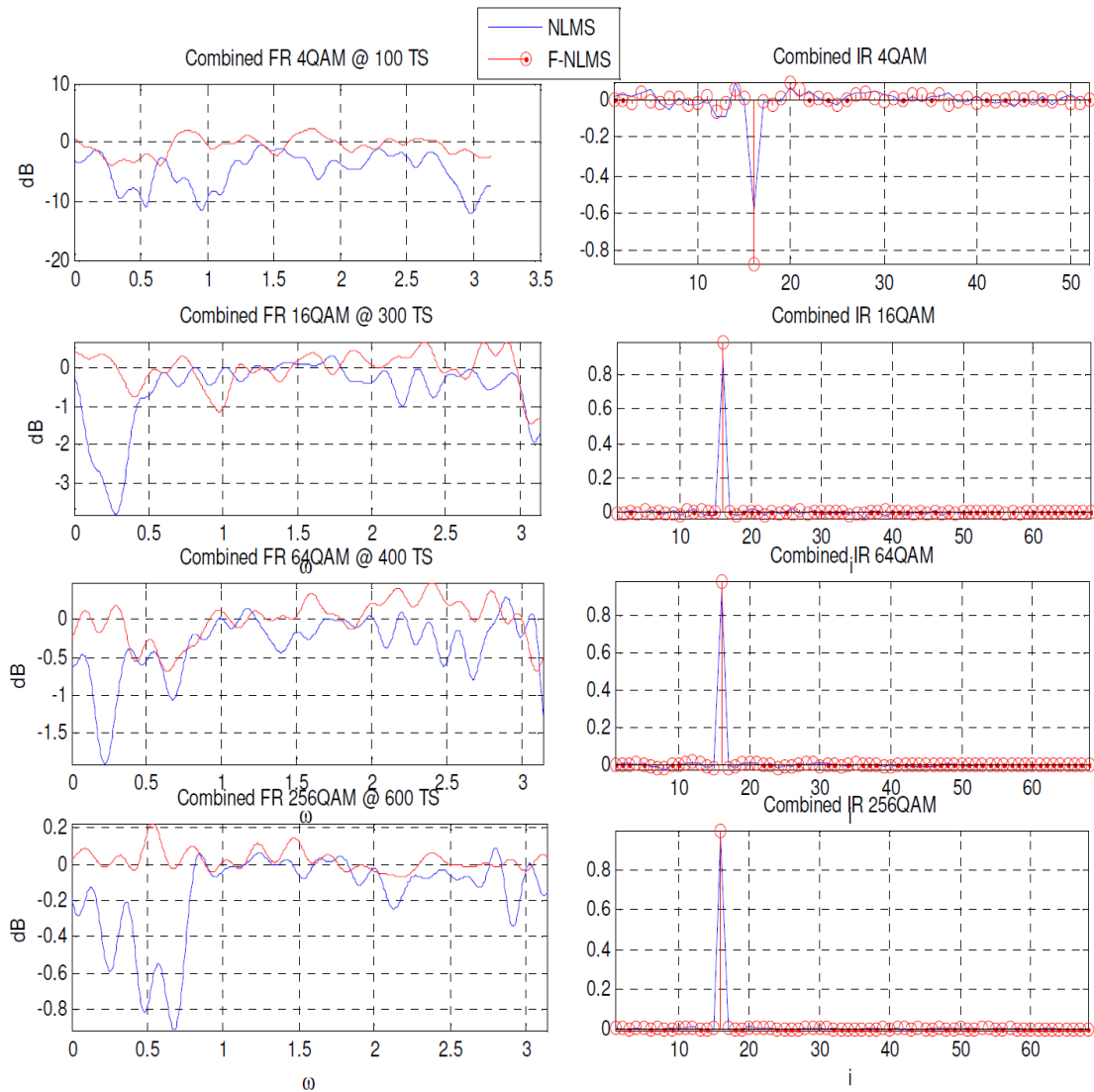


FIGURE 4.7: Combined channel and equalizer frequency response.

4.3.2 LMS, FLMS, NLMS and FNLMS Method 2

In Table 4.6, SERs are shown for LMS and FLMS for different number of training symbols. The sequence in these simulations corresponds to a slow fading channel with five taps, there are 10000 QPSK symbols used in decision directed mode. Note that, at 20 dB SNR, the FLMS has small SER with 300 Training Symbols (TS) as compared to the 500 TS used by standard LMS. Similarly, at 30 dB SNR, the SER of the FLMS algorithm at 100 symbols is not only smaller than standard LMS with 200 symbols, but comparable to that obtained with 400 TS of the LMS. It thus helps in reducing the number of TSs, resulting in increasing the packet

TABLE 4.6: SER of NLMS and Fractional NLMS Algorithms

SNR (dB)	Training	Errors LMS,	Errors FLMS	SER	SER
	Symbols	$\mu_l = 0.005$	$\mu_f = 0.005, \nu = 0.5$	LMS	FLMS
20	100	621587	180042	0.621587	0.180042
	150	498306	115111	0.498306	0.115111
	200	192945	1108	0.192945	0.001108
	300	12192	484	0.012192	0.000484
	400	2261	22	0.002261	0.000022
	500	724	17	0.000724	0.000017
30	100	734020	67782	0.73402	0.067782
	200	543595	4	0.543595	0.000004
	400	3018	0	0.003018	0

efficiency.

Figure 4.8 shows two different fading channels (almost flat) with scatter plots of the equalized symbols for various numbers of TSs. Effectively, FLMS has superior performance in terms of decreasing the variance of the error. Large values of variance result in high uncertainty and erroneous decisions by the demodulator.

In Table 4.7, SERs are shown for NLMS and FNLMS schemes for different numbers of TS. Since these algorithms exhibit faster convergence, only frequency selective fading channels are considered with up to 28 taps, 10000 QPSK symbols are used in decision directed mode. The SER is averaged over 200 independent runs and is obtained for SNR values of 5, 10, 15, 20, 25 and 30 dB. The results are generated using 100, 150 and 200 TSs. In the high SNR regime the FNLMS algorithm has superior error performance as compared to the standard NLMS scheme.

Next, the performance for different fading channels are illustrated using scatter plots for various numbers of TSs. Figure 4.9 (left) shows the scatter plots for the 27-tap channel and its equalization with 28 taps. Δ is set to 14. It can be seen that the FNLMS scheme is able to separate the symbols better than the NLMS, especially with 100 TS. It is observed that with FNLMS, there were no errors in

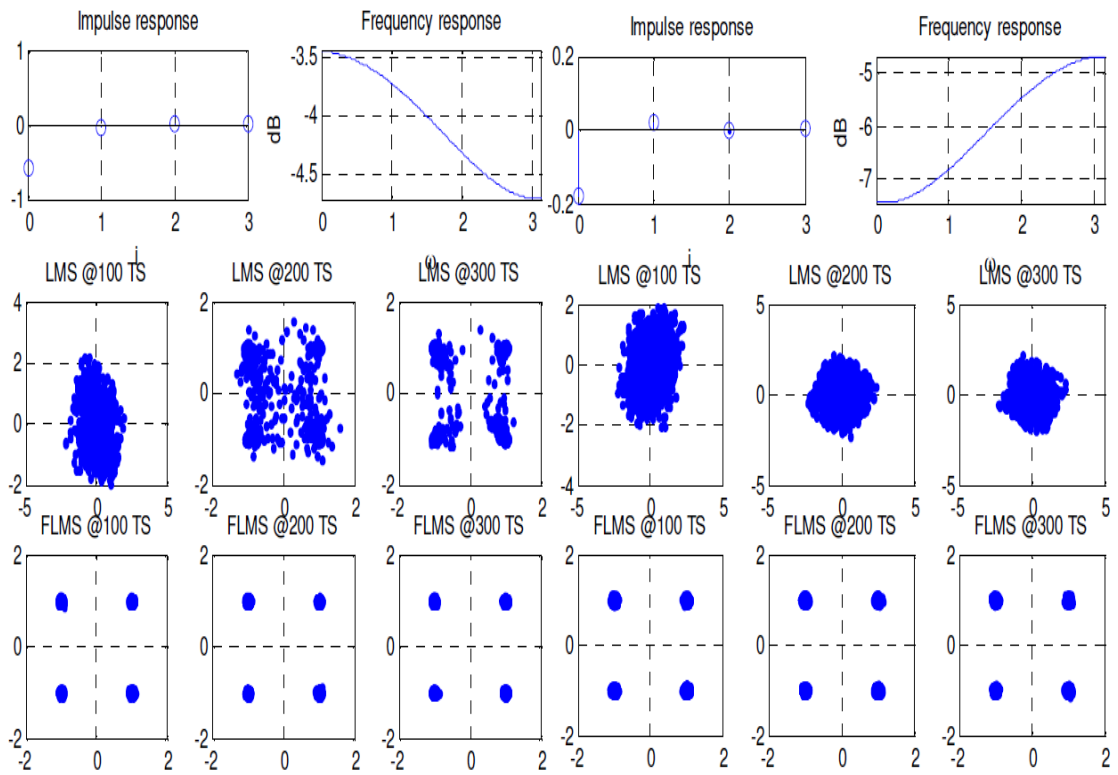


FIGURE 4.8: Scatter Plots for LMS, F-LMS, with Different Training Symbols.

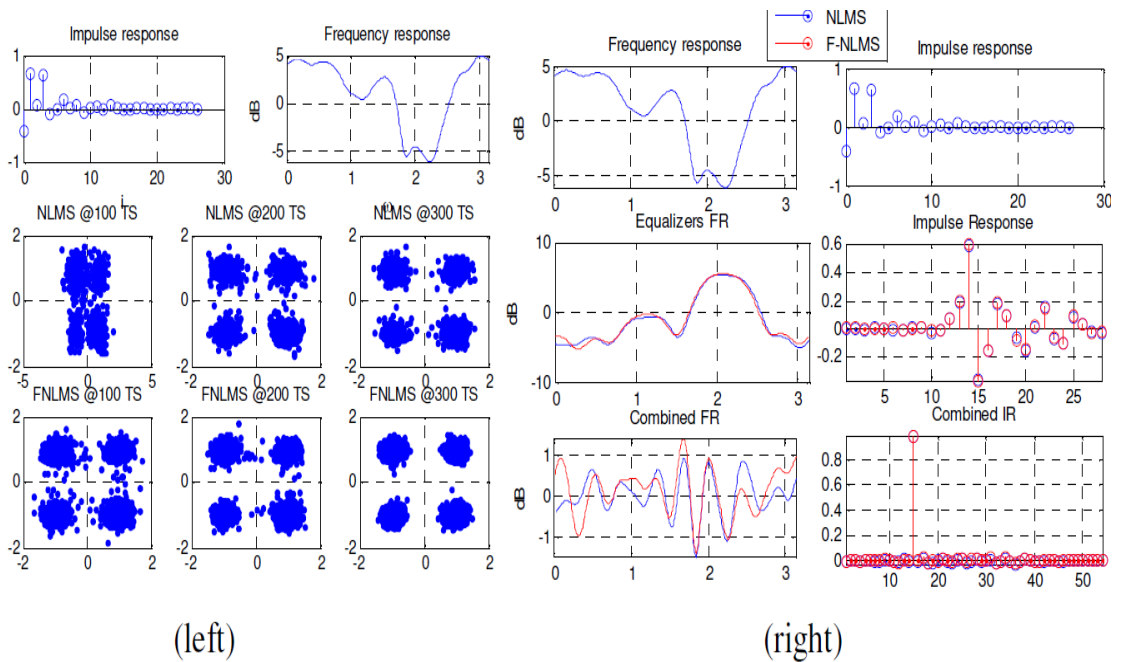


FIGURE 4.9: Scatter Plots with different training symbols and Responses of NLMS, F-NLMS.

TABLE 4.7: SER of NLMS and Fractional NLMS Algorithms

SNR (dB)	SER for NLMS, $\beta = 0.5$			SER for FNLMS, $\beta = 0.5, \gamma = 0.5$		
	100 TS	150 TS	200 TS	100 TS	150 TS	200 TS
5	0.001385	0.0010950	0.002025	0.003415	0.002165	0.002135
10	0.000165	0.0001450	0.000115	0.000150	0.000185	0.000210
15	0.000035	0.0000101	0.000010	0.000040	0.000016	0.000012
20	0.000015	0.0000015	0.000001	0.000005	0	0
25	0.000015	0	0	0	0	0
30	0.000010	0	0	0	0	0

the case of 150 and 200 training symbols while the NLMS have 10 and 5 erroneous symbols respectively.

Figure 4.9 (right) shows the frequency response of selective channel and the corresponding time domain impulse response (first row) for a 200 TS case. In this case, it is seen that no error for both algorithms, the plots correspond to correct decisions. The simulation parameters set were: $\beta = 0.4$, $\gamma = 0.4$ and SNR=20 dB. In the second row, the corresponding responses are shown for the NLMS and FNLMS equalizers, which are almost the same. Since the input to the equalizer is in the form of equation (4.1), the equalizer effectively reduces the effect of internal noise generated at the receiver. From the time domain combined response (right bottom plot), it is apparent that the symbol is correctly equalized. For a comparative analysis, the SER is given in Table 4.7. It can be inferred from the scatter plots that the FNLMS has much smaller variance than NLMS, the former, therefore, offer the attraction to use higher order QAM to enhance the data rate further and thereby increase the link efficiency.

Figure 4.10 (left) shows scatter plots for various numbers of training symbols for the given fading channel with the impulse response shown on the right side. The channel has 12 taps and is equalized with 24 taps LMS and FLMS filters. The simulation parameters are: $\mu_l = 0.01$, $\mu_f = 0.03$, $\nu = 0.1$, $\Delta = 12$, SNR=30 dB, a

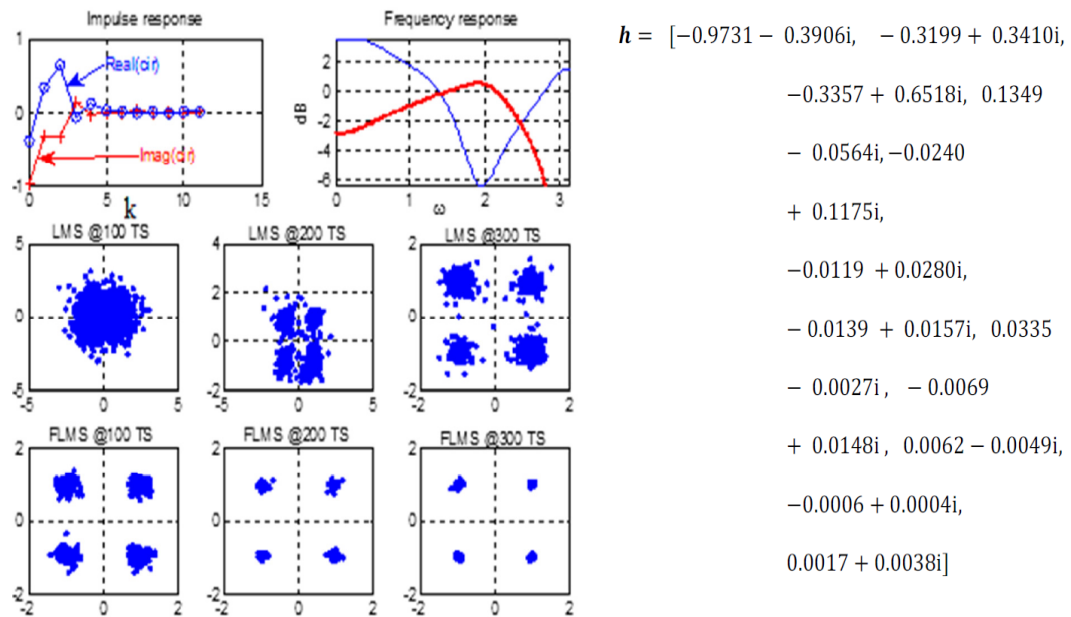


FIGURE 4.10: Scatter Plots for LMS, FLMS with [100,200,300] Training Symbols.

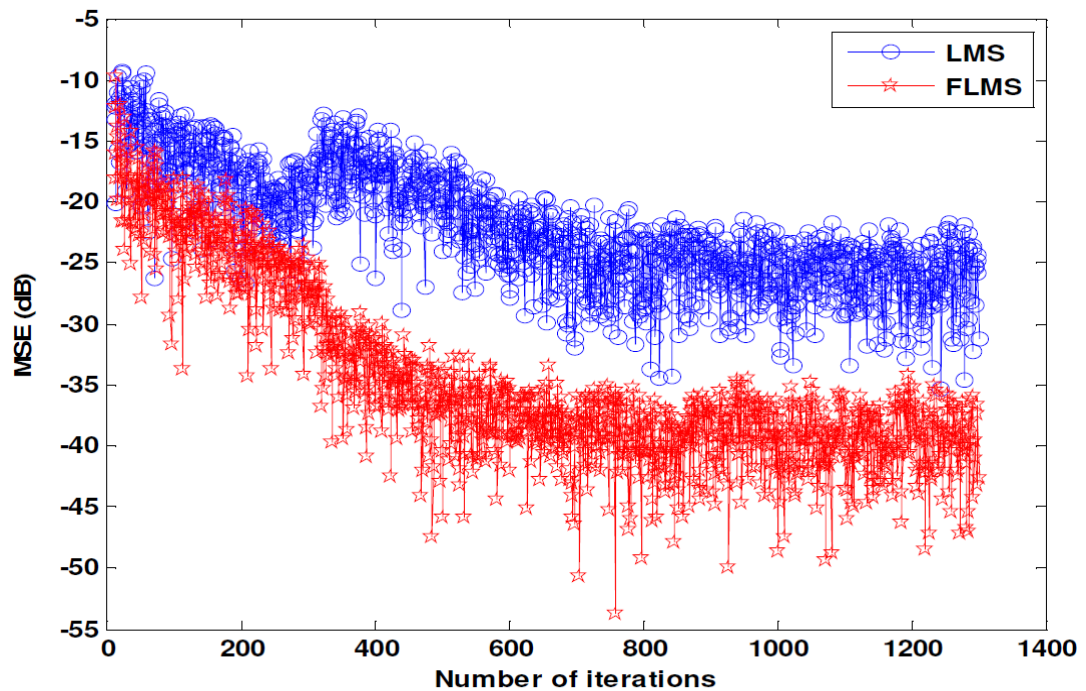


FIGURE 4.11: MSE for 300 Training Symbols of LMS and FLMS.

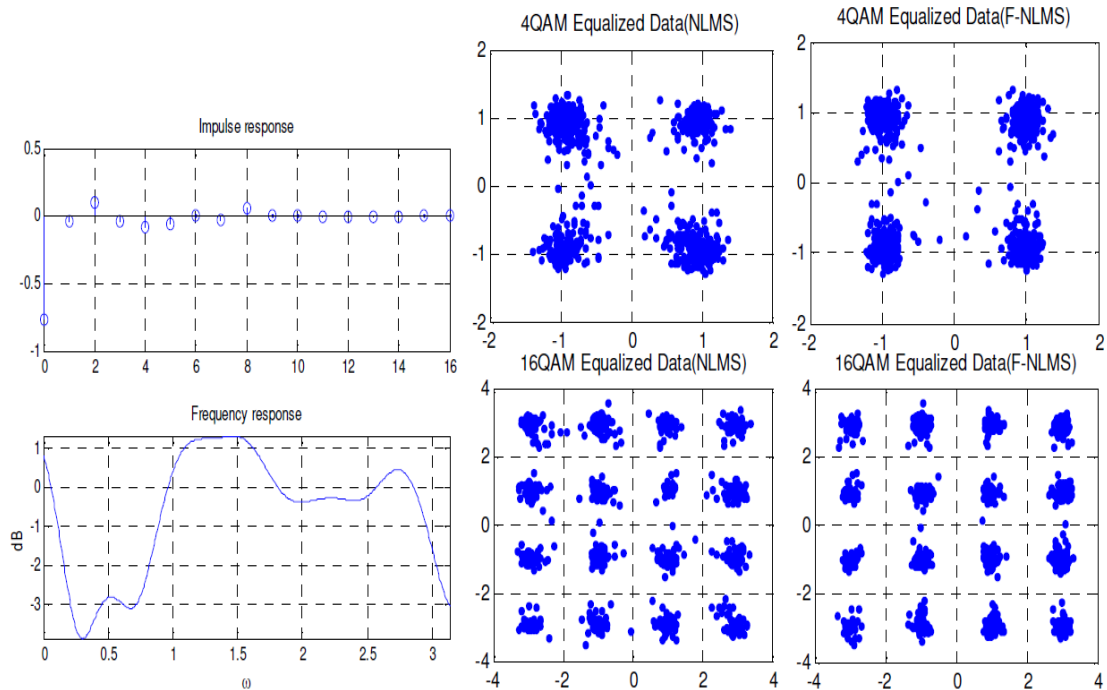


FIGURE 4.12: Scatter Plots of 16 & 64 QAM constellations Equalized with NLMS, F-NLMS.

total of 200 independent Monte Carlo runs were performed. Again it can be seen that the FNLMS is able to separate the symbols better than the NLMS algorithm, especially for 100 training symbols. It is seen that 573, 60 and 12 errors corresponding to 100, 200 and 300 training symbols respectively for the LMS while 8, 0 and 0 errors for the FNLMS algorithm. Figure 4.11 shows the corresponding MSE plot for both schemes, it can be seen that the MSE of the FLMS is much smaller than the LMS for the case of 300 training symbols.

Figure 4.12 (left) shows another frequency selective fading channel having 17 taps with delay $\Delta = 15$. For different step sizes and fractional orders of the FNLMS algorithm, the performance are evaluated for the channel. The constellation diagrams for equalized signal are shown. In Figure 4.13, QAM constellations are shown for a comparative analysis. The feed-forward filter has $N_f = 51$ taps; the feedback filter has $N_b = 1$ tap. Simulations results are generated for 50 independent iterations and 500 data symbols in the decision directed mode. The same number of training symbols have been used i.e. 100, 300, 400 and 600 for 4 QAM, 16 QAM, 64 QAM and 256 QAM respectively. It can be seen that FNLMS is able to isolate the symbols better than LMS. FNLMS has better performance in

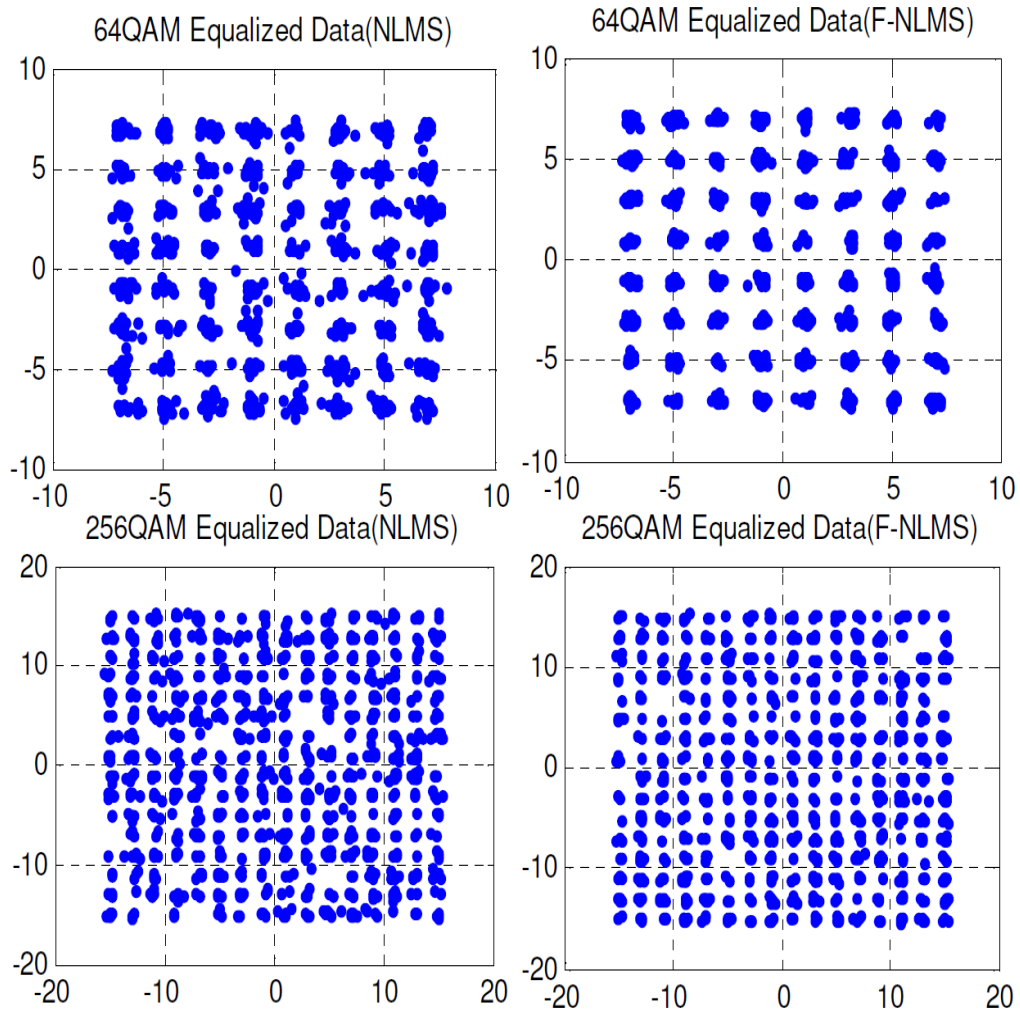


FIGURE 4.13: Scatter Plots of 64 & 256 QAM constellations Equalized with NLMS, F-NLMS.

decreasing the error variance. Large variations from desired symbol result in high uncertainty especially in higher order modulation schemes and more erroneous decisions by the demodulator.

Based on the simulation results provided in the form of symbol constellations through scatter plots, symbol error rate versus SNR curves, combined channel and equalizer responses, it can be observed that the proposed fractional order variant of the NLMS algorithm has superior performance than its standard counterpart. Meritoriously, the FNLMS has superior performance in decreasing the error and has stable steady state performance (as shown by the SER when the equalizer operates in decision directed mode). It can also be inferred that the FNLMS has faster convergence, otherwise the SER performance would have severely degraded.

4.4 Conclusions

In this chapter, fractional algorithms were proposed for both feedforward and decision feedback equalization by introducing fractional updates in both the LMS and NLMS algorithms. A filter structure was developed for the equalizer filter for the fractional variants of the LMS; the performances in terms of mean square error and symbol error rates of these fractional order techniques were compared with the standard LMS and NLMS algorithms. It is worth mentioning that there are three adaptation parameters in the proposed techniques, that is, two step sizes and the fractional order. The extra parameters facilitate to improve the convergence speed and steady state performance. The former helps in reducing the number of training symbols required for weight adaptation and the latter helps in reducing the symbol error rate performance.

Based on simulation results for the performance metrics of symbol error rate, mean squared error and combined channel and equalizer response, it is concluded that the fractional LMS and NLMS filters perform better than the standard LMS and normalized LMS algorithms. This is true for both flat frequency as well as frequency selective fading channels. Applying the new algorithms helps in significant improvement in the SNR of the order of $2 \sim 4$ dBs and increases the link efficiency by using less number of training symbols (a saving of about 60% training symbols) and minimizes the channel effects. The error rate performance evaluated considering fading channels and various parameters of the fractional versions; it was found that fractional algorithms provide better performance with SER of about 10^{-5} as compared to the traditional approaches which is the order of 10^{-3} . Further investigation of the fractional algorithms is required in other adaptive signal processing applications like system identification, beam-forming, active noise control, diffusion strategies in multi-agent systems, etc.

Chapter 5

Fractional Order Constant Modulus Blind Algorithms with Application to Channel Equalization

In this chapter, a novel methodology is developed for blind equalization where the output of the linear filter is passed through a nonlinear fractional update term derived from the cost function using fractional calculus. The final weight update is a combination of the conventional constant modulus algorithm (CMA) and the fractional part. The new fractional strategy helps capture the parameters of the model faster while keeping the error small. The algorithm is applied to blind equalization of flat and frequency selective channels. To assess the suitability of the proposed technique, different fractional orders and step sizes are used, the performance metric considered is mean squared error for a quadrature phase shift keying transmission scheme. Simulation results show that the proposed technique performs better than the conventional constant modulus algorithm, exhibits faster convergence, and yields an improved steady state response.

The presentation of the chapter is as follows: Firstly, the derivation and weight

updated mechanism of the fractional variants of the Godard algorithm are presented. In section 5.2, simulation results are shown for quadrature phase shift keying based modulations. Finally, conclusions are stated.

5.1 Fractional Order Blind Algorithms

Blind equalization techniques (in which the reference signal is not required) are very popular and especially useful for digital modulation [14, 136, 140], as the latter have known statistics which can be obtained from constellation used. One approach is based on variable tap-length filters [84], these can better model varying delays of frequency selective channels. The contribution of this chapter is to apply fractional order signal processing to improve the constant modulus algorithm (CMA-2). The cost function is based on the posterior error. Different fractional orders and step sizes are considered, and simulation results are shown for blind equalization of quadrature phase shift keying (QPSK) symbols in a multipath fading environment with mean squared error as the performance metric.

Consider the transmission of symbols drawn from a finite alphabet set \mathbf{s} through a channel with memory, modelled as an M^{th} order finite impulse response (FIR) filter represented by the vector $\mathbf{h} = [h(k), h(k-1) \dots h(k-M+1)]^T$; where $(\cdot)^T$ represents the vector transpose. The received signal is corrupted by the additive noise; the input regression vector is represented by $\mathbf{x} = [x(k), x(k-1) \dots x(k-N+1)]^T$ which carries the information of the present and previous N symbols that contribute to inter-symbol interference (ISI). To lessen the effect of ISI and separate the source symbol from noisy and distorted data, the design objective of the blind equalizer with N weights $\mathbf{w}(k) = [w_0(k), w_1(k) \dots w_{N-1}(k)]^T$; is to minimize the mean p-power generalized Godard cost function [14]:

$$\xi = E[(|\mathbf{w}^H(k)\mathbf{x}(k)|^q - \gamma_{\mathbf{q}})^p]. \quad (5.1)$$

where p and q are positive numbers and $p = q = 2$ is the CMA(2,2) case and $(\cdot)^H$ denotes Hermitian operator. The quantity $\mathbf{w}^H(k)\mathbf{x}(k)$ is the output $y(k)$ of

the equalizer filter, the constant γ_q is the constellation level to be achieved and is defined as:

$$\gamma_q = \frac{E[|s(k)|^{2q}]}{E[|s(k)|^q]}. \quad (5.2)$$

The error is the difference of output $y(k)$ and γ_q and is defined as:

$$e(k) = |\mathbf{w}^H(k)\mathbf{x}(k)|^q - \gamma_q. \quad (5.3)$$

A fractional constant modulus algorithm (FrCMA) is proposed having a configuration in which the final weights \mathbf{w} at sample index k are formed by combining the conventional CMA (Godard) weights \mathbf{w}_G and the fractional weight update \mathbf{w}_f such that:

$$\mathbf{w}(k) = \mathbf{w}_G(k) + \mathbf{w}_f(k). \quad (5.4)$$

In equation (5.4), $\mathbf{w}_G(k)$ is obtained by differentiating equation (5.3) with respect to the filter weights to obtain the gradient, the adaptation algorithm for the Godard (CMA) weights update is as given below [14, 140]:

$$\mathbf{w}_G(k) = \mathbf{w}(k-1) - \mu \left. \frac{\partial e^p(k)}{\partial \mathbf{w}} \right|_{\mathbf{w}_G = \mathbf{w}(k-1)}. \quad (5.5)$$

The gradient operation is performed on the instantaneous error, putting $e(k)$ from equation (5.3) in equation (5.5) results in:

$$\mathbf{w}_G(k) = \mathbf{w}(k-1) - \mu \frac{\partial (|y(k)|^q - \gamma_q)^p}{\partial \mathbf{w}}. \quad (5.6)$$

Applying the chain rule of integer derivatives, the gradient can be written as:

$$\frac{\partial e^p(k)}{\partial \mathbf{w}} = p (|y(k)|^q - \gamma_q)^{p-1} \frac{\partial e(k)}{\partial w}. \quad (5.7)$$

Putting for the error $e(k) = |y(k)|^q - \gamma_q$, this can be written as:

$$\frac{\partial e^p(k)}{\partial \mathbf{w}} = p (|y(k)|^q - \gamma_q)^{p-1} \frac{\partial (|y(k)|^q - \gamma_q)}{\partial w}. \quad (5.8)$$

which on further simplification becomes:

$$\frac{\partial e^p(k)}{\partial \mathbf{w}} = pq (|y(k)|^q - \gamma_q)^{p-1} |y(k)|^{q-1} \frac{\partial |y(k)|}{\partial \mathbf{w}}. \quad (5.9)$$

Since, $y(k) = \mathbf{w}^H(k)\mathbf{x}(k)$ is the output of the adaptive filter, whose weights are to be optimized, this partial derivative can be written as:

$$\frac{\partial |y(k)|}{\partial \mathbf{w}} = \mathbf{x}(k). \quad (5.10)$$

The equation (5.9) becomes:

$$\frac{\partial e^p(k)}{\partial \mathbf{w}} = pq (|y(k)|^q - \gamma_q)^{p-1} |y(k)|^{q-1} \mathbf{x}(k). \quad (5.11)$$

The term $|y(k)|^{q-1}$ can be set to $|y(k)|^{q-2}y^*(k)$, so the final update equation (5.6) for Gordad weights becomes:

$$\mathbf{w}_G(k) = \mathbf{w}(k-1) - \mu pq (|y(k)|^q - \gamma_q)^{p-1} |y(k)|^{q-2} y^*(k) \mathbf{x}(k). \quad (5.12)$$

where $(.)^*$ denotes the complex conjugate operation and μ is a positive step size. This can be written in terms of error $e(k)$ as:

$$\mathbf{w}_G(k) = \mathbf{w}(k-1) - \mu pq e^{p-1}(k) |y(k)|^{q-2} y^*(k) \mathbf{x}(k). \quad (5.13)$$

It is worth mentioning that $\mathbf{w}_G(k)$ relies on the final weights of the previous iteration updated through equation (5.4). The update part in equation (5.13) is derived from the gradient of the error measure $\varepsilon = J(w)$ such that $J(w + \Delta w) < J(w)$, where Δw is the change in weight from present iteration to the next one. Using Taylor series approximation, one arrives at the simplified LMS equation. Inspecting the update part of equation (5.13), the weights undergo large changes for steep gradients of error [127–130]. Conversely, for flat error surface, a small step size is required as the change in weights needs to be small [127]. Since each individual weight in the vector has a different value, the simplified gradient search mechanism in the LMS cannot provide the required intelligence, that is,

in the standard LMS, the output of the filter is more dependent on the weight(s) having larger values and does not provide significance to other weights [130]. This can be overcome to some extent by differentiating the cost function with respect to the weight vector using fractional derivatives [129]. To improve stability and convergence, a fractional update part is introduced as given in equation (5.4). The fractional derivative operator D^ν is used which for a function $f(x) = x^\beta$ is defined as [9, 42, 44, 108]:

$$D^\nu x^\beta = \frac{\Gamma(\beta + 1)}{\Gamma(\beta - \nu + 1)} x^{\beta - \nu}. \quad (5.14)$$

Here, $0 < \nu < 1$ is the fractional order and Γ is the gamma function [42, 44]. For $\beta - \nu + 1 > 0$, this is computed as: $\Gamma(\beta - \nu + 1) = \int_0^\infty e^{-t} t^{\beta - \nu} dt$. To obtain the fractional update part, the cost function is differentiated (first order derivative as given in equation (5.3)) with respect to the weights and fractional derivative of the last term in (5.9):

$$\mathbf{w}_f(k) = -\mu_f D^\nu \left(|\mathbf{w}^H(k) \mathbf{x}(k)|^q - \gamma_{\mathbf{q}} \right)^p \Bigg|_{\mathbf{w}=\mathbf{w}_G(k)}. \quad (5.15)$$

where μ_f is the positive step size to be chosen for the fractional update part. Taking the fractional derivative of $e^p(k)$ with respect to the filter weights \mathbf{w} and applying the chain rule approximation, the developed form is:

$$\mathbf{w}_f(k) = -\mu_f pq (|y_u(k)|^q - \gamma_q)^{p-1} |y_u(k)|^{q-2} y_u^*(k) \mathbf{x}(k) \odot \frac{\mathbf{w}_G^{(1-\nu)}(k)}{\Gamma(2-\nu)}. \quad (5.16)$$

Where \odot denotes element-wise multiplication of two vectors and $y_u(k) = \mathbf{w}_G^H(k) \mathbf{x}(k)$ is the output from the CMA stage and is computed before equation (5.16). Based on values of positive integer parameters p , q and the fractional order ν , variants of FrCMA can be generated. For $p=2$, $q=1$, equation (5.12) can be simplified to the following form:

$$\mathbf{w}_G(k) = \mathbf{w}(k-1) + \mu \frac{y_u^*(k)}{|y_u(k) + \epsilon|} (|y_u(k)| - \gamma_1) \mathbf{x}(k). \quad (5.17)$$

TABLE 5.1: FrCMA Based Adaptation Procedure.

$$y(k) = \mathbf{w}^H(k-1)\mathbf{x}(k)$$

$$\mathbf{w}_G(k) = \mathbf{w}(k-1) + \mu y^*(k)(|y(k)|^2 - \gamma_2)\mathbf{x}(k)$$

$$y_u(k) = \mathbf{w}_G^H(k)\mathbf{x}(k)$$

$$\mathbf{w}_f(k) = -\mu_f y_u^*(k)(|y_u(k)|^2 - \gamma_2)\mathbf{x}(k) \odot \frac{\mathbf{w}_G^{(1-\nu)}(k)}{\Gamma(2-\nu)}$$

$$\mathbf{w}(k) = \mathbf{w}_G(k) + \mathbf{w}_f(k)$$

The corresponding fractional update becomes:

$$\mathbf{w}_f(k) = -\mu_f \frac{y_u^*(k)}{|y_u(k) + \epsilon|} (|y_u(k)| - \gamma_1)\mathbf{x}(k) \odot \frac{\mathbf{w}_G^{(1-\nu)}(k)}{\Gamma(2-\nu)}. \quad (5.18)$$

where ϵ is a some small positive number. As discussed in chapter four, in implementation for the general case where the weights may have positive or negative values, equation (5.18) can be written as follows:

$$\mathbf{w}_f(k) = -\mu_f \frac{y_u^*(k)}{|y_u(k) + \epsilon|} (|y_u(k)| - \gamma_1)\mathbf{x}(k) \odot \frac{|\mathbf{w}_G(k)|^{(1-\nu)} \text{sgn}(\mathbf{w}_G(k))}{\Gamma(2-\nu)}. \quad (5.19)$$

For FrCMA (2, 2), the main adaptation procedure is summarized in Table: 5.1. The sequence of execution starts from the traditional CMA, the posterior output $\mathbf{w}_G^H(k)\mathbf{x}(k)$ of the first stage is calculated and used in the fractional update stage. It may be noted that the conventional update uses the final updated weights \mathbf{w} of the previous iteration; the fractional update term depends on the in-process CMA weights and the posterior output $y_u(k)$. Both terms are added to obtain the final weights.

Figure 5.1 shows plots of the term $\Delta f = \frac{w^{(1-\nu)}}{\Gamma(2-\nu)}$ against a scalar w to show the effect of varying the fractional order ν . It can be seen that for $\nu = 0$, the term is linear in w , however, increasing the fractional order results in a larger 'gain' for small weight values. In the plot, the effect of different fractional orders can be seen when the weight values are changed from 0 to 1; the nonlinear effect can be noted. The fractional update part provides prominence to each individual weight

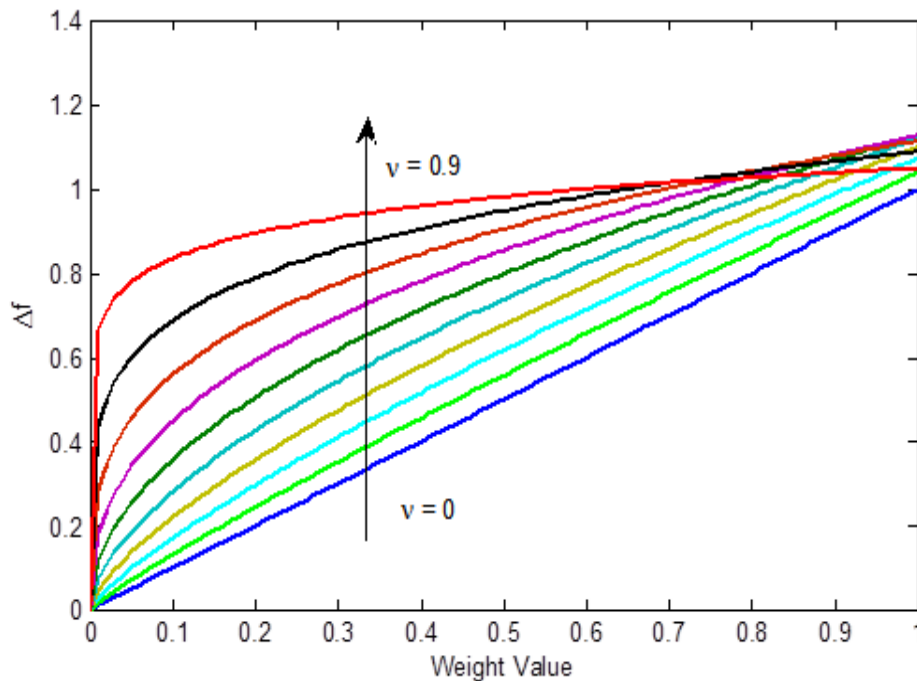


FIGURE 5.1: Effect of varying the fractional order on the term $\frac{w^{(1-\nu)}}{\Gamma(2-\nu)}$.

[127] based on its magnitude. This helps capture the required parameters at a faster rate and keeps the error small; which results in improved convergence.

5.2 Simulation Results

The transmission of symbols modulated through a quadrature phase shift keying scheme ($\gamma_1 = \sqrt{2}, \gamma_2 = 2$) is considered for the generation of simulation results. The results are provided for performance metric of mean squared error based on equation (5.1) for the cases when $p = 2$, and $q = 1, 2$ corresponding to standard and fractional order CMA (2, 1) and CMA (2, 2) respectively i.e.,

$$MSE = E[|y(k)|^q - \gamma_2]^2. \quad (5.20)$$

Two complex channels are considered, the first one (channel 1) is a three-tap case and exhibits an almost flat frequency response, the second (channel 2) is a fourteen-tap frequency selective fading channel having a deep null of -35 dB. The impulse response of channel 1 is $\mathbf{h} = [1.1 + 0.5i, 0.1 - 0.3i, -0.2 - 0.1i]$.

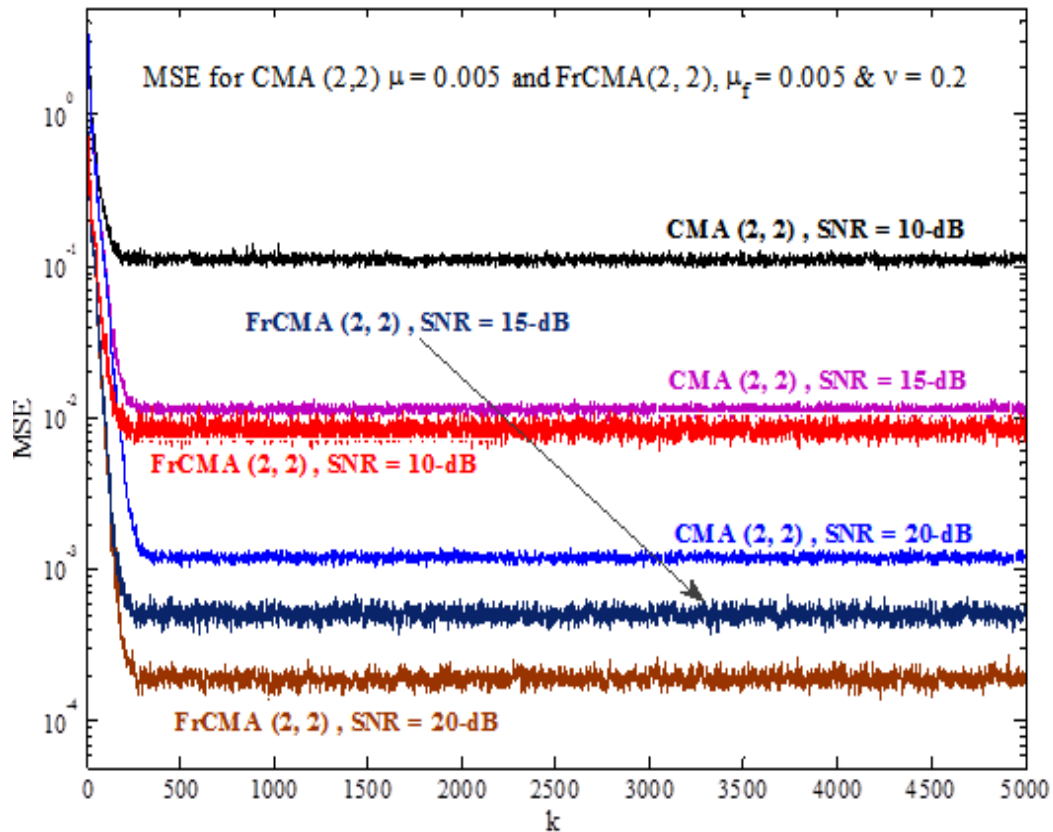


FIGURE 5.2: MSE Performance of CMA and FrCMA for channel 1.

The equalizer has 14 taps and a delay of 8 samples. The simulation results are generated for 5000 iterations and the mean squared error value is computed based on 1000 independent runs. Keeping the step size and fractional order constant ($\mu = \mu_f = 0.005, \nu = 0.2$), the MSE performance is calculated for different signal-to-noise ratios (SNRs). Results are plotted in Figure 5.2. It can be seen that FrCMA (2, 2) outperforms the conventional algorithm in terms of convergence and steady state performance. At iteration number 500 and for $SNR = [10, 15, 20]$ dB, the MSE of CMA is $[1.11 \times 10^{-1}, 1.16 \times 10^{-2}, 1.19 \times 10^{-3}]$, whereas for FrCMA, the corresponding MSE is $[7.48 \times 10^{-3}, 5.27 \times 10^{-4}, 1.64 \times 10^{-4}]$, respectively.

We now consider Equalizers with 45 taps having a delay of 25 samples for channel 2 with impulse response: $\mathbf{h} = [0.2231 - 0.1745i, -0.0077 + 0.1281i, 0.3312 + 0.4829i, 0.1703 + 0.0282i, -0.1024 + 0.1293i, 0.0743 - 0.0580i, 0.0070 - 0.0642i, 0.0340 - 0.0442i, -0.0191 + 0.0023i, 0.0060 - 0.0076i, 0.0035 + 0.0133i, -0.0015 - 0.0067i, 0.0092 - 0.0045i, -0.0022 - 0.0003i]$.

The results are generated for 5000 iterations and 1000 independent runs, the SNR

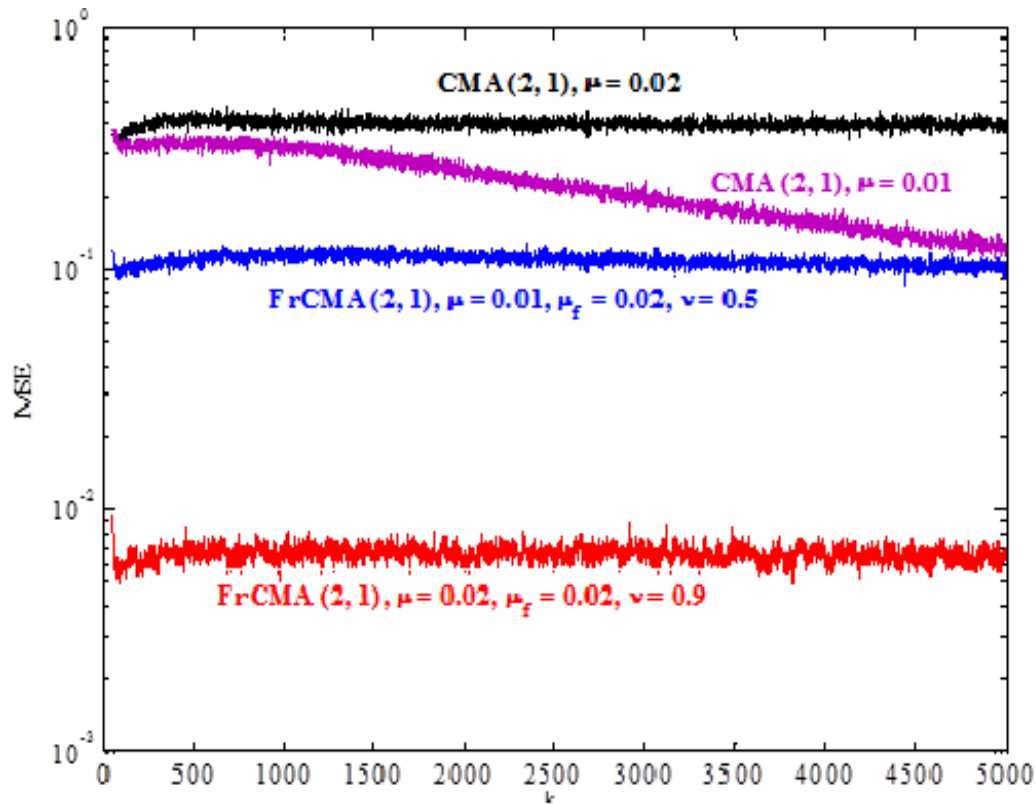


FIGURE 5.3: MSE Performance of CMA and FrCMA for channel 2 for different step sizes and fractional orders.

is fixed at 30 dB, $p=2$, $q=1$ and step sizes are varied. Results are shown in Figure 5.3. It is seen that the proposed fractional variant has faster convergence and smaller MSE for the given step sizes and fractional orders. For a step size of $\mu = 0.02$, the standard CMA (2, 1) algorithm has an MSE of 0.43, and limited convergence, whereas the fractional variant with $\nu = 0.9$ has a much better MSE of about 6×10^{-3} after 50 iterations, showing a very fast convergence. With a step size of 0.01, the CMA (2, 1) converges quite slowly to a lower MSE, but the FrCMA (2, 1) clearly outperforms the conventional CMA. These results indicate that the proposed fractional variants of CMA, where the posterior output is passed through the nonlinear adaptive stage improves convergence.

5.3 Conclusions

In this chapter, fractional order generalized blind algorithms were proposed for feedforward equalization. The new blind adaptive algorithms which exploits fractional derivatives in minimizing the Godard cost function is developed and applied to the inverse system identification problem of channel equalization. This is achieved by introducing fractional updates based on posterior output (implicit error based on the statistical parameter of modulus). The performance was measured in terms of mean squared error for different fractional orders, step sizes and channel types. It is again worth mentioning that the proposed algorithms have three adaptation parameters, that is, two step sizes and a fractional order. The proposed algorithm has 50-60% faster convergence when the steady state error of CMA is considered and improved mean squared error performance of the order of 100 times as compared to traditional CMA. It also achieves enhanced equalization and better MSE performance. For larger step sizes, the conventional CMA algorithm gives a poorer MSE and may also diverge, whereas the proposed algorithm shows stable steady state behavior while also alleviating the noise effects better than its traditional counterpart. FrCMA is, therefore, a useful addition in fractional order signal processing and is open to further research. Further investigation of the fractional algorithms is required in other adaptive signal processing applications like system identification, beamforming, multi-input multi output systems, active noise control, diffusion strategies in multi-agent systems, tracking of fading channels and so on.

Chapter 6

Fractional Order Adaptive Signal Processing Strategies for Active Noise Control Systems

Robust and computationally efficient adaptive algorithms are required in Active Noise Control Systems (ANCS) to cancel out the effects of noise in the presence of Secondary Path (SP) as the later makes the identification problem more challenging. To ensure stability in such applications, the step size parameter is kept small but it results in slow convergence which limits the usefulness of such algorithms in ANCS. In this chapter, two novel fractional order adaptive strategies are proposed such that the output from the conventional Filtered-x Least Mean Square (FxLMS) algorithm is passed through a new update equation derived from a cost function based on a-posteriori error and optimized using fractional derivatives. The proposed algorithms are designed for the feed-forward configuration of ANCS. The new schemes are validated using the performance metrics of mean squared error, mean squared deviation and mean relative modelling error. A number of scenarios are considered where different step sizes and fractional orders have been used for evaluation with input signals modeled as binary, Gaussian and impulsive noise sources. Simulation results shows that the proposed algorithms outperform the conventional counterparts with convergence improvements in the

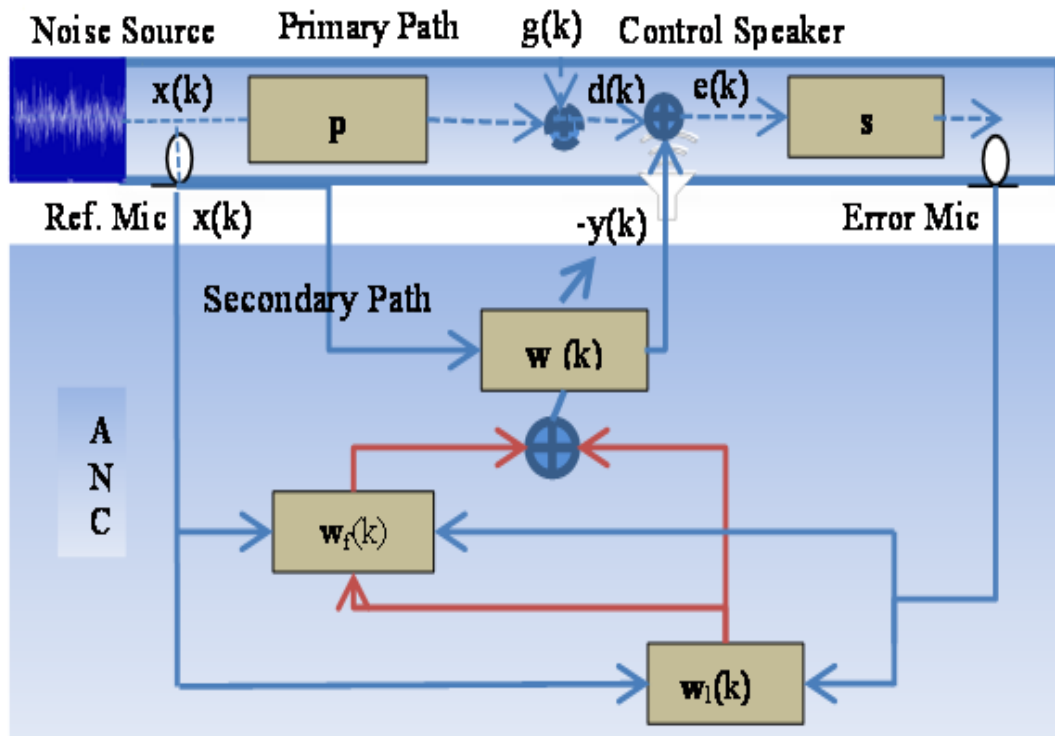


FIGURE 6.1: Schematic of ANC system for machine noise suppression in a channel using FN-FeLMS in a feedforward configuration.

range of 75-80%, while offer the same steady state behavior even for large step sizes; thereby provide better modeling in the presence of SP.

The presentation in the rest of the chapter is as follows: Section 6.1 presents the Fractional Normalized Filtered-error LMS (FN-FeLMS) algorithm. Section 6.2 presents modifications of FN-FeLMS algorithm and other FxLMS fractional order variants. Section 6.3 provides discussion about the computational complexity of the proposed algorithms along with comparative analysis. Simulation results and analysis for various fractional orders, different input noise sources and step sizes are given in Section 6.4. Conclusions are given in Section 6.5.

6.1 The FN-FeLMS Algorithm

A schematic of the FN-FeLMS based ANC system is shown in Figure 6.1. The noisy input signal $x(k)$ is passed through the primary path having impulse response $\mathbf{p} = [p_0, p_1 \dots p_{M-1}]^T$, its output $d(k)$ is corrupted by zero mean white Gaussian

noise $g(k)$. The objective is to design an adaptive filter $\mathbf{w} = [w_0, w_1 \dots w_{N-1}]^T$ such that its output $y(k)$ when fed to the control speaker generate an anti-noise signal and create a silence zone at the error microphone. The error $e(k)$ is the difference between $d(k)$ and the secondary path signal passed through the adaptive filter. The fixed filter $\mathbf{s} = [s_0, s_1 \dots s_{Q-1}]^T$ is used to filter the error. In the standard LMS algorithm, the cost function $J(k) = E[e^2(k)]^T$ is minimized by differentiating with respect to the filter weights \mathbf{w} . In the scheme proposed here, the weights \mathbf{w} at sample k are updated as:

$$\mathbf{w}(k) = \mathbf{w}_l(k) + \mathbf{w}_f(k). \quad (6.1)$$

where $\mathbf{w}_l(k)$ is the standard FeLMS weight adaptation vector and the term $\Delta\mathbf{w}_f(k)$ represents the weight update term corresponding to the fractional order (FO). With μ_l denoting the step size and $\mathbf{x}(k) = [x(k), x(k-1) \dots x(k-N+1)]^T$ as the input regression vector, the modified FeLMS equation is [33]:

$$\mathbf{w}_l(k) = \mathbf{w}(k) + \mu_l \mathbf{x}(k) \mathbf{s} [d(k) - \mathbf{w}^T(k-1) \mathbf{x}(k)]. \quad (6.2)$$

The term $\mathbf{s}[d(k) - \mathbf{w}^T(k-1) \mathbf{x}(k)] = \mathbf{s}[e(k)]$ represents filtering of the error by the fixed filter \mathbf{s} . The fractional correction term is calculated using the fractional operator [6, 42]. The fractional derivative of order ν of the cost function $J_w(k)$ is applied to obtain the weight update term is:

$$\mathbf{w}_f(k) = -\mu_f D^\nu e^2(k) \Big|_{\mathbf{w}=\mathbf{w}_l(k)}. \quad (6.3)$$

where μ_f is the step size for the fractional part adaptation. For $x > 0$, the R-L based differentiation of fractional order $\nu > -1$ of a function $f(x) = x^\beta$ is defined as [42]:

$$D^\nu x^\beta = \frac{\Gamma(\beta + 1)}{\Gamma(\beta - \nu + 1)} x^{\beta-\nu}. \quad (6.4)$$

Here Γ is the gamma function; for $\beta - \nu + 1 > 0$, it is computed as [42–44]:

$$\Gamma(\beta - \nu + 1) = \int_0^\infty e^{-t} t^{\beta-\nu} dt. \quad (6.5)$$

For $\nu \geq 0$, the operation in equation (6.4) is differentiation and for $\nu < 0$ it becomes integration. In equation (6.3), using the chain rule approximation and applying (6.4), the final fractional correction term at a given iteration can be written as:

$$\mathbf{w}_f(k) = \mu_f \mathbf{x}(k) \mathbf{s} [d(k) - \mathbf{w}^T(k-1) \mathbf{x}(k)] \odot \frac{\mathbf{w}_l^{(1-\nu)}(k)}{\Gamma(2-\nu)}. \quad (6.6)$$

where \odot denotes element-wise multiplication. For traditional LMS, a time-varying optimized step size is chosen for stability and performance and is given as [33, 34, 141]:

$$\mu_l(k) = \frac{\mu_l}{\|\mathbf{x}(k)\|^2}. \quad (6.7)$$

Similarly, as discussed in chapter four, the step size can be adapted for the fractional update part using fractional derivative in the objective function $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subject to the constraint: $d(k) = \mathbf{u}(k) \mathbf{w}(k+1)$. The step size parameter can be adjusted according to the FO as follows:

$$\mu_f(k) = \frac{\Gamma(3-\nu)}{\Gamma(2-\nu)\Gamma(3)} \frac{\mu_f}{\|\mathbf{x}(k)\|^2}. \quad (6.8)$$

As already discussed in, in implementation for the general case where weights may have positive or negative values, the equation $w^{1-\nu}(k) = |w(k)|^{1-\nu} \text{sgn}(w(k))$ can be applied. Using equations (6.2), (6.6), (6.7) and (6.8), the sequence of execution of weight update for the FN-FeLMS algorithm is summarized in Table 6.1. The result is a semi-cascade structure such that the FeLMS weight update uses the weight vector of the previous step, whereas the fractional correction term uses the in process FeLMS weights.

TABLE 6.1: Summary of ANC Algorithm for fractional normalized FeLMS.

$$\mathbf{w}_l(k) = \mathbf{w}(k-1) + \mu_l \frac{\mathbf{x}(k)\mathbf{s} [d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)]}{\|\mathbf{x}(k)\|^2 + \epsilon}$$

$$\mathbf{w}_f(k) = \mu_f \frac{\Gamma(\mathbf{3} - \nu)}{\Gamma(\mathbf{2} - \nu)} \frac{\mathbf{x}(k)\mathbf{s} [d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)]}{\|\mathbf{x}(k)\|^2 + \epsilon} \odot \frac{|\mathbf{w}_l(k)|^{(1-\nu)} \mathbf{sgn}(\mathbf{w}_l(k))}{\Gamma(\mathbf{2} - \nu)}$$

$$\mathbf{w}(k) = \mathbf{w}_l(k) + \mathbf{w}_f(k)$$

6.2 Modified Fractional FeLMS and its Variants

In this section, the techniques presented in previous section are further extended to achieve an improved Fractional Order (FO) variant of the FN-FeLMS. In both conventional FeLMS and the FN-FeLMS algorithms, the SP filter is applied on the errors $[d(k) - \mathbf{w}^T(\mathbf{k}-1)\mathbf{x}(\mathbf{k})]$ which is the filtered error instead of the filtered input. In previous section, it is shown that the FN-FeLMS algorithm achieves faster convergence than FeLMS; the final weights are updated through a combination of conventional and fractional update terms. However, the steady-state performance degrades when the fractional order is increased [67] and this is shown with simulation results in section 6.4. The approach provided in [67] is based on calculating the gradient using fractional derivatives of the *a-priori* error $e(k) = [d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)]$. To attain faster convergence and improve the steady-state response, the strategy in [67] is modified by introducing a second cost function $\xi_w(k)$ based on the *a-posteriori* error $e_p(k)$. The fractional correction term is obtained by taking the fractional derivative D^ν with respect to weights of the conventional update as in (6.3-6.8) as follows:

$$\mathbf{w}_f(k) = -\mu_f D^\nu \xi_w(k) \Big|_{\mathbf{w}=\mathbf{w}_l(k)} . \tag{6.9}$$

The final weights are obtained from combination of the weights updated through the conventional and fractional terms, i.e., $\mathbf{w}_l(k) + \mathbf{w}_f(k)$. The cost function relies on conventional weights and is defined as:

$$\xi_w(k) = e_p^2(k) = (d(k) - \mathbf{w}_l^T(k)\mathbf{x}(k))^2 . \tag{6.10}$$

Note that the *a-posteriori* error is based on the in-process available weights \mathbf{w}_l . Using the same approach, we also develop FO variants of other versions such as Fractional Normalized LMS, (FrNLMS), Fractional FxLMS (FrFxLMS), Fractional Modified FxLMS (FrMFxLMS) and Fractional FeLMS (FrFeLMS). Using the chain rule approximation for the fractional derivative, the expanded form can be written as:

$$D^\nu e_p(k) = \mu_f \frac{\Gamma(2)}{\Gamma(2-\nu)} \mathbf{x}(k) \odot \mathbf{w}_l^{1-\nu}(k). \quad (6.11)$$

To attain a normalized LMS update, the objective function $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ is minimized subject to the constraint that the posterior error is zero, that is, $d(k) - \mathbf{w}^T(k)\mathbf{x}(k) = 0$. For integer order case, the optimized step size in a given iteration is again stated as [34, 67]:

$$\mu_l(k) = \frac{\mu_l}{\|\mathbf{x}(k)\|^2 + \epsilon}. \quad (6.12)$$

As discussed in chapter four and the procedure for (6.8) above for the FN-FeLMS algorithm, the following equation is obtained for the time varying step size which also depends on the fractional order [67] as:

$$\mu_f(k) = \mu_f \frac{\Gamma(3-\nu)}{\Gamma(2-\nu)\Gamma(3)} \frac{\mu_f}{\|\mathbf{x}(k)\|^2 + \epsilon}. \quad (6.13)$$

The resulting fractional order update term for the normalized variant becomes:

$$\mathbf{w}_f(k) = \mu_f G \frac{e_p(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2 + \epsilon} \odot \mathbf{w}_l^{1-\nu}(k). \quad (6.14)$$

Here G is a new constant based on the Gamma function and for a given fractional order is calculated as:

$$G = \frac{\Gamma(3-\nu)}{\Gamma(2-\nu)^2}. \quad (6.15)$$

By combining equations (6.14) and (6.15), and using a small positive constant ϵ to avoid the denominator becoming too small, the time varying step size for the

TABLE 6.2: FrFeLMS Based ANC.

$e(k) = d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)$	(A)
$e_F(k) = \mathbf{s}^T \mathbf{e}(k)$	(B)
$\mathbf{w}_l(k) = \mathbf{w}(k-1) + \mu_l e_F(k) \frac{\mathbf{x}(k)}{\ \mathbf{x}(k)\ ^2 + \epsilon}$	(C)
$e_p(k) = d(k) - \mathbf{w}_l^T(k)\mathbf{x}(k)$	(D)
$\mathbf{w}_f(k) = \mu_f \frac{\Gamma(3-\nu)}{\Gamma(2-\nu)} e_p(k) \left(\frac{\mathbf{x}(k)}{\ \mathbf{x}(k)\ ^2 + \epsilon} \odot \frac{ \mathbf{w}_l(k) ^{(1-\nu)} \mathbf{sgn}(\mathbf{w}_l(k))}{\Gamma(2-\nu)} \right)$	(E)
$\mathbf{w}(k) = \mathbf{w}_l(k) + \mathbf{w}_f(k)$	(F)

fractional part can be further simplified to [67]:

$$\mu_f(k) = G \frac{\mu_f}{\|\mathbf{x}(k)\|^2 + \epsilon}. \quad (6.16)$$

Finally the conventional and fractional update terms are combined to yield the desired weights, which can be written in a simplified form as:

$$\mathbf{w}(k) = \mathbf{w}_l(k) + \mu_f(k) e_p(k) \mathbf{x}(k) \odot \mathbf{w}_l^{1-\nu}(k). \quad (6.17)$$

In implementation for the general case where weights may have positive or negative values, equation (6.17) can be written as below:

$$\mathbf{w}(k) = \mathbf{w}_l(k) + \mu_f(k) e_p(k) \mathbf{x}(k) \odot |\mathbf{w}_l(k)|^{(1-\nu)} \mathbf{sgn}(\mathbf{w}(k)). \quad (6.18)$$

Table 6.2 summarizes the steps for the modified version of this algorithm called as the FrFeLMS. The novelty here is the introduction of the *a-posteriori* error calculation (D) after the conventional stage and then its use in the fractional update part (E). Also the fractional part relies on the conventional part updated in (B) and the conventional part relies on the final weights of the previous iteration updated through (E).

Table 6.3 summarizes the steps for the fractional variant of NLMS, that is, FrNLMS.

TABLE 6.3: FrNLMS Based ANC.

$e(k) = d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)$	(A)
$\mathbf{w}_l(k) = \mathbf{w}(k-1) + \mu_l(k)e(k)\mathbf{x}(k)$	(B)
$e_p(k) = d(k) - \mathbf{w}_l^T(k)\mathbf{x}(k)$	(C)
$\mathbf{w}_f(k) = \mu_f(k)e_p(k)\mathbf{x}(k) \odot \mathbf{w}_l(k) ^{(1-\nu)}\mathbf{sgn}(\mathbf{w}_l(k))$	(D)
$\mathbf{w}(k) = \mathbf{w}_l(k) + \mathbf{w}_f(k)$	(E)

Only the main body of NLMS is shown here. A fixed fractional order case is considered so as to initialize G before the iterative process. It is assumed that the error $e(k)$, G and step sizes μ_l and μ_f are pre-computed as in equations (6.12), (6.13) and (6.15). Equation (E) combines the weights from the conventional and fractional parts of the algorithm. The fractional part uses the weights of the conventional update whereas the LMS relies on the final weights of the previous step. The FrNLMS algorithm is given as a benchmark algorithm; it does not incorporate the effects of SP and therefore has a faster convergence. However, the performance is poor when either the error signal is filtered or the input is filtered through the SP. For the NLMS or FrNLMS algorithms $F = \delta(k)$ which corresponds to an ideal discrete time impulse of unit magnitude.

Table 6.4 gives equations for the FrFxLMS algorithm. Note that the first two equations (A) and (B) are used in the conventional FxLMS algorithm; the last three equations are the modifications due to FrFxLMS. The fractional update part not only relies on the posterior error and the filtered input but also on the weights updated through the conventional part (B). The final weights are constructed from the conventional and the fractional parts. The fractional term $Gw^{1-\nu}$ behaves in a nonlinear fashion and therefore gives importance to individual weights according to the fractional order. Increasing the fractional order emphasizes the small weight values more as compared to larger values. The behavior of G is also a decreasing function of fractional order; use of the posterior error helps in improving convergence.

TABLE 6.4: FrFxLMS Based ANC.

$e(k) = d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)$	(A)
$\mathbf{w}_l(k) = \mathbf{w}(k-1) + \mu_l(k)e_F(k)\mathbf{x}_F^T(k)$	(B)
$e_p(k) = d(k) - \mathbf{w}_l^T(k)\mathbf{x}(k)$	(C)
$\mathbf{w}_f(k) = \mu_f(k)e_p(k)\mathbf{x}_F^T(k) \odot \mathbf{w}_l(k) ^{(1-\nu)}\mathbf{sgn}(\mathbf{w}_l(k))$	(D)
$\mathbf{w}(k) = \mathbf{w}_l(k) + \mathbf{w}_f(k)$	(E)

TABLE 6.5: FrMFxLMS Based ANC.

$e(k) = d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)$	(A)
$\mathbf{w}_l(k) = \mathbf{w}(k-1) + \mu_l(k)\mathbf{x}_F^T(k)\left(e_F(k) + y_F(k) - \tilde{\mathbf{y}}_F(k)\right)$	(B)
$e_p(k) = d(k) - \mathbf{w}_l^T(k)\mathbf{x}(k)$	(C)
$\tilde{\mathbf{y}}_{F(p)}(k) = \mathbf{w}_l^T(k)\mathbf{x}_F(k)$	(D)
$\mathbf{w}(k) = \mathbf{w}_l(k) + \mu_f(k)\left(e_F(k) + y_F(k) - \tilde{\mathbf{y}}_{F(p)}(k)\right)\left(\mathbf{x}_F^T(k) \odot \mathbf{w}_l(k) ^{(1-\nu)}\mathbf{sgn}(\mathbf{w}_l(k))\right)$	(E)

Table 6.5 gives a more detailed form for the fractional variant of MFxLMS which is called the FrMFxLMS algorithm. The final update is a function of the conventional and fractional correction parts which is the second term on the right hand side of (E). It can be seen that in addition to the posterior error (C), the fractional update part is also dependent on the posterior filtered output (D) which is based on the conventional weights updated through (B). The conventional update only relies on the final weights of the previous iteration for its adaptation. It can also be noted that both the conventional and fractional terms use the same filtered input regression vector. The filtered error $e_F(k) = \sum_{q=0}^{Q-1} s_q(k)e(k-q)$ the filtered output $y_F(k) = \sum_{q=0}^{Q-1} s_q(k)y(k-q)$ and the filtered output due to filtered input $\hat{y}_F(k) = \mathbf{w}^T(k-1)\mathbf{x}_F(k)$ are pre-computed before the conventional update (B). For fractional update the posterior filtered output $\hat{y}_{F(p)}(k)$ due to filtered input is based on the in-process weights updated through (B), that is, $\hat{y}_F(k) = \mathbf{w}_l^T(k)\mathbf{x}_F(k)$.

TABLE 6.6: Computational complexity of various algorithms.

Algorithm	Multiplications	Additions	Fractional Powers
NLMS	$2M+1$	$2M+1$	0
FrNLMS	$5M+2$	$4M+1$	M
FxLMS	$2M+2Q+1$	$2M+2Q-2$	0
FrFxLMS	$5M+2Q+2$	$4M+2Q-2$	M
MFxLMS	$2M+3Q+1$	$2M+3Q-1$	0
FrMFxLMS	$5M+5Q+2$	$5M+5Q+1$	M
FeLMS	$2M+2Q+1$	$2M+Q-1$	0
FrFeLMS	$4M+Q+2$	$4M+Q-1$	M
FN-FeLMS	$3M+Q+1$	$3M+Q$	M

It can be noted that in all the fractional order variants, the instantaneous error is based on the weights of the previous iteration which are updated through both conventional and fractional parts. The conventional filtered LMS (all variants) relies on these weights in conventional adaptation; a virtual output which does not generate any signal for the control speaker but is used internally in the algorithm is calculated and based on this output a new virtual or posterior error (since, it is not observed through any sensor or microphone) is calculated. The fractional term relies on this posterior error for its weight adaptation equation. In all the algorithms, the step size is a function of only the input signal energy, that is, $\frac{1}{\|x\|^2}$ and also depends on the fractional order in fractional variants.

6.3 Computational Complexity

The computational complexity of most algorithms is determined using the number of additions/subtractions and multiplications/divisions required per iteration [81, 142]. A new computational term is introduced for fractional power calculation of a real or complex number. The detailed computational complexity analysis of the proposed algorithms is shown in Table 6.6; the per-iteration complexity is

almost double in terms of the traditional computation requirements of additions and subtractions. It is assumed that there are M adjustable weights or taps, and the SP is modeled by the filter \mathbf{s} with Q taps, in general it is assumed that $M \gg Q$. There are M computations required for calculating the fractional power since there are M weights. The complexity further increases if a variable fractional order case is considered as the new parameter is based on Gamma functions of different orders.

Here a fixed order case is considered. In the simulation results shown in the next section, very fast convergence is achieved using the newly proposed algorithms, the overall computations required to achieve the desired model accuracy are much less than the traditional algorithms. For example considering a 10-tap case, the FrFeLMS achieves -60 dB MSD in 200 iterations while its conventional counterpart takes 2300 iterations, there are 59800 operations required in FeLMS while for FrFeLMS 13800 operations are required. It is observed that not only get faster convergence but also less computation to get the desired response. An option could be to use fractional variants in the start to achieve faster convergence and then use conventional algorithms in the steady-state to reduce computational load.

6.4 Simulation Results and Analysis

Extensive computer simulations are performed using different inputs, step-sizes and fractional orders, the PP output is corrupted by independent and identically distributed (i.i.d.) additive zero mean white Gaussian noise $g(k)$; in almost all cases its variance is set at 60 dB below the input signal power level. Although the step sizes μ_l and μ_f can be different, their values are kept equal for clear comparison. Reliability and accuracy of the proposed algorithms are assessed in terms of three parameters. The first is the mean squared error (MSE) calculated

as:

$$\begin{aligned} MSE(dB) &= 10 \log_{10} E[e^2(k)] \\ &= 10 \log_{10} E \left[\left(d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k) \right)^2 \right] \end{aligned} \quad (6.19)$$

The second performance metric is the mean squared deviation (MSD) which is defined as:

$$MSD(dB) = 10 \log_{10} E[\|\mathbf{w}_o - \mathbf{w}(k)\|^2] \quad (6.20)$$

The third performance metric which is very important in ANCS applications is the relative modeling error and is defined as:

$$\Delta S(dB) = 10 \log_{10} E \left[\frac{\|\mathbf{w}_o - \mathbf{w}(k)\|^2}{\|\mathbf{w}_o\|^2} \right] \quad (6.21)$$

where the vector \mathbf{w}_o represents the optimal weights. Four different primary path models of different orders are considered for the simulations. The first one is a basic model [30–33] having 10-taps and the rest are generated from the convolution operation. The second model is a 19-tap case and is represented by P_1 which is given below:

$$P_1(k) = \sum_{i=0}^{2M-1} p_i(k)p_i(k-i) \quad (6.22)$$

The third model is the convolution of the basic and the second models and has 28-taps. It is given by:

$$P_2(k) = \sum_{i=0}^{3M-2} p_i(k)p_1(k-i) \quad (6.23)$$

The last model is a 37-tap case and is derived from the basic model as follows:

$$P_3(k) = \sum_{i=0}^{4M-3} p_i(k)p_2(k-i) \quad (6.24)$$

The normalized frequency responses of the different models for the primary paths

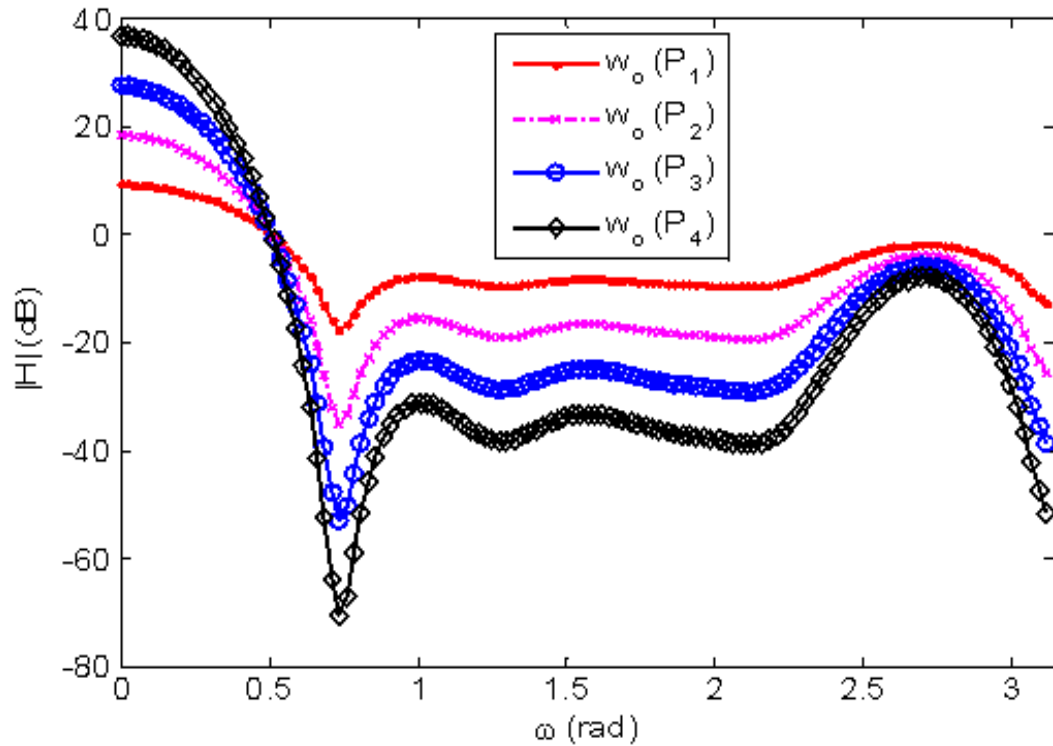


FIGURE 6.2: Normalized frequency responses of the different primary path models.

are shown in Figure 6.2. In the simulations, a fixed error path finite impulse response (FIR) filter is considered [30–33] with response:

$$S(z) = 1 - 1.2z^{-1} + 0.72z^{-2} \quad (6.25)$$

Simulation results are also provided for impulsive noise sources which are modelled as α -stable processes and are characterized by heavily tailed distribution having fractional lower order moments or statistics (FLOM/FLOS). There is no closed form expression for such density functions [143] but they have the following characteristic equation:

$$\varphi(x) = \exp \left(j\lambda x - \gamma|x|^\alpha [1 + j\beta \text{sign}(x)\omega(x, \alpha)] \right) \quad (6.26)$$

where

$$\omega(x, \alpha) = \begin{cases} \tan \frac{\alpha\pi}{2} & \text{if } \alpha \neq 1 \\ \frac{2}{\pi} \log |x| & \text{if } \alpha = 1 \end{cases} \quad (6.27)$$

and the sign function is defined as:

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x > 1 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 1 \end{cases} \quad (6.28)$$

Here, α is the characteristic exponent that decides the heaviness of the tails and its value should satisfy $0 < \alpha < 2$; λ is the location parameter and is within the range $-\infty < \lambda < \infty$; $\gamma > 0$ is the dispersion parameter and β is the index of symmetry ($-1 \leq \beta \leq 1$); its 0 value is for symmetric distributions. The Gaussian distribution is a special case of α -stable distribution [144–146] and is given by:

$$f_{\alpha=2, \beta=0}(\gamma, \lambda, x) = \frac{1}{\sqrt{4\pi\gamma}} \exp\left(-\frac{(x-\lambda)^2}{4\gamma}\right) \quad (6.29)$$

Its first order special case is the Cauchy distribution that models highly impulsive noises [146] and is given as:

$$f_{\alpha=1, \beta=0}(\gamma, \lambda, x) = \frac{\gamma}{\pi[\gamma^2 + (x-\lambda)^2]} \quad (6.30)$$

6.4.1 Simulation Results for FN-FeLMS Algorithm

Detailed simulations with different step sizes and fractional orders are performed, the primary path output is corrupted by additive zero mean white Gaussian noise $g(k)$, its variance is set at 60 dB below the input signal power level. Reliability and accuracy of FN-FeLMS is assessed in terms of MSE and MSD. Simulations are performed for 12,000 iterations for both FeLMS and FN-FeLMS algorithms; the

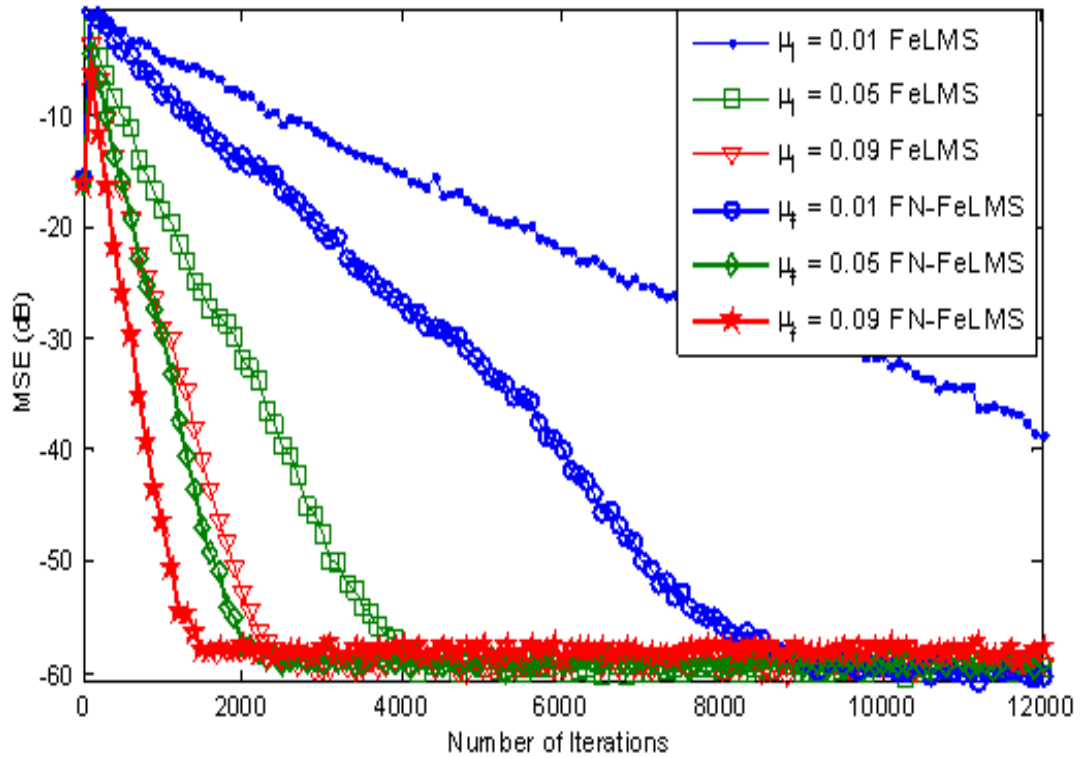


FIGURE 6.3: MSE learning curves for FeLMS & FN-FeLMS for different choices of μ_f and μ_l and for Gaussian noise, $\nu = 0.5$, SNR = 60 dB.

MSE and MSD values are averaged over 500 Monte Carlo runs. The step sizes are kept equal, that is, $\mu_f = \gamma\mu_l$ where γ is a multiplying factor; results are plotted in Figures 6.3-6.5. The FIR filter s is modelled as low pass [2], its impulse response is $\{1 \ -1.2 \ 0.72\}$.

Figure 6.3 shows the ensemble average MSE learning curves for Gaussian input data; step sizes for FeLMS are taken as: $\mu_l = \{0.01, 0.05, 0.09\}$. For FN-FeLMS, the fractional derivative order ν is taken as 0.5, and $\gamma = 3$.

Figure 6.4 shows the ensemble average MSD curves for Gaussian input data, ν is 0.1, the step size $\mu_l = \{0.02, 0.06, 0.1\}$, and $\gamma = 2$. Both figures confirm that FN-FeLMS has faster convergence than FeLMS.

Figure 6.5 shows MSD curves for binary input data, the fractional order is -0.25 and $\gamma = 4$. In all the simulation scenarios considered, the FN-FeLMS shows faster convergence than the FeLMS algorithm, also the steady state performance is seen to improve as the fractional order decreases. For cases $\mu_l = \{0.03, 0.08, 0.2\}$, the standard FeLMS method takes 6968, 2732, 1262 iterations to converge to an MSD

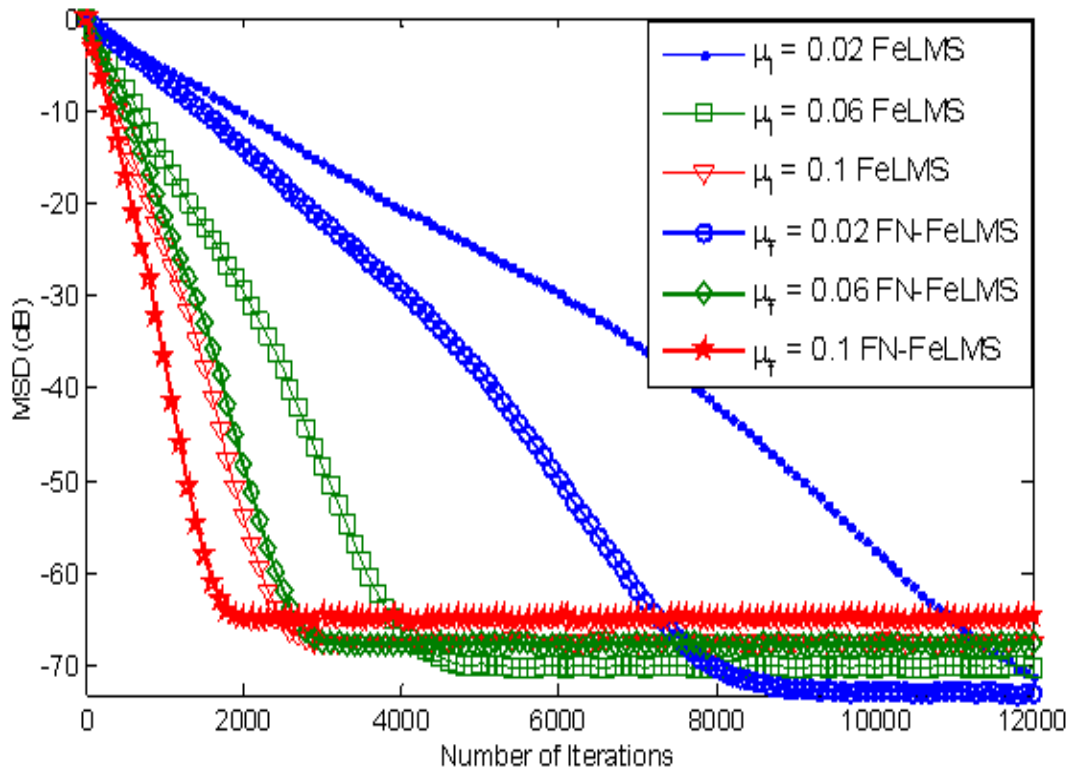


FIGURE 6.4: MSD learning curves for FeLMS & FN-FeLMS for different choices of μ_f and μ_l and for Gaussian input, $\nu = 0.1$, SNR = 60 dB.

of -60 dB, whereas the FN-FeLMS converges in 4324, 1748, 1001 iterations respectively.

Table 6.7 summarizes the effects of changing the fractional order on convergence and steady state performance for binary input data. Taking $\mu_l = 0.1$, $\gamma = 4$, the FeLMS algorithm yields a steady state MSD of -60 dB at iteration number 2500; to converge to -60 dB, a total of 2232 iterations are required. Column 2 of the table shows the number of samples required for the FN-FeLMS to converge to an MSD of -60 dB. The convergence improves as ν becomes more positive (more derivative action), while the steady state performance degrades. For negative values of ν , the convergence is slower but the steady state performance improves due to more integration action of the differintegral operator. The improvement over FeLMS is seen in the last column of the table.

Figure 6.6 shows normalized frequency responses for FeLMS and FN-FeLMS algorithms at iteration number 1500 for $\mu_l = 0.1$, $\mu_f = 0.5$; the latter more accurately matches the ideal response, indicating better path modelling.

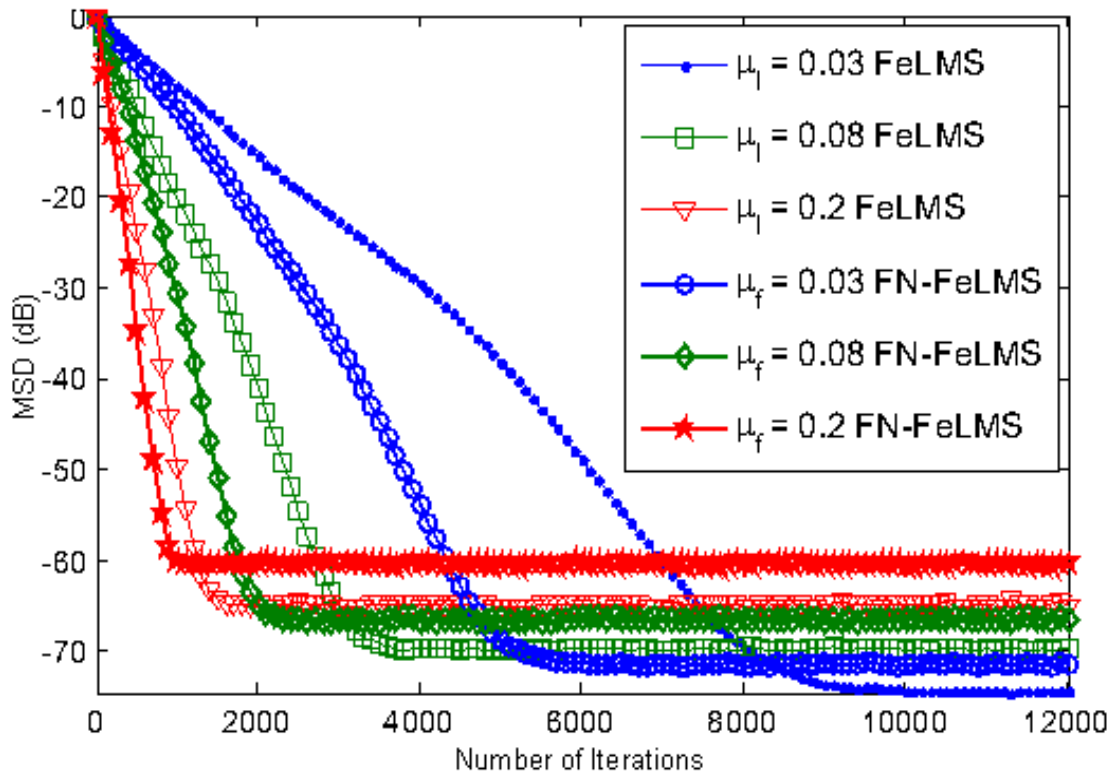


FIGURE 6.5: MSD learning curves for FeLMS & FN-FeLMS for different choices of μ_f and μ_l and for binary Input data, $\nu = 0. - 0.25$.

TABLE 6.7: Convergence and MSD for FN-FeLMS for $\mu_l=0.1, \gamma = 4$

ν	k for MSD of -60 dB	MSD at $k = 2500$	Gain in MSD over FeLMS
0.5	1063	-63.01	52.37%
0.25	1170	-63.73	47.58%
0.1	1259	-64.19	43.59%
-0.1	1371	-64.72	38.58%
-0.25	1442	-65.21	35.39%
-0.5	1596	-66.28	28.49%

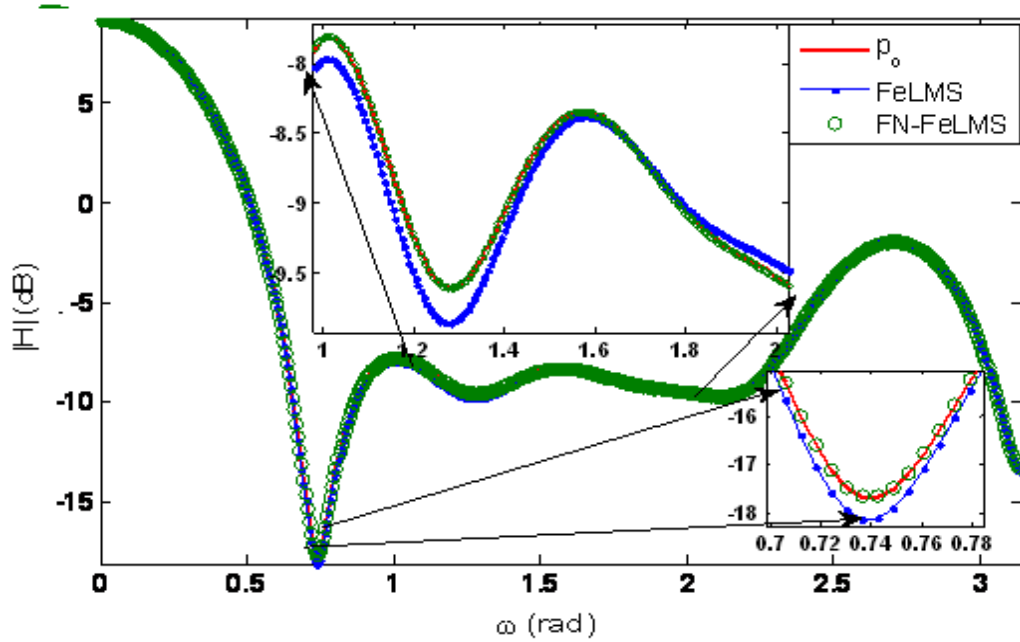


FIGURE 6.6: Frequency response of the primary path and secondary path models using FeLMS & FN-FeLMS methods, $\nu = -0.5$.

6.4.2 Simulation Results for Modified FO Algorithms

Figure 6.7 shows comparative MSD learning performance of the baseline FN-FeLMS [67] versus the proposed FrFeLMS in this work. The simulations are performed for 1400 samples and the ensemble MSD is obtained through 500 independent runs. The input is binary noise, the step size is 0.1 in both cases while the constant term is $\gamma = 4$ for both cases. The fractional order is varied from 0 to 1 in increments of 0.2. Convergence improves in both cases over the traditional FeLMS; however FrFeLMS exhibits faster convergence and better steady state performance as opposed to FN-FeLMS. In both cases, convergence increases with fractional order, however MSD degradation in the earlier scheme of [67] is more prominent as compared to the newly proposed scheme.

Table 6.8 provides both the -60 dB convergence and the steady state performance for $f = 1$ for 2000 samples, the ensemble MSD is obtained through 500 runs. Different fractional orders $\nu \in [0, 1]$ have been considered. For binary input, the step size $\mu_l = 0.1$, $\gamma = 1$, while for Gaussian input $\mu_l = 0.05$ and $\gamma = 4$. It is seen that for binary input, the proposed FrFeLMS has about 4 dB improved MSD performance, while for Gaussian input with smaller step size the improvement is about 5 dB over

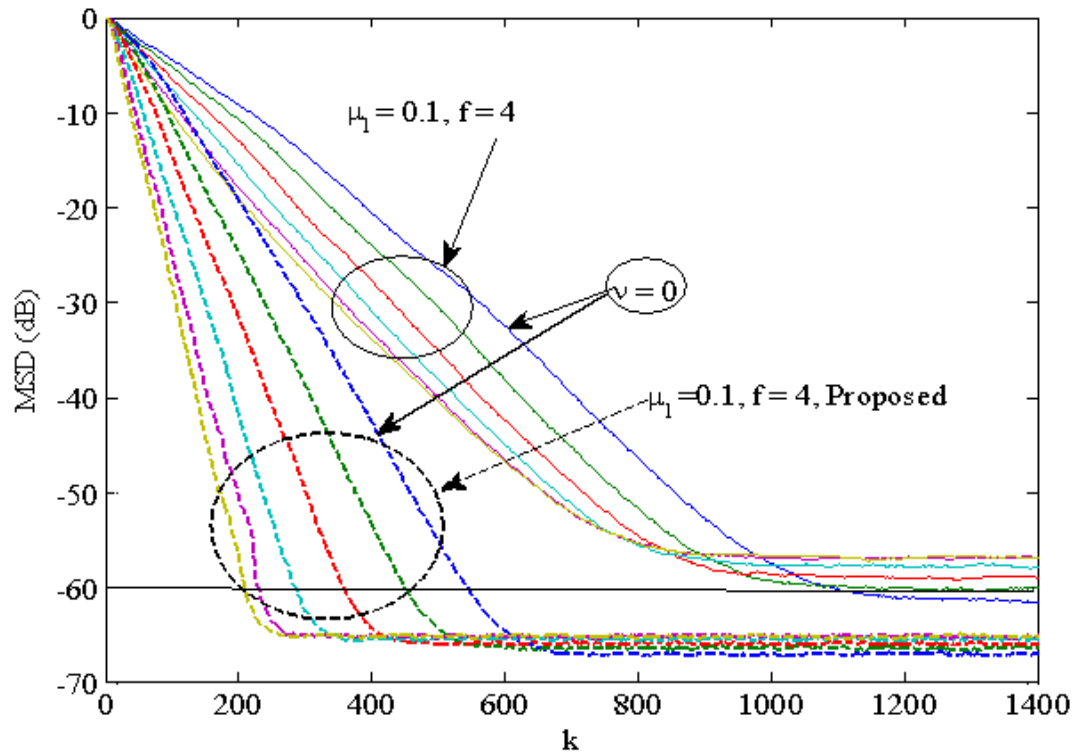


FIGURE 6.7: FrFeLMS vs. FN-FeLMS MSD Learning curves for different step sizes.

the FN-FeLMS [67]. The convergence improvement of the proposed FrFeLMS over the FN-FeLMS technique is from 43% to 67%; simulation results clearly establish the superiority of the proposed algorithm over that of the previous [67]. In [67] the MSD for the conventional case is -65.45 dB at 2500 samples, and for -60 dB, the required number of iterations is 2232. The convergence improvement over the conventional algorithm is in the range 54% to 84%.

Figure 6.8 shows the MSD performance of FxLMS and its fractional variant FrFxLMS for different step sizes for binary input noise. The fractional order chosen for FrFxLMS is 0.8. It can be seen that FrFxLMS has faster convergence for all step sizes and has a similar MSD performance in the steady-state. For a step size of 0.1, FxLMS converge to -60 dB in 8288 iterations whereas FrFxLMS requires 1946, resulting in 76.52% improvement. For $\mu = 0.08$ FxLMS converges in 1.034×10^4 iterations and FrFxLMS in 2386 iterations which results in 76.92% improvement. For $\mu = 0.05$, the improvement is 77.26% as the fractional variant requires 3721 iterations while its conventional counterpart takes 1.636×10^4 iterations.

TABLE 6.8: MSD Convergence performance comparison of FrFeLMS vs. FN-FeLMS [67]

Input noise	FO (ν)	k for FN-FeLMS	k for FrFeLMS	FN-FeLMS MSD at k=2000 [67]	FrFeLMS MSD at k=2000
Binary input $\mu_l = 0.1$ and $\gamma = 1$	0.1	1537	900	-65.59	-69.15
	0.3	1458	818	-65.56	-69.15
	0.5	1368	714	-65.36	-68.81
	0.7	1292	644	-64.96	-68.76
	0.9	1238	610	-64.96	-68.76
Gaussian input noise and $\mu_l = 0.05$ $\gamma = 4$	0	1808	1025	-64.09	-69.21
	0.2	1555	837	-63.64	-68.47
	0.4	1354	655	-63.53	-68.28
	0.6	1222	495	-62.39	-67.81
	0.8	1122	389	-62.32	-67.55
	1	1084	356	-62.32	-67.52

Figure 6.9 shows the MSD performance of MFxLMS and its fractional variant FrMFxLMS for different step sizes and Gaussian input noise. The noise level is kept 30 dB below the input signal level. Simulate results are obtained for 15000 iterations and each MSD is obtained by averaging over 200 independent runs. The fractional order chosen for FrMFxLMS is 0.95. It is seen that FrMFxLMS has faster convergence for all step sizes; the steady state MSD is a bit poorer. For a step size of 0.06, MFxLMS converges to -30 dB using 1999 iterations, whereas FrMFxLMS requires 1135 iterations resulting in a 43.22% improvement. For $\mu = 0.04$, the MFxLMS algorithm converges in 3022 iterations and FrMFxLMS in 1621 iterations which results in a 46.34% improvement. For $\mu = 0.06$, the improvement is 76.08% as the fractional variant requires 6253 iterations while its conventional counterpart takes 1.232×10^4 iterations.

Figure 6.10 shows the MSE learning curves of NLMS, FxLMS, FeLMS and their fractional variants FrNLMS, FrFxLMS and FrFeLMS for Gaussian noise as input.

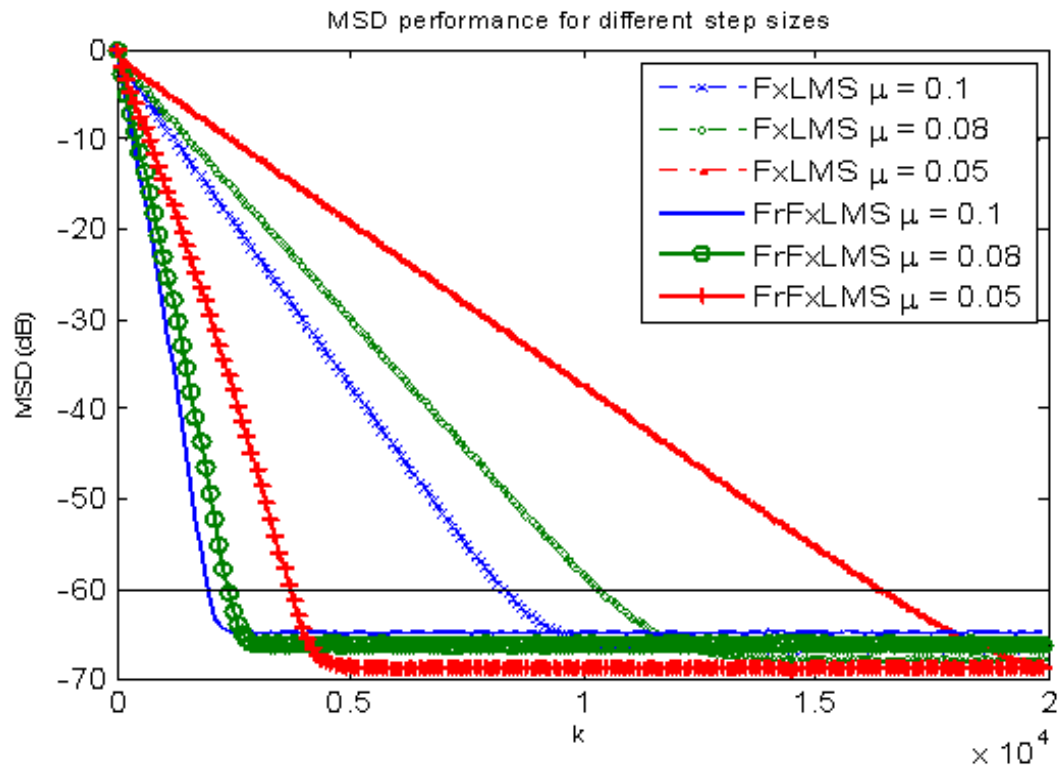


FIGURE 6.8: FxLMS vs. FrFxLMS MSD Learning curves for different step sizes.

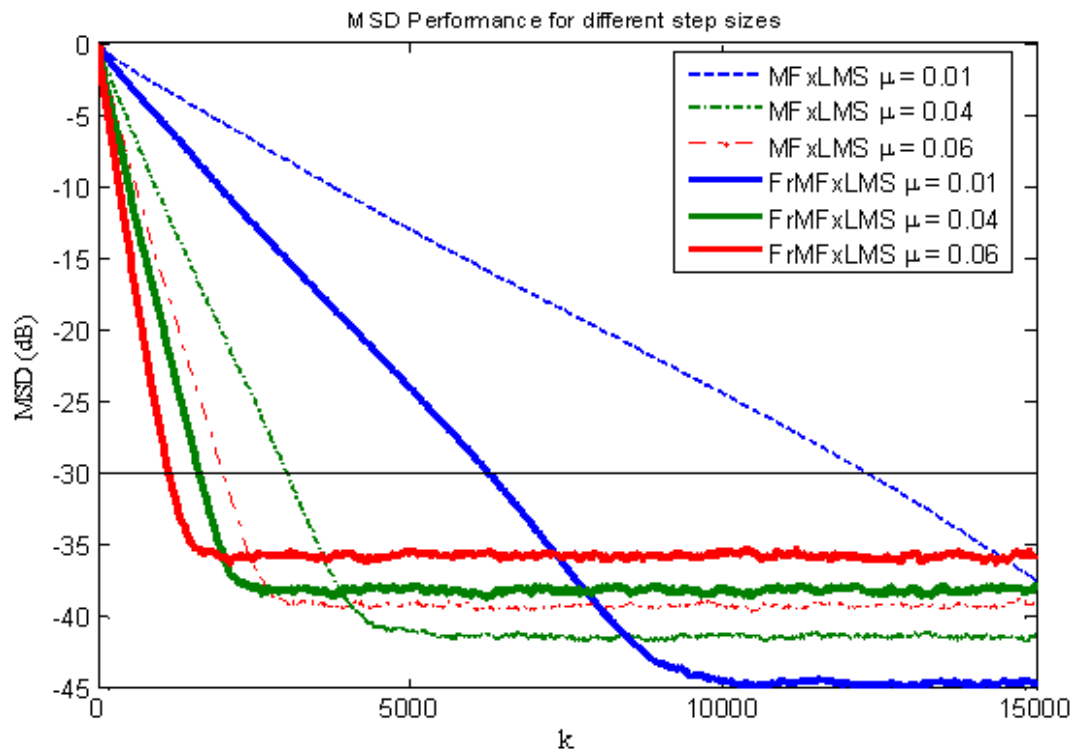


FIGURE 6.9: MFxLMS vs. FrMFxLMS MSD Learning curves for different step sizes.

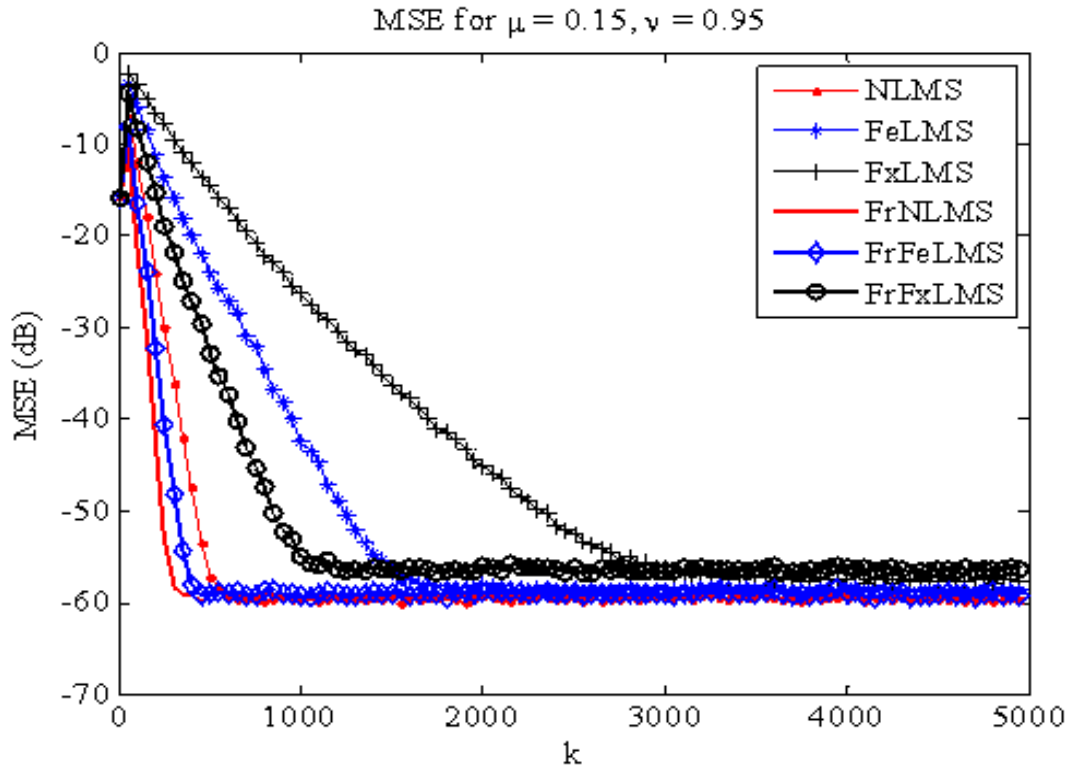


FIGURE 6.10: MSE performance of NLMS, FxLMS, FeLMS, FNLMS, FrFxLMS and FrFeLMS for Gaussian noise.

The fractional order is fixed at 0.95 and step size at 0.15. The basic 10-taps path is considered with results are shown for 5000 input samples and the ensemble average MSD is obtained from 1000 independent runs. The fractional variants have faster convergence rates than their corresponding conventional counterparts. The filtered variants show a better convergence performance. The FrNLMS has the fastest convergence, followed by the FrFeLMS, NLMS, FrFxLMS, FeLMS and FxLMS algorithms.

Figure 6.11 shows the MSD learning curves for different tap-lengths, input noise samples are drawn from standard normal distribution with zero mean and unit variance. Results are shown for 8000 input samples and ensemble average MSD is obtained from 1000 independent runs. The step size is fixed at 0.25 and the fractional order is set to 0.8. Tap lengths considered are 19, 28 and 37 which are obtained from the basic 10-tap primary path \mathbf{p} through the convolution operation. As can be seen the performance of FeLMS degrades severely as compared to FrFeLMS. With FrFeLMS the 60 dB MSD value is obtained in 587 iterations for 19-taps, 978 iterations for 28-taps and 1477 iterations for the 37-taps case. For the

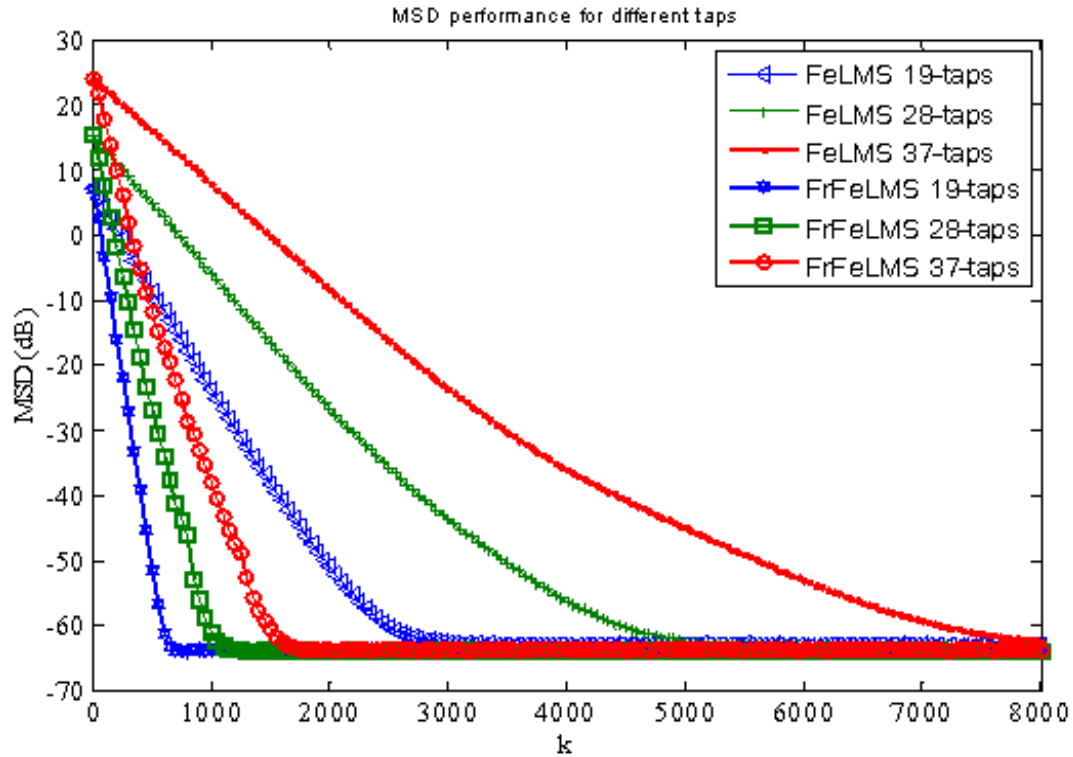


FIGURE 6.11: FeLMS vs. FrFeLMS MSD Learning curves for different tap-lengths.

FeLMS, it requires 2487, 4448 and 7168 iterations, respectively. The convergence improvement over FeLMS is therefore 76.4%, 78% and 79.4% for 19, 28 and 37 taps, respectively.

Figure 6.12 shows the MSD performance of FxLMS and FrFxLMS for binary input generated through an i.i.d. random process with equally probable bipolar outputs and different number of taps of the primary path, the step size chosen is 0.15 and the fractional order is 0.75. It can be seen that FrFxLMS has faster convergence rates for all taps-lengths, and has an almost similar MSD performance in the steady-state. Using 10-taps, FxLMS converges to -60 dB in 5670 iterations, whereas FrFxLMS requires 1400 iterations which correspond to a 76.31% improvement. For 19-taps FxLMS converges in 1.225×10^4 iterations and FrFxLMS in 2879 iterations which results in 76.50% improvement. For the 28-taps case, the improvement of FrFxLMS over FxLMS is 75.23% as the fractional variant requires 4954 iterations while FxLMS takes more than 20000 iterations.

Figure 6.13 shows the MSD learning curves of FrFeLMS for different fractional orders and step sizes. Again the noise samples are drawn from standard normal

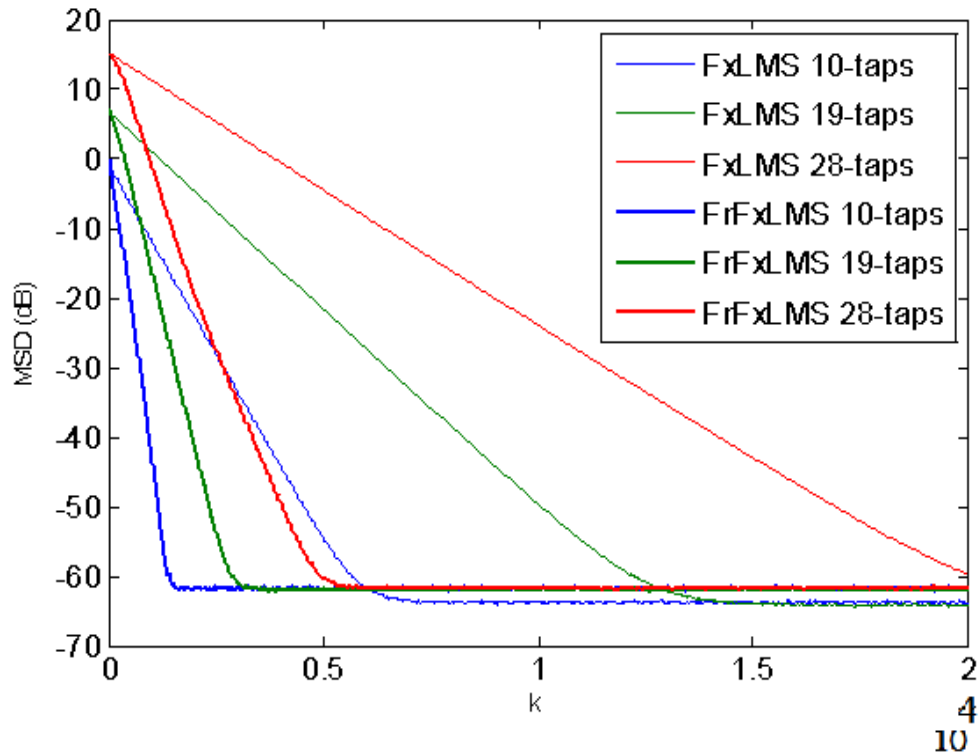


FIGURE 6.12: FxLMS vs. FrFxLMS MSD Learning curves for different tap-lengths.

distribution with zero mean and unit variance. The basic 10-taps path is considered with results are shown for 1000 input samples and the ensemble average MSD is obtained from 1000 independent runs. Results on the left side are shown for a fixed step size of 0.5 and the fractional order is varied from 0.1 to 1 in increments of 0.1. Results on the right side are shown for a step size of 0.1 and the same fractional orders. It is seen that the convergence speed increases as the fractional order is increased. The MSD is dependent on step size only; it improves as the step size is decreased.

Figure 6.14 shows effects of changing the number of taps of the primary path at different iteration numbers, the procedure is the same as for Figure 6.12. Here, the weights and regression vectors are kept as of the previous iteration. The numbers of taps are changed at iteration number [1000, 2000, 3000, and 4000] from 10 to 19, 19 to 28, 28 to 37, 37 to 28, and 28 to 19 and back to 10. The change in primary path does not affect the MSE performance of fast convergent NLMS, FrNLMS, and FrFeLMS since the paths are already modeled using weights in the previous iterations. In the case of FxLMS, its MSE performance is seen to be quite poor

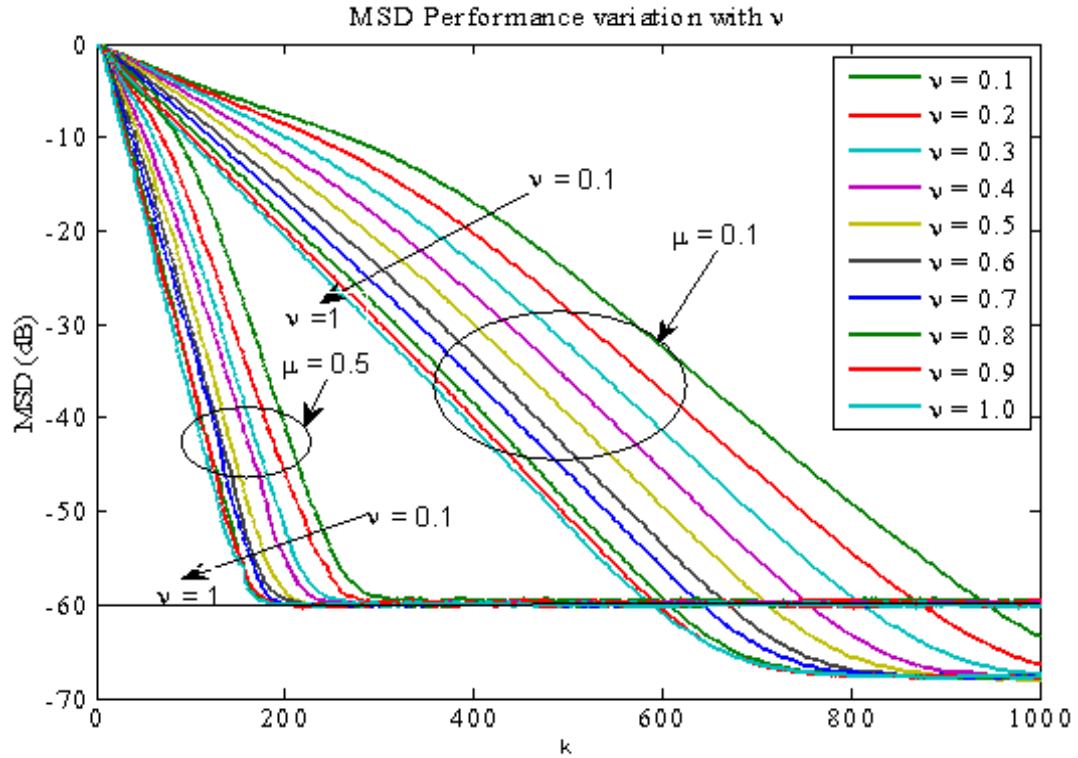


FIGURE 6.13: MSD Learning performance of FrFeLMS for different fractional orders and Gaussian input noise.

relative to FrFxLMS.

6.4.3 Simulation Results for Impulsive Noises

Figure 6.15 shows the relative modeling error performance of FeLMS and FrFeLMS for $S\alpha S(\beta = 0)$ based impulsive noise having 1st order and fractional order statistics. Simulations are provided for the basic path for 15000 iterations and the average obtained of 100 independent runs. The fractional order is fixed at 0.85, $\gamma = 1.5$ and $\mu_l = \mu_f$. The small figure shows a portion of the input noisy signal samples with outliers reaching amplitude of 1000; the input corresponds to Cauchy density function when $\alpha = 1, \beta = 0$. In such a scenario the FeLMS has a divergent behavior and is unstable even at low step sizes. The performance of FrFeLMS is, however, almost within the acceptable range. For $\alpha = 1.5$ and 1.75, the FrFeLMS has faster convergence at relatively bigger step sizes of 0.075 and 0.1 respectively. For $\alpha = 1.5$ and $\mu_l = 0.075$, the -60 dB performance is achieved by FrFeLMS

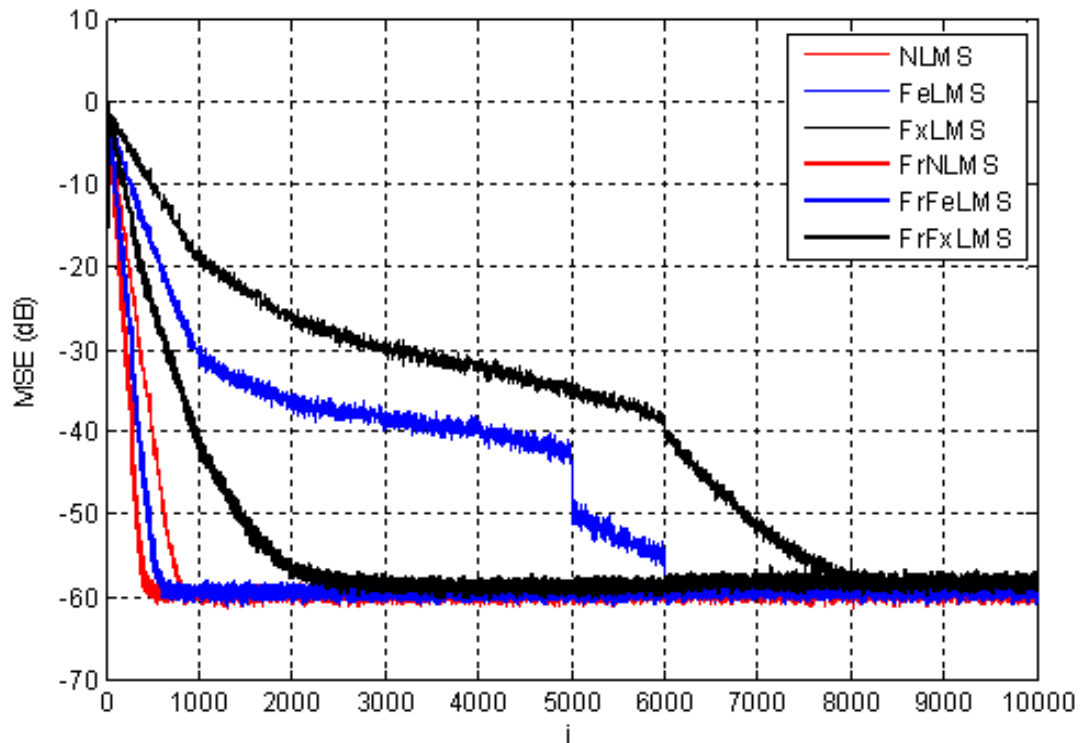


FIGURE 6.14: Effects of changing the number of taps of the primary path at different iteration number.

in 1346 iterations while 6603 iterations required in FeLMS; the improvement in convergence is 79.6%. Similarly For $\alpha = 1.75$ and $\mu_l = 0.1$, FrFeLMS converges in 1066 iterations and FeLMS in 4849 iterations, the improvement in convergence is 78%.

Figure 6.16 shows the relative modelling error performance of FeLMS and FrFeLMS for different α values and step sizes for A α S ($\beta = 1$) impulsive noise. The results are generated for the primary path P_1 with 19-taps. The RME is calculated for 10000 iterations and averaged over 200 independent runs. The fractional order is fixed at 0.85, $\gamma = 1.5$ and $\mu_l = \mu_f$. As opposed to the previous case, FeLMS has stable but slow convergent behavior. For $\alpha = [1.75, 1.5]$ with corresponding $\mu_l = [0.1, 0.075]$, the FeLMS converges in [4757, 7297] iterations while the proposed FrFeLMS in [1008, 1374] iterations, which corresponds to an improvement of [78.8%, 81%] respectively. For $\alpha = 1.75$, the FrFeLMS has a very fast convergence as compared to its counterpart.

Figure 6.17 shows the relative modelling error performance of FN-FeLMS and FrFeLMS, the impulsive noise is generated for the characteristic exponent $\alpha = 1.5$,

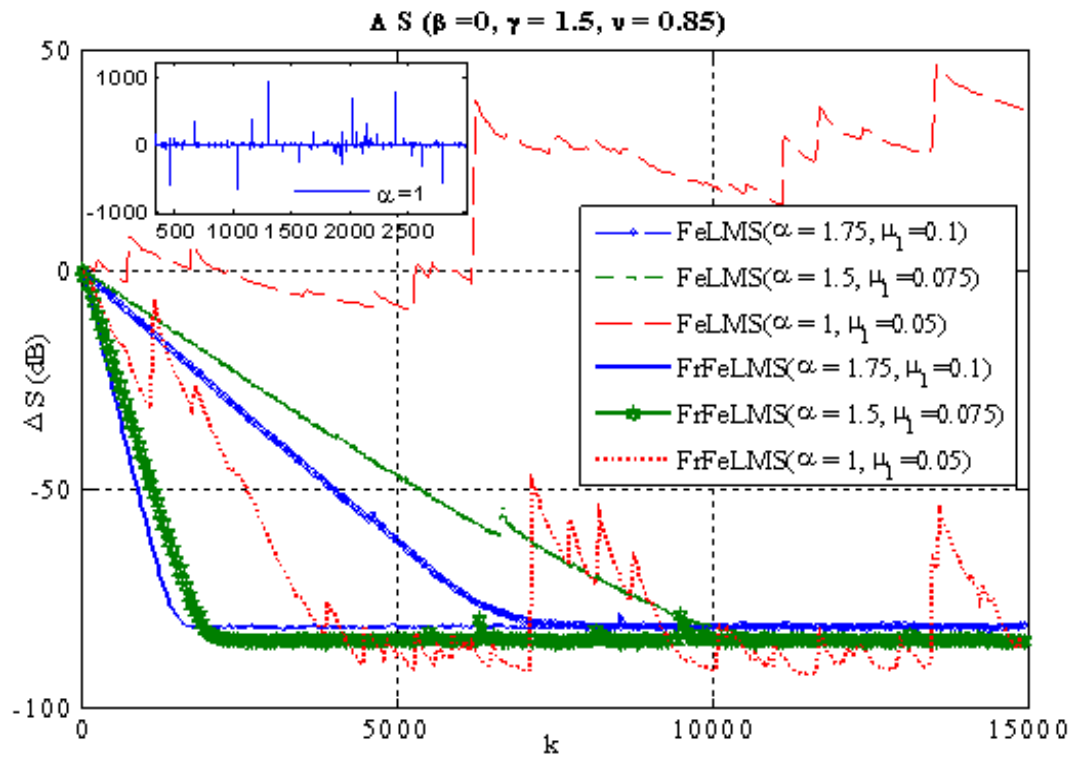


FIGURE 6.15: Relative Modelling Error of FeLMS and FrFeLMS algorithms for $S_\alpha S$ impulsive input noise.

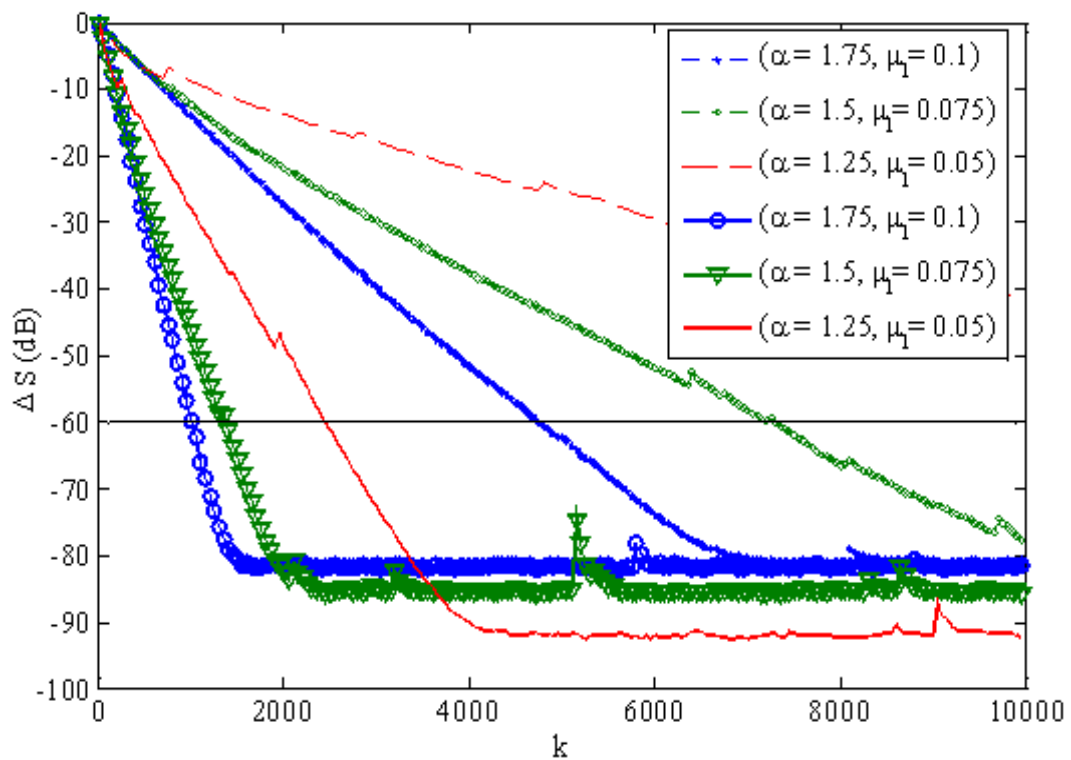


FIGURE 6.16: Relative Modelling Error of FeLMS and FrFeLMS for asymmetric α -stable impulsive input noise.

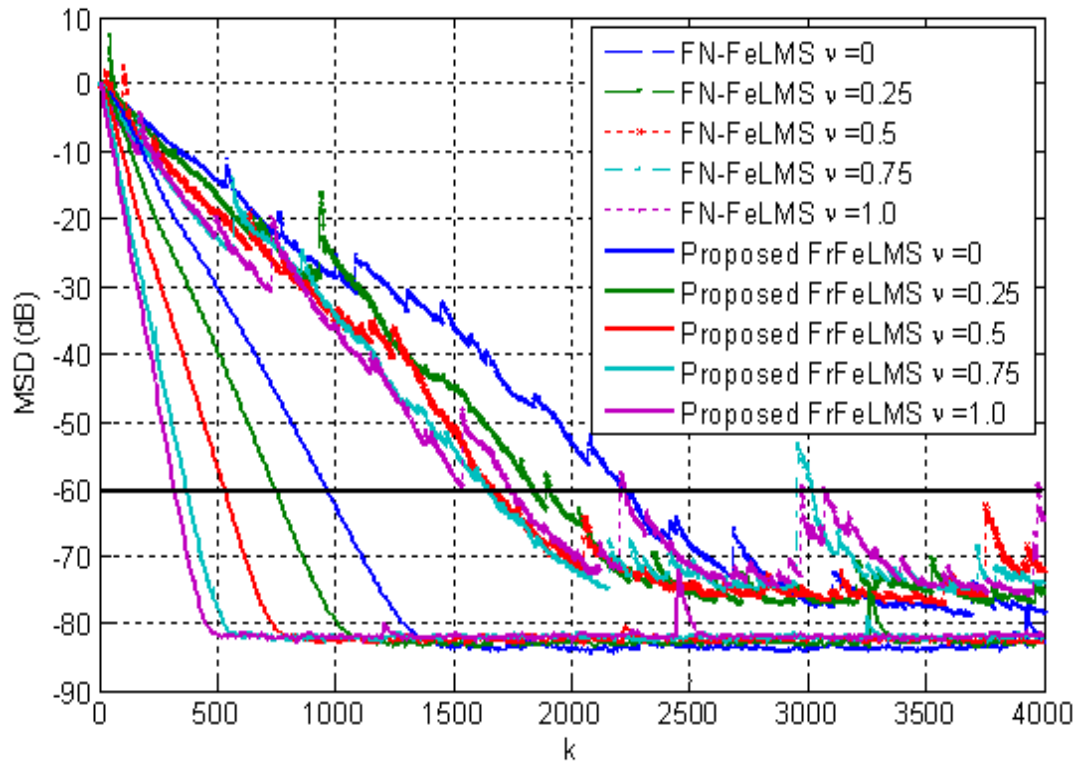


FIGURE 6.17: Relative Modelling Error of FeLMS and FrFeLMS for asymmetric α -stable impulsive input noise.

dispersion parameter of 5 and $\beta = 1$. The iterations considered are 4000 and Monte Carlo runs of 200. The same step size of 0.05 has been chosen and fractional orders of [0, 0.25, 0.5, 0.75, 1] are kept for both the algorithms. It can be seen that the proposed variant FrFeLMS has superior performance in terms of convergence rate as well as steady state response as compared to FN-FeLMS for all fractional orders.

It is seen from simulation results that FrFeLMS shows much improvement in convergence performance over FeLMS, followed by FrFxLMS over FxLMS. The improvements in these algorithms is about 75-80%. In case of FrMFxLMS the convergence improvement is approximately 45% in all cases. The steady-state performance of the fractional variants also shows improvement, especially when the relative modeling performance metric is considered. The limitation of the conventional algorithms is that the only adaptation parameter is the step size which controls the rate of convergence and provides robustness in ANC applications. The fractional order algorithms provide fast convergence by treating each weight

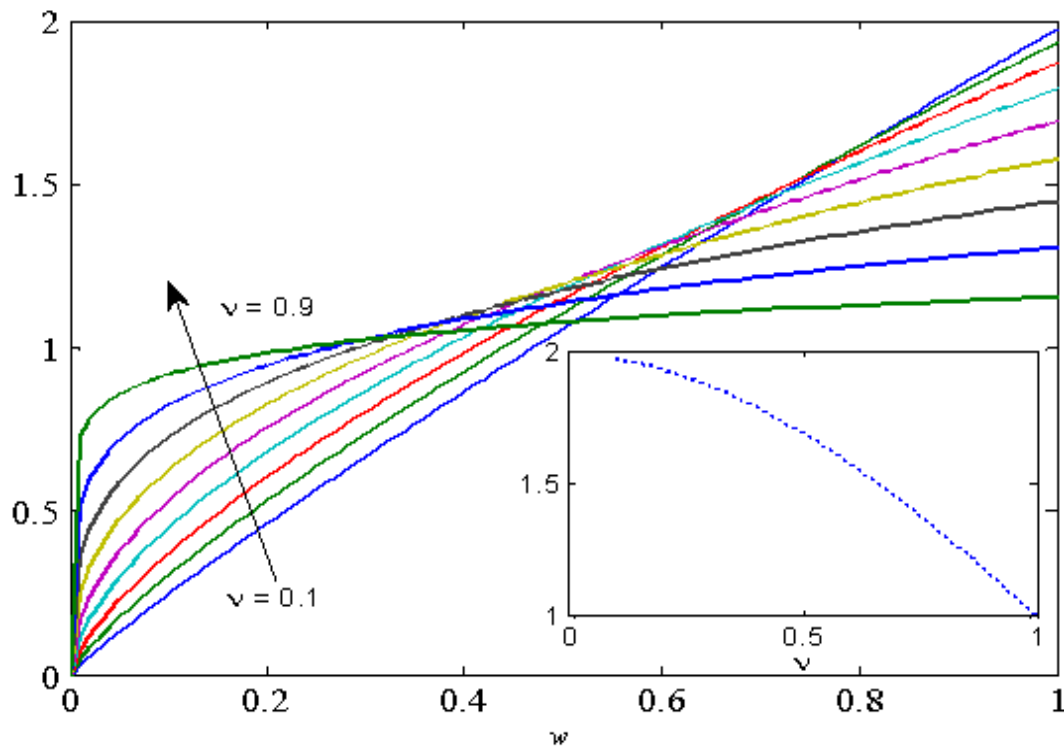


FIGURE 6.18: Effects of fractional order on $Gw^{1-\nu}$, the single plot shows the behavior of G for different fractional orders.

independently [127] according to the fractional order and weight values. In Figure 6.18, plots of the term $Gw^{1-\nu}$ are shown for a scalar w to show the effect of varying the fractional order. Increasing the fractional order results in a larger 'gain' for small weight values. The plot at the bottom right shows the variation of G with respect to the fractional order, which also helps adjusting the step size. The proposed strategies provide improvements over the traditional approaches, the new fractional technique helps capture the parameters of the model quickly, while keeping the error small.

6.5 Conclusions

New hybrid algorithms are designed to generate fractional alternatives of FxLMS and its variants. The algorithms are applied to the active noise control system in the feed-forward configuration using different non-integer orders, and performance

is compared with their traditional counterparts. The performance indicators considered are the mean squared error, the mean squared deviation and the relative modeling error. The performance of the algorithms are also considered for different primary path models changing during the steady state. Results show that the fractional variants converge faster for different types of input noise such as binary and Gaussian. The steady-state performance for the designed algorithms is almost the same as their conventional counterparts. It is seen that FrFeLMS shows highest improvement in convergence performance over FeLMS, followed by FrFxLMS over FxLMS. The improvements in these algorithms are of the order of 75-80% for the same step size. In case of FrMFXLMS the convergence improvement is approximately 45% in all cases.

Chapter 7

Tracking of Rayleigh Fading Channels

This chapter presents the tracking behavior of Fractional Order (FO) variants of the Normalized Least Mean Square (NLMS) algorithm in a nonstationary environment modeled as time varying Rayleigh fading sequence. In such cases, mostly the celebrated Recursive Least Squares (RLS) or its variant Extended RLS (E-RLS) algorithms have much degraded steady state response. The FO algorithms are obtained by applying fractional derivatives in the cost function; such schemes provide two step sizes and an FO to control the rate of convergence and have an acceptable steady state behavior. In evaluation, a high speed mobile environment is considered with a Rayleigh channel which results in different Doppler frequency shifts depending upon the transmission frequency and relative velocity of the transmitter and receiver. The proposed algorithms are compared with the NLMS, RLS and E-RLS schemes, numerical experiments show the superiority of the FO variants over these schemes in terms of stability and model accuracy in the steady state. A hybrid scheme is also shown where the weights of an FO variant are initially trained with RLS and then performs self-adaptation; the FO scheme is seen to have better performance than all traditional counterparts.

The presentation in the rest of the chapter is as follows: Section 7.1 presents introduction with importance of the tracking relating to literature. Section 7.2 states

the system model for the tracking of time varying sequences. Sections 7.3 and 7.4 respectively summarize the RLS and Fractional variants of NLMS algorithms. Section 7.5 shows simulation results and discussions for different parameters of the algorithms and characteristics of the fading sequences. The final section concludes the chapter.

7.1 Introduction

Fast and efficient implementation of algorithms is required to accurately track fast time varying channels which result from Doppler phenomena in a high speed mobility environment such as high speed airplanes, trains and vehicles. In these applications, Doppler shifts of about a tenth to a few thousands of Hertz may arise depending upon the transmission frequency, vehicle velocity and direction of motion [77, 78, 147]. The problem is made worse by multipath effects due to reflections, scattering and diffraction, which result in inter-symbol interference (ISI) [78, 147–149]. The LMS based algorithms are frequently applied in stationary environments due to their implementation ease and better steady state performance [79, 150]. However, they have poor rate of convergence; so alternative approaches such as recursive least squares (RLS) or extended RLS are used [11]. However, in time-varying nonstationary environments [122], the LMS, RLS and their variants compromise convergence with steady state performance; and at very high Doppler frequency shifts (with vehicular speeds of the order of 500 km/h for LTE [38, 151–153], 225km/h in IS-136 [41, 147, 154, 155] and other standards that can result in a maximum of Doppler shift of roughly 0.8KHz to 3KHz, the steady state performance also substantially degrades [25, 82, 83]. RLS achieves faster convergence but its steady state behavior degrades as the dynamics increase, while the LMS has superior performance in steady state but has the issue of poor rate of convergence. To get improved performance in both transient and steady state, the Kalman filter can be applied but it exhibits relatively high complexity due to on-line update of the Riccati equation [11], [156, 157].

The effects of multipath propagation include signal fading which is characterized

by delay spread and Doppler spread. Signal fading is caused by the interference between signals propagating through different paths. The delay spread is broadening in duration of received signal with respect to the transmitted signal as different delays are associated with the propagation paths. Doppler spread refers to the broadening of the frequency spectrum of the received signal with respect to the transmitted signal, when there is relative motion between the transmitter and the receiver. This is due to the different angles of arrival associated with the propagation paths.

The importance of tracking fast fading channels in high mobility vehicular communications [147, 149], motivates the authors to examine the application of fractional calculus to improve the performance metrics i.e., mean squared error (MSE) and mean squared deviation (MSD). It is worth mentioning that the performance of different tracking algorithms has been studied extensively in the literature for the last three decades [158–168], including variable step size adaptation in [158, 159, 169], amplitude tracking in OFDM systems in [160, 161], tracking in MIMO systems in [162, 163] and MIMO-OFDM systems in [164, 165]. For tracking channels, least squares [71, 166] and spline adaptive filters [167] have been applied recently, [168] provides a study of a third order amplitude tracking loop for online slow fading sequences.

7.2 System Model

Rayleigh fading channels provide a practical example of non-stationary environment. For tracking a Rayleigh fading sequence, a block diagram based on adaptive filtering is shown Figure 7.1. In such a case, the auto-correlation matrix of the input signal or the cross correlation vector of the input-output signals are time varying for which the optimal Wiener filter weight vector $\mathbf{w}_o(k)$ is also time varying. Estimation with a Wiener filter is a complex implementation issue requiring the calculation of an auto-correlation matrix, its inverse and the output cross-correlation vector. A gradient-based adaptive filter, on the other hand does not explicitly require the cross correlation vector or auto-correlation matrix in its

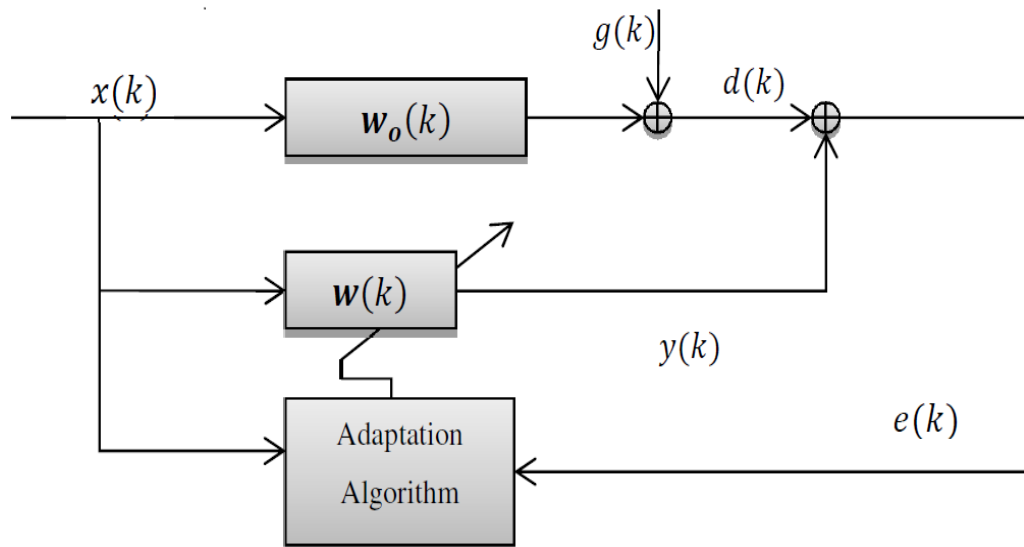


FIGURE 7.1: Schematic of adaptive tracking problem.

weight update equation to track the changes in the optimal weights. Finite Impulse Response (FIR) transversal structure is assumed for the adaptive filter with time varying weight vector \mathbf{w} with M taps. An error vector is defined with M elements:

$$\mathbf{e}(k) = [e(k), e(k-1) \dots e(k-M+1)]^T. \quad (7.1)$$

The objective of the adaptive filter is to minimize the instantaneous (squared) error between the observed output:

$$d(k) = \mathbf{w}_o^T(k)\mathbf{x}(k) + g(k). \quad (7.2)$$

and the output $y(k)$ of the adaptive filter. In equation (7.2), the time varying vector $\mathbf{w}_o(k)$ corresponds to the optimal Wiener weights required for optimal identification and $g(k)$ is the additive white noise modelled as a zero mean Gaussian random variable. Having stated the inputs and outputs of the filter, in the next section the adaptation algorithms are presented based on fractional order calculus as well as NLMS, RLS and E-RLS.

7.3 The Recursive Least Squares Algorithm

The weights in the least-squares method are optimized by minimizing the sum of squared values of the error samples of the filter output. The optimization procedure relies on observations from the time the filter starts until the present time. The objective function [5, 6, 11, 79], is defined as:

$$\xi(k) = \sum_{l=0}^k \rho(l) e^2(l) = \sum_{l=0}^k \lambda^{k-l} e^2(l). \quad (7.3)$$

where λ is the forgetting factor and its objective is to weight the recent samples more heavily as compared to the older ones. The input data is conveniently assumed to be zero for initialization. The error $e(k)$ is the difference of the desired response $d(k)$ and the filter output $y(k)$, that is:

$$e(k) = d(k) - \mathbf{w}^T(k) \mathbf{x}(k) \quad (7.4)$$

For derivation of RLS, standard literature can be seen such as [5, 6, 11, 79] and other references, the objective here is to state its update procedure. The auto-correlation matrix \mathbf{R} is defined as the expectation of the outer product of input vectors scaled by λ as given by:

$$\mathbf{R}(k) = \sum_{l=0}^k \lambda^{k-1} x^T(l) x(l). \quad (7.5)$$

Next, the input-output cross correlation vector is defined in terms of scale factor λ as:

$$\mathbf{p}(k) = \sum_{l=0}^k \lambda^{k-1} x_k^T(l) y_k(l). \quad (7.6)$$

A gain vector is calculated to simplify the calculations and later its usage in the update part, this is called the Kalman gain and in terms of forgetting factor and auto-correlation matrix [5, 11, 79], it is given by:

$$\mathbf{q}(k) = \frac{\mathbf{R}^{-1}(k-1) \mathbf{x}^T(k)}{\lambda + \mathbf{R}^{-1}(k-1) \mathbf{x}^T(k)}. \quad (7.7)$$

The final update equation is:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mathbf{q}(k)\mathbf{e}^*(k). \quad (7.8)$$

The current auto-correlation matrix is updated [5, 11, 79] as below:

$$\mathbf{R}^{-1}(k) = \frac{1}{\lambda} [\mathbf{R}^{-1}(k-1) + \mathbf{q}(k)\mathbf{x}^H(k)\mathbf{R}^{-1}(k-1)]. \quad (7.9)$$

The total computational complexity of the RLS algorithm is very high, especially when M is large [5, 6]. The complexity per iteration is about $4M^2$ multiplications and $3M^2$ additions/subtractions per iteration [6, 11]. The above algorithm is the standard RLS algorithm and the update equations are represented in terms of correlation matrix and gain vectors. Although there are many variants of the RLS algorithm, in the following, only the extended RLS (E-RLS) algorithm is presented using the alternate representation form used for describing the Kalman filter. The general RLS algorithm is a special case of Kalman filter [40, 157]. In [5, 40] it has been shown through general deterministic criterion that E-RLS is equivalent to a full-blown Kalman algorithm and is better suited for tracking of non-stationary inputs. The tracking problem can be formulated by having the measurement variable $d(k)$ as in equation (7.2) and set $h(k) = w(k)$. The recursions in terms of forgetting factor λ can be derived, the details of which can be seen in the references [5, 11, 79, 170–172]. The a-posteriori error in terms of forgetting factor can be written as:

$$r(k) = \lambda^k x(k)P(k-1)x^*(k) \quad (7.10)$$

The Kalman gain parameter can be written as:

$$q(k) = \frac{P(k-1)}{r(k)} \quad (7.11)$$

The a-priori error is calculated as:

$$e(k) = d(k) - \mathbf{h}^T(k-1)\mathbf{x}(k) \quad (7.12)$$

The estimated input is obtained through the scaled (weighted by α) weight update equation as below:

$$h(k) = \alpha h(k-1) + q(k)e(k-1) \quad (7.13)$$

Finally,

$$P(k) = |\alpha|^2 \left[P(k-1) - \frac{P(k-1)x^*(k)x(k)P(k-1)}{\lambda^k + x(k)P(k-1)x^H(k)} \right] + \lambda^k sI \quad (7.14)$$

By the change of variables $h(k) = w(k)$, $P(k) = \lambda^{-k}P(k+1)$, and multiplying the recursion for $P(k)$ by λ^{-k} , zero initialization of w , the following two update equations are obtained for weight adaptation and $P(k)$ respectively.

$$w(k) = \alpha w(k-1) + \frac{\lambda^{-1}\alpha P(k-1)x^*(k)}{1 + \lambda^{-1}x(k)P(k-1)x^*(k)} [d(k) - x(k)w^T(k-1)] \quad (7.15)$$

$$P(k) = \lambda^{-1}|\alpha|^2 \left[P(k-1) - \frac{\lambda^{-1}P(k-1)x^*(k)x(k)P(k-1)}{1 + \lambda^{-1}x(k)P(k-1)x^*(k)} \right] + sI \quad (7.16)$$

Here, s is some positive scalar and $|\alpha| < 1$. Equivalence among the many variants of the RLS algorithms has been established widely. Here, the tracking performance of fractional order strategies developed in this thesis are compared with the above well-known RLS algorithm which is a special case of the Kalman filter and the E-RLS which is full-blown equivalent of the Kalman filter as already discussed.

7.4 Fractional Least Mean Square Algorithm and Its variants

In this section, the fractional order algorithms are presented in a summarized form as most of them have been derived in previous chapters. A Fractional Normalized LMS (F-NLMS) algorithm (version 1) with step size dependent on the input signal energy and having two update parts [71], that is, a first order gradient based and a fractional update part is as follows:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_l \mathbf{g}(k) + \mu_f \mathbf{g}(k) \odot \frac{\mathbf{w}^{1-\nu}(k)}{\Gamma(2-\nu)} \quad (7.17)$$

where $\mathbf{g}(k)$ is the correction term based on integer order derivatives and is given by:

$$\mathbf{g}(k) = \frac{\mathbf{x}^H(k)e(k)}{\|\mathbf{x}(k)\|^2 + \epsilon} \quad (7.18)$$

The initially developed fractional least mean square (FLMS) algorithm [37, 68, 69] has some inherent issues. The final update was derived based on successive approximation assumption which is essentially not required and can be relaxed. In [67, 88, 114], it was proposed to calculate the standard update equation first, followed by the fractional update part and then both the weights are added to get the final weight update (version 2). The iterative processes is as follows:

$$\mathbf{w}_l(k) = \mathbf{w}(k-1) + \mu_l \mathbf{g}(k) \quad (7.19)$$

where, $\mathbf{g}(k)$ is calculated as in equation (7.18). Next the fractional part is calculated using the weights updated through equation (7.19) and the iteration number is kept fixed, that is:

$$\Delta \mathbf{w}_f(k) = \mu_f \mathbf{g}(k) \odot \frac{\mathbf{w}^{1-\nu}(k)}{\Gamma(2-\nu)} \quad (7.20)$$

TABLE 7.1: Summary of Fractional Normalized LMS v3.

$\mathbf{w}_l(k) = \mathbf{w}^{(k-1)} + \mu_l \frac{\mathbf{x}(k)[d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)]}{\ \mathbf{x}(k)\ ^2 + \epsilon}$	(A)
$\Delta \mathbf{w}_f(k) = \mu_f \frac{\Gamma(3 - \nu)}{\Gamma(2 - \nu)} \frac{\mathbf{x}(k)[d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)]}{\ \mathbf{x}(k)\ ^2 + \epsilon} \odot \frac{ \mathbf{w}_l(k) ^{(1-\nu)} \text{sgn}(\mathbf{w}_l(k))}{\Gamma(2 - \nu)}$	(B)
$\mathbf{w}(k) = \mathbf{w}_l(k) + \Delta \mathbf{w}_f(k)$	(C)

The final update is formed through the addition of equations (7.19) and (7.20).

$$\mathbf{w}(k) = \mathbf{w}_l(k) + \Delta \mathbf{w}_f(k) \quad (7.21)$$

For the general case where the weights in the fractional update part can be positive or negative, the implementation is done by using the *sgn* function. In such a case, the equation (7.20) can be written as follows:

$$\Delta \mathbf{w}_f(k) = \mu_f \mathbf{g}(k) \odot \frac{|\mathbf{w}_l(k)|^{(1-\nu)} \text{sgn}(\mathbf{w}_l(k))}{\Gamma(2 - \nu)} \quad (7.22)$$

It can be noted that in both the updates equation (7.17) and equation (7.20), the step size in the fractional updates is independent of the fractional order. The third modification (version 3) made to the normalized variant was the fractional order dependent step size adaptation; it has been applied in active noise control systems [67, 88]. For the tracking problem, the summary of the algorithm is given in Table 7.1; note that this is a modified version of the algorithm as presented in [67, 173]. The final modification (version 4) which is made is the use of posterior error based adaptation. The standard update uses the prior error, while the fractional part exploits the posterior error. The update is derived using the fractional derivative by taking the posterior error in the second cost function. Table 7.2 shows the main update equations. In the next section, the tracking performance results of these algorithms are shown and a comparative study is performed with NLMS, RLS and E-RLS algorithms.

TABLE 7.2: Summary of F-NLMS v4..

$e(k) = d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)$	(A)
$\mathbf{w}_l(k) = \mathbf{w}(k-1) + \mu(k)e(k)\mathbf{x}(k)$	(B)
$e_p(k) = d(k) - \mathbf{w}_l^T(k)\mathbf{x}(k)$	(C)
$\Delta\mathbf{w}_f(k) = \mu_f(k)e_p(k)\mathbf{x}(k) \odot \mathbf{w}_l(k) ^{(1-\nu)} \text{sgn}(\mathbf{w}_l(k))$	(D)
$\mathbf{w}(k) = \mathbf{w}_l(k) + \Delta\mathbf{w}_f(k)$	(E)

7.5 Simulation Results and Analysis

For performance evaluation of the proposed algorithms, the tracking of Rayleigh fading channels is considered; different channels are generated with different Doppler shifts. Fading occurs in wireless communications due to multi-paths and motion of the transmitter, receiver or the environment. The combined signal from all paths is the result of constructive or destructive interference depending upon the phases of the individual signals; among several mathematical models Rayleigh fading is generally the most practical. The composite signal magnitude exhibits Rayleigh distribution while the phase follows a uniform distribution. The Doppler frequency changes in direct proportion to the vehicle speed as well as the transmitted signal frequency. For simulation purpose and comparative analysis with other algorithms, two basic performance metrics of mean squared error (MSE) and mean squared deviation (MSD) are considered for the evaluation. All the algorithms are evaluated through extensive simulation results using the Monte Carlo approach, a comparison is provided with their traditional counterparts.

The above four fractional order algorithms are considered with equation (7.17) as version 1 (v1), the second algorithm (7.19-7.21) as v2, the third (Table 7.1 A, B, C) as v3 and the fourth (Table 7.2 A, B, C, D, E) as v4. It is shown in [40, 171, 172] that NLMS is robust in a nonstationary environment while the computationally complex RLS and E-RLS algorithms have degraded performance as the Doppler shift increases. The fractional variants are compared through simulation results for time varying channels generated for different Doppler shifts in frequency. The

observed output is corrupted by the measurement noise $g(k)$ which has a Gaussian distribution. It has been shown in previous chapters that convergence improves with increase in fractional order. In most cases, simulations for higher fractional orders are shown, all the results are obtained for 1000 samples and 500 independent runs. Each run is performed on different sets of time varying weights of the Rayleigh channel which assumes scattering from 20 objects and observations are drawn from a Rayleigh density function. In each run, the weights of all the filters are initialized to zero. The parameters α and s used in RLS and E-RLS algorithms are set such that $\alpha = J_0(2f_D T_s)$ and $s = (1 - \alpha^2)I$, where J_0 is a zero order Bessel function of the first order and is defined as:

$$J(2\pi f_D T_s) = \frac{1}{\pi} \int_0^\pi \cos(2\pi f_D T_s \sin(\theta)) d\theta \quad (7.23)$$

where T_s is the sampling period of the input $x(k)$, and f_D is the Doppler frequency which is a function of carrier frequency f_c and speed v of the mobile user [5], that is,

$$f_D = \frac{v f_c}{c} \quad (7.24)$$

$c = 3 \times 10^8$ m/s is the speed of light.

Figure 7.2 shows the MSE learning performance for step sizes $\mu_l = 0.2$ and $\mu_f = 0.5$. The Doppler shift is kept as $f_D = 100$ Hz, and the sampling frequency is 1.25 MHz. The fractional order is set as: $\nu = 0.99$ and the forgetting factor as: $\lambda = 0.999$. It is seen that the RLS and E-RLS algorithms converge to the MSE value of -25 dB in 10 iterations while the F-NLMS v1 converges in 37 iterations and NLMS in 101 iterations. This clearly shows the convergence improvement over the NLMS algorithm(63.37%). The steady state MSE for the NLMS, F-NLMSv1, RLS and E-RLS is [-28.01, -27.32,-10.06,-10.13] dBs respectively. It can be seen that NLMS has better MSE followed by F-NLMS v1 algorithm.

Figure 7.3 shows the corresponding MSD learning curve for the same parameter settings. The steady state MSD for the NLMS, F-NLM v1, RLS and E-RLS algorithms are [-64.11,-57.45, -19.29,-19.44] dBs respectively. It can be seen that

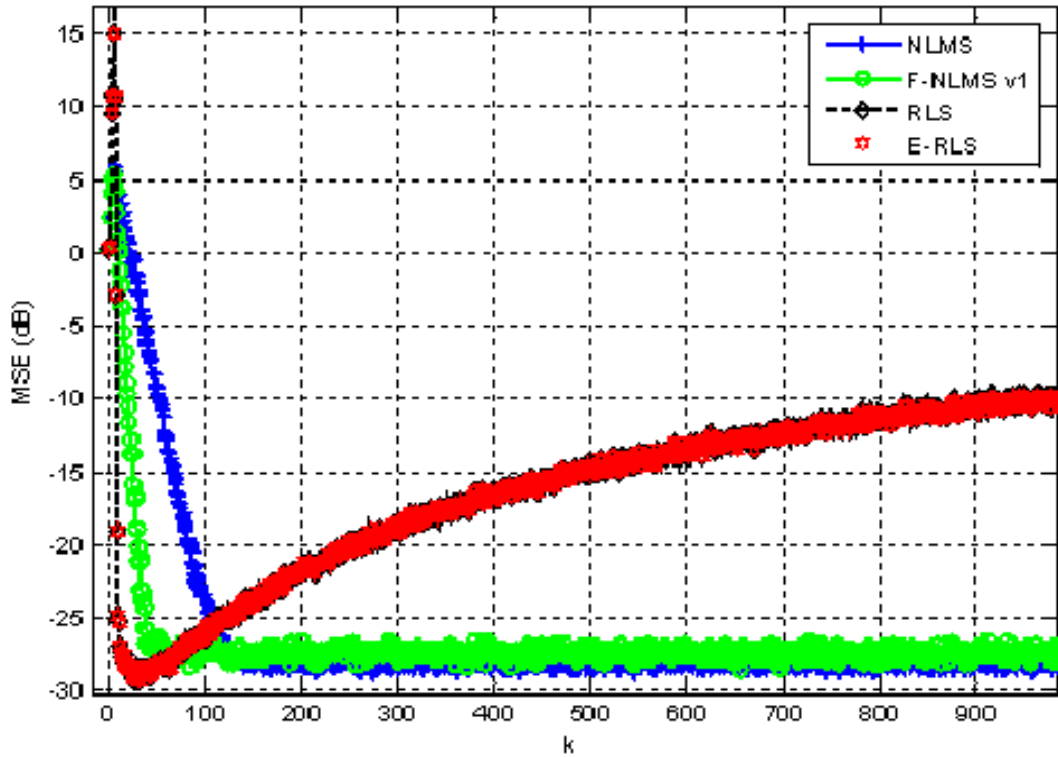


FIGURE 7.2: MSE Learning curves of NLMS, F-NLMSv1, RLS and E-RLS for $f_D = 100$ Hz.

RLS and E-RLS has essentially the same MSD performance. It is also seen that NLMS has better MSD followed by F-NLMS v1, however, the latter has superior convergence performance.

Figure 7.4 shows the MSE learning performance for $\mu_l = 0.2$ and $\mu_f = 0.3$. The Doppler shift $f_D = 400$ Hz, and the sampling frequency is 1.25 MHz. The fractional order is chosen to be $\nu = 0.85$, and $\lambda = 0.999$. It is seen that the RLS and E-RLS algorithms converge to the MSE value of -15 dB in 10 iterations while the F-NLMSv2 converges in 29 iterations and NLMS in 85 iterations. This clearly shows the convergence improvement of 65.88% over the NLMS algorithm. The steady state MSE values for the NLMS, F-NLMSv2, RLS and E-RLS algorithms are [-17.32, -16.67, 1.45, 0.42] dBs respectively. It can be seen that NLMS has better MSE followed by F-NLMS v2.

Figure 7.5 shows the corresponding MSD learning curve for the same parameter settings. The steady state MSD for the NLMS, F-NLMSv2, RLS and E-RLS algorithms are [-41.14, -34.78, 3.62, 1.53] dBs respectively. The RLS has the most degraded performance.

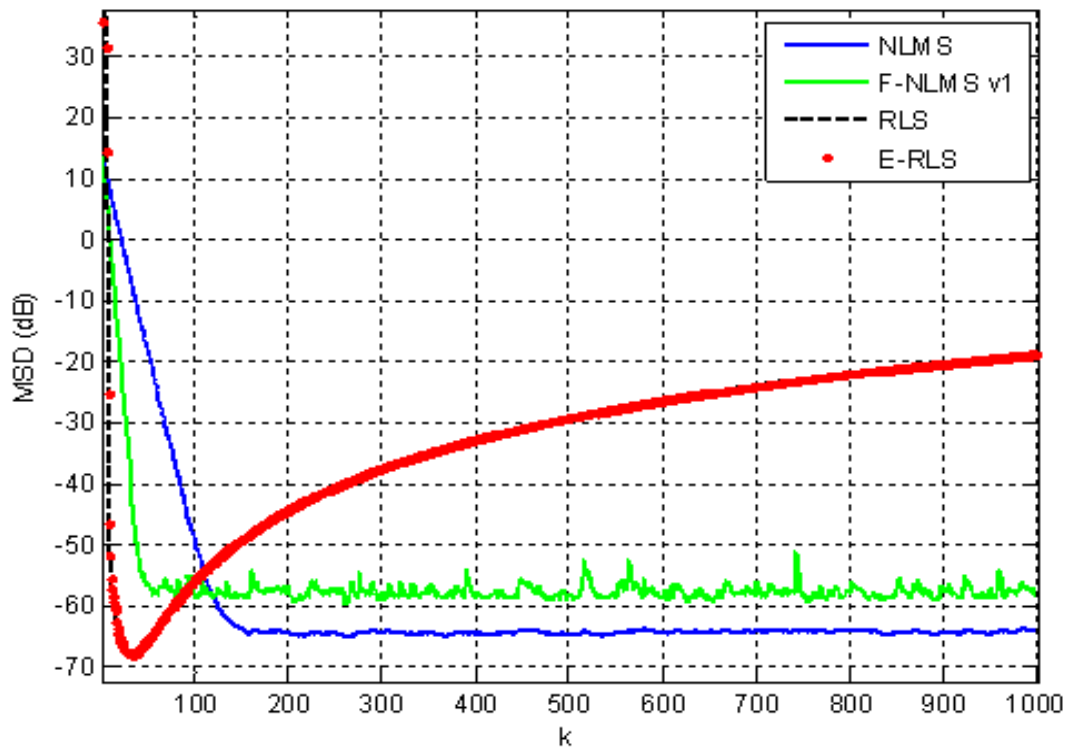


FIGURE 7.3: MSD Learning curves of NLMS, F-NLMSv1, RLS and E-RLS for $f_D = 100$ Hz.

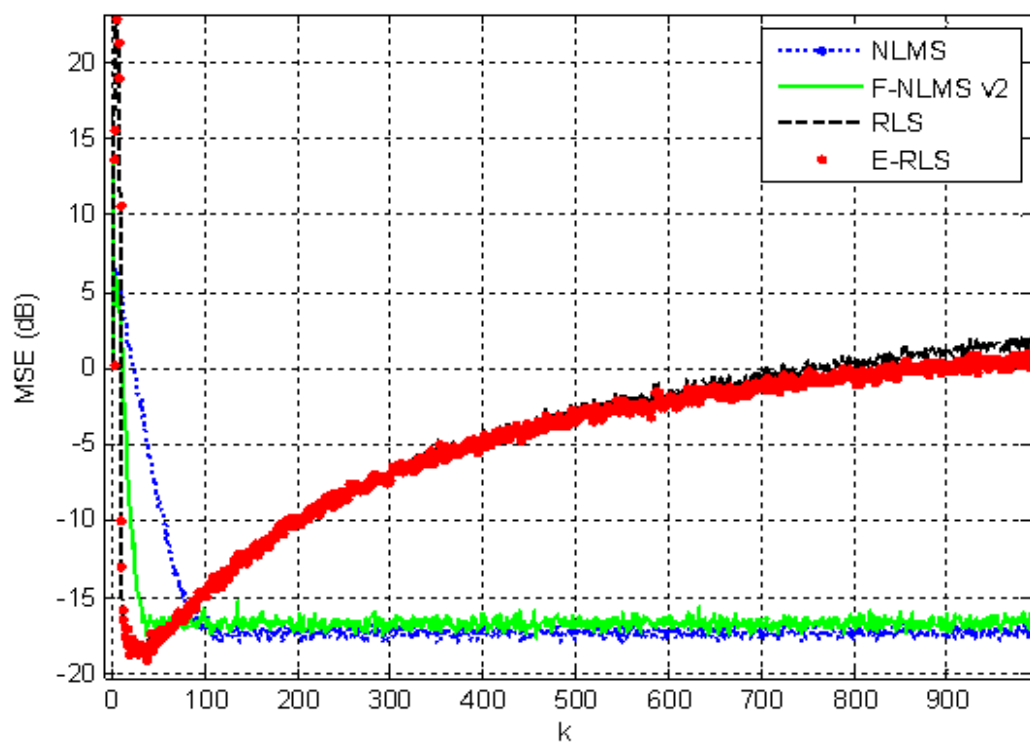


FIGURE 7.4: MSE Learning curves of NLMS, F-NLMSv2, RLS and E-RLS for $f_D = 400$ Hz.

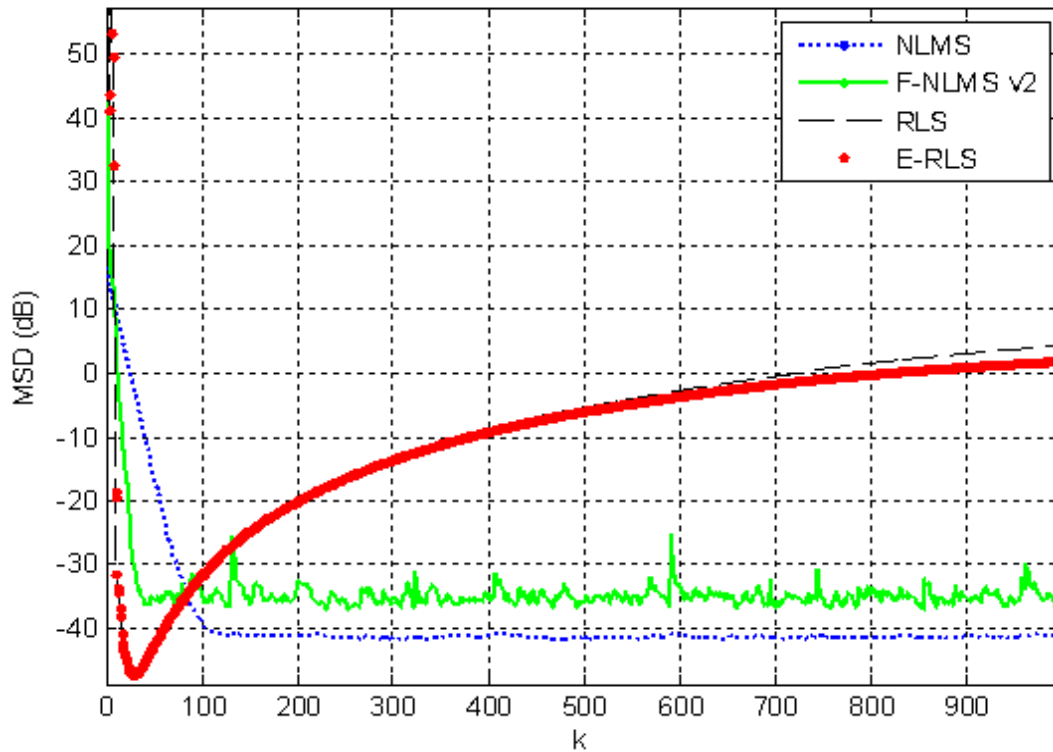


FIGURE 7.5: MSD Learning curves of NLMS, F-NLMSv2, RLS and E-RLS for $f_D = 400$ Hz.

Figure 7.6 shows the MSE learning performance for $\mu_l = 0.2$ and $\mu_f = 0.5$. The Doppler shift is 900Hz, fractional order is 0.95 and forgetting factor is 0.999. The steady-state MSE for NLMS, F-NLMSv3, RLS and E-RLS is [-13.51, -14.29, 6.11, 2.85] dBs while the corresponding MSD values are [-29.21, -27.14, 12.96, 6.43]dBs respectively. Noise is 20 dB below the input. Figure 7.7 shows the MSD learning curves for the same parameter settings. The RLS and E-RLS algorithms have severely degraded performance at such high Doppler shifts.

Figures 7.8 and 7.9 respectively show the MSE and MSD learning curves for $\mu_l = 0.2$ and $\mu_f = 0.5$. The Doppler shift is chosen to be 1300Hz, fractional order as 0.99 and forgetting factor is 0.999. The SNR is set at 20 dB. The steady-state MSE values for NLMS, F-NLMSv4, RLS and E-RLS are [-11.51, -21.83, 6.44, 2.95]dBs while the corresponding MSD values are [-23.27, -45.43, 13.81, 6.69]dBs respectively. Not only the RLS and E-RLS algorithms have severely degraded performance at such high Doppler shifts, the NLMS has also slow convergence and degraded steady state performance as compared to the F-NLMS v4 algorithm.

Figure 7.10 shows the tracking behavior with taps 1, 2, 4 and 5 for the above

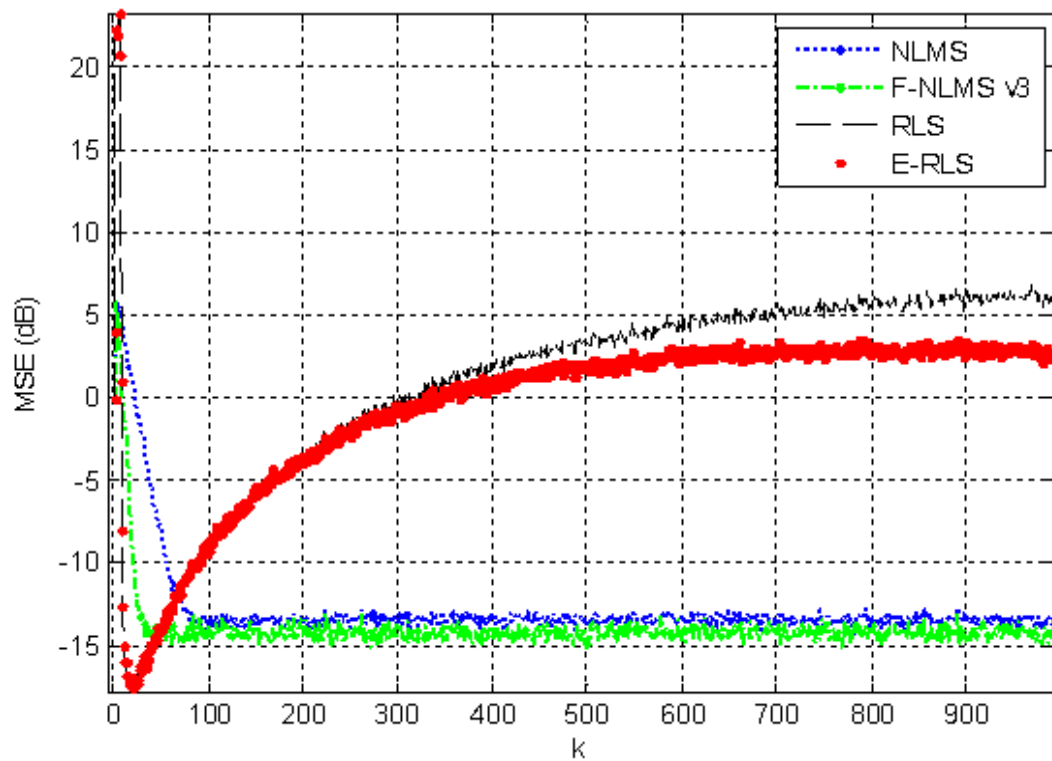


FIGURE 7.6: MSE Learning curves of NLMS, F-NLMSv3, RLS and E-RLS for $f_D = 900$ Hz.

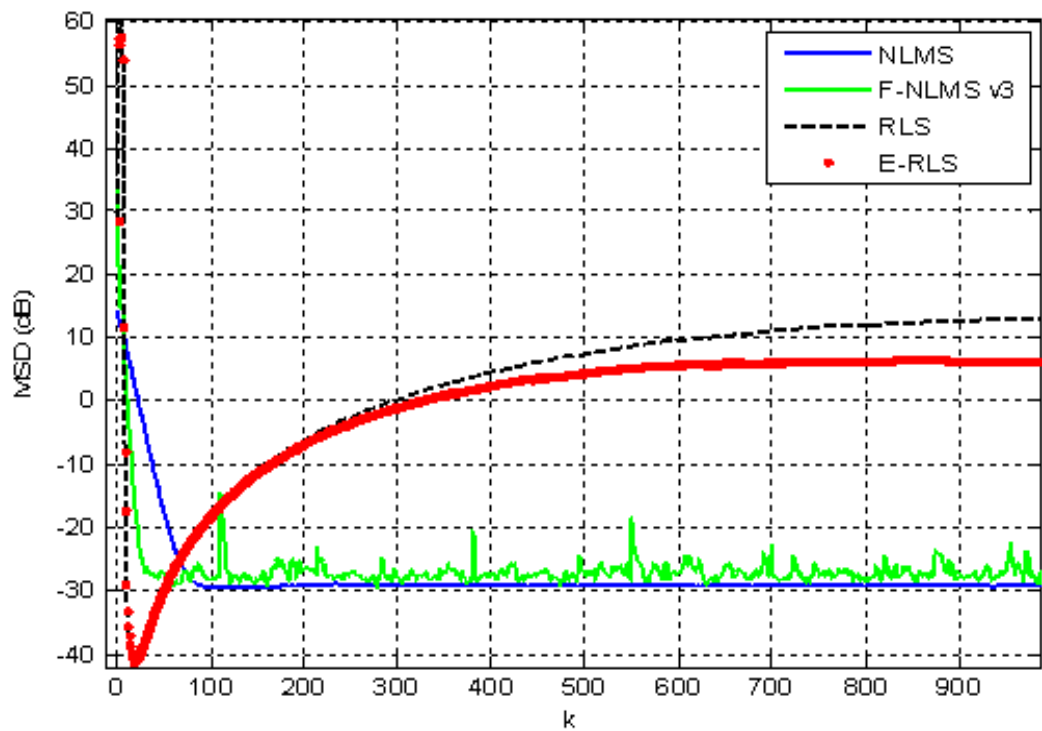


FIGURE 7.7: MSD Learning curves of NLMS, F-NLMSv3, RLS and E-RLS for $f_D = 900$ Hz.

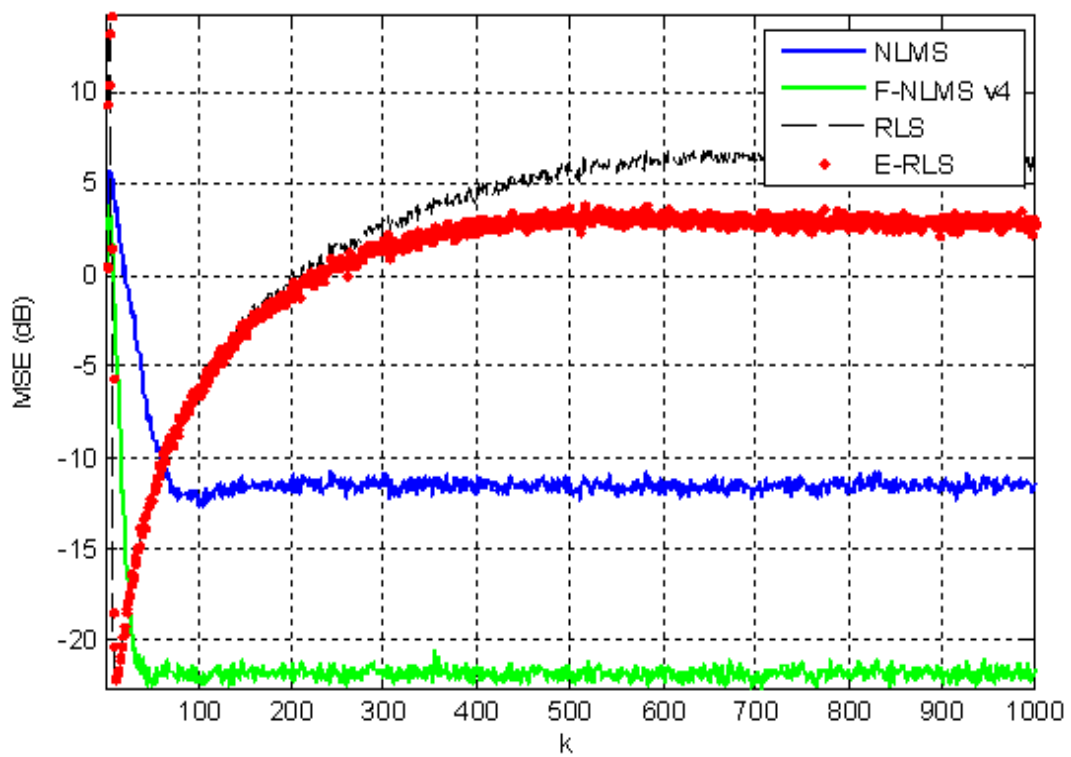


FIGURE 7.8: MSE Learning curves of NLMS, F-NLMSv4, RLS and E-RLS for $f_D = 1300$ Hz.

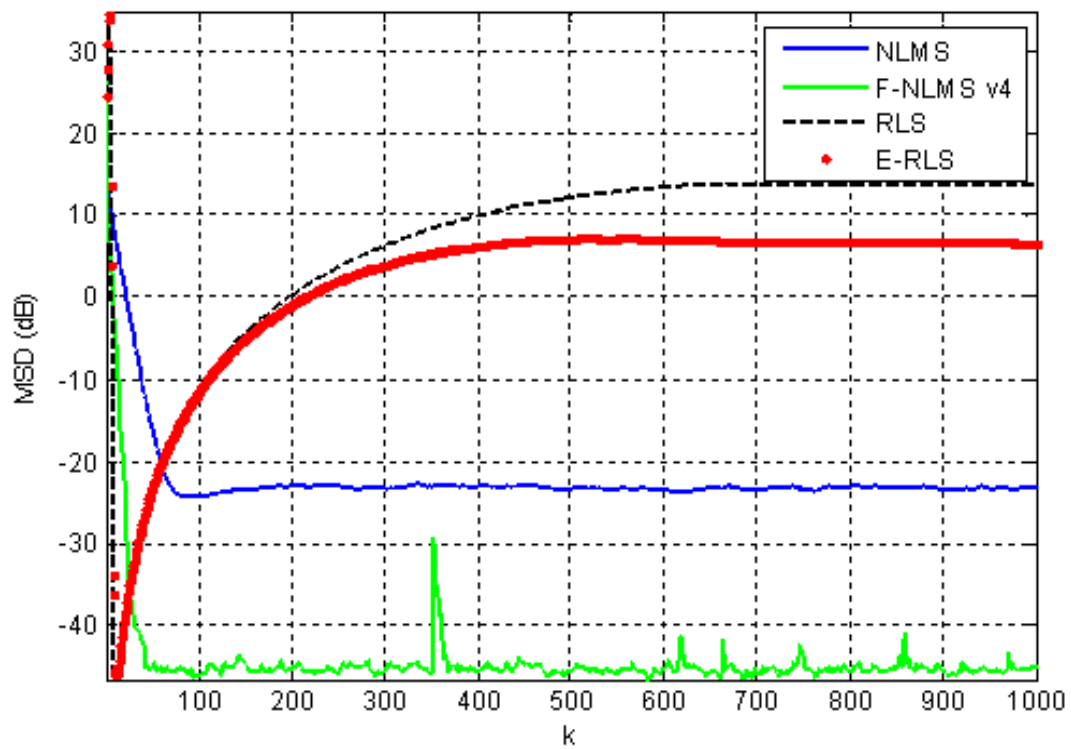


FIGURE 7.9: MSD Learning curves of NLMS, F-NLMSv4, RLS and E-RLS for $f_D = 1300$ Hz.

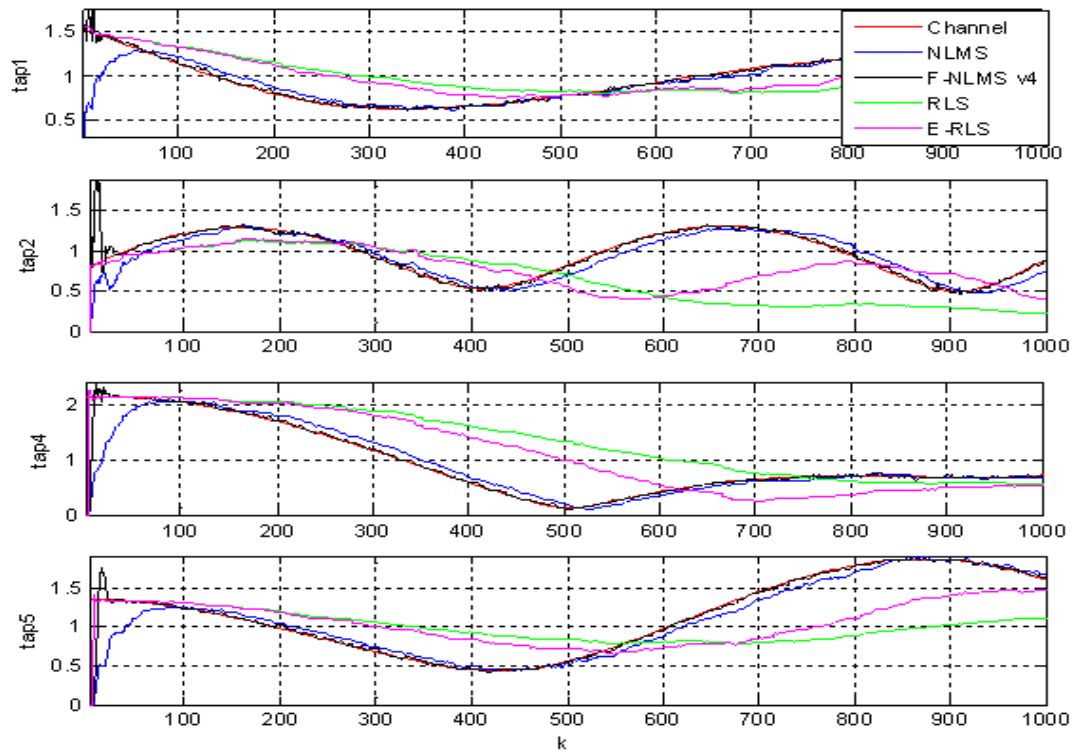


FIGURE 7.10: Tracking behavior of different taps of the algorithms.

parameter setting, the behavior of tap-3 is shown in Figure 7.11. The supremacy of F-NLMS v4 algorithm can be seen in these cases, the channel is almost perfectly followed by the tap weight. It can be noted that for the above cases, the parameters α and s have been kept small.

Figures 7.12 and 7.13 show the MSE and MSD Learning curves of NLMS, F-NLMSv4, RLS and E-RLS algorithms. In Figure 7.12, $\alpha = 0.95$, $s = 0.1$, while in Figure 7.13, $\alpha = 1$, $s = 0.1$, considering the generation of Rayleigh channel by deriving both of its real and imaginary components from independent and identically distributed processes, it can be noted that the performance of E-RLS and F-NLMS v4 is almost the same, followed by RLS and then NLMS, however, the complexity of F-NLMS v4 is smaller than the E-RLS algorithm.

Figure 7.14 shows the MSE and MSD learning curves of NLMS, F-NLMSv4, RLS and E-RLS algorithms in a hybrid configuration such that the weights of the NLMS and F-NLMS v4 algorithms are initialized with RLS weights till the first ten iterations and thereafter in the usual independent way of learning. The parameters are set as: $\alpha = 1$, $s = 0.1$, $\mu_l = 0.2$, $\mu_f = 0.5$, $f_D = 1500$ Hz, $\nu = 0.99$, $\lambda = 0.999$.

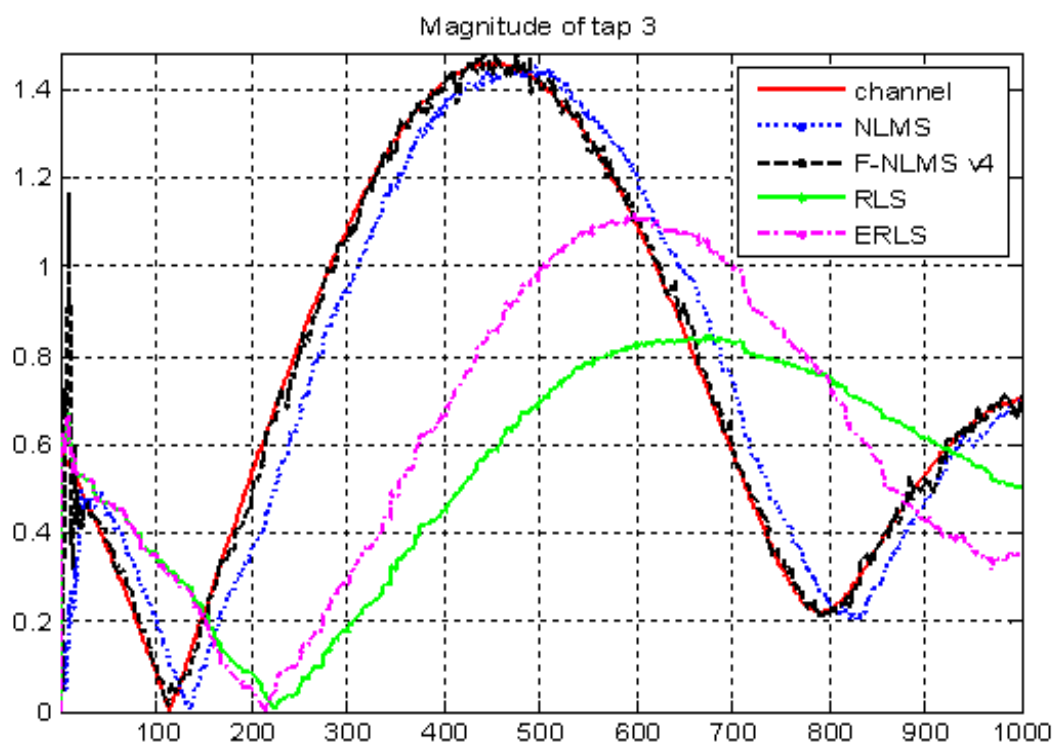


FIGURE 7.11: Tracking behavior of tap-3 for different algorithms.

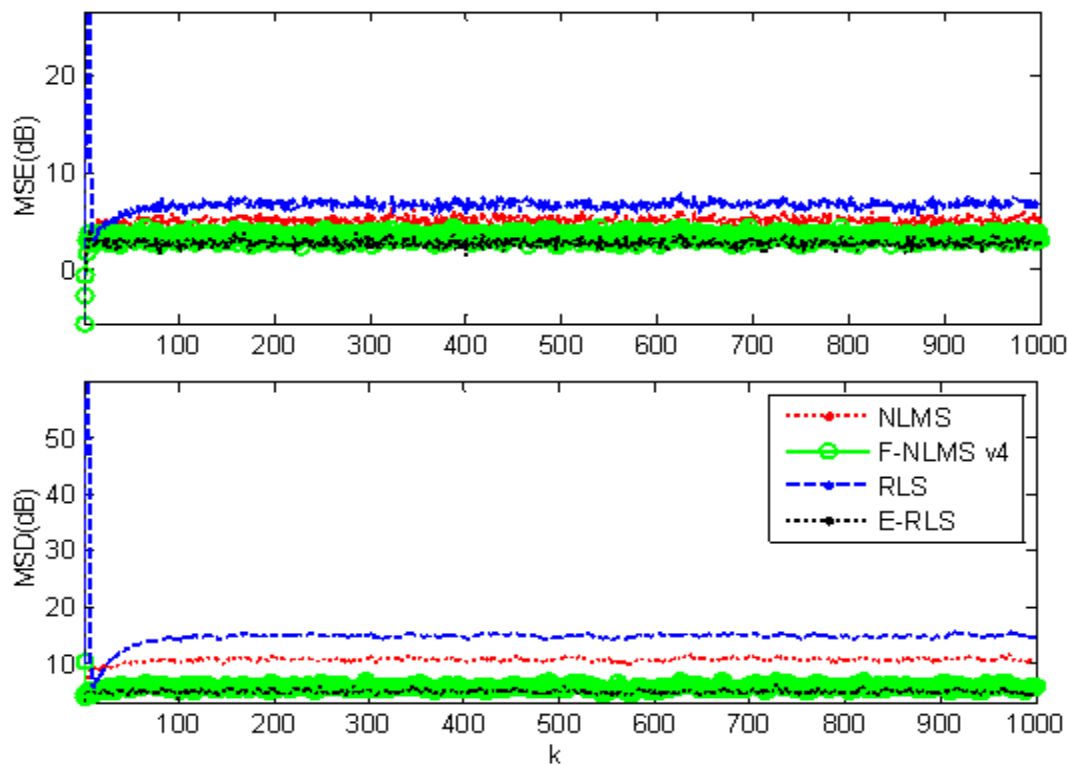


FIGURE 7.12: MSE and MSD Learning curves of NLMS, F-NLMSv4, RLS and E-RLS.

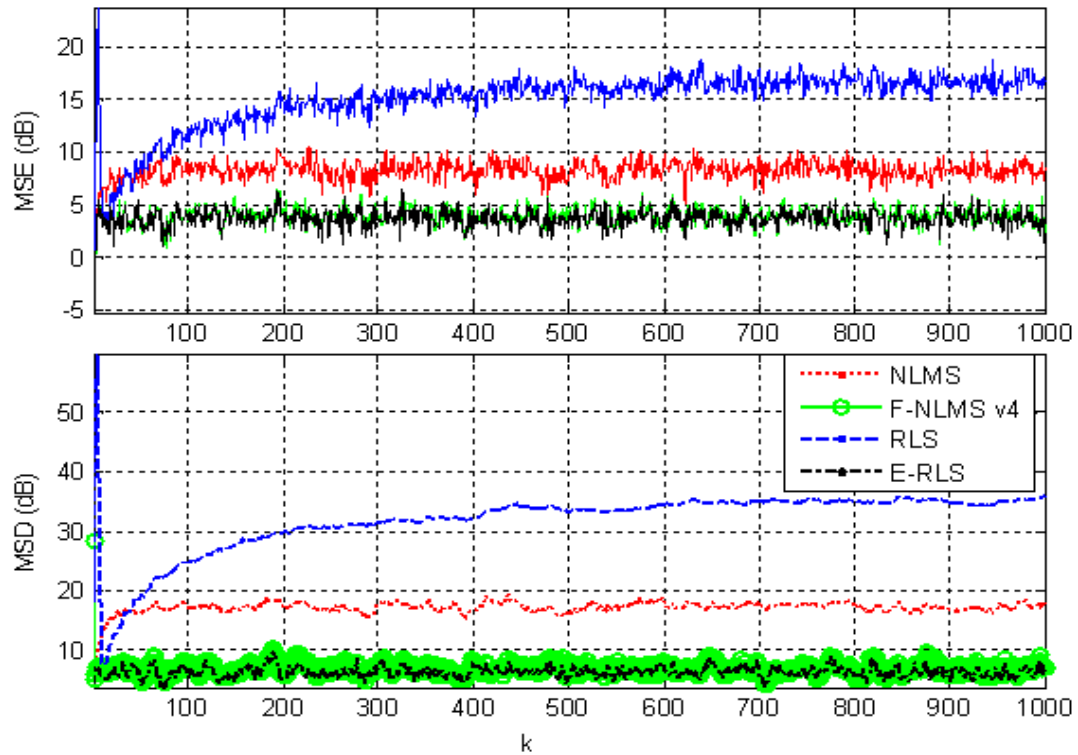


FIGURE 7.13: MSE and MSD Learning curves of NLMS, F-NLMSv4, RLS and E-RLS.

The additive noise is 30 dB below input which is 0 dB. It can be noted that F-NLMS v4 has the best performance followed by NLMS while RLS and E-RLS have degraded performance. Figure 7.15 shows the taps tracking behavior.

7.6 Conclusions

In this chapter, the tracking performance of the proposed fractional order variants was compared with its traditional counterparts as well as with the fast convergent RLS and E-RLS algorithms. The input was a time varying channel sequence based on different Doppler shifts, modelled by the Rayleigh distribution. The MSE and MSD performances were compared for the NLMS, F-NLMS (different versions), RLS and E-RLS algorithms. It was found that the F-NLMS variants mostly show superior performance in the steady state even at higher Doppler shifts, in such cases all other algorithms fail to perform. The RLS and E-RLS algorithms mostly converge in about eight iterations, the FNLMS variants converge in 30-35 iterations

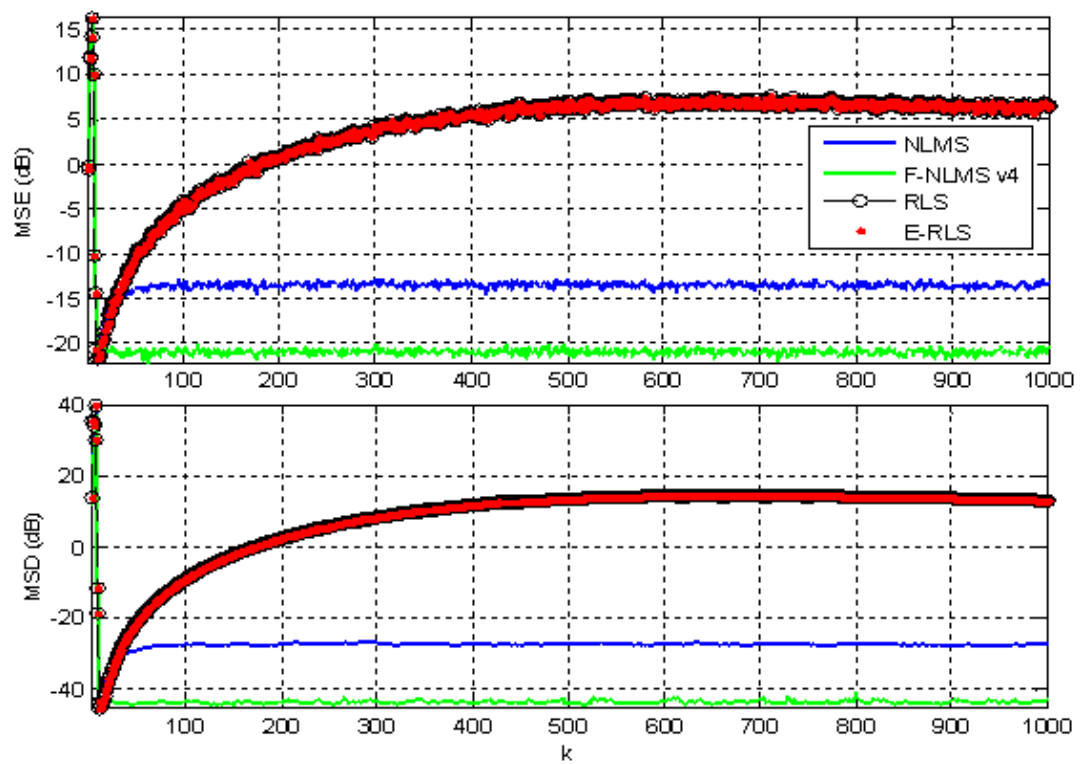


FIGURE 7.14: MSE and MSD Learning curves of NLMS, F-NLMSv4, RLS and E-RLS.

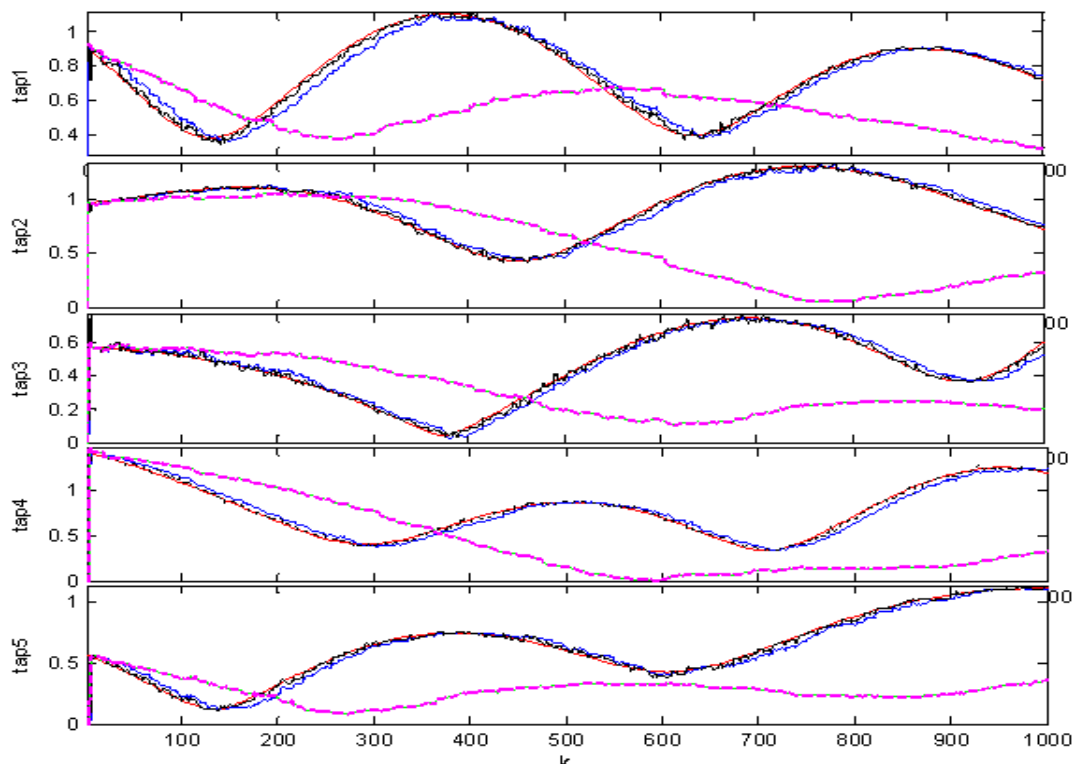


FIGURE 7.15: Tracking behavior of different taps of the algorithms in the hybrid case.

while the NLMS algorithm converges in 120 iterations. In the steady state NLMS and FNLMS algorithms have almost similar performance, but RLS and E-RLS diverge (especially for Doppler shifts greater than 100Hz). In the Hybrid case, the FNLMS algorithm matches the convergence of RLS and exhibits steady state performance better than both RLS and NLMS algorithms. The RLS and E-RLS have slightly faster convergence rates but degraded steady state behavior. FNLMS has better convergence as well as steady state response than NLMS for higher Doppler shifts.

Chapter 8

Summary and Future Work

The dissertation developed various fractional order adaptive signal processing algorithms which were evaluated in different applications. The performance was measured using standard signal processing as well as application specific metrics. Based on the simulation results, it is observed that the fractional order signal processing approaches have the potential of obtaining superior performance as compared to the integer order classical algorithms. The rest of this chapter summarizes the conclusions and presents some possible future work directions.

8.1 Summary and Conclusions

In this thesis, fractional adaptive signal processing algorithms have been applied in four applications, namely (1) linear and Decision Feedback Equalization (DFE) (2) constant modulus algorithm (CMA) based blind equalization (3) Active Noise Control Systems (ANCS) and (4) tracking of time varying Rayleigh fading sequences.

Firstly, a fractional order variant of the normalized least mean square (FNLMS) algorithm has been proposed for the DFE of multipath fading channels; it outperforms its traditional counterpart. Application of fractional calculus in combination with the standard first order derivative allows improvement of symbol error rate

performance by applying a more generalized modelling framework. The new hybrid filter structure exhibits better convergence and steady state performance, and has greater flexibility in terms of tuning parameters. Based on simulation results while considering different types of channels (such as frequency-flat and selective), it could be inferred that the FNLMS gives extra freedom because of its two step sizes options and choice of fractional orders, for better inverse modeling of the multipath channel. This can help increase the link efficiency not only by using higher order modulation schemes, but lesser number of training symbols and therefore, can reduce transmission power (to achieve acceptable symbol error rate at low SNR values).

Next, a new blind adaptive algorithm has been developed which exploits fractional derivatives in minimizing the Godard cost function, this is applied to the inverse system identification problem of channel equalization. The proposed algorithm has faster convergence and improved mean squared error performance, it achieves enhanced equalization and can better alleviate the effects of additive white noise. For larger step sizes, the conventional CMA algorithm can give a poorer MSE and may also diverge, whereas the proposed algorithm exhibits better steady state behavior.

For ANCS, five fractional algorithms have been developed, namely Fractional LMS (FrLMS), Fractional filtered-x LMS (FrFxLMS), Fractional Modified FxLMS (FrMFxLMS), Fractional Filtered-error LMS (FrFeLMS) and Fractional Normalized FeLMS (FN-FeLMS). It is seen that the FN-FeLMS converges much faster for both binary and Gaussian input data than the standard FeLMS. The MSE and MSD performance metrics were measured for different fractional orders and step sizes. Increasing the fractional order helps in improving the convergence speed at the expense of steady state performance.

Having noticed the convergence and steady state performance of the FN-FeLMS algorithm, new hybrid algorithms were designed to generate fractional alternatives of FxLMS and its variants. The algorithms were applied in ANCS using different fractional orders, and performance compared to their traditional counterparts as well as to FN-FeLMS. The algorithms were checked for different types of inputs,

especially impulsive noise which is modelled by fractional lower order moments or stable processes (both symmetric and asymmetric). The algorithms are also considered for different primary path models with various tap lengths. Results showed that fractional variants converge faster for different types of input noise such as binary and Gaussian. The steady-state performance for the designed algorithms is almost the same as their conventional counterparts. It is seen that FrFeLMS shows the most improved convergence performance over FeLMS, followed by FrFxLMS over FxLMS. The improvements are of the order of 75-80% for the same step size. In case of FrMFXLMS, the convergence improvement is approximately 45% in all cases.

Finally, the tracking performance of the proposed fractional order variants were compared with the traditional counterparts as well as with the fast convergence recursive least squares (RLS) and the extended RLS (eRLS) algorithms. The input was a time varying channel sequence based on different Doppler shifts and modelled by Rayleigh distribution. The MSE and MSD performance was compared for the NLMS, FNLMS, FrLMS, RLS and eRLS algorithms. It was found that the FrLMS algorithm has superior performance in the steady state even at higher Doppler shifts, in which case all other algorithms fail to perform. The RLS and eRLS have slightly faster convergence rates but degraded steady state behavior. It is worth mentioning that the FrLMS has better convergence and steady state behavior than NLMS for higher Doppler shifts.

8.2 Future Work Directions

The proposed algorithms can be seen as a useful addition in the field of adaptive (fractional order) signal processing which is open to further research. The newly designed algorithms have attraction for wide applications since adaptive signal processing techniques have been applied in a variety of fields such as bio-medical, control and instrumentation, communication and defence. These include (but not limited to) (1) system identification such as channel estimation, system model identification, active noise control systems, line and acoustic echo cancellations,

sensing, estimation and target localization in distributed environments, etc. (2) inverse system modelling such as adaptive channel equalization and deconvolution (3) spatial filtering (beamforming) such as interference and noise cancellation in Radars, Sonars and speech enhancement, just to name a few. Some of the possible future research directions could be the following:

- **Block Filters:** Fourier Transform (FT) of fractional FT (FrFT) based variants can be developed to have fractional order frequency domain adaptive filters and sub-band adaptive filters. These filters can be applied in acoustics such as RADARs, SONARs, Active noise control systems, line / acoustic echo cancellation, etc.
- **Channel Estimation, Equalization or Joint Channel Estimation and Equalization** in Orthogonal Frequency Division Multiplexing (OFDM) or Code Division Multiple Access (CDMA) schemes as these are among the latest communication technologies for higher data rates and to accommodate more users. These schemes can be evaluated with maximum likelihood sequence estimation techniques based on soft input soft output (soft output Viterbi Algorithm) or Turbo Codes.
- **Numerical Stability Analysis:** it is one aspect of checking the performance of fractional order adaptive filters. This is usually required for fixed point implementation of the algorithms. The performance can be compared with other classical algorithms.
- **H-infinity Robustness Analysis:** the stability and robustness of LMS algorithm was first established using the H-Infinity approach, similar techniques might be helpful to investigate the robustness and stability of fractional order adaptive filters.
- **Convergence and Tracking Analysis:** Mathematical proofs based on mean squared error, mean squared deviation and tracking performance analysis. This may require upper bounds on both the step sizes as well as the best

performance for a given fractional order. A distributed fractional order approach may also be applied where each fractional order can be optimized.

- Variants Based on Exact Chain Rule or Product Rule of Fractional Derivatives: Since an FD is a generalization of the integer derivative, it is going to lose many of its basic properties. For example, losing its geometric or physical interpretation, the derivative of the product of two functions is difficult to compute, and the chain rule cannot be straightforwardly applied. It is recommended to explore the application of these techniques for adaptive signal processing when such limitations are explored further.
- Diffusion LMS and application in WSNs: Fractional order variant (s) of diffusion LMS can be developed with target applications of distributed sensing and estimation, online machine learning, intrusion detection and target localization, etc. An example could be estimation of distributed channel gains in wireless sensor networks.
- Fractional order variants of Volterra LMS, eNLMS, LMF or VSSLMS algorithms can be developed and applied to various applications. These variants can also be based on Mittag-Leffler function.
- Applications of other fractional derivative definitions especially the centered fractional derivatives in fractional order adaptive signal processing.
- Optimization of adaptive signal processing especially when the performance metrics or signals are characterized by fractional lower order moments.

Bibliography

- [1] S. M. Kuo and D. R. Morgan, *Active noise control systems: algorithms and DSP implementations*. John Wiley & Sons, Inc., 1995.
- [2] D. R. Morgan, “History, applications, and subsequent development of the fxlms algorithm [dsp history],” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 172–176, 2013.
- [3] 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, 2013.
- [4] B. Widrow, D. Shur, and S. Shaffer, “On adaptive inverse control.” Proceeding of the 15th Asilomar Conference on Circuits, Systems and Computers,, 1981, pp. 185–189.
- [5] A. H. Sayed, *Adaptive Filters*. Wiley Interscience, June 2008.
- [6] S. R. Diniz, *Adaptive Filtering, Algorithms and Practical Implementations*. Springer Publisher, June 2008.
- [7] V. E. Tarasov, “No nonlocality. no fractional derivative,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 62, pp. 157–163, 2018.
- [8] V. V. Tarasova and V. E. Tarasov, “Concept of dynamic memory in economics,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 55, pp. 127–145, 2018.
- [9] B. Dumitru, D. Kai, and S. Enrico, *Fractional Calculus: Models And Numerical Methods*, second edition ed., vol. 3.

-
- [10] K. R. Martin, *Adaptive Equalization for wireless channels*. In Ibnkahla, M.: (Ed.). *Adaptive Signal Processing in Wireless Communications, Adaptation and cross layer design in wireless networks*. CRC Press, June 2009.
- [11] E. Eweda, "Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels," *Signal Processing, IEEE Transactions on*, vol. 42, no. 11, pp. 2937–2944, Nov 1994.
- [12] G. Gui, W. Peng, and F. Adachi, "Adaptive system identification using robust lms/f algorithm," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 2956–2963, 2014.
- [13] S. M. Jung and P. Park, "Normalised least-mean-square algorithm for adaptive filtering of impulsive measurement noises and noisy inputs," *Electronics Letters*, vol. 49, no. 20, pp. 1270–1272, 2013.
- [14] S. Abrar and A. K. Nandi, "Blind equalization of square-qam signals: a multimodulus approach," *IEEE Transactions on Communications*, vol. 58, pp. 1674–1685, 2010.
- [15] J. C. Burgess, "Active adaptive sound control in a duct: A computer simulation," *The Journal of the Acoustical Society of America*, vol. 70, no. 3, pp. 715–726, 1981.
- [16] D. Chang and F. Chu, "A new variable tap-length and step-size fxlms algorithm," *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1122–1125, 2013.
- [17] S. Elliott, I. Stothers, and P. Nelson, "A multiple error lms algorithm and its application to the active control of sound and vibration," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1423–1434, 1987.
- [18] G. E. Warnaka, L. A. Poole, and J. Tichy, "Active acoustic attenuator," Sep 1984, uS Patent 4,473,906.

- [19] A. V. Oppenheim, E. Weinstein, K. C. Zangi, M. Feder, and D. Gauger, "Single-sensor active noise cancellation," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 285–290, 1994.
- [20] Q. Shen and A. S. Spanias, "Time and frequency domain x block lms algorithms for single channel active noise control," *Proc. 2nd Int. Congr. Recent Developments in Air-and Structure-Borne Sound Vibration*, pp. 353–360, 1992.
- [21] K. M. Reichard and D. C. Swanson, "Frequency-domain implementation of the filtered-x algorithm with on-line system identification," in *Proc. Recent Advances in Active Control of Sound Vibration*, 1993, pp. 562–573.
- [22] S. J. Park, J. H. Yun, Y. C. Park, and D. H. Youn, "A delayless subband active noise control system for wideband noise control," *IEEE transactions on speech and audio processing*, vol. 9, no. 8, pp. 892–899, 2001.
- [23] V. DeBrunner, L. S. DeBrunner, and L. Wang, "Sub-band adaptive filtering with delay compensation for active control," *IEEE transactions on signal processing*, vol. 52, no. 10, pp. 2932–2941, 2004.
- [24] B. Siravara, N. Magotra, and P. Loizou, "A novel approach for single microphone active noise cancellation," in *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on*, vol. 3. IEEE, 2002, pp. 469–472.
- [25] M. Rupp, "Convergence properties of adaptive equalizer algorithms," *Signal Processing, IEEE Transactions on*, vol. 59, no. 6, pp. 2562–2574, June 2011.
- [26] M. Rupp and J. Garcia-Naya, "Equalizers in mobile communications: Tutorial 38," *Instrumentation Measurement Magazine, IEEE*, vol. 15, no. 3, pp. 32–42, June 2012.
- [27] W. Ping, F. Pingzhi, Y. Weina, and M. Darnell, "Data detection and coding for data-dependent superimposed training," *Signal Processing, IET*, vol. 8, no. 2, pp. 138–145, April 2014.

- [28] B. Shoaib and I. M. Qureshi, "Adaptive step-size modified fractional least mean square algorithm for chaotic time series prediction," *Chinese Physics B*, vol. 23, no. 5, p. 050503, 2014.
- [29] L. Lindbom, M. Sternad, and A. Ahlen, "Tracking of time-varying mobile radio channels-part i: The Wiener LMS Algorithm," *Communications, IEEE Transactions on*, vol. 49, no. 12, pp. 2207–2217, Dec 2001.
- [30] T. A. I. and W. H. Abdulla, "Root locus analysis and design of the adaptation process in active noise control," *The Journal of the Acoustical Society of America*, vol. 132, no. 4, pp. 2313–2324, 2012.
- [31] T. A. I. and H. Waleed Abdulla, "Effects of imperfect secondary path modeling on adaptive active noise control systems," *IEEE transactions on control systems technology*, vol. 20, no. 5, pp. 1252–1262, 2012.
- [32] M. Rupp and A. H. Sayed, "Two variants of the fxlms algorithm," in *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*. IEEE, 1995, pp. 123–126.
- [33] M. R. and A. H. Sayed, "A time-domain feedback analysis of filtered-error adaptive gradient algorithms," *IEEE transactions on Signal Processing*, vol. 44, no. 6, pp. 1428–1439, 1996.
- [34] M. Rupp and H. S. Ali, "Robust fxlms algorithms with improved convergence performance," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 1, pp. 78–85, 1998.
- [35] A. Gonzalez, M. Ferrer, M. de Diego, and G. Pinero, "Fast filtered-x affine projection algorithm for active noise control," in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*. IEEE, 2005, pp. 162–165.
- [36] H. Zhao, X. Zeng, X. Zhang, Z. He, T. Li, and W. Zhao, "Adaptive extended pipelined second-order volterra filter for nonlinear active noise controller,"

- IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1394–1399, 2012.
- [37] M. S. Aslam and M. A. Z. Raja, “A new adaptive strategy to improve online secondary path modeling in active noise control systems using fractional signal processing approach,” *Signal Processing*, vol. 107, pp. 433–443, 2015.
- [38] D. Chang and F. Chu, “Feedforward active noise control with a new variable tap-length and step-size filtered-x lms algorithm,” *IEEE/acm transactions on audio, speech, and language processing*, vol. 22, no. 2, pp. 542–555, 2014.
- [39] L. V. Wang, W. Gan, A. W. H. Khong, and S. M. Kuo, “Convergence analysis of narrowband feedback active noise control system with imperfect secondary path estimation,” *IEEE transactions on audio, speech, and language processing*, vol. 21, no. 11, pp. 2403–2411, 2013.
- [40] C. Komninakis, C. Fragouli, A. H. Sayed, and R. D. Wesel, “Multi-input multi-output fading channel tracking and equalization using kalman estimation,” *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1065–1076, 2002.
- [41] J. Y. Hua, D. H. Yuan, G. Li, and L. M. Meng, “Accurate estimation of doppler shift in mobile communications with high vehicle speed,” *International Journal of Communication Systems*, vol. 27, no. 12, pp. 3515–3525, 2014.
- [42] I. Podlubny, *Fractional differential equations*. Academic Press, San Diego-Boston-New York-London-Tokyo-Toronto, 1999.
- [43] K. Oldham and J. Spanier, *The fractional calculus theory and applications of differentiation and integration to arbitrary order*. Elsevier, 1974, vol. 111.
- [44] H. Sheng, Y. Chen, and T. Qiu, *Fractional Processes and Fractional-Order Signal Processing: Techniques and Applications*. Springer, 2011.

- [45] R. E. Gutiérrez, J. M. Rosário, and J. Tenreiro Machado, “Fractional order calculus: basic concepts and engineering applications,” *Mathematical Problems in Engineering*, vol. 2010, 2010.
- [46] V. Badri and M. Tavazoei, “On tuning FO [PI] controllers for FOPDT processes,” *Electronics Letters*, vol. 49, no. 21, pp. 1326–1328, 2013.
- [47] I. S. Jesus and J. A. T. Machado, “Development of fractional order capacitors based on electrolyte processes,” *Nonlinear Dynamics*, vol. 56, no. 1-2, pp. 45–55, 2009.
- [48] N. I. Chaudhary, M. A. Z. Raja, and A. R. Khan, “Design of modified fractional adaptive strategies for hammerstein nonlinear control autoregressive systems,” *Nonlinear Dynamics*, vol. 82, no. 4, pp. 1811–1830, 2015.
- [49] Y. Zhou, C. Ionescu, and J. A. T. Machado, “Fractional dynamics and its applications,” *Nonlinear Dynamics*, vol. 80, no. 4, pp. 1661–1664, 2015.
- [50] X. Yang, J. A. T. Machado, and J. Hristov, “Nonlinear dynamics for local fractional burgers’ equation arising in fractal flow,” *Nonlinear Dynamics*, vol. 84, no. 1, pp. 3–7, 2016.
- [51] A. H. Bhrawy, M. Taha, and J. A. T. Machado, “A review of operational matrices and spectral techniques for fractional calculus,” *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1023–1052, 2015.
- [52] C. C. Tseng and S. L. Lee, “Design of digital riesz fractional order differentiator,” *Signal Processing*, vol. 102, pp. 32–45, 2014.
- [53] C. Tseng and S. Lee, “Design of adjustable fractional order differentiator using expansion of ideal frequency response,” *Signal Processing*, vol. 92, no. 2, pp. 498–508, 2012.
- [54] D. Wei and Y. Li, “Novel tridiagonal commuting matrices for types i, iv, v, viii dct and dst matrices,” *IEEE signal processing letters*, vol. 21, no. 4, pp. 483–487, 2014.

- [55] J. A. T. Machado, "Optimal tuning of fractional controllers using genetic algorithms," *Nonlinear Dynamics*, vol. 62, no. 1-2, pp. 447–452, 2010.
- [56] S. Victor, R. Malti, H. Garnier, and A. Oustaloup, "Parameter and differentiation order estimation in fractional models," *Automatica*, vol. 49, no. 4, pp. 926–935, 2013.
- [57] R. Duma, P. Dobra, and M. Trusca, "Embedded application of fractional order control," *Electronics Letters*, vol. 48, no. 24, pp. 1526–1528, 2012.
- [58] A. Charef and T. Bensouici, "Digital fractional delay implementation based on fractional order system," *IET signal processing*, vol. 5, no. 6, pp. 547–556, 2011.
- [59] K. A. Moornani and M. Haeri, "Robustness in fractional proportional–integral–derivative-based closed-loop systems," *IET control theory & applications*, vol. 4, no. 10, pp. 1933–1944, 2010.
- [60] Z. Gao and X. Liao, "Robust stability criterion of fractional-order functions for interval fractional-order systems," *IET Control Theory & Applications*, vol. 7, no. 1, pp. 60–67, 2013.
- [61] H. Shen, X. Song, and Z. Wang, "Robust fault-tolerant control of uncertain fractional-order systems against actuator faults," *IET Control Theory & Applications*, vol. 7, no. 9, pp. 1233–1241, 2013.
- [62] D. Wei and Y. Li, "Sampling reconstruction of n-dimensional bandlimited images after multilinear filtering in fractional fourier domain," *Optics Communications*, vol. 295, pp. 26–35, 2013.
- [63] Wei and Y. Li, "Reconstruction of multidimensional bandlimited signals from multichannel samples in linear canonical transform domain," *IET Signal Processing*, vol. 8, no. 6, pp. 647–657, 2014.

- [64] L. J. Eriksson, M. C. Allie, D. E. Melton, S. R. Popovich, and T. A. Laak, "Fully adaptive generalized recursive control system for active acoustic attenuation," in *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. 2. IEEE, 1994, pp. 241–253.
- [65] F. J. C. Manuel D. Ortigueira and J. J. Trujillo, "Discrete-time differential systems," *Signal Processing*, 2014.
- [66] S. M. Shah, R. Samar, S. M. Naqvi, and J. A. Chambers, "Fractional order constant modulus blind algorithms with application to equalisation," *Electronics Letters*, vol. 50, pp. 1702–1704, July 2014.
- [67] S. Shah, R. Samar, M. Raja, and J. Chambers, "Fractional normalised filtered-error least mean squares algorithm for application in active noise control systems," *Electronics Letters*, vol. 50, no. 14, pp. 973–975, July 2014.
- [68] R. Muhammad A. Z. and N. I. Chaudhary, "Adaptive strategies for parameter estimation of box–jenkins systems," *IET Signal Processing*, vol. 8, no. 9, pp. 968–980, 2014.
- [69] M. A. Z. Raja and N. I. Chaudhary, "Two-stage fractional least mean square identification algorithm for parameter estimation of carma systems," *Signal Processing*, vol. 107, pp. 327–339, 2015.
- [70] Y. Tan, Z. He, and B. Tian, "A novel generalization of modified lms algorithm to fractional order," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1244–1248, 2015.
- [71] S. M. Shah, R. Samar, and M. A. Z. Raja, "Fractional-order algorithms for tracking of rayleigh fading channels," *Nonlinear Dynamics*, pp. 1–17, 2018.
- [72] R. Malti, S. Victor, and A. Oustaloup, "Advances in system identification using fractional models," *Journal of Computational and Nonlinear Dynamics*, vol. 3, no. 2, pp. 021 401–021 401–7, 2008.

- [73] M. D. Ortigueira and F. J. Coito, "On the usefulness of Riemann-Liouville and Caputo derivatives in describing fractional shift-invariant linear systems," *Journal of Applied Nonlinear Dynamics*, vol. 1, pp. 113–124, 2012.
- [74] M. D. Ortigueira and J. Machado, "Fractional signal processing and applications," *Signal processing*, vol. 83, no. 11, pp. 2285–2286, 2003.
- [75] A. Charef and D. Idiou, "Design of analog variable fractional order differentiator and integrator," *Nonlinear Dynamics*, vol. 69, no. 4, pp. 1577–1588, 2012.
- [76] M. Aoun, R. Malti, F. Levron, and A. Oustaloup, "Numerical simulations of fractional systems: an overview of existing methods and improvements," *Nonlinear Dynamics*, vol. 38, no. 1-4, pp. 117–131, 2004.
- [77] C. F. Mecklenbrauker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernadó, T. Zemen, O. Klemp, and N. Czink, "Vehicular channel characterization and its implications for wireless system design and performance," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1189–1212, 2011.
- [78] J. Karedal, N. Czink, A. Paier, F. Tufvesson, and A. F. Molisch, "Path loss modeling for vehicle-to-vehicle communications," *IEEE transactions on vehicular technology*, vol. 60, no. 1, pp. 323–328, 2011.
- [79] S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.
- [80] M. Rupp, "Robust design of adaptive equalizers," *Signal Processing, IEEE Transactions on*, vol. 60, no. 4, pp. 1612–1626, April 2012.
- [81] J. Hartmanis and R. E. Stearns, "On the computational complexity of algorithms," *Transactions of the American Mathematical Society*, vol. 117, pp. 285–306, 1965.
- [82] G. Gui and F. Adachi, "Sparse least mean fourth algorithm for adaptive channel estimation in low signal-to-noise ratio region," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3147–3157, 2014.

- [83] J. Y. B. Kang and P. Park, "Bias-compensated normalised lms algorithm with noisy input," *Electronics Letters*, vol. 49, pp. 538–539, April 2013. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/el.2013.0246>
- [84] Z. Liu, "Variable tap-length linear equaliser with variable tap-length adaptation step-size," *Electronics Letters*, vol. 50, pp. 587–589(2), April 2014. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/el.2014.0283>
- [85] C. Wang and T. Tang, "Several gradient-based iterative estimation algorithms for a class of nonlinear systems using the filtering technique," *Nonlinear Dynamics*, vol. 77, no. 3, pp. 769–780, 2014.
- [86] O. Rehman Khattak and A. Zerguine, "Leaky least mean fourth adaptive algorithm," *Signal Processing, IET*, vol. 7, no. 2, pp. 134–145, April 2013.
- [87] I. G. Muhammad, E. Abdel-Raheem, and K. E. Tepe, "Blind adaptive low-complexity time-domain equalizer algorithm for adsl systems by adjacent lag autocorrelation minimization (alam)," *Digital Signal Processing*, vol. 23, no. 5, pp. 1695–1703, 2013.
- [88] S. M. Shah, R. Samar, N. M. Khan, and M. A. Z. Raja, "Fractional-order adaptive signal processing strategies for active noise control systems," *Nonlinear Dynamics*, vol. 85, no. 3, pp. 1363–1376, 2016.
- [89] H. Jeon, T. Chang, S. Yu, and S. M. Kuo, "A narrowband active noise control system with frequency corrector," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 990–1002, 2011.
- [90] V. E. DeBrunner and D. Zhou, "Hybrid filtered error lms algorithm: another alternative to filtered-x lms," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 3, pp. 653–661, 2006.

- [91] R. M. Reddy, I. M. S. Panahi, and R. Briggs, "Hybrid FxRLS-FxNLMS adaptive algorithm for active noise control in fmri application," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 2, pp. 474–480, 2011.
- [92] Y. Xiao and J. Wang, "A new feedforward hybrid active noise control system," *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 591–594, 2011.
- [93] H. Zhao, X. Zeng, Z. He, and T. Li, "Adaptive rsov filter using the felms algorithm for nonlinear active noise control systems," *Mechanical Systems and Signal Processing*, vol. 34, no. 1, pp. 378–392, 2013.
- [94] J. Hu and T. Hsiao, "Adaptive feedforward active noise cancellation in ducts using the model matching of wave propagation dynamics," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1351–1356, 2012.
- [95] S. G. Osgouei and M. Geravanchizadeh, "Speech enhancement using convex combination of fractional least-mean-squares algorithm," in *Telecommunications (IST), 2010 5th International Symposium on*. IEEE, 2010, pp. 869–872.
- [96] M. Geravanchizadeh and O. S. Ghalami, "Speech enhancement by modified convex combination of fractional adaptive filtering," *Iranian Journal of Electrical and Electronic Engineering*, vol. 10, no. 4, pp. 256–266, 2014.
- [97] J. T. Machado, V. Kiryakova, and F. Mainardi, "Recent history of fractional calculus," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 3, pp. 1140–1153, 2011.
- [98] T. J. Freeborn, B. Maundy, and A. S. Elwakil, "Fractional-order models of supercapacitors, batteries and fuel cells: a survey," *Materials for Renewable and Sustainable Energy*, vol. 4, no. 3, p. 9, 2015.
- [99] C. Yin, Y. Chen, and S. Zhong, "Fractional-order sliding mode based extremum seeking control of a class of nonlinear systems," *Automatica*, vol. 50, no. 12, pp. 3173–3181, 2014.

- [100] W. Hsue and W. Chang, "Real discrete fractional fourier, hartley, generalized fourier and generalized hartley transforms with many parameters," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 10, pp. 2594–2605, 2015.
- [101] R. Zhou, R. Zhang, and D. Chen, "Fractional-order $1/\beta\alpha$ low-pass filter circuit," *J. Electr. Eng. Technol*, vol. 10, no. 4, pp. 1597–1609, 2015.
- [102] Y. Wei, B. Du, S. Cheng, and Y. Wang, "Fractional order systems time-optimal control and its application," *Journal of Optimization Theory and Applications*, vol. 174, no. 1, pp. 122–138, 2017.
- [103] Y. Wei, W. T. Peter, B. Du, and Y. Wang, "An innovative fixed-pole numerical approximation for fractional order systems," *ISA transactions*, vol. 62, pp. 94–102, 2016.
- [104] M. D. Ortigueira, "Introduction to fractional linear systems. part 1: Continuous-time case," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 147, no. 1, pp. 62–70, 2000.
- [105] D. Manuel, Ortigueira, "Introduction to fractional linear systems. part 2: discrete-time case," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 147, no. 1, pp. 71–78, 2000.
- [106] C. C. Tseng and S. Lee, "Design of linear phase fir filters using fractional derivative constraints," *Signal processing*, vol. 92, no. 5, pp. 1317–1327, 2012.
- [107] C. Tseng and S. Lee, "Designs of fractional derivative constrained 1-d and 2-d fir filters in the complex domain," *Signal Processing*, vol. 95, pp. 111–125, 2014.
- [108] M. D. Ortigueira, *Fractional calculus for scientists and engineers*. Springer Science & Business Media, 2011, vol. 84.
- [109] I. Podlubny, "Fractional derivatives: history, theory, application," *Utah State University, Logan*, 2005.

- [110] G. Bengochea and M. Ortigueira, “A recursive-operational approach with applications to linear differential systems,” *Journal of Applied Analysis*, vol. 22, no. 2, pp. 131–139, 2016.
- [111] R. Magin, M. D. Ortigueira, I. Podlubny, and J. Trujillo, “On the fractional signals and systems,” *Signal Processing*, vol. 91, no. 3, pp. 350–371, 2011.
- [112] M. Ortigueira and J. Machado, *Fractal and Fractional*, vol. 1, no. 1, p. 3, 2017.
- [113] M. D. Ortigueira and J. T. Machado, “What is a fractional derivative?” *Journal of computational Physics*, vol. 293, pp. 4–13, 2015.
- [114] S. M. Shah, R. Samar, N. M. Khan, and M. A. Z. Raja, “Design of fractional-order variants of complex lms and nlms algorithms for adaptive channel equalization,” *Nonlinear Dynamics*, vol. 88, no. 2, pp. 839–858, 2017.
- [115] M. D. Ortigueira and J. A. T. Machado, “What is a fractional derivative?” *Journal of computational Physics*, vol. 293, pp. 4–13, 2015.
- [116] R. Magin, M. D. Ortigueira, I. Podlubny, and J. Trujillo, “On the fractional signals and systems,” *Signal Processing*, vol. 91, no. 3, pp. 350–371, 2011.
- [117] G. Bengochea and M. D. Ortigueira, “A recursive-operational approach with applications to linear differential systems,” *Journal of Applied Analysis*, vol. 22, no. 2, pp. 131–139, 2016.
- [118] M. D. Ortigueira, J. A. T. Machado, M. Rivero, and J. J. Trujillo, “Integer/fractional decomposition of the impulse response of fractional linear systems,” *Signal Processing*, vol. 114, pp. 85–88, 2015.
- [119] J. Wang, Y. Ye, X. Pan, X. Gao, and C. Zhuang, “Fractional zero-phase filtering based on the riemann–liouville integral,” *Signal Processing*, vol. 98, pp. 150–157, 2014.
- [120] J. T. Machado, “Fractional order describing functions,” *Signal Processing*, vol. 107, pp. 389–394, 2015.

- [121] M. D. Ortigueira, J. J. Trujillo, V. I. Martynyuk, and F. J. V. Coito, "A generalized power series and its application in the inversion of transfer functions," *Signal Processing*, vol. 107, pp. 238–245, 2015.
- [122] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, "Stationary and nonstationary learning characteristics of the lms adaptive filter," *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151–1162, 1976.
- [123] Y. Pu, J. Zhou, Y. Zhang, N. Zhang, G. Huang, and P. Siarry, "Fractional extreme value adaptive training method: fractional steepest descent approach," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 4, pp. 653–662, 2015.
- [124] K. Jafari, J. Juillard, and M. Roger, "Convergence analysis of an online approach to parameter estimation problems based on binary observations," *Automatica*, vol. 48, no. 11, pp. 2837–2842, 2012.
- [125] H. Sheng, Y. Chen, and T. Qiu, "Tracking performance and robustness analysis of hurst estimators for multifractional processes," *IET signal processing*, vol. 6, no. 3, pp. 213–226, 2012.
- [126] Y. Q. C. H. Sheng and T. Qiu, "On the robustness of hurst estimators," *IET Signal Processing*, vol. 5, pp. 209–225, April 2011.
- [127] D. P. Mandic and J. A. Chambers, *Recurrent neural networks for prediction: Architectures, learning algorithms and stability*. John Wiley and Sons Ltd, 2001.
- [128] J. C. Principe, Y. N. Rao, and D. Erdogmus, "Error whitening wiener filters: Theory and algorithms. chapter 10:least mean square adaptive filters," vol. 31, pp. 445–489, 2003.
- [129] Y. N. Rao, D. Erdogmus, and J. C. Principe, "Error whitening criterion for adaptive filtering: theory and algorithms," *IEEE Transactions on signal processing*, vol. 53, no. 3, pp. 1057–1069, 2005.

- [130] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 3, pp. 209–216, 1997.
- [131] Y. Lee, "Adaptive equalization and receiver diversity for indoor wireless data communications," Ph.D. dissertation, Citeseer, 1997.
- [132] J. Wu and Y. R. Zheng, "Low complexity soft-input soft-output block decision feedback equalization," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 2, pp. 281–289, 2008.
- [133] M. T. Qaisrani and I. Barhumi, "Low-complexity equalisation of doubly selective channels for multiple-input-multiple-output transmission systems," *Signal processing*, vol. 7, pp. 101–111, 2013.
- [134] B. Han, Z. Yang, and Y. R. Zheng, "Efficient implementation of iterative multi-input–multi-output orthogonal frequency-division multiplexing receiver using minimum-mean-square error interference cancellation," *IET Communications*, vol. 8, no. 7, pp. 990–999, 2014.
- [135] K. Cumanan, Y. Rahulamathavan, and S. Lambotharan, "Minimum mean-square error transceiver optimisation for downlink multiuser multiple-input-multiple-output network with multiple linear transmit covariance constraints," *IET Signal Processing*, vol. 7, no. 1, pp. 47–58, 2013.
- [136] S. Abrar and A. K. Nandi, "Adaptive solution for blind equalization and carrier-phase recovery of square-qam," *IEEE Signal Processing Letters*, vol. 17, no. 9, pp. 791–794, 2010.
- [137] A. Ali, S. Abrar, A. Zerguine, and A. K. Nandi, "Newton-like minimum entropy equalization algorithm for apsk systems," *Signal Processing*, vol. 101, pp. 74–86, 2014.
- [138] S. M. Jung and P. Park, "Normalised least-mean-square algorithm for adaptive filtering of impulsive measurement noises and noisy inputs," *Electronics Letters*, vol. 49, no. 20, pp. 1270–1272, 2013.

- [139] C. Lin and J. Lin, "Physical-layer transceiving techniques on data-aided orthogonal frequency-division multiplexing towards seamless service on vehicular communications," *IET Communications*, vol. 7, no. 8, pp. 721–730, 2013.
- [140] M. Paskov, D. Lavery, and S. J. Savory, "Blind equalization of receiver in-phase/quadrature skew in the presence of nyquist filtering," *IEEE Photonics Technology Letters*, vol. 25, no. 24, pp. 2446–2449, 2013.
- [141] S. Ahmed, M. T. Akhtar, and X. Zhang, "Online acoustic feedback mitigation with improved noise-reduction performance in active noise control systems," *IET Signal Processing*, vol. 7, no. 6, pp. 505–514, 2013.
- [142] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.
- [143] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: stable processes and their applications," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 986–1010, 1993.
- [144] E. E. Kuruoglu, "Signal processing in α -stable noise environments: a least lp-norm approach," Ph.D. dissertation, University of Cambridge, 1999.
- [145] X. Sun, S. M. Kuo, and G. Meng, "Adaptive algorithm for active control of impulsive noise," *Journal of Sound and Vibration*, vol. 291, no. 1, pp. 516–522, 2006.
- [146] S. Miyoshi and Y. Kajikawa, "Statistical-mechanical analysis of the fxlms algorithm with nonwhite reference signals," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5652–5656.
- [147] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1650–1662, 2005.

- [148] N. R. Yousef, A. H. Sayed, and L. M. A. Jalloul, "Robust wireless location over fading channels," *IEEE transactions on vehicular technology*, vol. 52, no. 1, pp. 117–126, 2003.
- [149] S. Schwarz, J. C. Ikuno, M. Šimko, M. Taranetz, Q. Wang, and M. Rupp, "Pushing the limits of lte: A survey on research enhancing the standard," *IEEE Access*, vol. 1, pp. 51–62, 2013.
- [150] J. F. Galdino, E. L. Pinto, and M. S. de Alencar, "Analytical performance of the lms algorithm on the estimation of wide sense stationary channels," *IEEE Transactions on Communications*, vol. 52, no. 6, pp. 982–991, 2004.
- [151] M. Sternad, L. Lindborn, and A. Ahlén, "Wiener design of adaptation algorithms with time-invariant gains," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1895–1907, 2002.
- [152] A. M. A. Filho, E. L. Pinto, and J. F. Galdino, "Simple and robust analytically derived variable step-size least mean squares algorithm for channel estimation," *IET communications*, vol. 3, no. 12, pp. 1832–1842, 2009.
- [153] I. Barhumi, G. Leus, and M. Moonen, "Time-varying fir equalization for doubly selective channels," *IEEE Transactions on Wireless communications*, vol. 4, no. 1, pp. 202–214, 2005.
- [154] P. Sharma and K. Chandra, "Prediction of state transitions in rayleigh fading channels," *IEEE transactions on vehicular technology*, vol. 56, no. 2, pp. 416–425, 2007.
- [155] A. K. Kohli and D. K. Mehra, "Tracking of time-varying channels using two-step lms-type adaptive algorithm," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, pp. 2606–2615, 2006.
- [156] R. Gerzaguet, L. Ros, J. Brossier, S. Ghandour-Haidar, and F. Belvèze, "Self-adaptive stochastic rayleigh flat fading channel estimation," in *Digital Signal Processing (DSP), 2013 18th International Conference on*. IEEE, 2013, pp. 1–6.

- [157] H. Shu, L. Ros, and E. P. Simon, "Simplified random-walk-model-based kalman filter for slow to moderate fading channel estimation in ofdm systems," *IEEE transactions on signal processing*, vol. 62, no. 15, pp. 4006–4017, 2014.
- [158] A. Chatterjee and I. S. Misra, "Design and analysis of reward-punishment based variable step size lms algorithm in rayleigh faded channel estimation," in *Power, Communication and Information Technology Conference (PCITC), 2015 IEEE*. IEEE, 2015, pp. 223–228.
- [159] E. Mostafapour, C. Ghobadi, J. Nourinia, and M. C. Amirani, "Tracking performance of incremental lms algorithm over adaptive distributed sensor networks," *Journal of Communication Engineering*, vol. 4, no. 1, pp. 55–66, 2015.
- [160] L. Ros, H. Hijazi, and E. P. Simon, "Complex amplitudes tracking loop for multipath channel estimation in ofdm systems under slow to moderate fading," *Signal Processing*, vol. 97, pp. 134–145, 2014.
- [161] H. Shu, E. P. Simon, and L. Ros, "On the use of tracking loops for low-complexity multi-path channel estimation in ofdm systems," *Signal Processing*, vol. 117, pp. 174–187, 2015.
- [162] D. Goswami and K. K. Sarma, "Phase tracking and cci cancellation in severely faded rayleigh mimo channel," in *IJCA Proceedings on Mobile and Embedded Tech. International Conference*, 2013.
- [163] E. Karami, "Performance analysis of decision directed maximum likelihood mimo channel tracking algorithm," *International Journal of Communication Systems*, vol. 26, no. 12, pp. 1562–1578, 2013.
- [164] S. Pagadarai, A. M. Wyglinski, and C. R. Anderson, "Low-mobility channel tracking for mimo-ofdm communication systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, pp. 1–18, 2013.

- [165] M. C. Mah, H. S. Lim, and A. W. C. Tan, "Robust joint cfo and fast time-varying channel tracking for mimo-ofdm systems," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6494–6498.
- [166] S. Kaddouri, P. J. Beaujean, P. Bouvet, and G. Real, "Least square and trended doppler estimation in fading channel for high-frequency underwater acoustic communications," *IEEE journal of oceanic engineering*, vol. 39, no. 1, pp. 179–188, 2014.
- [167] M. Scarpiniti, D. Comminiello, G. Scarano, R. Parisi, and A. Uncini, "Steady-state performance of spline adaptive filters," *IEEE Transactions on Signal Processing*, vol. 64, no. 4, pp. 816–828, 2016.
- [168] H. Shu, L. Ros, and E. P. Simon, "Third-order complex amplitudes tracking loop for slow flat fading channel online estimation," *IET Communications*, vol. 8, no. 3, pp. 360–371, 2014.
- [169] A. Ozen, "A novel variable step size adjustment method based on channel output autocorrelation for the lms training algorithm," *International Journal of Communication Systems*, vol. 24, no. 7, pp. 938–949, 2011.
- [170] A. Benveniste and G. Ruget, "A measure of the tracking capability of recursive stochastic algorithms with constant gains," *IEEE Transactions on Automatic Control*, vol. 27, no. 3, pp. 639–649, 1982.
- [171] E. Eleftheriou and D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1097–1110, 1986.
- [172] N. R. Yousef and A. H. Sayed, "A unified approach to the steady-state and tracking analyses of adaptive filters," *IEEE Transactions on Signal Processing*, vol. 49, no. 2, pp. 314–324, 2001.
- [173] S. M. Shah, "Riemann-liouville operator-based fractional normalised least mean square algorithm with application to decision feedback equalisation

of multipath channels,” *IET Signal Processing*, vol. 10, no. 6, pp. 575–582, 2016.