

CAPITAL UNIVERSITY OF SCIENCE AND  
TECHNOLOGY, ISLAMABAD



**Processing-Efficient Distributed  
Adaptive RLS Filtering for  
Computationally-Constrained  
Platforms**

by

**Hasan Raza**

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

in the

Faculty of Engineering

Department of Electrical Engineering

2019

**Processing-Efficient Distributed Adaptive RLS  
Filtering for Computationally-Constrained  
Platforms**

By

Hasan Raza

(PE123010)

**Dr. Rodica Ramer**

University of New South Wales, Sydney, Australia

**Dr. Mohammad N. Patwary**

Staffordshire University, UK

**Dr. Noor Muhammad Khan**

(Thesis Supervisor)

**Dr. Noor Muhammad Khan**

(Head, Department of Electrical Engineering)

**Dr. Imtiaz Ahmad Taj**

(Dean, Faculty of Engineering)

**DEPARTMENT OF ELECTRICAL ENGINEERING  
CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY  
ISLAMABAD**

**2019**

Copyright © 2019 by Hasan Raza

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.

*I dedicate this thesis to my parents, aunt and wife for their love and support*



# 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](mailto: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 “**Processing-Efficient Distributed Adaptive RLS Filtering for Computationally-Constrained Platforms**” was conducted under the supervision of **Dr. Noor Muhammad Khan**. 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 **July 03, 2019**.

Student Name: Mr. Hasan Raza (PE123010)

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

### Examination Committee :

- (a) External Examiner 1: Dr. Najam ul Islam  
Professor  
Bahria University, Islamabad
- (b) External Examiner 2: Dr. Mukhtar Ullah  
Associate Professor  
FAST-NUCES, Islamabad
- (c) Internal Examiner : Dr. Imtiaz Ahmad Taj  
Professor  
CUST, Islamabad

Supervisor Name : Dr. Noor Muhammad Khan .  
Professor  
CUST, Islamabad

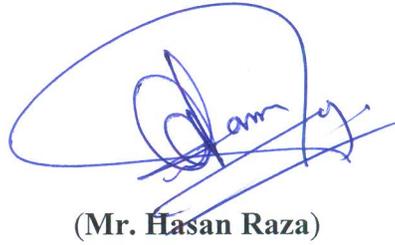
Name of HoD : Dr. Noor Muhammad Khan  
Professor  
CUST, Islamabad

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

## **AUTHOR'S DECLARATION**

I, **Mr. Hasan Raza (Registration No. PE-123010)**, hereby state that my PhD thesis titled, '**Processing-Efficient Distributed Adaptive RLS Filtering for Computationally-Constrained Platforms**' 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. Hasan Raza)**

Dated: 03 July, 2019

Registration No : PE-123010

## PLAGIARISM UNDERTAKING

I solemnly declare that research work presented in the thesis titled “**Processing-Efficient Distributed Adaptive RLS Filtering for Computationally-Constrained Platforms**” 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. Hasan Raza)

Dated: 3 July, 2019

Registration No : PE-123010

## *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:-

1. Noor M. Khan, Hasan Raza "Processing-Efficient Distributed Adaptive RLS Filtering for Computationally-Constrained Platforms", *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 1248796, 7 pages, 2017. [Impact factor: 1.396]
2. Hasan Raza, Noor M. Khan, "Low-Complexity Linear Channel Estimation for MIMO Communication Systems", *Wireless Personal Communications, Springer*, 97(4), p.p 5031-5044, 2017. [Impact factor: 0.929]
3. Hasan Raza, Noor M. Khan, "Processing-Efficient Distributed Diffusion-based RLS Filtering for Computationally-Constrained Platforms", *Journal of Signal Processing Systems, Springer* (**Submitted**)

**(Hasan Raza)**

Registration No: PE123010

## *Acknowledgements*

First of all, I would like to thank Almighty Allah for giving me an opportunity to get higher education. I would like to express my sincere gratitude to my supervisor Prof. Dr. Noor M. Khan for his guidance in all aspects of my education and research. It has been an honor for me to work with him. He has always encouraged my research work and has advised me patiently by providing insightful suggestions.

I would also like to thank all the faculty members of electrical engineering department of the Capital University of Science and Technology, for their precious concepts and valuable thoughts delivered in their lectures. I also want to thank all the members of the ARWiC research group who always provided me a pleasant environment to think of innovative ideas.

Finally, I thank my parents and aunt whose discontinued support always helped me a lot during my research. I also express my gratitude towards my wife who provided great encouragement during my thesis write up.

**(Hasan Raza)**

Registration No: PE123010

# *Abstract*

Achieving fast convergence on an energy-limited and computationally-constrained platform still remains a dream in spite of magnificent advancements in Integrated Circuit (IC) technologies. For instance, in telephony, the echo cancellation requires a high-definition adaptive-filtering algorithm that further needs a robust convergence performance while tracking the time varying uncertainties present in the communication link. Nevertheless, such high definition adaptive algorithm cannot be run on an energy-limited and computationally-constrained inexpensive platform.

The research work in this thesis focuses to propose the low-complexity distributed adaptive filtering solution for energy-constrained platforms. The thesis is organized in three parts. Part-1 aims to develop a low-complexity MIMO channel estimation algorithm for MIMO communication system. Part-II and III provide the distributed and diffusion based adaptive signal processing solutions for computationally-constrained inexpensive platforms.

The thesis begins with an overview of the adaptive algorithms with implementation constraints and then proceeds towards a comprehensive and detailed literature survey. The literature survey can be classified into two major areas, i.e. adaptive filter theory and adaptive algorithm implementation over low-cost platforms. Furthermore, a channel model is presented with the consideration of two multipath components for MIMO communication environment. Taking it as a reference as channel model, a spatiotemporal low-complexity adaptive estimation algorithm is proposed by assuming time-variant block fading channel with fixed number of training symbols. The proposed algorithm exhibits better results than those shown by some notable least square algorithms in the literature. The effect of varying doppler rates on the convergence performance of the algorithm is thoroughly observed to check the validation of the algorithm. Obtained simulated results show that the proposed algorithm entails low-complexity and provides independency on forgetting factor as compared to notable adaptive filtering algorithms.

In the second part of the thesis, a novel processing-efficient architecture of a group of inexpensive and computationally-constrained small platforms is proposed for a parallelly-distributed adaptive signal processing (PDASP) operation. The proposed architecture is capable of running computationally-expensive procedures like complex adaptive algorithms cooperatively. The proposed PDASP architecture operates properly even if perfect time alignment among the participating platforms is not available. Complexity and processing time of the PDASP scheme are compared with those of the sequentially-operated algorithms. The comparative analysis shows that the PDASP scheme exhibits much lesser computational complexity parallelly than the sequentially-operated algorithms. Moreover, for high and low doppler rates, the proposed architecture provides a parallelly-decreased processing time than the sequentially-operated MIMO algorithms.

In part III, a novel distributed diffusion-based adaptive signal processing (DDASP) architecture for computationally-constrained small platforms is introduced. In the proposed DDASP architecture, the adaptive algorithm is diffused into the desired number of processing devices. The number of processing nodes that are used in DDASP architecture is dependent upon the number of MIMO channel streams as well as on the number multipath components. Therefore, having more nodes and diffusion mechanism, the proposed DDASP architecture exhibits lesser and linear computational complexity parallelly on each processing node involved as compared to the proposed PDASP architecture.

# Contents

<b>Author's Declaration</b>	<b>v</b>
<b>Plagiarism Undertaking</b>	<b>vi</b>
<b>List of Publications</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>viii</b>
<b>Abstract</b>	<b>ix</b>
<b>List of Figures</b>	<b>xiv</b>
<b>List of Tables</b>	<b>xvii</b>
<b>Abbreviations</b>	<b>xix</b>
<b>Symbols</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Wireless Sensor Networks (WSNs) . . . . .	1
1.2 Multipath Radio Channel . . . . .	3
1.2.1 Refraction . . . . .	3
1.2.2 Reflection . . . . .	4
1.2.3 Scattering . . . . .	4
1.2.4 Diffraction . . . . .	4
1.2.5 Multipath Fading . . . . .	4
1.3 MIMO Communication Channel . . . . .	5
1.4 Adaptive Filters . . . . .	6
1.5 Adaptive Filter Structures . . . . .	7
1.6 Mean Squared Error (MSE) Criterion . . . . .	8
1.7 Wiener Filter Solution . . . . .	10
1.8 Adaptive Filtering Algorithms . . . . .	10
1.8.1 The Stochastic Gradient . . . . .	11
1.8.2 Kalman Filter . . . . .	11
1.8.3 Least Square Estimation . . . . .	12

---

1.9	Major Applications of Adaptive Filters . . . . .	13
1.9.1	MIMO Channel Estimation . . . . .	13
1.9.2	System Identification in Case of Missing Data . . . . .	13
1.10	Distributed Wireless Sensor Networks . . . . .	14
1.11	Research Aim . . . . .	15
1.12	Research Contributions . . . . .	16
1.12.1	Low Complexity Linear MIMO Channel Estimation Algorithm . . . . .	16
1.12.2	Parallel Distributed Adaptive RLS Filtering . . . . .	16
1.12.3	Parallel Distributed Diffusion-based Adaptive RLS Filtering . . . . .	16
1.12.4	Application of Diffusion-based RLS Architecture on Missing Data Systems . . . . .	17
1.13	Organization of the Thesis . . . . .	17
<b>2</b>	<b>Literature Survey and Problem Formulation</b>	<b>18</b>
2.1	Literature Survey . . . . .	18
2.1.1	Adaptive Filter Theory . . . . .	19
2.1.2	Implementation of Adaptive Algorithm Filtering over Low Cost Platforms . . . . .	21
2.2	Research Motivation and Problem Statement . . . . .	24
2.3	Research Methodology . . . . .	25
<b>3</b>	<b>System and Channel Model</b>	<b>26</b>
3.1	System Model . . . . .	26
3.2	MIMO Channel Model . . . . .	27
<b>4</b>	<b>Low-Complexity Channel Estimation for MIMO Communication Systems</b>	<b>32</b>
4.1	Overview . . . . .	32
4.2	Proposed Low-Complexity MIMO Algorithm . . . . .	33
4.3	Complexity Analysis . . . . .	37
4.4	Simulation Results and Discussion . . . . .	38
<b>5</b>	<b>Parallel Distributed Adaptive RLS Filtering</b>	<b>43</b>
5.1	Overview . . . . .	43
5.2	Proposed PDASP Technique for RLS with Non-aligned Time Indexes . . . . .	46
5.3	Complexity Analysis . . . . .	49
5.4	Simulation Results and Discussion . . . . .	49
<b>6</b>	<b>Parallel Distributed Diffusion based Adaptive RLS Filtering</b>	<b>57</b>
6.1	System Model for Diffusion based Distributed Adaptive Signal Pro- cessing . . . . .	58
6.2	DDASP Architecture for LoS-only Link . . . . .	60
6.3	DDASP Architecture for Multipath based Link . . . . .	64
6.4	Results and Discussion . . . . .	66
6.4.1	Computational Complexity Comparison . . . . .	66

---

6.4.2	Node Requirement . . . . .	67
6.4.3	Communication Burden . . . . .	70
<b>7</b>	<b>Application of Proposed DDASP Architecture on Missing Data Systems</b>	<b>74</b>
7.1	Overview . . . . .	74
7.2	System model and Missing Data Patterns . . . . .	76
7.2.1	Noise-free Discrete Missing Data System . . . . .	76
7.2.2	Noisy Discrete-time Missing Data System . . . . .	77
7.3	Auxiliary Model . . . . .	78
7.4	Implementation of DDASP on Missing Data System . . . . .	80
7.5	Complexity Comparison . . . . .	84
7.6	Simulation Results and Discussion . . . . .	88
<b>8</b>	<b>Conclusions and Future Directions</b>	<b>91</b>
8.1	Conclusions . . . . .	91
8.2	Future Directions . . . . .	93
	<b>Bibliography</b>	<b>95</b>

# List of Figures

1.1	Multipath propagation environment . . . . .	3
1.2	A typical $N \times N$ MIMO communication system . . . . .	6
1.3	Adaptive IIR filter . . . . .	8
1.4	Adaptive FIR filter . . . . .	9
1.5	Block diagram of adaptive filtering problem . . . . .	9
1.6	Distributed adaptive incremental strategy . . . . .	15
3.1	System model for MIMO communication system. . . . .	27
3.2	Frequency-selective channel model for MIMO communications. . . . .	30
4.1	Per-iteration multiplication complexity versus multipath components for $2 \times 2$ MIMO communication system . . . . .	38
4.2	Per-iteration multiplication complexity versus multipath components for $4 \times 4$ MIMO communication system . . . . .	38
4.3	Per-iteration multiplication complexity versus multipath components for $6 \times 6$ MIMO communication system . . . . .	39
4.4	Processing time versus number of iterations at SNR=20dB for a $4 \times 4$ MIMO system with LoS-only link . . . . .	39
4.5	Mean squared error at SNR=20dB for a $4 \times 4$ MIMO system with LoS-only link at $f_D T = 0.01$ . . . . .	40
4.6	Comparison of the MSE when the proposed algorithm is applied to various fading conditions at SNR=20dB for a $4 \times 4$ MIMO system with LoS-only link . . . . .	40
4.7	Behavior of minimum mean squared error (MMSE) against $E_b/N_o$ for a $4 \times 4$ MIMO system with LoS-only link when $f_D T = 0.004$ . . . . .	41
4.8	Probability of bit error rate (BER) against $E_b/N_o$ for a $4 \times 4$ MIMO system with LoS-only link when $f_D T = 0.004$ . . . . .	41
5.1	Sequential working of conventional RLS algorithm (a). sequential working of individual blocks (b). processes involved in a single block . . . . .	44
5.2	Proposed cooperative parallelly-operated RLS filtering . . . . .	45
5.3	Proposed PDASP architecture for MIMO RLS algorithm with non-aligned time indexes . . . . .	47
5.4	Per-iteration multiplication complexity comparison among $2 \times 2$ MIMO sequential algorithms and proposed PDASP technique . . . . .	50
5.5	Per-iteration multiplication complexity comparison among $4 \times 4$ MIMO sequential algorithms and proposed PDASP technique . . . . .	50

---

5.6	Per-iteration multiplication complexity comparison among $6 \times 6$ MIMO sequential algorithms and proposed PDASP technique . . . . .	51
5.7	Per-iteration addition complexity comparison among $2 \times 2$ MIMO sequential algorithms and proposed PDASP technique . . . . .	51
5.8	Per-iteration addition complexity comparison among $4 \times 4$ MIMO sequential algorithms and proposed PDASP technique . . . . .	52
5.9	Per-iteration addition complexity comparison among $6 \times 6$ MIMO sequential algorithms and proposed PDASP technique . . . . .	52
5.10	Mean square error (MSE) tracking performance versus training length for $4 \times 4$ MIMO when $f_D T = 10^{-6}$ . . . . .	53
5.11	Mean square error (MSE) tracking performance versus training length for $4 \times 4$ MIMO when $f_D T = 10^{-3}$ . . . . .	53
5.12	Mean square error difference among sequential algorithms and proposed PDASP scheme . . . . .	54
5.13	Processing time comparison among sequential algorithms and proposed PDASP technique . . . . .	55
5.14	Processing time difference for $4 \times 4$ MIMO system (a). sequential RLS vs proposed PDASP technique (b). sequential linear Kalman vs proposed PDASP technique . . . . .	55
6.1	Block Diagram of DDASP Architecture . . . . .	58
6.2	Proposed DDASP architecture for $2 \times 2$ MIMO RLS algorithm with LoS-only link . . . . .	60
6.3	Proposed DDASP architecture for $2 \times 2$ MIMO channel estimation with one multipath component . . . . .	65
6.4	Per-iteration multiplication complexity comparison between $2 \times 2$ proposed DDASP and proposed PDASP architectures . . . . .	68
6.5	Per-iteration multiplication complexity comparison between $4 \times 4$ proposed DDASP and proposed PDASP architectures . . . . .	68
6.6	Per-iteration multiplication complexity comparison between $6 \times 6$ proposed DDASP and proposed PDASP architectures . . . . .	69
6.7	Per-iteration addition complexity comparison between $2 \times 2$ proposed DDASP and proposed PDASP architectures . . . . .	69
6.8	Per-iteration addition complexity comparison between $4 \times 4$ proposed DDASP and proposed PDASP architectures . . . . .	70
6.9	Per-iteration addition complexity comparison between $6 \times 6$ proposed DDASP and proposed PDASP architectures . . . . .	70
7.1	Missing data error system . . . . .	78
7.2	Missing data error system with reference model . . . . .	79
7.3	Proposed DDASP architecture for low-complexity estimation algorithm with aligned time indexes . . . . .	79
7.4	Per-iteration multiplication complexity comparison among sequential algorithms and proposed DDASP technique on missing data system . . . . .	87

---

7.5	Per-iteration addition complexity comparison among sequential algorithms and proposed DDASP technique on missing data system .	87
7.6	Coefficients estimation error versus number of iterations at $30dB$ SNR . . . . .	89
7.7	Coefficients estimation error versus number of iterations at $20dB$ SNR . . . . .	89

# List of Tables

4.1	The proposed low-complexity MIMO channel estimation algorithm .	37
5.1	Pseudo code: Working procedure of PDASP architecture for MIMO communication system . . . . .	48
5.2	PDASP RLS algorithm with non-aligned time indexes if runs sequentially . . . . .	49
5.3	Percentage improvement in decreased processing of PDASP MIMO RLS with respect to sequential MIMO RLS . . . . .	56
5.4	Percentage improvement in decreased processing of PDASP with respect to sequential linear Kalman . . . . .	56
6.1	Number of processing devices require for various MIMO systems with or without multipath components along with one master node	59
6.2	Pseudo code: Working procedure of DDASP for $2 \times 2$ MIMO communication system . . . . .	62
6.3	Vector notations used in DDASP architecture for $2 \times 2$ MIMO system with one diffused component . . . . .	65
6.4	Computational complexity comparison for various distributed techniques using $N \times N$ MIMO system with LoS-only links . . . . .	71
6.5	Distributed multiplication complexity for various MIMO systems with LoS-only and multipath based link . . . . .	71
6.6	Distributed addition complexity for various MIMO systems with LoS-only and and multipath based link . . . . .	71
6.7	Processing time comparison between DDASP and PDASP in $\mu sec$ for different MIMO systems with LoS-only and multipath based link	72
6.8	Number of processing nodes required in various distributed techniques for different communication systems . . . . .	72
6.9	Communication burden comparison on various distributed architectures for different communication systems . . . . .	73
7.1	Multiplication and addition complexity of DDASP architecture with respect to each processing node . . . . .	84
7.2	Coefficient estimation of example scenario using conventional RLS algorithm at forgetting factor $\lambda = 0.9999$ . . . . .	84
7.3	Coefficient estimation of example scenario using proposed DDASP based low-complexity algorithm . . . . .	85

---

7.4	Sequential and distributed multiplication complexity for various system orders in missing data systems . . . . .	85
7.5	Sequential and distributed addition complexity for various system orders in missing data systems . . . . .	86
7.6	Sequential and proposed node-by-node distributed processing time in $\mu\text{sec}$ for various system orders . . . . .	86

# Abbreviations

AR	Autoregressive
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CLR	Complex Latex Reduction
CM-RLS	Constant Modulus Recursive Least Square
CSI	Channel State Information
CUDA	Compute Unified Device Architecture
DANSE	Distributed Adaptive Non-specific Signal Estimation
DDASP	Distributive Diffusion based Adaptive Signal Processing
DD-RLS	Decision Directed Recursive Least Square
DRLS	Distributed Recursive Least Square
FIR	Finite Impulse Response
GPU	Graphical Processing Unit
IIR	Infinite Impulse Response
I.I.D	Independent Identical Distribution
IoT	Internet of Things
Labview	Laboratory Virtual Instrument Engineering Workbench
LoS	Line of Sight
LMS	Least Mean Square
LS	Least Square
LR-MMSE	Latex Reduction Minimum Mean Square Error
LQE	Linear Quadratic Equation
MAFF-RLS	Modified Adaptive Forgetting Factor Recursive Least Square

---

MATLAB	Matrix Laboratory
MIMO	Multiple-Input Multiple-Output
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
OFDM	Orthogonal Frequency Division Multiple access
OP-RLS	Optimum weighted Recursive Least Square
PC	Personal Computer
PDASP	Parallel Distributive Adaptive Signal Processing
PE	Polynomial Expansion
QR	Orthogonal Matrix and Right Triangular Matrix
RLM	Recursive Least M-estimation
RLS	Recursive Least Square
RLSVF	Recursive Least Square Variable Forgetting factor
RRLS	Robust Recursive Least Square
RRLSS	Robust Recursive Least Square with Scale factor
RRWLSV	Robust Recursive Weighted Least Square with both Scale and Variable forgetting factors
RVFF-RLS	Robust Variable Forgetting Factor Recursive Least Square
SAF	Subband Adaptive Filtering
SDR	Software Define Radio
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
WSN	Wireless Sensor Network
5G	5th Generation Cellular Communication System

# Symbols

$E(\cdot)$	Expectation
$e_{Seq}$	Sequential algorithm error
$e_{NA}$	Non-aligned time indexes error
$f_D$	Doppler frequency
$I_N$	Identity matrix of order $N$
$J_0(\cdot)$	First kind zero order Bessel function
$L$	Number of multipath components
$N_{\mathcal{T}}$	Number of transmit antennas
$N_{\mathcal{R}}$	Number of receive antennas
$T_f$	Fetch Time
$T_c$	Computational time
$T_b$	Block processing time
$T_f$	Fetch time
$T_s$	Algorithm step time
$T_{f,c}$	cumulative fetch time
$T_{f,S\leftrightarrow M}$	Total fetch time among master and slave nodes
$\xi$	Mean square error performance function
$\alpha$	Step size parameter
$\vartheta$	White noise at receiving antenna
$\beta$	Channel coefficient
$\omega$	Independent identical distributed zero mean Gaussian process
$\sigma^2$	Noise variance
$\Omega$	i.i.d Gaussian process matrix

$\tilde{\Xi}$	Difference of original and estimated channel state
$\tilde{\Psi}$	$M \times M$ weight-error correlation matrix
$(.)^H$	Hermitian transpose
$\psi$	Input-output information vector
$\rho$	Missing data examination variable
$\lambda$	Forgetting factor
$\gamma$	Parameter estimation error

# Chapter 1

## Introduction

This chapter presents an introduction of the research topic. In Section 1.1, an overview of wireless sensor networks is given. Section 1.2 presents the multipath radio channel along with a note on its physical phenomena. Sections 1.3 and 1.4 provide overviews on MIMO communication system and adaptive filters, respectively, while Section 1.5 presents the designing structures of an adaptive filter. Mean square error criterion and non-adaptive Wiener filter solution are presented in section 1.6 and 1.7, respectively. Section 1.8 describes the adaptive filtering approaches used for the designing of an adaptive filter. Section 1.9 presents the major applications of the adaptive filters and Section 1.10 provides distributed strategies concerning wireless sensor networks. Section 1.11, 1.12 and 1.13 present the aim, contributions and publications of the research work contained in the dissertation, respectively. Finally, section 1.14 discusses the organization of this dissertation.

### 1.1 Wireless Sensor Networks (WSNs)

Wireless sensor networks (WSNs) [1, 2] have drawn considerable research interest recently because they can efficiently monitor the environment or physical conditions on behalf of reduced operational cost. However, the implementation of high definition adaptive filtering algorithms [3–5] over the low-cost wireless platforms of

these sensor networks [6] still remains a real challenge for the research community. For instance, in telephony, the echo cancelation requires a high definition adaptive filtering algorithm to avail a robust convergence performance while tracking the time varying uncertainties present in the communication link. Nevertheless, such high definition adaptive algorithm cannot be run on an energy-limited and computationally-constrained inexpensive platform. Moreover, due to high computational complexity of the signal processing algorithms, an energy-constrained node may not exhibit adequate performance with limited power source and existing hardware. To keep the effective utilization of existing power and hardware, the complexity of the adaptive algorithm must be reduced while implementing on energy-constrained platforms. On the other hand, advancement in distributed signal processing techniques [7] allows the wireless sensor nodes to run the high-definition adaptive filtering algorithm cooperatively. In this context, the resource-constrained WSN may run the high definition adaptive algorithm collaboratively. This mechanism may thus exert a significant impact on aggregate computational complexity as well as on the energy consumption of an individual node.

Since the beginning of the last decade, the cost of deployment of a Wireless Sensor Network (WSN) has dropped dramatically, due to which an exponential growth has been observed in its use from commercial and industrial applications to military areas. A WSN can be described generally a network of nodes that cooperatively work on a desired process and have the ability to interact with the computers or to the surrounding environment. The wireless sensor nodes may be deployed inside the sensor field to process and share data among them and finally transfer it towards the centralized node known as sink or gateway. A sensor node has a central importance in the network. It generally includes the battery, wireless transceiver, micro-controller and sensor unit. The limited memory and power constraints restrict the wireless sensor node to run the high definition adaptive algorithm on a single unit. In case of multiple input multiple output (MIMO) channel estimation or equalization, the complexity of the adaptive algorithm is directly dependent on the number of communication streams and the multipath components involving in the propagation environment. Therefore, the modern

WSNs have to operate cooperatively for such filtering applications with the best utilization of limited energy resources within the constraints posed by the existing hardware.

## 1.2 Multipath Radio Channel

In wireless communications, the multiple copies of the transmitted signal are received by the receiver at different timing instants. These multiple copies may undergo constructive or destructive fading depending upon their delays. The physical phenomena that affect the transmitted signal are shown in Fig. 1.1 and their brief descriptions are given in the subsequent subsections.

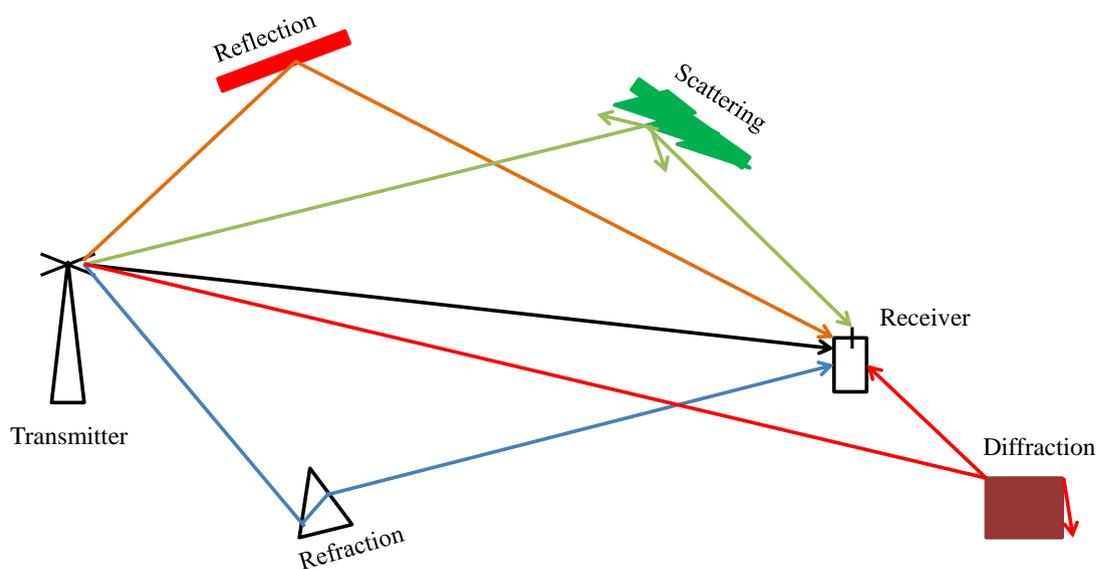


FIGURE 1.1: Multipath propagation environment

### 1.2.1 Refraction

Refraction is a physical phenomenon which refers to the change in the direction of the signal when it enters from one medium to another medium. The change in the density of the medium is the prime factor that is directly dependent on the physical form of the signal.

## 1.2.2 Reflection

Reflection arises when the signal is incident upon a smooth surface of an object. It occurs when the dimensions of the surface are larger compared to the wavelength of the signal.

## 1.2.3 Scattering

Scattering occurs when the dimensions of the obstacle are smaller or equal compared to the wavelength of the signal. It divides the energy of the propagating signal and reradiates the signal in multiple directions.

## 1.2.4 Diffraction

The diffraction occurs when the propagation signal encounters the edge of the building or object. In this phenomenon, the signal is bending around the stricken object. This gives benefit in the regions of shadowing caused due to big obstacles.

## 1.2.5 Multipath Fading

Due to the above-mentioned phenomena, the electromagnetic waves of varying lengths cause multipath fading which results in constructive or destructive interference. Multipath components cause time dispersion due to the arrival of the replicas of the transmitted signal at different timing instants. The situation becomes worse when the bandwidth of the channel is smaller than the bandwidth of the transmitted signal, then the channel undergoes frequency selective fading and induces intersymbol interference (ISI) [8]. Likewise, the multipath components experience a shift in frequency due to the relative motion between the transmitter and the receiver which results in rapid variations in the received signal over a span of short distance. This frequency shift is known as Doppler shift and this occurs when the coherence time of the channel is smaller than the symbol period

of the transmitted signal. The Doppler shift is directly proportional to the carrier frequency and the velocity of the receiver. The Doppler shift is also influenced by the direction of motion of the receiver with respect to the direction of arrival of the multipath components. This phenomenon also lead to distort the transmitted signal.

### 1.3 MIMO Communication Channel

Multiple Input Multiple Output (MIMO) refers to signal processing techniques which utilize multiple antenna components to enhance the performance of any wireless communication system [9]. Communication theories show that the MIMO communication system provides a significant impact on the data rate without increasing the operational bandwidth. A block diagram of a typical MIMO communication system is shown in Fig. 1.2 which consists of  $N$  transmit and  $N$  receive antenna elements. The capacity of the MIMO communication system is linearly dependent on the number of MIMO antennas. Theoretically, the capacity  $C$  provided by MIMO system with  $M$  spatial streams can be expressed as

$$C = MB \log_2(1 + P_S/P_N) \quad (1.1)$$

where  $B$  is the channel bandwidth and  $P_S$  and  $P_N$  are signal and noise power, respectively. The capacity of the MIMO communication system is based on the assumption that all the channels between transmit and receive antennas must be precisely known. However, in real time scenario, each antenna receives not only the direct line of sight component but also the other spatial streams coming from the other antennas elements. Even, the multipath radio environment makes the channel matrix at the receiver side more complex. Therefore, the true information of the channel states must be required at the receiver side to detect signals correctly in order to meet the demands of improved capacity and high data rate.

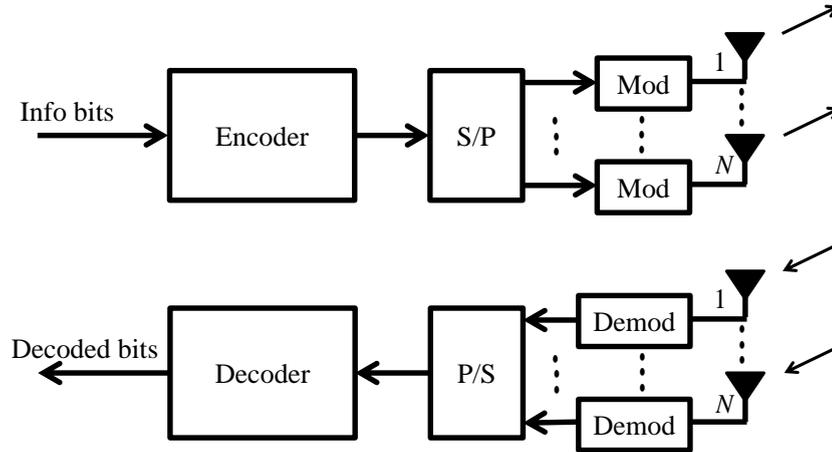


FIGURE 1.2: A typical  $N \times N$  MIMO communication system

## 1.4 Adaptive Filters

The term filter is commonly used for any system or device that processes the mixture of elements or particles given as its input and produces a desired set of elements or particles at its output according to certain principles or rules. However, in the perspective of signals and systems, the elements or particles can be referred to frequency components of the desired signal and the filter is used to pass all the frequencies belonging to the required frequency band and reject the rest of them. Filters may be either linear or non-linear. In this study, the emphasis revolves only around the linear filter theory and analysis.

The filter that adjusts its parameters according to unknown environmental conditions is known as adaptive filter. To keep the tracking of uncertainty, the adaption rate of adaptive filter must be faster than the rate at which change occurs in statistics. Since, the beginning of last decade, adaptive filtering techniques play a very important role in the emerging fields of science and technology [10–12]. Adaptive filters are evaluated by their numerical stability, steady-state mean squared error (MSE), convergence performance and computational complexity. Many adaptive algorithms have been introduced in the literature with the target to enhance the performance of adaptive algorithms in sense of tracking capability, steady-state MSE and convergence rate. The most important features of adaptive filtering algorithm are as follows:

- The convergence of the adaptive filter must be faster than the unknown changing statistics.
- The miss-adjustment should be minimum as much as it can be.
- The tracking performance should be high according to time variations provided by the propagation environment.
- The computational complexity should be minimal which makes possible of programming requirements and data processing.
- Numerical accuracy and stability should be comprehended
- The design of the adaptive filter should be efficient in sense of hardware implementation.

## 1.5 Adaptive Filter Structures

The most common structures used for the implementation of adaptive filters are finite impulse response (FIR) and infinite impulse response (IIR) [13]. The response of IIR filter is infinite in time which provides many difficulties while implementing the adaptive filtering algorithm on any practical application. Therefore, the applications of IIR filter in the domain of adaptive filtering theory are limited. In many cases, the digital IIR filter becomes unstable due to the poles being outside the unit circle. Moreover, the performance function of IIR filter results in many local-minimum points which forces the algorithm to converge to any one of the local-minimum points rather than global minimum point. A typical structure of IIR filter is shown in Fig. 1.3. The output signal  $y_k$  is the linear combination of the recursive and non-recursive products and can be expressed as

$$y_k = \sum_{i=0}^{N-1} a_{i,k} x_{k-i} + \sum_{j=1}^{N-1} b_{j,k} x_{k-j} \quad (1.2)$$

where  $N$  shows the filter order,  $x_k$  is the input signal and  $a_{i,k}$  and  $b_{j,k}$  are the feed-forward and feedback weights of the  $i$ th and  $j$ th tap, respectively, at time

instant  $k$ . On the other hand, the performance function of FIR filter has a single minimum point which makes the filter faster to converge. The structure of FIR filter is shown in Fig. 1.4. Likewise, the output  $y_k$  in FIR filter is the linear combination of the non-recursive products of filter weight taps  $w_{i,k}$  and delayed versions of input signal.  $y_k$  can be expressed as

$$y_k = \sum_{i=0}^{N-1} w_{i,k} x_{k-i} \quad (1.3)$$

The error signal  $e_k$  that is the difference of  $y_k$  and the desired signal  $d_k$  is used to update the filter weights. Furthermore, it can be realized that the computational complexity provided by IIR structure is greater than that of FIR filter. Therefore, the increased computational complexity restricts the IIR filter to be implemented on computationally constrained small platforms.

## 1.6 Mean Squared Error (MSE) Criterion

The block diagram of a typical adaptive filtering problem is shown in Fig. 1.5. The

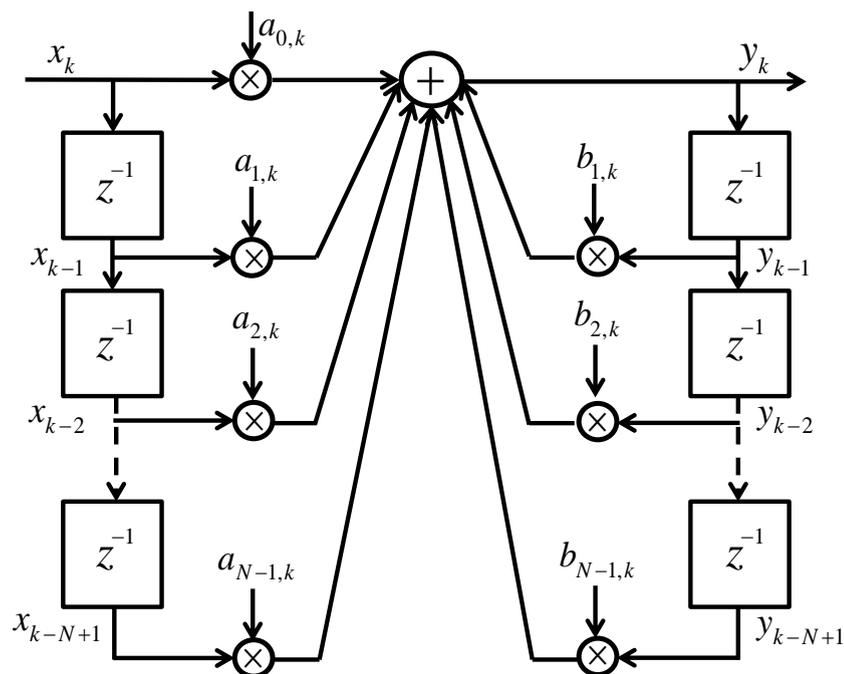


FIGURE 1.3: Adaptive IIR filter

input signal  $x_k$  and desired signal  $d_k$  are taken from the random white process. The estimation error that is the difference of output signal  $y_k$  and the desired signal  $d_k$  shows a key importance of the performance of linear discrete time filter. The output signal  $y_k$  of discrete time filter approaches towards  $d_k$  as the estimation error approaches zero. Therefore, the performance function  $\xi$  that makes the filter output approach towards the desired signal and permits the analysis of a discrete filter [14], can be expressed as

$$\xi = E[|e_k|^2] \tag{1.4}$$

where  $E[\cdot]$  denotes the expectation operator. The performance function  $\xi$  is the so-called mean squared error criterion. In mathematical perspective, the performance function can be handled easily if it has single global minimum or maximum point. Therefore, the performance function  $\xi$  is hyper-paraboloid that has a single global minimum point and can be calculated easily with second order statistics of random process.

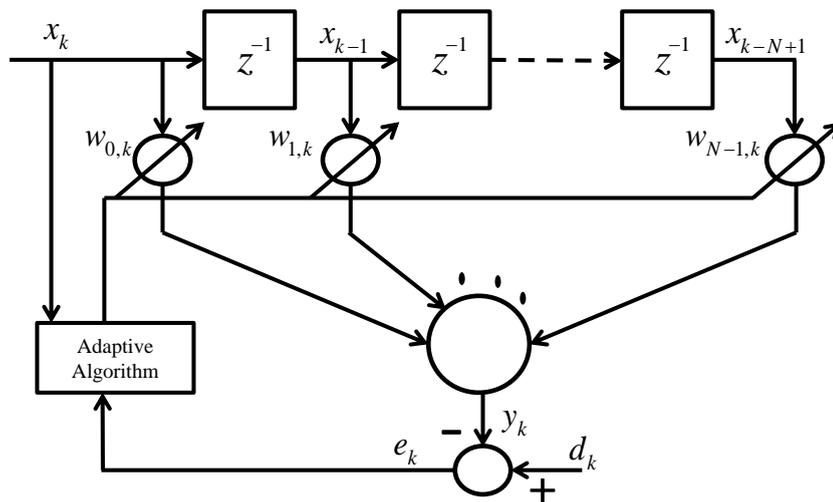


FIGURE 1.4: Adaptive FIR filter

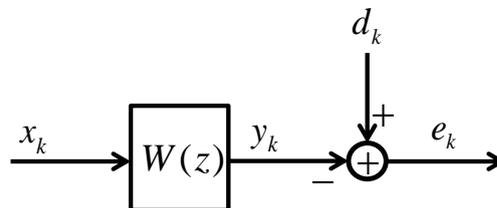


FIGURE 1.5: Block diagram of adaptive filtering problem

## 1.7 Wiener Filter Solution

The Wiener filter is used to develop an analytical non-recursive solution for the minimization problem of the mean squared error between the desired process and the estimated random process. The gradient and least square based approaches follow the mean squared error criterion and converge to Wiener optimal solution as well. To develop the solution of Wiener filter, we can rewrite Eq. (1.3) into vector form, as

$$y_k = \mathbf{w}_k^T \mathbf{x}_k \quad (1.5)$$

The performance function in Wiener filter solution is based on mean squared error (MSE) criterion that can be expressed in simplified form as

$$\begin{aligned} E[e_k]^2 &= E[d_k - \mathbf{w}_k^T \mathbf{x}_k]^2 \\ &= E[d_k^2] - 2\mathbf{w}_k^T E[\mathbf{x}_k d_k] + \mathbf{w}_k^T E[\mathbf{x}_k \mathbf{x}_k^T] \mathbf{w}_k \\ &= r_{dd} - 2\mathbf{w}_k^T \mathbf{r}_{xd} + \mathbf{w}_k^T \mathbf{R}_{xx} \mathbf{w}_k \end{aligned} \quad (1.6)$$

where  $r_{dd}$  is the mean square value of the desired signal,  $\mathbf{R}_{xx}$  is the input signal autocorrelation matrix and  $\mathbf{r}_{xd}$  is the cross-correlation vector. The optimum Wiener filter coefficients vector,  $\mathbf{w}_o$ , can be obtained by taking the gradient of Eq. (1.6) and setting the equation equal to zero, as

$$\mathbf{w}_o = \mathbf{R}_{xx}^{-1} \mathbf{r}_{xd} \quad (1.7)$$

To obtain the unique solution of Wiener filter, it must be noted that the number of filter coefficients must be equal to or greater than the number of multipath components which are present between the transmitter and the receiver.

## 1.8 Adaptive Filtering Algorithms

There are many approaches that are used to design the adaptive filtering algorithm. However, the most important approaches for the design and development of an

adaptive algorithm are as follows:

### 1.8.1 The Stochastic Gradient

The stochastic gradient approach uses the transversal or FIR structure for the designing of adaptive filtering algorithm. In this approach, the solution of the minimization problem of MSE criterion can be found by using the optimization method which is known as steepest decent. In steepest decent algorithm, the error function is minimized gradually after its gradient vector is taken. The most common Least Mean Square (LMS) adaptive filtering algorithm [13, 14] also works on the principle of stochastic gradient which was derived by B. Widrow in 1959. The LMS adaptive algorithm is the recursive form of Wiener filter solution which provides the computational complexity of order  $O(N)$  and gradually forces the error function towards minimum. The update equation of the weights of LMS filter can be written as

$$w_k = w_{k-1} + 2\alpha e_k x_k \quad (1.8)$$

where  $\alpha$  is the step size parameter. The convergence of LMS filter is basically dependent on the step size parameter. The selection of step size parameter in time varying environment is one of the promising tasks. The in-appropriate selection of step size while estimating an unknown parameters may cause miss-adjustment to the algorithm or may make it slower to converge. Likewise, the variants of LMS algorithm (e.g., Normalized LMS, Variable Step-Size LMS, Signed LMS, Leaky LMS [15], etc.) also have their own limitations and provide slow convergence. Nevertheless, all stochastic gradient based algorithms entail low computational complexity as well.

### 1.8.2 Kalman Filter

The Kalman filter is an extension of Wiener filter solution and it recursively minimizes the mean squared error function [14]. It is also known as linear quadratic

estimation (LQE). The algorithm in Kalman filter is based on the set of mathematical equations which provides an optimal estimate of the state of a process. The optimal estimate provided by Kalman filter is more accurate than those which are based on single measurement alone. The Kalman filter provides good tracking performance on behalf of very high computational complexity of order  $O(N^3)$ . The Kalman filter is too powerful filtering algorithm in many aspects, e.g. it precisely estimates the model parameters even when the system model is unknown or not available, and it can predict and estimate the future and present states, respectively. Nevertheless, the high computational complexity of Kalman filter restricts the filtering algorithm to be implemented sequentially on low-cost computationally-constrained small platforms.

### 1.8.3 Least Square Estimation

The least square estimation approach is based on minimization of an error function which is described as the weighted sum of error squares. This approach can be classified into two main categories, namely: 1) block estimation 2) recursive estimation [13]. In block estimation, the input data sequence is divided into blocks of equal time length and then processing procedure proceeds block by block. On the other hand, in recursive estimation, the processing procedure operates on the individual input data sample rather than a complete input data block. Therefore, the recursive estimation technique is more popular because of its low memory usage. The most popular Recursive Least Square (RLS) adaptive algorithm works on the principle of recursive approach and it can be seen as the special case of Kalman filter. The update equation of the weights of RLS filter can be written as

$$w_k = w_{k-1} + g_k e_k \quad (1.9)$$

where  $g_k$  is the Kalman gain at time instant  $k$ . As compared to gradient based approach, RLS filter and its variants (e.g., Modified RLS, Fast RLS, Extended RLS, Robust RLS, etc.) provide fast convergence performance on behalf of enhanced computational complexity of order  $O(N^2)$ .

## 1.9 Major Applications of Adaptive Filters

There are many applications of adaptive filters in signal processing domain [16–21]. The major applications of adaptive filters that are within the scope of this work are as follows:

### 1.9.1 MIMO Channel Estimation

As explained in the earlier section, MIMO antenna systems are used to increase the system capacity rather than increasing the operational bandwidth. In order to increase or improve the data transmission rate, the receiver must have the knowledge about the current channel state information (CSI) [22]. In frequency and time selective fading, the transmitted signal suffers from time-varying distortion in amplitude as well as in phase due to the multipath interference. To overcome such distortions, current channel state information is used in the detection process. For this purpose, some known bits are transmitted at the start of every communication session and channel is estimated by using the knowledge of the received and transmitted signals. In this context, adaptive filters are used which play a vital role to find unknown channel characteristics. The length of the training sequence and the duration of the communication session can be adjusted according to the coherence time of the channel, and the convergence analysis of the adaptive filter. The convergence performance of these adaptive filters is dependent upon how better they make use of training sequences. Therefore, to avail high data rate and better system performance, adaptive algorithms with fast convergence performance and low computational complexity are demanded.

### 1.9.2 System Identification in Case of Missing Data

Recursive least square based approaches are widely used in many applications, e.g. system identification or system parameters estimation while most of them are working on the assumption that the input-output data are available at every

sampling instant. However, in missing data system, the subset of output data is available rather than the complete output information restricts the directly use of standard identification algorithm. Therefore, in missing data system, the stochastic framework may only be used as reference model rather than the use of FIR filter [23]. Moreover, in missing data systems, the recursive algorithm continuously provides the estimated output data at every sample point. Therefore, due to high computational complexity of the adaptive algorithm, the efficiency of the system may degrade its performance. Hence, the distributed adaptive filtering is only being the choice which runs the adaptive algorithm distributively and provides a significant impact on the efficiency of the missing data systems.

## 1.10 Distributed Wireless Sensor Networks

A distributed wireless sensor network performs decentralized processing by using wireless sensor nodes which constitute the distributed network. In a distributed wireless sensor network, a group of nodes collaboratively estimates the parameters of interest. Therefore, a distributed wireless sensor network exhibits higher significance due to its applicability in many real life applications such as big data, environmental monitoring, distributed estimation, social networks, etc. Various strategies such as consensus, diffusion and incremental are used to connect the nodes in the distributed wireless sensor network. In consensus strategy [24], two time scales are used: one time scale for the collection of measurements across the distinct nodes while other time scale to iterate sufficiently enough over the collected data to accomplish agreement before going to next iteration. On the other hand, the diffusion strategy [25] uses only one time scale which provides a better performance as compared to consensus scheme. However, both the schemes have the best use in the application of distributed channel estimation. Furthermore, in incremental approach [26], each node in the network communicates with adjacent node, and consequently the data is processed cyclically throughout the distributed network. The model diagram of incremental network is shown in Fig. 1.6. The selection of number of nodes in the incremental network is based on the number of

iterations which are used for the complete convergence of the adaptive algorithm. Although all the distributed schemes exhibit high computational complexity as compared to the same high definition algorithm being run sequentially on a single node; however, their computational cost is distributed among all participating nodes.

## 1.11 Research Aim

The limited memory and energy constraint restrict a wireless sensor node to run the high definition adaptive filtering algorithm. Therefore, for the effective utilization of existing power and limited memory, the complexity of the adaptive algorithm must be reduced while implementing it on an energy-limited and computationally-constrained platform. In this context, the aim of this research is twofold: one of its objectives is to develop the low-complexity adaptive filtering algorithm that may be run as standalone on a wireless sensor node and the other objective is to develop distributed solutions for a group of low-cost computationally-constrained nodes to run computationally-expensive procedures parallelly which provide a significant impact on aggregate computational complexity as well as on the energy consumption of an individual node.

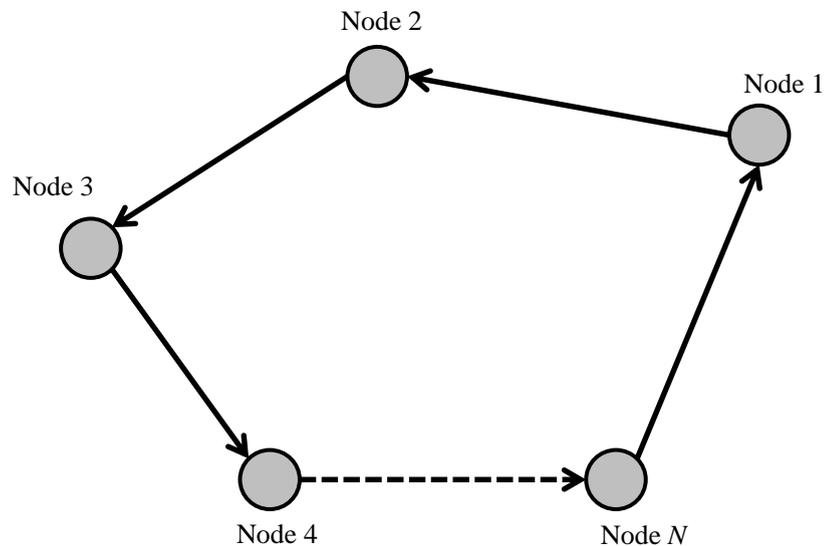


FIGURE 1.6: Distributed adaptive incremental strategy

## **1.12 Research Contributions**

The research contributions presented in this doctoral thesis can be listed in the following subsections.

### **1.12.1 Low Complexity Linear MIMO Channel Estimation Algorithm**

A new low-complexity MIMO channel estimator is introduced with the consideration of block fading channel environment. The proposed algorithm provides independency on forgetting factor parameter and entails lesser multiplication computations than RLS algorithm and variants.

### **1.12.2 Parallel Distributed Adaptive RLS Filtering**

A new time non-aligned distributed architecture for computationally-constrained small platforms is introduced which runs the complex adaptive algorithm parallelly. The proposed architecture provides parallelly much reduced complexity and processing time than the sequentially-operated algorithms.

### **1.12.3 Parallel Distributed Diffusion-based Adaptive RLS Filtering**

A new time aligned distributed-diffusion based architecture for computationally-constrained small platforms is introduced which runs the complex adaptive algorithm distributively. The proposed architecture provides parallelly linear computational complexity and gives significant improvement in decreased processing time than the sequentially-operated least square algorithms.

#### 1.12.4 Application of Diffusion-based RLS Architecture on Missing Data Systems

The proposed distributed diffusion-based architecture is implemented for system identification problem with missing data. It is observed that the proposed architecture provides parallelly much reduced complexity and processing time than the sequentially-operated algorithms. It is also observed that PDASP that uses non-aligned time indexes does not work for missing data application.

### 1.13 Organization of the Thesis

The rest of the thesis report is organized as follows: In Chapter 2, a detailed literature survey is presented that can be classified into two main categories, i.e. adaptive filter theory and adaptive filter implementation over low-cost platforms. Furthermore, the research motivation and problem formulation is also described at the end of this chapter. In Chapter 3, the system model is presented which describes the methodology of the MIMO-capable cluster head and its signal processing capabilities. Furthermore, the time varying MIMO channel model in terms of first order Markov process with the assumption of unit power channel matrix is presented. In Chapter 4, a low-complexity MIMO channel estimator is introduced. The proposed algorithm is derived through the expression of estimation error and enlarge into filter weight matrix by considering the assumption of block fading. Chapter 5 and 6 provide distributed solutions for the complex sequential adaptive filtering algorithm with time non-aligned and time aligned strategies, respectively. In Chapter 7, the problem related to output error models of data estimation and parameter identification for missing data systems is presented. Finally, Chapter 8 provides a summary of the thesis, presents the concluding remarks and discusses the future research work depending upon the obtained results of this thesis.

# Chapter 2

## Literature Survey and Problem Formulation

The literature survey presented in this chapter can be classified into two main categories, namely: 1) adaptive filter theory 2) implementation of adaptive algorithm over low-cost platforms. The research motivation and research methodology are also described at the end of this chapter.

### 2.1 Literature Survey

High-speed internet has a major impact on the socioeconomic development of the world. Online banking services, mobile-operated transactions and web based interactions demand seamless connectivity to the internet. Multiple Input Multiple Output (MIMO) systems are turning into the promising tools that can ensure to resolve the problems raised by the capacity bottleneck of the present and future fast data-rate communications. Even with the use of multiple antennas at the transmitting and receiving ends, high data-rates can be achieved without corresponding increase in the transmit power or the bandwidth. Furthermore, the rapid growth of wireless services has presented unbound innovative challenges in terms of high data rates and larger capacities for future mobile communications [27].

Utilization of space time coding [28–30] and antenna arrays [31] provides a new paradigm in signal processing research that further encourages better estimation techniques to address the above mentioned challenges. Although the problems related to MIMO channel estimation have been studied deeply in [32–35]; however, adaptive algorithms still need to be dealt in comprehensively in term of reduction in their computational complexities.

In order to address the two-fold research aim presented in Chapter 1, the literature documented in this dissertation spans over two major areas, i.e. adaptive filter theory and adaptive algorithm implementation over low-cost platforms.

### 2.1.1 Adaptive Filter Theory

Adaptive filtering techniques play a very important role in the emerging fields of science and technology, however, from the last two decades have witnessed tremendous research in the field of adaptive filtering for the improvement of their convergence and complexity requirements [36–38]. However, achieving fast convergence on an energy-constrained platform still remains a dream in spite of magnificent advancements in Integrated Circuit (IC) technologies. In [39], a maximum likelihood MIMO channel estimator is introduced, where the expressions of the estimator algorithm are derived by assuming stationary and quasi-stationary channel environments. However, in quasi-stationary environment, the algorithm shows undesirable behavior that exerts a critical impact on the convergence performance of the algorithm. Furthermore, the tracking performance of the maximum likelihood MIMO channel estimator is improved in [40] by modifying the structure of the algorithm. A major drawback of this algorithm is the need of *a priori* knowledge of the channel model. A number of linear adaptive algorithms are used for channel estimation that include Recursive Least Square (RLS), Recursive Least M-estimation (RLM) [41] and Robust RLS (RRLS) [42] algorithms. If we take MIMO channel estimation as an example application for our research work, some of the filtering algorithms like RRLS and RLM find their use only in SISO systems, because of their single-error constraint. The well known RLS algorithm has been in use for

the applications of channel estimation and channel equalization since its advent [14, 43, 44]. In the stationary environment, RLS algorithm provides good tracking performance by assuming the normal underlying noise. However, for time varying channel conditions, the fixed value of forgetting factor in RLS algorithm is not suitable to find unknown channel coefficients [44, 45]. This exerts adverse impact on the memory utilization of the algorithm and forces the algorithm to converge slowly. In order to improve the tracking performance of the RLS filtering algorithm for time varying channel conditions, a modified adaptive forgetting-factor RLS (MAFF-RLS) is introduced in [46]. However, the forgetting factor approximation in the algorithm severely degrades its tracking performance which is a major drawback of this algorithm. Furthermore, in [47], a decision directed RLS (DD-RLS) for MIMO channel tracking is introduced, where, a lower complexity was claimed with reference to the higher complexity of a Kalman filter. However, the DD-RLS is also not economic or feasible because of its enormous computational complexity for the finding channel coefficients. Likewise, in [48], a modified RLS algorithm is introduced which provides more enhanced tracking performance than the conventional RLS algorithm; however, the complexity of the modified RLS algorithm is greater than that of conventional RLS algorithm which makes a critical impact on the processing time of the algorithm. In [49], optimum weighted RLS (OP-RLS) algorithm is proposed for the estimation of rapid fading MIMO channel. However, the OP-RLS also suffers from an enhanced computational complexity. Furthermore, in [50], a robust recursive weighted least squares with both scale and variable forgetting factors (RRWLSV) is introduced. The RRWLSV algorithm outperforms recursive least square variable forgetting factor (RLSVF) and robustified recursive least square with scale factor (RRLSS) adaptive filtering algorithms in impulsive noise environment. The enhanced convergence performance provided by RRWLSV is on behalf of complex computational complexity which also provides a critical impact on the processing time of the algorithm. On the other hand, gradient based techniques proposed in [51] provide least mean square (LMS) solutions to tackle time-varying MIMO channels. However, the proposed

solutions are affected by the intrinsic fact that the gradient based algorithms converge slower than the recursive least square based schemes. Similarly, in adaptive multiuser detection, the same complexity problems appear, where matrix inversions exist in both MMSE and decorrelating detectors [52]. On the other hand in [53], a reduced rank linear interference suppression provides low-complexity solution with the use of polynomial expansion (PE), where the inverse of matrix is represented by  $P^{th}$  order matrix polynomial [52–55]; however, the selection of  $P$  is one of the promising tasks because its selection depends on the tradeoff between detection performance and complexity. In [56, 57], Banachiewicz inversion formulation is used to perform the matrix inversion for MIMO-OFDM based software defined radio (SDR) signal detection. The inversion of a  $4 \times 4$  matrix is divided into four  $2 \times 2$  matrices that reduce the computational operations. Likewise, the authors in [58] derive a low-complexity algorithm for Hermitian positive-definite recursive matrix inversion that provides lower computational complexity than [56] and [57] with the utilization of 52 operations for finding matrix inversion only. Nevertheless, using the concepts proposed in [56–58] for matrix inversion do not exhibit a significant impact on the computational cost of high definition adaptive filtering [42]. Meanwhile, many other approaches like a polynomial process model or autoregressive (AR) model (see ,e.g., [49, 59] and the references therein) for the time varying environment were also proposed. However, these algorithms provide enhanced performance to some extent in certain applications compared to RLS algorithm. In nutshell, most of the algorithms have major limitations such as high computational complexity and prior knowledge of noise power.

### 2.1.2 Implementation of Adaptive Algorithm Filtering over Low Cost Platforms

A typical wireless sensor node generally includes the battery, wireless transceiver, micro-controller and sensor unit. The limited memory and power constraints restrict the wireless sensor node to run the high definition adaptive algorithm on a single unit. On the other hand, the distributed solutions may allow the group of

wireless sensor nodes to run the high definition adaptive filtering algorithms collaboratively and may provide a significant reduction in the aggregate computational complexity while enhancing the performance of those low-cost devices. Several distributed filtering techniques have been introduced in the literature with the target to reduce the high computational complexity of the adaptive algorithm which permit the low-cost devices to run the high definition algorithms distributively. In [60], Lin *et al.* introduced an LR-MMSE algorithm based on QR decomposition and complex lattice reduction (CLR) which provides 35.5% lesser computational complexity than MMSE based scheme [61]; however, 35.5% lesser computational cost still restrict the computationally-constrained low-cost sensor node to run this filtering algorithm on a single unit. In [62], a comparison of renowned subband adaptive filtering (SAF) structures is presented with parallel arrangement of multi-rate filter banks. The SAF technique based on adaptive noise cancelation exhibits reduced complexity through the use of least mean square (LMS) adaptive filtering algorithm in acoustic noise environment. Therefore, due to phase, aliasing and amplitude distortions and extra processing delay, these systems may be ruled out for real time implementation. Another architecture configuration for reducing runtime is the MMSE signal estimation using wireless sensor nodes [63]. In this architecture, authors use the distributed adaptive node-specific signal estimation (DANSE) technique to estimate the channel coefficients by following Wiener Hopf equation. However, DANSE technique only follows the MMSE criterion rather than running the adaptive filtering algorithm. This makes DANSE incapable of estimating time-varying channel conditions.

Distributed network based architecture provides improved performance for many communication applications, such as environmental monitoring, channel estimation and source tracking [64–67]. The distributed and energy aware strategies for estimation over networks can be classified into two main groups, i.e., consensus strategies and incremental cooperation mode strategies [68]. In consensus strategies [69, 70], the procedure of the estimation is done in two stages. Unfortunately, this kind of implementation provides an increase of communication load and is not suitable for real time estimation in time varying channel environments. On the

other hand, in incremental strategies [26, 71–76], each node in the network communicates only with one neighbor node, and consequently the data is processed cyclically throughout the distributed network. Moreover, in incremental mode, all the nodes entail the complex computational complexity of the adaptive algorithm and each node in the network is being free for  $K - 1$  iterations, where  $K$  is the total number of iterations required for the complete convergence of the adaptive filtering algorithm. Furthermore, in [77], a distributed RLS (DRLS) algorithm with reduced communication load is introduced. However, in this technique, the RLS algorithm works in sequential form because each respective node waits until the information is not retrieved from the previous processing node. Therefore, this technique does not show a significant role in sense of reducing the computational complexity or even cannot run the algorithm in parallel fashion. Furthermore, the stability and performance analysis of DRLS algorithm is presented in [78]. The experimental validation of the DRLS algorithm only focusses on the performance and stability rather than the complexity or the processing time of the algorithm. Furthermore, in distributed estimation, different approaches [79–83] such as diffusion based techniques are used to regulate the information among different nodes. In this framework, each node in the network uses a local algorithm to update the desired parameters and then exchange the information towards central node to update the estimation. The diffusion techniques provide high computational cost and cause an increase of huge communication load while exchanging the information among the nodes. For example, in an  $M$ -node diffusion network, the communication load is written by  $M \times N$  times; where  $N$  shows the dimension of diffused vector. Therefore, this multiplication factor implies a crucial impact on the communication burden. Moreover, large neighborhood nodes require more power for information interchange [64–67] that is a major drawback of these distributed diffusion-based architectures.

## 2.2 Research Motivation and Problem Statement

As discussed earlier, vast research has been done on devising techniques for using wireless sensor nodes to monitor various parameters in unattended areas. However, implementation of high definition adaptive filtering algorithm over these low-cost wireless sensor nodes still remains a dream due to their low energy and limited memory constraints. The incremental strategies [26, 71–76] and DRLS [77] provide the distributed platforms that make these nodes run the high definition adaptive filtering algorithm distributively. In incremental strategies [26, 71–76], all of the  $K$  iterations of the adaptive algorithm required for its convergence are assigned to  $K$  nodes. Likewise, in DRLS [77], parts of the adaptive algorithm are assigned to the desired number of nodes to run the adaptive algorithm distributively. However, both techniques work in sequential form because each respective node waits till the information is retrieved from the previous processing node. Furthermore, in order to accommodate distributed processing Matlab<sup>©</sup> [84] and Labview<sup>©</sup> [85] use software parallelism which is available in various architectures of the present fast processing multi-core computers with perfect time-alignment processors. These software provide parallel processing toolbox to divide the large problems into smaller computations, hence requiring reduced running time. Likewise, graphical processing unit (GPU) enables to run high definition graphics on a personal computer (PC) by exploiting hundreds of cores [86]. Furthermore, compute unified device architecture (CUDA) [87] is NVIDIA's GPU architecture which provides multithreaded applications where cores can communicate and exchange information with each other. However, these cores have not been used to run adaptive algorithms in parallel with non-aligned time indexes due to being the components of a single system clock. According to the best of our knowledge; there is no parallel structure of recursive adaptive filtering algorithms in the literature where any of the complex adaptive algorithms runs in parallel fashion over computationally-constrained platforms with no perfect time alignment.

## 2.3 Research Methodology

The methodology to conduct the proposed research is divided into three stages. In the first stage of our research, we intend to reduce the complexity of adaptive filtering by proposing a sequential low complexity MIMO channel estimation algorithm for the time varying MIMO channel environments. For this purpose, we aim to take Kalman filter approach and modify it into low complexity algorithm. In order to simplify the system model, we assume block fading channel environment which is a valid assumption in all modern wireless communication systems. In the second stage, we aim to make a network based on low-cost computational-constrained wireless sensor nodes to run a high definition algorithm. For this purpose, we aim to propose a parallel distributed adaptive signal processing (PDASP) architecture using non-aligned time indexes. Finally, in the third stage, we intend to provide a distributed solution for perfect time aligned systems. In this regard, we aim to propose a distributed diffusion based adaptive signal processing (DDASP) architecture using aligned time indexes. The proposed solutions will be based on diffusion [25, 79–83] and distributed incremental strategies [26, 73, 74, 76] which provide the distributed platforms that run the high definition adaptive filtering algorithm distributively.

# Chapter 3

## System and Channel Model

This chapter presents the description of system and channel model for MIMO communication system. Section 3.1 describes the methodology of the MIMO capable cluster head and its signal processing capability. Section 3.2 presents the MIMO channel model with respect to frequency and time selective fading characteristics.

### 3.1 System Model

Consider a MIMO-capable cluster head which is engaged in communication with a far gate way is shown in Fig. 3.1. The MIMO-capable cluster head is considered to be the specific purpose node. As it is discussed earlier that the multipath fading characteristics and the number of MIMO antennas are directly dependent on the computational complexity of the adaptive signal processing algorithm. Therefore, the selection of sequential or distributed adaptive signal processing technique for MIMO channel estimation is dependent on the processing capability of MIMO-capable cluster head which is as follows:

- If the MIMO-capable cluster head has large signal processing capability then it uses the sequential low-complexity MIMO channel estimation algorithm as on a single unit which is presented in Chapter 4.

- If the MIMO-capable cluster head has limited signal processing capability then the specific purpose MIMO-capable cluster head iteratively gives the received data vector towards the general purpose PDASP or DDASP architecture nodes which are described in Chapter 5 and 6, respectively. The PDASP or DDASP architecture runs the highly complex adaptive filtering algorithm parallelly and send back the desired information of channel estimation towards the MIMO-capable cluster head.

### 3.2 MIMO Channel Model

An  $N_{\mathcal{T}} \times N_{\mathcal{R}}$  MIMO communication system with  $N$  independent spatial streams is considered, where the received signal  $y_k^{(m)}$  at the  $m^{\text{th}}$  ( $m = 1, \dots, N_{\mathcal{R}}$ ) receiver element at time index  $k$  can be written as

$$y_k^{(m)} = \sum_{n=1}^{N_{\mathcal{T}}} h_k^{(n,m)} x_k^{(n)} + \vartheta_k^{(m)} \quad (3.1)$$

where  $x_k^{(n)}$  is transmitted signal from  $n^{\text{th}}$  transmitting antenna,  $h_k^{(n,m)}$  is channel attenuation coefficient between  $n^{\text{th}}$  transmitter and  $m^{\text{th}}$  receiver and  $\vartheta_k^{(m)}$  is white noise at  $m^{\text{th}}$  receiving element. The autocorrelation of the channel parameters

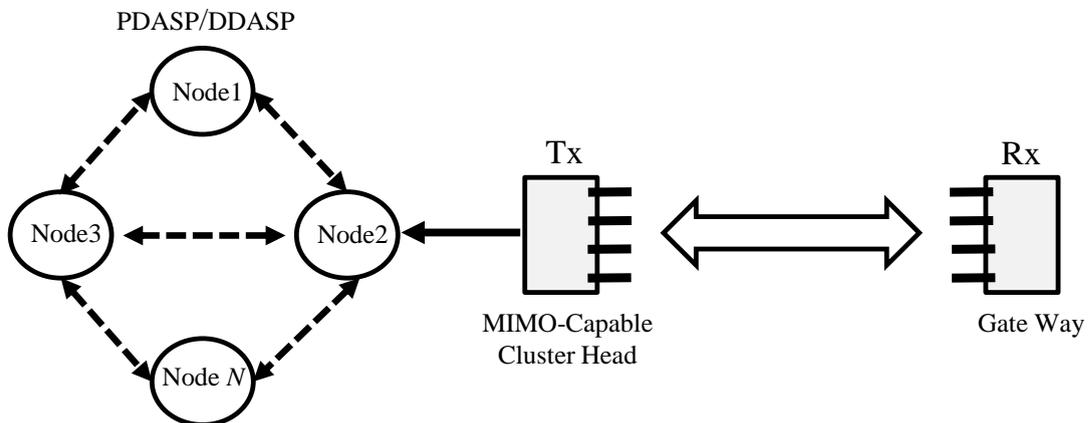


FIGURE 3.1: System model for MIMO communication system.

during block fading transmission [88] can be written as

$$E\{h_k^{(n,m)}[h_l^{(n,m)}]^*\} \cong Jo(2\pi f_D^{(n,m)}T | k - l |) \quad (3.2)$$

where  $E(\cdot)$  is the expectation operator, superscript “ $*$ ” represents the conjugate operator,  $Jo(\cdot)$  is first kind zero order Bessel function,  $f_D^{(n,m)}$  is Doppler frequency and  $T$  is symbol duration. According to uncorrelated scattering model [89], all channel streams are independent and approximately written by the following Autoregressive (AR) model of order  $C$ .

$$h_k^{(n,m)} = \sum_{c=1}^C \beta_c^{(n,m)} h_{k-c}^{(n,m)} + \omega_k^{(n,m)} \quad (3.3)$$

where  $\beta_c^{(n,m)}$  is  $c^{th}$  AR coefficient of the channel coefficient in between  $n^{th}$  transmitting and  $m^{th}$  receiving antenna at time epoch,  $k$  and  $\omega_k^{(n,m)}$  is independent identically distributed (i.i.d) zero mean Gaussian process. The variance of i.i.d Gaussian process can be written as

$$E\{\omega_k^{(n,m)}[\omega_k^{(n,m)}]^*\} = \sigma_{\omega_k^{(n,m)}}^2 \quad (3.4)$$

and

$$\sigma_{\omega_k^{(n,m)}}^2 = 1 - \sum_{c=1}^C (\beta_c^{(n,m)})^2 \quad (3.5)$$

The selection of optimal parameters for AR channel model can be achieved using (2) through the following Wiener equation [90],

$$Jo(2\pi f_D^{(n,m)}T | k - t |) = \sum_{c=1}^C Jo(2\pi f_D^{(n,m)}T | k - c - t |) \beta_c^{(n,m)} \quad (3.6)$$

for  $t = k - C, k - C + 1, k - C + 2, \dots, k - 1$ .

Due to the assumption of unit power channel matrix coefficients [91], the power of each time-varying channel coefficient can be written as

$$\sum_{m=1}^M \sum_{l=0}^{L-1} E[|h_k^{(n,m)}|^2] := 1 \quad \forall n, m \quad (3.7)$$

where  $L$  is the total number of multipaths. The time variation of channel is typically dependent on the velocity of the mobile user and its respective Doppler shift in the carrier frequency. A viable assumption, that is standard in vast majority of cases is

$$f_D^{(n,m)} = f_D \quad \forall n, m \quad (3.8)$$

Due to the viable assumption used in Eq. (3.8), the AR coefficient  $\beta$  [92] is assumed to be same for all  $f_D^{(n,m)}$  which can be written as

$$\beta = Jo(2\pi f_D T) \quad (3.9)$$

The smaller value of  $\beta$  corresponds to large Doppler shift which leads to fast variations in the channel. Due to large or increased Doppler shifts, a signal undergoes time-selective fading or fast fading. Therefore, to accommodate those variations, Markov chain of higher order can be used in Eq.(8). However, for the purpose of keeping simplicity, in this article, Markov chain of first order is applied in analysis as well as in simulations. The time varying attitude of the channel matrix  $\mathbf{H}_k$  can be written in terms of first order Markov process [93],

$$\mathbf{H}_k = \beta \mathbf{H}_{k-1} + \boldsymbol{\Omega}_k \quad (3.10)$$

where  $\boldsymbol{\Omega}_k$  is a matrix of i.i.d. Gaussian processes with variance  $\sigma_{\boldsymbol{\Omega}}^2$ . With the assumption made in (3.7), the variance of i.i.d. Gaussian processes can be written as

$$\sigma_{\boldsymbol{\Omega}}^2 = 1 - \beta^2 \quad (3.11)$$

Due to parallel interference [94], Eq. (3.1) can be expressed in matrix form

$$\mathbf{y}_k = \mathbf{H}_k^H \mathbf{x}_k + \mathbf{v}_k \quad (3.12)$$

where

$$\mathbf{H}_k = \begin{bmatrix} h_k^{(11)} & h_k^{(12)} & \dots & h_k^{(1N_{\mathcal{R}})} \\ h_k^{(21)} & h_k^{(22)} & \dots & h_k^{(2N_{\mathcal{R}})} \\ \vdots & \vdots & \ddots & \vdots \\ h_k^{(N_{\mathcal{T}}1)} & h_k^{(N_{\mathcal{T}}2)} & \dots & h_k^{(N_{\mathcal{T}}N_{\mathcal{R}})} \end{bmatrix}$$

is  $N_{\mathcal{T}} \times N_{\mathcal{R}}$  channel matrix,  $\mathbf{x}(n) = [x_k^{(1)} \ x_k^{(2)} \ \dots \ x_k^{(N_{\mathcal{T}})}]^T$  is the transmitted signal vector and  $\mathbf{v}_k = [v_k^{(1)} \ v_k^{(2)} \ \dots \ v_k^{(N_{\mathcal{R}})}]^T$  is zero-mean Additive White Gaussian Noise (AWGN) with variance  $\sigma_v^2$ .

Furthermore, if the bandwidth of the channel,  $B_C$  is smaller than the bandwidth of the transmitted signal,  $B_x$ , the channel undergoes frequency selective fading. Due to the frequency selectivity, the multiple versions of transmitted waveform which are faded and delayed in time are added into the received signal. Thus, the channel induces inter-symbol interference (ISI). To summarize, the channel induces ISI if

$$\begin{aligned} B_x &> B_C \\ T &< \sigma_\tau \end{aligned} \tag{3.13}$$

where  $B_x$  is the signal bandwidth,  $B_C$  is the channel bandwidth and  $\sigma_\tau$  is the rms delay spread. Therefore, due to the multi-path components that exist on the

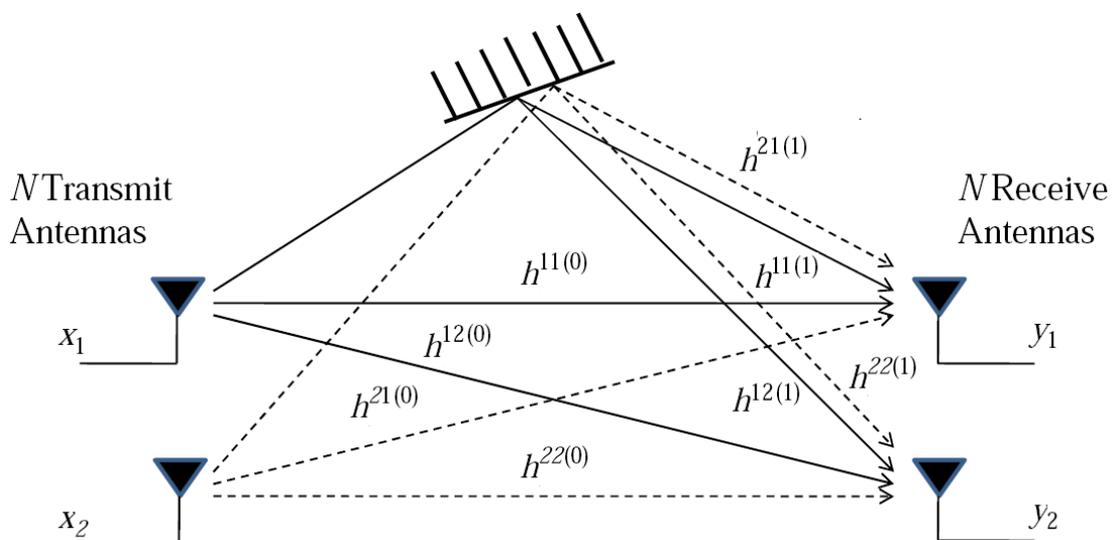


FIGURE 3.2: Frequency-selective channel model for MIMO communications.

$$\tilde{\mathbf{H}}_k = \begin{bmatrix} h_k^{(11(0))} & \dots & h_k^{(11(L))} & h_k^{(21(0))} & \dots & h_k^{(N_{\mathcal{T}}1(L))} \\ h_k^{(12(0))} & \dots & h_k^{(12(L))} & h_k^{(22(0))} & \dots & h_k^{(N_{\mathcal{T}}2(L))} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ h_k^{(1N_{\mathcal{R}}(0))} & \dots & h_k^{(1N_{\mathcal{R}}(L))} & h_k^{(2N_{\mathcal{R}}(0))} & \dots & h_k^{(N_{\mathcal{T}}N_{\mathcal{R}}(L))} \end{bmatrix}$$

$$\tilde{\mathbf{W}}_k = \begin{bmatrix} w_k^{(11(0))} & \dots & w_k^{(11(L_e))} & w_k^{(21(0))} & \dots & w_k^{(N_{\mathcal{T}}1(L_e))} \\ w_k^{(12(0))} & \dots & w_k^{(12(L_e))} & w_k^{(22(0))} & \dots & w_k^{(N_{\mathcal{T}}2(L_e))} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ w_k^{(1N_{\mathcal{R}}(0))} & \dots & w_k^{(1N_{\mathcal{R}}(L_e))} & w_k^{(2N_{\mathcal{R}}(0))} & \dots & w_k^{(N_{\mathcal{T}}N_{\mathcal{R}}(L_e))} \end{bmatrix}$$

way between transmit and receive antennas, the channel matrix  $\tilde{\mathbf{H}}_k$  becomes an  $N_{\mathcal{R}} \times (N_{\mathcal{T}} + N_{\mathcal{R}}(L))$  matrix which can be written on the top of the page, where  $L$  is the total number of multipaths, including line of sight (LoS). The dimensions of  $\tilde{\mathbf{H}}_k$  are not only dependent on the number of transmit and receive antennas but also on the number of multi-path components that exist between the transmit and receive antennas, as shown in Fig. 3.2.

Each entry of the channel matrix  $\tilde{\mathbf{H}}_k$  is now  $h_k^{(trl)}$ , where  $t = 1, 2, \dots, N_{\mathcal{T}}$ ,  $r = 1, 2, \dots, N_{\mathcal{R}}$  and  $l = 1, \dots, L$ . Likewise,  $\mathbf{x}_k$  also changes to  $\tilde{\mathbf{x}}_k = [x_k^{(1)} \dots x_{k-(L)}^{(1)} x_k^{(2)} \dots x_{k-(L)}^{(2)} \dots x_{k-(L)}^{(N_{\mathcal{T}})}]^{T}$  which is a concatenated transmitted signal vector with elements,  $x_k^{(i)}$ , where  $i$  shows the index of the transmit antenna element and  $k$  shows the time epoch. Time epochs other than current provide ISI in the model. From now onward “ $\sim$ ” will be used to denote the values that incorporate multipath processing. Furthermore, considering block fading, the multipath channel is assumed to be constant over a specific block. Likewise of channel matrix, the estimated channel matrix  $\tilde{\mathbf{W}}_k$  through the adaptive algorithm is the same as that of multipath channel matrix  $\tilde{\mathbf{H}}_k$  is shown on the top of the previous page.

# Chapter 4

## Low-Complexity Channel Estimation for MIMO Communication Systems

In this chapter, a new recursive Multiple Input Multiple Output (MIMO) channel estimation is introduced which is based on the recursive least square solution. Section 4.1 presents the overview of the proposed algorithm. Section 4.2 describes the formulation of proposed low-complexity MIMO channel estimation algorithm. Section 4.3 presents the complexity analysis and finally, Section 4.4 describes the simulation results and discussion.

### 4.1 Overview

Channel estimation is employed to get the current knowledge of channel states for an optimum detection in fading environments. The proposed algorithm is derived through the expression of estimation error and enlarge into filter weight matrix by considering the assumption of block fading environment. The proposed recursive algorithm utilizes short training sequence on one hand and requires low computational complexity on the other hand. It is pertinent to note that the

length of training sequence is dependent on the convergence performance of the adaptive algorithm. The proposed algorithm provides fast convergence as compared to Recursive Least Square (RLS) and Robust Variable Forgetting Factor RLS (RVFF-RLS) adaptive algorithms while utilizing lesser computational cost and provides independency on forgetting factor  $\lambda$ . Moreover, the proposed low complexity MIMO algorithm can be run as distributed or standalone depending upon the processing capability of the MIMO capable cluster head. The distributed architectures which can be used to run the low complexity MIMO algorithm parallelly on a number of nodes are discussed in Chapter 5 and Chapter 6.

## 4.2 Proposed Low-Complexity MIMO Algorithm

According to MIMO frequency selective fading channel impairments, Eq. (3.12) can also be written as

$$\mathbf{y}_k = \widetilde{\mathbf{H}}_k \widetilde{\mathbf{x}}_k + \mathbf{v}_k \tag{4.1}$$

The estimation error  $\mathbf{e}_k$  which is the difference of the received sequence  $\mathbf{y}_k$  and the estimate of the received sequence  $\widehat{\mathbf{y}}_k$  can be defined as

$$\mathbf{e}_k = \mathbf{y}_k - \widehat{\mathbf{y}}_k \tag{4.2}$$

and

$$\widehat{\mathbf{y}}_k = \widetilde{\mathbf{W}}_k \widetilde{\mathbf{x}}_k \tag{4.3}$$

where  $\widetilde{\mathbf{x}}_k =$

$[x_k^{(1)} \cdots x_{k-(L_e)}^{(1)} x_k^{(2)} \cdots x_{k-(L_e)}^{(2)} \cdots x_{k-(L_e)}^{(N_{\mathcal{T}})}]^T$  is the concatenated vector which is constructed with training symbols and is then passed through the filter of length equal to  $L_e \geq L$  and weight matrix  $\widetilde{\mathbf{W}}_k$  of the estimator of the order  $N_{\mathcal{T}} \times M$ ; where  $M = N_{\mathcal{R}} + N_{\mathcal{R}} L_e$  is the order of estimator and depends on  $L_e$ , the input of the estimator from each receiver antenna which can be written in Section 3.2 of Chapter 3. In case of LoS-only link, the dimension of  $M$  is equal to  $N_{\mathcal{R}}$  here it

by considering the condition of filter length  $L_e = L$ . Substituting the value of  $\widehat{\mathbf{y}}_k$  from Eq. (4.3) and  $\mathbf{y}_k$  from Eq. (4.1) and putting into Eq. (4.2), the error vector can also be written as

$$\mathbf{e}_k = \left\{ \widetilde{\mathbf{H}}_k - \widetilde{\mathbf{W}}_k \right\} \widetilde{\mathbf{x}}_k + \mathbf{v}_k \quad (4.4)$$

$$\mathbf{e}_k = \widetilde{\mathbf{\Xi}}_k \widetilde{\mathbf{x}}_k - \mathbf{v}_k \quad (4.5)$$

where

$$\widetilde{\mathbf{\Xi}}_k = \widetilde{\mathbf{H}}_k - \widetilde{\mathbf{W}}_k \quad (4.6)$$

Correlation function of the innovation process  $\mathbf{e}_k$  can be written as

$$c_k = \mathbb{E} \left\{ \mathbf{e}_k^H \mathbf{e}_k \right\} \quad (4.7)$$

where  $\mathbb{E}\{\cdot\}$  is the expectation operator.

$$c_k = \mathbb{E} \left\{ (\widetilde{\mathbf{\Xi}}_k \widetilde{\mathbf{x}}_k + \mathbf{v}_k)^H (\widetilde{\mathbf{\Xi}}_k \widetilde{\mathbf{x}}_k - \mathbf{v}_k) \right\} \quad (4.8)$$

or

$$c_k = \widetilde{\mathbf{x}}_k^H \widetilde{\mathbf{\Psi}}_k \widetilde{\mathbf{x}}_k + \sigma_{v,k}^2 \quad (4.9)$$

Where,  $\widetilde{\mathbf{\Psi}}_k = \mathbb{E} \left\{ \widetilde{\mathbf{\Xi}}_k^H \widetilde{\mathbf{\Xi}}_k \right\}$  is  $M \times M$  weight-error correlation matrix and is associated with the error correlation of the channel state. To find the estimate of the channel state matrix  $\widetilde{\mathbf{H}}_k$  in terms of  $\widetilde{\mathbf{W}}_k$  through innovation process  $\mathbf{e}_k$ , it is considered that the channel state estimate can be written as a sequence of innovation process,

$$\widetilde{\mathbf{W}}_{k+1} = \sum_{p=1}^{k+1} \mathbf{e}_p \widetilde{\mathbf{b}}_p^H \quad (4.10)$$

where  $\widetilde{\mathbf{b}}_p^H$  is an arbitrary deterministic vector at the timing instant  $q$ . According to the orthogonality principle, the difference of the inverse channel state matrix at time instant,  $k + 1$ , is orthogonal to the innovation process at any time instant,  $q$ ; therefore,

$$\begin{aligned} \mathbb{E} \left\{ \mathbf{e}_q^H \widetilde{\mathbf{\Xi}}_{k+1} \right\} &= 0 \\ q &= 1, 2, 3, \dots, k + 1. \end{aligned} \quad (4.11)$$

$$\mathbb{E}\left\{\mathbf{e}_q^H (\tilde{\mathbf{H}}_{k+1} - \tilde{\mathbf{W}}_{k+1})\right\} = 0 \quad (4.12)$$

Substituting the value of  $\tilde{\mathbf{W}}_{k+1}$  from Eq. (4.10) in Eq. (4.12),

$$\mathbb{E}\left\{\mathbf{e}_q^H (\tilde{\mathbf{H}}_{k+1} - \mathbf{e}_q \tilde{\mathbf{b}}_q^H)\right\} = 0 \quad (4.13)$$

$$\mathbb{E}\left\{\mathbf{e}_q^H \tilde{\mathbf{H}}_{k+1}\right\} - \mathbb{E}\left\{\mathbf{e}_q^H \mathbf{e}_q\right\} \tilde{\mathbf{b}}_q^H = 0 \quad (4.14)$$

Solving Eq. (4.14) for the value of  $\tilde{\mathbf{b}}_q$ ;

$$\tilde{\mathbf{b}}_q = \mathbb{E}\left\{(\tilde{\mathbf{H}}_{k+1})^H \mathbf{e}_q\right\} c_q^{-1} \quad (4.15)$$

Substituting the value of  $\tilde{\mathbf{b}}_q$  into Eq. (4.10);

$$\tilde{\mathbf{W}}_{k+1} = \sum_{p=1}^{k+1} \mathbf{e}_p \mathbb{E}\left\{\mathbf{e}_p^H \tilde{\mathbf{H}}_{k+1}\right\} c_p^{-1} \quad (4.16)$$

$$\tilde{\mathbf{W}}_{k+1} = \sum_{p=1}^k \mathbf{e}_p \mathbb{E}\left\{\mathbf{e}_p^H \tilde{\mathbf{H}}_k\right\} c_p^{-1} + \mathbf{e}_k \mathbb{E}\left\{\mathbf{e}_k^H \tilde{\mathbf{H}}_k\right\} c_k^{-1} \quad (4.17)$$

$$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k + \mathbf{e}_k \tilde{\mathbf{g}}_k^H \quad (4.18)$$

The Kalman gain vector,  $M \times 1$  is

$$\tilde{\mathbf{g}}_k = \mathbb{E}\left\{(\tilde{\mathbf{H}}_k)^H \mathbf{e}_k\right\} c_k^{-1} \quad (4.19)$$

Substituting the value of  $\mathbf{e}_k$  from Eq. (4.5) into Eq. (4.19), Kalman gain, becomes,

$$\tilde{\mathbf{g}}_k = \mathbb{E}\left\{(\tilde{\mathbf{H}}_k)^H \tilde{\mathbf{\Xi}}_k \tilde{\mathbf{x}}_k - (\tilde{\mathbf{H}}_k)^H \mathbf{v}_k\right\} c_k^{-1} \quad (4.20)$$

As  $\tilde{\mathbf{H}}_k$  and  $\mathbf{v}_k$  are uncorrelated so their product is equal to zero. Putting the value of  $\tilde{\mathbf{H}}_k$  from Eq. (4.6) into Eq. (4.20)

$$\tilde{\mathbf{g}}_k = \mathbb{E}\left\{(\tilde{\mathbf{\Xi}}_k + \tilde{\mathbf{W}}_k)^H \tilde{\mathbf{\Xi}}_k \tilde{\mathbf{x}}_k\right\} c_k^{-1} \quad (4.21)$$

Furthermore,  $\tilde{\mathbf{\Xi}}_k$ , the difference of the inverse predicted channel matrix  $\tilde{\mathbf{H}}_k$  and  $\tilde{\mathbf{W}}_k$  is orthogonal to  $\tilde{\mathbf{W}}_k$ , therefore, the Kalman gain matrix is

$$\tilde{\mathbf{g}}_k = \text{E} \left\{ \tilde{\mathbf{\Xi}}_k^H \tilde{\mathbf{\Xi}}_k \right\} \tilde{\mathbf{x}}_k C_k^{-1} \quad (4.22)$$

$$\tilde{\mathbf{g}}_k = \tilde{\mathbf{\Psi}}_k \tilde{\mathbf{x}}_k C_k^{-1} \quad (4.23)$$

To formulate the error covariance matrix  $\tilde{\mathbf{\Psi}}_k$ , rewriting the difference of  $\tilde{\mathbf{H}}_k$  and  $\tilde{\mathbf{W}}_k$  from (Eq. 4.6) for timing instant ' $k + 1$ ',

$$\tilde{\mathbf{\Xi}}_{k+1} = \tilde{\mathbf{H}}_{k+1} - \tilde{\mathbf{W}}_{k+1} \quad (4.24)$$

Substituting the value of  $\tilde{\mathbf{W}}_{k+1}$  from Eq. (4.18) into Eq. (4.24) and considering that the state  $\tilde{\mathbf{H}}_{k+1}$  is equal to  $\tilde{\mathbf{H}}_k$  according to the assumption of block fading,

$$\tilde{\mathbf{\Xi}}_{k+1} = \tilde{\mathbf{H}}_k - \tilde{\mathbf{W}}_k - \mathbf{e}_k \tilde{\mathbf{g}}_k^H \quad (4.25)$$

Rewriting the expression  $\tilde{\mathbf{\Xi}}_{k+1}$  by putting the value of  $\mathbf{e}_k$  from Eq. (4.5), we get

$$\tilde{\mathbf{\Xi}}_{k+1} = \tilde{\mathbf{\Xi}}_k \left( I_{M \times M} - \tilde{\mathbf{x}}_k \tilde{\mathbf{g}}_k^H \right) - \mathbf{v}_k \tilde{\mathbf{g}}_k^H \quad (4.26)$$

The correlation matrix of the difference of channel state can be written as

$$\tilde{\mathbf{\Psi}}_{k+1} = \text{E} \left\{ \tilde{\mathbf{\Xi}}_{k+1}^H \tilde{\mathbf{\Xi}}_{k+1} \right\} \quad (4.27)$$

Substituting the Eq. (4.26) into Eq. (4.27), we get the Riccati difference equation and new  $M \times M$  matrix is described by the recursive computation is

$$\begin{aligned} \tilde{\mathbf{\Psi}}_{k+1} &= \left( I_{M \times M} - \tilde{\mathbf{x}}_k \tilde{\mathbf{g}}_k^H \right)^H \text{E} \left\{ \tilde{\mathbf{\Xi}}_k^H \tilde{\mathbf{\Xi}}_k \right\} \\ &\quad \left( I_{M \times M} - \tilde{\mathbf{x}}_k \tilde{\mathbf{g}}_k^T \right) + \tilde{\mathbf{g}}_k \text{E} \left\{ \mathbf{v}_k^H \mathbf{v}_k \right\} \tilde{\mathbf{g}}_k^H \end{aligned} \quad (4.28)$$

$$\tilde{\mathbf{\Psi}}_{k+1} = \left( I_{M \times M} - \tilde{\mathbf{x}}_k \tilde{\mathbf{g}}_k^H \right)^H \tilde{\mathbf{\Psi}}_k \left( I_{M \times M} - \tilde{\mathbf{x}}_k \tilde{\mathbf{g}}_k^H \right) + \tilde{\mathbf{g}}_k \sigma_{v,k}^2 \tilde{\mathbf{g}}_k^H \quad (4.29)$$

$$\tilde{\mathbf{\Psi}}_{k+1} = \tilde{\mathbf{\Psi}}_k - \tilde{\mathbf{g}}_k \tilde{\mathbf{x}}_k^H \tilde{\mathbf{\Psi}}_k - \tilde{\mathbf{\Psi}}_k \tilde{\mathbf{x}}_k \tilde{\mathbf{g}}_k^H + \tilde{\mathbf{g}}_k \left( \tilde{\mathbf{x}}_k^H \tilde{\mathbf{\Psi}}_k \tilde{\mathbf{x}}_k + \sigma_{v,k}^2 \right) \tilde{\mathbf{g}}_k^H \quad (4.30)$$

$$\tilde{\Psi}_{k+1} = \tilde{\Psi}_k - \tilde{\mathbf{g}}_k \tilde{\mathbf{x}}_k^H \tilde{\Psi}_k - \tilde{\Psi}_k \tilde{\mathbf{x}}_k \tilde{\mathbf{g}}_k^H + \tilde{\mathbf{g}}_k c_k \tilde{\mathbf{g}}_k^H \quad (4.31)$$

multiplying both sides of Eq. (4.23) by  $c_k \tilde{\mathbf{g}}_k^H$ , we get

$$\tilde{\mathbf{g}}_k c_k \tilde{\mathbf{g}}_k^H = \tilde{\Psi}_k \tilde{\mathbf{x}}_k \tilde{\mathbf{g}}_k^H \quad (4.32)$$

substituting the value of  $\tilde{\mathbf{g}}_k c_k \tilde{\mathbf{g}}_k^H$  from Eq. (4.32) into Eq. (4.31); the correlation matrix becomes

$$\tilde{\Psi}_{k+1} = \tilde{\Psi}_k - \tilde{\mathbf{g}}_k \tilde{\mathbf{x}}_k^H \tilde{\Psi}_k \quad (4.33)$$

Eq. (4.33) exhibits the same results shown by RLS and RVFF-RLS with much reduced complexity, however, it does not dependent on the appropriate selection of forgetting factor  $\lambda$ .

### 4.3 Complexity Analysis

The implementation of the proposed algorithm is summarized in Table 4.1. The proposed algorithm requires  $2(N+NL)^2+2N^2(L+1)+2N(L+1)+1$  multiplications and  $2(N+NL)^2+2N^2(L+1)$  additions, where  $N$  represents the MIMO order and  $L$  shows the number of multipath components. On the other hand, the conventional RLS algorithm entails  $3(N+NL)^2+2N^2(L+1)+2N(L+1)+2$  multiplications and  $2(N+NL)^2+2N^2(L+1)$  additions per iteration, whereas, RVFF-RLS algorithm entails  $3(N+NL)^2+2N^2(L+1)+2N(L+1)+14$  multiplications and  $2(N+NL)^2+2N^2(L+1)+8$  additions. The multiplication computational complexity of the proposed algorithm is  $(N+NL)^2$  lesser than RLS and RVFF-RLS algorithms.

TABLE 4.1: The proposed low-complexity MIMO channel estimation algorithm

Initilize: $\tilde{\mathbf{W}}_k, \tilde{\Psi}_k$
$\hat{\mathbf{y}}_k = \tilde{\mathbf{W}}_k \tilde{\mathbf{x}}_k^H$
$\mathbf{e}_k = \mathbf{y}_k - \hat{\mathbf{y}}_k$
$c_k = \tilde{\mathbf{x}}_k^H \tilde{\Psi}_k \tilde{\mathbf{x}}_k + \sigma_{v,k}^2$
$\tilde{\mathbf{g}}_k = \tilde{\mathbf{x}}_k^H \tilde{\Psi}_k c_k^{-1}$
$\tilde{\Psi}_{k+1} = \tilde{\Psi}_k - \tilde{\mathbf{g}}_k \tilde{\mathbf{x}}_k^H \tilde{\Psi}_k$
$\tilde{\mathbf{W}}_{k+1} = \tilde{\mathbf{W}}_k + \mathbf{e}_k \tilde{\mathbf{g}}_k^H$

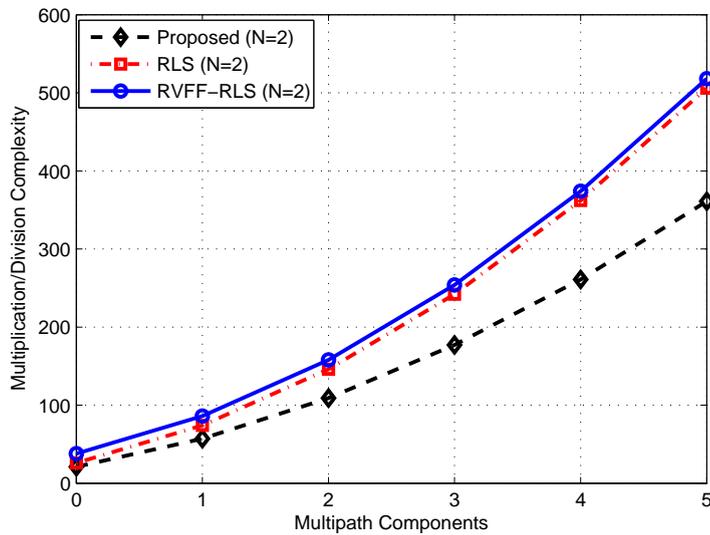


FIGURE 4.1: Per-iteration multiplication complexity versus multipath components for  $2 \times 2$  MIMO communication system

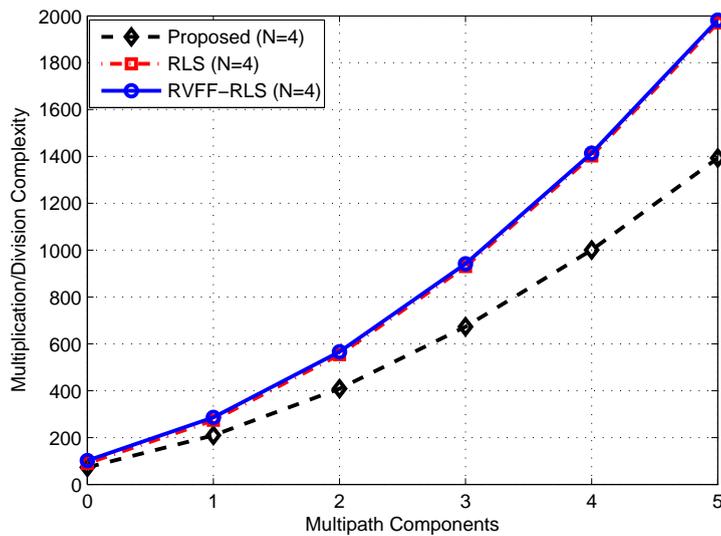


FIGURE 4.2: Per-iteration multiplication complexity versus multipath components for  $4 \times 4$  MIMO communication system

## 4.4 Simulation Results and Discussion

In this section, Monte Carlo simulations are performed on an  $N_T \times N_R$  MIMO communication system by considering two multipath components of BPSK modulated transmitted signal. The time varying channel states are taken from a first order Markovian process presented in Eq. (3.10). The proposed low-complexity MIMO channel estimator is compared with RVFF-RLS and conventional RLS algorithms

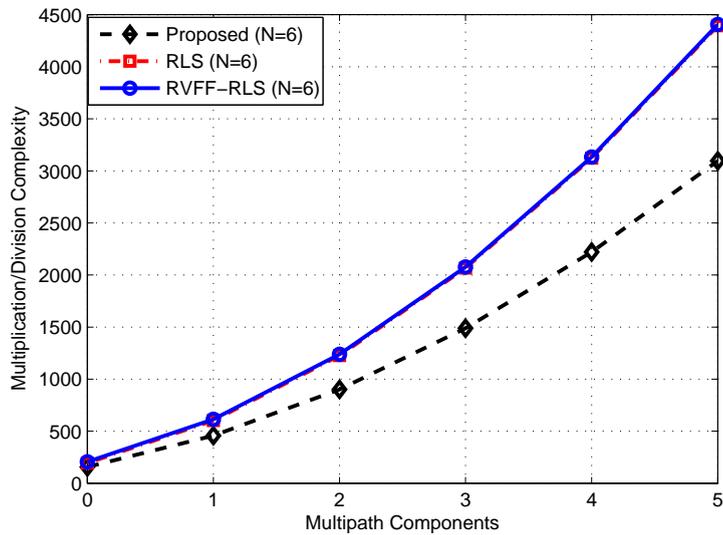


FIGURE 4.3: Per-iteration multiplication complexity versus multipath components for  $6 \times 6$  MIMO communication system

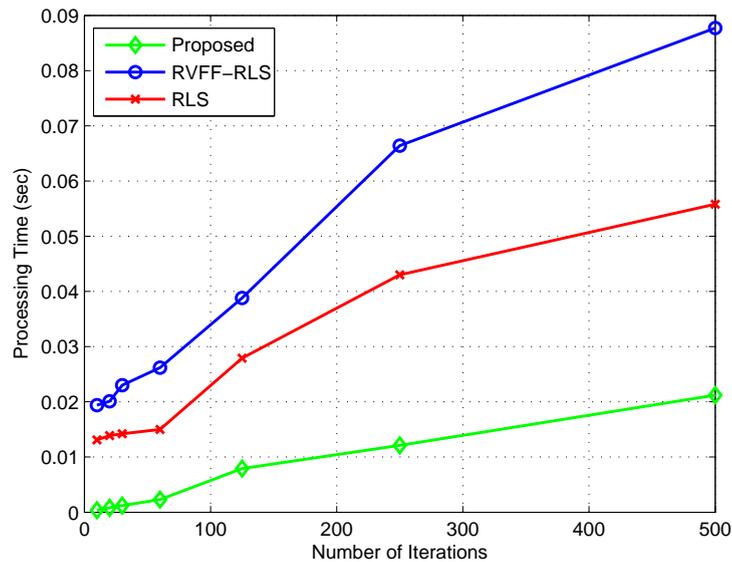


FIGURE 4.4: Processing time versus number of iterations at SNR=20dB for a  $4 \times 4$  MIMO system with LoS-only link

in terms of computational complexity, processing time and mean squared error (MSE). The per-iteration multiplication complexity comparisons of the proposed low-complexity MIMO algorithm with those of RVFF-RLS and conventional RLS algorithms for  $2 \times 2$ ,  $4 \times 4$  and  $6 \times 6$  MIMO communication systems are shown in Fig. 4.1, Fig. 4.2 and Fig. 4.3, respectively. It can be seen that at every number of multipath component, the proposed low-complexity MIMO algorithm provides

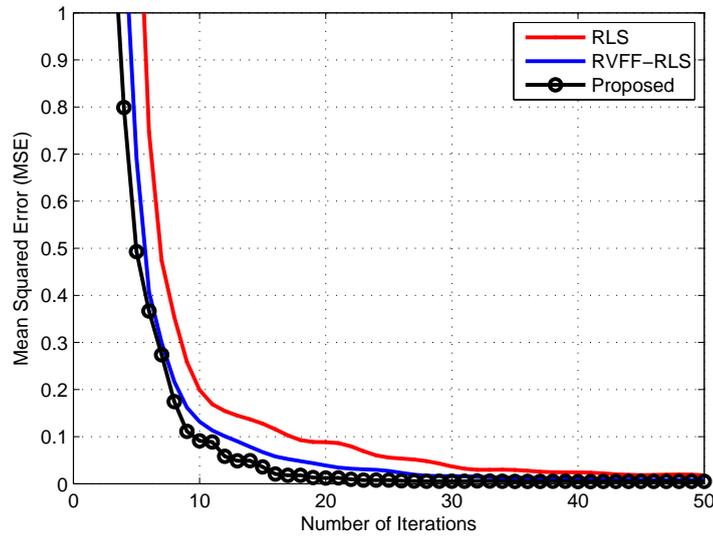


FIGURE 4.5: Mean squared error at SNR=20dB for a  $4 \times 4$  MIMO system with LoS-only link at  $f_D T = 0.01$

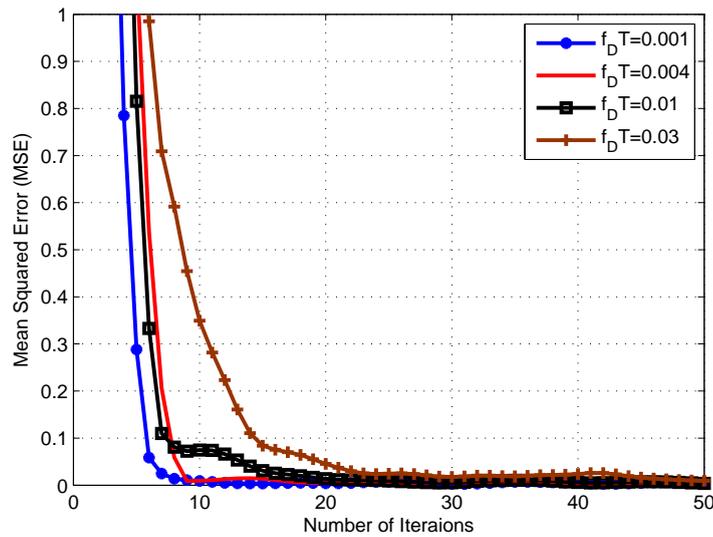


FIGURE 4.6: Comparison of the MSE when the proposed algorithm is applied to various fading conditions at SNR=20dB for a  $4 \times 4$  MIMO system with LoS-only link

lesser complexity than the conventional algorithms. Furthermore, the processing time with respect to number of iterations for a  $4 \times 4$  MIMO communication system at an SNR of 20dB is illustrated in Fig. 4.4. It can be observed that the proposed low-complexity MIMO estimator takes lesser processing time than that taken by both RVFF-RLS and RLS algorithms. Moreover, the MSE plot is presented in Fig. 4.5 where, it can be clearly seen that the proposed low-complexity MIMO

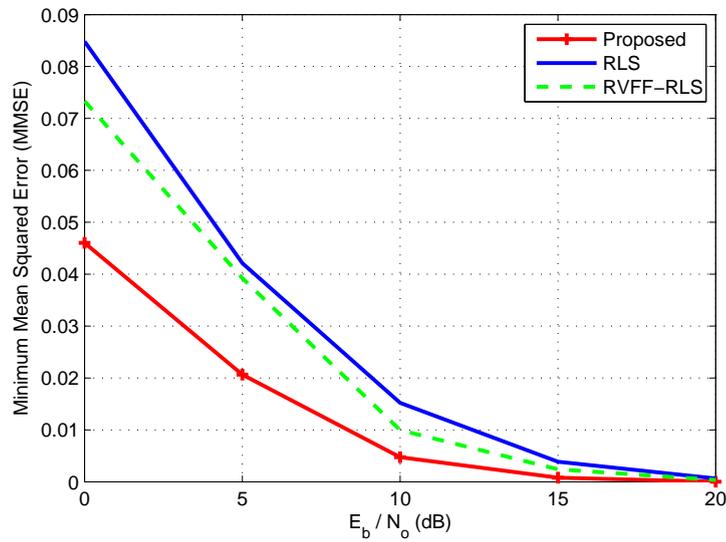


FIGURE 4.7: Behavior of minimum mean squared error (MMSE) against  $E_b/N_0$  for a  $4 \times 4$  MIMO system with LoS-only link when  $f_D T = 0.004$

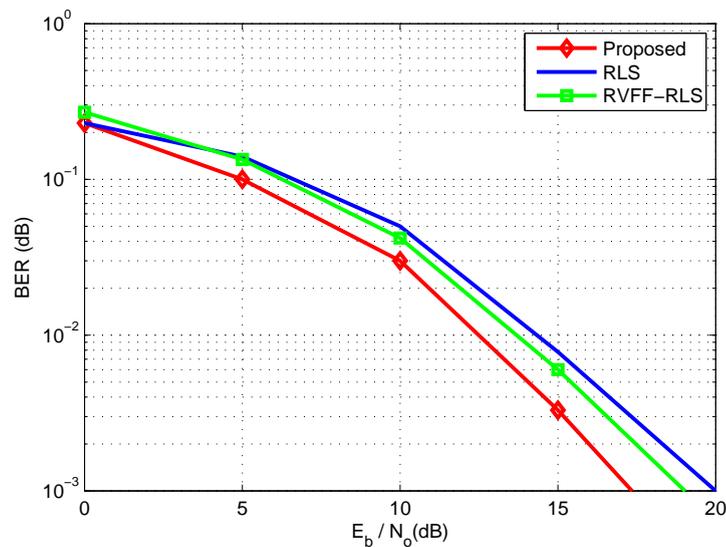


FIGURE 4.8: Probability of bit error rate (BER) against  $E_b/N_0$  for a  $4 \times 4$  MIMO system with LoS-only link when  $f_D T = 0.004$

algorithm outperforms RVFF-RLS and RLS filtering algorithms. It is realized that the proposed algorithm entails 20 iterations for its complete convergence. On the other hand, the conventional RLS and RVFF-RLS algorithms require 30 and 40 iterations, respectively. Likewise, Fig. 4.6 shows the MSE comparison of the proposed algorithm for different Doppler rates. It is obvious from the results that the proposed algorithm converges faster for lower fading rates, and requires more training data for fast fading channels. Furthermore, the minimum mean squared

error for different values of  $E_b/N_o$  is shown in Fig. 4.7. It can be seen that the proposed algorithm gives the smaller MMSE at all values of  $E_b/N_o$  than those of RLS and VFF-RLS filtering algorithms. Moreover, the proposed low-complexity adaptive algorithm can be run on any application like channel equalization. The channel equalizer behaves as the inverse of channel that is used to estimate the transmitted signal. Therefore, the performance of any adaptive filtering algorithm is measured through the probability of error which is the ratio of the number of untrue received data elements to total transmitted data elements. A plot showing the behavior of bit error rate (BER) against different values of  $E_b/N_o$  is presented in Fig. 4.8. It is clearly observed that the proposed low-complexity MIMO algorithm outperforms RVFF-RLS and RLS algorithms.

# Chapter 5

## Parallel Distributed Adaptive RLS Filtering

In this chapter, the working procedure of proposed parallelly-operated Recursive Least Square (RLS) filter in the light of its conventional sequential operation is introduced. Section 5.1 presents the overview of the proposed parallel distributed adaptive signal processing (PDASP) technique. Section 5.2 describes the formulation of proposed PDASP architecture for MIMO communication system. Section 5.3 presents the complexity analysis and finally, Section 5.4 describes the simulation results and discussion.

### 5.1 Overview

In conventional (RLS) adaptive algorithm and its variants, all filter subparts are interdependent on each other and operate sequentially. Before introducing the proposed parallel RLS operation over individual platforms with different clock systems, we define some timing variables with illustration shown in Fig. 5.1, where a single iteration of an RLS algorithm consists of  $N$  sequential blocks.

- **Computational time  $T_c$ :** This is the time taken by the processor for a single computation. It can be calculated simply by the speed of the processor.

- **Block processing time  $T_b$** : It is the processing time of a block of the algorithm. It depends on the number of computations involved in a block. It can thus be a multiple of  $T_c$ .
- **Fetch time  $T_f$** : This is the time in which one block fetches information from another block usually its predecessor.
- **Algorithm step time  $T_s$** : It is the processing time of a complete iteration of the algorithm.

If RLS filtering is operated on a single computationally-capable platform, all algorithm blocks would be executed sequentially as shown in Fig. 5.1 with fetch time  $T_f \rightarrow 0$ . However, if the same RLS filtering is operated on a group of computationally-constrained platforms using the proposed PDASP architecture, different algorithm blocks would be executed parallelly on various individual platforms with varying fetch times depending upon the media among the nodes as shown in Fig. 5.2.

The only possible way to operate the RLS algorithm in parallel fashion on individual platforms with different clock systems is by putting the time as non-aligned. While setting the time non-alignment, two things must be taken into account:

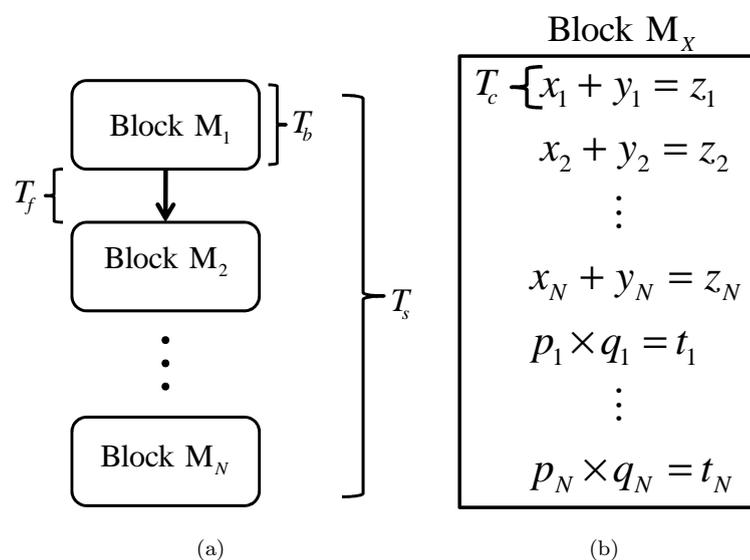


FIGURE 5.1: Sequential working of conventional RLS algorithm (a). sequential working of individual blocks (b). processes involved in a single block

First, it must be seen that the filter does not show the unstable behavior though implementing on any application. Secondly, all the filter sub parts are able to work in parallel manner with favorable fetch times with respect to block processing times. In this way, the sequential structure may be able to work parallelly even with non-aligned time indexes. In Fig. 5.2, the cooperative parallelly-operated RLS filtering architecture consists of four processing nodes, namely  $M_1$ ,  $M_2$ ,  $M_3$  and  $M_4$ . The selection of four nodes is dependent on the parts (e.g. Kalman gain, error covariance matrix, etc) of the adaptive filtering algorithm. Furthermore, it is assumed that the task time allocated to each processing node is sufficient and report the processing outcome to the cluster-head within a predefined time-stamp. The processing nodes  $M_1$  and  $M_4$  are interlinked with  $M_2$  and  $M_3$ , respectively while being connected to themselves also. Likewise,  $M_2$  is interconnected with  $M_1$  and  $M_4$  and  $M_3$  is only linked with  $M_4$ . All the processing nodes would first share information with one another and then would work out the desired process. The processing time of each block differs from one another and is known to all nodes; therefore, all processing nodes which complete their processing tasks earlier than others, wait the processing time equivalent to the block of maximum processing time till the processing of the block with maximum processing time ends. In this

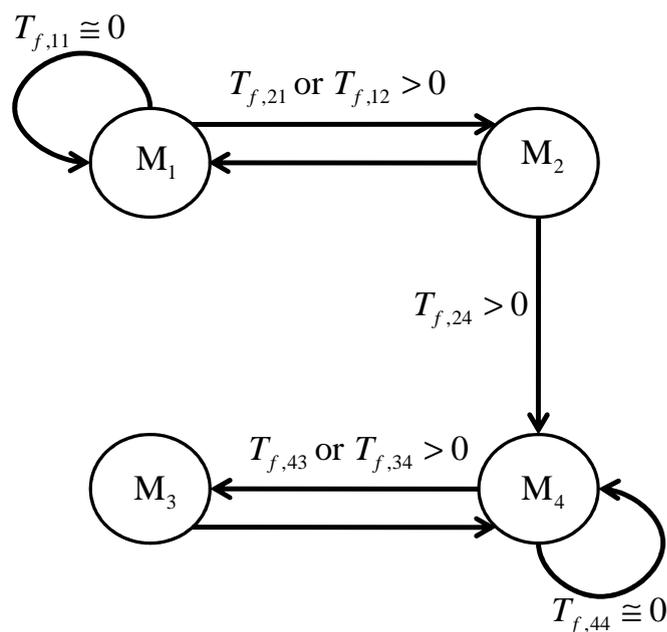


FIGURE 5.2: Proposed cooperative parallelly-operated RLS filtering

way, the inexpensive and computationally-constrained platforms work in parallel for a combined goal.

## 5.2 Proposed PDASP Technique for RLS with Non-aligned Time Indexes

In adaptive filtering, all the filter sub-parts are interdependent on one another. Due to cascaded fashion the algorithm takes mutual processing time while attaining its convergence with respect to uncertain channel conditions. By using the PDASP technique, RLS algorithm runs in parallel manner even with non-aligned time indexes while providing parallelly low processing time at each machine or processing node. The flow diagram of PDASP technique using RLS is shown in Fig. 5.3. The notation “ $T_{\mathbf{X}}$ ” is used to represent the time used in processing of  $\mathbf{X}$  whose computation is done inside the pointed block. Let the processing times taken by error covariance matrix “ $\Psi_k$ ”, Kalman gain “ $\mathbf{g}_k$ ”, received signal estimation  $\hat{\mathbf{y}}_k$ , estimation error “ $\mathbf{e}_k$ ” and update filter coefficient matrix “ $\hat{\mathbf{H}}_k$ ” be  $T_{\Psi}$ ,  $T_{\mathbf{g}}$ ,  $T_{\hat{\mathbf{y}}}$ ,  $T_{\mathbf{e}}$  and  $T_{\hat{\mathbf{H}}}$  respectively. Therefore, the total time taken by the whole algorithm that runs in cascaded fashion is

$$T_{\Psi} + T_{\mathbf{g}} + T_{\hat{\mathbf{y}}} + T_{\mathbf{e}} + T_{\hat{\mathbf{H}}} = T_{\text{tot}} \quad (5.1)$$

The maximum processing time among  $T_{\Psi} \dots T_{\hat{\mathbf{H}}}$  is  $T_{\Psi}$  because  $\Psi_k$  takes more multiplications than any of  $\mathbf{g}_k$ ,  $\hat{\mathbf{y}}_k$ ,  $\mathbf{e}_k$  and  $\hat{\mathbf{H}}_k$ . In order to operate RLS algorithm parallelly while distributing the operation of various blocks on individual nodes with non-aligned time indexes. The strict and sufficient conditions with respect to fast convergence rate in terms of multiplication and addition computations can thus be written as,

$$T_{\mathbf{g}}, T_{\hat{\mathbf{y}}}, T_{\mathbf{e}}, T_{\hat{\mathbf{H}}} \leq T_{\Psi} \quad (5.2)$$

and

$$T_{\Psi} + T_f \ll T_{\text{tot}}. \quad (5.3)$$

Due to non-aligned time indexes, the mismatch  $\zeta$  between the aligned and non aligned time indexes can be written as

$$\zeta = |e_{Seq} - e_{NA}| \quad (5.4)$$

where  $e_{Seq}$  is the error of sequential algorithm and  $e_{NA}$  is the error of the PDASP algorithm with non-aligned time indexes. The pseudocode of the working procedure of the PDASP architecture is presented in Table 5.1. Moreover, the proposed architecture can be run in sequential format for convergence calibration. The sequential implementation of PDASP RLS algorithm with non-aligned time indexes is nearly the same as that of a conventional RLS algorithm run on a single machine. Steps of this sequential format are tabulated in Table 5.2.

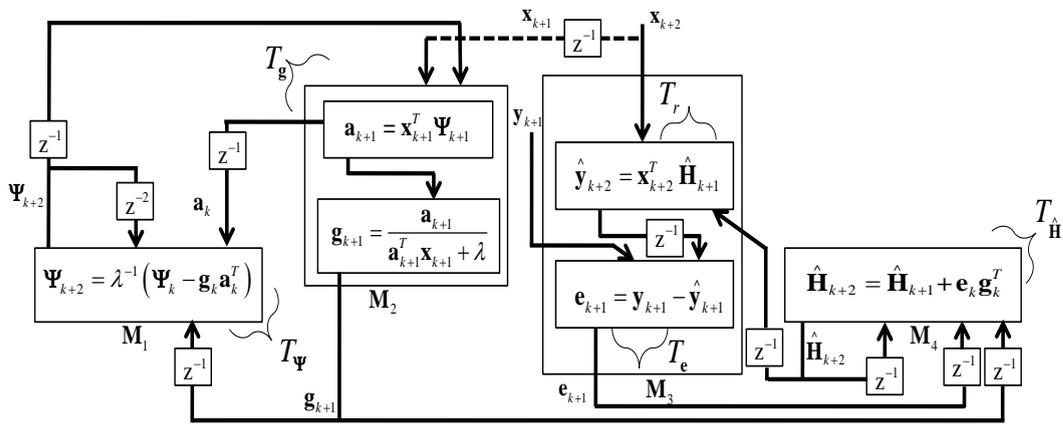


FIGURE 5.3: Proposed PDASP architecture for MIMO RLS algorithm with non-aligned time indexes

TABLE 5.1: Pseudo code: Working procedure of PDASP architecture for MIMO communication system

---

Initilize:  $\lambda, \hat{\mathbf{H}}_{k+1}, \mathbf{a}_k, \hat{\mathbf{y}}_{k+1}, \mathbf{\Psi}_{k+1}, \mathbf{\Psi}_k, \mathbf{e}_k, \mathbf{g}_k$   
for  $k=0:N$

Process running at node  $M_1$

at time  $t_1$ :  $\mathbf{\Psi}_{k+2} = \lambda^{-1} \{ \mathbf{\Psi}_k - \mathbf{g}_k \mathbf{a}_k^T \}$

at time  $t_2$ : Wait until receive  $\mathbf{g}_{k+1}$

at time  $t_3$ : Receive  $\mathbf{g}_{k+1}$  from  $M_2$

at time  $t_4$ : Wait

at time  $t_5$ : Transmit  $\mathbf{\Psi}_{k+2}$  from  $M_1$  to  $M_2$

at time  $t_6$ : Wait until receive  $\mathbf{a}_{k+1}^T$

at time  $t_7$ : Wait until receive  $\mathbf{a}_{k+1}^T$

at time  $t_8$ : Receive  $\mathbf{a}_{k+1}^T$  from  $M_2$

Process running at node  $M_2$

at time  $t_1$ :  $\mathbf{a}_{k+1}^T = \mathbf{x}_{k+1}^T \mathbf{\Psi}_{k+1}$

$$\mathbf{g}_{k+1} = \frac{\mathbf{a}_{k+1}}{\mathbf{a}_{k+1}^T \mathbf{x}_{k+1} + \lambda}$$

at time  $t_2$ : Transmit  $\mathbf{g}_{k+1}$  from  $M_2$  to  $M_1$

at time  $t_3$ : Transmit  $\mathbf{g}_{k+1}$  from  $M_2$  to  $M_4$

at time  $t_4$ : Wait until receive  $\mathbf{\Psi}_{k+2}$

at time  $t_5$ : Wait until receive  $\mathbf{\Psi}_{k+2}$

at time  $t_6$ : Receive  $\mathbf{\Psi}_{k+2}$  from  $M_1$

at time  $t_7$ : Transmit  $\mathbf{a}_{k+1}^T$  from  $M_2$  to  $M_1$

at time  $t_8$ : Ready for next iteration

Process running at node  $M_3$

at time  $t_1$ :  $\hat{\mathbf{y}}_{k+2}^T = \mathbf{x}_{k+2}^T \hat{\mathbf{H}}_{k+1}$

$$\mathbf{e}_{k+1} = \mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}$$

at time  $t_2$ : Wait until receive  $\hat{\mathbf{H}}_{k+2}$

at time  $t_3$ : Wait until receive  $\hat{\mathbf{H}}_{k+2}$

at time  $t_4$ : Wait until receive  $\hat{\mathbf{H}}_{k+2}$

at time  $t_5$ : Wait until receive  $\hat{\mathbf{H}}_{k+2}$

at time  $t_6$ : Receive  $\hat{\mathbf{H}}_{k+2}$  from  $M_4$

at time  $t_7$ : Transmit  $\mathbf{e}_{k+1}$  from  $M_3$  to  $M_4$

at time  $t_8$ : Ready for next iteration

Process running at node  $M_4$

at time  $t_1$ :  $\hat{\mathbf{H}}_{k+2} = \hat{\mathbf{H}}_{k+1} + \mathbf{e}_k \mathbf{g}_k^T$

at time  $t_2$ : Wait until receive  $\mathbf{g}_{k+1}$

at time  $t_3$ : Wait until receive  $\mathbf{g}_{k+1}$

at time  $t_4$ : Receive  $\mathbf{g}_{k+1}$  from  $M_2$

at time  $t_5$ : Transmit  $\hat{\mathbf{H}}_{k+2}$  from  $M_4$  to  $M_3$

at time  $t_6$ : Wait until receive  $\mathbf{e}_{k+1}$

at time  $t_7$ : Wait until receive  $\mathbf{e}_{k+1}$

at time  $t_8$ : Receive  $\mathbf{e}_{k+1}$  from  $M_3$

---

TABLE 5.2: PDASP RLS algorithm with non-aligned time indexes if runs sequentially

---


$$\begin{aligned} & \text{Initilize: } \lambda, \hat{\mathbf{H}}_{k+1}, \mathbf{a}_k, \hat{\mathbf{y}}_{k+1}, \mathbf{\Psi}_{k+1}, \mathbf{\Psi}_k, \mathbf{e}_k, \mathbf{g}_k \\ & \hat{\mathbf{y}}_{k+2}^T = \mathbf{x}_{k+2}^T \hat{\mathbf{H}}_{k+1} \\ & \mathbf{e}_{k+1} = \mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1} \\ & \mathbf{a}_{k+1}^T = \mathbf{x}_{k+1}^T \mathbf{\Psi}_{k+1} \\ & \mathbf{g}_{k+1} = \frac{\mathbf{a}_{k+1}}{\mathbf{a}_{k+1}^T \mathbf{x}_{k+1} + \lambda} \\ & \mathbf{\Psi}_{k+2} = \lambda^{-1} \{ \mathbf{\Psi}_k - \mathbf{g}_k \mathbf{a}_k^T \} \\ & \hat{\mathbf{H}}_{k+2} = \hat{\mathbf{H}}_{k+1} + \mathbf{e}_k \mathbf{g}_k^T \end{aligned}$$


---

### 5.3 Complexity Analysis

The complexity of the linear Kalman filter requires  $2(N + NL)^3 + 6N^2(L + 1) + 3N(L + 1) + 1$  multiplications and  $3(N + NL)^3 + 4N^2(L + 1) + 2$  additions per iteration; where  $N$  represents the MIMO order and  $L$  shows the number of multi-path components. Likewise, RLS algorithm that is the special case of Kalman filter entails  $3(N + NL)^2 + 2N^2(L + 1) + 2N(L + 1) + 2$  multiplications and  $2(N + NL)^2 + 2N^2(L + 1)$  additions per iteration; The implementation of the proposed PDASP technique on RLS algorithm exhibits much lesser computational cost for each parallelly-distributed entity block. The proposed PDASP technique with non-aligned time indexes entails parallelly  $2(N + NL)^2$  multiplications and  $(N + NL)^2$  additions per iteration at maximum. The proposed parallel technique thus provides much lesser processing time than that of sequential Kalman and RLS algorithms.

### 5.4 Simulation Results and Discussion

In this section, Monte Carlo simulations with binary phase shift keying (BPSK) are performed on  $4 \times 4$  MIMO communication system to substantiate the validation

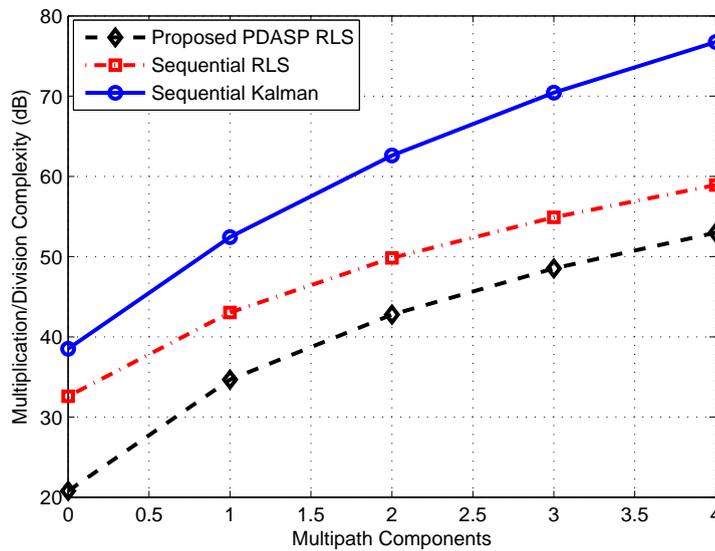


FIGURE 5.4: Per-iteration multiplication complexity comparison among  $2 \times 2$  MIMO sequential algorithms and proposed PDASP technique

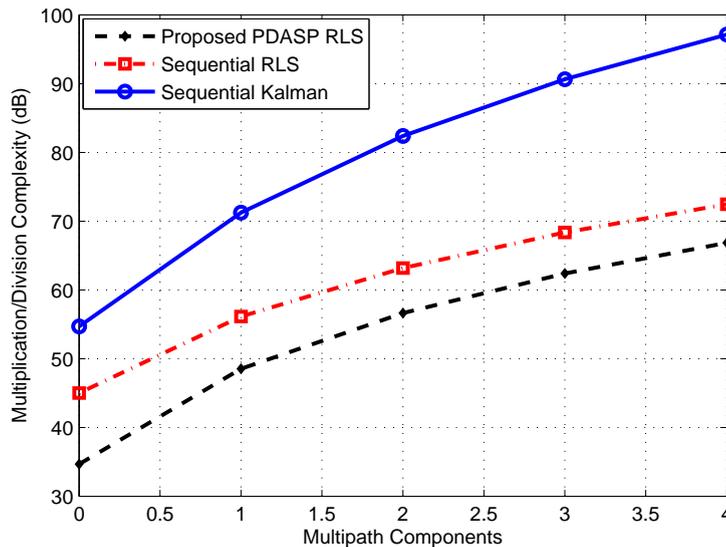


FIGURE 5.5: Per-iteration multiplication complexity comparison among  $4 \times 4$  MIMO sequential algorithms and proposed PDASP technique

of our proposed PDASP architecture. The forgetting factor  $\lambda$  is set to be 0.98 for both the proposed PDASP and sequential RLS algorithms.

The proposed parallel technique that is implemented on MIMO RLS is then compared with the sequential MIMO RLS adaptive algorithm and Kalman filter in terms of computational complexity, mean square error (MSE) and processing time with non-aligned time indexes. The implementation of the proposed PDASP

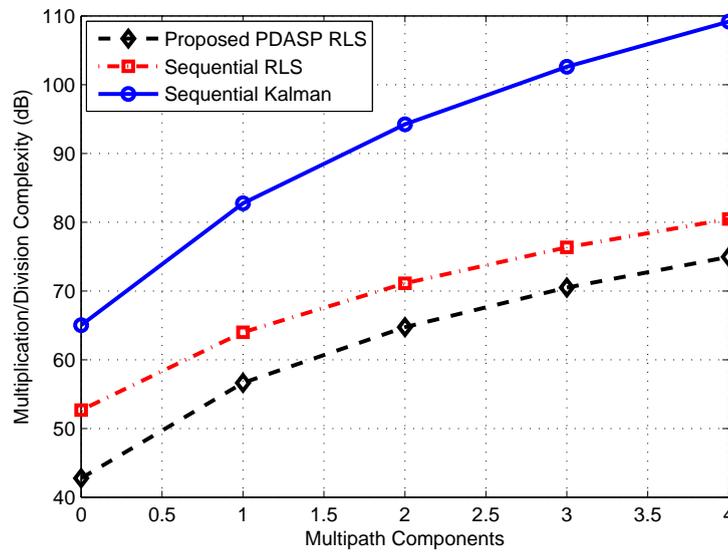


FIGURE 5.6: Per-iteration multiplication complexity comparison among  $6 \times 6$  MIMO sequential algorithms and proposed PDASP technique

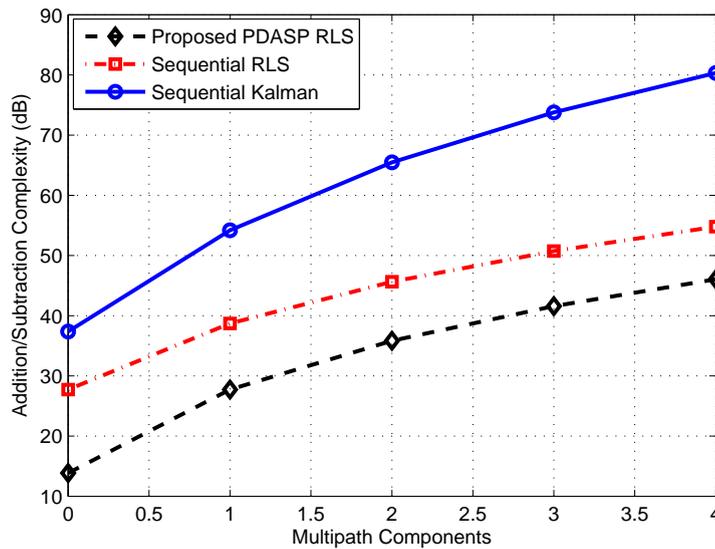


FIGURE 5.7: Per-iteration addition complexity comparison among  $2 \times 2$  MIMO sequential algorithms and proposed PDASP technique

scheme is done using MIMO RLS algorithm and its performance in terms computational complexity is then compared with a sequentially-operated non-distributed Kalman and RLS adaptive filtering algorithms. The parallel technique provides much lesser computational complexity parallelly than the sequential Kalman and MIMO RLS algorithms. The per-iteration multiplication complexity comparison of proposed PDASP technique than those of sequentially-operated Kalman and

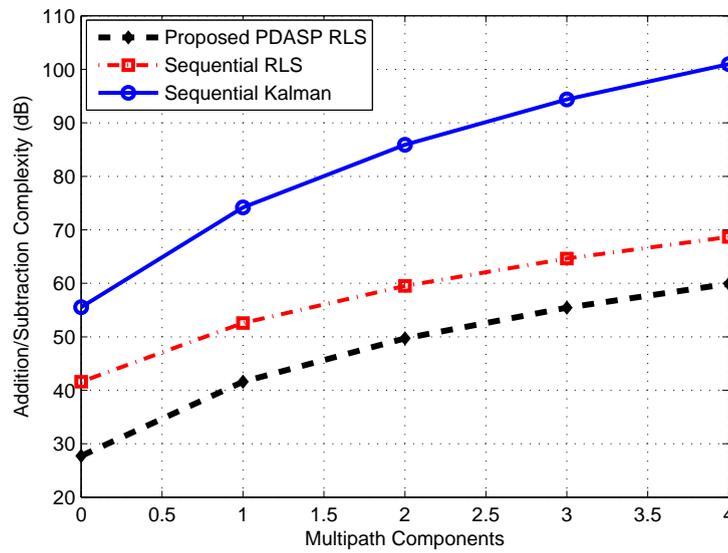


FIGURE 5.8: Per-iteration addition complexity comparison among  $4 \times 4$  MIMO sequential algorithms and proposed PDASP technique

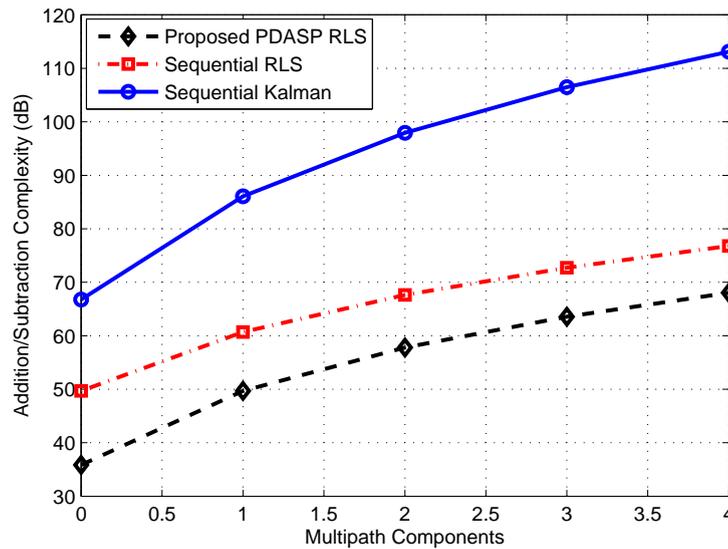


FIGURE 5.9: Per-iteration addition complexity comparison among  $6 \times 6$  MIMO sequential algorithms and proposed PDASP technique

MIMO RLS algorithms for  $2 \times 2$ ,  $4 \times 4$  and  $6 \times 6$  MIMO communication systems are shown in Fig. 5.4, Fig. 5.5 and Fig. 5.6, respectively. Likewise, the per-iteration addition complexity comparison of proposed PDASP technique than those of sequentially-operated Kalman and MIMO RLS algorithms for  $2 \times 2$ ,  $4 \times 4$  and  $6 \times 6$  MIMO communication systems are shown in Fig. 5.7, Fig. 5.8 and Fig. 5.9,

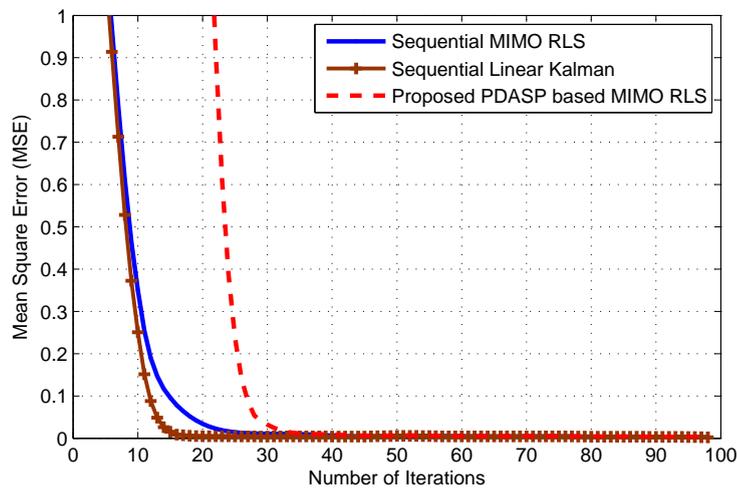


FIGURE 5.10: Mean square error (MSE) tracking performance versus training length for  $4 \times 4$  MIMO when  $f_D T = 10^{-6}$

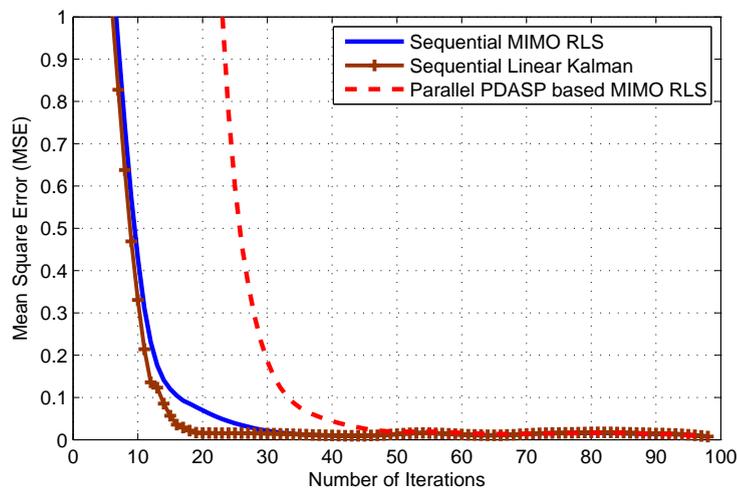


FIGURE 5.11: Mean square error (MSE) tracking performance versus training length for  $4 \times 4$  MIMO when  $f_D T = 10^{-3}$

respectively. It is observed that at every number of multipath component, the proposed PDASP technique using non-aligned time indexes provides parallelly much lesser multiplication and addition complexity than sequential Kalman and MIMO RLS algorithms. Furthermore, Fig. 5.10 and Fig. 5.11 show the MSE performance at low doppler rate  $f_D T = 10^{-6}$  and high doppler rate  $f_D T = 10^{-3}$ , respectively and Fig. 5.12 shows their MSE difference among the proposed PDASP MIMO RLS and sequentially-operated algorithms. It is realized that the difference in convergence performance of proposed PDASP scheme run with non-aligned time indexes and that of the sequential Kalman and MIMO RLS algorithms is only

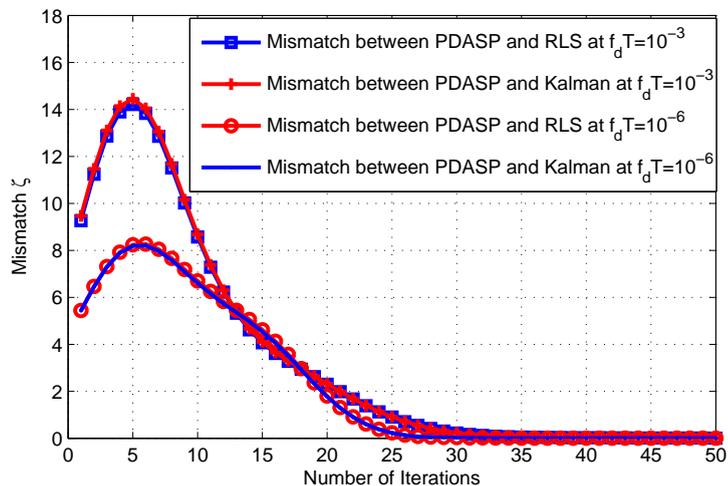


FIGURE 5.12: Mean square error difference among sequential algorithms and proposed PDASP scheme

of 20 and 10 iterations, respectively at low doppler spread and about 30 and 20 iterations with relatively high doppler spread, respectively. Considering the difference in Fig. 5.12, it can be seen that, due to initialization of the algorithm parameters of PDASP technique, the error difference that is small at the start, gradually increases, then reverses to decrease and eventually becomes zero on a specific number of iterations. The fast Ethernet speed of  $125\text{Mbits/s}$  is taken as the reference peak bit rate in the wired communication. In  $4 \times 4$  MIMO PDASP scheme, the maximum size of  $4 \times 4$  matrix is to be transmitted from one machine node to another through wired communication.

However, each entry in  $4 \times 4$  MIMO matrix is consisted on 4 bytes, in which two significant bytes are before the decimal point and two significant bytes are taken after the decimal point. The total number of bits for  $4 \times 4$  matrix are  $4 \times 16 \times 8 = 512$  bits. The fetch time  $T_f$  according to this number of bits is  $41\mu\text{s}$ . Therefore, the processing time comparison and the processing time difference at  $T_f = 41\mu\text{s}$  among the sequential algorithms and the proposed PDASP technique are presented in Fig. 5.13 and Fig. 5.14, respectively. It is clear that the proposed PDASP technique provides much lesser processing time than the sequentially-operated Kalman filter and MIMO RLS algorithm. The percentage improvement in decreased processing time is shown in Table. 5.3 and Table. 5.4. At low doppler

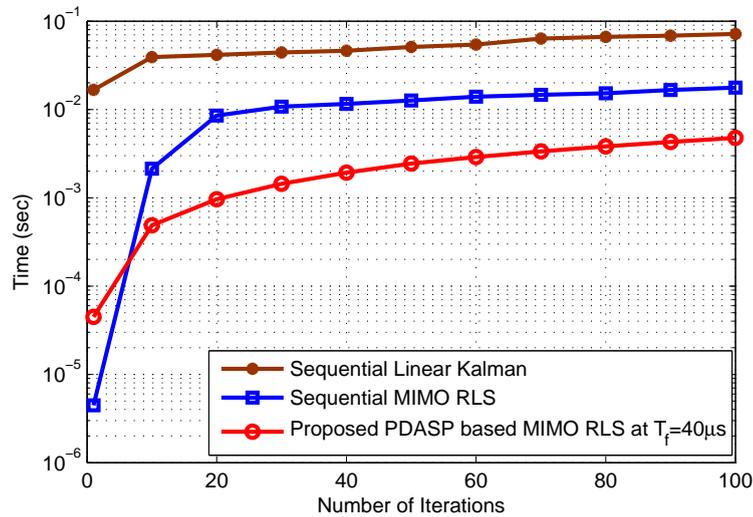


FIGURE 5.13: Processing time comparison among sequential algorithms and proposed PDASP technique

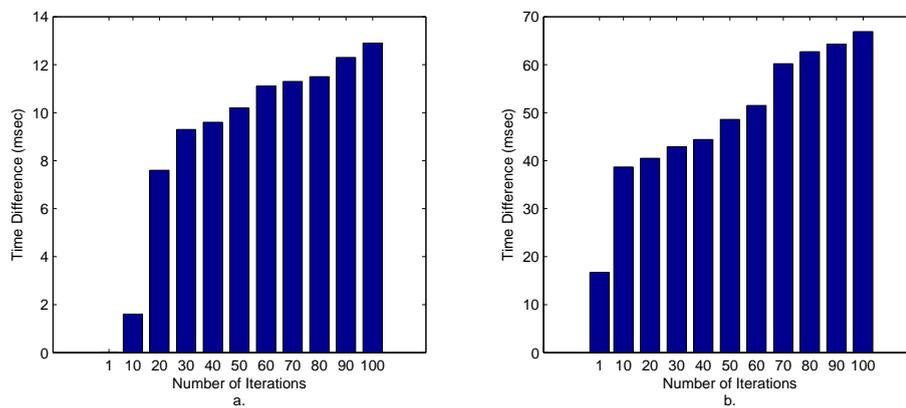


FIGURE 5.14: Processing time difference for  $4 \times 4$  MIMO system (a). sequential RLS vs proposed PDASP technique (b). sequential linear Kalman vs proposed PDASP technique

rate, it is realized that the proposed PDASP MIMO RLS algorithm converges about 35 iterations with the addition of  $T_f = 41\mu s$  at each iteration but still utilizes 95.83% and 82.29% lesser processing time than the sequential Kalman and MIMO RLS algorithms, respectively. Likewise, for high doppler rate, the proposed PDASP MIMO RLS takes 50 iterations for its convergence with the increase of 30 and 20 iterations than the sequential Kalman and MIMO RLS algorithms, respectively. It can be seen that, the proposed technique still entails 94.12% and 72.28% lesser processing time than the sequentially-operated Kalman and MIMO RLS algorithms, respectively.

TABLE 5.3: Percentage improvement in decreased processing of PDASP MIMO RLS with respect to sequential MIMO RLS

Doppler Rate	MIMO RLS		Proposed PDASP MIMO RLS		Improvement in Terms of Decreased Processing Time $\frac{T_{RLS}-T_{PDASP}}{T_{RLS}} \times 100$
	Convergence (iteration)	Processing Time $T_{RLS}$ (sec)	Convergence (iteration)	Processing Time $T_{PDASP}$ (sec) with $T_f = 41\mu sec$	
$f_D T = 10^{-6}$	25	0.0096373	35	0.0016825	82.29%
$f_D T = 10^{-3}$	30	0.0107447	50	0.0024410	77.28%

TABLE 5.4: Percentage improvement in decreased processing of PDASP with respect to sequential linear Kalman

Doppler Rate	Kalman Filter		Proposed PDASP MIMO RLS		Improvement in Terms of Decreased Processing Time $\frac{T_{Kalman}-T_{PDASP}}{T_{Kalman}} \times 100$
	Convergence (iteration)	Processing Time $T_{Kalman}$ (sec)	Convergence (iteration)	Processing Time $T_{PDASP}$ (sec) with $T_f = 41\mu sec$	
$f_D T = 10^{-6}$	15	0.0403	35	0.0016825	95.83%
$f_D T = 10^{-3}$	20	0.0415	50	0.0024410	94.12%

# Chapter 6

## Parallel Distributed Diffusion based Adaptive RLS Filtering

In this chapter, the working procedure of proposed distributed diffusion based Recursive Least Square (RLS) filtering for computationally-constrained platforms is introduced. Section 6.1 presents the system model for diffusion based distributed adaptive signal processing and discusses the number of nodes to be used in the requirement for the MIMO channel estimation. Section 6.2 describes the formulation of distributed diffusion based adaptive signal processing (DDASP) architecture for MIMO system with line of sight link. Likewise, Section 6.3 describes the proposed DDASP architecture for MIMO system with diffused components. Section 6.4 presents the complexity analysis and finally, Section 6.5 describes the results and presents the comparison of its computational complexity with those of PDASP architecture and the existing DDASP systems.

## 6.1 System Model for Diffusion based Distributed Adaptive Signal Processing

In the proposed PDASP architecture presented in Chapter 5, only four nodes are used to run recursive least square algorithm parallelly with non-aligned time indexes for MIMO channel estimation. However, in DDASP, the adaptive algorithm is diffused into the desired number of processing devices. Therefore, the number of processing nodes that are used to run the MIMO RLS algorithm on DDASP architecture is dependent upon the number of MIMO antennas as well as on multipath components. In case of no multipath components, the number of slave nodes ( $S_1, S_2 \dots S_N$ ) are equivalent to the number of MIMO spatial streams along with one master node  $M_o$ . The master node is used to operate the other processing or slave nodes in the distributed network. The block diagram of DDASP architecture for  $N \times N$  MIMO communication system is shown in Fig. 6.1.

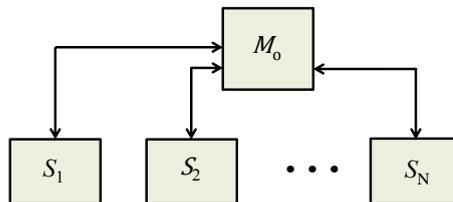


FIGURE 6.1: Block Diagram of DDASP Architecture

In the proposed DDASP architecture, the number of processing nodes  $P_n$  that are required to run any adaptive algorithm for  $N \times N$  MIMO communication system can be found through the following equation

$$P_n = N_T + N_R L + 1 \quad (6.1)$$

where  $N_T$  and  $N_R$  show the number of MIMO transmitting and receiving antennas, respectively, and  $L$  shows the number of diffused components. The number of processing nodes required for  $2 \times 2$ ,  $3 \times 3$  and  $4 \times 4$  MIMO communication systems with line of sight (LoS) and diffused components are shown in Table. 6.1,

TABLE 6.1: Number of processing devices require for various MIMO systems with or without multipath components along with one master node

Multipath Components	2 × 2 MIMO	3 × 3 MIMO	4 × 4 MIMO
No Multipath	3	4	5
One Multipath	5	7	9
Two Multipaths	7	10	13

respectively. Likewise, the dimensions of channel matrix  $\mathbf{W}_k$ , error covariance matrix  $\Phi_k$ , estimation error  $\mathbf{e}_k$ , and Kalman gain  $\mathbf{g}_k$  can be found as

$$\begin{aligned}
\mathbf{W}_k &= N_R \times (N_T + N_R L) \\
\Phi_k &= (N_T + N_R L) \times (N_T + N_R L) \\
\mathbf{e}_k &= 1 \times N_R \\
\mathbf{g}_k &= 1 \times (N_T + N_R L)
\end{aligned} \tag{6.2}$$

In the proposed DDASP architecture, the adaptive algorithm is diffused into the desired number of processing nodes which cooperatively run the adaptive algorithm. Having more nodes and diffusion mechanism, the proposed DDASP architecture must exhibit the following desirable features while implementing a kind of least square adaptive filtering algorithm as compared to PDASP architecture.

- The communication burden taken by DDASP must be lesser than the PDASP architecture
- The maximum computational complexity of DDASP architecture being linear up to some extent is lesser than that of PDASP architecture
- The overall processing time must be reduced than PDASP structure

## 6.2 DDASP Architecture for LoS-only Link

In this section, DDASP architecture with no diffused components is introduced. The DDASP architecture with the implementation of MIMO RLS algorithm is shown in Fig. 6.2. The diffused tasks for the estimated values of the received signals  $\hat{y}_k^{(1)}$  and  $\hat{y}_k^{(2)}$  can be defined as

$$\begin{aligned}\hat{\mathbf{y}}_k &= \mathbf{W}_k \mathbf{x}_k \\ \begin{bmatrix} \hat{y}_k^{(1)} \\ \hat{y}_k^{(2)} \end{bmatrix} &= \begin{bmatrix} w_k^{(11)} & w_k^{(12)} \\ w_k^{(21)} & w_k^{(22)} \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} \\ \hat{y}_k^{(1)} &= w_k^{(11)} x_k^{(1)} + w_k^{(12)} x_k^{(2)} \\ \hat{y}_k^{(2)} &= w_k^{(21)} x_k^{(1)} + w_k^{(22)} x_k^{(2)}\end{aligned}$$

Likewise, the diffused tasks for the components  $a_k^{(1)}$ ,  $a_k^{(2)}$ ,  $b_k^{(1)}$ ,  $b_k^{(2)}$ , and  $c$  that are used to make the formation of Kalman gain  $\mathbf{g}_k$ , can be expressed as

$$\mathbf{g}_k = \frac{\mathbf{x}_k^T \boldsymbol{\Psi}_k}{\mathbf{x}_k^T \boldsymbol{\Psi}_k \mathbf{x}_k + \lambda} \quad (6.3)$$

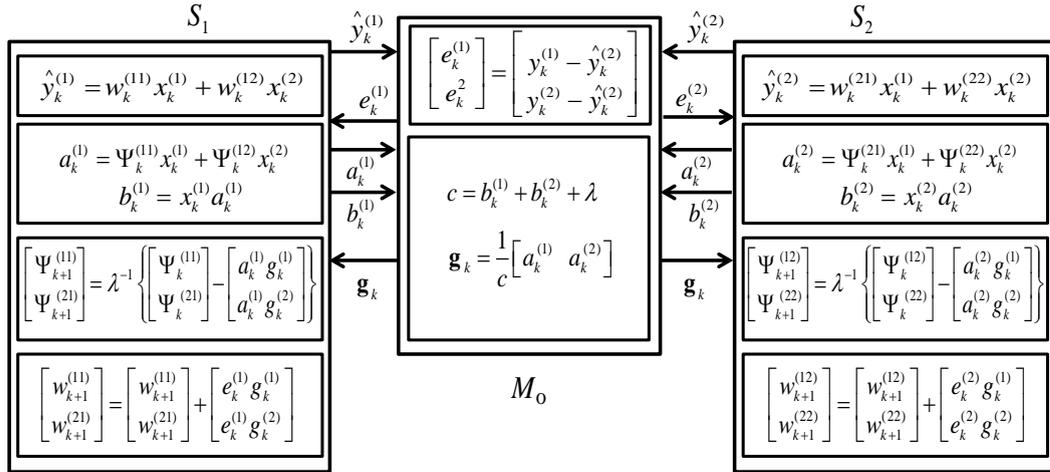


FIGURE 6.2: Proposed DDASP architecture for  $2 \times 2$  MIMO RLS algorithm with LoS-only link

$$\begin{aligned}
\mathbf{g}_k &= \frac{\begin{bmatrix} x_k^{(1)} & x_k^{(2)} \end{bmatrix} \begin{bmatrix} \Psi_k^{(11)} & \Psi_k^{(12)} \\ \Psi_k^{(21)} & \Psi_k^{(22)} \end{bmatrix}}{\begin{bmatrix} x_k^{(1)} & x_k^{(2)} \end{bmatrix} \begin{bmatrix} \Psi_k^{(11)} & \Psi_k^{(12)} \\ \Psi_k^{(21)} & \Psi_k^{(22)} \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} + \lambda} \\
\mathbf{g}_k &= \frac{\begin{bmatrix} a_k^{(1)} & a_k^{(2)} \end{bmatrix}}{\begin{bmatrix} a_k^{(1)} & a_k^{(2)} \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} + \lambda} = \frac{\begin{bmatrix} a_k^{(1)} & a_k^{(2)} \end{bmatrix}}{b_k^{(1)} + b_k^{(2)} + \lambda} \\
a_k^{(1)} &= \psi_k^{(11)} x_k^{(1)} + \psi_k^{(12)} x_k^{(2)} \\
a_k^{(2)} &= \psi_k^{(21)} x_k^{(1)} + \psi_k^{(22)} x_k^{(2)} \\
b_k^{(1)} &= x_k^{(1)} a_k^{(1)} \\
b_k^{(2)} &= x_k^{(2)} a_k^{(2)} \\
c &= b_k^{(1)} + b_k^{(2)} + \lambda
\end{aligned}$$

Similarly, the diffusion procedure of covariance matrix  $\mathbf{\Psi}_k$  and channel matrix  $\mathbf{W}_k$  for DDASP architecture can be defined as

$$\begin{aligned}
\mathbf{\Psi}_{k+1} &= \lambda^{-1} \left[ \mathbf{\Psi}_k - \mathbf{g}_k \mathbf{x}_k^T \mathbf{\Psi}_k \right] \\
\mathbf{\Psi}_{k+1} &= \lambda^{-1} \left\{ \begin{bmatrix} \Psi_k^{(11)} & \Psi_k^{(12)} \\ \Psi_k^{(21)} & \Psi_k^{(22)} \end{bmatrix} - \begin{bmatrix} g_k^{(1)} \\ g_k^{(2)} \end{bmatrix} \begin{bmatrix} x_k^{(1)} & x_k^{(2)} \end{bmatrix} \begin{bmatrix} \Psi_k^{(11)} & \Psi_k^{(12)} \\ \Psi_k^{(21)} & \Psi_k^{(22)} \end{bmatrix} \right\} \\
\mathbf{\Psi}_{k+1} &= \lambda^{-1} \left\{ \begin{bmatrix} \Psi_k^{(11)} & \Psi_k^{(12)} \\ \Psi_k^{(21)} & \Psi_k^{(22)} \end{bmatrix} - \begin{bmatrix} g_k^{(1)} \\ g_k^{(2)} \end{bmatrix} \begin{bmatrix} a_k^{(1)} & a_k^{(2)} \end{bmatrix} \right\} \\
\begin{bmatrix} \Psi_{k+1}^{(11)} \\ \Psi_{k+1}^{(21)} \end{bmatrix} &= \lambda^{-1} \left\{ \begin{bmatrix} \Psi_k^{(11)} \\ \Psi_k^{(21)} \end{bmatrix} - \begin{bmatrix} a_k^{(1)} g_k^{(1)} \\ a_k^{(1)} g_k^{(2)} \end{bmatrix} \right\} \\
\begin{bmatrix} \Psi_{k+1}^{(12)} \\ \Psi_{k+1}^{(22)} \end{bmatrix} &= \lambda^{-1} \left\{ \begin{bmatrix} \Psi_k^{(12)} \\ \Psi_k^{(22)} \end{bmatrix} - \begin{bmatrix} a_k^{(2)} g_k^{(1)} \\ a_k^{(2)} g_k^{(2)} \end{bmatrix} \right\}
\end{aligned}$$

$$\begin{aligned} \mathbf{W}_{k+1} &= \mathbf{W}_k + \mathbf{g}_k \mathbf{e}_k^T \\ \begin{bmatrix} w_{k+1}^{(11)} & w_{k+1}^{(12)} \\ w_{k+1}^{(21)} & w_{k+1}^{(22)} \end{bmatrix} &= \begin{bmatrix} w_k^{(11)} & w_k^{(12)} \\ w_k^{(21)} & w_k^{(22)} \end{bmatrix} + \begin{bmatrix} g_k^{(1)} \\ g_k^{(2)} \end{bmatrix} \begin{bmatrix} e_k^{(1)} & e_k^{(2)} \end{bmatrix} \\ \begin{bmatrix} w_{k+1}^{(11)} \\ w_{k+1}^{(21)} \end{bmatrix} &= \begin{bmatrix} w_k^{(11)} \\ w_k^{(21)} \end{bmatrix} + \begin{bmatrix} e_k^{(1)} g_k^{(1)} \\ e_k^{(1)} g_k^{(2)} \end{bmatrix}, \quad \begin{bmatrix} w_{k+1}^{(12)} \\ w_{k+1}^{(22)} \end{bmatrix} = \begin{bmatrix} w_k^{(12)} \\ w_k^{(22)} \end{bmatrix} + \begin{bmatrix} e_k^{(2)} g_k^{(1)} \\ e_k^{(2)} g_k^{(2)} \end{bmatrix} \end{aligned}$$

Let the processing time taken by any slave node  $S_1$  or  $S_2$  and master node  $M_o$  being  $T_S$  and  $T_{M_o}$ , respectively, can be expressed as

$$\begin{aligned} T_S &= T_{\hat{y}} + T_a + T_b + T_{p_{i,j}, p_{i,j}} + T_{w_{i,j}, w_{i,j}} \\ T_{M_o} &= T_{e_k} + T_c + T_{\mathbf{g}_k} \end{aligned} \quad (6.4)$$

where, processing time  $T$  is either subscripted by the respective variable or by  $i$  and  $j$  that show the index number. It is pertinent to note that the processing times of all the slave nodes are the same in our current case, i.e.  $T_{S_1} = T_{S_2} = T_S$ ; therefore, the strict and sufficient condition with respect to processing time of DDASP and PDASP architectures can thus be written as

$$T_S + T_{M_o} + T_{f,(S_{1,2} \leftrightarrow M_o)} = T_{DDASP} \ll T_{PDASP} \quad (6.5)$$

where  $T_{DDASP}$  is the total time taken by DDASP with no diffused component for one iteration and  $T_{f,S \leftrightarrow M_o}$  is the total fetch time for the communication of data elements among the master and slave nodes. The pseudocode of the working procedure of master and slave nodes in the DDASP architecture with no diffused components is presented in Table. 6.2.

TABLE 6.2: Pseudo code: Working procedure of DDASP for  $2 \times 2$  MIMO communication system

---

Initilize: $\Psi_k^{(11)}, \Psi_k^{(12)}, \Psi_k^{(21)}, \Psi_k^{(22)}, w_k^{(11)}, w_k^{(12)}, w_k^{(21)}, w_k^{(22)}$
for k=0:N
Process running at node $M_1$
at time $t_1$ : $\hat{y}_k^{(1)} = w_k^{(11)} x_k^{(1)} + w_k^{(12)} x_k^{(2)}$
at time $t_2$ : Transmit $\hat{y}_k^{(1)}$ from $S_1$ to $M_o$

---

- 
- at time  $t_3$ : Wait until receive  $e_k^{(1)}$   
 at time  $t_4$ : Wait until receive  $e_k^{(1)}$   
 at time  $t_5$ : Wait until receive  $e_k^{(1)}$   
 at time  $t_6$ : Receive  $e_k^{(1)}$  from  $M_o$   
 at time  $t_7$ :  $a_k^{(1)} = \Psi_k^{(11)} x_k^{(1)} + \Psi_k^{(12)} x_k^{(2)}$   
 $b_k^{(1)} = x_k^{(1)} a_k^{(1)}$   
 at time  $t_8$ : Transmit  $a_k^{(1)}$  and  $b_k^{(1)}$  from  $S_1$  to  $M_o$   
 at time  $t_9$ : Wait until receive  $\mathbf{g}_k$   
 at time  $t_{10}$ : Wait until receive  $\mathbf{g}_k$   
 at time  $t_{11}$ : Wait until receive  $\mathbf{g}_k$   
 at time  $t_{12}$ : Receive  $\mathbf{g}_k$  from  $M_o$   
 at time  $t_{13}$ : 
$$\begin{bmatrix} \Psi_{k+1}^{(11)} \\ \Psi_{k+1}^{(21)} \end{bmatrix} = \lambda^{-1} \left\{ \begin{bmatrix} \Psi_k^{(11)} \\ \Psi_k^{(21)} \end{bmatrix} - \begin{bmatrix} a_k^{(1)} g_k^{(1)} \\ a_k^{(1)} g_k^{(2)} \end{bmatrix} \right\}$$
  
 at time  $t_{14}$ : 
$$\begin{bmatrix} w_{k+1}^{(11)} \\ w_{k+1}^{(21)} \end{bmatrix} = \begin{bmatrix} w_k^{(11)} \\ w_k^{(21)} \end{bmatrix} - \begin{bmatrix} e_k^{(1)} g_k^{(1)} \\ e_k^{(1)} g_k^{(2)} \end{bmatrix}$$

Process running at node  $M_o$

- at time  $t_1$ : Wait until receive  $\hat{y}_k^{(1)}$  and  $\hat{y}_k^{(2)}$   
 at time  $t_2$ : Wait until receive  $\hat{y}_k^{(1)}$  and  $\hat{y}_k^{(2)}$   
 at time  $t_3$ : Receive  $\hat{y}_k^{(1)}$  and  $\hat{y}_k^{(2)}$  from  $S_1$  and  $S_2$   
 at time  $t_4$ : 
$$\begin{bmatrix} e_k^{(1)} \\ e_k^{(2)} \end{bmatrix} = \begin{bmatrix} y_k^{(1)} - \hat{y}_k^{(1)} \\ y_k^{(2)} - \hat{y}_k^{(1)} \end{bmatrix}$$
  
 at time  $t_5$ : Transmit  $e_k^{(1)}$  and  $e_k^{(2)}$  towards  $S_1$  and  $S_2$ , respectively  
 at time  $t_6$ : Wait until receive  $a_k^{(1)}$ ,  $a_k^{(2)}$ ,  $b_k^{(1)}$ ,  $b_k^{(2)}$   
 at time  $t_7$ : Wait until receive  $a_k^{(1)}$ ,  $a_k^{(2)}$ ,  $b_k^{(1)}$ ,  $b_k^{(2)}$   
 at time  $t_8$ : Wait until receive  $a_k^{(1)}$ ,  $a_k^{(2)}$ ,  $b_k^{(1)}$ ,  $b_k^{(2)}$   
 at time  $t_9$ : Receive  $a_k^{(1)}$ ,  $b_k^{(1)}$  and  $a_k^{(2)}$ ,  $b_k^{(2)}$  from  $S_1$  and  $S_2$ , respectively  
 at time  $t_{10}$ :  $c = b_k^{(1)} + b_k^{(2)} + \lambda$   

$$\mathbf{g}_k = \frac{1}{c} \begin{bmatrix} a_k^{(1)} & a_k^{(2)} \end{bmatrix}$$
  
 at time  $t_{11}$ : Transmit  $\mathbf{g}_k$  from  $M_o$  towards  $S_1$  and  $S_2$   
 at time  $t_{12}$ : Ready for next iteration  
 at time  $t_{13}$ : Ready for next iteration
-

---

Process running at node $S_2$
at time $t_1$ : $\hat{y}_k^{(2)} = w_k^{(21)}x_k^{(1)} + w_k^{(22)}x_k^{(2)}$
at time $t_2$ : Transmit $\hat{y}_k^{(2)}$ from $S_2$ to $M_o$
at time $t_3$ : Wait until receive $e_k^{(2)}$
at time $t_4$ : Wait until receive $e_k^{(2)}$
at time $t_5$ : Wait until receive $e_k^{(2)}$
at time $t_6$ : Receive $e_k^{(2)}$ from $M_o$
at time $t_7$ : $a_k^{(2)} = \Psi_k^{(21)}x_k^{(1)} + \Psi_k^{(22)}x_k^{(2)}$ $b_k^{(2)} = x_k^{(2)}a_k^{(2)}$
at time $t_8$ : Transmit $a_k^{(2)}$ and $b_k^{(2)}$ from $S_2$ to $M_o$
at time $t_9$ : Wait until receive $\mathbf{g}_k$
at time $t_{10}$ : Wait until receive $\mathbf{g}_k$
at time $t_{11}$ : Wait until receive $\mathbf{g}_k$
at time $t_{12}$ : Receive $\mathbf{g}_k$ from $M_o$
at time $t_{13}$ : $\begin{bmatrix} \Psi_{k+1}^{(12)} \\ \Psi_{k+1}^{(22)} \end{bmatrix} = \lambda^{-1} \left\{ \begin{bmatrix} \Psi_k^{(12)} \\ \Psi_k^{(22)} \end{bmatrix} - \begin{bmatrix} a_k^{(2)} g_k^{(1)} \\ a_k^{(2)} g_k^{(2)} \end{bmatrix} \right\}$
at time $t_{14}$ : $\begin{bmatrix} w_{k+1}^{(12)} \\ w_{k+1}^{(22)} \end{bmatrix} = \begin{bmatrix} w_k^{(12)} \\ w_k^{(22)} \end{bmatrix} - \begin{bmatrix} e_k^{(2)} g_k^{(1)} \\ e_k^{(2)} g_k^{(2)} \end{bmatrix}$

---

### 6.3 DDASP Architecture for Multipath based Link

The flow diagram of DDASP for  $2 \times 2$  MIMO system with one diffused component other than the LoS link is shown in Fig. 6.3, where, the vector notations used in the DDASP architecture are defined in Table. 6.3. Due to the diffused components, the DDASP architecture requires more processing nodes than the PDASP architecture. The working procedure of each node in this architecture is the same as that of  $2 \times 2$  MIMO system with no diffused components. Therefore, the strict and sufficient condition with respect to processing time can be written as

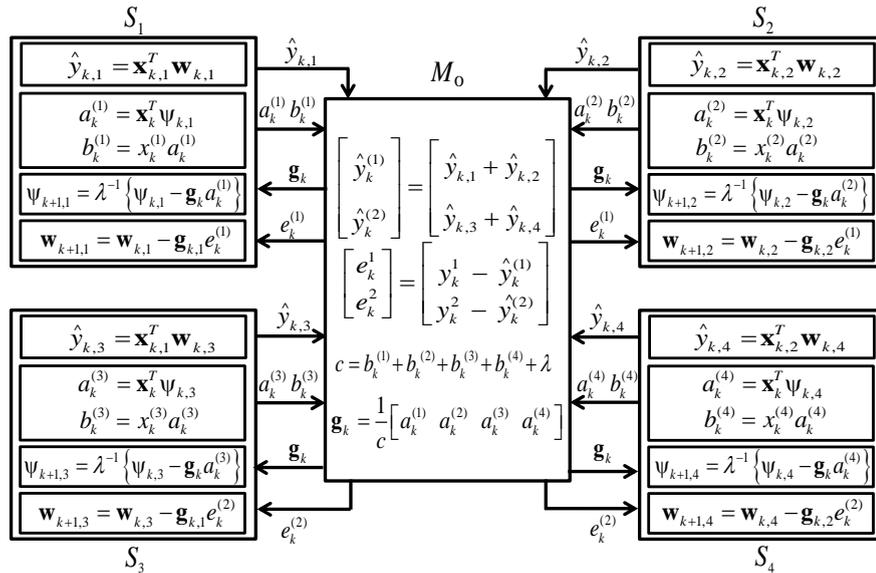
TABLE 6.3: Vector notations used in DDASP architecture for  $2 \times 2$  MIMO system with one diffused component

$$\begin{aligned}
 \mathbf{x}_{k,1}^T &= [x_k^{(1)} \ x_k^{(2)}], \quad \mathbf{x}_{k,2}^T = [x_k^{(3)} \ x_k^{(4)}] \\
 \mathbf{w}_{k,1}^T &= [w_k^{(11)} \ w_k^{(12)}], \quad \mathbf{w}_{k,2}^T = [w_k^{(13)} \ w_k^{(14)}] \\
 \mathbf{w}_{k,3}^T &= [w_k^{(21)} \ w_k^{(22)}], \quad \mathbf{w}_{k,4}^T = [w_k^{(23)} \ w_k^{(24)}] \\
 \mathbf{x}_k^T &= [x_k^{(1)} \ x_k^{(2)} \ x_k^{(3)} \ x_k^{(4)}] \\
 \Psi_{k,1}^T &= [\Psi_k^{(11)} \ \Psi_k^{(21)} \ \Psi_k^{(31)} \ \Psi_k^{(41)}], \\
 \Psi_{k,2}^T &= [\Psi_k^{(12)} \ \Psi_k^{(22)} \ \Psi_k^{(32)} \ \Psi_k^{(42)}] \\
 \Psi_{k,3}^T &= [\Psi_k^{(13)} \ \Psi_k^{(23)} \ \Psi_k^{(33)} \ \Psi_k^{(43)}], \\
 \Psi_{k,4}^T &= [\Psi_k^{(14)} \ \Psi_k^{(24)} \ \Psi_k^{(34)} \ \Psi_k^{(44)}] \\
 \mathbf{g}_{k,1}^T &= [g_k^{(1)} \ g_k^{(2)}], \quad \mathbf{g}_{k,2}^T = [g_k^{(3)} \ g_k^{(4)}]
 \end{aligned}$$

$$T_{S_1} = T_{S_2} = T_{S_3} = T_{S_4} = T_S$$

$$T_S + T_{M_0} + T_{f,(S_1,2,3,4 \leftrightarrow M_0)} = T_{DDASP} \quad (6.6)$$

$$T_{DDASP} \ll T_{PDASP}$$


 FIGURE 6.3: Proposed DDASP architecture for  $2 \times 2$  MIMO channel estimation with one multipath component

where  $T_S$  is the processing time taken by each slave node,  $T_{M_o}$  is the processing time of master node and  $T_{f,(S_{1,2,3,4} \leftrightarrow M_o)}$  is the total fetch time.

## 6.4 Results and Discussion

### 6.4.1 Computational Complexity Comparison

In this section, the theoretical results of the computational complexity of proposed PDASP, proposed DDASP and existing distributed architectures [74, 75] are discussed for different MIMO communication systems considering LoS-only and multipath based links. The maximum computational complexity of PDASP architecture entails  $2(N + NL)^2$  multiplications and  $(N + NL)^2$  additions per iteration. However, in DDASP architecture, the computational complexity of any of the slave node provides  $3(N + NL) + 2N + 1$  multiplications and  $2(N + NL) + 2N - 2$  additions per iteration; likewise, the master node entails  $N + NL + 1$  multiplications and  $2(N + NL)$  additions. Therefore, the combined computational complexity of the slave and master node provides  $4(N + NL) + 2N + 2$  multiplications and  $4(N + NL) + 2N - 2$  additions. The per-iteration multiplication complexity comparison between proposed DDASP and proposed PDASP architecture for  $2 \times 2$ ,  $4 \times 4$  and  $6 \times 6$  MIMO communication systems are shown in Fig. 6.4, Fig. 6.5 and Fig. 6.6, respectively. Likewise, the per-iteration addition complexity comparison between proposed DDASP and proposed PDASP architectures for  $2 \times 2$ ,  $4 \times 4$  and  $6 \times 6$  MIMO communication systems are shown in Fig. 6.7, Fig. 6.8 and Fig. 6.9, respectively. It is observed that at every number of multipath component, the proposed DDASP technique provides parallelly much lesser multiplication and addition complexity than the proposed PDASP architecture. Furthermore, the computational complexity comparison among the proposed PDASP, proposed DDASP and other notable existing distributed architectures is shown in Table. 6.4. It is realized that the computational complexity provided by the existing notable distributed techniques on each processing node is much higher than those of the proposed

PDASP and DDASP architectures. Moreover, among all these distributed techniques, DDASP architecture provides linear computational complexity parallelly on each node involved. Furthermore, the multiplication and addition complexity comparisons between the proposed PDASP and DDASP for different MIMO communication systems are shown in Table. 6.5 and Table. 6.6, respectively. It is realized that DDASP architecture provides much lower multiplication and addition complexity for all MIMO communication system with diffused components. However, in case of LoS-only MIMO communication link, the addition and multiplication complexity provided by PDASP architecture is lower than DDASP architecture. As it is discussed earlier that the complexity of the adaptive algorithm is directly dependent on the number of communication streams and the multipath components. Therefore, in case of multipath based link, the computational complexity of the adaptive algorithm is very high which can be found easily using Eq. 6.2 and DDASP becomes the only suitable choice to run the computationally complex adaptive algorithm on a distributed and diffused platform. On the other hand, in case of LoS-only communication link, the computational complexity of the adaptive algorithm is not so high and PDASP architecture come to be the right choice. Furthermore, the processing time based on their computational complexity taken by the PDASP and DDASP architectures using Arduino UNO platform is shown in Table. 6.7. In case of diffused component addition, it is observed that the processing time taken by DDASP architecture is lower than the PDASP architecture. However, in case of LoS-only link, the PDASP architecture provides a significant improvement in processing time than the DDASP architecture.

### 6.4.2 Node Requirement

The number of devices required in the proposed PDASP, proposed DDASP and notable existing architectures [74, 75] for  $1 \times 1$ ,  $2 \times 2$  and  $3 \times 3$  MIMO communication systems are shown in Table. 6.8. It is observed that the proposed PDASP architecture requires only 4 nodes for any order of MIMO communication system.

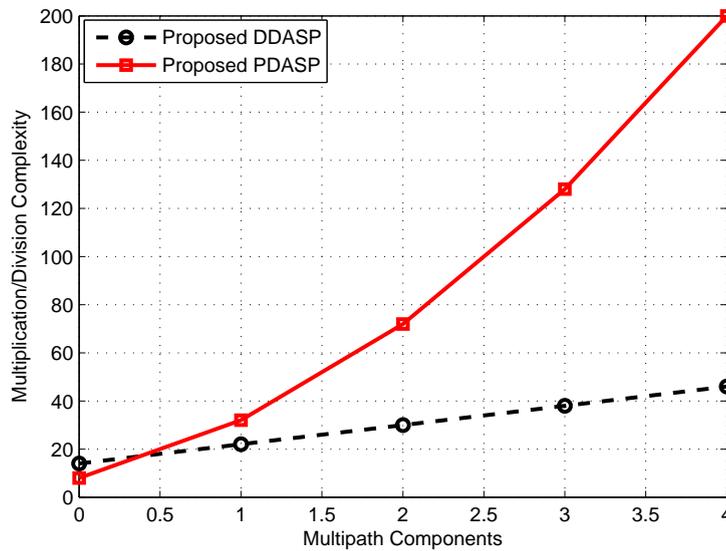


FIGURE 6.4: Per-iteration multiplication complexity comparison between  $2 \times 2$  proposed DDASP and proposed PDASP architectures

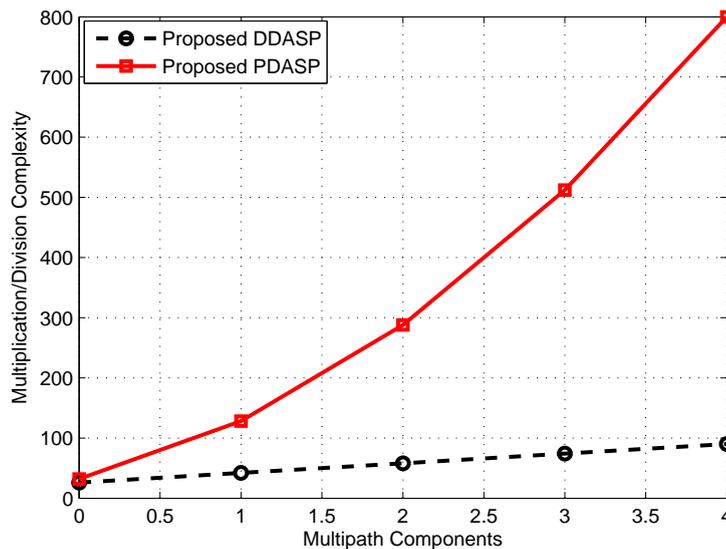


FIGURE 6.5: Per-iteration multiplication complexity comparison between  $4 \times 4$  proposed DDASP and proposed PDASP architectures

On the other hand, the number of nodes required for the proposed DDASP architecture are dependent upon the system order and increases with an increase in the system order. However, the existing distributed techniques [74, 75] require  $K$  nodes to run the adaptive algorithm distributively, where  $K$  is equal to the total number of iterations required for the complete convergence of the adaptive filtering algorithm. Moreover, in the existing distributed techniques, each iteration task of

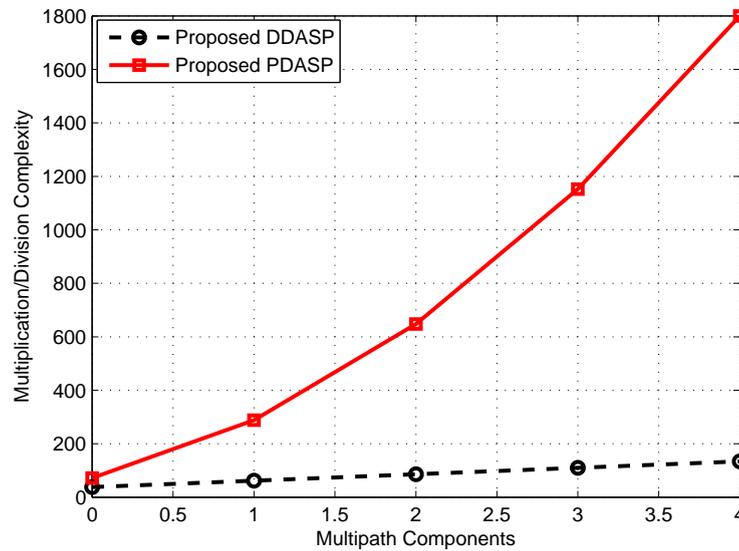


FIGURE 6.6: Per-iteration multiplication complexity comparison between  $6 \times 6$  proposed DDASP and proposed PDASP architectures

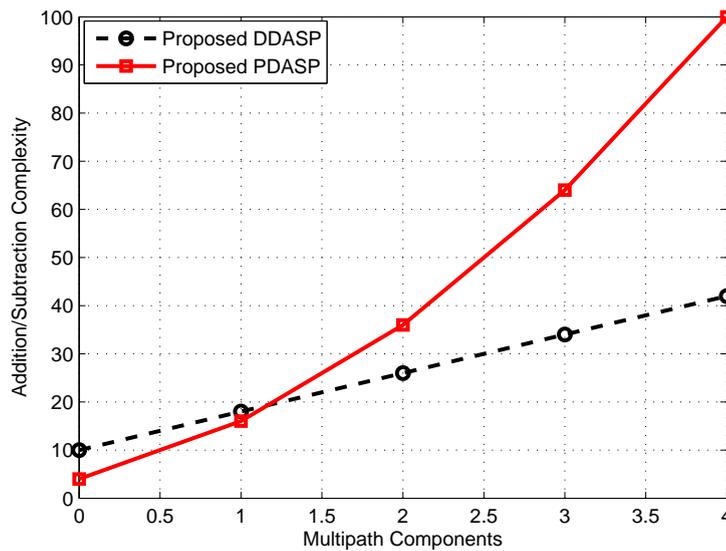


FIGURE 6.7: Per-iteration addition complexity comparison between  $2 \times 2$  proposed DDASP and proposed PDASP architectures

the adaptive algorithm is assigned to the specific node in the distributed network. In this context, all the nodes entail the complex computational complexity of the adaptive algorithm and each node is being free for  $K - 1$  iterations which are the major limitations of these existing techniques.

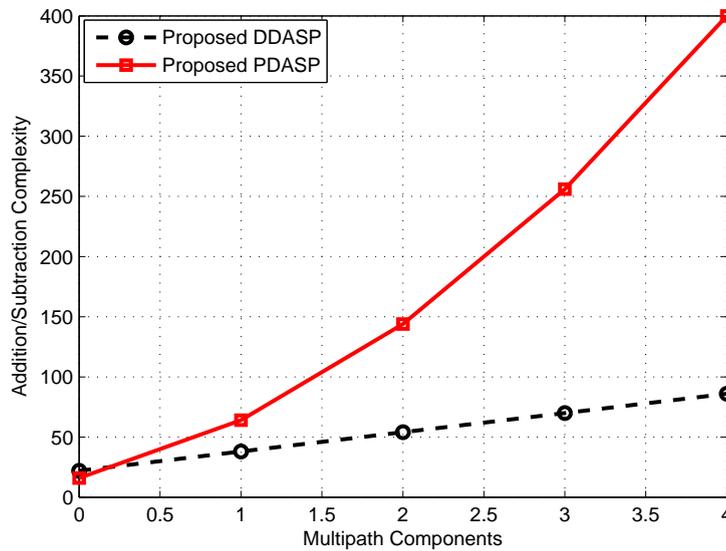


FIGURE 6.8: Per-iteration addition complexity comparison between  $4 \times 4$  proposed DDASP and proposed PDASP architectures

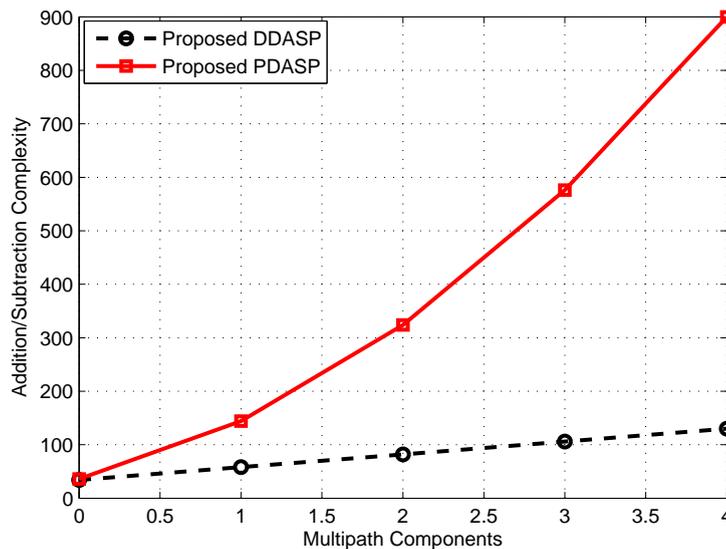


FIGURE 6.9: Per-iteration addition complexity comparison between  $6 \times 6$  proposed DDASP and proposed PDASP architectures

### 6.4.3 Communication Burden

The communication burden comparison among the proposed PDASP, proposed DDASP and notable existing architectures [74, 75] for  $1 \times 1$ ,  $2 \times 2$  and  $3 \times 3$  MIMO communication systems is shown in Table. 6.9. It is observed that the communication burden provided by the PDASP architecture is almost equivalent or lesser

TABLE 6.4: Computational complexity comparison for various distributed techniques using  $N \times N$  MIMO system with LoS-only links

Computational Complexity	PDASP	DDASP	dRLS [74]	LC-dRLS [74]	BC-RLS [75]
Multiplication Complexity	$2N^2$	$6N + 2$	$5N^2 + 2N + 2$	$5N^2 + 2N + 2$	$6N^2 + 3N + 3$
Addition Complexity	$N^2$	$6N - 2$	$4N^2$	$4N^2$	$5N^2$

TABLE 6.5: Distributed multiplication complexity for various MIMO systems with LoS-only and multipath based link

Architecture Name	Multipath Components	$2 \times 2$ MIMO	$3 \times 3$ MIMO	$4 \times 4$ MIMO
PDASP	No Multipath	8	17	32
DDASP	No Multipath	14	20	26
PDASP	One Multipath	32	72	128
DDASP	One Multipath	22	32	42
PDASP	Two Multipaths	72	162	288
DDASP	Two Multipaths	30	44	58

TABLE 6.6: Distributed addition complexity for various MIMO systems with LoS-only and and multipath based link

Architecture Name	Multipath Components	$2 \times 2$ MIMO	$3 \times 3$ MIMO	$4 \times 4$ MIMO
PDASP	No Multipath	4	9	16
DDASP	No Multipath	10	16	22
PDASP	One Multipath	16	36	64
DDASP	One Multipath	18	28	38
PDASP	Two Multipaths	36	81	144
DDASP	Two Multipaths	26	40	54

TABLE 6.7: Processing time comparison between DDASP and PDASP in  $\mu sec$  for different MIMO systems with LoS-only and multipath based link

Architecture Name	Multipath Components	$2 \times 2$ MIMO	$3 \times 3$ MIMO	$4 \times 4$ MIMO
PDASP	No Multipath	196	348	512
DDASP	No Multipath	268	396	602
PDASP	One Multipath	524	1008	1772
DDASP	One Multipath	396	490	652
PDASP	Two Multipaths	1020	2268	4544
DDASP	Two Multipaths	448	652	976

TABLE 6.8: Number of processing nodes required in various distributed techniques for different communication systems

Communication system	PDASP	DDASP	dRLS [74]	LC-dRLS [74]	BC-RLS [75]
$1 \times 1$ SISO system	4	1	$K$	$K$	$K$
$2 \times 2$ MIMO system	4	3	$K$	$K$	$K$
$3 \times 3$ MIMO system	4	4	$K$	$K$	$K$

than the dRLS [75] and BCRLS [74] algorithms. Likewise, the communication burden entailed by DDASP architecture is slightly greater than the both dRLS and BCRLS algorithms. On the other hand, LC-dRLS technique [75] provides lesser communication burden than all the other techniques including PDASP and DDASP. This reduction in the communication burden by LC-dRLS is due to the assumption that each node in the distributed network uses the initialized value of error covariance matrix  $\Psi_k$  and it is not needed to be transmitted from one node to another. Although LC-dRLS reduces the network communication burden; however, it degrades the convergence performance of the adaptive algorithm significantly.

TABLE 6.9: Communication burden comparison on various distributed architectures for different communication systems

Communication system	PDASP	DDASP	dRLS [74]	LC-dRLS [74]	BC-RLS [75]
$1 \times 1$ SISO system	$3K$	–	$2K$	$K$	$2K$
$2 \times 2$ MIMO system	$8K$	$12K$	$8K$	$4K$	$8K$
$3 \times 3$ MIMO system	$15K$	$21K$	$18K$	$9K$	$18K$

# Chapter 7

## Application of Proposed DDASP Architecture on Missing Data Systems

In this chapter, the proposed DDASP architecture presented in Chapter 6 is used to estimate the parameters of auxiliary model of missing data application. Section 7.1 presents an overview of the topic. Section 7.2 describes the system model and presents missing data patterns. Section 7.3 presents an auxiliary model description of missing data systems. Section 7.4 describes the implementation of DDASP architecture on missing data systems. Section 7.5 presents the complexity analysis of the proposed DDASP architecture with respect to its implementation on missing data systems and compares it with complexity of the sequential RLS. Finally, Section 7.6 presents the simulation results and their discussion.

### 7.1 Overview

In the case of the implementation of PDASP architecture for the estimation of MIMO channel coefficients, the RLS adaptive algorithm perfectly runs parallelly on a group of nodes even with non-aligned time indexes. However, as far as its

implementation for missing data application is concerned it shows unstable behavior due to its inherent working based on non-aligned time indexes. Since, missing data application needs perfect time alignment among various parts of the algorithm in order to reach a specific goal. Therefore, sequentially-run algorithm (e.g. RLS, etc) or diffusion-oriented architecture (e.g. DDASP) become the only suitable choice for system identification in case of missing data systems. In the literature, a number of auxiliary model-based methods have been presented to find the coefficients of reference system [95]. Likewise, in [96], an auxiliary model for dual-rate sampled data system with the affect of colored noise is introduced. The extended least square adaptive filtering algorithm is used for finding coefficients in the auxiliary model of missing data systems [97]. In [98–100], a least square based filtering algorithm for system identification is used in multi-rate systems. In [23], an RLS algorithm is used for two-fold applications, i.e. to find the coefficients of auxiliary model and to estimate the missing data in stochastic framework. In [101], an auxiliary model using RLS filtering algorithm is introduced with state-space model for dual-rate systems. From the above discussion, it is realized that all the sequential techniques used for finding coefficients of auxiliary model provide high complex computational complexity. Nevertheless, such high definition adaptive algorithm cannot be run on an energy-limited and computationally-constrained inexpensive platform. Moreover, due to high computational complexity of these signal processing algorithms, an energy-constrained node may not exhibits adequate performance with limited power source and existing hardware. Therefore, the proposed distributed diffusion-based adaptive signal processing (DDASP) become the only suitable choice.

## 7.2 System model and Missing Data Patterns

### 7.2.1 Noise-free Discrete Missing Data System

Consider the following input-output relationship of noise-free discrete time system of missing data

$$\begin{aligned} y_k &= X(z)x_k \\ &= \frac{N(z)}{D(z)}x_k \end{aligned} \quad (7.1)$$

where,  $x_k$  and  $y_k$  are the input and output of the above-mentioned discrete system, respectively, and  $X(z)$  shows its transfer function. The numerator  $N(z)$  and denominator  $D(z)$  can be defined as

$$\begin{aligned} N(z) &= \bar{b}_1 z^{-1} + \cdots + \bar{b}_N z^{-N} \\ D(z) &= 1 + \bar{a}_1 z^{-1} + \cdots + \bar{a}_N z^{-N} \end{aligned} \quad (7.2)$$

Assume that the order of polynomials in (7.2) is known *a priori*. Let  $\mathbf{h}$  be the transfer function coefficients vector and  $\mathbf{u}_k$  be the input-output information vector defined as

$$\mathbf{h} = [\bar{a}_1, \bar{a}_2, \cdots, \bar{a}_N, \bar{b}_1, \bar{b}_2, \cdots, \bar{b}_N] \quad (7.3)$$

$$\mathbf{u}_k = [-y_{k-1}, -y_{k-2}, \cdots, -y_{k-N}, x_{k-1}, x_{k-2}, \cdots, x_{k-N}] \quad (7.4)$$

Rewriting Eq. (7.1), we get

$$y_k = \mathbf{u}_k^T \mathbf{h} \quad (7.5)$$

In the System identification problem, it is assumed that the output data from the unknown system is available at every sample point. However, due to sensory limitations and infrequent data sampling, the subset of output data is available rather than the complete output information. Therefore, the missing data examination function can be expressed as

$$\rho_k = \begin{cases} 1 & \text{if } y_k \text{ available} \\ 0 & \text{otherwise} \end{cases} \quad (7.6)$$

For instance, the irregular missing data output pattern and the respective discrete time indexes corresponding to its values can be defined as

$$\mathbf{y}_k = [y_0, y_1, y_3, y_6, y_7, y_9, y_{10}, y_{14}, y_{15}, \dots]$$

$$k_0 = 0, k_1 = 1, k_2 = 3, k_3 = 6, k_4 = 7, k_5 = 9, k_6 = 10, k_7 = 14, k_8 = 15, \dots$$

Likewise, their corresponding missing data examination function can also be written as

$$\rho_0 = \rho_1 = \rho_3 = \rho_6 = \rho_7 = \rho_9 = \rho_{10} = \rho_{14} = \rho_{15} = 1, \dots$$

In the missing data output framework, there is no regular pattern of the output data available. In this context, a stochastic framework can only be used rather than the use of FIR filter. In the stochastic framework, complete known training input is given to the adaptive algorithm with partial output data for parameter identification.

### 7.2.2 Noisy Discrete-time Missing Data System

Consider the missing data output system in stochastic framework shown in Fig. 7.1. The input output relationship for the missing data system can be written as

$$y_k = r_k + \omega_k \tag{7.7}$$

where  $\omega_k$  is additive white noise with variance  $\sigma_\omega^2$ ,  $y_k$  is the measurable system output and  $r_k$  is the unmeasurable noise free output given as

$$r_k = \frac{N(z)}{D(z)}x_k \tag{7.8}$$

The vector  $\mathbf{h}$  of transfer function coefficients is the same as defined in section 7.2 and the updated form of input-output information vector  $\mathbf{u}_k$  can be expressed as

$$\mathbf{u}_k = [-r_{k-1}, -r_{k-2}, \dots, -r_{k-N}, x_{k-1}, x_{k-2}, \dots, x_{k-N}] \tag{7.9}$$

Rewriting Eq. (7.7) and Eq. (7.8) in terms of  $\mathbf{u}_k$  and  $\mathbf{h}$ , we get

$$r_k = \mathbf{u}_k^T \mathbf{h} \quad (7.10)$$

$$y_k = \mathbf{u}_k^T \mathbf{h} + v_k \quad (7.11)$$

### 7.3 Auxiliary Model

The reference/auxiliary model  $N_a(z)/D_a(z)$  with input  $x_m$  is shown in Fig. 7.2, where, the numerator  $N_a(z)$  and denominator  $D_a(z)$  of transfer function auxiliary system are the same as those of  $N(z)$  and  $D(z)$  of original system, respectively. Therefore, rewriting the Eq. (7.10) in terms of time index  $m$  for auxiliary model, gives

$$\hat{r}_m = \mathbf{u}_m^T \hat{\mathbf{h}} \quad (7.12)$$

likewise, the updated form of input-output information vector according to auxiliary model can thus be written as

$$\mathbf{u}_m = [-\hat{r}_{m-1}, -\hat{r}_{m-2}, \dots, -\hat{r}_{m-N}, x_{m-1}, x_{m-2}, \dots, x_{m-N}] \quad (7.13)$$

where  $\mathbf{u}_m$  is the information vector of an auxiliary model. Considering a minimization problem, the cost function according to least square error criterion can be defined as

$$J(\mathbf{h}) = [y_m - \mathbf{u}_m^T \mathbf{h}]^2 \quad (7.14)$$

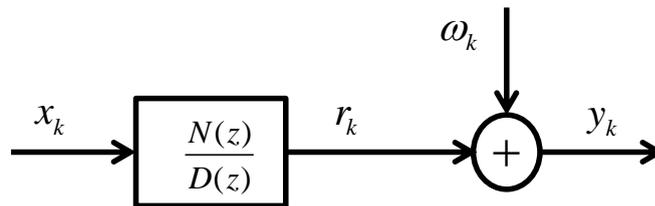


FIGURE 7.1: Missing data error system

According to this minimization criteria, the proposed low-complexity algorithm (proposed in Chapter 4) based on the above-defined auxiliary model for missing data system can be outlined as follows:

$$\hat{\mathbf{h}}_m = \hat{\mathbf{h}}_{m-1} + e_m \Psi_m \mathbf{u}_m \quad (7.15)$$

$$e_m = y_m - \mathbf{u}_m^T \hat{\mathbf{h}}_{m-1} \quad (7.16)$$

$$\hat{r}_m = \mathbf{u}_m^T \hat{\mathbf{h}}_{m-1} \quad (7.17)$$

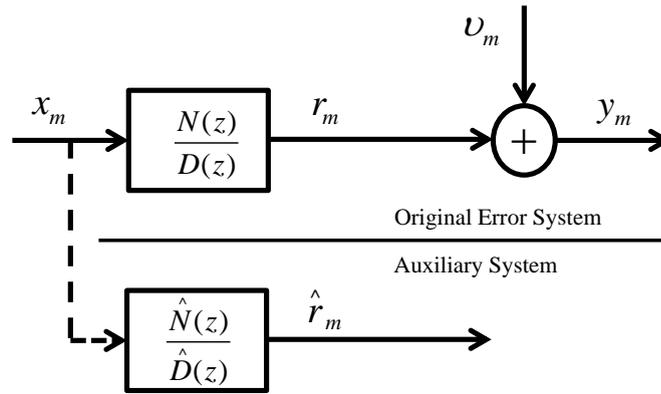


FIGURE 7.2: Missing data error system with reference model

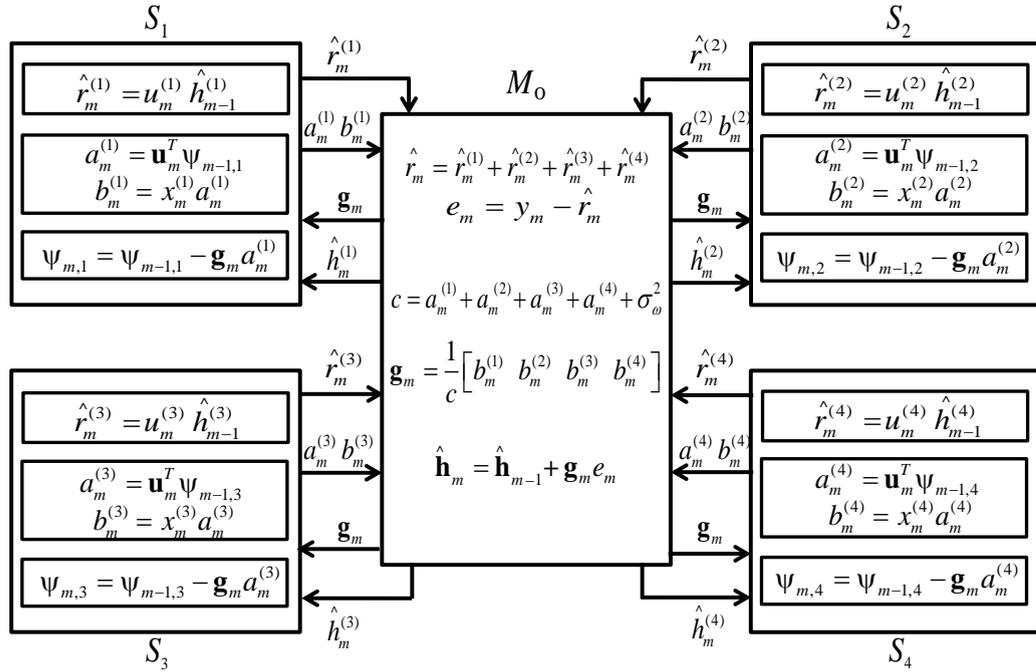


FIGURE 7.3: Proposed DDASP architecture for low-complexity estimation algorithm with aligned time indexes

$$\mathbf{g}_m = \Psi_m \mathbf{u}_m = \frac{\Psi_{m-1} \mathbf{u}_m}{\sigma_v^2 + \mathbf{u}_m^T \Psi_{m-1} \mathbf{u}_m} \tag{7.18}$$

$$\begin{aligned} \Psi_m &= \left[ \Psi_{m-1} - \frac{\Psi_{m-1} \mathbf{u}_m \mathbf{u}_m^T \Psi_{m-1}}{\sigma_v^2 + \mathbf{u}_m^T \Psi_{m-1} \mathbf{u}_m} \right] \\ &= \left[ \Psi_{m-1} - \mathbf{g}_m \mathbf{u}_m^T \Psi_{m-1} \right] \end{aligned} \tag{7.19}$$

where,  $\Psi_m$  is the covariance matrix and  $\mathbf{g}_m$  is the Kalman gain vector. It is pertinent to note that the initialization of covariance matrix  $\Psi_0$  must be a large positive constant matrix which may affect the convergence performance of the algorithm [13].

## 7.4 Implementation of DDASP on Missing Data System

The flow diagram of the proposed DDASP architecture with the implementation of low-complexity estimation algorithm for missing data application is shown in Fig. 7.3. The distributed architecture consists of one master node  $M_o$  and four slave nodes  $S_1, S_2, S_3$  and  $S_4$  for  $N = 4$  estimation system. The number of slave nodes is proportional to the length of the estimation vector  $N$ . The diffused tasks for the estimated values of the received signals  $\hat{r}_m^{(1)}, \hat{r}_m^{(2)}, \hat{r}_m^{(3)}$  and  $\hat{r}_m^{(4)}$  in missing data system can be defined as

$$\begin{aligned} \hat{\mathbf{r}}_m &= \mathbf{u}_m^T \hat{\mathbf{h}}_{m-1} \\ \hat{r}_m^{(1)} &= u_m^{(1)} \hat{h}_{m-1}^{(1)} = -\hat{r}_{m-1} \hat{a}_1 \\ \hat{r}_m^{(2)} &= u_m^{(2)} \hat{h}_{m-1}^{(2)} = -\hat{r}_{m-2} \hat{a}_2 \\ \hat{r}_m^{(3)} &= u_m^{(3)} \hat{h}_{m-1}^{(3)} = \hat{x}_{m-1} \hat{b}_1 \\ \hat{r}_m^{(4)} &= u_m^{(4)} \hat{h}_{m-1}^{(4)} = \hat{x}_{m-2} \hat{b}_2 \end{aligned}$$

Likewise, diffused tasks for components  $a_m^{(1)}, a_m^{(2)}, a_m^{(3)}, a_m^{(4)}, b_m^{(1)}, b_m^{(2)}, b_m^{(3)}, b_m^{(4)}$  and  $c$  that are used to make the formation of Kalman gain  $\mathbf{g}_m$ , can be expressed as

$$\begin{aligned}
\mathbf{g}_m &= \frac{\mathbf{u}_m^T \boldsymbol{\Psi}_{m-1}}{\mathbf{u}_k^T \boldsymbol{\Psi}_{m-1} \mathbf{u}_m + \sigma_\omega^2} \\
\mathbf{g}_m &= \frac{\begin{bmatrix} u_m^{(1)} & u_m^{(2)} & u_m^{(3)} & u_m^{(4)} \end{bmatrix} \begin{bmatrix} \Psi_{m-1}^{(11)} & \Psi_{m-1}^{(12)} & \Psi_{m-1}^{(13)} & \Psi_{m-1}^{(14)} \\ \Psi_{m-1}^{(21)} & \Psi_{m-1}^{(22)} & \Psi_{m-1}^{(23)} & \Psi_{m-1}^{(24)} \\ \Psi_{m-1}^{(31)} & \Psi_{m-1}^{(32)} & \Psi_{m-1}^{(33)} & \Psi_{m-1}^{(34)} \\ \Psi_{m-1}^{(41)} & \Psi_{m-1}^{(42)} & \Psi_{m-1}^{(43)} & \Psi_{m-1}^{(44)} \end{bmatrix}}{\begin{bmatrix} u_m^{(1)} & u_m^{(2)} & u_m^{(3)} & u_m^{(4)} \end{bmatrix} \begin{bmatrix} \Psi_{m-1}^{(11)} & \Psi_{m-1}^{(12)} & \Psi_{m-1}^{(13)} & \Psi_{m-1}^{(14)} \\ \Psi_{m-1}^{(21)} & \Psi_{m-1}^{(22)} & \Psi_{m-1}^{(23)} & \Psi_{m-1}^{(24)} \\ \Psi_{m-1}^{(31)} & \Psi_{m-1}^{(32)} & \Psi_{m-1}^{(33)} & \Psi_{m-1}^{(34)} \\ \Psi_{m-1}^{(41)} & \Psi_{m-1}^{(42)} & \Psi_{m-1}^{(43)} & \Psi_{m-1}^{(44)} \end{bmatrix} \begin{bmatrix} u_m^{(1)} \\ u_m^{(2)} \\ u_m^{(1)} \\ u_m^{(2)} \end{bmatrix} + \sigma_\omega^2} \\
\mathbf{g}_m &= \frac{\begin{bmatrix} a_m^{(1)} & a_m^{(2)} & a_m^{(3)} & a_m^{(4)} \end{bmatrix}}{\begin{bmatrix} a_m^{(1)} & a_m^{(2)} & a_m^{(3)} & a_m^{(4)} \end{bmatrix} \begin{bmatrix} u_m^{(1)} \\ u_m^{(2)} \\ u_m^{(3)} \\ u_m^{(4)} \end{bmatrix} + \sigma_\omega^2} = \frac{\begin{bmatrix} a_m^{(1)} & a_m^{(2)} & a_m^{(3)} & a_m^{(4)} \end{bmatrix}}{b_m^{(1)} + b_m^{(2)} + b_m^{(3)} + b_m^{(4)} + \sigma_\omega^2}
\end{aligned}$$

$$a_m^{(1)} = \psi_{m-1}^{(11)} u_m^{(1)} + \psi_{m-1}^{(21)} u_m^{(2)} + \psi_{m-1}^{(31)} u_m^{(3)} + \psi_{m-1}^{(41)} u_m^{(4)}$$

$$a_m^{(2)} = \psi_{m-1}^{(12)} u_m^{(1)} + \psi_{m-1}^{(22)} u_m^{(2)} + \psi_{m-1}^{(32)} u_m^{(3)} + \psi_{m-1}^{(42)} u_m^{(4)}$$

$$a_m^{(3)} = \psi_{m-1}^{(13)} u_m^{(1)} + \psi_{m-1}^{(23)} u_m^{(2)} + \psi_{m-1}^{(33)} u_m^{(3)} + \psi_{m-1}^{(43)} u_m^{(4)}$$

$$a_m^{(4)} = \psi_{m-1}^{(14)} u_m^{(1)} + \psi_{m-1}^{(24)} u_m^{(2)} + \psi_{m-1}^{(34)} u_m^{(3)} + \psi_{m-1}^{(44)} u_m^{(4)}$$

$$b_m^{(1)} = u_m^{(1)} a_m^{(1)}$$

$$b_m^{(2)} = u_m^{(2)} a_m^{(2)}$$

$$b_m^{(3)} = u_m^{(3)} a_m^{(3)}$$

$$b_m^{(4)} = u_m^{(4)} a_m^{(4)}$$

$$c = b_m^{(1)} + b_m^{(2)} + b_m^{(3)} + b_m^{(4)} + \sigma_\omega^2$$

Similarly, the diffusion procedure of covariance matrix  $\boldsymbol{\Psi}_m$  for DDASP architecture in missing data system can be defined as

$$\begin{aligned}
\mathbf{\Psi}_m &= \mathbf{\Psi}_{m-1} - \mathbf{g}_m \mathbf{u}_m^T \mathbf{\Psi}_{m-1} \\
\mathbf{\Psi}_m &= \begin{bmatrix} \Psi_{m-1}^{(11)} & \Psi_{m-1}^{(12)} & \Psi_{m-1}^{(13)} & \Psi_{m-1}^{(14)} \\ \Psi_{m-1}^{(21)} & \Psi_{m-1}^{(22)} & \Psi_{m-1}^{(23)} & \Psi_{m-1}^{(24)} \\ \Psi_{m-1}^{(31)} & \Psi_{m-1}^{(32)} & \Psi_{m-1}^{(33)} & \Psi_{m-1}^{(34)} \\ \Psi_{m-1}^{(41)} & \Psi_{m-1}^{(42)} & \Psi_{m-1}^{(43)} & \Psi_{m-1}^{(44)} \end{bmatrix} \\
&\quad - \begin{bmatrix} g_m^{(1)} \\ g_m^{(2)} \\ g_m^{(3)} \\ g_m^{(4)} \end{bmatrix} \begin{bmatrix} u_m^{(1)} & u_m^{(2)} & u_m^{(3)} & u_m^{(4)} \end{bmatrix} \begin{bmatrix} \Psi_{m-1}^{(11)} & \Psi_{m-1}^{(12)} & \Psi_{m-1}^{(13)} & \Psi_{m-1}^{(14)} \\ \Psi_{m-1}^{(21)} & \Psi_{m-1}^{(22)} & \Psi_{m-1}^{(23)} & \Psi_{m-1}^{(24)} \\ \Psi_{m-1}^{(31)} & \Psi_{m-1}^{(32)} & \Psi_{m-1}^{(33)} & \Psi_{m-1}^{(34)} \\ \Psi_{m-1}^{(41)} & \Psi_{m-1}^{(42)} & \Psi_{m-1}^{(43)} & \Psi_{m-1}^{(44)} \end{bmatrix} \\
\mathbf{\Psi}_m &= \begin{bmatrix} \Psi_{m-1}^{(11)} & \Psi_{m-1}^{(12)} & \Psi_{m-1}^{(13)} & \Psi_{m-1}^{(14)} \\ \Psi_{m-1}^{(21)} & \Psi_{m-1}^{(22)} & \Psi_{m-1}^{(23)} & \Psi_{m-1}^{(24)} \\ \Psi_{m-1}^{(31)} & \Psi_{m-1}^{(32)} & \Psi_{m-1}^{(33)} & \Psi_{m-1}^{(34)} \\ \Psi_{m-1}^{(41)} & \Psi_{m-1}^{(42)} & \Psi_{m-1}^{(43)} & \Psi_{m-1}^{(44)} \end{bmatrix} - \begin{bmatrix} g_m^{(1)} \\ g_m^{(2)} \\ g_m^{(3)} \\ g_m^{(4)} \end{bmatrix} \begin{bmatrix} a_m^{(1)} & a_m^{(2)} & a_m^{(3)} & a_m^{(4)} \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\begin{bmatrix} \Psi_m^{(11)} \\ \Psi_m^{(21)} \\ \Psi_m^{(31)} \\ \Psi_m^{(41)} \end{bmatrix} &= \begin{bmatrix} \Psi_{m-1}^{(11)} \\ \Psi_{m-1}^{(21)} \\ \Psi_{m-1}^{(31)} \\ \Psi_{m-1}^{(41)} \end{bmatrix} - \begin{bmatrix} a_m^{(1)} g_m^{(1)} \\ a_m^{(1)} g_m^{(2)} \\ a_m^{(1)} g_m^{(3)} \\ a_m^{(1)} g_m^{(4)} \end{bmatrix} \\
\begin{bmatrix} \Psi_m^{(12)} \\ \Psi_m^{(22)} \\ \Psi_m^{(32)} \\ \Psi_{m+1}^{(42)} \end{bmatrix} &= \begin{bmatrix} \Psi_{m-1}^{(12)} \\ \Psi_{m-1}^{(22)} \\ \Psi_{m-1}^{(32)} \\ \Psi_{m-1}^{(42)} \end{bmatrix} - \begin{bmatrix} a_m^{(2)} g_m^{(1)} \\ a_m^{(2)} g_m^{(2)} \\ a_m^{(2)} g_m^{(3)} \\ a_m^{(2)} g_m^{(4)} \end{bmatrix} \\
\begin{bmatrix} \Psi_m^{(13)} \\ \Psi_m^{(23)} \\ \Psi_m^{(33)} \\ \Psi_m^{(43)} \end{bmatrix} &= \begin{bmatrix} \Psi_{m-1}^{(13)} \\ \Psi_{m-1}^{(23)} \\ \Psi_{m-1}^{(33)} \\ \Psi_{m-1}^{(43)} \end{bmatrix} - \begin{bmatrix} a_m^{(3)} g_m^{(1)} \\ a_m^{(3)} g_m^{(2)} \\ a_m^{(3)} g_m^{(3)} \\ a_m^{(3)} g_m^{(4)} \end{bmatrix} \\
\begin{bmatrix} \Psi_m^{(14)} \\ \Psi_m^{(24)} \\ \Psi_m^{(34)} \\ \Psi_m^{(44)} \end{bmatrix} &= \begin{bmatrix} \Psi_{m-1}^{(14)} \\ \Psi_{m-1}^{(24)} \\ \Psi_{m-1}^{(34)} \\ \Psi_{m-1}^{(44)} \end{bmatrix} - \begin{bmatrix} a_m^{(4)} g_m^{(1)} \\ a_m^{(4)} g_m^{(2)} \\ a_m^{(4)} g_m^{(3)} \\ a_m^{(4)} g_m^{(4)} \end{bmatrix}
\end{aligned}$$

Let the processing time taken by covariance matrix  $\Psi_m$ , Kalman gain  $\mathbf{g}_m$ , parameter estimation error  $e_m$  and coefficient vector  $\hat{\mathbf{h}}_m$  be  $T_\Psi$ ,  $T_g$ ,  $T_e$  and  $T_{\hat{\mathbf{h}}}$ , respectively. Therefore, the overall processing time taken by the sequentially-operated adaptive algorithm when it runs in distributed manner is

$$T_\Psi + T_g + T_e + T_{\hat{\mathbf{h}}} = T_{\text{seq}} \quad (7.20)$$

where,  $T_{\text{seq}}$  is the time taken by the sequential algorithm when it runs in non parallel fashion. The cumulative fetch time  $T_{f,c}$  for the communication of data elements among the master and slave nodes can be written as

$$T_{f,\hat{r}_m^{(1,2,3,4)}} + T_{f,a_m^{(1,2,3,4)}} + T_{f,b_m^{(1,2,3,4)}} + T_{f,\hat{h}_m^{(1,2,3,4)}} + T_{f,\mathbf{g}(m)} = T_{f,c} \quad (7.21)$$

The strict and sufficient condition for the distributive parameter estimation of irregular missing data output system can thus be expressed as

$$T_{\Psi} + T_{\mathbf{g}} + T_e + T_{\hat{\mathbf{h}}} + T_{f,c} \leq T_{\text{seq}} \tag{7.22}$$

TABLE 7.1: Multiplication and addition complexity of DDASP architecture with respect to each processing node

Processing Node	Multiplication Complexity	Addition Complexity
$M_o$	$2N + 1$	$3N$
$S_1$	$2N + 2$	$2N - 1$
$S_2$	$2N + 2$	$2N - 1$
$S_3$	$2N + 2$	$2N - 1$
$S_4$	$2N + 2$	$2N - 1$

TABLE 7.2: Coefficient estimation of example scenario using conventional RLS algorithm at forgetting factor  $\lambda = 0.9999$

No. of Iterations $K$	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_1$	$\hat{b}_2$	$\gamma$
100	-0.5190	-0.1651	0.4457	0.6185	1.4247
200	-0.6341	-0.1749	0.2791	0.4819	1.4118
300	-0.7470	-0.0347	0.3198	0.4623	1.1543
500	-0.8985	0.0883	0.2962	0.3973	0.8503
1000	-1.2137	0.4089	0.3305	0.3195	0.3100
1500	-1.2935	0.4930	0.3193	0.2638	0.2079
2000	-1.3922	0.5764	0.3144	0.2266	0.1073
2500	-1.4577	0.6429	0.3050	0.2009	0.0412
3000	-1.4963	0.6819	0.3054	0.1918	0.0125
3500	-1.5143	0.7107	0.3078	0.1886	0.0122
4000	-1.5164	0.7190	0.3099	0.1903	0.0167
True Values	-1.5000	0.7000	0.3000	0.2000	

## 7.5 Complexity Comparison

In this section, the computational cost of the distributed low-complexity estimation algorithm is compared with both of sequentially-operated low-complexity

TABLE 7.3: Coefficient estimation of example scenario using proposed DDASP based low-complexity algorithm

No. of Iterations $K$	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_1$	$\hat{b}_2$	$\gamma$
100	-0.4635	-0.2441	0.4261	0.5879	1.5960
200	-0.6952	-0.1245	0.2603	0.4513	1.3255
300	-0.8816	0.0936	0.2969	0.4255	0.8698
500	-1.1003	0.2825	0.2824	0.3520	0.4898
1000	-1.4022	0.6089	0.3065	0.2725	0.0964
1500	-1.4121	0.6185	0.3020	0.2468	0.0818
2000	-1.4405	0.6375	0.3044	0.2337	0.0574
2500	-1.4758	0.6624	0.2981	0.2220	0.0398
3000	-1.4790	0.6765	0.2998	0.2105	0.0110
3500	-1.4854	0.6852	0.3029	0.2074	0.0098
4000	-1.4978	0.6980	0.3015	0.2027	0.0021
True Values	-1.5000	0.7000	0.3000	0.2000	

TABLE 7.4: Sequential and distributed multiplication complexity for various system orders in missing data systems

Estimation Vector of Order $N$	Sequential RLS Algorithm	Sequential Low-Complexity Algorithm	Distributive Complexity of Node $M_o$	Distributive Complexity of any Specific Node $S$
2	22	17	5	6
3	41	31	7	8
4	66	49	9	10
5	97	71	11	12
6	137	97	13	14
7	177	127	15	16
8	226	161	17	18
9	281	199	19	20
10	342	241	21	22

algorithm and conventional RLS algorithm. The multiplication and addition complexity of RLS algorithm entails  $3N^2+4N+2$  and  $2N^2+2N$ , respectively. Likewise, the low-complexity estimation algorithm provides  $2N^2 + 4N + 1$  multiplications and  $2N^2 + 2N$  additions per iterations. On the other hand, the proposed DDASP

TABLE 7.5: Sequential and distributed addition complexity for various system orders in missing data systems

Estimation Vector of Order $N$	Sequential RLS Algorithm	Sequential Low-Complexity Algorithm	Distributive Complexity of Node $M_o$	Distributive Complexity of any Specific Node $S$
2	12	12	6	3
3	24	24	9	5
4	40	40	12	7
5	60	60	15	9
6	84	84	18	11
7	112	112	21	13
8	144	144	24	15
9	180	180	27	17
10	220	220	30	19

TABLE 7.6: Sequential and proposed node-by-node distributed processing time in  $\mu\text{sec}$  for various system orders

Estimation Vector of Order $N$	Sequential RLS Algorithm	Sequential Low-Complexity Algorithm	Processing Time of Node $M_o$	Processing Time of any Specific Node $S$	Total Distributed Processing Time of All Nodes
2	568	532	1.33	4.1	5.34
4	1532	1432	8.42	20.23	28.65
6	3088	2700	12.33	24.13	36.46
8	4987	4412	14.13	28.42	42.55
10	7132	6414	15.92	31.89	47.81

based low-complexity estimation algorithm provides  $4N + 3$  multiplications and  $5N - 1$  additions at maximum. The step by step addition and multiplication complexity is shown in Table. 7.1. It is realized that the DDASP architecture provides linear computational complexity parallelly on each node involved. Furthermore, the multiplication and addition complexity comparisons among the master and slave nodes of DDASP architecture and sequentially-operated low-complexity and RLS algorithms are shown in Table. 7.4 and Table. 7.5, respectively. Likewise, the per-iteration multiplication and addition complexity comparisons of the combined

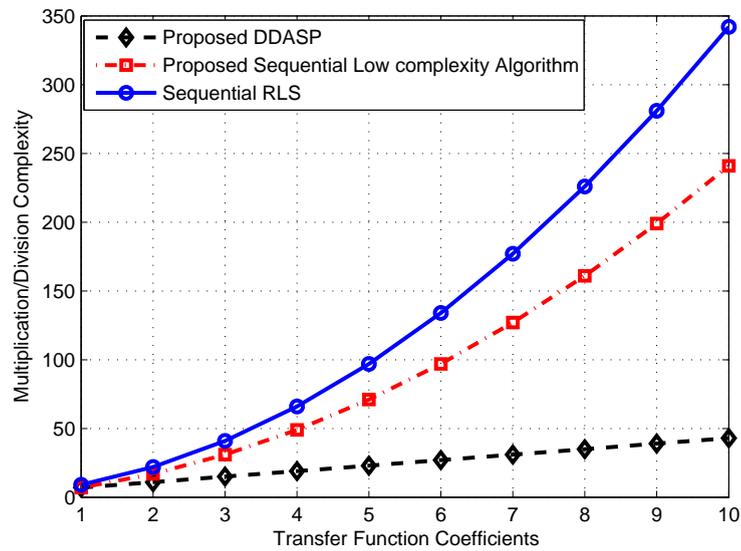


FIGURE 7.4: Per-iteration multiplication complexity comparison among sequential algorithms and proposed DDASP technique on missing data system

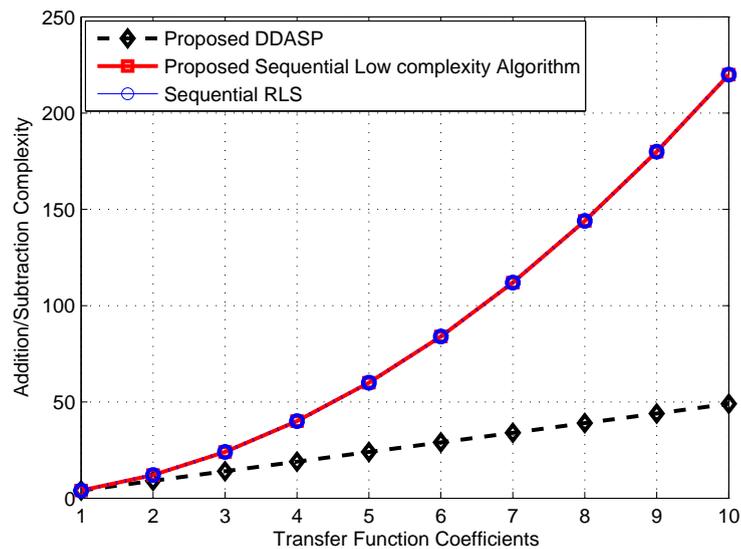


FIGURE 7.5: Per-iteration addition complexity comparison among sequential algorithms and proposed DDASP technique on missing data system

master and slave node of proposed DDASP architecture with those of sequentially-operated low-complexity and RLS algorithms are shown in Fig. 7.4 and Fig. 7.5, respectively. It is observed that at every order of transfer function coefficients, the proposed DDASP technique provides parallelly much lesser multiplication and addition complexity than the sequentially-operated low-complexity and RLS algorithms.

## 7.6 Simulation Results and Discussion

In this section, the measurement results are presented by considering the following second order system for missing data application

$$\begin{aligned}
 y_m &= \frac{N(z)}{D(z)}x_m + v_m \\
 &= \frac{b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}x_m + v_m \\
 &= \frac{0.30z^{-1} + 0.20z^{-2}}{1 + 1.50z^{-1} + 0.70z^{-2}}x_m + v_m
 \end{aligned}$$

where  $x_m$  is the input signal with zero mean and variance  $\sigma_x^2 = 1$  and  $v_m$  is additive white noise with zero mean and variance  $\sigma_w^2$ . The signal to noise ratio (SNR) is set to be 20dB according to the following relation

$$SNR = \frac{\sigma_x^2}{\sigma_w^2}$$

The proposed low-complexity estimation algorithm proposed in Chapter 4 is implemented on distributive architecture to substantiate the validation of parameter estimation of missing data system in terms of convergence performance, processing time and computational complexity. The coefficients estimation for the above example scenario by using the sequentially-operated RLS algorithm and the proposed DDASP based low-complexity algorithm are shown in Table. 7.2 and Table. 7.3, respectively. It is observed that the proposed DDASP based low-complexity algorithm provides fast convergence performance than the sequentially-operated RLS filtering algorithm.

Furthermore, the parameter estimation error  $\gamma = \|(\hat{\mathbf{h}}_m - \mathbf{h})\|/\|\hat{\mathbf{h}}_m\|$  using DDASP based low-complexity algorithm and sequentially-operated RLS algorithm with forgetting factor  $\lambda = 0.999999$  and  $\lambda = 0.98$  are shown in Fig. 7.6 and Fig. 7.7, respectively. It is realized that the DDASP based low-complexity algorithm provides better estimation error as compared to RLS filtering algorithm at  $\lambda = 0.999999$ .

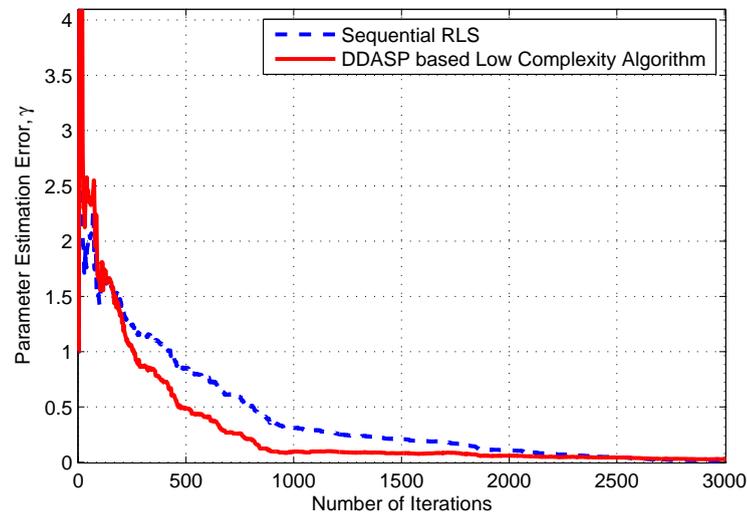


FIGURE 7.6: Coefficients estimation error versus number of iterations at  $30dB$  SNR

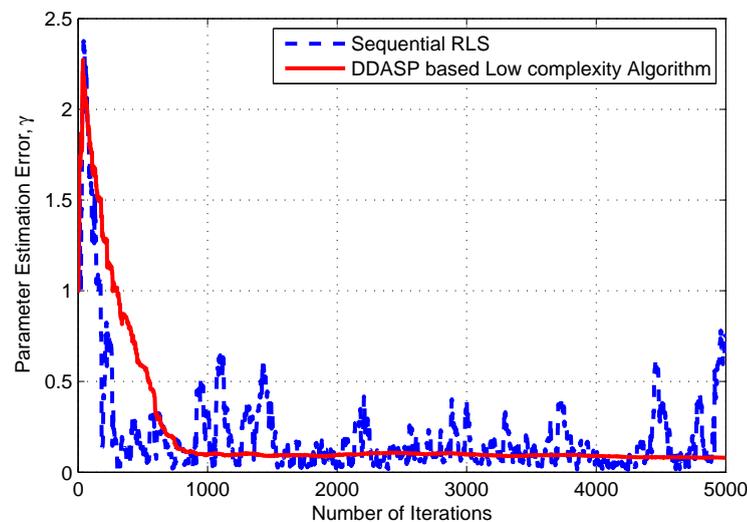


FIGURE 7.7: Coefficients estimation error versus number of iterations at  $20dB$  SNR

However, for  $\lambda = 0.98$ , the estimation error of RLS algorithm decreases as compared to the proposed DDASP based low-complexity algorithm but the final settling value is not be obtained. Moreover, the convergence performance of the proposed DDASP based low-complexity algorithm is compared with sequentially-operated RLS algorithm for  $\lambda = 0.999999$ . It is analyzed that the proposed DDASP based low-complexity algorithm provides better convergence performance than that of RLS filtering algorithm which is clearly envisioned in Table. 7.2 and

Table. 7.3, respectively. Furthermore, the processing time comparison among the DDASP based low-complexity and sequentially-operated algorithms on Arduino UNO platform is presented in Table. 7.6. It is realized that the combined processing time of the master node  $M_o$  and any of the slave nodes  $S$  is much lesser than those of both sequentially-operated standard RLS and sequentially-operated low-complexity estimation algorithm.

# Chapter 8

## Conclusions and Future

## Directions

This chapter presents conclusion of the dissertation in Section 8.1 and discusses the prospective future work in Section 8.2 on the basis of obtained results.

### 8.1 Conclusions

In this dissertation, a new fast low-complexity MIMO channel estimator has been proposed that performs well in multipath fading environment. The proposed algorithm uses the concept of Kalman gain to provide fast convergence performance while keeping its complexity lower than the standard RLS or Kalman filter. Obtained simulation results show that the proposed algorithm exhibits low probability of error compared to RVFF-RLS and RLS algorithms. Furthermore, the proposed algorithm provides  $(N + NL)^2$  lesser multiplication complexity and independency on forgetting factor  $\lambda$  than RVFF-RLS and RLS adaptive filtering algorithms.

Usually, in a wireless sensor network, the limited memory and energy constraint restrict a wireless sensor node to run the high definition adaptive filtering algorithm as a single unit. In this context, a novel low-complexity architecture for the parallelly distributed adaptive signal processing (PDASP) operation of inexpensive and

computationally-constrained small platforms has been proposed. The proposed architecture makes the inexpensive and computationally-constrained devices run computationally expensive procedures like complex adaptive Kalman and RLS algorithms cooperatively. The proposed architecture can even be run in the presence of time non-alignment. The complexity and processing time of the proposed PDASP scheme with RLS algorithm have been compared with those of sequentially operated Kalman and RLS algorithms. It has been observed that PDASP scheme exhibits much lesser computational complexity and processing time parallelly than the sequentially-operated Kalman and RLS algorithms. The proposed PDASP technique with non-aligned time indexes entails parallelly  $2(N + NL)^2$  multiplications and  $(N + NL)^2$  additions per iteration at maximum. Likewise, the proposed technique utilizes 95.83% and 82.29% lesser processing time than the sequential Kalman and MIMO RLS algorithms, respectively, for low doppler rate. Similarly, for high doppler rate, the proposed technique entails 94.12% and 77.28% decreased processing than sequentially-operated Kalman and MIMO RLS algorithms, respectively. In nutshell, processing time and parallel complexity of the proposed PDASP based MIMO RLS scheme have been observed to be much lesser than those of Kalman and RLS adaptive filtering algorithms, if operated sequentially on a single unit.

Although, the computational complexity provided by the PDASP architecture is much lesser than the sequential Kalman and MIMO RLS algorithms; however, the PDASP architecture still shows non-linear behavior in terms of computational cost. In this context, a new low-complexity architecture for the distributed diffusion based adaptive signal processing (DDASP) operation of computationally constrained small platforms has been proposed. In the DDASP architecture, the adaptive algorithm is diffused into the desired number of low-cost processing devices. Therefore, the number of processing nodes that are used to run the MIMO RLS algorithm on DDASP architecture is dependent upon the number of MIMO spatial streams as well as on the number of multipath components. It has been observed that, in case of no multipath components, the number of slave nodes  $(S_1, S_2 \cdots S_N)$  are equivalent to the number of MIMO spatial streams along with

one master node  $M$ . The master node is used to operate the other processing or slave nodes in the distributed network. It has been observed that the PDASP architecture exhibits lesser and linear computational complexity parallelly on each processing node involved as compared to the PDASP architecture and existing distributed architectures [74, 75]. The computational complexity of any of the slave nodes in the DDASP architecture provides  $3(N + NL) + 2N + 1$  multiplications and  $2(N + NL) + 2N - 2$  additions per iteration; likewise, the master node in the DDASP architecture entails  $N + NL + 1$  multiplications and  $2(N + NL)$  additions. Therefore, the combined computational complexity of a slave and master node provides  $4(N + NL) + 2N + 2$  multiplications and  $4(N + NL) + 2N - 2$  additions.

In addition to the application of MIMO channel estimation, parametric estimation of the auxiliary model for the retrieval of missing data can be considered another good example of cooperative distributed processing of small platforms for a common goal. It has been observed that the proposed PDASP architecture can perfectly run a group of nodes for the estimation of MIMO channel coefficients; however, as far as its implementation for missing data application is concerned, it shows unstable behavior due to its inherent working based on non-aligned time indexes. Since, missing data application needs perfect time alignment among various parts of the algorithm in order to reach a specific common goal. Therefore, sequentially-run algorithms or diffusion-oriented architectures become the only suitable choices. It has been realized that the implementation of the proposed DDASP technique using the proposed low-complexity algorithm provides much lesser and linear computational complexity parallelly on each node involved as compared to any sequentially-operated least square algorithm.

## 8.2 Future Directions

On the basis of the research carried out in the dissertation, the following future research directions may be furnished:

- The proposed low-complexity distributed architectures for energy-inefficient and computationally-constrained platforms can be extended to massive MIMO communication systems, in order to accommodate the use of such energy-inefficient and computationally-constrained devices in internet of things (IoT) networks. In recent years, massive MIMO antenna infrastructure has been proposed to facilitate high data rates over *mm*waves in prospective 5G communication systems which are supposed to provide communication framework for IoT solutions.
- The reduction of communication burden in DDASP architecture can be carried out in future to make sure the feasibility of distributed techniques in highly time-varying channel environments.
- The experimental validation of PDASP and DDASP architectures for quasi-stationary and time-varying channel environments with stability and communication burden analyses can be carried out in future. Moreover, the distributed solutions related to master-slave node systems can also be deployed to highly complex systems, e.g. distributed estimation etc.
- The reduction of training sequence in PDASP architecture can be carried out in future by reducing communication overload to make sure the fast convergence of the adaptive filtering algorithm.
- The task allocation and time thresholding regarding distributed network or individual nodes can further be analyzed by implementing the distributed architectures on low cost computationally-constrained platforms.

# Bibliography

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, pp. 393–422, 2002.
- [2] D. Culler, D. Estrin, and M. Srivastava, “Overview of sensor networks,” *Computer*, vol. 37, no. 8, pp. 41–49, 2004.
- [3] I. Sheikh, N. M. Khan, and P. Rapajic, “Kalman filter based channel estimation in OFDM multi-user signal-input-multi-output system,” in *Proc. IEEE International Symposium on Spread Spectrum Techniques and Applications*, Sydney, NSW, Australia, 2004, pp. 802–806.
- [4] A. H. Syed and T. Kailath, “A state-space approach to adaptive rls filtering,” *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18–60, 1994.
- [5] R. Merched and A. H. Syed, “Extended fast fixed-order rls adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3015–3031, 2001.
- [6] E. Unsal, M. Milli, and Y. Cebi, “Low cost wireless sensor networks for environment monitoring,” *The Online Journal of Science and Technology*, vol. 6, no. 2, pp. 61–67, 2016.
- [7] E. Serpedin, H. Li, A. Dogandzic, H. Dai, and P. Cotae, “Distributed signal processing techniques for wireless sensor networks,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 540176, 2 pages, doi:10.1155/2008/540176, 2008.

- 
- [8] N. M. Khan, "Modeling and characterization of multipath fading channels in cellular mobile communication systems," *Ph. D. Dissertation, University of New South Wales*, 2006.
- [9] J. R. Hampton, *Introduction to MIMO communications*. Cambridge University Press, 2013.
- [10] P. Konstantinos and M. Chitre, "Robust equalization of mobile underwater acoustic channels," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 4, pp. 775–784, 2015.
- [11] B. Zhang, L. Zhang, L. Zhang, and F. Karray, "Retinal vessel extraction by matched filter with first-order derivative of gaussian," *Computers in Biology and Medicine*, vol. 40, no. 4, pp. 438–445, 2010.
- [12] S. Bidyadhar and R. Pradhan, "An adaptive predictive error filter based maximum power point tracking algorithm for a photovoltaic system," *The Journal of Engineering*, vol. 1, no. 1, 2016.
- [13] B. Farhang-Boroujeny, *Adaptive filters: theory and applications*. John Wiley & Sons, 2013.
- [14] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.
- [15] M. Kamenetsky and B. Widrow, "A variable leaky lms adaptive algorithm," in *Proc. IEEE Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, PacificGrove, California, vol. 1, 2004, pp. 125–128.
- [16] S. A. Hadei and P. Azmis, "A novel adaptive channel equalization method using variable step size partial rank algorithm," in *Proc. IEEE Sixth Advanced International Conference on Telecommunications (AICT)*, Barcelona, Spain, 2010, pp. 201–206.
- [17] Y. Li, Z. Jin, and Y. Wang, "Adaptive channel estimation based on an improved norm-constrained set-membership normalized least mean square algorithm," *Wireless Communications and Mobile Computing*, vol. 2017, 2017.

- 
- [18] D. Tufts, L. Griffiths, B. Widrow, J. Glover, J. McCool, and J. Treichler, "Adaptive line enhancement and spectrum analysis," in *Proceedings of the IEEE*, vol. 65, no. 1, 1977, pp. 169–173.
- [19] J. M. Gorriz, J. Ramirez, S. Cruces-Alvarez, C. G. Puntonet, E. W. Lang, and D. Erdogmus, "A novel lms algorithm applied to adaptive noise cancellation," *IEEE Signal Processing Letters*, vol. 16, no. 1, pp. 34–37, 2009.
- [20] M. S. Razzaq and N. M. Khan, "performance comparison of adaptive beamforming algorithms for smart antenna systems," *World Applied Sciences Journal*, vol. 11, no. 7, pp. 775–785, 2010.
- [21] Z. Peric and J. Nikolic, "An adaptive waveform coding algorithm and its application in speech coding," *Digital Signal Processing*, vol. 22, no. 1, pp. 199–209, 2012.
- [22] A. M. Tulino, A. Lozano, and S. Verdu, "Mimo capacity with channel state information at the transmitter," in *Proc. IEEE Eighth International Symposium on Spread Spectrum Techniques and Applications*, Sydney, Australia, 2004, pp. 22–26.
- [23] F. Ding and J. Ding, "Least-squares parameter estimation for systems with irregularly missing data," *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 7, pp. 540–553, 2010.
- [24] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Consensus algorithm design for distributed estimation," in *Proceedings of the Workshop on Network Control Systems Tolerant to Faults*, 2007.
- [25] A. Rastegarnia, M. A. Tinati, and A. Khalili, "A diffusion least mean square algorithm for distributed estimation over sensor networks," *International Journal of Electrical, Computer and Systems Engineering*, vol. 2, no. 1, pp. 15–19, 2008.

- [26] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.
- [27] R. Prasad and A. Mihovska, *New Horizons in Mobile and Wireless Communications: Radio Interfaces*. Artech House, Inc, 2009.
- [28] H. El Gamal and A. R. Hammons, "On the design of algebraic space-time codes for mimo block-fading channels." *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 151–163, 2003.
- [29] M. R. Amin and S. D. Trapasiya, "Space time coding scheme for mimo system-literature survey," *Procedia Engineering*, vol. 38, pp. 3509–3517, 2012.
- [30] S. K. Mohammed, A. Zaki, A. Chockalingam, and B. S. Rajan, "High-rate space time coded large-mimo systems: Low-complexity detection and channel estimation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 6, pp. 958–974, 2009.
- [31] C. A. Balanis, *Advanced engineering electromagnetics*. John Wiley and Sons, 2012, vol. 111.
- [32] H. Dokhanchi and A. Falahati, "Adaptive mimo ls estimation technique via mse criterion over a maekov channel model," in *Proc. IEEE Symposium on Computers and Communications (ISCC)*, Riccione, Italy, 2010, pp. 151–154.
- [33] M. Kashoob and Y. Zakharov, "Selective detection with adaptive channel estimation for mimo ofdm," in *Proc. IEEE Signal Processing Workshop on Sensor Array and Multichannel(SAM)*, RiodeJanerio, Brazil, 2016, pp. 1–5.
- [34] X. L. S. L. Gui, G and F. Adachi, "Adaptive mimo channel estimation using sparse variable step-size nlms algorithms," in *Proc. IEEE International Conference on Communication Systems (ICCS)*, Macau, China. IEEE, 2014, pp. 605–609.

- [35] X. Liu, J. Wang, Z. Li, and J. Si, "Soft-output mmse mimo detector under different channel estimation models," *IET Communications*, vol. 11, no. 2, pp. 192–197, 2017.
- [36] C. Schuldt, F. Lindstrom, and I. Claesson, "Low-complexity adaptive filtering implementation for acoustic echo cancellation," in *Proc. IEEE TENCON 2006: 2006 IEEE Region 10 Conference: HongKong, China, 2006*, pp. 1–4.
- [37] R. A. Soni, K. A. Gallivan, and W. K. Jenkins, "Low-complexity data reusing methods in adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 394–405, 2004.
- [38] C. Schuldt, F. Lindstrom, and I. Claesson, "A low-complexity delayless selective subband adaptive filtering algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 12, pp. 5840–5850, 2008.
- [39] L. Tong, "Blind sequence estimation," *IEEE Transactions on communications*, vol. 43, no. 12, pp. 2986–2994, 1995.
- [40] A. Vosoughi and A. Scaglione, "Channel estimation for precoded mimo systems," in *Proc. IEEE Workshop on Statistical Signal Processing*, St.Louis, Mo, USA, 2003, pp. 442–445.
- [41] S.-C. Chan and Y.-X. Zou, "A recursive least m-estimation algorithm for robust adaptive filtering in impulsive noise: fast algorithm and convergence performance analysis," *IEEE Signal Processing Letters*, vol. 52, no. 4, pp. 975–991, 2004.
- [42] M. Z. A. Bhotto and A. Autoniou, "Robust recursive least-squares adaptive-filtering algorithm for impulsive-noise environments," *IEEE Signal Processing Letters*, vol. 18, no. 3, pp. 185–188, 2011.
- [43] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach*. Cambridge university press, United Kingdom, 2007.
- [44] B. Kovacevic, Z. Banjac, and M. Milosavljevic, *Adaptive digital filters*. Springer Science and Business Media, 2013.

- [45] A. Benveniste and M. Basseville, "Detection of abrupt changes in signals and dynamical systems: some statistical aspects. analysis and optimization of systems," pp. 143–155, 1984.
- [46] S. Song, "Self-tuning adaptive algorithm and applications," *Ph. D. Dissertation, Seoul National University*, pp. 15–25, 2003.
- [47] E. Karami and M. Shiva, "Decision-directed recursive least squares mimo channels tracking," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, no. 1, pp. 1–10, 2006.
- [48] R. Arablouei and K. Dogancay, "Modified rls algorithm with enhanced tracking capability for mimo channel estimation," *Electronics letters*, vol. 47, no. 19, pp. 1101–1103, 2011.
- [49] T. K. Akino, "Optimum-weighted rls channel estimation for rapid fading mimo channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11.
- [50] B. Kovavcevi, Z. Banjac, and I. K. Kovavcevic, "Robust adaptive filtering using recursive weighted least squares with combined scale and variable forgetting factors," *EURASIP journal on advances in signal processing*, vol. 2016, no. 1, p. 37, 2016.
- [51] Y. Yapici and A. O. Yilmaz, "Low-complexity iterative channel estimation and tracking for time-varying multi-antenna systems," in *Proc. IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications 2009*, Tokyo, Japan, 2009, pp. 1317–1321.
- [52] K. E. G. Moshavi, Shimon and D. L. Schilling, "Multistage linear receivers for ds-cdma systems," *International Journal of Wireless Information Networks*, vol. 3, no. 1, pp. 1–17, 1996.
- [53] M. L. Honig and W. Xiao, "Performance of reduced-rank linear interference suppression," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1928–1946, 2001.

- [54] G. M. Sessler and F. K. Jondral, “Low complexity polynomial expansion multiuser detector for CDMA systems,” *IEEE Transactions on Vehicular Technology*, vol. 54, no. 4, pp. 1379–1391, 2005.
- [55] D. M. Hoydis, Jakob and M. Kobayashi, “Asymptotic moments for interference mitigation in correlated fading channels,” in *Proc. IEEE International Symposium on Information Theory Proceedings (ISIT)*, SaintPeterburg, Russia, 2011, pp. 2796–2800.
- [56] S. Puntanen and G. P. H. Syuan, *The Schur Complement and Its Applications: Numerical Methods and Algorithms*. Springer-Verlag, New York, 2005.
- [57] R. R. Muller, L. Cottatellucci, and V. Mikko, “Blind pilot decontamination,” *IEEE Journal of Selected Topics Signal Processing*, vol. 8, no. 5.
- [58] S. Eberli, D. Cescato, and W. Fichtner, “Divide-and-conquer matrix inversion for linear mmse detection in SDRMIMO receivers,” in *Proc. IEEE 26th Norchip Conference*, Tallinn, EE, 2008, pp. 162–167.
- [59] S.-H. Leung and C. So, “Gradient-based variable forgetting factor rls algorithm in time-varying environments,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3141–3150, 2005.
- [60] L. Xiao, S. Liu, and D. Yang, “A low-complexity block diagonalization algorithm for mu-mimo two-way relay systems with complex lattice reduction,” *Internation Journal of Distributed Sensor Networks*, vol. 11, no. 6, 2015.
- [61] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, “Near-maximum likelihood detection of mimo systems using mmse-based lattice reduction,” in *Proc. IEEE Interantional Conference on Communications*, vol. 2, 2004, pp. 798–802.
- [62] A. O. A. Noor, S. A. Samad, and A. Hussain, “A review of advances in subband adaptive filtering,” *World Applied Sciences Journal*, vol. 21, no. 1, pp. 113–124, 2013.

- [63] A. Bertrand and M. Moonen, "Distributed adaptive node-specific mmse signal estimation in sensor networks with a tree topology," in *Proc. IEEE 17th European Conference on Signal Processing*, Glasgow, UK, 2009, pp. 794–798.
- [64] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, 2013.
- [65] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.
- [66] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Signal Processing Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [67] D. Estrin, L. Girod, G. Pottie, and m. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, New York, USA, vol. 4, 2001, pp. 2033–2036.
- [68] L. Fan, L. Jia, R. Tao, and Y. Wang, "Distributed bias-compensated normalized least-mean squares algorithms with noisy input," *Science China Information Sciences*, vol. 61, no. 11, 2018.
- [69] N. Bogdanovie, J. Plata-Chaves, and K. Berberidis, "Consensus problems in networks of agents with switching topology and time delays," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASP)*, Florence, Italy, 2014, pp. 7223–7227.
- [70] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed lms for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

- [71] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 151–164, 2010.
- [72] K. Dimple, D. K. Kotary, and S. J. Nanda, "An incremental rls for distributed parameter estimation of IIR systems present in computing nodes of a wireless sensor network," *Procedia Computer Science*, vol. 115, pp. 699–706, 2017.
- [73] J. Plata-Chaves, N. Bogdanovic, and K. Berberidis, "Distributed incremental-based rls for node-specific parameter estimation over adaptive networks," in *Proc. 21st IEEE European Signal processing Conference*, Marrakech, Morocco, 2013, pp. 1–5.
- [74] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. IEEE Fortieth Asilomar conference on Signals, systems and Computers (ACSSC)*, PacificGrove, CA, USA, 2006, pp. 233–237.
- [75] J. Jian, J. Lijuan, T. Ran, and W. Yue, "Distributed incremental bias-compensated rls estimation over multi-agent networks," *Science China Information Sciences*, vol. 60, no. 3, pp. 1–15, 2017.
- [76] W. B. Lopes and C. G. Lopes, "Incremental combination of rls and lms adaptive filters in nonstationary scenarios," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 5676–5680.
- [77] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4583–4588, 2009.
- [78] G. Mateos and G. B. Georgios, "Distributed recursive least-squares: Stability and performance analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3740–3754, 2012.

- [79] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.
- [80] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.
- [81] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, 2008.
- [82] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010.
- [83] Y. Yu, H. Zhao, R. de Lamare, Y. Zakharov, and L. Lu, "Robust distributed diffusion recursive least squares algorithms with side information for adaptive networks," *IEEE Transactions on Signal Processing*, vol. 67, no. 6, pp. 1566–1581, 2019.
- [84] S. Kajan, I. Sekaj, and M. Oravec, "The use of matlab parallel computing toolbox for genetic algorithm-based mimo controller design," in *Proc. 17th International Conference on Process Control*, StrbskePles, Slovakia, vol. 9, 2009, pp. 9–12.
- [85] E. Starkloff, "Designing a parallel, distributed test system," in *In Proc. 2000 IEEE AUTOTESTCON*, Anaheim, CA, 2000, pp. 564–567.
- [86] S. W. Keckler, W. J. Dally, B. Khailnay, M. Garland, and D. Glasco, "Gpus and the future of parallel computing," *IEEE Micro*, vol. 31, no. 5, pp. 7–17, 2011.
- [87] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron, "A performance study of general-purpose applications on graphics processors

- using cuda,” *Journal of Parallel and Distributed Computing*, vol. 68, no. 10, pp. 1370–1380, 2008.
- [88] W. C. Jakes and D. C. Cox, *Microwave mobile Communications*. Wiley-IEEE Press, 1994.
- [89] P. Bello, “Characterization of randomly time-variant linear channels,” *IEEE transactions on Communications Systems*, vol. 11, no. 4, pp. 360–393, 1963.
- [90] M. K. Tsatsanis, G. B. Giannakis, and G. Zhou, “Estimation and equalization of fading channels with random coefficients,” *Signal Processing*, vol. 53, no. 2-3, pp. 1093–1096, 1996.
- [91] O. E. Weikert, “Blinde demodulation in mimo-ubertragungssystemen,” *Ph.D. dissertation, Helmut Schmidt University / University of the Federal Armed Forces Hamburg, dissertation.de, ISBN 978-3-86624-273-9*, 2007.
- [92] Z. Liu, X. Ma, and G. B. Giannakis, “Space-time coding and kalman filtering for time-selective fading channels,” *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 183–186, 2002.
- [93] 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.
- [94] K. Yano and S. Yoshida, “Cdma non-linear interference canceller with multi-beam reception,” in *Proc. IEEE Fifth International Conference on Information, Communications and Signal Processing*, Bangkok, Thailand, 2005, pp. 6–10.
- [95] F. Ding, Y. Wang, and J. Ding, “Recursive least squares parameter identification for systems with colored noise using the filtering technique and the auxiliary model,” *Digital Signal Processing*, vol. 37, pp. 100–108, 2015.

- 
- [96] J. Ding, C. Fan, and J. Lin, "Auxiliary model based parameter estimation for dual-rate output error systems with colored noise," *Applied Mathematical Modeling*, vol. 37, no. 6, pp. 4051–4058, 2013.
- [97] Y. Hu, "Iterative and recursive least squares estimation algorithms for moving average systems," *Simulation Modeling Practice and Theory*, vol. 34, pp. 12–19, 2013.
- [98] K. H. Choi, W.-S. Ra, S.-Y. Park, and J. B. Park, "Robust least squares approach to passive target localization using ultrasonic receiver array," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 1993–2002, 2014.
- [99] D. Wang, "Least squares-based recursive and iterative estimation for output error moving average systems using data filtering," *IET Control Theory and Applications*, vol. 5, no. 14, pp. 1648–1657, 2011.
- [100] X. Wang and F. Ding, "Recursive parameter and state estimation for an input nonlinear state space system using the hierarchical identification principle," *Signal Processing*, vol. 117, pp. 208–218, 2015.
- [101] F. Ding, X. Liu, and Y. Gu, "An auxiliary model based least squares algorithm for a dual-rate state space system with time-delay using the data filtering," *Journal of the Franklin Institute*, vol. 353, no. 2, pp. 398–408, 2016.