CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Low Communication-Parallel Distributed Adaptive Signal Processing (LC-PDASP) Architecture for Processing-Inefficient Low Cost Platforms

by

Ghalib Hussain

A thesis submitted in partial fulfillment for the degree of Master of Science

in the

Faculty of Engineering
Department of Electrical Engineering

2019

To my parents and beloved family

CERTIFICATE OF APPROVAL

Low Communication-Parallel Distributed Adaptive Signal Processing (LC-PDASP) Architecture for Processing-Inefficient Low Cost Platforms

by

Ghalib Hussain

(MEE163006)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Junaid Nawaz	CIIT, Islamabad
(b)	Internal Examiner	Dr. Imtiaz Ahmed Taj	CUST, Islamabad
(c)	Supervisor	Dr. Noor Muhammad Khan	CUST, Islamabad

Dr. Noor Muhammad Khan
Thesis Supervisor
April, 2019

Dr. Noor Muhammad Khan
Head
Dept. of Electrical Engineering
April, 2019

Dr. Imtiaz Ahmad Taj
Dean
Faculty of Engineering
April, 2019

Author's Declaration

I, **Ghalib Hussain** hereby state that my MS thesis titled "**Low Communication-Parallel Distributed Adaptive Signal Processing (LC-PDASP) Architecture for Processing-Inefficient Low Cost 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/abroad.

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

(Ghalib Hussain)

Registration No:MEE163006

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**Low Communication-Parallel Distributed Adaptive Signal Processing (LC-PDASP) Architecture for Processing-Inefficient Low Cost Platforms**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been dully 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 MS Degree, the University reserves the right to withdraw/revoke my MS 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 work.

(Ghalib Hussain)

Registration No: MEE163006

Acknowledgements

First of all I would like to thank ALMIGHTY ALLAH for giving me the nerve, health and mental ability during my research work. I am grateful to Prof. Dr. Noor Muhammad Khan for believing in me to work under his supervision. He steered me in the direction which was the best for me. I would also acknowledge Mr. Hasan Raza who's sincere suggestions always proved to be fruitful in the way of completing my thesis. I am thankful to all faculty members of Electrical Engineering department of the Capital University of Science and Technology for giving me the knowledge that proved to be helpful. I am also thankful to all the members of Acme Center for Research in Wireless Communications (ARWiC), Group. It would be an ultimate pleasure to acknowledge my father, who taught me the patience, devotion and commitment, and my mother who is symbol of ultimate affection.

(Ghalib Hussain)

Registration No:MEE163006

Abstract

In order to reduce the extensive processing of the centralized processor, distributed adaptive techniques are used that provide a cooperative solution for high definition adaptive algorithms. The distributed adaptive filtering techniques can be used in those applications, that utilize processing incapable platforms, such as military surveillance, industry, transportation instrumentation, environmental parameters estimation and agriculture development. In this thesis, a Low Communication Parallel Distributed Adaptive Signal Processing (LC-PDASP) architecture for a group of computationally-incapable and inexpensive small platforms is introduced. The proposed architecture is capable of running computationally-expensive procedures like complex adaptive algorithms parallely with minimally low communication overhead. The RLS algorithm with the application of MIMO channel estimation is deployed on the proposed architecture. complexity and communication burden of the proposed LC-PDASP architecture are compared with the complexity and communication burden of conventional PDASP architecture. The comparative analysis shows that the proposed LC-PDASP architecture provides low computational complexity and exhibits minimally reduced communication burden per iteration with an improvement of 85% as compared to the conventional PDASP architecture.

Contents

List of Figures

List of Tables

Abbreviations

WSNs	Wireless Sensor Networks
DSP	Digital Signal Processor
IoT	Internet of Things
CPU	Central Processing Unit
GPU	Tensor Processing Unit
TPU	Tensor Processing Unit
ISI	Inter-Symbol Interference
MIMO	Multiple-Input and Multiple-Output
CSI	Channel State Information
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
MSE	Mean Square Error
MMSE	Minimum Mean Square Error
LMS	Least Mean Square
NLMS	Normalized Least Mean Square
VSLMS	Variable Step Size Least Mean Square
RLS	Recursive Least Square
LC-RLS	Low Communication-Recursive Least Square
CLR	Complex Lattice Reduction
PDASP	Parallel Distributive Adaptive Signal Processing
LC-PDASP	Low Communication-Parallel Distributive Adaptive Signal Processing
BPSK	Binary Phase Shift Keying
LoS	Line of Sight

Chapter 1

Introduction

In the domain of science and technology, role of adaptive filtering techniques is vital. These techniques have grabbed researchers' attention in the past few decades. As for as the implementations of these filter on low cost energy-constrained wireless sensor nodes is concerned, they are usually assumed to be incapable of running these adaptive filtering techniques because of high memory requirement and fast processing. However, recently some techniques were proposed by the researchers to run these highly complex algorithms on low cost platforms through operating multiple nodes cooperatively [1]. Nevertheless, these parallel operations need significant overheads which make them difficult to implement. This chapter presents the introduction about the wireless sensore nodes (WSNs) and their effective utilization towards the distributed adaptive signal processing techniques

1.1 Low Cost Processing Inefficient Platforms: Wireless Sensor Nodes

1.1.1 History of Wireless Sensor Networks

The research in (WSNs) may have began in 1980s, from the experiments conducted by a research group in the United States (US) for military purpose. Furthermore,

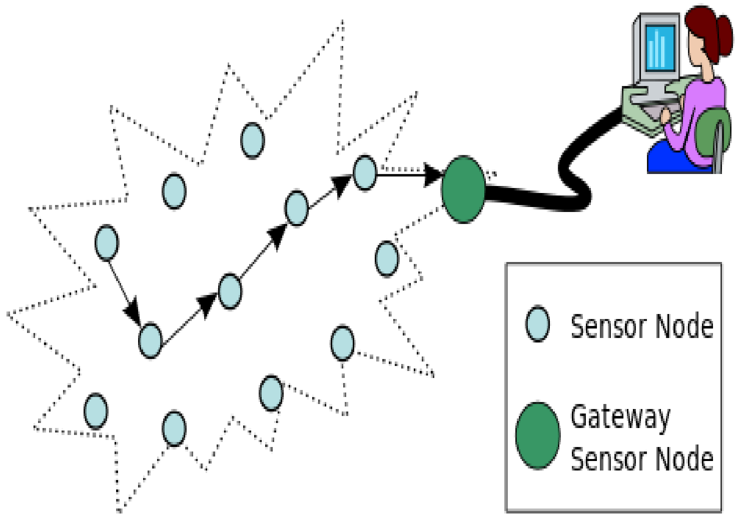

Figure 1.1: A typical Wireless Sensor Network

in the late 90s, a new upsurge was examined in the field wireless sensor network which made possible to manufacture miniature nodes. In the present era of globalization, internet of things (IoT), a modified form of WSNs is the utmost requirement in our daily life and may be used in different applications like monitoring of crops in fields, recording of physical environmental conditions, watching mechanical stress in bridges and beams after earthquake, surveillance of critical patients, etc [2].

Nevertheless, the present advancement in technology and popularity of these tiny nodes not only push the WSNs to apply on broader scale in various fields but also draw the attention of researchers to bring further enhancement in various subfields of WSNs. Wireless sensor nodes are the indispensable mechanism modules of the wireless sensor networks. WSNs consists of tiny, low power, low cost and auto-configure nodes that collaboratively exchange information with a central node, i.e. sink [3]. WSNs composed of large number of nodes according to the desired application as shown in FIGURE 1.1. Wireless sensor nodes are playing vital role in field of wireless communication system. A WSN consists of small, low power, low cost and auto-configure nodes that collaboratively exchange information with a master node.

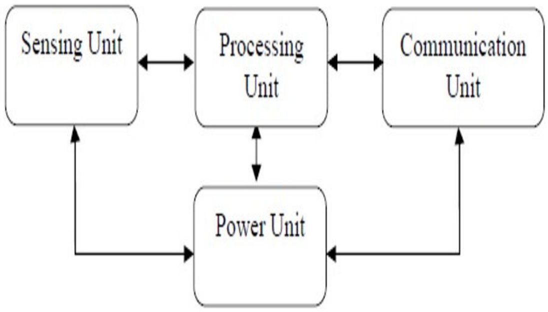

FIGURE 1.2: Block Diagram Of a Sensor Node

1.1.2 Architecture of a Wireless Sensor Node

A typical wireless sensor network consists of processing unit, communication unit, sensor unit and power unit [4]. The block diagram of sensor node is shown in FIGURE. 1.2. A brief overview about each unit is as follows:

1.1.2.1 Processing Unit

In wireless senor node, the role of processing unit is foremost as it processes the data which is dependent upon the type of processor. The performance of any wireless sensor node is measured by its processor speed. Processing unit consist of different types of processors according to the nature of work. The most excessively used processors are DSPs, ATMEGA-128, ASICS, FPGA and ATMEGA-16.

1.1.2.2 Power Unit

Power unit is considered as a key component in a wireless senor node which supplies energy to other units of the node. The cost of any wireless sensor node is measured by its total energy consumption. During exchange of data within the network, some nodes may die out due to inadequate power backup which may decrease the proficiency of the wireless sensor nodes. There are different energy sources used to support power unit. Power unit is supported by solar cell, batteries and capacitors. However, for continuous supply of energy is necessary which can be cultivated through solar cells, etc [5].

1.1.2.3 Sensor Unit

Sensor unit consists of different sensors as per desired application. It includes vibration sensor, light sensor and temperature sensor, etc. Sensors are used to measure external parameters of the environment. Various sensors take analog signals as input from the external environment and converted them to digital signals before sending to the processing unit.

1.1.2.4 Communication Unit

The key component of the transceiver is antenna which is used to transmit and received small packets of data in form of signals. The transceiver of a typical node is usually operate on 2.4GH industrial scientific and medical (ISM) band [6].

1.1.3 Internet of Things (IoT)

IoT has become the topic of great interest these days. This is an idea to enhance the life quality by interconnecting small devices, smart technologies and applications together. IoT also establishes the connection among edge devices to different servers via internet which provides a role of distributed processing unit in the IoT platform. Moreover, the performance of IoT platform depends on its components which include sensing, communication, identification, computational, cloud and lastly are the services and applications. Whereas, the sensing component is responsible for the sensing physical or environmental conditions like temperature and pressure, etc. Communication and identification components represent the communication technologies and unique ID of each IoT object, respectively. Likewise, computational and service components are responsible for processing of data and represent all kind services provided by the IoT platform, respectively [7]. All these components work together to get reliable and distributed data from IoT platform.

1.1.4 Graphical Processing Unit (GPU)

Graphical Processing Unit (GPU) is an electronic chip that can perform rapid mathematical operations and process data with very high speed due to its parallel architecture. Processing speed of GPU is greater than Central Processing Unit (CPU) this is because of its ability to perform different tasks at the same time and this quality makes the GPU palmary. GPU is one of the most promising platform for parallel applications. However, the Clock speed of GPU is low as compared to CPU. Moreover, the GPU is a power hungry device which utilizes more power as compared to the bunch of wireless sensor nodes.. Calculations done by GPU are applied in many fields such as, sound processing, physical simulation, adaptive radiation therapy, computer vision, data mining and bioinformatics etc [8].

1.1.5 Tensor Processing Unit (TPU)

In 2011, google Professionals realized they had serious problem about deep learning network with increased computational demand. CPUs were introduced to handle this problem but, CPUs were not able to handle large number of computational tasks at a time, this was major limitation of CPUs. On the other hand GPUs can carry out wide range of tasks at the same time. GPUs are best for deep learning applications because deep learning network performs millions of computations at a same time. But google requirements were still high they need more efficient processing chip so they decided to build their own processing chip. Google developed an Al accelerator integrated circuit which is used by the help of its tensor flow Al frame work and this chip has given the name of tensor processing unit (TPU). Tensor flow, is google open source software library for its internal use. Tensor flow name is derived from the operation of neural network that it performs on multidimensional data arrays, these arrays are called tensors. Tensor processing units are specially designed for machine learning they may not use for other work load, thereby limiting the flexibility they have to shift work around.

1.1.6 Distributed Wireless Sensor Nodes

The demand of improving processing efficiencies is increasing every day. Central Processing unit (CPU), communication hardware and power unit are the key components that are used to measure the performance of wireless sensor node. The main challenge in the development of wireless sensor node is low processing capability and better quality of service due to limit power resources. So, powerful combination of distributive wireless sensor nodes is used for complex processing. Overall cost is justified by the cost of the single node. Zigbee and android BT are excessively used low cost wireless sensor nodes for communication [9, 10].

1.2 Adaptive Filtering

Filter is just like a black box which processes the input signal and generates a modified output signal. In linear filter, output is always linear function of its input signal, whereas, output behavior of nonlinear filter is comparatively different from linear filter. Hence, complex mathematics is involved to retain the output form of nonlinear signal [11].

1.2.1 Linear Adaptive Filter

A filter detects a desired signal adaptively which is coming from the unknown channel environment is known as adaptive filter. Adaptive filter continuously track the changes which are happen in the channel and updating its filter coefficients values accordingly [12]. The filter weight adjustment is a primary task of any adaptive filter. Adaptive filter always tries to reduce the difference between the input and the output signal and refine its transfer function. Adaptive filters are used in many applications, e.g. channel state information (CSI), channel equalization, adaptive noise cancelation, and adaptive beam forming. Furthermore, Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) are the two main categories which are used to model the adaptive filter. However, FIR filter has a preference over IIR due to its casuality and lesser computational complexity requirements [11].

1.2.2 Mean Squared Error (MSE) Criterion

A mean squared error (MSE) criterion plays a vital role for the analysis of any adaptive filtering algorithm. MSE is simply refers to the mean squared difference between the desired signal and the estimated signal [13]. FIR filter strictly follows the MSE criterion. The error signal e_n which is the difference of desired signal d_n and measured signal y_n is an important parameter used to update the filter coefficients can be written as

$$e_n = d_n - y_n \tag{1.1}$$

The MSE criterion according to error function can be expressed as

$$MSE = \mathrm{E}[|e_n|^2] \tag{1.2}$$

Here, E represents the expectation operator. The MES criterion has a great importance while designing of any adaptive filtering algorithm.

1.2.3 Gradient Based Approach

The gradient based methods have received intensive interest in research and scientific applications [14]. The most important and well used gradient based algorithms are least mean square (LMS), normalized LMS (NLMS) and variable step-size LMS (VSLMS). A brief description about each gradient based approach is as follow.

1.2.3.1 Least Mean Square (LMS) Algorithm

A well known LMS adaptive algorithm is most widely used in the world. The convergence performance of LMS algorithm is dependent on the step size parameter and its selection is one of the promising task [15]. The update filter weight equation of LMS algorithm can be expressed as

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu e_n \mathbf{y}_n \tag{1.3}$$

where μ is the step size parameter, $\mathbf{w}$ represents the filter weights and $\mathbf{y}_n$ is the measure output signal[16].

1.2.3.2 Normalized LMS (NLMS) Algorithm

In NLMS adaptive algorithm, a normalize step size parameter is used to update the filter coefficients. The normalize step size parameter depends upon the signal variations. However, for highly time varying channel conditions, the NLMS algorithm rules out its convergence performance which is a major drawback of this algorithm [17, 18]. In NLMS step size parameter is adjusted according to input signal power variation.

1.2.3.3 Variable Step-Size (VSSLMS) Algorithm

VSSLMS algorithm utilizes two step size parameters which is suitable for slow and time varying channel environments. For highly time varying channel conditions, the larger step size parameter is used and vice versa. The stability and convergence performance of the VSSLMS algorithm is dependent upon the proper setting of step size parameters; however, their setting are considered to be one of the challenging task [19]. So in fast time varying channel conditions VSSLMS may lost stability and computational cost of VSSLMS also increase just like LMS algorithm.

1.3 Least Square Based Approach

The least square based approach recursively finds the optimum filter coefficients and it minimizes the weighted cost function. This method is able to track the fast variations which are happen into the signal process. The least square based approach has relatively fast tracking performance as compared to gradient based approach. However, the price of these benefits is on behalf of complex computational complexity of the recursive based algorithms [20].

1.3.1 Recursive Least Square (RLS) Algorithm

Recursive Least Square (RLS) is the most popular adaptive filtering algorithm. The RLS algorithm is suitable for non-stationary channel environments. The RLS algorithm provides fast rate of convergence performance according to the time variations in the process. The RLS algorithm utilizes forgetting factor λ to reduce the influence of past samples to error covariance matrix and it makes the filter more sensitive to the present and recent samples. This algorithm is very suitable in many applications, such as channel equalization, speech enhancement, channel estimation, radar signal processing and echo cancelation. However, due to complex computational cost, RLS algorithm is considered to be ruled out for real time implementation on computationally-incapable wireless sensor nodes [21].

1.4 MIMO System and MIMO Channel Estimation

In MIMO communication system, multiple antennas are used at both the transmitter and receive side to improve the link capacity or spectral efficiency. MIMO system has become a key element of many communication standards, such as IEEE 802.22ac (Wi-Fi), IEEE 802.11n (Wi-Fi), WiMAX (4G), HSPA^{+} (3G) and Long

Term Evolution (LTE 4G). The capacity in MIMO communication system is proportional to the number of antenna elements which are used in the communication link. MIMO systems are also used to increase the data throughput without an increase in operational bandwidth. However, this increased data throughput is based on the getting of true information of non-coherent Rayleigh fading channel. The non-coherency of the MIMO channel is due to the Inter-Symbol Interference (ISI) [22] that may be either parallel or based on mutipath components which is discussed in Chapter 3. Therefore, adaptive filters are only the choice which are used to estimate the non-coherent statistics of the MIMO channel. Moreover, the computational complexity of the adaptive algorithm is directly dependent on the parallel interference as well as on multipath components that provides a critical impact on the overall execution time of the adaptive algorithm [23–25].

1.5 Parallel Distributed Adaptive Filtering Architectures

Incremental strategy and Parallel Distributed Adaptive Signal Processing (PDASP) are the two architectures which are used to run the high definition adaptive filtering algorithm in parallel fashion. A brief overview about each architecture is as follows:

1.5.1 Incremental Strategy

In incremental strategy, all the network nodes use the cyclic pattern to find the estimate of the unknown coefficients with minimum power requirements. The model diagram of incremental strategy is shown in FIGURE. 1.3. Furthermore, in the incremental network, each individual node performs local computations and then share the updated information towards the adjacent node [26]. The number of N nodes in the incremental network is dependent upon the M iterations which make possible of the convergence of the adaptive algorithm. As compared to centralized solution, the incremental approach reduces the power requirements and improves the autonomy of the network. However, in case of MIMO channel estimation scenario, the incremental based network technique facing high communication burden in sense of transferring of complex MIMO channel matrices among the nodes of the network. Moreover, each node in the incremental network is being free for $M-1$ iterations, where M is the total number of iterations required for the complete convergence of the adaptive filtering algorithm [27–29].

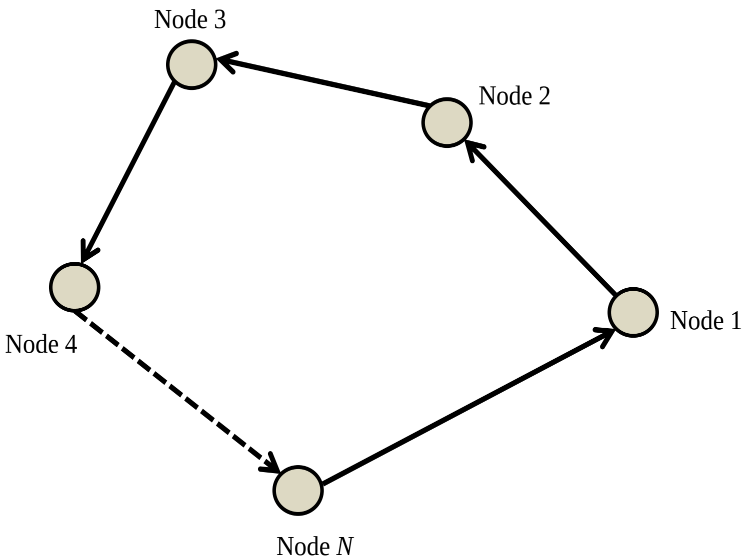

FIGURE 1.3: Model diagram of incremental strategy

1.5.2 Parallel Distributed Adaptive Signal Processing (PDASP) Architecture

Parallel Distributed Adaptive Signal Processing (PDASP) architecture utilizes only four nodes for the execution of the adaptive filtering algorithm. PDASP architecture is able to run RLS algorithm in parallel fashion. The flow diagram of PDASP architecture is shown in FIGURE. 1.4. The PDASP scheme exhibits much lesser computational complexity and processing time parallely than the sequentially-operated algorithms and it also provides improvement in mean squared error as compared to sequential operated RLS algorithm, however; the

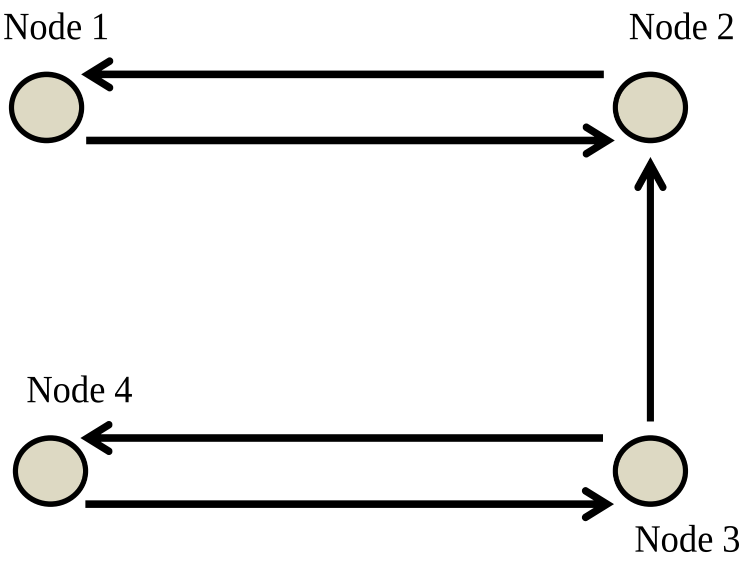

FIGURE 1.4: Model diagram of PDASP architecture

communication burden among the participating nodes is very high which makes a very crucial impact on the execution time of the algorithm [30].

1.6 Research Objectives

The research objectives of this thesis are as follows:

- The first objective of this research is to develop a distributive low complexity solution of high definition adaptive filtering algorithm, e.g. RLS algorithm.
- The second objective of this research is that the proposed distributive architecture exhibits minimally reduced communication burden as compared to the existing solutions
- The third objective of this research is that all the nodes use the collaborative strategy though requires limited interaction with the other nodes in the distributed network.

1.7 Thesis Composition

In Chapter 2, the literature survey is presented. In Chapter 3, the system model and existing solutions are described. The proposed low communication PDASP (LC-PDASP) architecture is presented in Chapter 4 and Chapter 5 draws the conclusions and future work.

Chapter 2

Literature Review

This chapter presents literature survey of WSNs, adaptive and non adaptive filtering techniques with pros and cons. This chapter also describes problem statement, research methodology and thesis contribution.

2.1 Literature Survey

A typical wireless sensor node generally consists on micro controller, sensor unit, wireless transceiver and power unit. The power constraints and limited memory restrict the low-cost sensor node to run the complex adaptive filtering algorithm. In the existing adaptive filtering strategies, fast convergence and computational complexity are the main parameters in which more research is needed to improve the convergence and complexity requirements of the high definition algorithms so that they can efficiently run on a single unit [31, 32]. Several techniques have been introduced in the literature that may apply on the high definition adaptive filtering algorithms to reduce their computational cost [23, 24, 33, 34]. In [35, 36], Banachiewics inversion formulation is introduced. In this technique, the 4×4 matrix is divided into four 2×2 matrix which makes the reduction in computation operations. Likewise, in [37], Hermitian positive definite inversion method is introduced which requires only 52 operations for the solving of 4×4 matrix inversion.

Nevertheless, both the matrix inversion techniques do not provide a significant role in the reduction of computational complexity if we apply them on the high definition adaptive filtering algorithm like RLS. Furthermore, in [38], subband adaptive filtering (SAF) for multirate filter banks is introduced. The SAF technique utilizes reduced computational complexity through by using LMS algorithms. However, due to extra processing delays these system are not able to be implemented in real life applications. In [39], a distributed adaptive node-specific signal estimation (DANSE) is introduced. The DANSE techniques uses the wireless sensor nodes to estimate the channel coefficients by using the adaptive Wiener Hopf formulation. The major drawback of DANSE techniques is that it only follows the adaptive Wiener Hopf equation rather than by using gradient or recursive based adaptive filtering algorithms.

Furthermore, one of the particular objectives is that the distributive adaptive solution has a potential to run the complex adaptive filtering algorithm distributively. Moreover, the low cost individual nodes in the distributed adaptive network have the ability to share the bandwidth, computational complexity and power usage and reduced the over all aggregate complexity as compared to the centralized solution [31, 32, 34]. In [10, 40, 41], the consensus based distributed solution is presented. The consensus technique requires two time scales while working on the estimation problem. During the initial time period, each node in the distributed network produces the individual estimate; however, in consensus stage, each node combines the overall estimates of all the nodes in the network and reaching towards the desire estimate. The consensus technique relies on network topology and particular conditions which make ruled out of its implementation in real time environment. In [42] anther type of algorithms were introduced which based on duffision based strategies,these strategies are basically used for distributed estimation and adaptive filtering problems. The main idea behind the diffusion based strategies was to introduce distributed estimation based scheme. Furthermore, the incremental distributed technique is introduced in [26, 27]. In incremental strategy, all the network nodes use the cyclic pattern to find the estimate of the

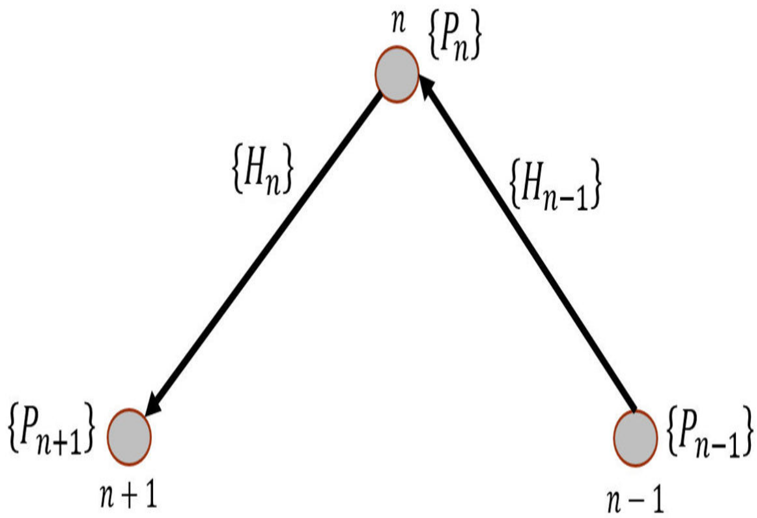

FIGURE 2.1: LC-DRLS Incremental Technique

unknown coefficients with minimum power requirements. Furthermore, in the incremental network, each individual node performs local computations and then share the updated information towards the adjacent node. The number of nodes in the incremental network is dependent upon the total iterations which are used to make possible of the convergence of the adaptive algorithm. As compared to centralized solution, the incremental approach reduces the power requirements and improves the autonomy of the network. However, in [43] case of MIMO channel estimation scenario, the consensus and incremental based network techniques facing high communication burden in sense of transferring of complex MIMO channel matrices among the nodes of the network. Likewise, in [29] low communication distributed recursive least square (LC-DRLS) technique is introduced. In this technique, the communication burden is reduced by initializing the error covariance matrix at each node, which is shown in FIGURE. 2.1. However, all the distributed nodes still 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 (PDASP) [30], a novel processing-efficient PDASP architecture of a group of inexpensive and computationally-incapable small platforms is proposed for a parallely-distributed adaptive signal processing (PDASP) operation. The PDASP scheme exhibits much lesser computational complexity parallely than the sequentially-operated algorithms. PDASP architecture has ability to run RLS algorithm parallely, the wireless sensor nodes used in PDASP architecture have less multiplication and additional complexity. PDASP architecture consist of four nodes one is called master and other three are called slave, master node is defined on basis of its higher computational complexity however; the communication burden among the participating nodes is very high which makes a very crucial impact on the execution time of the algorithm.

2.2 Research Motivation and Problem Statement

As it is discussed earlier, the PDASP architecture requires only four nodes to run the high definition RLS algorithm parallely. Moreover, the PDASP architecture provides reduced computational complexity and processing time parallely as compared to sequential RLS and distributed LC-RLS adaptive algorithms. However, the communication burden provided by the PDASP architecture is very high which makes a very crucial impact on the overall execution time of the algorithm which restricts the PDASP architecture to run in time varying channel environments. To overcome such type of uncertainties, there must be the distributed architecture which provides low computational complexity and processing time parallely solution with minimally low communication overheads and provides fast convergence as compared to the PDASP architecture.

2.3 Research Methodology

In this thesis low-communication parallel distributed adaptive signal processing (LC-PDASP) architecture for a group of computationally-incapable small platforms is introduced. In the proposed architecture each node uses collaborative strategy though required limited interaction with the other nodes; therefore proposed architecture utilize only two nodes for complete communication setup which provides the best utilization of low cost devices than the LC-RLS scheme [44]. Furthermore the proposed LC-PDASP scheme exhibits reduced multiplication complexity and communication burden than conventional PDASP architecture. Moreover the proposed architecture provides an improvement in mean square error (MSE) than PDASP architecture. The convergence performance of proposed scheme tends to be almost same as that of sequentially-operated RLS algorithm.

2.4 Thesis Contributions

Net-shell research contributions of this thesis are given below.

- Processing- efficient Low-communication parallel distributed adaptive signal processing (LC-PDASP) architecture is a group of small, inexpensive and incapable platform is introduced.
- LC-PDASP presented reduced multiplication and communication burden as compared to the recently proposed PDASP.
- Mean Square error is improved in as compared to conventional PDASP while keeping performance same

Chapter 3

System Model and Working of PDASP

3.1 System Model

The block diagram of MIMO communication system with N transmitting and receiving antennas is shown in FIGURE 3.1. In typical MIMO communication system, the input signal is divided into N subblocks which are transmitted separately through the use of multiple antennas. At the receiver end, all the transmitted subblock signals are distorted by the parallel intersymbol interference (ISI) [22], as shown in FIGURE 3.2. Therefore, the received signal $\mathbf{y}_n$ can be expressed as

$$\mathbf{y}_n = \mathbf{H}_n^H \mathbf{x}_n + \boldsymbol{\vartheta}_n \tag{3.1}$$

where

$$\mathbf{H}_n = \begin{bmatrix} h_{11,n} & h_{12,n} & \cdots & h_{1N,n} \\ h_{21,n} & h_{22,n} & \cdots & h_{2N,n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1,n} & h_{N2,n} & \cdots & h_{NN,n} \end{bmatrix}$$

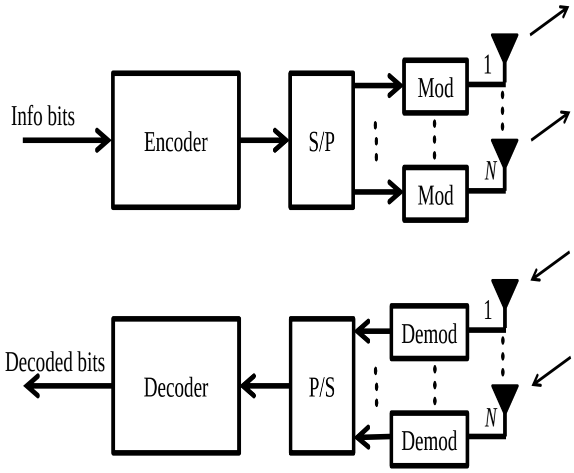

FIGURE 3.1: MIMO System Model

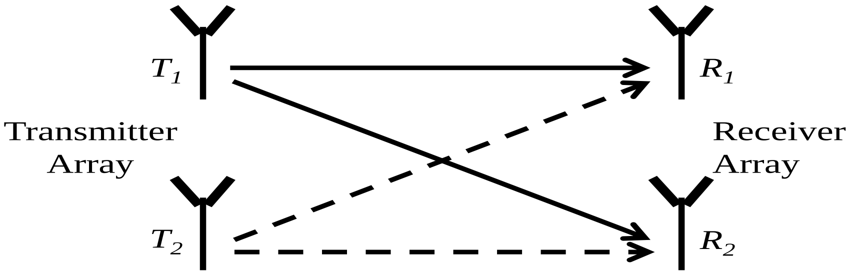

FIGURE 3.2: 2×2 MIMO Communication System

$$\widetilde{\mathbf{H}}_n = \begin{bmatrix} h_{11(0),n} & \cdots & h_{11(L),n} & h_{21(0),n} & \cdots & h_{N1(L),n} \\ h_{12(0),n} & \cdots & h_{12(L),n} & h_{22(0),n} & \cdots & h_{N2(L),n} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ h_{1N(0),n} & \cdots & h_{1N(L),n} & h_{2N(0),n} & \cdots & h_{NN(L),n} \end{bmatrix}$$

---

is $N \times N$ channel matrix, $\mathbf{x}_n = [x_{1,n}\ x_{2,n}\ \cdots\ x_{N,n}]^T$ is the transmitted signal vector, $\boldsymbol{\vartheta}_n = [\vartheta_{1,n}\ \vartheta_{2,n}\ \cdots\ \vartheta_{N,n}]^T$ is the White Gaussion Noise with variance σ_ϑ^2 and subscript n shows the discrete time index. In case of multipath components that exist on the way between transmitter and receiver, make a critical impact on the size of the matrix; therefore, the channel matrix $\widetilde{\mathbf{H}}_n$ becomes an $N \times N(L+1)$ matrix, where L shows the total number of multipath components. The dimensions of $\widetilde{\mathbf{H}}_n$ are not only dependent on the number of transmit and receive antennas but also dependent on the number of multi-path components that exist between the transmit and receive antennas, as shown in FIGURE 3.3.

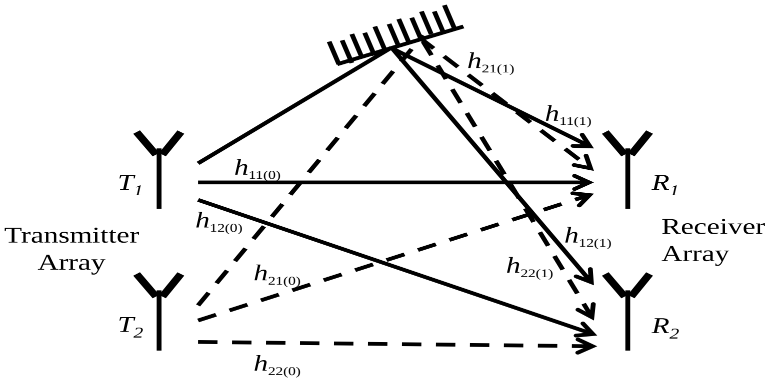

FIGURE 3.3: Frequency Selective Channel Model for 2×2 MIMO Communication System

Each entry of the channel matrix $\widetilde{\mathbf{H}}_n$ is now $h_n^{(trl)}$, where $t = 1, 2, \cdots, N$, $r = 1, 2, \cdots, N$ and $l = 0, 1, \cdots, L-1$. Likewise, $\mathbf{x}_n$ also changes to $\widetilde{\mathbf{x}}_n = [x_{1,n} \ \cdots \ x_{1,n-(L)} \ x_{2,n} \ \cdots \ x_{2,n-(L)} \cdots \ x_{N,n-(L)}]^T$ which is a concatenated transmitted signal vector with elements, $x_{(i),n}$, where i shows the index of the transmit antenna element and n shows the time epoch. Time epochs other than current provide ISI in the model.

3.2 Working Procedure of PDASP Architecture

This section describes the parallel working procedure of sequential RLS algorithm with different time indexes. The important timing variables involved in parallel operation of RLS algorithm are defined as;

- **Computational Time** (T_c): The total time taken by processor for single iteration.
- **Block Processing Time** (T_b): Processing time taken by any specific block of algorithm.
- **Fetch Time** (T_f): Time required to fetch the information one block to another.
- **Algorithm Step time** (T_s): Total time required for one complete iteration of algorithm.

Sequential working of blocks is shown in FIGURE 3.5. Whereas, the working procedure of single block is shown in FIGURE 3.4.

$$
\begin{aligned}
T_c \Big\{\, & x_1 + y_1 = z_1 \\
& x_2 + y_2 = z_2 \\
& \quad \vdots \\
& x_N + y_N = z_N \\
& p_1 \times q_1 = t_1 \\
& \quad \vdots \\
& p_N \times q_N = t_N
\end{aligned}
$$

FIGURE 3.4: A Block Processes

In sequentially operated RLS algorithm, all filter subparts are interdependent on each other and algorithm takes mutual processing time for convergence. Whereas, in PDASP architecture, RLS algorithm runs parallely at different time indexes with low processing time compared to sequential operated RLS algorithm. PDASP architecture also provides reduction in computational complexity and communication burden. Before setting the nonaligned time indexes, two things must be considered. First filter behavior should not be uncertain and second filter subparts should be able to work in parallel fashion with acceptable fetch time with respect to block processing time. Parallel operated RLS filtering architecture has four nodes namely; M1, M2, M3 and M4 as shown in FIGURE 3.6. M1 and M4 are connected with M2 and M3, respectively, and also linked with themselves. M2 is linked with M1 and M4 whereas M3 is only connected to M4. Before starting work on desired goal all processing nodes exchange information with each other. And node with smaller processing time waits until the processing of the other nodes complete. In this way, computationally incapable and low cost platforms work in parallel fashion for desired goal [30]. This is communication procedure of PDASP architecture without considering diffused components, by considering diffused components communication burden among the nodes also increase.

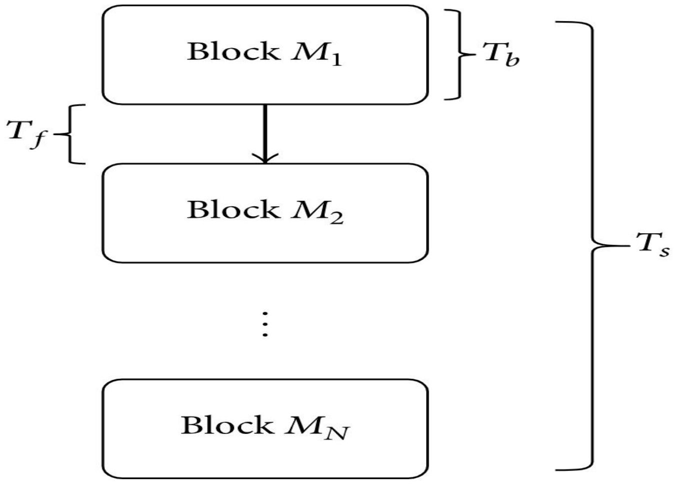

FIGURE 3.5: Sequential Working of Blocks

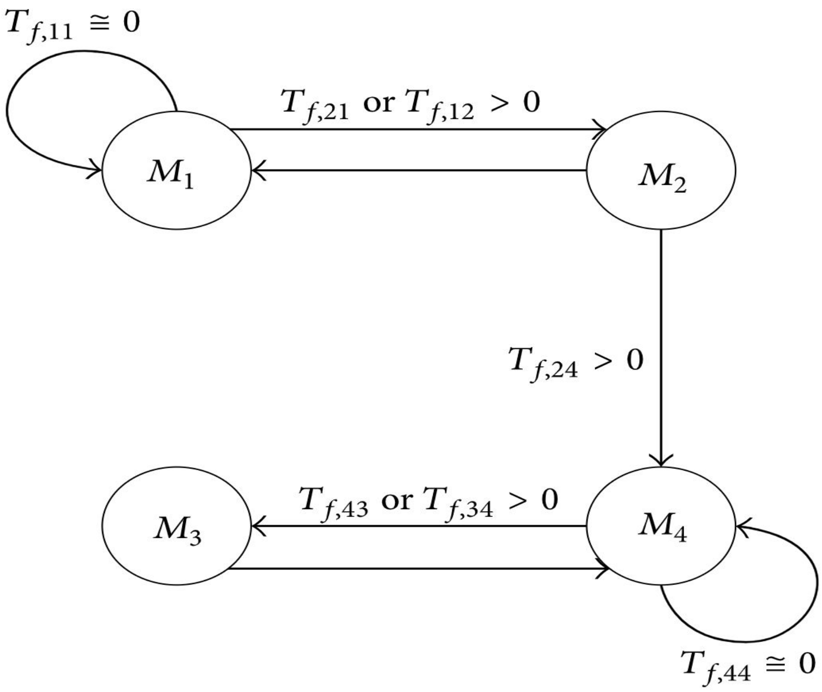

FIGURE 3.6: Parallel Operated RLS Filtering

3.3 Channel Estimation using PDASP Architecture

In the conventional Parallel Distributed Adaptive Signal Processing (PDASP) architecture, as shown in FIGURE 3.7, the RLS algorithm runs in parallel fashion

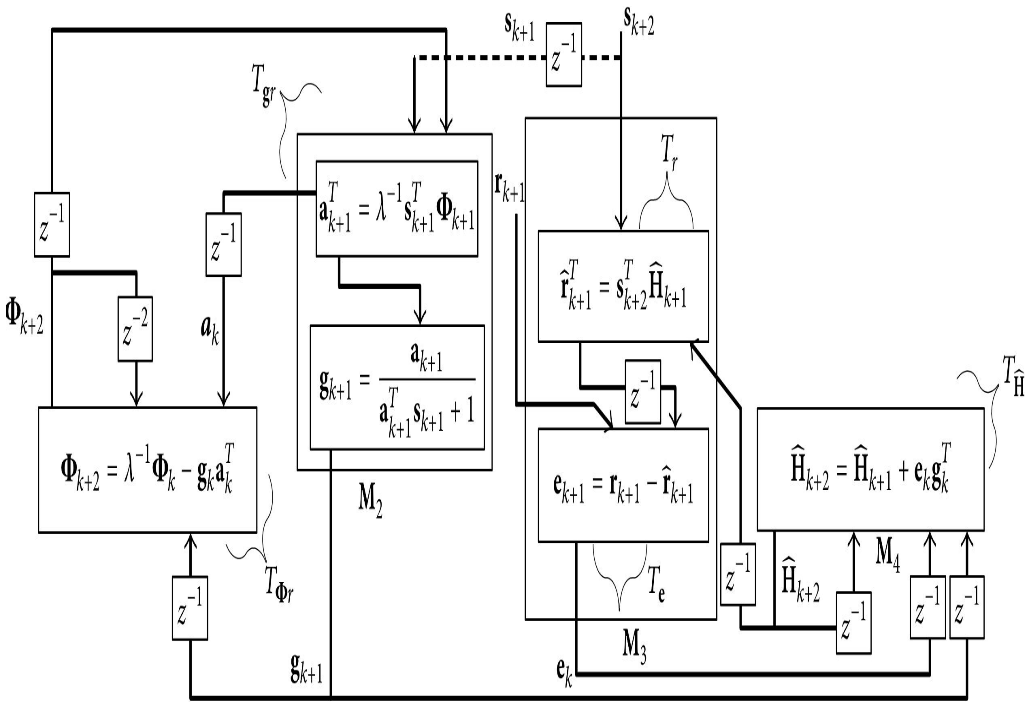

FIGURE 3.7: PDASP architecture for MIMO RLS algorithm with non-aligned time indexes

even with non-aligned time indexes while utilizing low processing time at each processing node. The PDASP architecture provides an average improvement of 94.97% in sense of decreased processing time then the sequential RLS algorithm. However, this improvement in decreased processing time is on behalf of high communication burden which makes a crucial impact on the overall execution time of the adaptive filtering algorithm.

TABLE 3.1: Maximum Communication Burden Specified for one Complete Iteration using LoS and Diffused Components

LoS/Diffused Components	2×2 MIMO	3×3 MIMO	4×4 MIMO
LoS Component	10	18	24
One Diffused Component	28	54	88
Two Diffused Components	54	108	180

3.4 Load Balancing Procedure of PDASP

This section elaborates the information interchanging procedure in PDASP architecture. PDASP architecture consists of four nodes namely; M1, M2, M3 and M4 which are shown in figure. 3.8. Among these four wireless sensor nodes M2 is set to be a master node. This selection has been done on the basis of maximum computational complexity. So M2 behaves as master node and M1, M3 and M4 acts as slave nodes. Before running the actual procedure M2 sends beacon message to all slave nodes to make them ready. The next communication procedure does not start until or unless the process of master node is completed.

The communication burden provided by the PDASP architecture varies for both the LoS and diffused components. In case of LoS communication, the dimensions of the channel matrix $\mathbf{H}_n$ and error covariance matrix $\mathbf{P}_n$ are the same which implies the PDASP architecture follows the totally balancing communication procedure. In this procedure, first of all the update information of Kalman gain $\mathbf{g}_k$ is transmitted from node M_2 towards M_1 and M_4 then the nodes M_1 and M_4 would capable to send the updated information of error covariance matrix P_n and H_n or weight vector towards M_2 and M_3, respectively. Likewise, the nodes M_2 and M_3 send the information of same order towards M_1 and M_4, respectively. However, in case of diffused components, the dimensions of channel matrix $\mathbf{H}_n$ and error covariance matrix $\mathbf{P}_n$ are varies and depend upon the number of transmitting and receiving antennas as well as on the number of multipath components. In case of diffused components, the error covariance matrix $\mathbf{P}_n$ provides overwhelm transmission delay in the communication link [45]. The per-iteration PDASP based maximum communication burden of data elements with LoS and diffused components is shown in Table 3.1. It can be seen that the communication burden provided by the PDASP architecture for diffused components provides a critical impact on the execution time of the algorithm over the distributed adaptive network. By considering multipath components the communication burden of PDASP architecture increase due to change in dimensions of error covariance matrix P_n and channel matrix H_n.

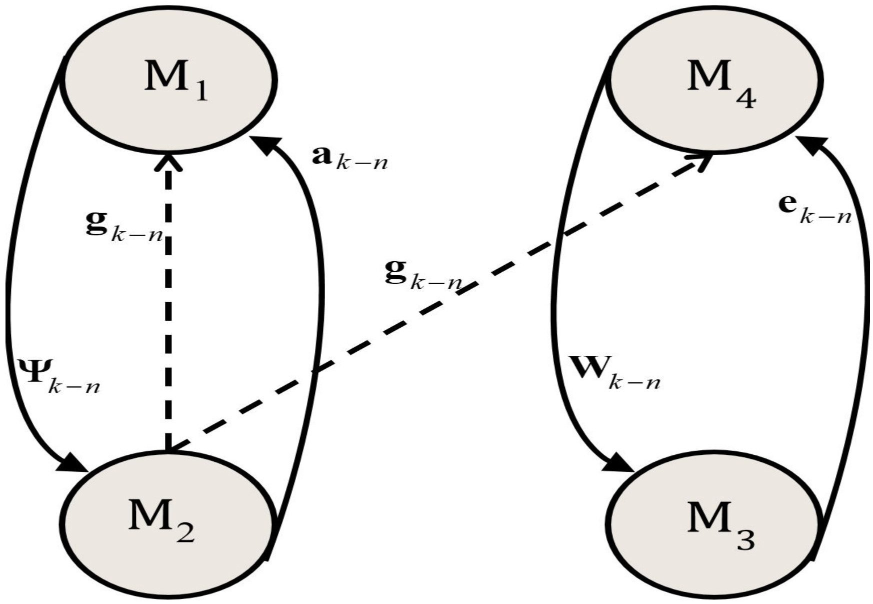

FIGURE 3.8: Load Balancing Procedure of PDASP Architecture

3.5 Validation of PDASP using NRF24L01 Nodes and Memory Usage Comparison

In PDASP architecture, four nodes communicating with each other by using NRF24L01 trans-multiplexer module. The module is based on 2.4 GHz industrial scientific and medical (ISM) band. NRF24L01 module is low cost and low current consumption device with maximum data rate of 2 Mbps.

3.5.1 Sequential Implementation

In sequential operated algorithm all subparts run on single unit. The memory limitation comparison of 2x2, 3x3 and 4x4 MIMO systems by considering zero, one and two diffused components during the execution of sequential algorithm on single unit is shown in Table 3.2. It has been observed that number of antennas and diffused components increase low memory devices like NANO and UNO. NANO and UNO are unable to run sequential algorithm on single unit whereas, MEGA is able to run sequential algorithm on single unit 2x2, 3x3 and 4x4 MIMO systems.

TABLE 3.2: Memory limitation comparison on different sequential MIMO systems with LOS and diffused components using sequential implementation

Arduino platform	Diffused Components	2x2 MIMO System	3x3 MIMO System	4x4 MIMO System
NANO	Zero Diffused Components	Is Working	Is Working	Is Working
UNO	Zero Diffused Components	Is Working	Is Working	Is Working
MEGA	Zero Diffused Components	Is Working	Is Working	Is Working
NANO	One Diffused Components	Is Working	Memory Error	Memory Error
UNO	One Diffused Components	Is Working	Is Working	Is Working
MEGA	One Diffused Components	Is Working	Is Working	Memory Error
NANO	Two Diffused Components	Memory Error	Memory Error	Memory Error
UNO	Two Diffused Components	Is Working	Memory Error	Memory Error
MEGA	Two Diffused Components	Is Working	Is Working	Is Working

3.5.2 Parallel Implementation

The group of the computationally incapable and inexpensive platforms are able to work in parallel fashion with lesser communication burden. The memory improvements comparison of 2x2, 3x3 and 4x4 by considering zero, one and two diffused components is shown in Table 3.3. However among these three nodes NANO is not working on 3x3 and 4x4 MIMO communication system with two diffused components. Whereas by considering two multipath components UNO and MEGA are working on 3x3 and 4x4 MIMO communication system. However for LoS and one multipath component NANO, UNO and MEGA are working on 3x3 and 4x4

MIMO communication system.

Table 3.3: Memory improvement for different MIMO systems with LOS and diffused components using PDASP architecture

Arduino platform	Diffused Components	2x2 MIMO System	3x3 MIMO System	4x4 MIMO System
NANO	Zero Diffused Components	Is Working	Is Working	Is Working
UNO	Zero Diffused Components	Is Working	Is Working	Is Working
MEGA	Zero Diffused Components	Is Working	Is Working	Is Working
NANO	One Diffused Components	Is Working	Memory Error	Is Working
UNO	One Diffused Components	Is Working	Is Working	Is Working
MEGA	One Diffused Components	Is Working	Is Working	Is Working
NANO	Two Diffused Components	Is Working	Memory Error	Memory Error
UNO	Two Diffused Components	Is Working	Memory Error	Is Working
MEGA	Two Diffused Components	Is Working	Is Working	Is Working

3.6 Processing Time Comparison

The processing time of any algorithm is dependent upon the computational complexity of the algorithm as well as on the fetch time of variables while acquiring from the memory. The Processing time comparison between low complexity MIMO channel estimation algorithm and its implementation on PDASP architecture is shown in Table 3.4. It has been observed that processing time provided by

PDASP architecture is much lesser as compared to sequential operated low complexity MIMO channel estimation algorithm. In case no of diffused components, processing time provided by M1 is greater as compared to slave nodes.

Table 3.4: Sequential and distributed Processing Time in Micro sec for different MIMO systems.

Processing Algorithm	Diffused Component	2x2 MIMO System	3x3 MIMO System	4x4 MIMO System
Sequential Algorithm	One Diffused Component	1780	3532	6116
M1	One Diffused Component	524	1008	1612
M2	One Diffused Component	508	1000	1772
M3	One Diffused Component	164	332	680
M4	One Diffused Component	212	508	920
Sequential Algorithm	Two Diffused Component	3272	6852	11816
M1	Two Diffused Component	1004	1992	3080
M2	Two Diffused Component	1020	2268	4544
M3	Two Diffused Component	248	504	884
M4	Two Diffused Component	332	772	1232

The processing time of M1 is greater than all the other nodes due to its maximum computational complexity. However, in some cases of diffused components, the processing time of M2 is increased than M1. This increased processing time of M2 is due to the increased fetch time of variables while acquiring from the memory as compared to the node M1. Furthermore, the sequential and distributed multiplication and addition complexity comparisons for different MIMO systems are shown in Table 3.5 and Table 3.6, respectively. It can be seen that among of all the nodes, M1, entails larger computational complexity which is much lesser than the sequentially-operated low complexity MIMO algorithm.

TABLE 3.5: Sequential and distributed multiplication complexity for different MIMO systems with one and two diffused components.

Processing Algorithm	Diffused Component	2x2 MIMO System	3x3 MIMO System	4x4 MIMO System
Sequential Algorithm	One Diffused Component	57	121	209
M1	One Diffused Component	25	49	81
M2	One Diffused Component	16	36	64
M3	One Diffused Component	8	18	32
M4	One Diffused Component	8	18	32
Sequential Algorithm	Two Diffused Component	109	235	409
M1	Two Diffused Component	149	100	169
M2	Two Diffused Component	36	81	144
M3	Two Diffused Component	12	27	48
M4	Two Diffused Component	12	27	48

The processing time of M1 is greater than all the other nodes due to its maximum computational complexity. However, in some cases of diffused components, the processing time of M2 is increased than M1. This increased processing time of M2 is due to the increased fetch time of variables while acquiring from the memory as compared to the node M1. Furthermore, the sequential and distributed multiplication and addition complexity comparisons for different MIMO systems are shown in Table 3.5 and Table 3.6, respectively. It can be seen that among of all the nodes, M1, entails larger computational complexity which is much lesser than the sequentially-operated low complexity MIMO algorithm.

TABLE 3.6: Sequential and distributed aditional complexity for different MIMO systems with one and two diffused components.

Processing Algorithm	Diffused Component	2x2 MIMO System	3x3 MIMO System	4x4 MIMO System
Sequential Algorithm	One Diffused Component	48	108	192
M1	One Diffused Component	16	36	64
M2	One Diffused Component	16	36	64
M3	One Diffused Component	8	18	32
M4	One Diffused Component	8	18	32
Sequential Algorithm	Two Diffused Component	96	216	384
M1	Two Diffused Component	36	81	144
M2	Two Diffused Component	36	81	144
M3	Two Diffused Component	12	27	48
M4	Two Diffused Component	12	27	48

Chapter 4

Proposed Low Communication PDASP (LC-PDASP) Architecture

In this chapter the proposed LC-PDASP architecture is presented insection 4.1. Moreover comparative analysis of the proposed architecture and the existing PDASP architecture in terms of computational complexity, communication burden and mean squared error is given in section 4.2, 4.3 and 4.4. And it is observed that proposed LC-PDASP architecture outperforms as compared to existing PDASP architecture.

4.1 Proposed LC-PDASP Architecture

The flow diagram of the proposed LC-PDASP architecture is shown in FIGURE 4.1. In the proposed Low Communication Parallel Distributed Adaptive Signal Processing (LC-PDASP) through computationally constrained low cost communication platform is introduced. However, by using the LC-PDASP architecture, the computational complexity respective to each processing node is much lesser than that of sequentially operated low complexity MIMO estimation algorithm and PDASP. The proposed LC-PDASP architecture exhibits reduced multiplication complexity and communication burden than the conventional PDASP architecture. Moreover proposed LC-PDASP architecture provides an improvement in Mean Square Error (MSE) than PDASP.

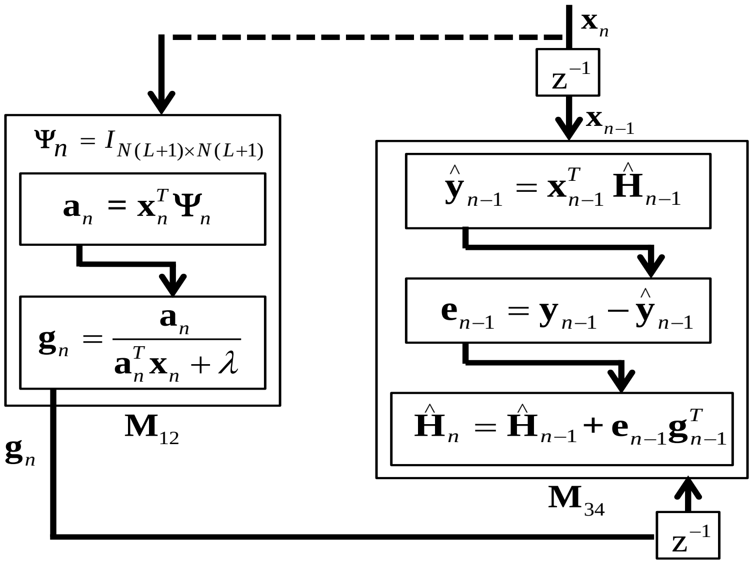

FIGURE 4.1: Proposed LC-PDASP Architecture for MIMO RLS Algorithm

4.1.1 Reduction Of Computational Complexity

In the proposed LC-PDASP architecture, each node uses the collaborative strategy though requires limited interaction with the other nodes in the distributed network. In this context, the individual computational complexity of all the nodes $M_1 \cdots M_4$ in PDASP architecture with diffused components is shown in Table 4.1. Likewise, the combined multiplication and addition complexity of nodes $M_{1,2}$ and nodes $M_{3,4}$ are shown in Table 4.2 and Table 4.3, respectively. It can be seen that the multiplication and addition complexity of the nodes M_1 and M_2 is greater or equal than the both nodes M_3 and M_4. Therefore, if we combine the computational complexity of the nodes M_3 and M_4 that would be less or equal than the individual complexity of the nodes M_1 and M_2.

4.1.2 Reduction Of Communication Burden

Furthermore, the communication burden and the computational complexity implies by the node M_1 is greater than all the other nodes in the PDASP architecture. Our proposition is that the node M_2 locally initialize the error covariance matrix P_n every time and make themselves independent form the node M_1. In this way, the node M_1 has no need to operate in the distributed network. The communication burden of proposed LC-PDASP architecture for different MIMO communication systems is shown in Table 4.4.

Table 4.1: Individual Computational Complexity of all the PDASP Nodes with Diffused Components (DCs)

	Node 1	Node 2	Node 3	Node 4
Multiplication Complexity	$2(N+NL)^2$	$(N+NL)^2+2N(L+1)$	$N^2(L+1)$	$N^2(L+1)$
2×2 with 1 DC	32	24	8	8
2×2 with 2 DC	72	48	12	12
3×3 with 1 DC	72	48	18	18
3×3 with 2 DC	162	99	27	27
Additional Complexity	$(N+NL)^2$	$(N+NL)^2$	$N^2(L+1)$	$N^2(L+1)$
2×2 with 1 DC	16	16	8	8
2×2 with 2 DC	36	36	12	12
3×3 with 1 DC	36	36	18	18
3×3 with 2 DC	81	81	27	27

4.1.3 Reduction Of Processing Time

In the proposed LC-PDASP architecture, only two low cost processing nodes are used for the complete communication setup. Let the processing time taken by

TABLE 4.2: Combined Multiplication Complexity of Node $1, 2$ and Node $3, 4$

Multiplication Complexity	Node 12	Node 34
2×2 MIMO with 1 Diffused Component	24	16
2×2 MIMO with 2 Diffused Components	48	24
3×3 MIMO with 1 Diffused Component	48	36
3×3 MIMO with 2 Diffused Components	99	54

TABLE 4.3: Combined Additional Complexity of Node $1, 2$ and Node $3, 4$

Additional Complexity	Node 12	Node 34
2×2 MIMO with 1 Diffused Component	16	16
2×2 MIMO with 1 Diffused Component	36	36
3×3 MIMO with 1 Diffused Component	36	36
3×3 MIMO with 2 Diffused Components	81	81

TABLE 4.4: Communication burden of proposed LC-PDASP architecture for different MIMO systems

Diffused Components	2x2 MIMO	3x3 MIMO	4x4 MIMO
One Diffused Component	4	6	8
Two Diffused Components	6	9	11

Kalman gain $\mathbf{g}_n$ and channel matrix $\mathbf{H}_n$ be $T_{\mathbf{g}}$ and $T_{\mathbf{H}}$ respectively. Due to the diffused components, the maximum time taken between the two nodes is of node M_{12} because Kalman gain $\mathbf{g}_n$ taken by node M_{12} . Node M_{12} takes more multiplications than the finding of update channel coefficients matric $\mathbf{H}_n$ occupied by node M_{34}; therefore, the processing time of the node M_{12} is greater than the node M_{34} that can be expressed as;

$$T_{\mathbf{H}} < T_{\mathbf{g}} \tag{4.1}$$

Likewise

$$\begin{aligned} T_{\mathbf{H}} &\ll T_{maxPDASP} \\ T_{\mathbf{H}} &\ll T_{seqRLS} \end{aligned} \tag{4.2}$$

where $T_{maxPDASP}$ is the maximum processing time taken by the conventional PDASP architecture and T_{seqRLS} is the time taken by the RLS algorithm when it operates sequentially.

4.2 Complexity Analysis

In this section, the complexity analysis among the sequential RLS, PDASP and proposed LC-PDASP architecture is presented and It is also observed that the communication burden is directly dependent on the number of MIMO streams and number of multipath components. Multiplication complexity. The computational complexity of the sequential RLS algorithm requires $3(N+NL)^2+2N^2(L+1)+2N(L+1)+2$ multiplications and $2(N+NL)^2+2N^2(L+1)$ additions per iteration; where L shows the number of multipath components and N shows the dimensions of filter order. Furthermore, the conventional PDASP based RLS algorithm entials $2(N+NL)^2$ multiplications and $(N+NL)^2$ additions per iteration at maximum. On the other hand, the implementation of LC-PDASP architecture on RLS algorithm exhibits $(N+NL)^2+2N(L+1)$ multiplications and $(N+NL)^2$ additions per iteration at maximum. The multiplication complexity comparisons among sequential RLS, PDASP and proposed LC-PDASP for $L=1$ and $L=2$ diffused components are shown in FIGURE 4.2 and FIGURE 4.3, respectively. It can be seen that the proposed LC-PDASP architecture provides much lesser multiplication complexity than the conventional PDASP and sequential RLS algorithms. Whereas additional complexity of PDASP architecture and LC-PDASP architecture is same but due to increase in multiplication complexity of PDASP the overall computational complexity of PDASP architecture is greater than LC-PDASP architecture.

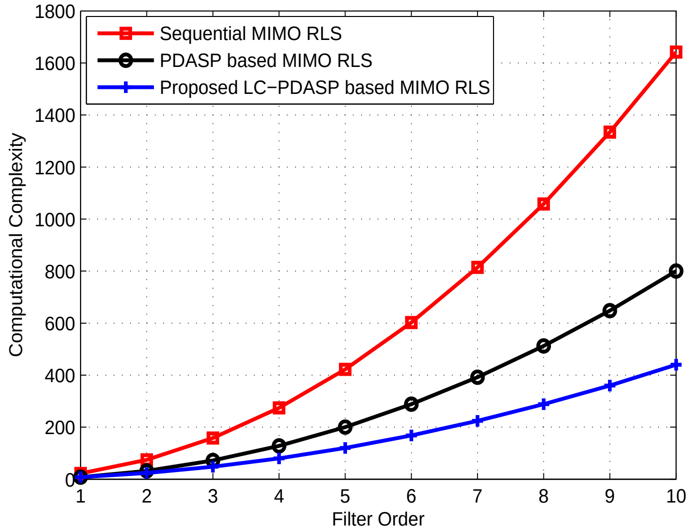

FIGURE 4.2: Multiplication Complexity Comparison among Sequential and Distributed Techniques with One Diffused Component ($L = 1$)

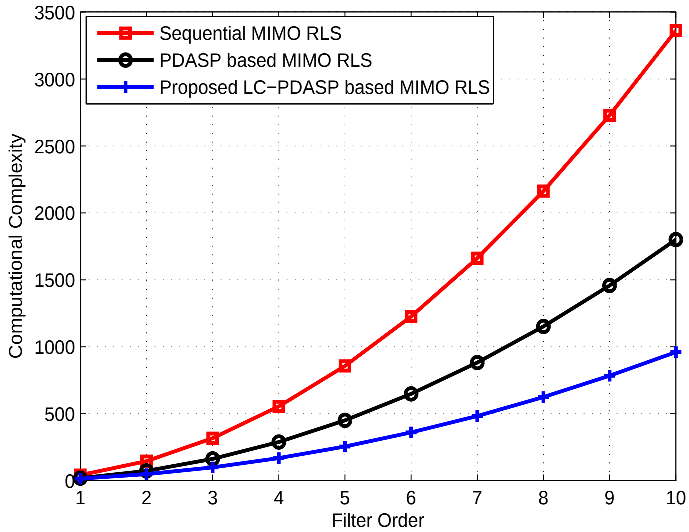

FIGURE 4.3: Multiplication Complexity Comparison among Sequential and Distributed Techniques with Two Diffused Components ($L = 2$)

4.3 Communication Burden Analysis

In this section, the communication burden analysis is presented. The communication burden provided by the proposed LC-PDASP technique is much lesser than the conventional PDASP architecture. The communication burden comparison for different MIMO systems with diffused components is shown in Table 4.5. It can be seen that the proposed LC-PDASP architecture provides $N(L+1)$ communication load; where N shows the MIMO system order and L shows the number of diffused components. Furthermore, the proposed scheme provides an improvement of more than of 85% in sense of decreased communication burden which provides a significant impact on the overall execution time of the algorithm.

TABLE 4.5: Communication burden Comparison between the proposed LC-PDASP and conventional PDASP architecture for different MIMO systems

System Order	Conventional PDASP	Proposed LC-PDASP	% difference
2×2 MIMO with 1 Diffused Component	28	4	85.71%
2×2 MIMO with 2 Diffused Components	54	6	88.88%
3×3 MIMO with 1 Diffused Component	54	6	88.88%
3×3 MIMO with 2 Diffused Components	108	9	91.66%
4×4 MIMO with 1 Diffused Component	88	8	90.90%
4×4 MIMO with 2 Diffused Components	180	12	93.33%

4.4 Mean Square Performance

To substantiate the validation of proposed LC-PDASP scheme, the Monte Carlo simulations are performed on 4×4 MIMO communications system with binary phase shift keying (BPSK). The forgetting factor λ is taken to be 0.98. The proposed LC-PDASP technique is implemented on MIMO RLS is then compared with conventional PDASP architecture and sequential RLS algorithm in terms of mean square error (MSE) performance. FIGURE. 4.4 and FIGURE. 4.5 are presenting the MSE performance of proposed LC-PDASP architecture, conventional PDASP architecture and sequential operated RLS algorithm at low doppler rate $f_D T = 10^{-6}$ and high doppler rate $f_D T = 10^{-3}$, respectively. It can be seen that the convergence performance of the proposed LC-PDASP technique tends to be almost same as that of sequential RLS algorithm whereas; convergence performance of LC-PDASP is much lesser than the conventional PDASP. This improved performance of the proposed technique for both the low and high doppler rates is due to the little involvement of time non-alignments as compared to the PDASP architecture.

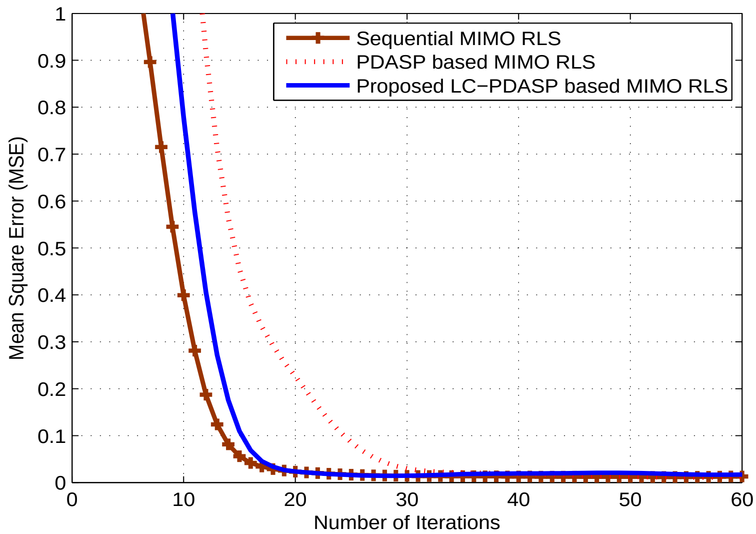

FIGURE 4.4: Mean Square Error (MSE) tracking performance versus training length for 4×4 MIMO when $f_D T = 10^{-6}$

It has been observed that at higher doppler rate LC-PDASP converging faster as compared to PDASP architecture. And convergence performance of LC-PDASP architecture is little slow as compared to sequentially operated MIMO RLS algorithm.

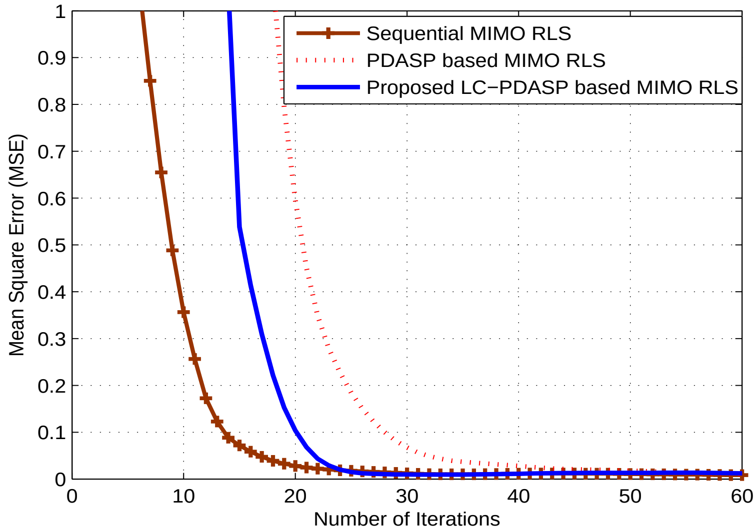

FIGURE 4.5: Mean Square Error (MSE) tracking performance versus training length for 4×4 MIMO when $f_D T = 10^{-3}$

Chapter 5

Conclusions and Future Work

5.1 Conclusion

A new low complexity parallel distributed adaptive signal processing (LC-PDASP) for a group of computationally-incapable and incapable small platforms has been proposed. The proposed architecture with the implementation of MIMO RLS algorithm is capable to run computationally-expensive procedures parallely. The validation of LC- PDASP architecture has been evaluated on the basis of mean square error, computational complexity and communication delay. Processing time and computational complexity of LC-PDASP are compared with sequentially operated low complexity MIMO algorithm and PDASP algorithm. The proposed LC-PDASP architecture using collaborative strategy though required limited interaction with other nodes in the distributed network. It has been seen that the proposed LC-PDASP architecture provides lesser multiplication complexity than the conventional PDASP and sequential RLS algorithms. Furthermore, the communication burden provided by the proposed LC-PDASP architecture is much lesser than the PDASP architecture. Moreover, the mean square error (MSE) provided by the proposed architecture tends to be the lesser and almost same as compared to PDASP and sequential RLS algorithms, respectively. Moereover, the complexity of LC- PDASP architecture is much lesser than the sequentially operated low complexity MIMO channel estimation algorithm and PDASP algorithm. The percentage improvement of LC-PDASP algorithm is more than of 85% in reduction of communication burden that significantly impact on overall execution time of the algorithm as compared to PDASP algorithm and sequential RLS algorithms.

5.2 Future Work

It has been observed that communication burden ,computational complexity and mean square error of Low-Communication Parallel Distributed Adaptive Signal Processing (LC-PDASP) is very good, so the strategy of the future work is divided into two plans which are as follows:

5.2.1 Plan 1: Hardware Implementation Of The Proposed Scheme By Using NRL24101 Transceiver

The NRF24101 is a low cost module which is operated on 2.5GHz ISM band. The maximum data rate provided by this module is 2Mbps. Therefore, the NRF24101 based distributed network provides the access to substantiate the validation of (LC-PDASP) architecture in terms of individual node computational complexity, power utilization and memory utilization and over all communication burden that has to be required for the one complete iteration. Moreover, the attained measured results are also compared to those which are obtained from the conventional hardware based PDASP architecture [30].

5.2.2 Plan 2: Validation Of The Proposed Scheme By Using Network Simulator NS-2

NS-2 is a discrete simulator which is used for the analysis of research work done in the area of networking [46]. NS-2 simulator offers considerable support for the network simulations regarding routing and TCP protocols over wireless and wired networks. Moreover, the NS-2 provides the complete access about the packet index number, end to end delay and packet transfer rate. Nevertheless, the NS-2 simulator is not only the best platform for the analysis of LC-PDASP architecture but can also offer the room to compare the simulations with those of hardware based results.

Bibliography

[1] N. M. Khan and H. Raza, "Processing-efficient distributed adaptive rls filtering for computationally constrained platforms," *Wireless Communications and Mobile Computing*, vol. 10, pp. 1–7, 2017.

[2] J. M. Kahn, H. K. Randy, and S. P. Kristofer, "Next century challenges: mobile networking for smart dust," in *Proceedings of the 5th annual international conference on Mobile computing and networking,* Seattle, Washington, USA. IEEE, 1999, pp. 271–278.

[3] D. Estrin, G. Lewis, P. Greg, and S. Mani, "Instrumenting the world with wireless sensor network," *IEEE Journal on Selected Areas in Communications*, pp. 2033–2036, 2001.

[4] C. Karlof, S. Naveen, and W. David, "Tinysec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems,* $New York, NY, USA$, 2005, pp. 162–175.

[5] A. Sinha and C. Anantha, "Dynamic power management in wireless sensor networks," *IEEE Design and Test of Computers*, vol. 18, no. 2, pp. 62–74, 2001.

[6] S. Muruganathan, D. Ma, R. Bhasin, and A. Fapojuwo, *A centralized energy efficient routing protocol for wireless sensor networks*, 2005, vol. 43, no. 3.

[7] I. Lee and L. Kyoochun, "The internet of things applications, investments, and challenges for enterprises," *Business Horizons*, vol. 54, no. 4, pp. 431–440, 2015.

[8] J. F. Croix and P. K. Sunil, "Introduction to gpu programming for eda," in *Proceedings of the 2009 International Conference on Computer-Aided Design,* UnitedState, vol. 40, no. 3, 2009, pp. 276–280.

[9] J. B. Predd, S. B. Kulkarni, , and H. V. Poo, "Distributed learning in wireless sensor network," *IEEE Signal Process*, vol. 23, no. 4, pp. 134–142, 2006.

[10] L. Xiao, B. Stephen, and L. Sanjay, "A scheme for robust distributed sensor fusion based on average consensus," *Information Processing in Sensor Networks*, vol. 8, pp. 63–70, 2005.

[11] B. B. Farhang, "Adaptive filters: theory and applications," *John Wiley and Sons*, vol. 55, no. 9, pp. 240–248, 2013.

[12] S. O. Haykin, "Adaptive filter theory," *Pearson Higher Ed*, vol. 10, no. 4, pp. 672–680, 2013.

[13] A. N. Delopoulos and G. B. Giannakis, "Strongly consistent identification algorithms and noise insensitive mse criteria," *IEEE Transactions on signal processing*, vol. 40, no. 8, pp. 1955–1970, 1992.

[14] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics and Data Analysis*, vol. 38, no. 4, pp. 467–478, 2002.

[15] R. H. Kwong and W. J. Edward, "A variable step size lms algorithm," *IEEE Transactions on signal processing*, vol. 40, no. 7, pp. 6133–6142, 1992.

[16] B. Widrow, M. M. John, G. L. Michael, and J. Richard, "Stationary and nonstationary learning characteristics of the lms adaptive filter," in *Proceedings of the IEEE,* T.Kummura, 1976, pp. 1151–1162.

[17] D. Bismor, K. Czyz, and Z. Ogonowski, "Normalized least-mean-square algorithm with a pseudo-fractional number of orthogonal correction factors," *International Journal of Acoustics and Vibration*, vol. 21, no. 1, pp. 24–39, 2016.

[18] S. G. Sankaran and A. A. Beex, "Normalized lms algorithm with orthogonal correction factors," in *Conference Record of the Thirty-First Asilomar Conference,* T.Kummura, 1997, pp. 1670–1673.

[19] S. M. Jung, J. H. Seo, and P. G. Park, "Review and comparison of variable step-size lms algorithms," *Journal of Advances in Computer Networks*, vol. 3, no. 2, pp. 465–472, 2015.

[20] P. Whittle, P. Whittle, and G. Britain, *Prediction and regulation by linear least-square methods.* English Universities Press, 1963, vol. 31, no. 5.

[21] M. Z. A. Bhotto and A. Andreas, "Robust recursive least-squares adaptive-filtering algorithm for impulsive-noise environments," *IEEE Signal processing letters*, vol. 8, no. 3, pp. 185–188, 2011.

[22] K. Yano and Y. Susumu, "Cdma non-linear interference canceller with multi-beam reception," vol. 3, pp. 6–10, 2005.

[23] K. Pelekanakis and C. Mandar, "Robust equalization of mobile underwater acoustic channels," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 21, pp. 775–785, 2015.

[24] B. Subudhi and R. Pradhan, "Adaptive predictive error filter-based maximum power point tracking algorithm for a photovoltaic system," *The Journal of Engineering*, no. 4, pp. 54–61, 2016.

[25] T. Yoo and A. Goldsmith, "Capacity and power allocation for fading mimo channels with channel estimation error," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2203–2214, 2006.

[26] C. G. Lopes and H. S. Ali, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4070, 2007.

[27] A. H. Sayed and G. L. Cassio, "Distributed adaptive incremental strategies," in *IEEE International Conference on Acoustics Speech and Signal Processing ,Toulouse, France*, vol. 3, no. 9. IEEE, 2007.

[28] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 90, no. 8, pp. 1504–1510, 2007.

[29] A. H. Sayed, "Distributed recursive least-squares strategies over adaptive networks," in *Signals, Systems and Computers, $PacificGrove, CA, USA$*. IEEE, 2007, pp. 233–237.

[30] I. Ahmad, "Validation of pdasp for quasi-stationary mimo channel estimation through processing-inefficient low-cost communication platforms," *MS thesis,Capital University of Science and Technology*, vol. 18, pp. 1–62, 2018.

[31] 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.

[32] 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.

[33] B. Zhang, "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.

[34] C. Schuldt, L. Fredric, and C. Ingvar, "Low-complexity adaptive filtering implementation for acoustic echo cancellation," in *TENCON,$HongKong$*, 2006, pp. 1–4.

[35] F. Zhang, "The schur complement and its applications," *Springer Science and Business Media*, vol. 4, no. 6, pp. 220–228, 2006.

[36] R. R. Muller, L. Cottatellucci, and V. Mikko, "Blind pilot decontamination," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 773–786, 2014.

[37] Z.-Z. Bai, H. G. Gene, and K. N. Michael, "Hermitian and skew-hermitian splitting methods for non-hermitian positive definite linear systems," *SIAM*

Journal on Matrix Analysis and Applications, vol. 24, no. 3, pp. 603–626, 2003.

[38] A. O. A. Noor, S. A. Samad, and Hussain, "A review of advances in subband adaptive filtering," *World Applied Sciences Journal*, vol. 21, no. 1, pp. 113–124, 2013.

[39] A. Bertrand and M. Marc, "Distributed adaptive node-specific mmse signal estimation in sensor networks with a tree topology," in *Processing of SignalConference,GlasgowUK*, vol. 8, no. 3, 2009, pp. 794–798.

[40] R. Olfati-Saber and S. S. Jeff, "Consensus filters for sensor networks and distributed sensor fusion," in *Decision and Control and control Conference,Seville, Spain.* IEEE, 2005, pp. 6698–6703.

[41] S. R. Olfati, "R. olfati-saber, distributed kalman filter with embedded consensus filters," in *Conference on Decision and Control,seville, spain.* IEEE, 2005, pp. 8179–8184.

[42] Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks formulation and performance analysis," *IEEE Trans on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.

[43] S. Eberli, C. Davide, and F. Wolfgang, "Divide-and-conquer matrix inversion for linear mmse detection in sdr mimo receivers," *NORCHIP*, vol. 37.

[44] Z. X. Ding, Z.Z. and X. Zhu, "A low complexity rls-pastd algorithm for blind multiuser detection in dispersive cdma channels," *IEEE transactions on wireless communications*, vol. 4, no. 6, pp. 165–176, 2007.

[45] A. H. S. Komninakis, C. Fragouli and R. D. Wesel, "Multi-inputmulti-output fading channel tracking and equalization using kalman estimation," *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1665–1676, 2002.

[46] T. Issariyakul and E. Hossain, "Introduction to network simulator," *Springer, Boston, MAComputer*, vol. 10, pp. 4–21, 2012.