Distributed Estimation in Wireless Sensor Networks: Robust Nonparametric and Energy Efficient Environment Monitoring

Upendra Kumar Sahoo

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India

Distributed Estimation in Wireless Sensor Networks: Robust Nonparametric and Energy Efficient Environment Monitoring

Thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

in

Electronics and Communication Engineering

by

Upendra Kumar Sahoo

(Roll: 507EC007)

under the guidance of

Prof. Ganapati Panda

IIT Bhubaneswar, India

and

Prof. Bernard Mulgrew

University of Edinburgh, UK

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

September 2015

dedicated to my parents with love...

Dept. of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

September 30, 2015

Certificate

This is to certify that the work in the thesis entitled ***Distributed Estimation in Wireless Sensor Networks: Robust Nonparametric and Energy Efficient Environment Monitoring*** by ***Upendra Kumar Sahoo*** is a record of an original research work carried out under our supervision and guidance in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Electronics and Communication Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. Ganapati Panda
Professor
School of Electrical Sciences
Indian Institute of Technology Bhubaneswar
Odisha, India - 751 013

Prof. Bernard Mulgrew
Professor
Institute for Digital Communication
The University of Edinburgh
Scotland, UK

Acknowledgment

I owe deep gratitude to the ones who have contributed greatly in completion of this thesis.

Foremost, I would like to express my sincere gratitude to my advisor, Prof. Ganapati Panda for providing me with a platform to work on challenging areas of distributed signal processing in Wireless Sensor Network. His profound insights and attention to details have been true inspirations to my research. I am grateful to my co-supervisor, Prof. Bernard Mulgrew who has provided me with continuous encouragement and support to carry out research. His dedication and consistent notation in my writings has motivated me to work for excellence.

I am thankful to Prof. S. Meher, Prof. S. K. Patra, Prof. K. K. Mahapatra, Prof. D.P. Acharya, Prof. S. K. Behera of Electronics and Communication Engg. Department and Prof. B.D. Subudhi of Electrical Engg. Department for extending their valuable suggestions and help whenever I approached them.

It is my great pleasure to show indebtedness to my friends like Amitav, Trilochan, Pyari, Ajit, Sudhansu, Sitanshu, Satyasai, Babita, George P. Gera, Rakesh, Kallol and Nithin for their help during the course of this work.

I am grateful to NIT Rourkela for providing me adequate infrastructure to carry out the present investigations. I am also grateful to UKIERI project for providing me financial support for traveling to Edinburgh and staying there for three months during summer (2008-2010). I am also grateful to the University of Edinburgh for providing me the resources available at the Department of IDCOM.

I take this opportunity to express my regards and obligation to my family members whose support and encouragement I can never forget in my life.

Upendra Kumar Sahoo

Abstract

Wireless sensor networks estimate some parameters of interest associated with the environment by processing the spatio-temporal data. In classical methods the data collected at different sensor nodes are combined at the fusion center(FC) through multihop communications and the desired parameter is estimated. However, this requires a large number of communications which leads to a fast decay of energy at the sensor nodes. Different distributed strategies have been reported in literature which use the computational capability of the sensor nodes and the estimated local parameters of the neighborhood nodes to achieve the global parameters of interest. However all these distributed strategies are based on the least square error cost function which is sensitive to the outliers such as impulse noise and interference present in the desired and/or input data. Therefore there is need of finding the proper robust cost functions which would be suitable for wireless sensor network in terms of communication and computational complexities.

This dissertation deals with the development of a number of robust distributed algorithms based on the notion of rank based nonparametric robust cost functions to handle outliers in the (i) desired data; (ii) input data; (iii) in both input and desired data; and (iv) desired data in case of highly colored input data. Exhaustive simulation studies show that the proposed methods are robust against 50% outliers in the data, provide better convergence and low mean square deviation.

Further this thesis deals with a real world application of energy efficient environment monitoring. A minimum volume ellipsoid is formed using distributed strategy covering those sensor nodes which indicate the event of interest. In addition a novel technique is proposed for finding the incremental path for regularly placed sensor nodes. It is shown mathematically that the proposed distributed strategy enhances the lifetime of the entire network drastically.

***Keywords*:** Wireless Sensor Networks, Distributed Signal processing, Incremental Minimum Wilcoxon Norm, Outliers, Incremental generalized rank norm, Pseudo Least Squares, Minimum Volume Ellipsoid, Block Householder transformation

Contents

List of Figures

List of Tables

List of Abbreviations

APA	Affine Projection Algorithm
AdIMGRN	Adaptive Incremental Minimum GR Norm
AdIMHBRN	Adaptive Incremental Minimum HBR Norm
AWGN	Additive White Gaussian Noise
BHT	Block Householder Transformation
FC	Fusion Center
GR	Generalized Rank
HBR	High Breakdown
FA	False Alarm
IMGRN	Incremental Minimum Generalized Rank Norm
IMMWN	Incremental Minimum Modified Wilcoxon Norm
IMWN	Incremental Minimum Wilcoxon Norm
LMS	Least Mean Squares
LRT	Likelihood Ratio Test
MAD	Median Absolute Deviation
MVE	Minimum Volume Ellipsoid
PDF	Probability Density Function
PLS	Pseudo Least Squares
QR-IMGRN	QR based Incremental Minimum Generalized Rank Norm
QR-IMWN	QR based Incremental Minimum Wilcoxon Norm
QR-LCIMWN	QR based Low Communication Incremental Minimum Wilcoxon Norm
RLS	Recursive Least Squares
R-APA	Robust Affine Projection Algorithm
SR-IMGRN	Sign-Regressor Incremental Minimum Generalized Rank Norm
SR-IMMWN	Sign-Regressor Incremental Minimum Modified Wilcoxon Norm
SR-IMWN	Sign-Regressor Incremental Minimum Wilcoxon Norm
SS-IMMWN	Sign-Sign Incremental Minimum Modified Wilcoxon Norm
SS-IMWN	Sign-Sign Incremental Minimum Wilcoxon Norm
SR-AdIMGRN	Adaptive Sign Regressor Incremental Minimum GR Norm

SR-AdIMHBRN	Adaptive Sign Regressor Incremental Minimum HBR Norm
SR-IMGRN	Sign Regressor Incremental Minimum GR Norm
SR-IMHBRN	Sign Regressor Incremental Minimum HBR Norm
WSN	Wireless Sensor Network
NE	North-East
NW	North-West
WE	West
EA	East
SO	South
NO	North
SE	South-East
SW	South-West

Chapter 1

Introduction

1.1 Introduction

Necessity is the mother of invention. Digital computer was invented with an objective to compute a large amount of data within a small time interval. This became feasible only due to the development of the notion of stored programme architecture. Development of very-large-scale-integration (VLSI) system helps to decrease the size of the computer to a large extent such that a number of functional units can be fabricated on a single chip. This type of system is called embedded system. The applications of embedded system extend from daily life use to the extreme applications like missile and control of nuclear reactor. However, a single processor system cannot solve a problem when the decision or action requires knowledge of different parts of a system or an environment at the same time. In order to solve this type of problem a multi agent system is required. The multi-agent system can monitor a process that changes in both space and time. Wireless sensor network is the subset of this multi agent type system, in which each agent is equipped with a sensor, a processor and connected to the other agents by a wireless networking system.

1.2 Wireless Sensor Networks

Wireless sensor networks (WSNs) consist of spatially distributed autonomous sensor nodes for monitoring physical or environmental parameters, such as temperature, sound, vibration, pressure, humidity, motion or pollutants cooperatively. Each sensor node comprises a sensing unit, a processing unit, a memory unit, a transmission unit and a power supply unit [1, 2]. Sensor nodes are connected to each other through a wireless networking system. In most of the applications of WSNs, the state of the environment is to be estimated and appropriate action is to be taken accordingly. The data is collected through the sensing unit of the sensor nodes. Transmission unit of the sensor node is responsible for the transmission and reception of the data. The sensor node powered externally by a battery.

The state of the environment is estimated by using the spatiotemporal data recorded by the sensor nodes. Let us consider a popular application of WSNs in precision agriculture. The objective is to save the crop by supplying appropriate

amount of water to the plants. For this the sensor nodes equipped with humidity measurement sensors are spread over the area of interest. The sensor nodes measure the water content in their local regions and the data is sent to the fusion center (FC). The event of interest may be considered mathematically as the vector containing the position of the sensor nodes which measured less than the threshold. However, there are several challenges associated with the practical use of WSNs. Since the sensors have limited processing capability and are powered by external battery, there is need of developing techniques which require less amount of computation and power to process the measured data. Usually, the transmission unit of the sensor nodes requires more power than the other units. Moreover, the sensor nodes share a common wireless medium for communication, thus there is a need to decrease the communication overheads among the sensor nodes by intelligently using the available bandwidth. These challenges motivate the researchers worldwide to design energy efficient strategies for enhancing the lifetime of the WSNs as well as to efficiently fulfill the objective of the state estimation.

1.3 Parameter Estimation in WSNs

This section deals with the mathematical formulation of the state estimation. As previously discussed the objective is to estimate some parameters of interest associated with the environment. Let there be N number of sensor nodes present in the environment and the measured data at the k^{th} sensor node is denoted by $\mathbf{y}_k \in \Re^n, k = 1, \cdots, N$. Thus the data collected from the environment can be represented as $\mathbf{y} = \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots & \mathbf{y}_N^T \end{bmatrix}^T$. Let the global parameter of interest to be estimated be $\mathbf{w} \in \Re^p$. This problem of parameter estimation can be formulated as an optimization problem using the theory of estimation as

$$\mathbf{w} = \arg\max_{\mathbf{w}} \mathrm{E}\left(f_{\mathbf{w},\mathbf{y}}\left(\mathbf{w}, \mathbf{y}\right)\right) \tag{1.1}$$

where E is the expectation operator and $f_{\mathbf{w},\mathbf{y}}$ is the joint probability density function(PDF). This problem of maximization can also be viewed as the minimization of

the negative logarithm of the joint PDF. If the model is linear, the measured data can be modeled as

$$\mathbf{y} = \mathbf{X}^T\mathbf{w} + \mathbf{v} \tag{1.2}$$

The parameter estimation problem in WSNs can be represented as

$$\mathbf{w} = \arg\min_{\mathbf{w}} \left\|\mathbf{y} - \mathbf{X}^T\mathbf{w}\right\|_2^2 \tag{1.3}$$

1.3.1 Classical Method

In classical method sensor nodes measure the environment data and then send it to the FC through multihop communications. This requires a large number of communications. Moreover, this strategy requires a proper organization of the sensor nodes to route the data to the FC and hence is not robust against the failure of a sensor node. Therefore after deployment the sensor nodes need to organize themselves to find the shortest path to the FC. During multihop transmission sensor nodes present near the FC lose more energy than those away from the FC. Subsequently the energy of the sensor nodes near the FC decays below the threshold level. This is called as a dead condition of the WSNs. However, there is sufficient amount of energy remains unused. To increase the lifetime of the WSNs in-network processing capability of the sensor nodes should be facilitated. This can be achieved by compressing the data before transmission to the FC [3, 4]. Moreover, these methods are not robust and not adaptive to the change in environment.

1.3.2 Distributed Estimation in WSNs

To increase the lifetime of the WSNs and to make it robust against failure of any sensor node, the sensors need to process its own data and only the estimated parameters should be shared among the neighboring sensors to obtain the global objective. Based on different cooperation schemes, distributed strategies such as incremental and diffusion techniques have been developed. The distributed estimation in WSN

necessitates [5, 6]

$$J(\mathbf{w}) = \sum_{i=1}^{N} J_k(\mathbf{w}) \tag{1.4}$$

where $J(\mathbf{w})$ and $J_k(\mathbf{w})$ are the global objective and local objective at any sensor node k, respectively. This means the global objective should be factorized into a sum of local objectives.

1.3.3 Incremental Strategy

This strategy requires a predefined incremental path connecting each sensor node present in the environment. Finding an incremental path for a large number of sensor nodes is very difficult. In this strategy each node receives the estimate from the previous node and updates it using its own data to achieve a new estimate. This strategy requires less number of communications among the sensor nodes. The distributed incremental estimation based on the gradient of the cost function is given by

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \nabla \left(\sum_{k=1}^{N} \left(J_k(\mathbf{w}_i) \right) \right) \tag{1.5}$$

1.3.4 Diffusion Strategy

In case of diffusion strategy there is no need for any cyclic path connecting each sensor node [7–10]. A sensor node uses a weighted sum of the estimates from the neighborhood nodes and its own data to generate the *aposteriori* estimate which is again transmitted back to its neighborhood nodes. The weight associated with the estimates from neighborhood nodes can be calculated as [11]

$$\begin{cases} \boldsymbol{\phi}_{k,i-1} = \sum\limits_{i=1}^{N} p_{1,l,k} \mathbf{w}_{l,i-1} \\ \boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \mu_k \sum\limits_{l=1}^{N} s_{l,k} \nabla \left(J_l \left(\boldsymbol{\phi}_{k,i-1} \right) \right) \\ \mathbf{w}_{k,i} = \sum\limits_{l=1}^{N} p_{2,l,k} \boldsymbol{\psi}_{l,i} \end{cases} \tag{1.6}$$

where $\boldsymbol{\phi}_{k,i-1}$ is the convex combination of the previous iteration estimates, $\boldsymbol{\psi}_{k,i}$ is the updated estimate, $\mathbf{w}_{k,i}$ is the convex combination of updated estimates and $p_{1,l,k}$, $p_{2,l,k}$ are nonnegative real coefficients.

The aforementioned gradient based distributed strategies are based on the least squares error cost function which is sensitive to the model uncertainty and outliers present in the data. In addition to the additive white Gaussian noise(AWGN) interference and impulse noise can be viewed as outliers in the data. These problems can be easily solved using a robust cost function.

1.4 Robust Statistics

Robust statistics provides an alternative approach to standard statistical methods, for estimating location, scale and regression parameters [12–14]. The motivation behind the development of robust estimators is that they are not affected by small departure from the model. For example, small departure in a data set alters the mean and variance by a large factor in comparison to the median and MAD. A robust method is quantified by its influence function, breakdown point and sensitivity. Most robust estimation methods use the following principle. Firstly, the samples affected by outliers are detected and then their values are either made zero or decreased using some function. These estimators are broadly categorized into three groups: M-type, L-type and R-type.

1.4.1 M-Type Estimators

M-type estimators are called maximum likelihood type estimators. In this case the likelihood function of some function of the error value is maximized [13, 15]. These estimators may be called monotone or redescending estimators based on the function being used. The former uses a convex function whereas the later uses a redescending function. Huber's function based estimator is one example of monotone M- estimator and is given by

$$\rho_H(e) = \begin{cases} \frac{1}{2}e^2, & \text{for } e \leq k \\ k|e| - \frac{k^2}{2}, & \text{for } |e| > k \end{cases} \tag{1.7}$$

where k is a threshold value. The estimator based on the Huber's function (1.7) is not scale equivariant. Therefore a scale factor needs to be incorporated into it. The threshold value k depends on the deviation of the error distribution from Gaussian distribution. For heavy tailed noise distribution these estimators provide poor performance. Therefore, bisquare weight function which is a redescending function is more appropriate. The bisquare weight function is given by

$$\rho_B(e) = \begin{cases} \frac{k^2}{6}\left\{1 - \left[1 - \left(\frac{e}{k}\right)^2\right]^3\right\} & \text{for } |e| \leq k \\ \frac{k^2}{6} & \text{for } |e| > k \end{cases} \tag{1.8}$$

The gradient descent algorithm based a redescending function may get trapped a local minima. This can be avoided by starting the estimation process with a robust scale method to get a raw estimated near the global minima and then use redescending estimators to reach global minima. However these methods require a large number of computations.

1.4.2 L-Type Estimators

L-type estimators are called a linear combination of order statistics based estimators [15]. In these estimators the residual errors are sorted in increasing order and the middle error values are taken into account for designing the estimation criteria. These estimators are scale equivariant but provides low estimation accuracy.

1.4.3 R-Type Estimators

R-type estimators are called as rank based estimators [16–18]. In these estimates, a block of residual errors are taken into account and the score value corresponding to the individual error is obtained from the rank value of that error. The score and error values are used to design the norm. These norms are scale equivariant and provide

better estimation accuracy. The Wilcoxon norm and the generalized rank norm are the two popular norms.

1.5 Background and Scope of the Thesis

A number of reported materials are available for distributed estimation in wireless sensor networks. Incremental and diffusion least mean squares (LMS) algorithms have been proposed [5, 7, 11, 19] which require less computational complexity. Due to development of low power and efficient VLSI architectures, more sophisticated algorithms like recursive least squares (RLS) and Kalman filter can also be implemented in distributed scenario using less power. Accordingly incremental and diffusion recursive least squares (RLS) algorithms have been suggested [20–23]. Distributed Gossip based algorithms have also been developed for distributed estimation [24, 25]. All these above methods only work well only in the presence of additive white Gaussian noise(AWGN) in the desired data and very sensitive to the presence of outliers in the data. This thesis deals with the development of some novel algorithms to handle outliers in the desired data and/or input data.

As explained in Section 1.4 there are a number of robust cost functions are being used by statisticians for robust regression analysis. However, there is a need of choosing the proper cost functions and novel implementation so that the computational and communication overheads can be reduced. In this thesis some novel robust nonparametric algorithms based on the Wilcoxon norm is designed which require less computations and/or provide faster convergence compared to the existing algorithms.

In practical situations outliers are also present in the input data [26, 27]. Hence there is a need to develop robust methods to handle the outliers in the input data. In this thesis some novel distributed algorithms based on the GR and HBR estimator are developed which requires less computations and/or provide faster convergence compared to previous existing algorithms.

In an environmental monitoring system, the FC always does most of the field estimation work using the entire measured data set. As a result the life time of the network substantially decreases. Hence there is a requirement of designing an en-

ergy efficient environment monitoring system so that life time of the entire network is enhanced. A novel distributed minimum volume ellipsoid based method is developed for environment monitoring which requires less number of communication and computation among the sensor node. Hence the proposed method is energy efficient.

1.6 Motivation Behind the Research Work

Impulse noise and co-channel or adjacent channel interference is also present in the environment along with the additive white Gaussian noise(AWGN) [28, 29]. During measurement process the sensor nodes may capture the impulsive noise and interference. Since The estimation of the probability density function(PDF) of impulse requires large amount computation the estimation process can be made easy by viewing these as outliers in the input and desired data and to handle outliers robust cost function based approach can be used. By this way the parameters associated with the environment can be effectively estimated in real world environment.

However, R-type estimation using simple gradient descent method possess less convergence speed. Hence, there is a need to design appropriate cost function based on the notion R-type estimators, that should have better convergence speed and can be implemented requiring less communication overheads.

Environmental monitoring is an important area of research in wireless sensor network. In most of the cases the fusion center estimates the area where there is abnormality in the environment. Hence, there is also a need for distributed estimation in the environment which is affected by an abnormality. Ellipsoid is a generalized shape which has been used by computer science people for cluster analysis. Moreover, there is also a good amount of literature for finding the minimum volume ellipsoid (MVE) covering a finite set of data. This motivates to use MVE method in distributed manner to indicate the area where the abnormal condition has occurred.

1.7 Objective of the Thesis

The objective of present research work is to develop novel distributed robust algorithms for wireless sensor network under different situations . These are:

- To design a distributed algorithm which are robust against outliers present in the desired and input and input data.
- To enables the new algorithms computationally and communication wise efficient.
- To develop methodology to enhance the convergence speed of the algorithm and performance wise superior.
- To design efficient distributed algorithms suitable for real world situation.

1.8 Structure and Chapter Wise Contribution of the Thesis

Chapter 1 Introduction

The concept of distributed signal processing in wireless sensor network as well as the basics of robust cost function based estimation are presented in this chapter. The motivation behind the use of a rank based robust cost function method is outlined. Moreover, the motivation for designing computationally efficient and communication wise efficient algorithms are also outlined. The chapter wise contributions are also dealt.

Chapter 2 Development of Robust Distributed Strategies for Wireless Sensor Networks

In this chapter the general method of distributed signal processing is reformulated to facilitate the batch processing implementation to suit to sensor network environment. Then the Wilcoxon norm and the sign Wilcoxon norm cost function based distributed signal processing methods are proposed to provide robust estimation performance in presence of outliers in the desired data.

One of the measure problems associated with the Wilcoxon norm and sign Wilcoxon norm is low convergence speed. In order to accelerate the convergence speed a sign regressor Wilcoxon norm and sign sign Wilcoxon norm have been proposed. Simulation studies exhibit that the proposed methods are not only robust against outliers in the desired data but also offers improved convergence speed compared to the other available robust two norms.

However, when the input data is biased, its mean value is not zero, the convergence speed of proposed methods, i.e. sign regressor Wilcoxon and sign sign Wilcoxon, decrease. Moreover when the input data is either positive or negative, the proposed methods do not converge. In order to alleviate the stated shortcomings, a novel cost function is proposed. Mathematically the proposed norm is shown to be convex and also its sign sign and sign regressor counter part are proposed. By simulation it is shown that the proposed norm is robust against outliers in the desired data and its convergence speed is faster than the previous norm for both bias and unbias input data.

Chapter 3
Robust Incremental Distributed Strategy to Handle Outliers both in Input and Desired Data

The methods proposed in the previous chapter are only robust against outliers present in the desired data. In this chapter a new method is proposed which provides robustness against outliers in both input and desired data. An indicator function based a newly proposed approach is used for the mathematical analysis. Being motivated by the tap delay structure of the input, a novel median based approach is incorporated, that requires less number of computation. Simulation studies illustrate that the proposed norm is robust against outliers both in the desired and input data. In addition a sign regressor norm is proposed which not only provides robust performance in the presence of outliers in the input and desired data but also offers improved convergence speed compared to that of the previous norm.

Chapter 4
QR based Incremental Adaptive Strategies to Handle Outliers in the Desired Data as well as in both Input and Desired Data

Some novel approaches based on QR decomposition, are proposed in this chapter which provide faster convergence speed as well as better performance compared to the methods suggested in chapter II and III. However, these require more computation and communication complexities. Due to development of low power VLSI and efficient VLSI architectures these algorithms can be implemented efficiently in wireless sensor nodes. Using simulation based experiments it is demonstrated that the proposed methods are robust against outliers in the desired data, provide better convergence speed and also yield improved performance.

In order to decrease the communication complexity, a low communication QR decomposition based approach is proposed which also provides robust performance.

Chapter 5
Robust Incremental Pseudo Affine Projection Algorithm to Handle Outliers in the Desired Data

The incremental minimum-Wilcoxon-norm(IMWN) proposed in Chapter-II provides very less convergence speed in presence of correlated input data and QR decomposition based approach given in Chapter-IV requires large number of computation. This chapter deals with the development of a pseudo affine projection algorithm which is robust and also provides better convergence speed than the IMWN and less computation than QR-IMWN. In presence of highly correlated input data it provides better convergence speed and estimation performance compared to IMWN. Thus it acts as a compromise between IMWN and QR-IMWN in terms of computation and convergence speed. In order to design this algorithm the Wilcoxon norm is changed to a pseudo least square cost function, which further changed to the pseudo affine projection algorithm by considering the block approximation of the gradient.

Chapter 6
Energy Efficient Environment Monitoring using Minimum Volume Ellipsoid

In this chapter a novel method is proposed for energy efficient environment monitoring using incremental distributed strategy. This strategy requires a predefined path connecting every sensor node present in the network. A simple method is proposed for this. Further, it is proved that by the method of induction that by the local decision of the sensor node, the global incremental path can be found out.

By mathematical analysis it is shown that the proposed method increases the lifetime of the entire sensor network. In order to decrease the computational complexity, the core set is designed and the Langrage multiplier based approach is used.

The above proposed method exhibits less error when the area in the environment can be approximated by a convex shape. But for non convex area a search, make and break approach is used which increases the life time of the entire network.

In order to decrease the quantization effect and further to increase the lifetime of a sensor network, a robust approach is also suggested, which decreases the communication complexity but its performance error increases.

Chapter 7
Conclusion and Future Work

In this chapter the overall contribution of the thesis is reported. Different distributed strategies have been proposed for wireless sensor network to handle outliers only in desired or in both input and desired. Some of the algorithms require less communication and computation complexity but provide poor convergence speed as well as poor performance, where as other algorithms need more computation and communication overheads but provide improved performance and convergence speed. Some novel methods are proposed for energy efficient environment monitoring for a different scenario.

In this chapter future research problems are also outlined for further investigation on the same/related topics. Computation complexity and communication complexity

of the algorithms based upon the different type of implementation can be calculated and compared. In this thesis all the algorithms are applied by using incremental strategy, different other distributed strategies,such as diffusion strategy, adaptive diffusion strategy and probabilistic diffusion strategy can also be applied with an an objective to obtain better performance. Convergence analysis such as steady state and transient analyses can be done using asymptotic linearity of the rank test. In case of the energy efficient environment monitoring the problem can be extended considering the presence of additive white Gaussian noise and impulse noise in the environment.

Chapter 2

Development of Robust Distributed Strategies for Wireless Sensor Networks.

Distributed signal processing is an important area of research in wireless sensor networks (WSNs) which aims to increase the life time of the entire network. In WSNs the data collected by nodes are affected by both additive white Gaussian noise(AWGN) and impulsive noise. The classical square error based distributed techniques used for parameter estimation are sensitive to impulse noise and provide inferior estimation performance. In this chapter, novel robust distributed learning strategies are proposed based on the Wilcoxon norm and its variants. The Wilcoxon norm based learning strategy provides very slow convergence speed. In order to circumvent this, improved distributed learning strategies based on the notion of the Wilcoxon norm are proposed for the different type of environmental data. Simulation based experiments demonstrate that the proposed techniques provide faster convergence speed than the previously reported techniques.

2.1 Introduction

WSNs consisting of large number of sensor nodes are envisioned to solve a large number of modern day problems like environment monitoring, precision agriculture, designing smart house etc [1]. These applications require processing of the data acquired by the sensor nodes. In classical methods the entire data set is sent to the FC where the decision is taken after required processing. This type of solution to the problems requires a large number of communication overheads and leads to the fast decay of energy at the sensor node. Distributed signal processing based upon the processing of data at the individual node and cooperation among the sensors to get the global objective has recently been proposed in the literature [5, 7, 10, 11]. Mostly two types of strategies have been reported in the literature: incremental and diffusion. The incremental strategy requires a predefined cyclic path connecting every sensor nodes present in the environment. This strategy requires less communication overheads and is most suitable for a small number of sensor nodes. The advantage with the diffusion strategy is that it does not require a cyclic path connecting each sensor node, thus suitable for a large number of sensor nodes based WSNs. Using these strategies incremental and diffusion LMS have been proposed [5, 7], for distributed training of

adaptive system which require less computation and attain the global solution. For wireless sensor nodes equipped with low power very-large-scale-integration(VLSI) and efficient VLSI architecture, distributed strategies based on sophisticated algorithms have recently been proposed [21, 23, 30]. However, all these methods are based on least squares error cost function and are very sensitive to the outliers present in the data. In addition to the additive white Gaussian noise(AWGN), impulsive noise is also present in the environment [28]. This impulsive noise is also captured by the measuring instruments during the collection of data. Such type of noise may be viewed as outliers in the data.

Generally the robust cost functions based approach is being used for analysis of data in the presence of outliers. These are broadly classified into three groups, i.e. M-type, L-type and R-type [12, 13, 15]. Since the sensor nodes are operated by finite battery power, the cost functions which require less computation are more suitable. Keeping this in view there is a need to choose appropriate cost functions among all type of robust cost functions.

The M-type estimators are called as maximum likelihood type estimators in which the likelihood function of some function of error values are often used. More discussion about the M-type estimators is given in Section 1.4.1. The important drawbacks associated with these estimators are:

1. the functions associated with these estimators depend on some predefined parameters, which need to be fine tuned in order to get good performance. The predefined parameters depend upon the deviation of the noise distribution from Gaussianity. Since data is spread through out the environment, the estimation of the predefined parameters requires a large number of communication overheads;

2. these estimators in simple form are not scale equivariant [13]. In order to make it scale equivariant, a scale parameter needs to be introduced. Hence, along with the estimation of parameters, the scale factor needs to estimated, which requires large computation and memory space.

The L-type estimators are called as a linear combination of order statistics based

estimator. These estimators are scale equivariant. However the main drawback is that these estimators provide poor performance compared to the M- and R-type estimators [13, 15].

The remaining one is R-type estimators,i.e. rank based estimators. The major advantages of rank based estimators: are simple to implement; are scale equivarint; do not depend upon any predefined parameter; and provide good estimation accuracy.

Due to obvious advantages, the R-type estimators are chosen for distributed implementation to handle outliers in the desired data. The Wilcoxon norm is one of the R-based estimators. Recently it has drawn the attention of the signal processing community for designing robust learning algorithms and robust identification of system parameters [31, 32].

The main contributions made in the chapter are:

1. Two distributed norms: the incremental minimum Wilcoxon norm (IMWN) and incremental minimum sign Wilcoxon norm (IMSWN) are developed based upon the notion in [31, 32];

2. The sign regressor incremental minimum Wilcoxon norm (SR-IMWN) and sign sign incremental minimum Wilcoxon norm (SS-IMWN) are proposed using the notion in [33] and are implemented in distributed sensor networks. These methods provide better convergence speed than the previous methods with the slightly inferior performance compared to that of IMWN and IMSWN.

3. A robust cost function is suggested based on the notion of the Wilcoxon norm to handle outliers in the desired data in the presence of biased input data and offers faster convergence compared to the conventional ones.

2.2 Problem Formulation

Suppose N number of sensor nodes are deployed in a locality of interest. Each sensor takes its measurement after fixed time intervals. Let the spreading time be t_0 and the measurement starts at time t_0 and continues up to time t_1 with interval of T. Thus, the total number of measurements obtained by one sensor within the time t_0 and t_1

is $(t_1 - t_0)/T + 1 = n$. Let us consider one measuring time as one time instant so that there are n number of time instants. The measured training and input data at the k^{th} node at time instant i are denoted by $y_{k,i}$ and $\mathbf{x}_{k,i}$, respectively. Let the input and desired data of the physical system be related by a linear model [34, 35] as

$$y_{k,i} = \mathbf{x}_{k,i}^T \mathbf{w} + v_{k,i} \tag{2.1}$$

where $\mathbf{x}_{k,i} \in \Re^p$, p is the order of the system and $v_{k,i}$ is the AWGN and impulsive noise present at the environment during the i^{th} time instant.

The entire spatial desired and input data from the first node to the N^{th} node during the i^{th} time instant are denoted by $\mathbf{y}_i$ and $\mathbf{X}_i$ respectively, which is represented by

$$\mathbf{y}_i = \begin{bmatrix} y_{1,i} & y_{2,i} & \cdots & y_{N,i} \end{bmatrix}^T \qquad \mathbf{X}_i = \begin{bmatrix} \mathbf{x}_{1,i} & \mathbf{x}_{2,i} & \cdots & \mathbf{x}_{N,i} \end{bmatrix} \tag{2.2}$$

.

Further the spatio-temporal desired and input data upto n^{th} time instant are denoted by $\mathbf{y}(n)$ and $\mathbf{X}(n)$ respectively and are given by

$$\mathbf{y}(n) = \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots & \mathbf{y}_n^T \end{bmatrix}^T \qquad \mathbf{X}(n) = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix} \tag{2.3}$$

.

The objective is to estimate the parameter $\mathbf{w}$ from the desired data $\mathbf{y}(n)$ and the input data $\mathbf{X}(n)$. The objective can be formulated an optimization problem given in (2.4)

$$\mathbf{w} = \arg\min_{\mathbf{w}} \left\| \mathbf{y}(n) - \mathbf{X}^T(n)\mathbf{w} \right\|_* \tag{2.4}$$

When the noise $\mathbf{v}$ is Gaussian, the optimization problem is changed to

$$\mathbf{w} = \arg\min_{\mathbf{w}} \left\| \mathbf{y}(n) - \mathbf{X}^T(n)\mathbf{w} \right\|_2^2 \tag{2.5}$$

2.2.1 Wilcoxon Norm

The Wilcoxon norm is a pseudo norm [16] which is defined on a vector $\mathbf{v} \in \Re^l$

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_l \end{bmatrix} \tag{2.6}$$

.

The Wilcoxon norm of the vector (2.6) is given by

$$\|\mathbf{v}\|_w = \sum_{i=1}^{L} \left[\varphi\left(v_i\right) v_i\right] = \sum_{i=1}^{L} \left[\sqrt{12}\left(R\left(v_i\right)/(L+1) - 0.5\right) v_i\right] \tag{2.7}$$

where $\varphi(v_i)$ is the score function associated with the element v_i present in the vector $\mathbf{v}$. The score function exhibits the following properties

1. $\varphi : \begin{bmatrix} 0 & 1 \end{bmatrix} \rightarrow \Re$
2. It is bounded as $\int \varphi^2\left(u\right) d\left(u\right) < \infty$
3. It is a pseudo norm and hence

$\int_0^1 \varphi\left(u\right) d\left(u\right) = 0.$

For the Wilcoxon norm the score value is defined as

$$\varphi\left(u\right) = \sqrt{12}\left(u - 0.5\right) \tag{2.8}$$

The score value corresponds to the element v_k of the vector $\mathbf{v}$ is $\sqrt{12}\left(\frac{\mathrm{R}(v_k)}{l+1} - 0.5\right)$, where $R\left(v_k\right)$ is the rank order of the element v_k among all the elements present in the vector. The rank order defines the position of an element when all the elements are arranged in ascending order.

2.2.2 Block Formulation of the Problem

Since the Wilcoxon norm requires a block of data for implementation hence block processing is necessary for distributed implementation of the Wilcoxon norm. This is achieved by introducing the notion of block time. Thus, n number of time instants can be divided into $\lceil n/l \rceil$ number of block times. Without loss of generality we can assume n is an integer multiple of l. Therefore i^{th} block time is the time from $((i-1)l+1)^{th}$ time instant to $(il)^{th}$ time instant. The input and desired data in block form length l at i^{th} block time of k^{th} node are given as

$$\begin{aligned} \mathbf{X}_{k,i} &= \begin{bmatrix} \mathbf{x}_{k,(i-1)l+1} & \mathbf{x}_{k,(i-1)l+2} & \cdots & \mathbf{x}_{k,il} \end{bmatrix} \\ \mathbf{y}_{k,i} &= \begin{bmatrix} \mathbf{y}_{k,(i-1)l+1} & \mathbf{y}_{k,(i-1)l+2} & \cdots & \mathbf{y}_{k,il} \end{bmatrix}^T \end{aligned} \tag{2.9}$$

. The spatial input data and corresponding desired data at i^{th} block time are denoted by

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{X}_{1,i} & \mathbf{X}_{1,i} & \cdots & \mathbf{X}_{1,i} \end{bmatrix} \qquad \mathbf{y}_i = \begin{bmatrix} \mathbf{y}_{1,i}^T & \mathbf{y}_{1,i}^T & \cdots & \mathbf{y}_{1,i}^T \end{bmatrix}^T \tag{2.10}$$

.

Moreover, the entire spatial input and desired data from the first block time to n^{th} block time are denoted as

$$\mathbf{X}(n) = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix} \qquad \mathbf{y}(n) = \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots & \mathbf{y}_n^T \end{bmatrix} \tag{2.11}$$

.

Similar to (2.4) the objective is to estimate $\mathbf{w}$ from the input and desired data shown in (2.11) using block optimization form as

$$\mathbf{w} = \arg\min_{\mathbf{w}} \left\| \mathbf{y} - \mathbf{X}^T\mathbf{w} \right\|_{wil} = \arg\min_{\mathbf{w}} \chi(\mathbf{w}) \tag{2.12}$$

The Wilcoxon norm (2.12) is a nonlinear function, which depends on the number

of data present in the residual error vector corresponding to the desired and input vectors shown in (2.11). The distributed implementation requires $\chi(\mathbf{w}) = \sum_{k=1}^{N} \chi_k(\mathbf{w})$. However, it is not true for the Wilcoxon norm and hence the objective (2.12) can not be achieved by distributed strategies.

To facilitate the distributed implementation of the MWN, define the local cost function for node k as

$$\min_{\mathbf{w}} \left\| \mathbf{y}^k - \left(\mathbf{X}^k\right)^T \mathbf{w} \right\|_{wil} \tag{2.13}$$

where

$$\mathbf{y}^k = \begin{bmatrix} \mathbf{y}_{1,1}^T & \mathbf{y}_{1,2}^T & \cdots & \mathbf{y}_{1,n/l}^T \end{bmatrix}^T \text{ and } \mathbf{X}^k = \begin{bmatrix} \mathbf{X}_{1,1}^T & \mathbf{X}_{1,2}^T & \cdots & \mathbf{X}_{1,n/l}^T \end{bmatrix}^T \tag{2.14}$$

The local objective depends upon the data collected by the k^{th} sensor node during the measurement process. Based upon the local objective (2.13), the global objective is defined as

$$\min_{\mathbf{w}} \sum_{i=1}^{N} \chi_k(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^{N} \left\| \mathbf{y}^i - \left(\mathbf{X}^i\right)^T \mathbf{w} \right\|_{wil} \tag{2.15}$$

Since (2.15) is an affine combination of the local Wilcoxon norm, which is a pseudo norm, the global objective (2.15) is also a pseudo norm [16]. However, the objective (2.15) provides less performance than the former objective function defined in (2.4). This can be verified by using the theory of rank statistics [16]. The estimation efficiency of a norm depends upon the variance of the estimation error. In case of the Wilcoxon norm it directly depends on the number of data present in the residual errors vector. However, by adding the number of cost functions, the variance decreases linearly where as by adding a number of data in the vector variance decreases by the power of two. Further, the drawback of the cost function(2.15) is that it depends upon the entire data acquired during the measurement process. Thus, the estimation is to be initiated after the end of the measurement process. It requires more number of

computations and a large memory, which increases with the increase of measurement data. Thus, there is need of recursive implementation of the (2.15) at the node. This is not possible because of the dependence of (2.15) on the rank of each data.

In order to achieve recursive implementation, the local objective is modified to

$$\boldsymbol{\chi_k} = \sum_{i=1}^{L} \left\| \mathbf{y}_{k,iL} - \mathbf{X}_{k,iL}^{T} \mathbf{w} \right\|_{wil} \tag{2.16}$$

It is termed as temporal local objective. From the performance point of view the temporal local objective differs from the local objective (2.12) in a similar way the objective function (2.15) differs from that of (2.4). Similar to (2.15) the global objective function is given by

$$\left\| \mathbf{y} - \mathbf{X}^{T} \mathbf{w} \right\|_{wl} = \sum_{j=1}^{N} \sum_{i=1}^{n/l} \left\| \mathbf{y}_{j,((i-1)l+1):(il)} - \mathbf{X}_{j,((i-1)l+1):(il)}^{T} \mathbf{w} \right\|_{wl} \tag{2.17}$$

The efficiency may further deteriorate in such formulation but it can be implemented recursively by using distributed strategies.

2.3 Incremental Minimum-Wilcoxon-Norm (IMWN) Incremental Minimum-Sign-Wilcoxon-Norm (IMSWN)

As discussed in the previous section the objective function to be minimized is given in (2.17). In order to achieve this the gradient based method is used. Hence, the parameter is updated in the direction of negative gradient [34, 35] of the cost function and the corresponding update equation may be written as

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \mu \nabla \left(\chi \left(\mathbf{w} \right) \right) \tag{2.18}$$

Equation (2.18) requires all the data collected from the environment at the block time i. Hence, the incremental strategy is used to employ the data from first to last

node to estimate the parameters. In this strategy during i^{th} time instant a node receives the update parameters from previous node and uses its own data to further update the parameters. Subsequently these parameters are sent to the next node. Let the estimated parameter at node k during the spatial recursion in the i^{th} block time be represented as $\boldsymbol{\psi}_{k,i}$. In this case the $\mathbf{w}_i$ is obtained from $\mathbf{w}_{i-1}$ as

$$\mathbf{w}_i \equiv \boldsymbol{\psi}_{N,i} \leftarrow \cdots \boldsymbol{\psi}_{2,i} \leftarrow \boldsymbol{\psi}_{1,i} \equiv \mathbf{w}_{i-1} \tag{2.19}$$

Further taking the gradient and using it in (2.18), we obtain

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \mu \sum_{i=1}^{N} \sum_{j=1}^{l} \varphi\left(e_{i,j}\right) \mathbf{X}_j \tag{2.20}$$

Thus the distributed incremental algorithm is outlined as

$$\begin{aligned}
&\boldsymbol{\psi}_{0,i} \leftarrow \mathbf{w}_{i-1} \\
&\text{For } k = 1 : N \\
&\quad \boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k-1,i} + \mu \\
&\quad \times \left[\sum_{k=1}^{l} \sqrt{12} \left(\frac{\mathrm{R}\left(y_{j,(i-1)l+k} - \mathbf{x}_{j,(i-1)l+k}^T \boldsymbol{\psi}_{k-1,i}\right)}{l+1} - 0.5 \right) \right. \\
&\quad \left. \times \left(\mathbf{x}_{j,(i-1)l+k}\right) \right] \\
&\text{end} \\
&\mathbf{w}_i \leftarrow \boldsymbol{\psi}_{N,i}
\end{aligned} \tag{2.21}$$

In case of sign Wilcoxon norm the score value in (2.21) is changed to

$$\varphi\left(u\right) = \mathrm{Sign}(u - 0.5) \tag{2.22}$$

2.4 Proposed Sign-Regressor Incremental Minimum-Wicoxon-Norm (SR-IMWN) and Sign-Sign Incremental Minimum Wilcoxon Norm (SS-IMWN)

It is known that the sign-regressor LMS and the sign-sign LMS provide better convergence speed compared to the LMS [36] where it is demonstrated that the sign regressor Wilcoxon and sign-sign Wilcoxon are robust against outliers and converges faster than the Wilcoxon norm. Motivated by these findings two algorithms, SR-IMWN and SS-IMWN(Which are variants of the incremental minimum Wilcoxon norm) are proposed to achieve better convergence speed than the IMWN as well as robust performance against outliers in the desired data. The derivation proceeds as follows. The incremental LMS algorithm [5] is given as

$$\begin{cases} \boldsymbol{\psi}_{o,i} \leftarrow \mathbf{w}_{i-1} \\ \text{For } k = 1 : N \\ \quad \boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k-1,i} + \mu \left(y_{k,i} - \mathbf{x}_{k,i}^T \boldsymbol{\psi}_{k-1,i} \right) \mathbf{x}_{k,i} \\ \text{end} \\ \mathbf{w}_i \leftarrow \boldsymbol{\psi}_{N,i} \end{cases} \tag{2.23}$$

The IMWN in (2.21) can be changed to matrix-vector multiplication form as

$$\begin{cases} \boldsymbol{\psi}_{o,i} \leftarrow \mathbf{w}_{i-1} \\ \text{For } k = 1 : N \\ \quad \boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k-1,i} + \mu \text{Score}\left(\mathbf{e}\right) \mathbf{X}_{((i-1)L+1):L} \\ \text{end} \\ \mathbf{w}_i \leftarrow \boldsymbol{\psi}_{N,i} \end{cases} \tag{2.24}$$

where $\text{Score}\left(\text{e}\right) = \sqrt{12} \left[\begin{array}{cccc} \frac{\text{R}\left(\text{e}_{(\text{i}-1)\text{l}+1}\right)}{l+1} - 0.5 & \frac{\text{R}\left(\text{e}_{(\text{i}-1)\text{l}+2}\right)}{l+1} - 0.5 & \cdots & \frac{\text{R}(\text{e}_{\text{il}})}{l+1} - 0.5 \end{array} \right]$ and $\mathbf{X}_{k,(k-1)l:k} = \left[\begin{array}{ccccc} \mathbf{x}_{k,(k-1)l+1} & \mathbf{x}_{k,(k-1)l+2} & \cdots & \mathbf{x}_{k,kl} \end{array} \right]$

Comparing (2.23) and (2.24), it is observed that the Score($\mathbf{e}$) and $\mathbf{x}_{il+1}$ in IMWN act like e and $\mathbf{x}$ in the ILMS algorithm. For the case of the sign-regressor ILMS and

the sign-sign ILMS the parameters update equations are

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\psi}_{k-1,i} + \mu \left(y_{k,i} - \mathbf{x}_{k,i}^T \boldsymbol{\psi}_{k-1,i}\right) \text{Sign}\left(\mathbf{x}_{k,i}\right) \tag{2.25}$$

and

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\psi}_{k-1,i} + \mu \text{Sign}\left(y_{k,i} - \mathbf{x}_{k,i}^T \boldsymbol{\psi}_{k-1,i}\right) \text{Sign}\left(\mathbf{x}_{k,i}\right) \tag{2.26}$$

respectively.

Comparing (2.23),(2.24)and (2.25), the algorithm for SR-IMWN may be outlined as

$$\begin{cases} \boldsymbol{\psi}_{0,i} \leftarrow \mathbf{w}_{i-1} \\ \text{For } k = 1 : N \\ \quad \boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k-1,i} + \mu \times \text{Score}\left(\mathbf{e}_{((i-1)L+1):L}\right) \text{Sign}\left(\mathbf{X}_{((i-1)L+1):L}\right) \\ \text{end} \\ \mathbf{w}_i \leftarrow \boldsymbol{\psi}_{N,i} \end{cases} \tag{2.27}$$

Further, comparing (2.23),(2.24) and (2.26), the update operation for SS-IMWN is obtained as

$$\begin{cases} \boldsymbol{\psi}_{0,i} \leftarrow \mathbf{w}_{i-1} \\ \text{For } k = 1 : N \\ \quad \boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k-1,i} + \mu \times \text{Sign}\left(\text{Score}\left(\mathbf{e}_{((i-1)L+1):L}\right)\right) \text{Sign}\left(\mathbf{X}_{((i-1)L+1):L}\right) \\ \text{end} \\ \mathbf{w}_i \leftarrow \boldsymbol{\psi}_{N,i} \end{cases} \tag{2.28}$$

Changing the matrix vector multiplication term in (2.27) and (2.28) to summation of scalar vector multiplication term we get

$$
\begin{aligned}
&\boldsymbol{\psi}_{0,i} \leftarrow \mathbf{w}_{i-1} \\
&\text{For } k = 1 : N \\
&\quad \boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k-1,i} + \mu \\
&\quad \times \left[\sum_{k=1}^{l} \sqrt{12} \left(\frac{\mathrm{R}\left(y_{j,(i-1)l+k} - \mathbf{x}_{j,(i-1)l+k}^{T} \boldsymbol{\psi}_{k-1,i}\right)}{l+1} - 0.5 \right) \right. \\
&\quad \left. \times \mathrm{Sign}\left(\mathbf{x}_{j,(i-1)l+k}\right) \right] \\
&\text{end} \\
&\mathbf{w}_i \leftarrow \boldsymbol{\psi}_{N,i}
\end{aligned}
\tag{2.29}
$$

and

$$
\begin{aligned}
&\boldsymbol{\psi}_{0,i} \leftarrow \mathbf{w}_{i-1} \\
&\text{For } k = 1 : N \\
&\quad \boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k-1,i} + \mu \\
&\quad \times \left[\sum_{k=1}^{l} \sqrt{12} \mathrm{Sign} \left(\frac{\mathrm{R}\left(y_{j,(i-1)l+k} - \mathbf{x}_{j,(i-1)l+k}^{T} \boldsymbol{\psi}_{k-1,i}\right)}{l+1} - 0.5 \right) \right. \\
&\quad \left. \times \mathrm{Sign}\left(\mathbf{x}_{j,(i-1)l+k}\right) \right] \\
&\text{end} \\
&\mathbf{w}_i \leftarrow \boldsymbol{\psi}_{N,i}
\end{aligned}
\tag{2.30}
$$

2.5 Simulation Results and Discussions

This section deals with simulation based experiments to assess the robust estimation performance of the proposed norms based distributed strategies. For simulation purpose five sensor nodes are taken into consideration. Total 4000 number of random input data is generated with the magnitude between $(-0.5, 0.5)$. The parameters $\mathbf{w}$ of model (2.1) is $\left[\begin{array}{ccccc} 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} \end{array} \right]$. In order to obtain the desired value, the input data is passed to the model and the output is mixed with white Gaussian noise of 30dB SNR as well as outliers. In order to mix outliers in the desired data, random positions are chosen depending upon the percentage of outliers. Then random magnitudes are generated depending upon the strength of the outliers and then these are added with the selected samples. For example to generate training

samples having 30% outliers of strength 10, firstly, 30 number of desired data points are chosen in every 100 measured samples and random values between $(-10, 10)$ are added with them. Results are obtained for different percentage and strength of outliers. But in this chapter the simulation results corresponding to 10% and 50% of outliers having magnitudes between $(-10, 10)$ are presented. The step size used is 0.001. The block length considered during simulation is 40. The simulation results shown are averaged over 20 independent experiments.

The convergence characteristics are obtained for each of the ILMS,IMWN,IMSWN,SS-IMWN and SR-IMWN based algorithms and are presented in Figures 2.1-2.2 for 10% and 50% outliers respectively. It is observed that the IMWN, IMSWN, SS-IMWN and SR-IMWN based algorithms exhibit robust performance against outliers in the desired data, where as ILMS offers the worst convergence in presence of outliers. Further, the SS-IMWN and SR-IMWN norm based algorithms provide better convergence speed but the inferior performance compared to the IMWN and IMSWN based algorithms. As the percentage of outliers increases, both the convergence speed and performance of all the algorithms deteriorate.

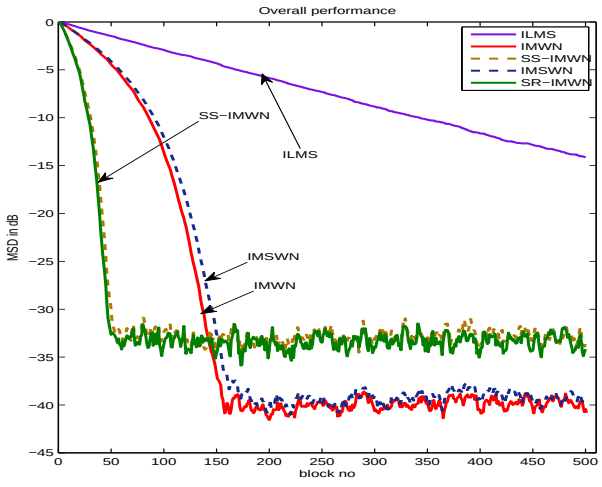

Figure 2.1: Overall convergence performance with 10% outliers in the desired data with magnitude between (-10,10)

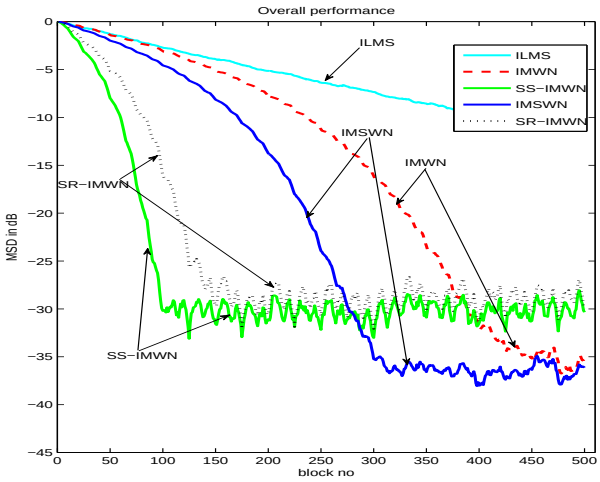

Figure 2.2: Overall convergence performance with 50% outliers in the desired data with magnitude between (-10,10)

2.5.1 Drawbacks of SR-IMWN and SS-IMWN

The simulation results shown in Figures 2.1-2.2 correspond to the unbiased input data i.e. the mean of the input data is zero. The convergence characteristics obtained from the simulation results for biased input data lying between $(-0.1, 0.9)$ and $(0, 1)$ are presented in Figures 2.3-2.4 respectively. The observations made from these plots are:

1. The convergence speed of proposed SR-IMWN and SS-IMWN based algorithms decrease when the biased input data is used;

2. The SR-IMWN and SS-IMWN based algorithms do not converge if the input data is either positive or negative.

The theoretical explanation is as follows. The update equation for sign-regressor GR norm is given as

$$\boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k-1,i} + \mu \left[\sum_{j=1}^{l} \sqrt{12} \text{Sign} \left(\frac{\text{R} \left(y_{k,(i-1)l+j} - \mathbf{x}_{k,(i-1)l+j}^{T} \boldsymbol{\psi}_{k-1,i} \right)}{l+1} - 0.5 \right) \text{Sign} \left(x_{(k,i-1)l+j} \right) \right] \tag{2.31}$$

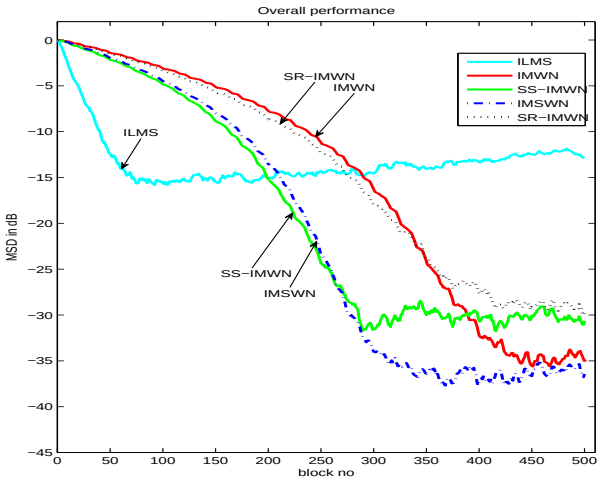

Figure 2.3: Overall convergence performance with 50% outliers in the desired data with magnitude between (-10,10) for input data between (-0.1,0.9)

Without loss of generality we can assume that the input data in a block is in the same order as the rank of the error. It means that if $\mathbf{x}_{(i-1)l+1} < \mathbf{x}_{(i-1)l+2} < \cdots < \mathbf{x}_{il}$ then $\mathrm{R}\left(e_{k,(i-1)l+1}\right) < \mathrm{R}\left(e_{k,(i-1)l+2}\right) < \cdots < \mathrm{R}\left(e_{k,il}\right)$. It is observed that the score value $R\left(e_{k,(i-1)l+1}\right)/(l+1) - 0.5$ is negative for $j = 1, ..., l/2$ and positive for $j = l/2+1, ..., l$. In addition

$$\left(\mathrm{R}\left(e_{k,(i-1)l+j}\right)/(l+1) - 0.5\right) = -\left(R\left(e_{k,il-j+1}\right)/(l+1) - 0.5\right) \tag{2.32}$$

The input data is generated uniformly between $(-0.5, 0.5)$. So the input data corresponding to the lower and upper halves of the error are negative and positive respectively. Thus the negative and positive inputs are multiplied with the negative and positive score values respectively. Hence the gradient value is maximum and convergence speed becomes fast.

Consider for the case of bias input data. Suppose the input data is random and lies between $(-0.1, 0.9)$. According to the assumption the input data in a block is in the same order as that of the rank of the error. The gradient in this case is less than the gradient for the data between $(-0.5, 0.5)$. This is due to the following factors: (i) In this scenario input data between (-0.1, 0.4) and (0.4, 0.8) have negative and positive

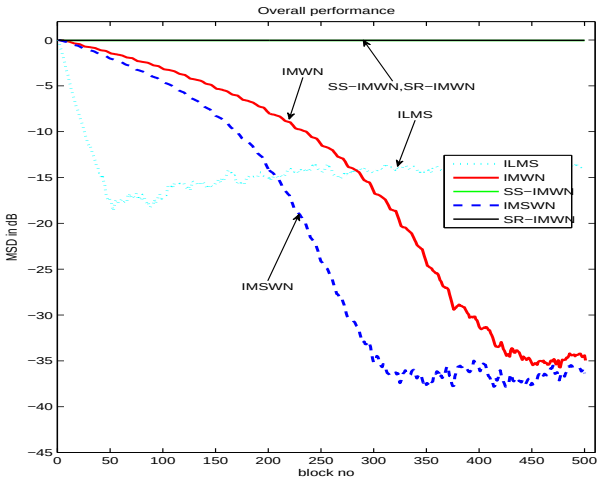

Figure 2.4: Overall convergence performance with 50% outliers in the desired data with magnitude between (-10,10)for input data between (0,1)

sores respectively. Using (2.32) it is observed that for every score corresponding to the input value between (0.4, 0.8) there will be a score of same magnitude with opposite sign for the input data between (0, 0.4). Since the score is multiplied with the sign of the input data and the sign is same for both the cases these two scores canceled each other during addition in the update equation. (ii) Both score and sign for the data between (-0.1, 0) and (0.8, 0.9) are negative and positive respectively. Thus the multiplication of score with the sign of the input data for this case is positive. Hence only these data are responsible for the gradient factor. Thus it is less than for the data between (-0.5, 0.5).

As bias magnitude increases the data corresponding to factor (i) increases and factor (ii) decreases. Thus gradient decreases. When all the input data are either positive or negative at that time all data belong to factor (i). Hence the gradient value is zero and the algorithm does not converge to the optimum value.

In order to alleviate these drawbacks a new norm is proposed in the next section. Its computational complexity is more than the previous norm but provide faster convergence speed.

2.6 Proposed Robust Modified Wilcoxon Norm Based Learning Strategy

In this section a modified Wilcoxon norm is proposed to overcome the problem mentioned in the previous section. The new norm is studied in details and a distributed strategy is proposed. Similar to the Wilcoxon norm the modified Wilcoxon norm is also applied to a set of data. In the present case the error difference instead of error is used so that the score value of the modified Wilcoxon norm changes more compared to the score value of the Wilcoxon norm with change of some elements present in the set. The proposed modified Wilcoxon norm of the vector (2.6) defined by

$$\|\mathbf{v}\|_{mwil} = \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \varphi\left(\mathrm{R}\left(v_i - v_j\right)\right)\left(v_i - v_j\right) \tag{2.33}$$

where φ is the score function defined in (2.8). The score function is applied over the difference of two vector elements. The function (2.33) is shown in Section A.1 as a pseudo-norm. By taking the difference of error terms the sensitivity of the norm is increased compared to the Wilcoxon norm. Similar to the SR-IMWN and SS-IMWN, the modified minimum Wilcoxon norm is used to propose SS-IMMWN and SR-IMMWN to design other fast distributed algorithms. The score function is defined as

$$\|\mathbf{v}\|_{mwn} = \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \sqrt{12}\left(\frac{\mathrm{R}\left(v_i - v_j\right)}{0.5 \times l(l-1)} - 0.5\right)\left(v_i - v_j\right) \tag{2.34}$$

The sign-modified-Wilcoxon-norm is given by

$$\|\mathbf{v}\|_{mwn} = \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \mathrm{Sign}\left(\frac{\mathrm{R}\left(v_i - v_j\right)}{0.5 \times l(l-1)} - 0.5\right)\left(v_i - v_j\right) \tag{2.35}$$

The estimation of the required parameters is based on the gradient of the modified Wilcoxon norm of the residual errors defined as

$$\|\mathbf{e}\|_{mwn} = \left\|\mathbf{y} - \mathbf{X}^T\mathbf{w}\right\|_{mwn} \tag{2.36}$$

In case of incremental strategy the parameters are estimated at the individual node taking the parameters from the previous node as a priori information and this is updated at the node using its own data. For updation the gradient based technique is used. Hence similar to (2.18), the time update of the parameters is given by

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \mu\nabla \|\mathbf{e}\|_{mwn} \tag{2.37}$$

Equation (2.37) is divided to N spatial updates as in (2.20) and the corresponding update at the k^{th} node is given by

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\psi}_{k-1,i} - \nabla \|\mathbf{e}_{k,i}\|_{mmwn} \tag{2.38}$$

Using the gradient of proposed norm, the spatial update equation is obtained as

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\psi}_{k-1,i} - \mu \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \sqrt{12} \left(\frac{\mathrm{R}\left(e_i - e_j\right)}{0.5 \times l(l-1) + 1} - 0.5 \right) (\mathbf{x}_j - \mathbf{x}_i) \tag{2.39}$$

where the second term in the right hand side of (2.39) is the instant gradient computed using the data present at the node k. The error term is given by

$$e_i = y_i - \mathbf{x}_i^T\mathbf{w} \tag{2.40}$$

2.6.1 Sign Regressor Modified Wilcoxon Norm and Sign Sign Modified Wilcoxon Norm

Similar to that of (2.27-2.30), the update equation for SR-IMMWN and SS-IMMWN are obtained from (2.39) as

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \left(\sqrt{12} \left(\frac{\mathrm{R}(e_i - e_j)}{0.5 \times l(l-1)+1} - 0.5 \right) \times \mathrm{Sign}\left(\mathbf{x}_i - \mathbf{x}_j\right) \right) \tag{2.41}$$

and

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \left(\sqrt{12}\mathrm{Sign} \left(\frac{\mathrm{R}(e_i - e_j)}{0.5 \times l(l-1)+1} - 0.5 \right) \times \mathrm{Sign}\left(\mathbf{x}_i - \mathbf{x}_j\right) \right) \tag{2.42}$$

respectively.

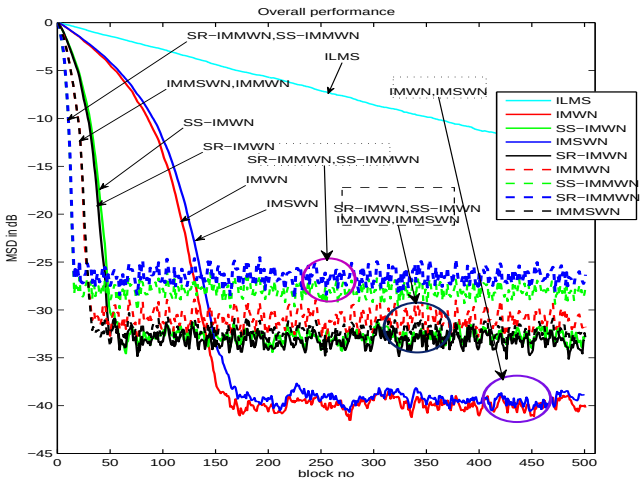

Figure 2.5: Overall convergence performance with 10% outliers in the desired data with magnitude between (-10,10)

2.7 Learning Behavior of IMMWN

This section deals with simulation based experiments to assess the robust estimation performance of the IMMWN. For simulation purpose number of sensor nodes, a number of random input data, parameters, output generation are similar to the Section-2.5. Simulations are carried out for different percentage and strength of outliers. But only the simulation results for 10% and 50% of outliers having magnitude 10 are shown. The block length considered during simulation is 40. The simulation

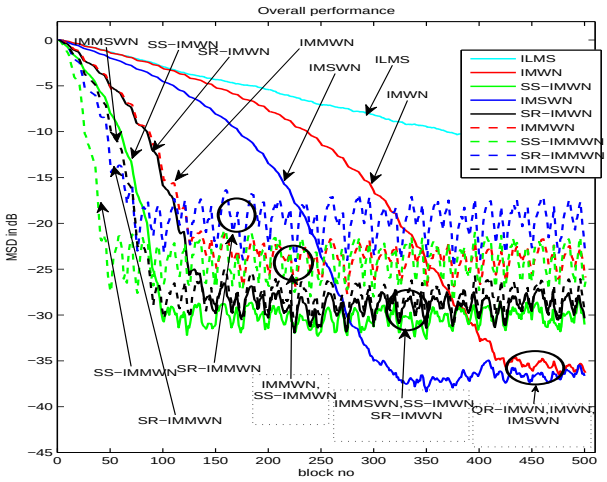

Figure 2.6: Overall convergence performance with 50% outliers in the desired data with magnitude between (-10,10)

results shown are averaged over 20 independent experiments. The step size is 0.0002.

Comparative convergence performance among the ILMS,IMWN,IMSWN,SSIMWN, SRIMWN, IMMWN,IMMSWN, SR-IMMWN and SS-IMMWN are plotted in Figures 2.5-2.8. From these plots it is evident that the IMMWN,IMMSWN, SS-IMMWN and SR-IMMWN are robust against outliers in the desired data in the presence of both biased and unbiased input data. The IMMWN and its variants provide better convergence speed compared to the IMWN and its variants, even though the step size of IMMWN is less than the step size of IMWN. As the percentage of outliers increases, both the convergence speed and the steady state performance of all the methods decreases. In Table-2.1 the steady state performance for all the proposed methods are given for different block sizes. For this steady state performance comparison, the parameter of the environment is $\left[\begin{array}{ccccc} 1/\sqrt{5} & 1/\sqrt{3} & 1/\sqrt{5} & 1/\sqrt{3} & 1/\sqrt{5} \end{array} \right]$. For the same parameters the steady state performance between 5, 10 and 15 number of sensor nodes based wireless sensor network are given in Table-2.2. In this case some parameters are different from the other parameters. Centralized performance of the proposed algorithm is given in the Figures 2.9, 2.10. Comparing simulation results of the centralized method with the distributed method it is found that the performance of both the algorithms is same. Because the centralized cost function is the sum of

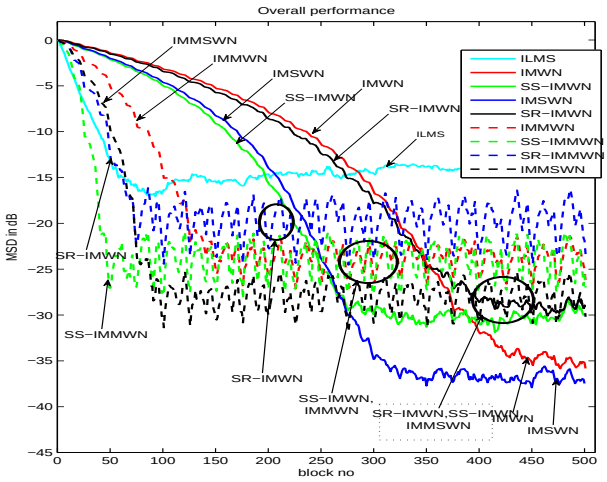

Figure 2.7: Overall convergence performance with 50% outliers in the desired data with magnitude between (-10,10)for input data between (-0.1,0.9)

Algorithms	5	10	15	20	25	30	35	40
ILMS	-20.2801	-22.3251	-24.0092	-21.6554	-26.4973	-21.1574	-17.5870	-21.9188
IMWN	-40.1491	-39.7431	-41.9459	-41.0735	-39.8789	-38.8731	-39.5113	-39.6202
IMSWN	-38.1951	-39.1210	-40.3224	-40.0848	-38.9507	-39.4155	-39.5456	-39.3503
SRIMWN	-34.1768	-33.8340	-33.6544	-33.5863	-32.2761	-33.5502	-32.7823	-33.2312
SSIMWN	-32.5545	-33.2619	-34.3665	-33.8093	-31.6090	-32.7876	-33.0385	-32.8978
IMMWN	-42.4013	-36.8849	-36.8167	-34.3093	-33.4830	-32.4537	-30.7164	-30.8123
IMMSWN	-42.8775	-39.5286	-38.3377	-35.1569	-34.5851	-33.6229	-32.3705	-31.9942
SR-IMMWN	-36.8815	-34.0472	-32.9454	-30.6145	-29.5798	-28.8233	-26.9214	-26.3592
SS-IMMWN	-38.3421	-35.6921	-34.7523	-31.5365	-30.5074	-29.9765	-28.6574	-27.6969

Table 2.1: Comparison of steady state performance for different block sizes

all the local cost function present in the every sensor node.

2.8 Computational Complexity

The Wilcoxon norm depends on the rank of the element. The rank of an element in a group of data can be calculated using the indicator function. Let $\mathrm{R}(v_i), i = 1, \cdots, L$ be the rank of the element v_i in the vector $\mathbf{v}$ defined in (2.6). The rank of the element can be calculated using the indicator function as

$$\mathrm{R}\left(v_i\right) = \sum_{j=1}^{L} \mathrm{I}_{(v_i - v_j)} \tag{2.43}$$

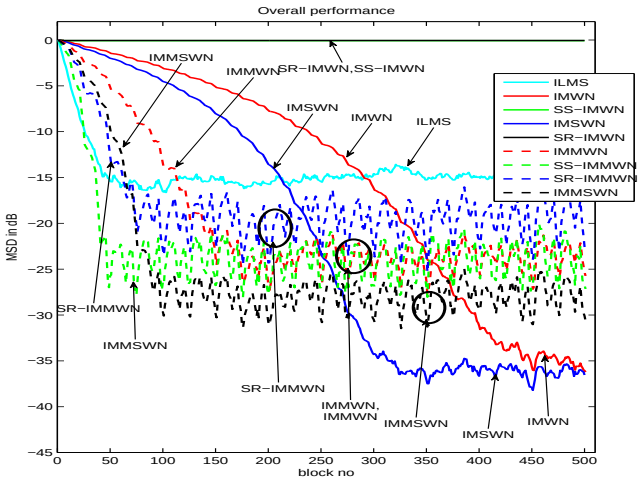

Figure 2.8: Overall convergence performance with 50% outliers in the desired data with magnitude between (-10,10)for input data between (0,1)

Algorithms	5	10	15
ILMS	-6.0360	-12.0854	-21.5233
IMWN	-40.8821	-41.1476	-41.4604
IMSWN	-40.5875	-41.5194	-40.4329
SS-IMWN	-34.2807	-34.5988	-33.2432
IMMWN	-31.6626	-32.0614	-32.2958
IMMSWN	-33.1592	-33.5503	-33.6202
SR-IMWN	-27.3415	-27.9644	-27.6364
SS-IMWN	-28.7503	-29.3808	-29.1598

Table 2.2: Comparison of steady state performance for different block sizes

where I_p is the indicator function given as

$$\mathrm{I}_p = \begin{cases} 1 & p \geq 0 \\ 0 & p < 0 \end{cases} \tag{2.44}$$

The number of operations required to compute the norm in (2.44) is to computed.

$$\|\mathbf{v}\|_w = \sum_{i=1}^{L} \sqrt{12} \left(\frac{\mathrm{R}\left(v_i\right)}{L+1} - 0.5 \right) v_i \tag{2.45}$$

Since the rank of one element requires $l-1$ number of comparisons then to calculate the rank of all elements $0.5 \times l(l-1)$ number of comparisons are required. In order

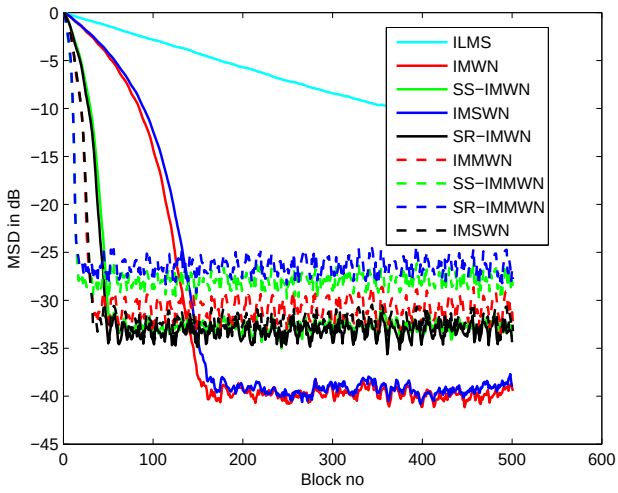

Figure 2.9: Overall convergence performance with 10% outliers in the desired data with magnitude between (-10,10)for input data between (0,1)

to calculate the score value l number of divisions and l number of subtractions are to be performed. However, this score needs to be calculated once in one block. Since the score value is same for other blocks, depending upon the rank of the data the corresponding score value is to be assigned. Thus for other blocks the score can be calculated using only $0.5 \times l(l-1)$ number of comparisons. In order to calculate one error element in a block p number of multiplications and $p-1$ number of additions are required. Hence for a block of error terms lp number of additions or subtractions and lp number of multiplications are required. During the update operation $2lp$ number of multiplications and lp number of additions are needed. Thus the total operations required to compute (2.45) are: $0.5 \times l(l-1)$ number of comparisons; $3lp$ number of multiplications; and $2lp$ number additions.

In a similar way computation of (2.34) requires $(l^2-1)^2$ number of comparisons, $3l(l-1)$ number of multiplications and $2l(l-1)p$ number of additions. The comparisons between the two algorithms are given in Table 2.3.

2.9 Conclusion

The chapter proposes different robust distributed gradient descent algorithms which are essentially based on the Wilcoxon norm. The Wilcoxon norm and sign Wilcoxon

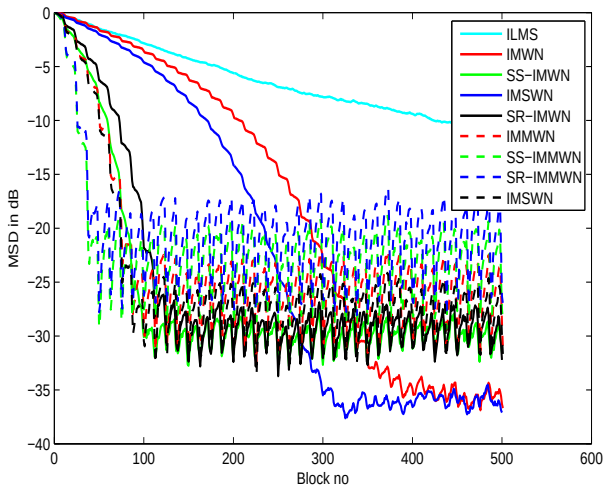

Figure 2.10: Overall convergence performance with 10% outliers in the desired data with magnitude between (-10,10)for input data between (0,1)

Table 2.3: Comparison of Computational Complexity

Operation	IMWN	IMMWN
Comparisons	l^2	$(l^2-l)^2$
Additions	$2lp$	$2l(l-1)p$
Multiplications	$3lp$	$3l(l-1)p$

l = length of the bock
p = order of the system

norm are minimized in a distributed manner to handle outliers in the desired data. It is observed to provide slow convergence speed. In order to increase the convergence speed during training sign regressor and sign sign Wilcoxon norms based algorithms have been proposed. However, these algorithms yield poor performance in the presence of biased input data. In order to circumvent this, a novel modified Wilcoxon norm is also proposed in the distributed algorithms. Simulation results demonstrate that the proposed methods are robust against outliers in the desired data. The computation complexity of the various algorithms have been assessed and compared. It is observed that the IMWN and IMMWN require less number of computations compared to the QR-IMWN.

The proposed algorithms can easily be extended to diffusion based distributed implementation. Interested readers can also extend the study for finding out the convergence and steady state analyses based upon the theory of the rank test. More

sophisticated robust algorithms can also be designed by examining the analogy between the IMWN with other LMS variant algorithms such as normalized LMS and variable step size LMS.

Chapter 3

Robust Incremental Adaptive Strategies to Handle Outliers in Both Input and Desired Data

In previous chapter some novel distributed algorithms are proposed to handle outliers in the desired data. In some cases the input data is also affected by outliers. This chapter employs the generalized-rank (GR) technique is as cost function instead of least square error cost function to control the effects of outliers present in both input and desired data. A novel indicator function and median based approach are proposed to decrease the computational complexity requirement at the sensor nodes. Further, to increase the convergence speed a sign regressor GR norm is also proposed and used. Simulation based experiments show that the performance obtained using proposed methods are robust against outliers in the desired and input data. In order to enhance both the performance and convergence speed, high breakdown estimator is also proposed and improved performance is demonstrated.

3.1 Introduction

Robust methods developed in the previous chapter are robust against outliers in the desired data. However, the input data along with the desired data is also affected by the outliers. The data is affected by outliers due to the following reasons:

1. In addition to the additive white Gaussian noise(AWGN), impulsive noise, whose source is atmospheric noise [28], is also present in the input and/or desired data;

2. Since the input and desired data are collected using transducers, the measured data may also be affected by outliers due to temporal instrumental error, which may occur at the time of measurement;

3. In general, data is processed using finite length registers that leads to quantization errors.

This type problem has been addressed in the literature [26], where total least square is used. However, it is assumed that the strength of the data uncertainty is bounded, which may not occur in the real environment due to the presence of impulsive noise [28]. Therefore, it seems natural to use robust cost functions based methods for distributed estimation of environmental parameters. Robust cost functions are

broadly classified into three major groups such as M-type, L-type and R-type [12, 13, 15]. Among the M-estimators, the GM and MM estimators [13] are often used by statisticians to handle outliers in both desired and input data. The former is a generalized method of M-estimators which depends upon some predefined parameters that affect its performance. The predefined parameters depend on the deviation of the noise distribution from the Gaussian assumption. Since the data is distributed among the nodes, estimation of these parameters requires more communication and computation energy. In simple form the estimator is not scale equivariant. In order to make it scale equivariant a robust scale factor needs to be used. Hence the scale equivariant method requires estimation of the robust scale factor and then estimation of the parameters using this robust scale, which needs more computation at the sensor node. In addition to higher computational and communication overheads, the GM estimator provides inferior performance in the presence of heavy tailed noise. The MM estimator is more suitable for a heavy tailed noise scenario. In addition it employs a redescending function [13], which may trap the soution to a local minima. In order to avoid this problem, the scale based estimators such as median absolute deviation (MAD) or the least median squares (LMS) estimators may be used at the beginning of the estimation to get a parameter near to the global minima and then switch to the MM estimator to achieve an improved estimation performance. Although this method gives good estimation accuracy in the presence of heavy tailed noise, it also suffers from the problem of fine tuning of predefined parameters and estimation of scaling factor similar to the GM estimator. Thus these M-estimators are not suitable for implementation in distributed sensor network scenario.

The L-estimators are themselves scale equivariant and do not depend upon any predefined parameters but suffer from poor estimation performance [12, 15]. On the other hand, the R-based methods are scale equivariant, independent of any predefined parameters and offer good estimation capability. Hence these type of estimator have drawn the attention of the signal processing community. The Wilcoxon norm which belongs to the R-estimators has been introduced in designing robust learning machines [37] and in designing robust system identification [32]. However, this is only

robust against outliers in the desired data. The GR estimator which comes under R estimators is a good technique to estimate the regression parameters in the presence of outliers both in the desired and input data [16]. It does not depend upon any predefined parameters and also provides better estimation accuracy. Therefore the GR estimator is chosen in this chapter and appropriately used for distributed intended parameters.

In this chapter we have developed two new algorithms using the GR technique for distributed estimation of the parameters in presence of outliers in the input as well as in the desired data. All these techniques are implemented using incremental distributed strategy. The first technique is a direct block implementation of the GR estimator in the incremental distributed strategy using a gradient based method. A novel median based approach is used here which takes less computational energy of the sensor nodes. The second one is a sign regressor GR estimator which yields better convergence speed but its performance is slightly inferior to the previous one. Furthermore to increase the performance and convergence of these two proposed methods a median based high breakdown estimator is also proposed. However, this new technique requires relatively more computation than the previous methods.

The bold capital and small letters are meant for matrices and vectors, respectively and realization of elements in a matrix or vector are denoted by small letters. The input data at node k and time instant i is represented by $\mathbf{x}_{k,i}$, whereas the entire spatial input data in the environment at same time instant i is $\mathbf{X}_i$. The entire spatio-temporal data from the first iteration to n^{th} iteration is represented by $\mathbf{X}(n)$. The notation for desired data is similar to the input but with a letter y.

3.2 Problem Formulation

As described in Section 2.2 the entire spatio-temporal input and desired data can be related as

$$\mathbf{y}(n) = \mathbf{X}^T(n)\,\mathbf{w} + \mathbf{v}(n) \tag{3.1}$$

Due to the presence of outliers, the available input is a corrupted version of the original input. The corrupted version of input is given by $\mathbf{X}^c(n)$. When outliers are presented in the input data the matrix version of the measured input is given as

$$\mathbf{X}^c\left(n\right) = \mathbf{X}^T\left(n\right) + \mathbf{D}\left(n\right) \tag{3.2}$$

where $\mathbf{D}(n)$ is the outliers present in the input data. The objective is to estimate $\mathbf{w}$ from $\mathbf{y}(n)$ and $\mathbf{X}^c(n)$. In the presence of outliers, the estimation problem can be formulated as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\| \mathbf{y} - \left(\mathbf{X}^c\right)^T \mathbf{w} \right\|_{norm} \tag{3.3}$$

In terms of geometry, (3.3) can be viewed as to estimate $\mathbf{w}^*$ so that norm distance between the estimated output $(\mathbf{X}^c)^T\mathbf{w}^*$ and the desired output is minimized.

3.3 Generalized Rank (GR) Norm

In order to obtain the estimate $\mathbf{w}$ present in the problem (3.3), an incremental minimum generalize rank norm is proposed. To achieve this objective let us first discuss the Wilcoxon norm [16]. This section deals with the sensitivity and robustness behavior of rank based methods taking one example, i.e. the Wilcoxon norm which is demonstrated in Section 2.2.1. This is a very simple norm based upon the rank order statistics. Though this norm is merely robust against outliers in the desired data [16, 37], using the idea behind this robustness property, it is possible to devise an improved rank based norm to handle outliers both in the input and desired data.

The Wilcoxon norm of the vector $\mathbf{v}$ is shown in (2.7). The Wilcoxon norm is robust against outliers in the desired data. In order to understand the logic behind the robustness, its sensitive analysis is required [16]. Sensitivity of a norm defines the rate of change in norm value with respect to change in the error value. If the sensitivity of a norm is a bounded function of the error then the norm is robust. As the Wilcoxon norm in (2.7) depends upon the magnitude and the rank order of the

error, a small change in error changes its magnitude part but not the rank. Thus its sensitivity is less and the norm is robust. However in the case of the l_2 norm, which is used in [5,7,10,11], the square of the errors is to be minimized. Therefore, the change in norm value is the square times the change in error value. Thus the norm is very sensitive to the error. Therefore, the solution is biased towards the data yielding a large error which is essentially due to the outliers in the desired data.

However, both the Wilcoxon and l_2 norms are not robust against outliers present in the input data. This is explained as follows. Consider a system with its output as the convolution of the input and system parameter vectors [34, 35]. In this case the vectors of $\mathbf{X}(n)$ in the model (3.1), is the tap delay version of the input samples. Hence, if one sample is affected by an outlier it affects at total p estimated outputs. This is similar to the occurance of a burst error in the desired output. Moreover, if two or more consecutive input samples are affected by outliers, then the estimated output value is due to the additive effect of such input outliers. Hence under such condition it is difficult to estimate the parameters in(3.3) using either l_2 or Wilcoxon norm. However this effect can be mitigated by studying absolute magnitude of the error difference. The Wilcoxon norm (3.2) for the vector $\mathbf{v}$ in (2.7) can be expressed in an alternative form as

$$\|\mathbf{v}\|_w = \frac{\sqrt{3}}{L+1} \sum_{i<j} |v_i - v_i| \tag{3.4}$$

The proof of representation as in (3.4) is given in the Section B.1. An indicator function based approach is used in this derivation. It may be noted that this indicator function is the basis of deriving our proposed algorithms.

It can be observed from (3.4) that in case of the Wilcoxon norm, the weightage for different absolute error deviation is same ,i.e. $\frac{\sqrt{3}}{L+1}$. This indicates that the Wilcoxon norm gives equal importance to every absolute error difference. Since the error is related to the desired and input in affine manner, the notion that the contribution of the Wilcoxon norm is the same for every absolute error difference, can also be extended to both input and desired signals. In other words the Wilcoxon norm provides the

same importance to every absolute input and desired differences. Therefore, the corruption of the input data leads to a new absolute input difference which is more than the previous input difference. The aforesaid explanation indicates that outliers in the input lead to change in the absolute error difference. On the other hand, taking different coefficients in (3.4) for different absolute error deviation, the effect of outliers in input data can be decreased [16]. Using this important concept robust distributed algorithms are designed to handle outliers both in the input and desired data.

The function defined in (3.4) with different coefficients, is called generalized rank norm. The GR norm of the vector $\mathbf{v}$ is

$$\|\mathbf{v}\|_{GR} = \sum_{i<j} (w_{ij} |v_i - v_j|) \tag{3.5}$$

This can be proved to be a pseudo norm for $w_{ij} > 0$ and $w_{ij} = w_{ji}$ [16]. Considering the structure of the input data, i.e. tap delay structure, a median based approach is used to calculate w_{ij}. The median based approach and indicator function are the novelty of the proposed robust design. For distributed implementation of the generalized R norm for WSNs the block formulation of the problem is needed.

3.4 Proposed Method of Distributed Parameter Estimation using Generalized R Norm

In order to estimate the parameter $\mathbf{w}$ the GR norm defined in (3.5) is used. In the present case the objective is

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\| \mathbf{y} - (\mathbf{X}^c)^T \mathbf{w} \right\|_{GR} \tag{3.6}$$

Since $\mathbf{X}^C(n)$ and $\mathbf{y}(n)$ contain the entire spatio-temporal data up to time n, the solution of (3.6) can only be achieved by assessing the entire data to one node. This requires a large number of communication overheads similar to that which occurs in classical methods. But the incremental method of distributed approach requires that

$$\left\|\mathbf{y} - \mathbf{X}^C\mathbf{w}\right\|_{GR} = \sum_{i=1}^{N} \left\|\mathbf{y}^i - (\mathbf{X}^{i,C})^T\mathbf{w}\right\|_{GR} \tag{3.7}$$

where, $\mathbf{y}^i$ and $\mathbf{X}^{ci}$ are the desired and input data sets at the i^{th} node. Since GR norm is a nonlinear function of errors, it is impossible to achieve the global optimum solution (3.6) by incremental distributed strategy. On the other hand to design a global cost function, which (i) should use global information, (ii) should be implemented by incremental method and (iii) should be robust against outliers in desired and input signal the local minimum generalized R cost function is defined. It is given by

$$\min_{\mathbf{w}} \left\|\mathbf{y}^i - (\mathbf{X}^{i,C})^T\mathbf{w}\right\|_{GR} \tag{3.8}$$

Based on the local cost function(3.8), the global cost function is defined as

$$\min_{\mathbf{w}} \sum_{j=1}^{N} \left\|\mathbf{y}^j - (\mathbf{X}^j)^T\mathbf{w}\right\|_{GR} \tag{3.9}$$

The solutions of (3.6) and (3.9) are not the same. The new cost function is the sum of all the generalized R cost functions at individual nodes. Since the cost function at every node is a norm then sum of all cost functions in(3.9), which is an affine function of each cost function, can also be shown as a pseudo norm by using convexity properties [38]. Solution of this cost function can be obtained by an incremental strategy [5]. In order to achieve this, each node is to find the minimum generalized R solution for the parameter $\mathbf{w}$ and then transmit this estimation to the next node. The present node employs the estimated values of the previous node as priori information and then uses its own data to further refine the estimation. However to calculate the local cost function(3.8) it is required to wait until the end of the measurement process. This requires more computation and also the process is very slow. In order to circumvent this problem, the local cost function is again modified to a block sum of cost function whose optimal solution can be calculated

using block processing of the data.

To facilitate such processing the local cost function is defined as

$$\min_{\mathbf{w}} \sum_{k=1}^{n/L} \left\| \mathbf{y}_{i,k} - (\mathbf{X}_{i,k,c})^T \mathbf{w} \right\|_{GR} \tag{3.10}$$

This cost function is not equal to the cost function defined as(3.8). The difference between (3.8) and (3.10) is similar to the difference between the global cost functions defined in (3.6) and (3.9). Taking the sum of all local cost functions defined as in (3.10), the global cost function is obtained as

$$\min_{\mathbf{w}} \sum_{k=1}^{n/L} \sum_{j=1}^{N} \left\| \mathbf{y}_{k,j} - \mathbf{X}_{k,j}^T \mathbf{w} \right\|_{GR} \tag{3.11}$$

The present objective is to get the optimal solution for (3.11) using incremental distributed strategy. Here the block formulation of the problem is used which is given in Section 2.2.2. Collecting all the spatial data from the first block measurement time to the p^{th} block measurement time into a matrix, we get

$$\mathbf{X}(p) = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_p \end{bmatrix} \tag{3.12}$$

and the corresponding measured data which acts as desired vector as

$$\mathbf{y}(p) = \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots & \mathbf{y}_p^T \end{bmatrix}^T \tag{3.13}$$

Let the estimated parameters using (3.12)and (3.13) be $\mathbf{w}_p$. Similarly $\mathbf{w}_{p+1}$ is obtained using $\mathbf{X}(p+1)$ and $\mathbf{y}(p+1)$. However, with time the size of input and desired data increases. Hence large number of memory space and computation unit is required to estimate the optimum parameters. In order to avoid this an iterative estimation method is introduced by which $\mathbf{w}_{p+1}$ is estimated from the previous estimate $\mathbf{w}_p$ and the new data set at the $(p+1)^{th}$ block measurement time. Since the

data collected during $(p+1)^{th}$ block measurement time is present in every sensor node throughout the environment, the incremental method is used to achieve the iterative estimation of the parameters. Similar to the incremental method given in Section 2.3, every sensor uses its previous estimate which is received from the previous node and its own data. Further to investigate this spatial iterative formulation, let the spatial estimated parameters at node k during $(p+1)$ iteration be represented as $\boldsymbol{\psi}_{k,p+1}$.

Therefore in an incremental distributed strategy during the $(p+1)^{th}$ spatial recursion, the $(k)^{th}$ node transmits the data $\boldsymbol{\psi}_{k,p+1}$ to the $(k+1)^{th}$ node. Using $\boldsymbol{\psi}_{k,p+1}$ and the $(p+1)^{th}$ measured data, the $(k+1)^{th}$ sensor node calculates $\boldsymbol{\psi}_{k+1,p+1}$. Using the gradient descent technique the iterative estimation is given by

$$\boldsymbol{\psi}_{j+1,k+1} = \boldsymbol{\psi}_{j+1,k} - \mu \nabla J\left(\boldsymbol{\psi}_{j+1,k}\right) \tag{3.14}$$

The GR norm is used in (3.14) to update the parameters. Applying the generalize R norm for vector of errors in $(k+1)^{th}$ node we get

$$J_{j+1,k}\left(\boldsymbol{\psi}_{j+1,k}\right) = \sum_{l<m} w_{lm} \left| e_{j+1,k+1,l} - e_{j+1,k+1,m} \right| \tag{3.15}$$

where

$$e_{j+1,k+1,l} = y_{j+1,k+1,l} - \mathbf{x}_{j+1,k+1,l}^{T} \boldsymbol{\psi}_{j+1,k} \tag{3.16}$$

.

and

w_{lm} is the weighting factor,(the procedure for finding this is given in the subsequent subsections) which depends upon the input $\mathbf{x}_{j+1,p+1}^{c}$. This factor helps to decrease the sensitivity of the outliers in the input data.

Taking the gradient of the cost function with respect to the parameter $\mathbf{w}_p$ and then using this gradient to update the parameter, the final update equation is obtained as

$$\begin{aligned}\boldsymbol{\psi}_{j+1,k} = \boldsymbol{\psi}_{j+1,k} + \mu \sum_{i<j} w_{ij} \text{sign}\left(\left(y_{k+1,j} - \mathbf{x}_{k+1,j}^T \boldsymbol{\psi}_{k,j}\right)\right. \\ \left. - \left(y_{k+1,j} - \mathbf{x}_{k+1,j}^T \boldsymbol{\psi}_{k,j}\right)\left(\mathbf{x}_{k+1,j} - \mathbf{x}_{k+1,j}\right)\right)\end{aligned} \tag{3.17}$$

Following the above procedure for all the sensor node starting from node 1 to node N during $(p+1)$ block measurement time, $\mathbf{w}_{p+1}$ is calculated from $\mathbf{w}_p$ as

$$\mathbf{w}_{p+1} \equiv \boldsymbol{\psi}_{N,p+1} \leftarrow \cdots \boldsymbol{\psi}_{2,p} \leftarrow \boldsymbol{\psi}_{1,p} \equiv \mathbf{w}_p \tag{3.18}$$

3.4.1 Calculation of w_{ij} in GR Norm

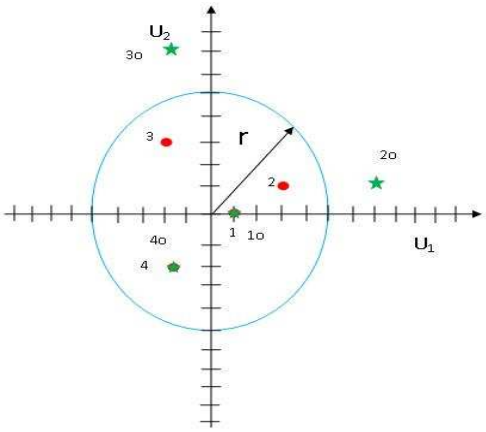

Figure 3.1: Input data in multidimensional space with and without outliers

This subsection describes the method for finding of the weighting factor w_{ij} present in (3.18). This is responsible for the stability of the algorithm against outliers in the input space [16]. Since the input data contributing to the output error is a vector of dimension p the investigation of w_{lm} requires multivariate outliers analysis of dimension p. Multivariate outliers analysis is based on the multivariate distribution of the random vector variable of interest. In the present case the multivariate random variable is a vector random variable of input data. It may be denoted that the multivariable distribution is the position distribution of time realization vectors corresponding to the multivariable of interest in multidimensional space. The input

data free from outliers remains close to each other in multidimensional space, where as data affected by outliers remains away from the correct data as well as from each other depending upon the strength of the outliers. Parametrically a vector is judged whether it is affected by outliers by threshold value which depends on some norm distance from the center of points having no outliers to the point of interest. In order to achieve this the first step is to identify the input vectors not affected by outliers. Since the points not affected by outliers remain close to each other, a shape defined by some parameters can be formed enclosing these points. In other words the shape should enclose the convex hull of the points not affected by outliers. For this purpose, an ellipsoid is a suitable geometry because (i) it is being used for clustering, (ii) it gives less error to enclose a random structure and (iii) it can be implemented using the Khachiyan algorithm [39]. Hence if a minimum volume ellipsoid is formed enclosing the points not affected by outliers then using the parameters of the ellipsoid we can differentiate the points affected by outliers. Furthermore, the strength of the outliers from the distance of the point of interest from the center of the ellipsoid can also be measured. This distance information is a key parameter in designing the weighting factor.

An ellipsoid is defined by a matrix, say $\mathbf{Q}$, that describes the spread and orientation of the ellipsoid in different direction, and a vector, say $\mathbf{c}$, that represents the center of the ellipsoid. With these parameters an ellipsoid is defined as

$\chi = \left\{ \mathbf{x} \middle| (\mathbf{x}-\mathbf{c})^T \mathbf{Q}^{-1} (\mathbf{x}-\mathbf{c}) \leq 1 \right\}$ where, $\mathbf{c} \in \Re^p$ and $\mathbf{Q} \in \Re^{p \times p}$.

For a vector of $\mathbf{x}$ of dimension p, $(\mathbf{x}-\mathbf{c})^T \mathbf{Q}^{-1} (\mathbf{x}-\mathbf{c})$ is called the Mahalanobis distance of the vector position from the center of the ellipsoid. Large Mahalanobis distance indicates that the point is affected by strong outliers. Since the points not affected by outliers remain close to each other the formation of a minimum volume ellipsoid covering any 50% of the data(assuming 50% of data are not affected by outliers), can differentiate outliers affected data from rest of the data. Therefore at first a minimum volume ellipsoid covering 50% of the input data is formed. Then Mahalanobis distance of each point using the parameters of the minimum volume ellipsoid is caculated. Subsequently the weight factor corresponds to each point is

obtained.

3.4.2 Construction of MVE Algorithm Corresponding to the Vectors Obtained From Tap Delay Filter

In most of the cases the system can be explained by a tap delay system [34,35]. Input data obtained from these system exhibits maximum relation with the previous and future input data. This similarity motivates to design efficient method for estimation of MVE. In the present case a two dimensioned vector is illustrated to facilitate a pictorial representation . However this method can easily be extended for vectors having more dimensions. Let us define a vector $\mathbf{u}$ as

$\mathbf{u}(i) = \begin{bmatrix} u_{1,i} & u_{2,i} \end{bmatrix}^T$

where $\mathbf{u}(i)$ refers to the i^{th} realization of the random vector variable $\mathbf{u}$. Then $u_{1,i}$ and $u_{2,i}$ refer to the i^{th} realization of the first and second elements of the random vector variable, respectively. If the position of the above random vector variable $\mathbf{u}$ is plotted in a space of dimension 2, then the first and second coordinate will be the first and second element of the random variable, respectively as given in Figure 3.1. Particularly, for a tap delay linear input system, the time realization of input random vector variable is obtained through the shifting and, then loading operation of a single dimension random variable, which is the input to the system. Let the input random variable corresponds to the input vector random variable be v. Thus, the i^{th} and $(i+1)^{th}$ realizations of the random vector variable $\mathbf{u}$ are given by

$$\mathbf{u}(i) = \begin{bmatrix} v(i) & v(i-1) \end{bmatrix}^T \qquad \mathbf{u}(i+1) = \begin{bmatrix} v(i+1) & v(i) \end{bmatrix}^T \tag{3.19}$$

respectively. From (3.19) it can be observed that the first vector of first coordinate is same as the second coordinate of the second vector. This observation can also be found for input vector variable of higher dimensions, e.g. for an input vector of dimension m, first element of the one vector is same as the second element of the next realization and also same with the third element of the third vector and so on. This observation indicates that for a tap delay system, the vectors take a position in multidimensional space with equal distribution along every coordinate. Similarly, if

an outlier affected input value $v^*(i)$ at any realization, say i , would enter the tap delay system then the realization of the input vector variable at the same instant and in next instant would be

$$\mathbf{u}(i) = \begin{bmatrix} v^*(i) & v(i-1) \end{bmatrix}^T \qquad \mathbf{u}(i+1) = \begin{bmatrix} v(i+1) & v^*(i) \end{bmatrix}^T \tag{3.20}$$

respectively. Therefore an outlier affects every coordinate with the same magnitude until it comes out of the system. Thus if the position of all these vectors would be plotted it looks symmetrical along every coordinate. Similar situation can also be explained for more than two dimensioned case. As a result affected and not affected data lie symmetrically with respect to the overall mean. Hence for this case the MVE enclosing the data not affected by outliers becomes a circle. The circle is one particular case of an ellipsoid in which all the off diagonal elements of the matrix $\mathbf{Q}$ are zero and diagonal elements are same. Hence, for such a tap delay system the objective is to find the center and radius of the circle covering 50% of the input vector data which may occur in extreme bad situation.

Median based technique differentiates the lower half elements of a data set from the upper half. This notion helps to use median based technique to find a circle enclosing 50% of the input data. For further explanation about the proposed method of finding data not affected by outliers, consider a vector $\mathbf{v} \in \Re^n$.

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \tag{3.21}$$

Take the median of the vector $\mathbf{v}$ to get the middle value element when the percentage of outliers is less than 50%. In this case the probability that the median value is not affected by outliers is maximum. In order to collect the points not affected by outliers take the median of the absolute difference between the median value and the elements. This median value gives the threshold value to decide whether an element is affected by outlier or not.

For vector of data this process can be carried out by relating the median value

of an individual vector with the median of the median of all vectors. Subsequently the radius and center of the circle is calculated using a median based procedure detailed below. Let n number of points say $\mathbf{u}_1$,$\mathbf{u}_2$,...,$\mathbf{u}_n$ be arranged in one matrix as $\mathbf{u} = \begin{bmatrix} \mathbf{u}_1^T & \mathbf{u}_2^T & \cdots & \mathbf{u}_n^T \end{bmatrix}^T$ Let $\eta_{\mathbf{u}_1}$ and $\eta_{\boldsymbol{\eta}_\mathbf{u}}$ represented as median($\mathbf{u}_1$) and median($\boldsymbol{\eta}_\mathbf{u}$) respectively, where $\boldsymbol{\eta}_\mathbf{u} = \begin{bmatrix} \eta_{\mathbf{u}_1} & \eta_{\mathbf{u}_2} & \cdots & \eta_{\mathbf{u}_n} \end{bmatrix}$ Let the difference of the median is given as $\chi_{\mathbf{u}_1} = \left|\eta_{\boldsymbol{\eta}_\mathbf{u}} - \eta_{\mathbf{u}_1}\right|$. Similarly let $\eta_{\boldsymbol{\chi}_u} = \text{median}\left(\boldsymbol{\chi}_\mathbf{u}\right)$, where $\boldsymbol{\chi}_\mathbf{u} = \begin{bmatrix} \chi_{\mathbf{u}_1} & \chi_{\mathbf{u}_2} & \cdots & \chi_{\mathbf{u}_n} \end{bmatrix}$. All the input data whose $\chi_{\mathbf{u}_1}$ is less than equal to $\eta_{\boldsymbol{\chi}_u}$ are collected into a vector u_new. Then the following two extreme values are computed $u_{new}^{max} = \text{Maximum}(u_{new})\&u_{new}^{min} = \text{Minimum}(u_{new})$ Subsequently the center and radious of the circle is obtained as $\mathbf{c} = \left(u_{new}^{max} + u_{new}^{min}\right)/2, \left(u_{new}^{max} + u_{new}^{min}\right)/2$ and $r = \left(u_{new}^{max} - u_{new}^{min}\right)/2$. Let $\mathbf{Q} = \text{Diag}\left\{\mathbf{1}_{1,N} \times r\right\}$. Using $\mathbf{Q}$, w_i is computed as [16]

$$w_i = \min\left\{1, \frac{\chi_{p,n-1}}{\sqrt{\left(\mathbf{u}_i - \mathbf{c}_i\right)^T \mathbf{Q}^{-1} \left(\mathbf{u}_i - \mathbf{c}_i\right)}}\right\} \tag{3.22}$$

Subsequently w_j is computed from (3.22) by replacing i by j. Finally w_{ij} is evaluated as

$$w_{ij} = w_i \times w_j \tag{3.23}$$

3.4.3 Stepwise Description of the Update Equation

Step1:Transmission of data from $k-1^{th}$ node is $\boldsymbol{\psi}_{k-1,i}$. Data from k^{th} node is $\mathbf{u}_{k,i}$ and $\mathbf{d}_{k,i}$.

Step2:Calculation of $w_{i,j}$ from the input data.

Step3:
$$\boldsymbol{\psi}_{k+1,i} = \boldsymbol{\psi}_{k,i} + \mu \sum_{i<j} w_{ij} \text{sign}\left(\left(y_{k+1,i} - \mathbf{x}_{k+1,i}^T \boldsymbol{\psi}_{k,i}\right)\right.$$
$$\left. - \left(y_{k+1,i} - \mathbf{x}_{k+1,i}^T \boldsymbol{\psi}_{k,i}\right)\left(\mathbf{x}_{k+1,i} - \mathbf{x}_{k+1,j}\right)\right)$$

Step4:Transmission to $k+1^{th}$ node is $\boldsymbol{\psi}_{k+1,i}$

3.5 Sign Regressor GR Estimator

The (3.5) can also be rewritten using an indicator function as shown below, whose derivation is established in Section B.2.

$$\|\mathbf{v}\|_{GR} = \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} \left(w_{ij} \mathrm{I}_{(e_i - e_j)} - w_{ij} \mathrm{I}_{(e_j - e_i)} \right) e_i \tag{3.24}$$

The gradient of the GR norm in the above stated form is

$$\sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} \left(w_{ij} \mathrm{I}_{(e_i - e_j)} - w_{ij} \mathrm{I}_{(e_j - e_i)} \right) \mathbf{x}_i \tag{3.25}$$

This leads to the update equation as

$$\boldsymbol{\psi}_{k+1,i} = \boldsymbol{\psi}_{k,i} + \ \mu \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} \left(w_{ij} \mathrm{I}_{\left(e_{k+1,i} - e_{k+1,j}\right)} - w_{ij} \mathrm{I}_{\left(e_{k+1,j} - e_{k+1,i}\right)} \right) \mathbf{x}_{k+1,i} \tag{3.26}$$

Changing the sum of vector scalar multiplication form that is in right rand side of (3.26) to a matrix vector multiplication form, it yields

$$\boldsymbol{\psi}_{k+1,i} = \boldsymbol{\psi}_{k,i} + \mu \mathbf{W}_{k+1,i} \mathbf{X}_{k+1,i} \tag{3.27}$$

As shown in [5], the incremental LMS update equation is

$$\boldsymbol{\psi}_{k+1,i} = \boldsymbol{\psi}_{k,i} + \mu e_{k+1,i} \mathbf{x}_{k+1,i} \tag{3.28}$$

Comparing (3.27) with (3.28), it is observed that the gradient term in both equations contain an input term multiplied with another term, i.e. a score function in case of GR norm and error value in case of incremental LMS. This observation concludes that the work of the score term in the case of the generalized R norm is similar to error term in the case of ILMS. This analogy motivates the use of other LMS variants in the GR norm with an objective to achieve good convergence speed or good

estimation performance. For example, it is known that sign regressor LMS is faster than the LMS with less estimation performance. Thus the question arises whether the notion of sign regressor LMS in the GR norm can be incorporated to get an algorithm that would provide faster convergence speed than the previous one and also would be robust against outliers in both input and desired data.

This has been investigated here. The update equation for the sign regressor LMS is

$$\boldsymbol{\psi}_{k+1,i} = \boldsymbol{\psi}_{k,i} + \mu e_{k+1,i} \text{sign}\left(\mathbf{x}_{k+1,i}\right) \tag{3.29}$$

Comparing (3.28) and (3.29), it is observed that the update equation for sign regressor LMS is same as the LMS except with sign of the input data. Since input data term is also there in GR norm, sign regressor GR norm is proposed by taking the sign of the input data in the update equation. Hence by taking the sign of the input data in (3.27), we get

$$\boldsymbol{\psi}_{k+1,i} = \boldsymbol{\psi}_{k,i} + \mu \mathbf{W}_{k+1,i} \text{sign}\left(\mathbf{X}_{k+1,i}\right) \tag{3.30}$$

Subsequently changing the sum of vector matrix multiplication term in (3.30) to sum of scalar vector multiplication term the desired form is obtained as

$$\boldsymbol{\psi}_{k+1,i} = \boldsymbol{\psi}_{k,i} + \mu \sum_{i<j} \text{sign}\left(e_{k+1,i} - e_{k+1,j}\right)\left(\text{sign}\left(\mathbf{x}_{k+1,i}\right) - \text{sign}\left(\mathbf{x}_{k+1,j}\right)\right) \tag{3.31}$$

3.6 Simulation Results and Discussions

This section deals with simulation based experiments to assess the robust estimation performance of the proposed methods. The number of nodes taken in the environment is 5. The model parameters to be estimated from the environment is $\left[\begin{array}{ccccc} 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} \end{array}\right]$. The input data generated is a zero mean uniformly distributed random number lying between $(-0.5, 0.5)$. The output data is

generated by passing the white input data through the model having feed forward parameters given above. The output data is added with additive white Gaussian noise having an SNR 30dB. Then the desired data is contaminated with outliers at random positions having magnitude randomly varying between $(-10, 10)$. Further the input data is also mixed with outliers continuously with a random magnitude between $(-3, 3)$. The block size is taken to be 40. The number of input samples used are 8000. Simulations have been carried out for separately with $10\%, 20\%, 30\%, 40\%, 50\%$ outliers in both input and desired data. However only the simulation results for 10% and 50% are shown in the chapter. The results presented are the average result over 50 independent experiments.

The overall convergence performance results at node 1 are shown in Figures 3.2-3.5. The step size chosen is 0.001. From the simulation results it is observed that incremental LMS and incremental minimum-Wilcoxon-norm do not lead to optimum convergence while incremental minimum GR norm and incremental minimum sign GR norm have the potentiality to achieve improved convergence in presence of outliers both in desired and input data. It can be observed that the convergence speed for incremental sign regressor minimum GR norm is faster than the incremental minimum GR norm but residual MSD is high. These comparative results are identical to convergence behavior between LMS and sign regressor LMS. As percentage of outliers increases both convergence speed and steady state performance of the proposed algorithms correspondingly decreases. However in presence of constant outliers in input data if the outliers is increased in the desired data the the performance decreases but convergence speed increases. This happens due to the masking effect of the desired outliers with these at input. Steady state performance between 5, 10 and 15 number of sensor nodes based wireless sensor network is given in the Table-5.1 . For this steady state performance comparison the parameter of the environment is $\left[\begin{array}{ccccc} 1/\sqrt{5} & 1/\sqrt{3} & 1/\sqrt{5} & 1/\sqrt{3} & 1/\sqrt{5} \end{array}\right]$. Here some parameters are different from the other parameters. Centralized performance of the proposed algorithm is given in the Figures 3.6, 3.7, 3.8 and 3.9. Comparing simulation results of the centralized method with the distributed method it is found that the performance of both the

Algorithms	5	10	15
ILMS	-4.6947	-7.9699	-10.8031
IMWN	-0.8861	-1.9555	-3.1579
IMGRN	-25.8303	-24.8967	-24.4575
IMHBR	-33.6345	-33.1457	-32.4596

Table 3.1: Comparison of steady state performance for different block sizes

algorithms are same. Because the centralized cost function is the sum of all the local cost function present in the every sensor node.

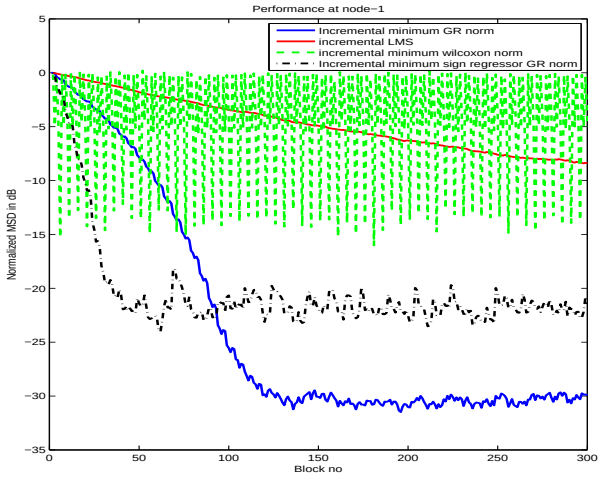

Figure 3.2: Convergence performance at node 1 with 10% outliers in input data with magnitude between $(-3, 3)$, and 10% outliers in desired with magnitude between $(-10, 10)$

The conventional GR suffers from slow convergence particularly when outliers are present in both the input and output data. Hence sign-regressor has been proposed as an extension of GR. This new algorithm offers faster convergence. However its performance slightly less than the GR. In case of sign regressor GR norm sign of the input data is considered in pace of the original data. As a result the variance of the gradient term increases. Hence for the same step size the convergence speed of sign regressor GR norm increases but the optimum MSD performance decreases.

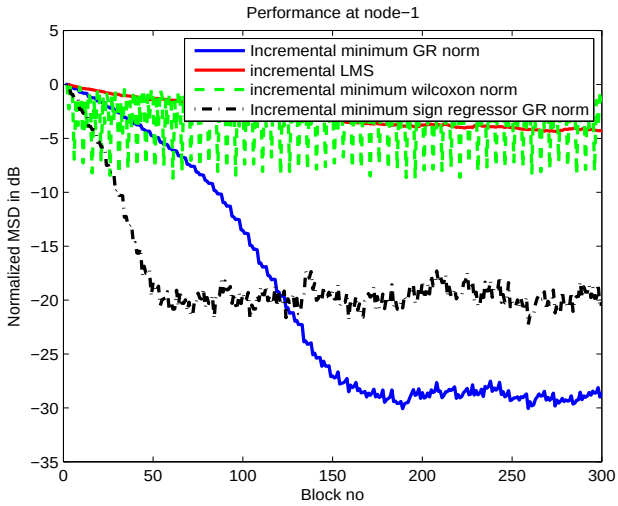

Figure 3.3: Convergence performance at node 1 with 10% outliers in input data with magnitude between $(-3, 3)$, and 40% outliers in desired with magnitude between $(-10, 10)$

3.7 High Breakdown Estimator & Adaptive Generalized R Estimator

3.7.1 High Breakdown (HBR) Estimator

In the previous case the weights in (3.23), are designed using only the information from the input data and hence the performance is dependent on the input data only. Therefore to design an improved algorithm which is superior to the previous method both in convergence and performance, the concept of HBR estimator [16] is incorporated. The HBR estimator is function of both the input and desired data. Here the same proposed median based approach given in Section 3.4.2 is used in finding the parameters of the HBR function. In order to employ the HBR estimator let us define

$$\psi(t) = \begin{cases} 1 & t \geq 1 \\ t & -1 < t < 1 \\ -1 & t \leq 1 \end{cases} \tag{3.32}$$

Further let $m_i = \psi\left[b \Big/ \left(\left(\mathbf{x}^c_{k,(i-1)l+j} - \mathbf{c}\right)^T \mathbf{Q}^{-1} \left(\mathbf{x}^c_{k,(i-1)l+j} - \mathbf{c}\right)\right)\right]$. Here $\mathbf{c}$ and $\mathbf{Q}$ are robust estimates of location and scatter in input space using the same tech-

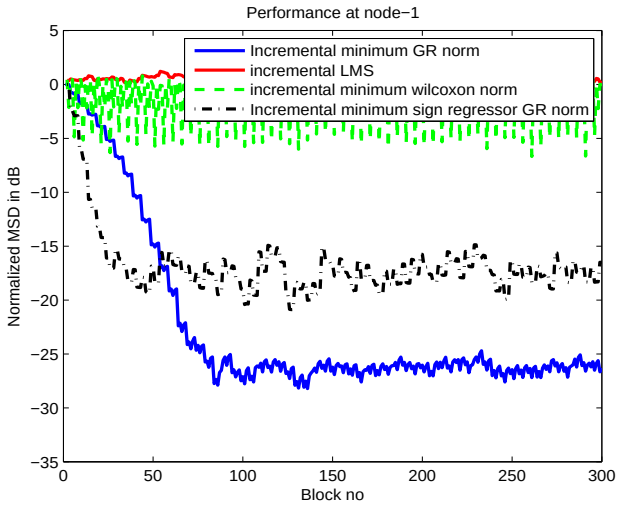

Figure 3.4: Convergence performance at node 1 with 40% outliers in input data with magnitude between $(-3, 3)$, and 10% outliers in desired with magnitude between $(-10, 10)$

nique like as shown in Section 3.4.2, b is the tuning constant. If $\boldsymbol{\psi}_{k-1,i}$ is the estimate of the optimum parameter then the initial residual $e_{k-1,k,i}(j) = y_{k,(i-1)l+j} - (\mathbf{x}^c_{k,(i-1)l+j})^T \boldsymbol{\psi}_{k-1,i}$. Considering the weights as

$$w_{lm} = \psi \left[\left| \frac{t m_l m_m}{\left(e_{k-1,k,i}\,(l)/\sigma_{k,i} \right) \left(e_{k-1,k,i}\,(m)/\sigma_{k,i} \right)} \right| \right] \tag{3.33}$$

where t is the tuning constants and $\sigma_{k,i}$ is the MAD of the error which is given by

$$\text{MAD} = 1.483 \text{med}_s \left| e_{k-1,k,i}(s) - \text{med}_t \left\{ e_{k-1,k,i}(t) \right\} \right| \tag{3.34}$$

Taking $q_m = \left(\mathbf{x}^c_{k,(i-1)l+m} - \mathbf{c} \right)^T \mathbf{Q}^{-1} \left(\mathbf{x}^c_{k,(i-1)l+m} - \mathbf{c} \right)$, we can write

$$m_l = \boldsymbol{\psi} \left(\frac{b}{q_l} \right) = \min \left\{ 1, \frac{b}{q_l} \right\} \tag{3.35}$$

Hence the weights can be estimated as

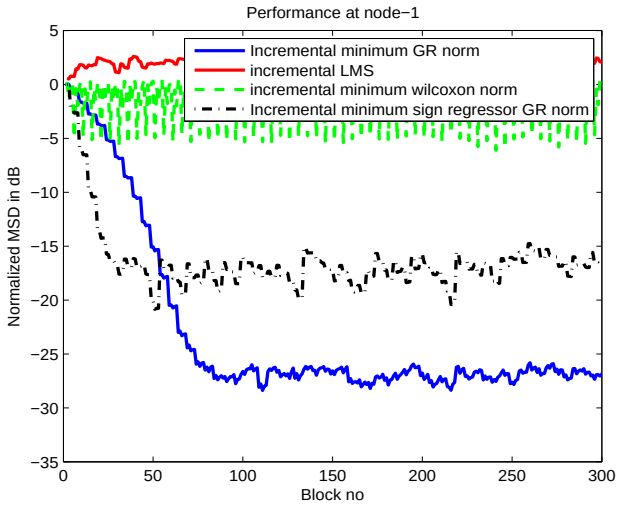

Figure 3.5: Convergence performance at node 1 with 40% outliers in input data with magnitude between $(-3, 3)$, and 40% outliers in desired with magnitude between $(-10, 10)$

$$w_{lm} = \min\left\{1, \frac{c\sigma_{k,i}}{|e_{k-1,k,i}(l)|}\frac{\sigma_{k,i}}{|e_{k-1,k,i}(m)|}\min\left\{1, \frac{b}{q_l}\right\}\min\left\{1, \frac{b}{q_m}\right\}\right\} \tag{3.36}$$

This technique is applied to both incremental minimum GR and incremental minimum sign GR norms. The results obtained are depicted in Figures 3.10-3.13. These figures illustrate the comparisons among all four estimators. In this case the number of data used is 12000 with block size of 40. The simulation results shown are the average over 50 independent experiments. The step size is 0.005 for GR norm case and 0.001 for HBR norm case. As expected it is observed that the convergence and performance of the HBR estimator is improved.

3.7.2 Adaptive Generalize R Estimator

The derivation for the circle in Section 3.4.2 is based on the assumption that 50% of the input data are affected by outliers. However in some cases if the percentage of outliers is less then some information bearing data would be treated as outliers. In that case the convergence speed and performance of the algorithm deteoriate. In order to avoid this, an adaptive threshold can be obtained by which the parameters of the circle can be adaptively chosen. The simuation results for adaptive GR and HBR

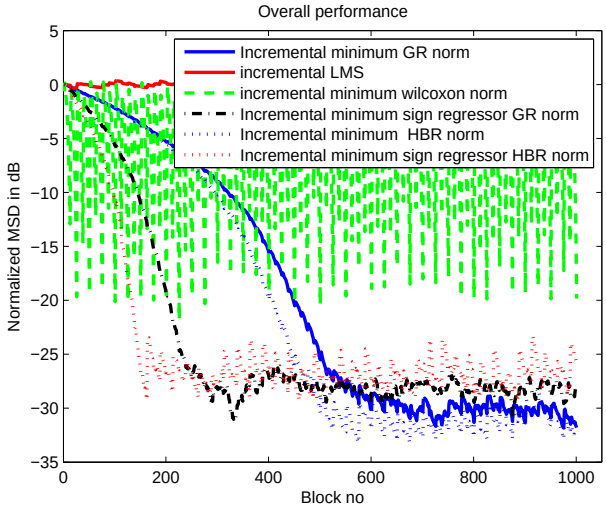

Figure 3.6: Overall convergence performance with 10% outliers in input data with magnitude between $(-3, 3)$, and 10% outliers in desired with magnitude between $(-10, 10)$

Algorithm	Computation	Performance	Convergence
IMGRN	Less	Medium	Less
SR-IMGRN	Less	Less	Medium
IMHBRN	More	Good	Medium
SR-IMHBRN	More	Good	Fast
AdIMGRN	Less	Medium	Fast
SR-AdIMGRN	Less	Less	Fast
AdIMHBRN	More	Good	Fast
SR-AdIMHBRN	More	Medium	Fast

Table 3.2: Overall comparison among all the adaptive strategies to handle outliers both in the input and the desired data

estimators are given in Figure 3.15. In this case 10% input data is contaminated by outliers. The estimators are designed for adaptive case and general case by considering 25% and 50% input data is affected by outliers. From simulation results it is found that the adaptive estimators show faster convergence speed than the general case.

3.8 Conclusions

From the investigation and simulation results it is shown that the proposed techniques are robust against outliers in both desired and input data. Hence these approaches can successfully applied in impulse noise environment. Since the median based approach

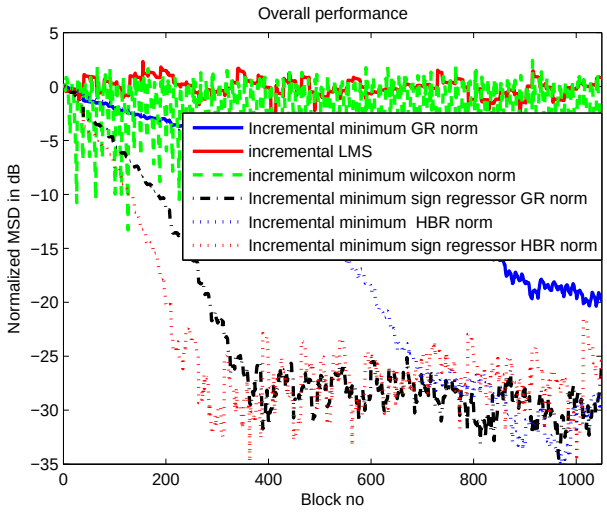

Figure 3.7: Overall convergence performance with 40% outliers in input data with magnitude between $(-3, 3)$, and 10% outliers in desired with magnitude between $(-10, 10)$

requires less computational complexity it can easily be applied at a wireless sensor network node. The HBR estimator can further increase estimator performance as it employs the information of both input and desired data.

In this chapter the technique is applied only using incremental strategy. This work can easily be extended to the problem of robust estimation by using diffusion strategy in which there is no need of a cyclic path connecting each sensor node present in the environment. In the present case the model is a lumped linear parameter model. The same approach can also be extended for nonlinear and distributed parameter based plants or systems. The convergence analysis of the proposed methods can be carried out as a future work using the sophisticated mathematical analysis based upon rank statistics [16].

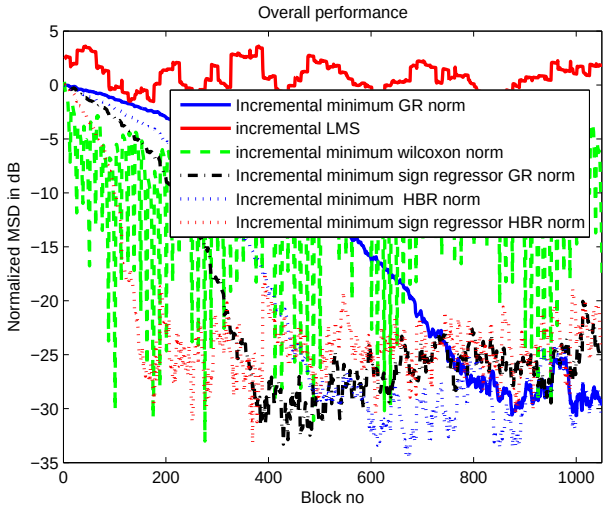

Figure 3.8: Overall convergence performance with 10% outliers in input data with magnitude between $(-3, 3)$, and 40% outliers in desired with magnitude between $(-10, 10)$

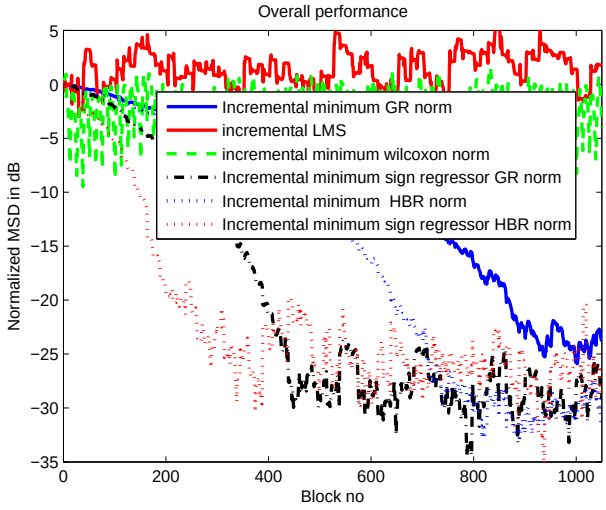

Figure 3.9: Convergence performance with 40% outliers in input data with magnitude between $(-3, 3)$, and 40% outliers in desired with magnitude between $(-10, 10)$

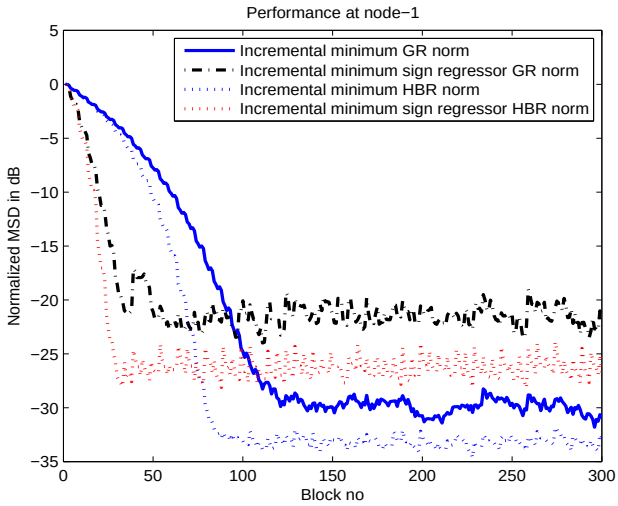

Figure 3.10: Convergence performance at node 1 with 10% outliers in input data with magnitude between $(-3, 3)$, and 10% outliers in desired with magnitude between $(-10, 10)$

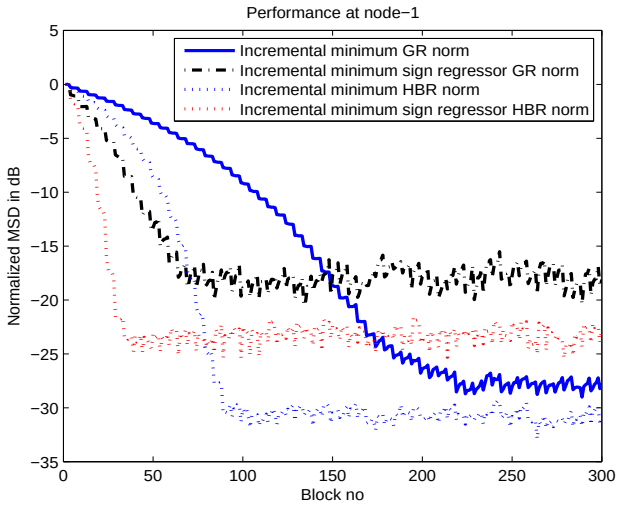

Figure 3.11: Convergence performance at node 1 with 10% outliers in input data with magnitude between $(-3, 3)$, and 50% outliers in desired with magnitude between $(-10, 10)$

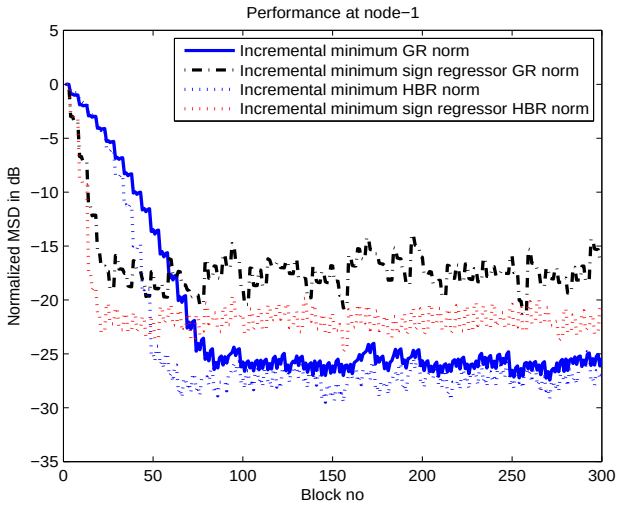

Figure 3.12: Convergence performance at node 1 with 50% outliers in input data with magnitude between $(-3, 3)$, and 10% outliers in desired with magnitude between $(-10, 10)$

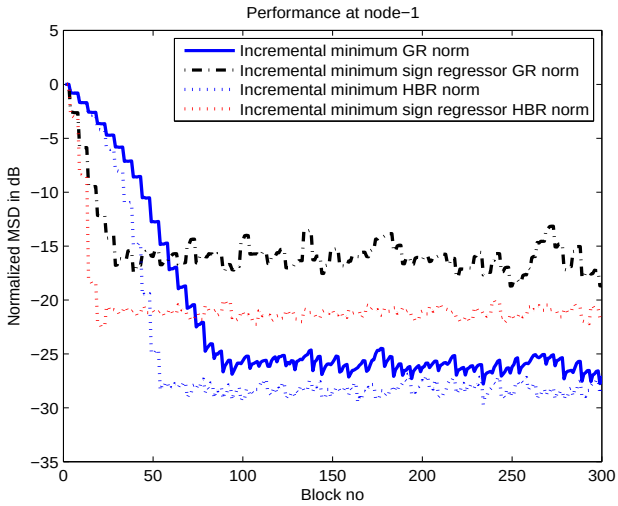

Figure 3.13: Convergence performance at node 1 with 50% outliers in input data with magnitude between $(-3, 3)$, and 50% outliers in desired with magnitude between $(-10, 10)$

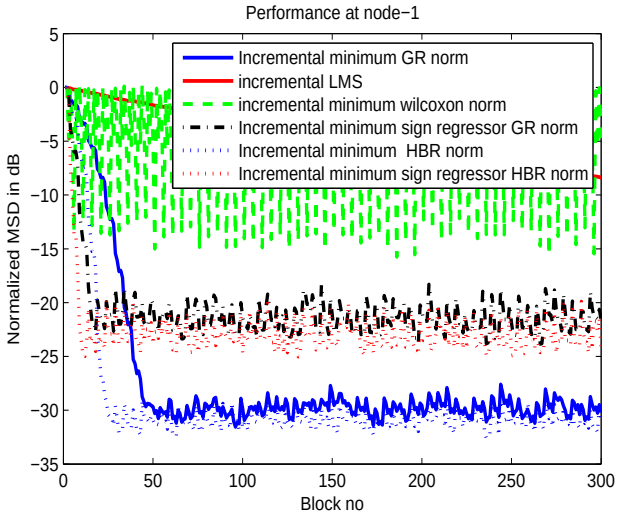

Figure 3.14: Simulation for adaptive estimator(10% outliers with magnitude (-3,3) and 10% outliers in the desired data with magnitude(-10,10))

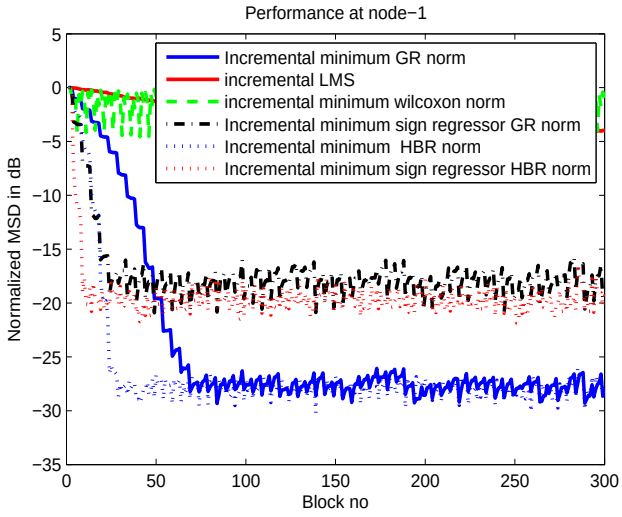

Figure 3.15: Simulation for adaptive estimator(10% outliers with magnitude (-3,3) and 50% outliers in the desired data with magnitude(-10,10))

Chapter 4

QR-Based Incremental Adaptive Strategies to Handle Outliers in the Desired Data as well as in Both Input and Desired Data

The previous chapters deal with the development of some novel algorithms to handle outliers in the desired or in both the input and the desired data. The previous problems shown in Chapters 2 and 3 are reformulated in this chapter as a pseudo least squares(PLS) problem. The solution to the problem is obtained by using QR decomposition. Through this formulation both the convergence speed and performance of the algorithms increase. To demonstrate the potential of this algorithm simulation study is carried out for the distributed estimation of parameters in the presence of weak to strong outliers in the data. The simulation results show that the performance of the new algorithm is robust against outliers and provides better convergence compared to the previous algorithms. Further to achieve low communication overhead, a new scheme is introduced and its performance has been assessed through simulation study. It is observed that the proposed scheme even though exhibits slightly inferior performance but offers the substantial reduction in terms of communication overhead.

4.1 Introduction

Wireless sensor nodes comprising the sensing unit, the processing unit, the memory unit, transmission unit and power supply unit are envisioned to solve many real life problems. In many interesting applications the spatio-temporal data present in the environment is required for the estimation of some parameters of interest. The classical method based on the FC requires more power to fulfill the objectives. For this problem the in-network processing capability of the sensor nodes should be used as described in the previous chapters.

This chapter deals with the development of two novel algorithms to handle outliers in the data using the Wilcoxon norm and the generalize R(GR) norm techniques by formulating these cost functions as PLS cost functions. Then these are used for distributed estimation of the parameters in presence of outliers in the desired as well as in both the input and the desired data. However, the main drawback associated with this type of formulation is that it requires large computations compared to that of the previous methods. Due to the development of low power VLSI and powerful processors this can be easily implemented in WSNs. The techniques are called QR-

based incremental minimum-Wilcoxon-norm (QR-IMWN) and QR-based incremental minimum-GR-norm (QR-IMGRN). For QR-IMGRN the median based approach, which is already employed in Chapter 3, is used to design a computationally efficient algorithm. Further, to decrease the communication overheads a QR decomposition based low communication incremental minimum Wilcoxon norm (QR-LCIMWN) and QR based low communication incremental minimum GR norm (QR-LCIMGRN) are also proposed. From exhaustive simulation studies it is found that the proposed methods provide superior convergence and performance compared to the previous methods.

4.2 Problem Formulation

As given in Chapters 2 and 3, the entire spatio-temporal data of an environment can be related by

$$\mathbf{y}(n) = (\mathbf{X}^u)^T(n)\mathbf{w} + \mathbf{v}(n) \tag{4.1}$$

When there is the presence of outliers in the input data the available input $\mathbf{X}(n)$ is a corrupted version of the original input data $\mathbf{X}^u(n)$. Recall from (3.3) that the objective is to estimate the parameters $\mathbf{w}$ from $\mathbf{y}(n)$ and $\mathbf{X}(n)$. This can be formulated as an optimization problem based on weighted LS formulation as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\|\mathbf{y}(n) - \mathbf{X}^T(n)\mathbf{w}\right\|^2_{\mathbf{\Lambda}(n)} \tag{4.2}$$

where $\mathbf{X}(n) = \begin{cases} \mathbf{X}^u(n) & \text{if input is not affected by outliers} \\ \mathbf{X}^u(n) + \mathbf{D}(n) & \text{if input is affected by outliers} \end{cases}$

In terms of geometry (4.2) can be viewed as to estimate $\mathbf{w}^*$ so that the norm distance between $\mathbf{X}^T(n)\mathbf{w}^*$ and the desired space is minimum.

4.3 Pseudo-Least-Squares(PLS) Formulation

In this section the Wilcoxon norm and the generalize rank norm are formulated as PLS cost functions. Further this cost function is used for the estimation of the unknown parameters $\mathbf{w}$ present in (4.1).

4.3.1 PLS Formulation of the Wilocoxon Norm

The Wilcoxon norm, which is explained in Section 2.2.1 is usually applied over a vector. The Wilcoxon and the sign Wilcoxon norms of the vector $\mathbf{v}$ shown in (2.7) are $\|\mathbf{v}\|_w = \sum_{i=1}^{l} \left(\sqrt{12}\left(\frac{\mathrm{R}(v_i)}{l+1} - 0.5\right) v_i\right)$ and $\|\mathbf{v}\|_{sw} = \sum_{i=1}^{l} \left(\sqrt{12}\mathrm{sign}\left(\frac{\mathrm{R}(v_i)}{l+1} - 0.5\right) v_i\right)$

This functions can be represented as a PLS cost function as shown below

$$\|\mathbf{v}\|_* = \sum_{i=1}^{l} \varepsilon_i^* \left(v_i\right)^2 \tag{4.3}$$

Here the symbol $*$ is used to indicate that this method can be applied both for the Wilcoxon norm and sign Wilcoxon norm. For the Wilcoxon norm the ε_i^* is

$$\varepsilon_i^w = \begin{cases} \frac{a(\mathrm{R}(v_i))}{v_i} & for\ v_i \neq 0 \\ a^{'}\left(\mathrm{R}\left(0\right)\right) & for\ v_i = 0 \end{cases} \tag{4.4}$$

where $a^{'}\left(\mathrm{R}\left(0\right)\right)$ is defined by $\left|\partial a\left(\mathrm{R}\left(v_i\right)\right)/\partial v_i\right|_{v_i=0}$, which is obtained by applying the L-Hospital rule to the weighting factor associated with the nonzero element case and then limiting the value for zero element. Since the rank order of the element is discrete hence derivative needs to be replaced by the finite difference. Then performing finite difference for $a^{'}\mathrm{R}(0)$, we find it is the difference of score values corresponding to two consecutive elements in ordered vector where one is of magnitude zero. Then using (2.6) for $a^{'}(\mathrm{R}(0))$, it yields $a^{'}\left(\mathrm{R}\left(0\right)\right) = \frac{\sqrt{12}/(l+1)}{v^{(i+1)}}$

For the sign Wilcoxon norm the ε_i^* is

$$\varepsilon_i^{sw} = \begin{cases} \frac{\mathrm{sign}(a(\mathrm{R}(v_i)))}{v_i} & for\ v_i \neq 0 \\ a^{'}(\mathrm{sign}((\mathrm{R}\left(0\right)))) & for\ v_i = 0 \end{cases} \tag{4.5}$$

where, $a^{'}(\text{sign}((\text{R}\,(0))))$ is defined by $|\partial \text{sign}(a\,(\text{R}\,(v_i)))/\partial v_i|_{v_i=0}$, that is obtained by applying the L-Hospital rule to the weighting factor associated with the nonzero element case and then limiting the value for zero element. In case of Sign Wilcoxon norm the score value is either 1 or -1. Hence, the finite difference of score values corresponding to two consecutive elements, where one element having zero magnitude, is 2, since $1-(-1)=2$. Therefore, the weighting factor corresponding to zero element is $\text{a}^{'}(\text{sign}(R(0))) = \frac{2}{v^{(i+1)}}$

4.3.2 PLS Formulation of the GR Norm

The generalized-rank(GR) norm is also applied over a vector. The GR norm of the vector (2.6) is given by

$$\|\mathbf{v}\|_{GR} = \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} \left(w_{ij}\text{I}_{(v_i - v_j)} - w_{ij}\text{I}_{(v_j - v_i)}\right) v_i \tag{4.6}$$

.

The GR norm in (4.6) is shown as the summation of the multiplication of the element of the vector with a weighting factor. This weighting factor can be considered as score value associated with the element v_i. The expression of GR norm can be expressed in PLS form as

$$\|\mathbf{v}\|_{GR} = \sum_{i=1}^{L} \epsilon_i v_i^2 \tag{4.7}$$

where

$$\varepsilon_i = \begin{cases} \sum\limits_{j=1, j\neq i}^{l} \frac{w_{ij}\left(\text{I}_{\left(v_i - v_j\right)} - \text{I}_{\left(v_j - v_i\right)}\right)}{v_i} & v_i \neq 0 \\ \chi_i & v_i = 0 \end{cases} \tag{4.8}$$

In order to find χ_i present in (4.8) the L-Hospital rule is applied with the ε_i for $v_i \neq 0$. Then the value is limited to $v_i = 0$. The denominator is a continuous function of v_i, where as the numerator is a discrete function of v_i. Thus the derivative of the

denominator is taken with respect to v_i where as the finite difference of the numerator is taken with respect to v_i. Then the value is limited for $v_i = 0$. The denominator is 1 and the numerator is the difference between two score values associated with two consecutive elements among which one element is of magnitude 0. The χ_i can be found out to be

$$\chi_i = \frac{\sum_{j=1,j\neq i}^{l} w_{ij}\left(I_{(v_i-v_j)} - I_{(v_j-v_i)}\right) - \sum_{j=1,j\neq i+1}^{l} w_{ij}\left(I_{(v_{i+1}-v_j)} - I_{(v_j-v_{i+1})}\right)}{v_{i+1}} \tag{4.9}$$

It is known that the algorithm based on recursive estimation of autocorrelation of the input data and cross correlation between the input and the desired value provides better convergence speed and performance than instant gradient based estimation performance. For an example recursive least squares(RLS) provides better convergence and performance compared to least mean squares (LMS) method. However, due to quantization error the recursive algorithm suffers from the problem of instability. For such case the QR decomposition based method is the best alternative which decreases the dynamic range of the data thus increases the stability of the algorithm [34, 35]. Hence in the present case a QR decomposition based incremental minimum Wilcoxon norm (QR-IMWN) and a QR decomposition based incremental minimum generalized R norm(QR-IMGRN) is proposed. The weighting factor ε_i shown in (4.4) and (4.8) are used to change the forgetting factor $\Lambda(n)$ to achieve the QR-IMWN and QR-IMGRN.

4.4 Block Incremental RLS Strategies using Block Householder Transformation(BHT)

The proposed method requires a block of data for calculation of the cost function therefore block processing is needed for distributed implementation. This section provides a mathematical framework for the proposed method. The formulation proceeds as follows. The entire time period is divided into different block time constituting a number of consecutive time instant. Suppose there are n number of time instants

for each node and l number of time instants are grouped together to make one block time. Thus, there are n/l number of block times corresponding to n number of time instants. The i^{th} block time constitutes the $(i-1)l+1$ to il time instants. Let the input and the measured data over the entire network at j^{th} block be given as

$$\mathbf{X}_{1:N,j} = \begin{bmatrix} \mathbf{X}_{1,j} & \mathbf{X}_{2,j} & \cdots & \mathbf{X}_{N,j} \end{bmatrix} \quad \mathbf{y}_{1:N,j} = \begin{bmatrix} \mathbf{y}_{1,j} & \mathbf{y}_{2,j} & \cdots & \mathbf{y}_{N,j} \end{bmatrix}^T \tag{4.10}$$

Collecting all the input data up to n^{th} block gives

$$\mathbf{X}(n) = \begin{bmatrix} \mathbf{X}_{1:N,1} & \mathbf{X}_{1:N,2} & \cdots & \mathbf{X}_{1:N,n} \end{bmatrix}^T \tag{4.11}$$

.

and similarly the measured data upto n^{th} block is given by

$$\mathbf{y}(n) = \begin{bmatrix} \mathbf{y}^T_{1:N,1} & \mathbf{y}^T_{1:N,2} & \cdots & \mathbf{y}^T_{1:N,n} \end{bmatrix}^T \tag{4.12}$$

the problem is to estimate the parameter vector $\mathbf{w}$ of the model using $\mathbf{X}(n)$ and $\mathbf{y}(n)$. The least squares estimate is

$$\min_{\mathbf{w}} \|\mathbf{y}(n) - \mathbf{X}(n)\mathbf{w}\|^2_{\mathbf{\Lambda}(n)} \tag{4.13}$$

where $\mathbf{\Lambda}(n)$ is the block forgetting factor for the data. Here the norm of a column vector $\mathbf{x}$ is defined in terms of a suitably dimensioned square matrix $\mathbf{A}$, i.e. $\|\mathbf{x}\|^2_{\mathbf{A}} = \mathbf{x}^T\mathbf{A}\mathbf{x}$ and

$$\mathbf{\Lambda}(n) = \text{Diag}\left\{\lambda^{n-1}\mathbf{D}_{NL}, \lambda^{n-2}\mathbf{D}_{NL}, \cdots, \lambda^1\mathbf{D}_{NL}, \mathbf{D}_{NL}\right\} \tag{4.14}$$

where λ is the block forgetting factor for the data from all nodes in one block, $\mathbf{D}_{NL} = 1_{NL}$. Assuming $\mathbf{X}(n)$ as a full column rank matrix, the QR decomposition of the input and desired gives rise to

$$\sqrt{\mathbf{\Lambda}(n)}\begin{bmatrix}\mathbf{X}(n) & \mathbf{y}(n)\end{bmatrix} = \begin{bmatrix}\mathbf{Q}_1^T(n) & \mathbf{Q}_2^T(n)\end{bmatrix}\begin{bmatrix}\mathbf{R}(n) & \mathbf{u}(n)\\ 0 & \mathbf{v}(n)\end{bmatrix} \tag{4.15}$$

where $\mathbf{Q}_1^T(n) \in \Re^{n\times n}$ and $\mathbf{Q}_2^T(n) \in \Re^{((n)k-n)\times n}$ are the orthogonal basis matrices for the range and null space of the input data respectively. The optimum parameter $\mathbf{w}_n$ can be calculated from (4.15) as [40]

$$\mathbf{w}_n = \mathbf{R}^{-1}(n)\,\mathbf{u}(n) \tag{4.16}$$

Since, the estimation 4.16 requires the data available up to n^{th} block, it needs more computation to solve the problem (4.16) in one step. Hence a recursive method must be used to get the optimum parameter $\mathbf{w}_n$ using the estimated parameter $\mathbf{w}_{n-1}$ and the data available during n^{th} block. But the problem is that the data at n^{th} block is present among the sensor nodes dispersed throughout the environment. Hence the spatial recursive method is used using the incremental strategy [5].

4.4.1 Exact Block Incremental RLS Strategies using Block Householder Transformation

The aim is to estimate the parameter $\mathbf{w}_n$ using $\mathbf{w}_{n-1}$ and the data available from each node at the n^{th} block in turn. To this end, define a matrix $\mathbf{X}_{k-1}(n)$ which contains all the available input data in all the nodes up to the $(n-1)^{th}$ block plus the corresponding data at the n^{th} block for nodes 1 to $k-1$

$$\mathbf{X}_{k-1}(n) = \begin{bmatrix}\mathbf{X}^T(n-1) & \mathbf{X}_{1,n} & \mathbf{X}_{2,n} & \cdots & \mathbf{X}_{k-1,n}\end{bmatrix}^T \tag{4.17}$$

where $\mathbf{X}_{k-1}(n) \in \Re^{((n-1)NL+(k-1)L)\times p}$ and p=order of the system vector. In a similar manner the measurement data is represented as

$$\mathbf{y}_{k-1}(n) = \begin{bmatrix} \mathbf{y}^T(n-1) & \mathbf{y}_{1,n}^T & \mathbf{y}_{2,n}^T & \cdots & \mathbf{y}_{k-1,n}^T \end{bmatrix}^T \tag{4.18}$$

where $\mathbf{y}_{k-1}(n) \in \Re^{((n-1)NL+(k-1)L)}$.

Defining the estimated weight vector at $(k-1)^{th}$ node during the incremental strategy at n^{th} iteration, we have

$$\boldsymbol{\psi}_{k-1,n} = \min_{\mathbf{w}} \left\| \mathbf{y}_{k-1}(n) - \mathbf{X}_{k-1}(n)\,\mathbf{w} \right\|^2_{\mathbf{\Lambda}_{k-1}(n)} \tag{4.19}$$

The forgetting factor, i.e. $\mathbf{\Lambda}_{k-1}(n)$, is given by

$$\mathbf{\Lambda}_{k-1}(n) = \mathrm{Diag}\left\{ \lambda^{n-2}\mathbf{D}_{NL}, \cdots, \lambda^{1}\mathbf{D}_{NL}, \mathbf{D}_{(k-1)L} \right\} \tag{4.20}$$

The present problem is to solve

$$\boldsymbol{\psi}_{k-1,n} = \min_{\mathbf{w}} \left\| \mathbf{y}_{k-1}(n) - \mathbf{X}_{k-1}(n)\,\mathbf{w} \right\|^2_{\mathbf{\Lambda}_{k-1}(n)} \tag{4.21}$$

given $\boldsymbol{\psi}_{k-2,n}$ and the data at node k at block n, i.e. $\mathbf{X}_{k,n}$ and $\mathbf{y}_{k,n}$. Since $\mathbf{X}_{k-1}(n)$ is a full column rank matrix and hence the QR decomposition of the input and measured data can be written as

$$\sqrt{\mathbf{\Lambda}_{k-1}(n)} \begin{bmatrix} \mathbf{X}_{k-1}(n) & \mathbf{y}_{k-1}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{1,k-1}^T & \mathbf{Q}_{2,k-1}^T \end{bmatrix} \begin{bmatrix} \mathbf{R}_{k-1}(n) & \mathbf{u}_{k-1}(n) \\ 0 & \mathbf{v}_{k-1}(n) \end{bmatrix} \tag{4.22}$$

$$\text{where} \quad \mathbf{Q}_{1,k-1}(n)\sqrt{\boldsymbol{\Lambda_{k-1}(n)}}\mathbf{X}_{k-1}(n) = \mathbf{R}_{k-1}(n) \tag{4.23}$$

Using the matrix $\mathbf{R}_{k-1}(n)$ and vector $\mathbf{u}_{k-1}(n)$ from (4.22), the optimal weight vector is computed as $\boldsymbol{\psi}_{k-1,n} = \mathbf{R}_{k-1}^{-1}(n)\mathbf{u}_{k-1}(n)$. Incremental estimation of the pa-

rameter vector can be achieved as follows. Define $\overline{\mathbf{Q}}_{1,k}(n)$ as

$$\overline{\mathbf{Q}}_{1,k}(n) = \begin{bmatrix} \mathbf{Q}_{1,k-1}(n) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_L \end{bmatrix} \tag{4.24}$$

Multiplying (4.24) with $\sqrt{\mathbf{\Lambda}_k(n)}\mathbf{X}_k(n)$ gives

$$\overline{\mathbf{Q}}_{1,k}(n)\sqrt{\mathbf{\Lambda}_k(n)}\mathbf{X}_k(n) = \begin{bmatrix} \mathbf{R}_{k-1}(n) \\ \mathbf{x}_{k,n}^T \end{bmatrix} \tag{4.25}$$

and applying the BHT to (4.25) leads to

$$\mathbf{H}_k(n)\,\overline{\mathbf{Q}}_{1,k}(n)\sqrt{\mathbf{\Lambda}_k(n)}\mathbf{X}_k(n) = \begin{bmatrix} \mathbf{R}_k(n) \\ 0 \end{bmatrix} \tag{4.26}$$

The operator $\mathbf{H}_k(n)$ represents the block Householder transformation (BHT). The dimension of the BHT matrix is $(p+L)\times(p+L)$. The expression $\mathbf{H}_k(n)$ is derived in Section C.1.

In preparation for processing the next data block $\mathbf{Q}_{1,k}$ is required as in (4.24). Defining $\mathbf{T} = \begin{bmatrix} \mathbf{I}_p & \mathbf{0}_{p\times l} \end{bmatrix}$ and multiplying it by (4.26) gives

$$\mathbf{T}\mathbf{H}_k(n)\overline{\mathbf{Q}}(n)\sqrt{\mathbf{\Lambda}}_k(n)\mathbf{X}_k(n) = \mathbf{R}_k(n) \tag{4.27}$$

and $\mathbf{Q}_{1,k}(n) = \mathbf{T}\mathbf{H}_k(n)\overline{\mathbf{Q}}_{1,k}(n)$. Performing similar operation as in (4.25) with the desired data, we have

$$\mathbf{T}\mathbf{H}_k(n)\,\overline{\mathbf{Q}}_{1,k}(n)\sqrt{\mathbf{\Lambda}_k(n)}\mathbf{y}_k(n) = \mathbf{u}_k(n) \tag{4.28}$$

A combination of (4.27) and (4.28) gives

$$\mathbf{TH}_k(n)\begin{bmatrix} \mathbf{R}_{k-1}(n) & \mathbf{u}_{k-1}(n) \\ \mathbf{X}_{k,n}^T & \mathbf{y}_{k,n} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k(n) & \mathbf{u}_k(n) \end{bmatrix} \tag{4.29}$$

We can get $\boldsymbol{\psi}_{k,n}$ using $\mathbf{R}_k(n)$ and $\mathbf{u}_k(n)$. In the distributed scenario the information to be transmitted from node k to $k+1$ are $\mathbf{R}_k(n)$ and $\mathbf{u}_k(n)$.

Thus the distributed algorithm is outlined as

$$\begin{cases} \boldsymbol{\psi}_{0,n} \leftarrow \mathbf{w}_{n-1} \\ \mathbf{R}_0(n) \leftarrow \sqrt{\lambda}\mathbf{R}(n-1) \\ \mathbf{u}_0(n) \leftarrow \sqrt{\lambda}\mathbf{u}(n-1) \\ \text{For } \ k = 1:N \\ \begin{bmatrix} \mathbf{R}_k(n) & \mathbf{u}_k(n) \end{bmatrix} \leftarrow \mathbf{TH}_k(n)\begin{bmatrix} \mathbf{R}_{k-1}(n) & \mathbf{u}_{k-1}(n) \\ \mathbf{X}_{k,n}^T & \mathbf{y}_{k,n} \end{bmatrix} \\ \boldsymbol{\psi}_{k,n} \leftarrow \mathbf{R}_k^{-1}(n)\,\mathbf{u}_k(n) \\ \text{end} \\ \mathbf{w}_n \leftarrow \boldsymbol{\psi}_{N,n} \end{cases}$$

4.4.2 Derivation of QR-IMWN and QR-IMGRN Algorithms

This subsection deals with the development of QR-IMWN and QR-IMGRN algorithms using the PLS formulation of the Wilcoxon and GR norms. This is obtained as follows. The objective is to estimate

$$\mathbf{w} = \arg\min_{\mathbf{w}} \left\|\mathbf{y}(n) - \mathbf{X}^T(n)\,\mathbf{w}\right\|_{\mathrm{N}} \tag{4.30}$$

where N may be the GR norm or the Wilcoxon norm depending upon the presence of outliers in the input data or not. This can be represented as

$$\mathrm{N} = \begin{cases} \text{The Wilcoxon Norm} & \text{for } \mathbf{D} = \mathbf{0} \\ \text{The GR Norm} & \text{for } \mathbf{D} \neq \mathbf{0} \end{cases} \tag{4.31}$$

By changing the term $\left\|\mathbf{y}(n) - \mathbf{X}^T(n)\,\mathbf{w}\right\|_{\mathrm{N}}$ to a form as in (4.19), (4.30) can be

shown as $\mathbf{w} = \arg\min_{\mathbf{w}} \left\|\mathbf{y}\left(n\right) - \mathbf{X}^{T}\left(n\right)\mathbf{w}\right\|^{2}_{\mathbf{\Lambda}^{\mathrm{N}}(n)}$ The weighting factor $\mathbf{\Lambda}^{\mathrm{N}}\left(n\right)$ depends upon the total spatial data up to n^{th} time. Thus

$$\left\|\mathbf{y}\left(n\right) - \mathbf{X}^{T}\left(n\right)\mathbf{w}\right\|^{2}_{\mathbf{\Lambda}^{\mathrm{N}}(n)} \neq \sum_{i=1}^{N}\left\|\mathbf{y}^{i}\left(n\right) - \left(\mathbf{X}^{i}\left(n\right)\right)^{T}\mathbf{w}\right\|^{2}_{\mathbf{\Lambda}^{(i)\mathrm{N}}(n)}. \tag{4.32}$$

where $\left\|\mathbf{y}\left(n\right) - \mathbf{X}^{T}\left(n\right)\mathbf{w}\right\|^{2}_{\mathbf{\Lambda}^{\mathrm{N}}(n)}$ is the global cost function and $\left\|\mathbf{y}^{i}\left(n\right) - \left(\mathbf{X}^{i}\left(n\right)\right)^{T}\mathbf{w}\right\|^{2}_{\mathbf{\Lambda}^{(i)\mathrm{N}}(n)}$ is the local cost function at node i. Hence it is not amenable for distributed implementation in WSNs [5].

In order to design a cost function which will be robust against outliers as well as can be implemented using the incremental strategy define the local cost function as

$$\mathbf{w} = \arg\min_{\mathbf{w}} \left\|\mathbf{y}^{i}\left(n\right) - \left(\mathbf{X}^{i}\right)^{T}\left(n\right)\mathbf{w}\right\|^{2}_{\mathbf{\Lambda}^{(i)\mathrm{N}}(n)} \tag{4.33}$$

where, $\mathbf{y}^{i}\left(n\right)$ and $\mathbf{X}^{i}\left(n\right)$ are the entire desired and input data at node i. Based on the local cost function (4.33), the global cost function is defined as

$$\min_{\mathbf{w}} \sum_{i=1}^{N}\left\|\mathbf{y}^{i}\left(n\right) - \mathbf{X}^{i}\left(n\right)\right\|^{2}_{\mathbf{\Lambda}^{(i)\mathrm{N}}(n)} \tag{4.34}$$

In order to estimate this cost function the measurement process need to be completed. In order to circumvent this and to estimate the process block by block manner the local cost function is redefined as

$$\min_{\mathbf{w}} \sum_{i=1}^{n/L}\left\|\mathbf{y}_{k,(iL+1):(i+1)L} - \mathbf{X}^{T}_{k,(iL+1):(i+1)L}\mathbf{w}\right\|^{2}_{\Lambda^{(i)b-\mathrm{N}}(n)} \tag{4.35}$$

Similar to (4.34), define the global cost function for the local cost function given in (4.35) as

$$\min_{\mathbf{w}} \sum_{k=1}^{N}\sum_{i=1}^{n/L}\left\|\mathbf{y}_{k,(iL+1):(i+1)L} - \mathbf{X}^{T}_{k,(iL+1):(i+1)L}\mathbf{w}\right\|^{2}_{\mathbf{\Lambda}^{(i)b-\mathrm{N}}(n)} \tag{4.36}$$

Equation (4.36) can be simplified as

$$\overline{\mathbf{w}}_{L,i}^{*} = \arg\min_{\mathbf{w}} \sum_{j=1}^{n/L} \left(\lambda^{(n/L-j)} \sum_{j=1}^{N} \left\| \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}} \mathbf{y}_{k,j} - \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}} \mathbf{X}_{k,j}^{T} \mathbf{w} \right\|^{2} \right) \tag{4.37}$$

-

which can be expressed as

$$\overline{\mathbf{w}}_{L,i}^{*} = \arg\min_{\mathbf{w}} \sum_{j=1}^{n/L} \left(\sum_{j=1}^{N} \left\| \sqrt{\lambda^{(n/L-j)} \boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}} \mathbf{y}_{k,j} - \sqrt{\lambda^{(n/L-j)} \boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}} \mathbf{X}_{k,j}^{T} \mathbf{w} \right\|^{2} \right) \tag{4.38}$$

Further, it can be modified to a similar form like (4.13) with a different weighting factor, which is given by

$$\overline{\mathbf{w}}_{L,i}^{*} = \arg\min_{\mathbf{w}} \left\| \mathbf{y}\left(n\right) - \mathbf{X}^{T}(n)\mathbf{w} \right\|_{\boldsymbol{\Lambda}^{b,\mathrm{N}}(n)}^{2} \tag{4.39}$$

4.4.3 Calculation of Forgetting Factor for the PLS Method

This subsection deals with the calculation of the weighting factor $\boldsymbol{\Lambda}^{b,\mathrm{N}}\left(n\right)$ of the cost function (4.39). For this consider the n^{th} block at the k^{th} node, the Wilcoxon or generalize R norm of the error vector is

$$\left\| \mathbf{e}_{k-1,k,n} \right\|_{\mathrm{N}} = \sum_{i=1}^{L} a(e_{k-1,k,n})\left(i\right) e_{k-1,k,n}\left(i\right) \tag{4.40}$$

where $e_{k-1,k,n}(i) = y_{k,(n-1)L+i} - \mathbf{x}_{k,(n-1)L+i}^{T} \boldsymbol{\psi}_{k-1,n}$. Following (4.3) and (4.7),(4.40) can be rewritten as a PSL problem as

$$\left\| \mathbf{e}_{k-1,k,n} \right\|_{\mathrm{N}} = \sum_{i}^{L} \varepsilon_{k-1,k,n}^{\mathrm{N}}\left(i\right) e_{k-1,k,n}^{2}\left(i\right) \tag{4.41}$$

.

Here $\epsilon_{k-1,k,n}^{\mathrm{N}}(i)$ is interpreted as the weighting factor for the i^{th} error of n^{th} block. This weighting factor depends upon the norm being used. Multiplying the input

and desired data by this weighting factor decreases the effect of outliers. The block weighting factor for the block of errors at the n^{th} iteration in the k^{th} node is

$$\boldsymbol{\varepsilon}^{\mathrm{N}}_{k-1,k,n} = \mathrm{Diag}\left[\begin{array}{cccc} \varepsilon^{\mathrm{N}}_{k-1,k,n}(1) & \varepsilon^{\mathrm{N}}_{k-1,k,n}(2) & \cdots & \varepsilon^{\mathrm{N}}_{k-1,k,n}\ (l) \end{array}\right] \tag{4.42}$$

The weighting factor matrix for the entire spatial data set available up to k^{th} node at n^{th} iteration is

$$\Lambda^{\mathrm{N}}_{k-1}(n) = \mathrm{Diag}\left[\begin{array}{cccc} \lambda^{n-1}\boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,N,n} & \lambda^{n-2}\boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,N,n-1} & \cdots & \boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,k-1,1} \end{array}\right] \tag{4.43}$$

where

$$\boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,N,n} = \mathrm{Diag}\left\{\boldsymbol{\varepsilon}^{\mathrm{N}}_{N,1,n}, \boldsymbol{\varepsilon}^{\mathrm{N}}_{1,2,n}, \cdots, \boldsymbol{\varepsilon}^{\mathrm{N}}_{N-1,N,n}\right\} \tag{4.44}$$

4.4.4 Formulation of QR-IMWN or QR-IMGRN Problem

The proposed method is obtained by multiplying the weighting factor (4.43)with the forgetting factor (4.14). The superscript N is used to signify that this mathematical technique can be used for both weighting factor made from Wilcoxon and generalize rank scores. Hence the block modified forgetting factor with superscript N is represented as

$$\boldsymbol{\Lambda}^{\mathrm{N}}_{k-1}(n) = \mathrm{Diag}\left[\begin{array}{cccc} \lambda^{n-1}\boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,N,n} & \lambda^{n-2}\boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,N,n-1} & \cdots & \boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,k-1,1} \end{array}\right] \tag{4.45}$$

Let $\boldsymbol{\Lambda}^{\mathrm{N}}_{k}(n)$ represents the forgetting factors for the proposed method. It is given by

$$\boldsymbol{\Lambda}^{\mathrm{N}}_{k}(n) = \mathrm{Diag}\left[\begin{array}{cccc} \boldsymbol{\lambda}^{n-1}\boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,N,1} & \cdots & \boldsymbol{\lambda}^{1}\boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,N,n-1} & \boldsymbol{\varepsilon}^{\mathrm{N}}_{N,\cdots,k,n} \end{array}\right] \tag{4.46}$$

The forgetting factor $\boldsymbol{\Lambda}^{\mathrm{N}}_{k}(n)$ is used instead of $\boldsymbol{\Lambda}_{k}(n)$, to design the proposed method. The weight vector is given by

$$\boldsymbol{\psi}^{\mathrm{N}}_{k-1,n} = \arg\min_{\mathbf{w}} \|\mathbf{y}_{k-1}(n) - \mathbf{X}_{k-1}(n)\,\mathbf{w}\|^2_{\boldsymbol{\Lambda}^{\mathrm{N}}_{k-1,n}} = \arg\min_{\mathbf{w}} \|\mathbf{e}_{k-1}(n)\|^2_{\boldsymbol{\Lambda}^{\mathrm{N}}_{k-1,n}} \quad (4.47)$$

Thus the proposed technique is

$$\sqrt{\boldsymbol{\Lambda}^{\mathrm{N}}_{k-1}(n)} \begin{bmatrix} \mathbf{X}_{k-1}(n) & \mathbf{y}_{k-1}(n) \end{bmatrix} = \begin{bmatrix} \left(\mathbf{Q}^{\mathrm{N}}_{1,k-1}(n)\right)^T & \left(\mathbf{Q}^{\mathrm{N}}_{2,k-1}(n)\right)^T \end{bmatrix} \begin{bmatrix} \mathbf{R}^{\mathrm{N}}_{k-1}(n) & \mathbf{u}^{\mathrm{N}}_{k-1}(n) \\ 0 & \mathbf{v}^{\mathrm{N}}_{k-1}(n) \end{bmatrix} \quad (4.48)$$

The weight vector is

$$\boldsymbol{\psi}^{\mathrm{N}}_{k-1,n} = \left(\mathbf{R}^{\mathrm{N}}_{k-1}(n)\right)^{-1} \mathbf{u}^{\mathrm{N}}_{k-1}(n) \quad (4.49)$$

where $\mathbf{Q}^{\mathrm{N}}_{1,k-1}(n) \in \Re^{p\times(n-1)NL+(k-1)L}$ and $\mathbf{Q}^{\mathrm{N}}_{2,k-1}(n) \in \Re^{((n-1)NL+(k-1)L-p)\times(n-1)NL+(k-1)L}$ constitute an orthogonal transformation. Here $\mathbf{R}^{\mathrm{N}}_{k-1}(n) \in \Re^{p\times p}$ is an upper triangular matrix, $\mathbf{u}^{\mathrm{N}}_{k-1}(n) \in \Re^{p\times 1}$ and $\mathbf{v}^{\mathrm{N}}_{k-1}(n) \in \Re^{(n-1)NL+(k-1)L\times 1}$. Then to make the algorithm incremental, operations similar to (4.24)-(4.28) are used with $\mathbf{R}^{\mathrm{N}}_{k-1}(n)$, $\mathbf{u}^{\mathrm{N}}_{k-1}(n)$ and further using BHT matrix $\mathbf{H}^{\mathrm{N}}_{k}(n)$ to get $\mathbf{R}^{\mathrm{N}}_{k}(n)$ and $\mathbf{u}^{\mathrm{N}}_{k}(n)$. A procedure for finding $\mathbf{H}^{\mathrm{N}}_{k}(n)$ is derived in Section C.2. To get the optimum parameter iteratively the following procedure is carried out

$$\mathbf{TH}^{\mathrm{N}}_{k}(n) \begin{bmatrix} \mathbf{R}^{\mathrm{N}}_{k-1}(n) & \mathbf{u}^{\mathrm{N}}_{k-1}(n) \\ \sqrt{\boldsymbol{\varepsilon}^{\mathrm{N}}_{k-1,k,n}}\mathbf{X}^T_{k,n} & \sqrt{\boldsymbol{\varepsilon}^{\mathrm{N}}_{k-1,k,n}}\mathbf{y}_{k,n} \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{\mathrm{N}}_{k}(n) & \mathbf{u}^{\mathrm{N}}_{k}(n) \end{bmatrix} \quad (4.50)$$

The updated weight vector is given as

$$\boldsymbol{\psi}^{\mathrm{N}}_{k,n} = \left(\mathbf{R}^{\mathrm{N}}_{k}(n)\right)^{-1} \mathbf{u}^{\mathrm{N}}_{k}(n) \quad (4.51)$$

4.4.5 Communication Complexity

In the proposed method, one node sends $\mathbf{R}^{\mathrm{N}}_{k}(n)$ and $\mathbf{u}^{\mathrm{N}}_{k}(n)$ to the next node. $\mathbf{R}^{\mathrm{N}}_{k}(n)$ is an upper triangular matrix of order $\Re^{p\times p}$. Therefore the number of data values

transfered to the next node is $(0.5)p(p+1)$. For transmitting $\mathbf{u}_k^{\mathrm{N}}(n)$, p data values are required. The amount of data to communicate after every block of L time samples is $((0.5)p(p+1)+p)/L$. In case of incremental RLS strategy the ensemble average which is order of $\Re^{p\times p}$ and estimated parameter which order of p are to be sent to the next node. Hence the communication complexity is p^2+p. The communication complexity of the proposed technique is rewritten as $(p^2+p)/(2L)+(p/L)$. Comparing the communication complexity of the proposed technique with conventional technique, it is observed that the proposed technique requires much less communication between nodes.

4.4.6 Stepwise Description of the Algorithm

Operation at k^{th} node in n^{th} iteration the steps are

(i) Transfer the compute data matrix $\mathbf{R}_{k-1}^{\mathrm{N}}(n)$ and $\mathbf{u}_{k-1}^{\mathrm{N}}(n)$ from $(k-1)^{th}$ node to k^{th} node

(ii) Process the data at k^{th} node

step-1: Compute $\boldsymbol{\psi}_{k-1,n}^{\mathrm{N}} = \left(\mathbf{R}_{k-1}^{\mathrm{N}}(n)\right)^{-1}\mathbf{u}_{k-1}^{\mathrm{N}}(n)$

step-2: If the norm is GR norm then calculate of w_{ij} from the input.

step-3: Calculation of error $e_{k-1,k,n}(i) = y_{k,(n-1)L+i} - \mathbf{x}_{k,(n-1)L+i}^T\boldsymbol{\psi}_{k-1,n}^{\mathrm{N}}$

step-4: Calculate the weighting factor.

$\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}} = \mathrm{Diag}\left\{\varepsilon_{k-1,k,n}^{\mathrm{N}}(1), \varepsilon_{k-1,k,n}^{\mathrm{N}}(2), \cdots, \varepsilon_{k-1,k,n}^{\mathrm{N}}(L)\right\}$

step-5:

compute

$$\overline{\mathbf{R}}_k^{\mathrm{N}}(n) = \begin{bmatrix} \mathbf{R}_{k-1}^{\mathrm{N}}(n) \\ \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{X}_{k,n}^T \end{bmatrix}$$

$$\overline{\mathbf{u}}_k^{\mathrm{N}}(n) = \begin{bmatrix} \mathbf{u}_{k-1}^{\mathrm{N}}(n) \\ \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{y}_{k,n} \end{bmatrix}$$

step:6

Calculation of BHT matrix $\mathbf{H}_k^{\mathrm{N}}(n)$.

step:7

Multiply BHT matrix $\mathbf{H}_k^{\mathrm{N}}(n)$ to (4.31) to make the column zero and multiply $\mathbf{T}$ to erase the zero rows

$\mathbf{TH}_k^{\mathrm{N}}(n)\overline{\mathbf{R}}_k^{\mathrm{N}}(n) = \mathbf{R}_k^{\mathrm{N}}(n)$

Similarly

$\mathbf{TH}_k^{\mathrm{N}}(n)\overline{\mathbf{u}}_k^{\mathrm{N}}(n) = \mathbf{u}_k^{\mathrm{N}}(n)$

step-8: Obtain optimum weight vector

$\boldsymbol{\psi}_{k,n}^{\mathrm{N}} = \left(\mathbf{R}_k^{\mathrm{N}}(n)\right)^{-1}\mathbf{u}_k^{\mathrm{N}}(n)$

(iii) Transmit data $\mathbf{R}_k^{\mathrm{N}}(n)$ and $\mathbf{u}_k^{\mathrm{N}}(n)$ to the $(k+1)^{th}$ node

4.4.7 Simulation Results and Discussion

For QR-IMWN

For the simulation studies five nodes are considered. At each node the model parameter considered is $\left[\begin{array}{ccccc} 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} \end{array}\right]$. Uniformly distributed random data between $(-0.5, 0.5)$ is used as input to the model to give the output data. White Gaussian noise is added to the output data having a signal to noise ratio(SNR) of 30dB at each of the five nodes. Further, outliers ranging between (-10,10) are added to the desired data at random positions to simulate the effect of data corruption in a real world environment. The percentage of outliers used in the study ranged between 10% and 50% of the desired data. However, in this chapter results for 10% and 50% outliers are shown. The block size chosen is $L = 40$. In case of overall performance the mean square deviation(MSD) of the updated parameter is taken from first node to N^{th} node starting from first block to last block. A total of 8000 data samples at every sensor node were used. The forgetting factor for QR-IMWN, QR-IMSWN and QR based block incremental RLS strategies are 0.71, 0.84 and 1 respectively. These forgetting factors are chosen empirically to get the best performance for each algorithm. The simulation results are obtained by taking the average of 50 independent experiments.

The overall performance is shown in Figures 4.1-4.2. Here the performance of QR based block incremental RLS, QR-IMWN, QR-IMSWN and IMWN are plotted for comparison purpose. From these figures it is observed that the QR-IMWN, QR-IMSWN and previous proposed method are robust against outliers in the desired signal but QR-IMWN outperforms the previously proposed method in terms of convergence speed and the steady state performance. The QR based block incremental

Algorithms	5	10	15
IMWN	-41.4296	-42.2643	-42.7238
QR-IMWN	-50.4048	-51.2527	-51.6486
QR-IMSWN	-49.7641	-53.1692	-55.9304

Table 4.1: Comparison of steady state performances for different block sizes

RLS provides poor performance and with increase in outliers the performance further degrades. The convergence of QR-IMSWN is good in the first few iterations then it slowly tends to the steady state. Since this algorithm uses the sign of the score value,it is more computationally efficient than the other algorithms. It is clear from the results that the proposed method is good both in convergence and performance when compared with previous methods. Here we have simulated the algorithm with different forgetting factors and found out satisfactory performance in the range 0.50.9. This range depends on the magnitude and percentage of the outliers. Sophisticated mathematical analysis based on the asymptotic linearity of rank test [11] can be used to find out the range of the forgetting factor. If the forgetting factor is nearer to 1 then the effect of outliers remain in the estimation process until the last iteration and hence its performance degrades. If this factor is low then the effect of outliers as well as the information about the parameter decreases with iteration,hence the performance also degrades. Performance comparison between different sensor nodes are given in the Table 4.1. For this steady state performance comparison the parameter of the environment is $\begin{bmatrix} 1/\sqrt{5} & 1/\sqrt{3} & 1/\sqrt{5} & 1/\sqrt{3} & 1/\sqrt{5} \end{bmatrix}$, where some parameters are different from the other. Centralized performance of the proposed algorithm is given in the Figures 4.7 and 4.8. Comparing simulation results of the centralized method with the distributed method it is found that the performance of both the algorithms is same. Because the centralized cost function is the sum of all the local cost function present in the every sensor node.

For QR-IMGRN

In case of this the input data, model parameter, number of sensor nodes, noise and desired outliers strength are same as in the Section 4.4.7. A total of 4000 data samples at every sensor node were used. Further the input data is also added with outliers

continuously with a random magnitude between $(-3, 3)$. The block size is taken to be 40. Simulations have been carried out for separately with $10\%, 20\%, 30\%, 40\%, 50\%$ outliers in both input and desired data. However only the simulation results for 10% and 50% are shown in this chapter. The results presented are the average result over 50 independent experiments. Here the block forgetting factor is chosen to be 0.7. The simulation results are given in Figures $4.3 - 4.6$. In the simulation results the comparisons are done among incremental RLS, QR-IMWN, IMGRN and QR-IMGRN. From simulation results it is found that the IMGRN and QR-IMGRN are robust against outliers both in desired and input space. It can also be found from simulation results that the convergence speed of QR-IMGRN is faster than the IMGRN. From figures it can be observed that with fixed input outliers as the percentage of outliers in desired increases the convergence speed of QR-IMGRN decreases but QR-IMWN is performing well because of the masking effect of the desired outliers with input outliers. Therefore some input outliers are canceled by the desired outliers. As the outliers in the input data increases the QR-IMGRN remains robust against outliers in the input and desired data whereas QR-IMWN and IRLS fail to estimate the optimum parameters. Thus from performance and convergence point of view it is concluded that the QR-IMGRN is a superior method compared to IMGRN.

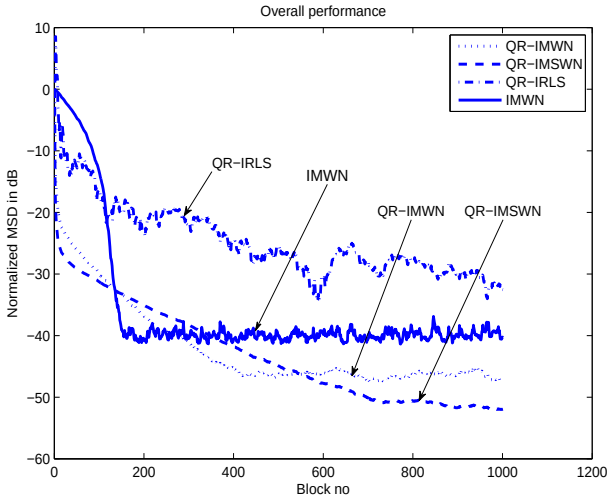

Figure 4.1: 10% outliers in output with random magnitude between $(-10, 10)$

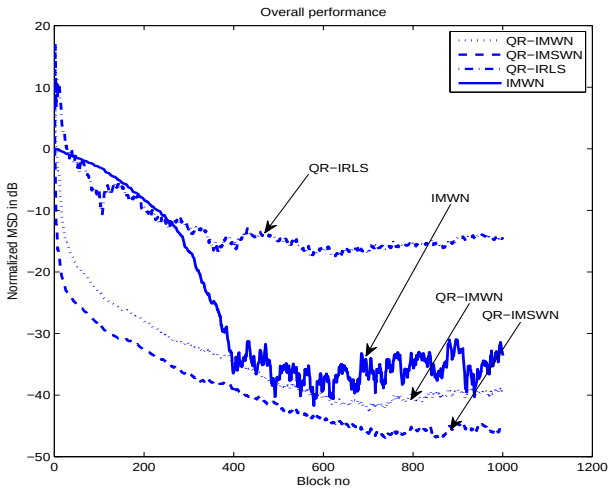

Figure 4.2: 50% outliers in output with random magnitude between $(-10, 10)$

4.5 QR based Low Communication Incremental Strategy for WSNs

4.5.1 Problem Formulation

This subsection deals with a QR based low communication incremental strategy method. To reduce the communication overheads and to exploit the PLS formulation proposed in Section 4.4.2, the weight vector itself is communicated rather than the matrices $\mathbf{R}_k^{\mathrm{N}}$ and $\mathbf{u}_k^{\mathrm{N}}$. At each node the weight vector from the neighboring node is used to calculate the error sequence and hence the forgetting factor. Local data is used to update the weights using the resultant forgetting factor. To achieve this objective the data matrix is changed to

$$\mathbf{X}_k(n-1) = \begin{bmatrix} \mathbf{X}_{k,1}^T \\ \mathbf{X}_{k,2}^T \\ \vdots \\ \mathbf{X}_{k,n-1}^T \end{bmatrix} = \begin{bmatrix} \mathbf{X}_k(n-2) \\ \mathbf{X}_{k,n-1}^T \end{bmatrix} \tag{4.52}$$

where $\mathbf{X}_k\,(n-1) \in \Re^{(n-1)L \times p}$. Similarly the desired data vector is represented as

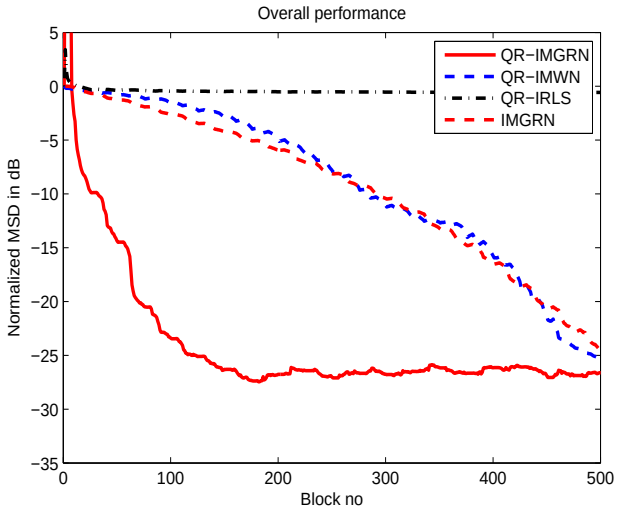

Figure 4.3: 10% outlier in input with random magnitude between $(-3, 3)$ and 10% outlier in output with random magnitude between $(-10, 10)$

$$\mathbf{y}_k(n-1) = \begin{bmatrix} \mathbf{y}_{k,1} \\ \mathbf{y}_{k,2} \\ \vdots \\ \mathbf{y}_{k,n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_k(n-2) \\ \mathbf{y}_{k,n-1} \end{bmatrix} \tag{4.53}$$

where $\mathbf{y}_k\left(n-1\right) \in \Re^{(n-1)L}$

Following the operation in (4.48) the representations of data in (4.52) and (4.53) for QR based low communication strategy can be expressed in the form

$$\begin{aligned} &\Lambda_k^{\mathrm{N}}\left(n-1\right) \begin{bmatrix} \mathbf{X}_k\left(n-1\right) & \mathbf{y}_k\left(n-1\right) \end{bmatrix} = \\ &\begin{bmatrix} \left(\mathbf{Q}_{1,k}^{\mathrm{N}}\left(n-1\right)\right)^T & \left(\mathbf{Q}_{2,k}^{\mathrm{N}}\left(n-1\right)\right)^T \end{bmatrix} \times \begin{bmatrix} \mathbf{R}_k^{\mathrm{N}}\left(n-1\right) & \mathbf{u}_k^{\mathrm{N}}\left(n-1\right) \\ 0 & \mathbf{v}_k^{\mathrm{N}}\left(n-1\right) \end{bmatrix} \end{aligned} \tag{4.54}$$

The weight vector is given as $\boldsymbol{\psi}_{k,n-1}^{\mathrm{N}} = \left(\mathbf{R}_k^{\mathrm{N}}\left(n-1\right)\right)^{-1} \mathbf{u}_k^{\mathrm{N}}\left(n-1\right)$.

Operations similar to (4.24-4.27) are performed with $\mathbf{R}_k^{\mathrm{N}}(n)$ and $\mathbf{u}_k^{\mathrm{N}}(n)$ to get the forgetting factor $\boldsymbol{\varepsilon}_{N,\cdots,N,1}^{\mathrm{N}}$. Then the following matrix is obtained performing similar operation like (4.28) with $\mathbf{R}_k^{\mathrm{N}}(n)$, $\mathbf{u}_k^{\mathrm{N}}(n)$, $\mathbf{X}_{k,n}$ and $\mathbf{y}_{k,n}$, which is given by

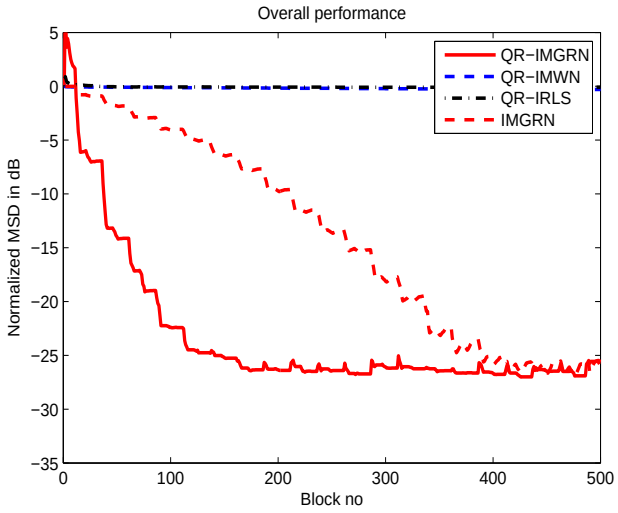

Figure 4.4: 50% outlier in input with random magnitude between $(-3, 3)$ and 10% outlier in output with random magnitude between $(-10, 10)$

$$\left[\begin{array}{cc} \overline{\mathbf{R}}_k^{\mathrm{N}}(n) & \overline{\mathbf{u}}_k^{\mathrm{N}}(n) \end{array}\right] = \left[\begin{array}{cc} \sqrt{\lambda}\mathbf{R}_k^{\mathrm{N(n)}} & \sqrt{\lambda}\mathbf{u}_k^{\mathrm{N(n)}} \\ \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{X}_{k,n}^T & \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{y}_{k,n} \end{array}\right] \tag{4.55}$$

Then simultaneously multiplying $\mathbf{H}_k^{\mathrm{N}}(n)$, which is the Householder transformation matrix for $\overline{\mathbf{R}}_k^{\mathrm{N}}(n)$, and and $\mathbf{T}$ to obtain $\mathbf{R}_k^{\mathrm{N}}(n)$ and $\mathbf{u}_k^{\mathrm{N}}(n)$. Mathematically it is given by

$$\mathbf{T}\mathbf{H}_k^{\mathrm{N}}(n)\left[\begin{array}{cc} \sqrt{\lambda}\mathbf{R}_k^{\mathrm{N}}(n-1) & \sqrt{\lambda}\mathbf{u}_k^{\mathrm{N}}(n-1) \\ \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{X}_{k,n}^T & \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{y}_{k,n} \end{array}\right] = \left[\begin{array}{cc} \mathbf{R}_k^{\mathrm{N}}(n) & \mathbf{u}_k^{\mathrm{N}}(n) \end{array}\right] \tag{4.56}$$

Calculation of block Householder transformation matrix, i.e. $\mathbf{H}_k^A(n)$ is dealt in Section C.3. The updated weight vector is obtained from (4.56) as $\boldsymbol{\psi}_{k,n}^N = \left(\mathbf{R}_k^{\mathrm{N}}(n)\right)^{-1}\mathbf{u}_k^{\mathrm{N}}(n)$

The distributed algorithm is outlined as

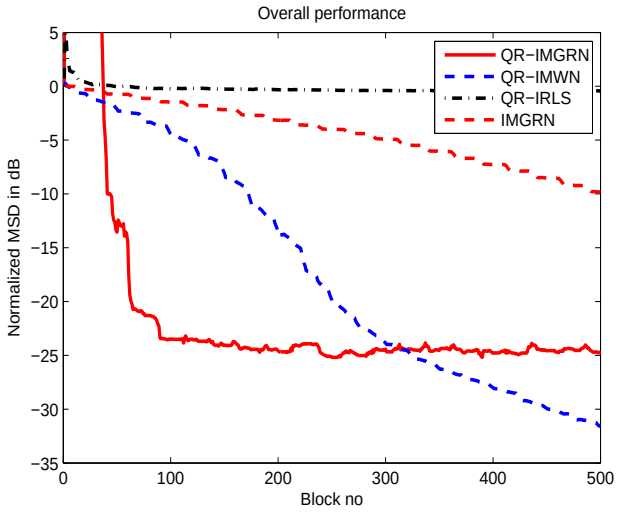

Figure 4.5: 10% outlier in input with random magnitude between $(-3, 3)$ and 50% outlier in output with random magnitude between $(-10, 10)$

$$\begin{cases} \boldsymbol{\psi}_{0,n}^{\mathrm{N}} \leftarrow \mathbf{w}_{n-1}^{\mathrm{N}} \\ \text{For } k = 1 : N \\ \quad \begin{bmatrix} \mathbf{R}_k^{\mathrm{N}}(n) & \mathbf{u}_k^{\mathrm{N}}(n) \end{bmatrix} \leftarrow \mathbf{TH}_k^{\mathrm{N}}(n) \begin{bmatrix} \sqrt{\lambda}\mathbf{R}_{k-1}^{\mathrm{N}}(n) & \sqrt{\lambda}\mathbf{u}_{k-1}^{\mathrm{N}}(n) \\ \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{X}_{k,n}^{T} & \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{y}_{k,n} \end{bmatrix} \\ \text{end} \\ \mathbf{w}_n^{\mathrm{N}} \leftarrow \boldsymbol{\psi}_{N,n}^{\mathrm{N}} \end{cases}$$

4.5.2 Stepwise Representation of QR based Low Communication Incremental Minimum Generalized R Norm

For n^{th} iteration at k^{th} node

(i) Transfer of the estimated parameter vector $\boldsymbol{\psi}_{k-1,n}^{\mathrm{N}}$ from node $k-1$ to node k

(ii) Process the data at k^{th} node

step-1: compute $\boldsymbol{\psi}_{k-1,n}^{\mathrm{N}} = \left(\mathbf{R}_{k-1}^{\mathrm{N}}(n)\right)^{-1} \mathbf{u}_{k-1}^{\mathrm{N}}(n)$

step-2: If the norm is GR norm then calculate of w_{ij} from the input.

step-3: Calculation of error $e_{k-1,k,n}(i) = y_{k,(n-1)L+i} - \mathbf{x}_{k,(n-1)L+i}^{T}\boldsymbol{\psi}_{k-1,n}^{\mathrm{N}}$

step-4: Calculation of forgetting factor based upon the indicator function

$$\lambda_{k-2,k-1,n}(i) = \frac{\sum_{j=1,j\neq i}^{L} w_{ij} I\left(t_{ij}\right) I\left(t_{ji}\right)}{e_{k-2,k-1,n}(i)}$$

$$t_{ij} = e_{k-2,k-1,n}(i) - e_{k-2,k-1,n}(j)$$

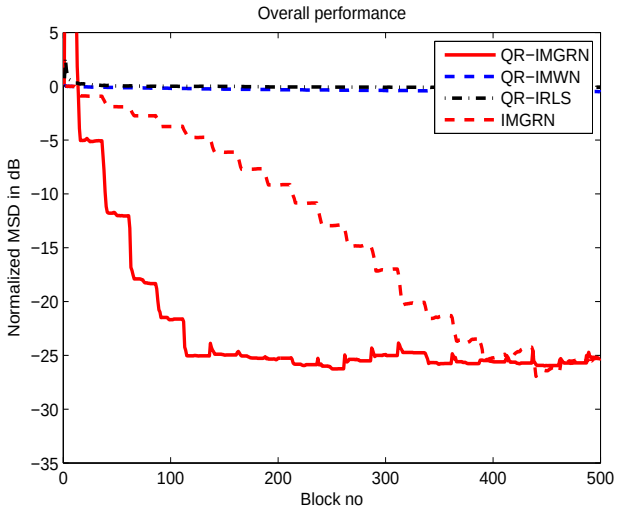

Figure 4.6: 50% outlier in input with random magnitude between $(-3, 3)$ and 50% outlier in output with random magnitude between $(-10, 10)$

$\boldsymbol{\varepsilon}^{\mathrm{N}}_{k-1,k,n} = \mathrm{diag}\left\{\varepsilon^{\mathrm{N}}_{k-1,k,n}(1), \varepsilon^{\mathrm{N}}_{k-1,k,n}(2), \cdots, \varepsilon^{\mathrm{N}}_{k-1,k,n}(L)\right\}$

step-5: Compute

$$\overline{\mathbf{R}}^{\mathrm{N}}_{k}(n) = \begin{bmatrix} \sqrt{\lambda}\mathbf{R}^{\mathrm{N}}_{k}(n-1) \\ \sqrt{\boldsymbol{\varepsilon}^{\mathrm{N}}_{k-1,k,n}}\mathbf{X}^{T}_{k,n} \end{bmatrix}$$

$$\overline{\mathbf{u}}^{\mathrm{N}}_{k}(n) = \begin{bmatrix} \sqrt{\lambda}\mathbf{u}^{\mathrm{N}}_{k}(n-1) \\ \sqrt{\boldsymbol{\varepsilon}^{\mathrm{N}}_{k-1,k,n}}\mathbf{y}_{k,n} \end{bmatrix}$$

step:6

Multiplying block Householder transformation matrix $\mathbf{H}^{\mathrm{N}}_{k}(n)$ to the (4.37) and (4.38) to update and then $\mathbf{T}$ to erase the zero rows to obtain $\mathbf{R}^{\mathrm{N}}_{k}(n)$ and $\mathbf{u}^{\mathrm{N}}_{k}(n)$

$\mathbf{T}\mathbf{H}^{\mathrm{N}}_{k}(n)\overline{\mathbf{R}}^{\mathrm{N}}_{k}(n) = \mathbf{R}^{\mathrm{N}}_{k}(n)$

Similarly

$\mathbf{T}\mathbf{H}^{\mathrm{N}}_{k}(n)\overline{\mathbf{u}}^{\mathrm{N}}_{k}(n) = \mathbf{u}^{\mathrm{N}}_{k}(n)$

step-7:Compute

$\boldsymbol{\psi}^{\mathrm{N}}_{k,n} = \left(\mathbf{R}^{\mathrm{N}}_{k}(n)\right)^{-1}\mathbf{u}^{\mathrm{N}}_{k}(n)$

(iii) Transmit the processed data to the $(k+1)^{th}$ node

$\mathbf{R}^{\mathrm{N}}_{k}(n)$ and $\mathbf{u}^{\mathrm{N}}_{k}(n)$

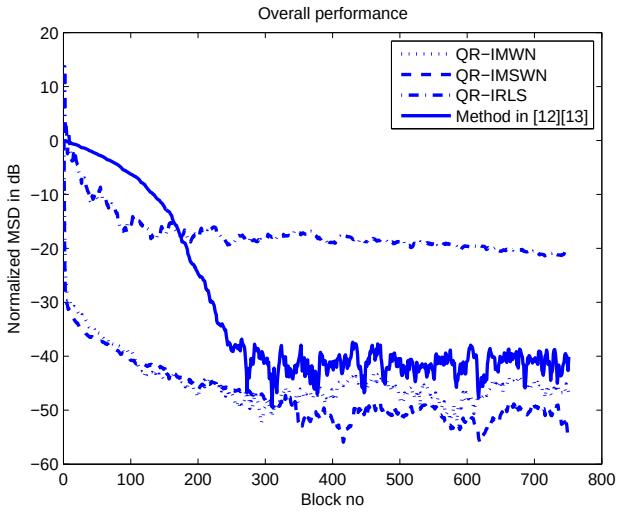

Figure 4.7: 10% outliers in output with random magnitude between $(-10, 10)$

4.5.3 Communication Complexity

In this low communication scheme, the k^{th} node transfers the estimated parameter vector $\boldsymbol{\psi}_{k,n}^{A}$ to the $(k+1)^{th}$ node after every block. The number of data samples to communicate in one communication is p, which is the order of the model. Therefore the communication complexity on average, after one time interval, is p/L. It is shown in Section 4.4.5, that the communication complexity of QR-IMWN is $((0.5)p(p+1)+p)/L$, which is higher by $(0.5)p(p+1)/L$ than the amount in the low communication strategy. Hence a reduced communication over head is obtained with the low communication scheme compared to general scheme.

4.5.4 Simulation Results and Discussion

In this simulation the number of sensor nodes, input data, model parameters, block size and number of input data are same as given in previous simulation. The block forgetting factor is chosen to be 0.8.

For QR-LCIMWN

In the present case the model parameter, the number of nodes and the noise power are same as used in the simulation study for QR-IMWN. The generation of input and desired data are also identical. The results of this scheme obtained through simulation

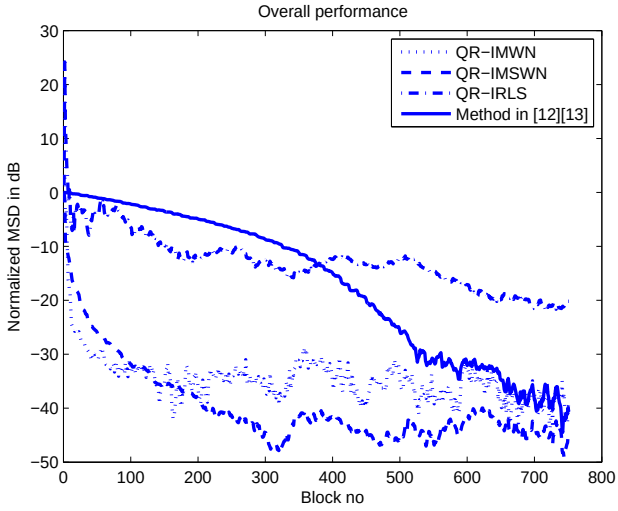

Figure 4.8: 50% outliers in output with random magnitude between $(-10, 10)$

are compared with the QR-IMWN. The density of outliers considered in this case are 10% and 50% of the desired data having magnitude randomly varying between $(-10, 10)$. The block forgetting factor is set at 0.71. The number of data samples used at every node is 3000. The simulation results are shown in Figures 4.9 and 4.10, which are obtained by averaging over 50 independent experiments. These figures demonstrate that the proposed QR-LCIMWN algorithm is robust against outliers. Comparing with QR-IMWN, the MSD converges faster but optimum performance is less. In the QR-IMWN algorithm both the R and u are passed from node (k-1) to node k during block n. These two terms effectively summarize all the available data up to block (n-1) plus all the data at nodes 1 to (k-1) at block n. These two terms are used: (i)to provide a weight estimate based on all this earlier data with which to calculate the block error vector and hence the forgetting factor associated with a Wilcoxon norm;(ii)in combination with local data available at node k at block n to update these terms themselves. In the QR-LCIMWN algorithm only the weight vector estimate is passed between the nodes. This allows step(i) to be implemented at node k since the error vector can be calculated and hence the block forgetting factor. However step(ii) is replaced with a local update of the two terms based only on the local data available at that node. Thus in the QR-IMWN algorithm each node benefits from its neighbors data both in terms of modifying the cost function (through

the forgetting factor) and in updating R and u, whereas in QR-LCIMWN algorithm only the former is possible. This accounts for the degradation in performance of the QR-LCIMWN algorithm compared with the QR-IMWN.

The reason behind the fast convergence of the QR-LCIMWN is due to the following facts. The outliers added to the desired data are random values between $(-10, 10)$, then there is a possibility of some node having weak outliers than the other nodes in the same block. Consider the data in the first time block. As explained in Section 3, in case of QR-IMWN, the previous node, say $k-1$, sends $\mathbf{R}_k^{\mathrm{N}}(1)$ and $\mathbf{u}_k^{\mathrm{N}}(1)$ to the next node ,i.e. k. At node k, the block Householder transformation $\mathbf{H}_k^{\mathrm{N}}(1)$ and the T are multiplied with $\begin{bmatrix} \overline{\mathbf{R}}_k^{\mathrm{N}}(1) & \overline{\mathbf{u}}_k^{\mathrm{N}}(1) \end{bmatrix}$ to get $\begin{bmatrix} \mathbf{R}_k^{\mathrm{N}}(1) & \mathbf{u}_k^{\mathrm{N}}(1) \end{bmatrix}$. Due to the effect of $\mathbf{u}_{k-1}^{\mathrm{N}}(1)$ (which may contain strong outliers) in the multiplication the total information from the new data cannot be extracted. Hence convergence speed decreases. Whereas,in case of QR-LCIMWN, the previous node sends only the estimated parameter, hence if the data in next node contains weak outliers then from $\begin{bmatrix} \sqrt{\varepsilon_{k-1,k,n}^{\mathrm{N}}}\mathbf{R}_k^{\mathrm{N}}(1) & \sqrt{\varepsilon_{k-1,k,n}^{\mathrm{N}}}\mathbf{u}_k^{\mathrm{N}}(1) \end{bmatrix}$ we are able to extract the entire information.

For QR-LCIMGRN

The simulation results given are the average over 50 independent experiments. For this case the simulation is done for every 10% to 50% of outliers in the desired and input data. But only the simulation results for the 10% and 50% are given in Figures 4.11-4.12. The comparision is done between QR-IMGRN and QR-LCIMGRN. These figures demonstrates that the the proposed QR-LCIMGRN is robust against outliers in the desired and input data with inferior performance compared to QR-IMWN. The reson is similar to the reason given for QR-LCIMWN.

4.6 Conclusion and Future Work

This chapter proposes a new QR based robust distributed algorithm using the Wilcoxon and generalize rank scores which exhibits robust performance against outliers in the desired data. The results are compared with those obtained by the general block

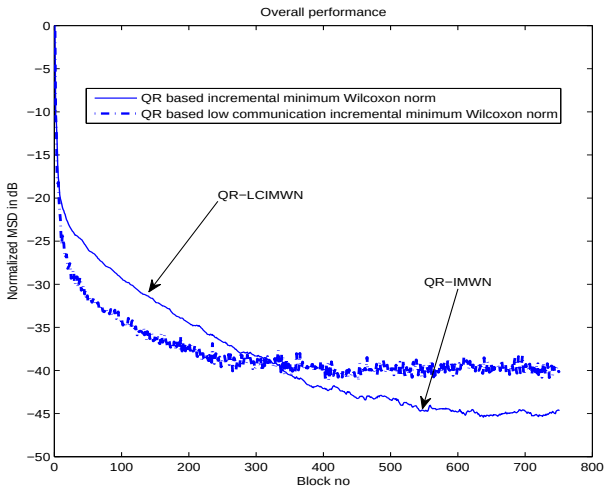

Figure 4.9: 10% outlier in output with random magnitude between $(-10, 10)$

distributed RLS algorithm using BHT. It is in general observed that the performance of conventional block distributed RLS algorithm substantially degrades in the presence of outliers in the desired data. However the proposed method provides robust performance even in the presence of strong and high density outliers. Thus the proposed method is expected to perform well in real world environment. The proposed QR based low communication scheme is attractive as it offers less communication overhead and hence a saving in power.

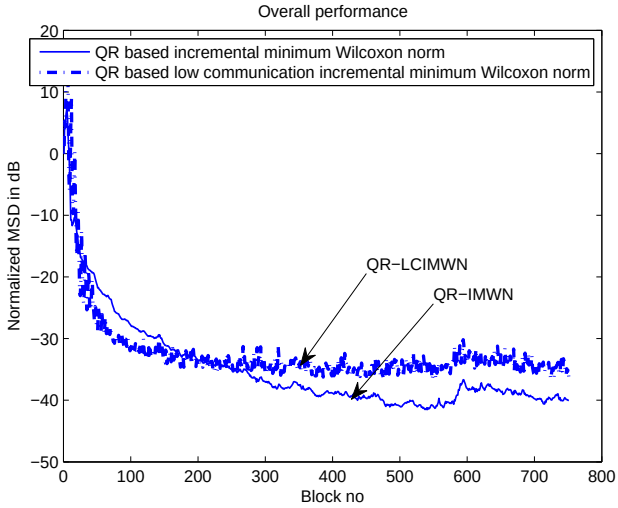

Figure 4.10: 50% outlier in output with random magnitude between $(-10, 10)$

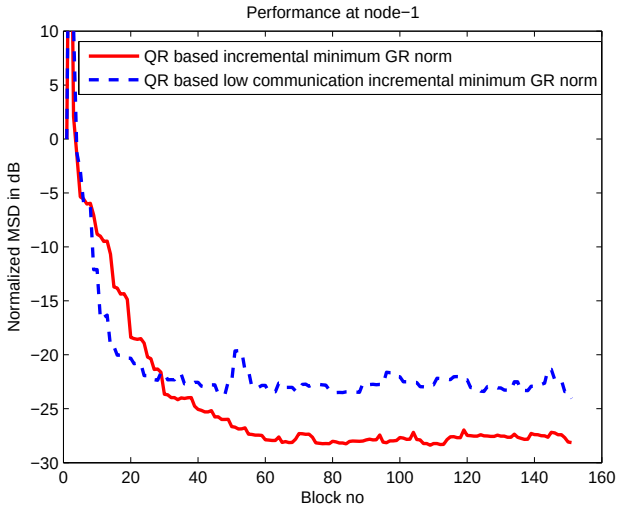

Figure 4.11: 40% outlier in input with random magnitude between $(-3, 3)$ and 10% outlier in output with random magnitude between $(-10, 10)$

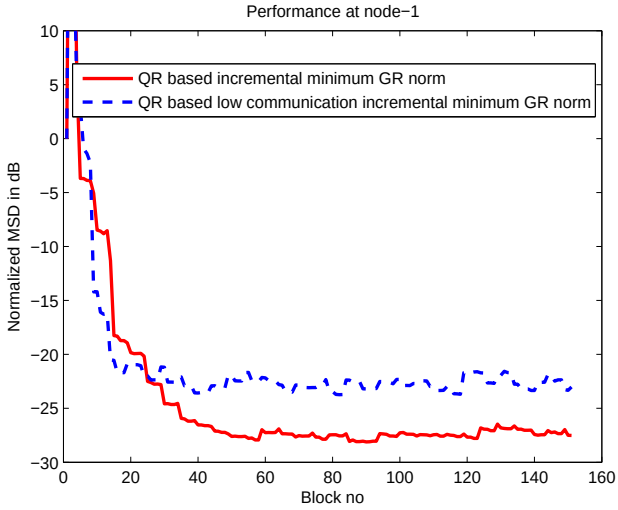

Figure 4.12: 40% outlier in input with random magnitude between $(-3, 3)$ and 40% outlier in output with random magnitude between $(-10, 10)$

Chapter 5

Robust Distributed Affine Projection Algorithm to Handle Outliers in the Desired Data

Conventional affine projection algorithm(APA) based on the least square error cost function is not robust against outliers in the desired data. This chapter deals with the development of a robust affine projection algorithm(R-APA) based on the PLS formulation of the Wilcoxon norm. The new algorithm is then applied to the estimation problem using the WSNs. In this chapter the Wilcoxon norm based cost function is used as the robust norm for minimization purpose. The proposed algorithm acts as a compromise between the IMWN and QR-IMWN in terms of its convergence speed and the computational complexity. It is shown that in case of colored input the R-APA performs similarly to the QR-IMWN with less number of computations and faster convergence whereas the performance of the IMWN and its variants deteriorate. It is demonstrated through the simulation based experiments that the proposed method is robust against outliers present in the desired data.

5.1 Introduction

As discussed in Section 2.1, distributed signal processing algorithm can be suitably formulated for the efficient use of in-network processing capability of the sensor nodes to increase the lifetime of the WSNs. In this chapter the impirements of the environments considered are the AWGN and impulsive noise. The impulsive noise can be modeled as outliers in the measured data. Different robust distributed strategies have been proposed in the previous chapters to handle outliers in the desired data. Some of the strategies provide faster convergence speed with less parameter estimation capability whereas the others require more number iterations with good estimation potentiality. In Chapter 4 the PLS formulation of the Wilcoxon norm has been made to handle outliers in the desired data. This strategy provides faster convergence speed as well as good estimation capability. However, this strategy requires a large number of computational complexity. Moreover, the gradient based robust distributed strategies given in Chapter-2 (which requires less number of computations) provide poor convergence speed in the presence of the colored input data. If the input data is highly correlated then the performance of these methods deteriorate. It is well known that the affine projection algorithm (APA) performs well in presence of colored input.

This chapter deals with the development a novel R-APA algorithm based on the PLS formulation of the Wilcoxon norm. This strategy requires less computational complexity than offered by the QR-IMWN proposed in Section 4.5 and achieve similar performance to the QR-IMWN method. In addition, this proposed method provides superior convergence and performance compared to the algorithms proposed in Chapter-2 with more computational complexity. Thus, the new algorithm acts as a good compromise between the QR-IMWN and the IMWN in terms of computational complexity and convergence speed. The proposed R-APA is motivated from [41], where a distributed affine projection algorithm is proposed.

5.2 Problem Formulation

The problem undertaken here is same as given in Section 4.2 i.e. to estimate the parameters associated with a given environment using the environmental data measured by the sensor nodes dispersed across the environment of interest. Considering the model to be a linear model, the entire spatio-temporal output and input data can be related by

$$\mathbf{y}(n) = \mathbf{X}^T(n)\,\mathbf{w} + \mathbf{v}(n) \tag{5.1}$$

The objective is to estimate $\mathbf{w}$ in (5.1) from $\mathbf{y}(n)$ and $\mathbf{X}(n)$ which can be formulated as an optimization problem as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\|\mathbf{y}(n) - \mathbf{X}^T(n)\,\mathbf{w}\right\|_* \tag{5.2}$$

In terms of geometry the objective of (5.2) is to estimate $\mathbf{w}^*$ so that norm distance between the $\mathbf{X}^T(n)\,\mathbf{w}^*$ to the desired space is minimum.

5.3 Proposed Robust Affine Projection Algorithm(R-APA)

For the estimation of the parameter $\mathbf{w}$ in (5.1) the R-IAPA algorithm based on the Wilcoxon norm is proposed. The Wilcoxon norm as explained in Section 4.3.1 is a pseudo norm. As given in Section 4.3.1 the Wilcoxon norm can be represented as a PLS as shown in (5.3).

$$\|\mathbf{v}\|_w = \sum_{i=1}^{l} \varepsilon_i^w \left(v_i\right)^2 \tag{5.3}$$

where,

$$\varepsilon_i^w = \begin{cases} \frac{a(R(v_i))}{v_i} & for\ v_i \neq 0 \\ a^{'}\left(R\left(0\right)\right) & for\ v_i = 0 \end{cases} \tag{5.4}$$

5.3.1 Proposed Methodology

In order to estimate the parameter $\mathbf{w}$, the proposed R-APA algorithm based on the PLS formulation of the Wilcoxon norm is used. This is achieved by using the weighting factor obtained from the Wilcoxon norm for modification of the auto- and cross-correlation matrices of the IAPA algorithm [41]. Using the PLS formulation (5.3) the objective (5.2) can be changed to

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\|\mathbf{y}\left(n\right) - \mathbf{X}^T\left(n\right)\mathbf{w}\right\|^2_{\Lambda^A(n)} \tag{5.5}$$

where

$$\Lambda^A\left(n\right) = \Lambda\left(n\right) \times \boldsymbol{\varepsilon}(n) \tag{5.6}$$

$$\boldsymbol{\varepsilon}(n) = \left[\begin{array}{cccc} \boldsymbol{\varepsilon}^a_{N,\cdots,N,n} & \boldsymbol{\varepsilon}^a_{N,\cdots,N,n-1} & \cdots & \boldsymbol{\varepsilon}^a_{N,\cdots,N,1} \end{array}\right] \tag{5.7}$$

(which is given in (4.44))

$$\Lambda(n) = \text{diag}\left[\begin{array}{cccc} \lambda^{n-1}\mathbf{1}_N & \lambda^{n-2}\mathbf{1}_N & \cdots & \lambda^{0}\mathbf{1}_N \end{array}\right] \tag{5.8}$$

The $\boldsymbol{\varepsilon}(n)$ is the weighting factor (which is present in (5.3)) corresponds to the entire spatio-temporal error vector.

The cost function (5.5) can be written as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left(\mathbf{y}(n) - \mathbf{X}(n)\mathbf{w}\right)^T \Lambda^A(n) \left(\mathbf{y}(n) - \mathbf{X}(n)\mathbf{w}\right) \tag{5.9}$$

The problem (5.9) can be written as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left(\overline{\mathbf{y}}(n) - \overline{\mathbf{X}}(n)\mathbf{w}\right)^T \left(\overline{\mathbf{y}}(n) - \overline{\mathbf{X}}(n)\mathbf{w}\right) \tag{5.10}$$

where, $\overline{\mathbf{y}}(n) = \sqrt{\boldsymbol{\Lambda}^A(n)}\mathbf{y}(n)$ and $\sqrt{\boldsymbol{\Lambda}^A(n)}\mathbf{X}(n)$

The solution to the problem (5.10) is given as [34, 35]

$$\overline{\mathbf{w}}^* = \overline{\mathbf{R}}^{-1}\overline{\mathbf{p}} \tag{5.11}$$

where, $\overline{\mathbf{R}} = E\left[\overline{\mathbf{X}}^T(n)\overline{\mathbf{X}}(n)\right]$ and $\overline{\mathbf{p}} = E\left[\overline{\mathbf{X}}^T(n)\overline{\mathbf{y}}(n)\right]$. Thus, the solution is based on the autocorrelation of the weighted input and cross correlation of the weighted input and weighted desired values. However, the estimation of (5.11)(which is a PLS cost function) requires the entire spatio-temporal data set due to the presence of weighting factor $\boldsymbol{\varepsilon}(n)$ (which depends upon the entire spatio-temporal error vector). Thus, it is not amenable for distributed implementation. The incremental strategy [41] requires

$$J(\mathbf{w}) = \sum_{k=1}^{N} J_k(\mathbf{w}) \tag{5.12}$$

where, $J(\mathbf{w})$ is the global cost function and $J_k(\mathbf{w})$ is the local cost function at any node say k which is given as

$$J_k(\mathbf{w}) = \min_{\mathbf{w}} \left\| \mathbf{y}^k(n) - \mathbf{X}^k(n) \right\|^2_{\mathbf{\Lambda}^{k,A}(n)} \tag{5.13}$$

Here $\mathbf{y}^k(n)$ and $\mathbf{X}^k(n)$ is the entire input and desired data at node k respectively. The forgetting factor present in (5.13) is given as

$$\mathbf{\Lambda}^{k,A}(n) = \left[\begin{array}{cccc} \lambda^{n-1}\varepsilon_{k-1,k,n} & \lambda^{n-2}\varepsilon_{k-1,k,n-1} & \cdots & \lambda^{0}\varepsilon_{k-1,k,1} \end{array} \right] \tag{5.14}$$

In order to implement the incremental strategy the global cost function can be changed similar to (3.9). However, it requires entire data set present in the sensor node starting from first time to n^{th} time. Therefore, it can not be implemented in recursive manner. In order to implement in the recursive manner the global cost function can be changed to a similar form like (3.11), which is given as

$$\min_{\mathbf{w}} \sum_{k=1}^{n} \sum_{i=1}^{n/L} \left\| \mathbf{y}_{k,(iL+1):(i+1)L} - \mathbf{X}^T_{k,(iL+1):(i+1)L}\mathbf{w} \right\|^2_{\mathbf{\Lambda}^{(i)b-\mathrm{N}}(n)} \tag{5.15}$$

Following (5.9-5.11) the optimum parameter corresponds to the cost function (5.15) can be calculated as

$$\overline{\mathbf{w}}^* = \overline{\mathbf{R}}^{-1}\overline{\mathbf{p}} \tag{5.16}$$

In order to achieve the optimum parameter (5.16) recursively, a sliding window is used. This method helps to facilitate the implementation of R-APA which requires a block of data for calculation of the auto correlation as well as cross correlation and also suitable for the calculation of the weighting factor based on the Wilcoxon norm. In order to facilitate the implementation of sliding window method let us define the entire spatio-temporal input and desired data arranged by block manner up to p^{th} block as

$$\mathbf{X}(p) = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_p \end{bmatrix} \qquad \mathbf{y}(p) = \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T & \cdots & \mathbf{y}_p^T \end{bmatrix}^T \tag{5.17}$$

where, $\mathbf{X}_j$ and $\mathbf{y}_j$ are given in (2.2).

Let the estimated parameters by IAPA using $\mathbf{X}(p)$ and $\mathbf{y}(p)$ be $\mathbf{w}_p$. Similarly, $\mathbf{w}_{p+1}$ is obtained using $\mathbf{X}(p+1)$ and $\mathbf{y}(p+1)$. For this purpose, a recursive method is introduced by which $\mathbf{w}_{p+1}$ is estimated from previous estimate $\mathbf{w}_p$ based on the APA [41] given as

$$\mathbf{w}_{p+1} = \mathbf{w}_p - \mu \nabla J(\mathbf{w}_p) \tag{5.18}$$

Since the data collected during $(p+1)^{th}$ block measurement time is present in every sensor node throughout the environment the IAPA method is used to achieve the spatial recursive estimation of the parameters. In IAPA each sensor receives the estimated parameters from the previous node and uses its own measured data to update it. In order to investigate this spatial recursive formulation based on IAPA, let us denote the spatial estimated parameters at node k during $(p+1)$ iteration as $\boldsymbol{\psi}_{k,p+1}$. Therefore, in incremental distributed strategy during $(p+1)^{th}$ spatial recursion, the k^{th} node transmits the data $\boldsymbol{\psi}_{k,p+1}$ to $(k+1)^{th}$ node. Using $\boldsymbol{\psi}_{k,p+1}$ and the measured data at $(p+1)^{th}$ block, $(k+1)^{th}$ sensor node calculates $\boldsymbol{\psi}_{k+1,p+1}$. Using IAPA recursive estimation is given by

$$\boldsymbol{\psi}_{k+1,p+1} = \boldsymbol{\psi}_{k,p+1} - \mu \nabla J\left(\boldsymbol{\psi}_{k,p+1}\right) \tag{5.19}$$

Following (5.18) for all the sensor nodes starting from node 1 to node N during $(p+1)^{th}$ block, $\mathbf{w}_{p+1}$ is calculated as

$$\mathbf{w}_{p+1} \equiv \boldsymbol{\psi}_{N,p+1} \leftarrow \cdots \boldsymbol{\psi}_{1,p+1} \leftarrow \boldsymbol{\psi}_{0,p+1} \equiv \mathbf{w}_p \tag{5.20}$$

5.3.2 Proposed R-IAPA Algorithm

This section deals with the development an R-IAPA based on the weighted input and weighted desired data given in the Section 5.3.1. This algorithm is designed based on the following approximation

$$\mathbf{w}_{p+1} = \mathbf{w}_p - \sum_{k=1}^{N} \mu_k \left(\delta \mathbf{I} + \nabla^2 J_k\left(\boldsymbol{\psi}_{k-1,p+1}\right)\right)^{-1} \nabla \left[J_k\left(\boldsymbol{\psi}_{k-1,p+1}\right)\right] \tag{5.21}$$

Further, using the approximation based upon the data in a sliding window in (5.21), we get

$$\mathbf{w}_{p+1} = \mathbf{w}_p - \mu_k \sum_{k=1}^{N} \overline{\mathbf{X}}_{k,p+1} \left(\delta \mathbf{I} + \overline{\mathbf{R}}_{x,k}\right)^{-1} \overline{\mathbf{e}}_{k,i} \tag{5.22}$$

where,

$$\overline{\mathbf{X}}_{k,p+1} = \left[\begin{array}{cccc} \mathbf{x}_{k,pl+1} & \mathbf{x}_{k,pl+2} & \cdots & \mathbf{x}_{k,(p+1)l} \end{array} \right] \times \sqrt{\lambda_{k,p+1}^{T}} \tag{5.23}$$

$$\overline{\mathbf{R}}_{x,k} = \frac{1}{T}\left[\overline{\mathbf{X}}_{k,i} \times \overline{\mathbf{X}}_{k,i}^{T}\right] \tag{5.24}$$

$$\overline{\mathbf{e}}_{k,p+1} = \sqrt{\lambda_{k,p+1}} \times \left[\mathbf{y}_{k,p+1} - \mathbf{X}_{k,p+1}^{T} \psi_{k-1,p+1}\right] \tag{5.25}$$

Further, incorporating $\overline{\mathbf{X}}_{k,p+1}$, $\overline{\mathbf{R}}_{x,k}$ and $\overline{\mathbf{e}}_{k,p+1}$, we get

$$\mathbf{w}_{p+1} = \mathbf{w}_p - \sum_{k=1}^{N} \mu_k \mathbf{X}_{k,p+1}^{T} \sqrt{\lambda_{k,p+1}^{T}} \left(\delta \mathbf{I} + \mathbf{X}_{k,p+1} \lambda_{k,p+1}^{T} \mathbf{X}_{k,p+1}^{T}\right)^{-1} \sqrt{\lambda_{k,p+1}^{T}} \left[\mathbf{y}_{k,p+1} - \mathbf{X}_{k,p+1} \boldsymbol{\psi}_{k-1,p+1}\right] \tag{5.26}$$

The spatial updation of (5.26) is given by

$$\boldsymbol{\psi}_{k,p+1} = \boldsymbol{\psi}_{k-1,p+1} + \mu_k \overline{\mathbf{X}}_{k,p+1} \left(\delta \mathbf{I} + \overline{\mathbf{X}}_{k,p+1} \overline{\mathbf{X}}_{k,p+1}^T \right)^{-1} \left[\overline{\mathbf{y}}_{k,p+1} - \overline{\mathbf{X}}_{k,p+1}^T \boldsymbol{\psi}_{k,p+1} \right] \quad (5.27)$$

5.4 Simulation Results

This section deals with simulation based experiments to assess the robust estimation performance of the proposed methods. The number of nodes taken in the environment is 5. The model parameters to be estimated from the environment is

$$\left[\begin{array}{ccccc} 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} & 1/\sqrt{5} \end{array} \right] \quad (5.28)$$

.

For the simulation study the white and the colored input data are taken into consideration. For the case of white input, the data generated is a zero mean uniformly distributed random number lying between $(-0.5, 0.5)$. For the generation of the colored input the white input is passed through a FIR filter [41] given as

$$H(z) = 0.1 - 0.2z^{-1} - 0.3z^{-2} + 0.4z^{-3} + 0.4z^{-4} - 0.2z^{-5} - 0.1z^{-6} \quad (5.29)$$

This model gives the colored input data having eigen value spread $\lambda_{\max}/\lambda_{\min} \approx 91.85$ The output data is generated by passing the input data through the model (5.28) having feed forward parameters. The output data is added with white Gaussian noise having an SNR 30dB. Then the desired data is contaminated with outliers at random positions having magnitude randomly varying between $(-10, 10)$. The outliers are modeled as uniform random variables between the range (-10, 10). This is similar to the situation where the variance of the desired data increases due to outliers. In the present case the form of IMWN, QR-IMWN and R-APA does not change with the distribution of outliers' random variable change. The block size is taken to be 40. The number of input samples used are 8000. Simulations have been carried out separately for colored and white inputs with outliers of $10\%, 20\%, 30\%$ of the desired data. However, only the simulation results for 10% and 30% are shown in

this case. The results presented are the average of 50 independent experiments. In the simulation results the comparision is done among QR-IMWN, IMWN and R-APA based on the Wilcoxon norm. Figures 5.1 and 5.2 correspond to the case of white input data. From these simulation results it is observed that for the white input data the convergence speed of R-APA is in between the QR-IMWN and IMWN having the same performance. Figures 5.3 and 5.4 correspond to the colored input data. It is observed that in case of highly correlated input data and the little large step size the IMWN does not converge whereas the QR-IMWN and R-APA algorithms provide a better solution after converge. Steady state performance of the proposed robust distributed affine projection algorithm is given in the Tabe -5.1 for different block size. Centralized performance of the proposed algorithm is given in the Figures 5.5, 5.6. From the simulation results it is found that for small step size the IMWN convergence to the optimal solution with less convergence speed compared to the previous two algorithms.

In case of colored input data both the RAPA and the QR-IMWN achieve satisfactory convergence due to the whitening term present in the update equation. In case of QR-IMWN the whitening term i.e. the inverse of the pseudo autocorrelation matrix depends on the previous data by some forgetting factor. Thus, the effect of previous outliers is present in the whitening term. Therefore, performance decreases. However, in case of RAPA the whitening term does not depend on the previous data hence the effect of previous outliers is less in the update equation. Hence, in case of colored input data R-APA performs better than QR-IMWN.

5.5 Calculation of Computational Complexity

This section deals with the computational complexity calculation of the IMWN, R-APA and QR-IMWN algorithms. Firstly, the computational complexity of the IMWN is given. Based on the indicator function, the calculation of the rank of one element in a vector of dimension l requires $l-1$ number of comparisons. Thus, to calculate the rank of all l elements $0.5 \times l(l-1)$ number of comparisons are required. In order to calculate the score value, l number divisions and a same number of subtractions

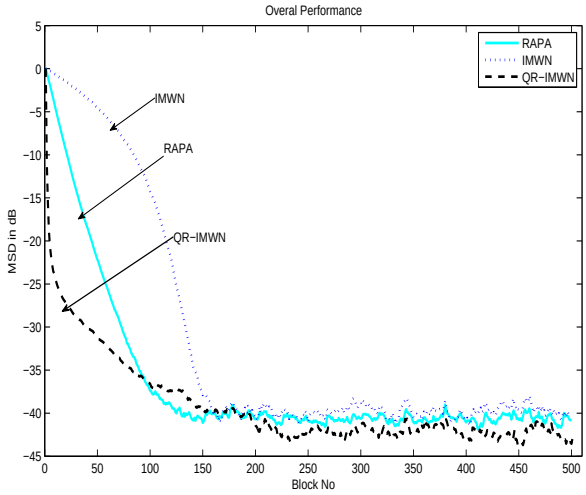

Figure 5.1: Overall convergence performance with 10% outliers in the desired data with magnitude between (-5,5)

are required. Since, the same score value is to be used in every iteration there is no need of the calculation of the score value always. Therefore, the score value needs to be calculated once and then it will be continued for the next iterations. Thus, in the other iteration only the rank is to be calculated and then the corresponding score value is to be assigned for different rank values. Thus, for other iterations other than the first iteration $0.5 \times l(l-1)$ number of comparisons are required to calculate the score value. For calculation of the errors in one block pl number of multiplications and pl number of additions are needed. During the update operation $2lp$ number of multiplications and lp number of additions or subtractions are required. Thus, the total number of operations are: $0.5 \times l(l-1)$ number of comparisons; $3lp$ number of multiplications; and $2lp$ number additions.

Secondly the computational complexity of QR-IMWN is given for the comparison purposes. Similar to the previous explanation about the calculation of the score, the same number of computations are also required for the calculation of the score in the case of QR-IMWN. Further, there is a need for the calculation of the weighting factor. In order to calculate the weighting factor corresponding to one block p divisions are required. Then this weighting factor is to be multiplied with the data so there is need of $l(p+1)$ number of multiplications. Further, the computational complexity is to be

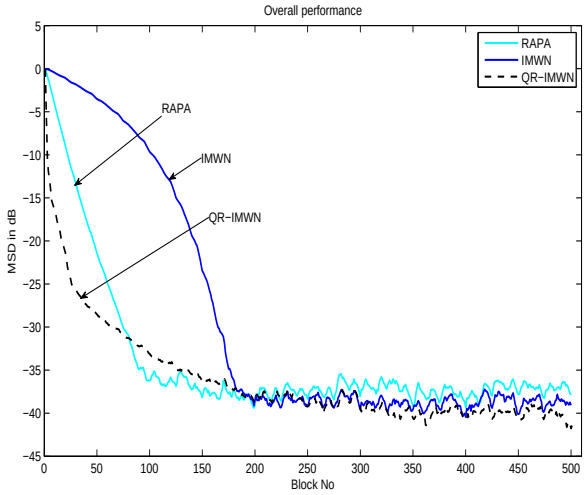

Figure 5.2: Overall convergence performance with 30% outliers in the desired data with magnitude between (-5,5)

calculated for the BHT matrix formation and multiplication of the BHT matrix with the input and the desired data. The number of operations requires for the calculation of the BHT matrix depends upon the column number. Consider for the case of column number m. The number of non zero elements present in the column m is $(l+1)$. Thus, the number of operations required to calculate the term $\frac{\mathbf{z}\mathbf{z}^T}{\|\mathbf{z}\|^2}$ are: $(l+1)^2 + (l+1)$ number of multiplications; $(2l+1)$ number of additions or subtractions; and $(l+1)^2$ number of divisions. Further, the number of operations required to calculate the first element of the vector are: $l+1$ number of multiplications; $l+1$ number of additions. Hence, in total $(l+1)^2$ number of multiplications are required for the multiplication of the BHT with the data matrix. For the multiplication of the entire BHT matrix with the data matrix of size $p-m+2$. Thus, the total number of multiplications is $(p-m+2) \times (l+1)^2$.

This paragraph deals with the computational complexity calculation for the R-APA algorithm. The same score function and the forgetting factor are to be calculated in this case whose computational complexity is already given in previous paragraphs. There is only need to calculate the computational complexity for the update equation shown in (5.27). To calculate the correlation matrix present in (5.27) l^2p number of multiplications and $l^2(p-1)$ number of additions are required. Then the multiplication

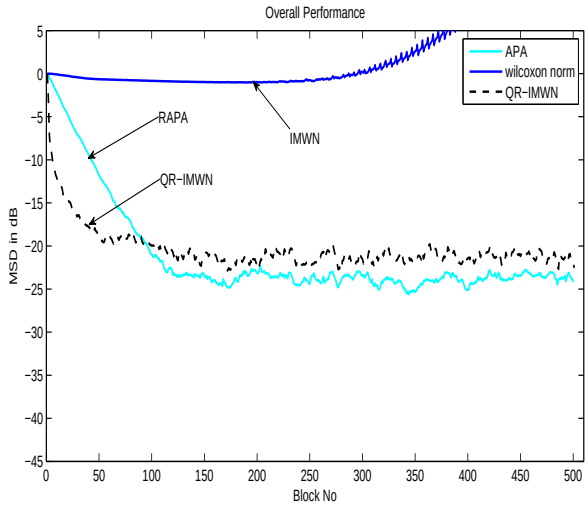

Figure 5.3: Overall convergence performance with 10% outliers in the desired data with magnitude between (-5,5)

Block size	5	10	15
Steady state performance	-50.7671	-50.2746	-51.4548

Table 5.1: Steady state performance of the robust APA for different block size

of the correlation term with the left term i.e. requires a pl^2 number of multiplications and $pl(l-1)$ number of additions. Further, the last multiplication operation requires a pl number of multiplications and $p(l-1)$ number of additions. Computational complexity comparison for different methods is given in Table 5.2.

Operation	IMWN	QR-IMWN	R-APA
Comparison	$0.5 \times l(l-1)$	$0.5 \times l(l-1)$	$0.5 \times l(l-1)$
Addition	$2lp$	$4l+3$	$2l^2p - l^2 + lp - p$
Multiplication	$3lp$	$0.5p^2l^2 + p^2l + 1.5pl^2 + 2l^2$ $+0.5p^2 + 4pl + 8l + 1.5p$	$l^2p + pl^2$ $3pl + l$
Division	0	$l^2 + l + p$	p
Square root	0	l	l
Matrix Inverse	0	1	1

Table 5.2: Comparison of computational complexity

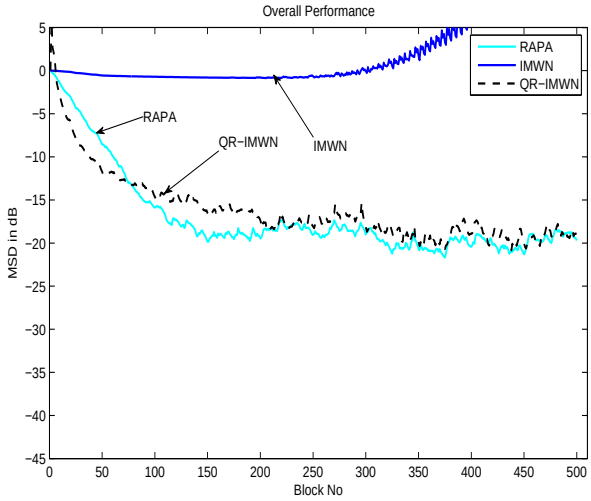

Figure 5.4: Overall convergence performance with 10% outliers in the desired data with magnitude between (-5,5)

5.6 Conclusion

It is observed that the R-APA based on the minimum-Wilcoxon-norm is robust against outliers in the desired data for both white and colored input data. Since the computational complexity of the proposed algorithm is less than the QR-LCIMWN one it is advantageous to use this algorithm in distributed case. Based on the choice of block length the computational complexity of the algorithm is controlled. Therefore, the block length can be chosen based on the availability of the computation units. The convergence analysis of the algorithm can be carried out as future work by using the the theory of rank statistics [16, 42].

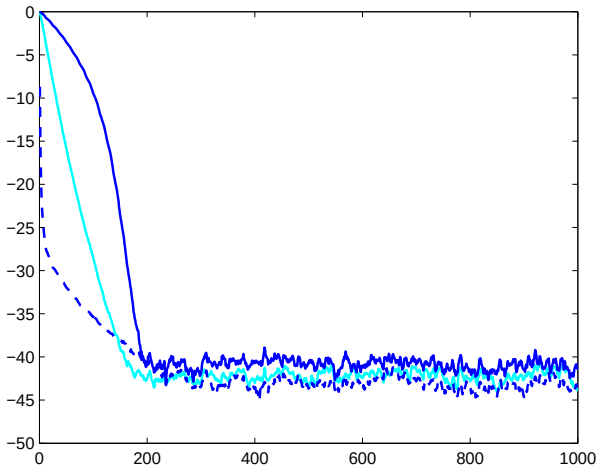

Figure 5.5: Overall central convergence performance with 10% outliers in the desired data with magnitude between (-10,10)

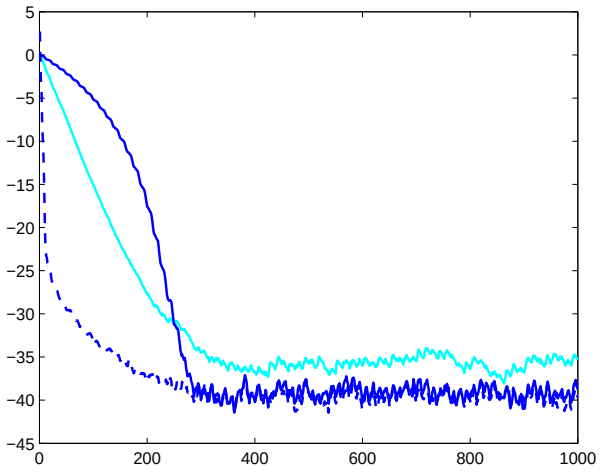

Figure 5.6: Overall central convergence performance with 10% outliers in the desired data with magnitude between (-10,10)

Chapter 6

Energy Efficient Environment Monitoring Using Minimum Volume Ellipsoid Method

Environment monitoring is one important application of WSNs. In this chapter, a novel way is proposed to estimate the state of the environment using distributed signal processing. A novel distributed strategy is proposed to find the incremental path connecting each sensor node. In order to send the data in the wireless channel, the sensor data needs to be quantized. To decrease the effect of the quantization error, a robust technique is also proposed. The proposed techniques have been validated through simulation based experiments.

6.1 Introduction

The WSNs consist of large number of sensor nodes connected through wireless communication and are dispersed in an environment of interest to collect some useful environmental data [1]. This data is used to estimate some parameters of interest that could explain the physical condition of the environment. Effective solution of such problems finds a number of applications in real life situation. Some interesting applications are precision agriculture, monitoring hazardous material in the environment, monitoring hazardous environments such as a fire in the forest and volcanic eruption. This type of problem requires the estimation of the location where the environment is changing to an abnormal condition. For an example consider the case of precision agriculture, the objective is to find the area where the humidity level has been decreased below some threshold value. Similarly for the case of fire in the forest the objective is to find the location where the temperature value has gone above some threshold value. All these problems are based on the assumption that each sensor knows its position. Since the sensors are spread randomly in a remote area, it is not possible to find the position of each sensor node manually. For this case the position of the sensor can be calculated by using some algorithms already reported in the literature [43, 44].

Classical estimation method requires a fusion center (FC) with high power processing capability. In the case of environment monitoring each sensor node sends its position and measured data to the FC. The FC processes the received data and estimates the location where the abnormal condition has occurred. This type of so-

lution for such problem requires a large number of communication overheads. Hence one of the research ares in WSNs is to develop energy efficient strategy that can be achieved using in-network processing capability of sensor nodes. This problem has been addressed in the literature and has been dealt in previous chapters in which a common parameters of interest are estimated in distributed manner by processing the measured data locally and sharing their estimated parameters with neighbor nodes to achieve the global estimation.

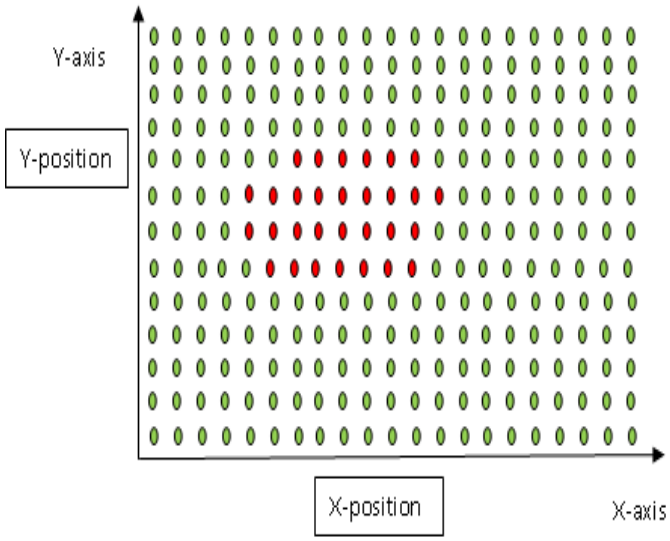

Figure 6.1: Sensors are spread in regular manner in the environment of interest

From the environment monitoring viewpoint, the estimation problem is explained here. In the beginning, a large number of sensor nodes are spread in the area of interest. Then these nodes measure the environmental data after a regular time interval. In practice it requires all sensor positions where the temperature is above the threshold level at the FC. Using this data the FC identifies the abnormal location. Accordingly appropriate action is initiated by the control center to overcome this problem. Due to large communication overheads to route position data from the sensor to the FC, the energy of the sensors near the FC decreases fast and after some time it becomes zero. Then the FC would not able to communicate with other sensor

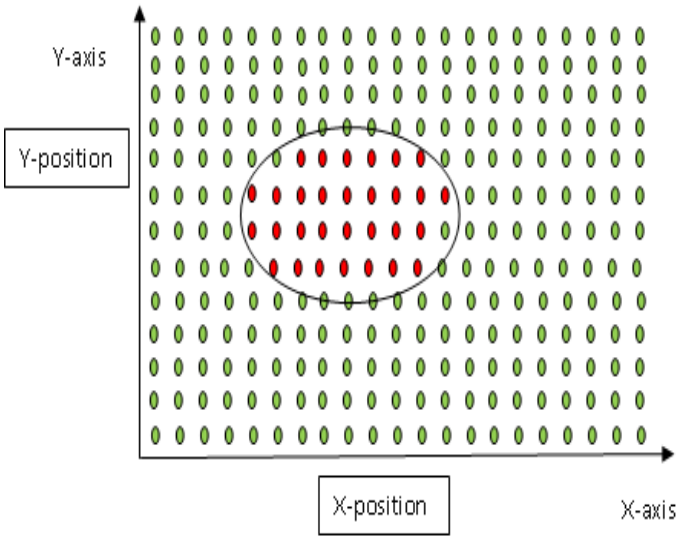

Figure 6.2: Sensors in red and blue measure event above threshold and below threshold level respectively

nodes. This situation is called the dead condition because the objective of the WSNs cannot be fulfilled. However, a considerable amount of energy of the sensor network remains unused. Thus, the aim is how to use almost equal amount of energy at every node. The solution can only be achieved by using in-network processing based distributed strategy.

In this chapter, the incremental strategy is used for distributed estimation of the area. Previous literature [5] indicates that the Hamiltonian cycle is required for finding the incremental path. In this case all the sensor nodes' positions need to send to the FC for finding the Hamiltonian cycle. This also requires large number of communication overheads. In order to get rid of this, a novel method is proposed to find the incremental path based on the local information. A logical proof of the proposed method is provided. The present distributed method to find the incremental path is incorporated in the method of estimation to achieve the desired objective.

To further clarify the problem an example is given. Suppose the sensor nodes are dispersed in a regular manner in an area as shown in Figure 6.1. Let the left

bottom sensor represents the FC of the sensor network. Consider red sensors measure temperature values above the threshold level. Classically these sensors' positions are required at the FC and the FC estimates the abnormal area. The sensors 3,4,5,6,7,8 near to the FC have to route all sensor nodes' positions. Thus, the sensor near to the FC losses substantial amount of energy during one process. Hence, within short time after the deployment these sensors lose their energy. This leads to a situation where the sensors near to the FC become out of energy where as sensors away from the FC still contain appreciable amount of energy. However, these sensors can not transmit their measured data to the FC by multihop communication. Thus, the objective of sensor network can not be fulfilled.

In order to avoid the above multihop communication based strategy and to facilitate the estimation process, a distributed scheme needs to be found out. The above problem can be solved if by distributed manner a shape is formed enclosing the area where measured value is above the threshold level and the parameters associated with the shape are sent to the FC. Ellipsoid is a general shape which can be used to enclose any type of shape with less amount of error. It is often used for clustering of data [45, 46]. In this case an MVE covering positions of the sensor nodes in the abnormal area is formed by distributed manner to estimate the abnormal area. Finding the MVE covering a finite set is a convex optimization problem. In order to find the MVE using distributed strategy a token is passed across the entire network along the incremental path. When a sensor node receives the token, it receives also priori information about the shape of the MVE and uses its own information to update this and passes the token as well as the parameters of the MVE to the next node defined by the incremental strategy. In short the chapter deals with the development of:

1. a novel method to find out the global incremental cyclic path connecting each sensor node by using the local decision;

2. an incremental distributed strategy based MVE covering the positions where abnormality has occurred in absence of environmental noise;

3. an incremental distributed strategy based MVE covering the position where abnormality has occurred in presence of environmental noise;

4. a robust distributed MVE estimation based on the quantization parameter data;

5. an incremental distributed strategy for nonconvex scenario.

The symbols used in the chapter are listed:

$\mathbf{S}$	Sensor position matrix;
A	Area of the environment;
$\mathbf{E} - \mathbf{E_a}$	The element is present in set $\mathbf{E}$ but not in set $\mathbf{E}_a$;
vol	Volume of an ellipsoid;
$\mathbf{0}_p$	A $p \times p$ zero matrix;
$\mathbf{0}_{p\times q}$	A $p \times q$ zero matrix;
I_p	A $p \times p$ identity matrix;
$\mathbf{1}_p$	A $p \times p$ matrix having elements; 1
Span	Range space of the column vectors;
$\mathbf{e}$	A column vector whose all elements are one;
$\mathbf{e}_j$	A zero column vector in which the j^{th} element is of value 1;
[.]	Void matrix or vector
$[.]^T$	Transpose of a matrix or vector
diag(x)	If x is a vector then it returns a matrix whose diagonal elements are the elements of the vector otherwise if x is matrix it returns a vector containing only the diagonal elements of the matrix

6.2 Problem Formulation

Suppose there are N number of sensor nodes are spread through out the environment of interest. Let the sensors be spread at the time T_1 and the measurement process starts from T_1 after a regular time interval, say t_0, to time T_2. Thus, the sensors measure $(T_2 - T_1)/t_o + 1 = n$ number of measurements. The objective is to estimate the area where the abnormality has occurred at any time instant.

The following assumptions are used:

1. the sensors are spread in a near flat surface so that their z-coordinates are same;
2. the sensors are spread in a regular manner;
3. the sensors know their positions and also the positions of their neighboring sensor nodes (this is defined latter).

Representing the positions of the sensors by a matrix $\mathbf{S}$, we get

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_N \end{bmatrix} \tag{6.1}$$

where, $\mathbf{s}_j \in \Re^2$ is the position of the j^{th} sensor in x and y coordinates. Let the area of the environment of interest be $\mathbf{a}$. The sensors are spread in such a manner that all the sensors able to estimate condition of the entire environment. Thus, $\mathbf{S} \equiv \mathbf{a}$. The area $\mathbf{a}$ is represented as $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}^T$, where a_i the area corresponds to the sensor with position $\mathbf{s}_i$.

The objective is to estimate the area where the sensors measure the environmental value above the threshold level at every measured time instant. The area where the change has occurred is denoted by $\mathbf{a}^c$ and let it be a convex set. The total area vector $\mathbf{a}$ can be denoted as $\mathbf{a} = [\mathbf{a}^c \ (\mathbf{a} - \mathbf{a}^c)]$. Suppose the positions of these sensor nodes are $\mathbf{S}^c$. The problem is to estimate the MVE covering the area a^c using distributed strategy. Then the parameters of the MVE are to be sent to the control center.

Using the parameters of an Ellipsoid the problem can be reformulated as given below. An ellipsoid is defined by

$$E = \{\mathbf{c} + \mathbf{Q}\mathbf{u}| \ \|\mathbf{u}\|_2^2 \leq 1\} \tag{6.2}$$

where, $\mathbf{c} \in \Re^2$ and $\mathbf{Q} \in \Re^{2\times 2}$ are defined as the center and the spreading matrix of the MVE, respectively. Any point $\mathbf{x}_i$ in the ellipsoid must satisfy the following

$$\|\mathbf{A}\mathbf{x}_i + \mathbf{c}\|_2^2 \leq 1 \tag{6.3}$$

where, $\mathbf{A} = \mathbf{Q}^{1/2}$.

So the objective is to estimate the parameters of the MVE, $\mathbf{c}$ and $\mathbf{Q}$, covering the area $\mathbf{a}^c$. This can be formulated as an optimization problem

$$\min_{\mathbf{A},\mathbf{c}} \left\{ \text{vol}\left(\mathbf{E}\right) \middle| \mathbf{E} = \mathbf{c} + \mathbf{A}\mathbf{u}, \|\mathbf{u}\|_2^2 \leq 1, \|\mathbf{A}\mathbf{s}_i + \mathbf{c}\|_2^2 \leq 1, a_i \in \mathbf{a}^c \right\} \tag{6.4}$$

Since the objective (6.4) is based on the knowledge of positions of all the sensor nodes present in the WSNs, it can be called global objective. An algorithm is called fully distributed if a sensor only uses the data from its neighborhood nodes only. Hence, the global objective cannot be computed in fully distributed manner. Before formulating the problem as a distributed optimization problem, there is a requirement of finding neighborhood sensor nodes corresponds to a particular sensor.

6.2.1 Neighborhood Sensor Nodes

It is assumed that the sensor nodes are placed in regular manner. Suppose the distance between two nearest sensor nodes in X- or Y-directions is d. The neighborhood sensors of any sensor node, say j, is defined by

$$\mathbf{S}_{n_j} = \left\{ \forall s_j \middle| \sqrt{\left(s_i\left(1,1\right) - s_j\left(1,1\right)\right)^2 + \left(s_i\left(2,1\right) - s_j\left(2,1\right)\right)^2} \leq \sqrt{2}d \right\} \tag{6.5}$$

6.3 Proposed Distributed Strategy to Find the Incremental Path

The incremental strategy requires a predefined cyclic path connecting each sensor node. Thus, there is a need to find out an incremental path connecting each sensor node. In the existing literature, the problem of finding the incremental path connecting each sensor node is called as the Hamiltonian path problem [5]. This is a NP hard problem whose computational complexity increases with number of sensor nodes [5] and this algorithm can be applied to both regular and irregular problem. However, the main drawback of this algorithm is that, it requires all sensor node positions value

at one central processor which requires large communication overheads. It would be better if a distributed algorithm is proposed in which local sensor position values can be used to obtain the global incremental path which is done in this thesis. In this thesis a simple distributed technique is proposed to find the incremental path for regular network (that is all sensors are placed in a rectangular grid). This distributed estimation of the incremental path is attractive from a sensor network life time view point. In distributed implementation each node takes the decision after receiving the information from its neighbor sensors. The decision is in such a way that (i) the global incremental path can be found out and (ii) less amount of communication energy will be used. Two proposed strategies are given in Figure 6.3. Anyone of the strategy leads to the formation of the incremental path. The arrows given in the figure show the direction of transmission of data. The directions are $g1$, $g2$ and $g3$. If a sensor is in position (i, j) then the sensors corresponding to $g1$, $g2$ and $g3$ directions are sensors with position coordinates $(i+1, j)$, $(i, j+1)$ and $(i-1, j)$, respectively. The directions are chosen based on the following criteria:

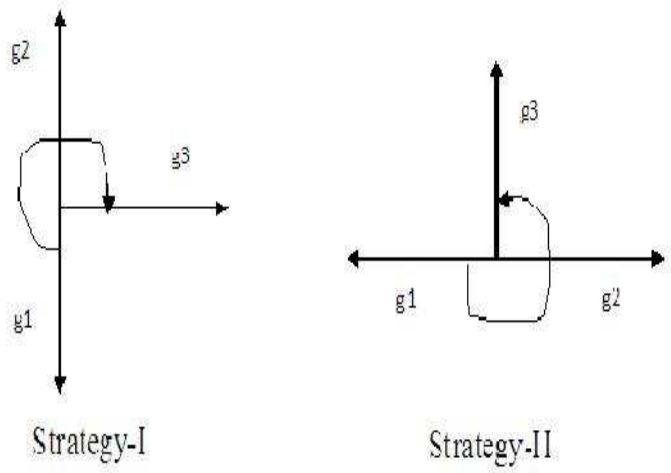

Figure 6.3: Decision at individual sensor node to obtain incremental strategy

1. first, the sensors choose the $g1$ direction sensor for transmission of data, if this

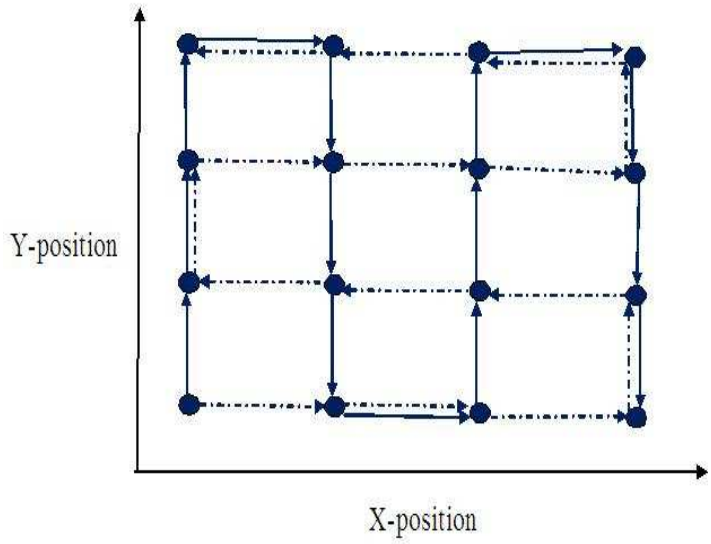

Figure 6.4: The incremental path obtained by the proposed strategy

is not possible then in $g2$ direction sensor, if not then finally to $g3$ direction sensor. Thus, they should choose a direction in the order $g1, g2$ and $g3$;

2. the sensors should not send the data to the sensor from which it has just received the data;

3. there is a predefined routing path from the N^{th} sensor node to the 1^{st} sensor node.

The proof of this strategy is given in Section D.1, in which it is shown that the above proposed strategy based on the local information leads to establish a global incremental path.

6.4 Proposed Techniques for Environment Monitoring

This section deals with the development of environment monitoring algorithm for simple case, i.e. for regular network. The estimation process starts after every measurement time. Based on this measurement value the MVE is to be formed in a

distributed manner. Assume that the region where the changes occur in the environment is a convex shape. In the proposed strategy after every measurement time a token passes from the first node to the N^{th} node. Along with the token the parameters of the MVE also pass. After the completion of one complete cycle the token value increases by one and it sends back to the first sensor node. Thus, the token value indicates the number of measurement times. When a sensor node receives the token from the neighbor sensor node at that time it does the operation. Along with the token a sensor node receives the MVE parameters. After reception of the MVE parameters the sensor node (i) updates the MVE parameters by using its own sensor position if the measurement value is above the threshold value or (ii) just route the MVE parameters along with the token to the next sensor node.

At the starting of the process the center and spreading matrix of the MVE are $\mathbf{c} = \mathbf{0}_{2\times 1}$ and $\mathbf{A} = \mathbf{0}_2$, respectively. Then these values pass along with the token through the sensor nodes. When the measurement value of any node is above the threshold value, then the sensor updates the parameters of the ellipsoid using its position value and passes the updated MVE parameters along with the token to the next node. A sensor chooses the next node by using the method given in Section 6.3.

In order to formulate the problem mathematically, consider the parameters of the estimated MVE for all the nodes present in the incremental path from the first sensor node to the $(k-1)^{th}$ sensor node as $\mathbf{c}_{k-1}$ and $\mathbf{Q}_{k-1}$ for center and spreading matrix respectively. These parameters are sent to the k^{th} sensor node along with the token defined by the incremental strategy. Based on the measured data, the k^{th} sensor node does the operation on the parameters $\mathbf{c_{k-1}}$ and $\mathbf{Q_{k-1}}$ using the position of it to obtain $\mathbf{c}_k$ and $\mathbf{Q}_k$. The relation between the previous ellipsoid and the new ellipsoid can be written as $\mathbf{Q}_{k-1} \subseteq \mathbf{Q}_k$. The updation can be formulated as a constraint convex optimization problem as [38]

$$\begin{aligned} &\mathbf{Q}_k, \mathbf{c}_k = \underset{\mathbf{Q}_k, \mathbf{c}_k}{\arg\min}\ (\log \mathrm{Det}\,(\mathbf{Q}_k)) \\ &s.t.\ \ \mathbf{Q}_{k-1} \subseteq \mathbf{Q}_k,\ \ (\mathbf{s}_k - \mathbf{c}_k)^T\, \mathbf{Q}_k^{-1}\, (\mathbf{s}_k - \mathbf{c}_k) \leq 1,\ \ t_k \geq T_h \end{aligned} \tag{6.6}$$

Here, positions of the k^{th} sensor node and the previous information about the

MVE are used as the constraint. If the measured temperature value by the sensor k is not greater than the threshold level T_h then it sends the information $\mathbf{c}_{k-1}$ and $\mathbf{Q}_{k-1}$ to the next sensor without any change.

If the previous MVE parameters are $\mathbf{c}_{k-1} = \mathbf{0}_{2\times 1}$ and $\mathbf{Q}_{k-1} = \mathbf{0}_2$, then the parameters of MVE at the k^{th} node are

$$\mathbf{c}_k = \mathbf{s}_k, \quad \mathbf{A}_k = k \times \mathrm{diag}\left(\mathrm{I}_{1\times 2}\right) \tag{6.7}$$

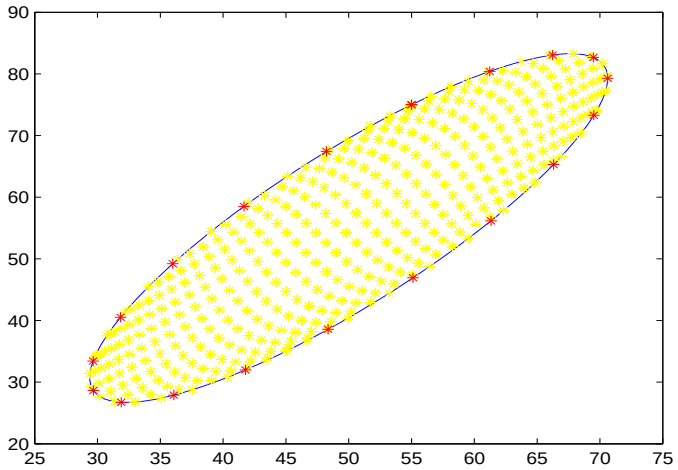

Figure 6.5: Ellipsoid and Ellipse

On the other hand if it is greater than T_h then it estimates the center and spreading matrix based on the convex optimization problem (6.6). The optimization problem (6.6) is meant to form an MVE enclosing the previous MVE and the k^{th} sensor node's position point. If the points only on the border area of the previous ellipsoid $\mathbf{Q}_{k-1}$ are considered then the entire ellipsoid $\mathbf{Q}_{k-1}$ becomes a subset of the new ellipsoid $\mathbf{Q}_k$. Therefore to decrease the computational complexity, the outer points of the MVE are considered in the optimization problem (6.6). As shown in Figure 6.5 the yellow area is an ellipsoid whose mathematical explanation is given in (6.8).

$$\boldsymbol{\chi}_{ellipsoid} = \left\{ \mathbf{p} \in \Re^2 \,\middle|\, (\mathbf{p}-\mathbf{c})^T \mathbf{Q}^{-1} (\mathbf{p}-\mathbf{c}) \leq 1 \right\} \tag{6.8}$$

The ellipse is the border line of the ellipsoid. The mathematical formula for the ellipse is

$$\boldsymbol{\chi}_{ellipse} = \left\{ \mathbf{p} \in \Re^2 \,\middle|\, (\mathbf{p}-\mathbf{c})^T \mathbf{Q}^{-1} (\mathbf{p}-\mathbf{c}) = 1 \right\} \tag{6.9}$$

In order to select the 20 points in the ellipse with equal interval, the following algorithm is employed

$\boldsymbol{\theta} \leftarrow 0 : \frac{2\pi}{20} : 2\pi$

$\boldsymbol{\chi}^{k-1} \leftarrow \varphi$

For $i = 1 : 20$

$x_i^{pos} = c_{k-1,1,1} + A_{k-1,1,1} \times \cos(\theta_i) + A_{k-1,1,2} \times \sin(\theta_i)$

$y_i^{pos} = c_{k-1,2,1} + A_{k-1,1,1} \times \cos(\theta_i) + A_{k-1,1,2} \times \sin(\theta_i)$

$$\boldsymbol{\chi}^{k-1} \leftarrow \boldsymbol{\chi}^{k-1} \cup \begin{bmatrix} x_i^{pos} \\ y_i^{pos} \end{bmatrix}$$

end

where, φ is a null set, $\mathbf{A}_{k-1} = \mathbf{Q}_{k-1}^{\frac{1}{2}}$ and the $A_{k-1,i,j}$ is the element present in i^{th} row and j^{th} column of the matrix $\mathbf{A}_{k-1}$.

With all these, the objective is to find the MVE covering the 20 points of the previous ellipsoid , i.e. $\boldsymbol{\chi}^{k-1}$ and the new sensor position. Thus, the problem is

$$\begin{aligned} \mathbf{Q}_k, \mathbf{c}_k = \arg \min_{\mathbf{Q}_k, \mathbf{c}_k} \; & \log \mathrm{Det}\,(\mathbf{Q}_k) \\ \text{s.t. } & (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{Q}_k^{-1} (\mathbf{x}_i - \mathbf{c}_k) \leq 1 \\ & \mathbf{x}_i \in \overline{\boldsymbol{\chi}}^k, t_k \geq T_h \end{aligned} \tag{6.10}$$

where, $\overline{\boldsymbol{\chi}}^k = \boldsymbol{\chi}^{k-1} \cup \mathbf{s}_k$.

6.4.1 Core Sets

The computational complexity of the problem (6.10) depends on 21 constraints present in it. Further to reduce the number of constraints a core-set of the 21 points is calculated and then the MVE is obtained for the core-set points. It is known that an ellipse is uniquely defined by the at most $2d$ number of points [39, 47], where d is

the dimension of each point. Since for our case $d = 2$, $2d = 4$ points are sufficient to uniquely construct the MVE. The core-set algorithm [39] is given by

$$
\begin{aligned}
&\text{For} \quad i = 1 : 2d \\
&\quad \text{If } n \leq 2d \\
&\quad\quad \boldsymbol{\chi}_0 \leftarrow \overline{\boldsymbol{\chi}}^k \\
&\quad \text{else} \\
&\quad \boldsymbol{\chi}_0 \leftarrow \emptyset \\
&\quad\quad \text{While} \\
&\quad\quad \Re^d / \boldsymbol{\Psi} \neq \emptyset \\
&\quad\quad \text{Pick an arbitrary direction } \mathbf{b}^i \in \Re^d \\
&\quad\quad \text{orthogonal complement of } \boldsymbol{\Psi} \\
&\quad\quad \alpha \leftarrow \underset{k=1,\cdots,n}{\arg\max} \left(\mathbf{b}^i\right)^T \overline{\boldsymbol{\chi}}_i^k, \quad \boldsymbol{\chi}_0 \leftarrow \left\{\boldsymbol{\chi}_0 \cup \left\{\overline{\boldsymbol{\chi}}_\alpha^k\right\}\right\} \\
&\quad\quad \beta \leftarrow \underset{k=1,\cdots,n}{\arg\min} \left(\mathbf{b}^i\right)^T \overline{\boldsymbol{\chi}}_i^k, \quad \boldsymbol{\chi}_0 \leftarrow \left\{\boldsymbol{\chi}_0 \cup \left\{\overline{\boldsymbol{\chi}}_\beta^k\right\}\right\} \\
&\quad\quad \boldsymbol{\Psi} \leftarrow \text{Span}\left(\boldsymbol{\Psi}, \left\{\overline{\boldsymbol{\chi}}_\beta^k - \overline{\boldsymbol{\chi}}_\alpha^k\right\}\right). \\
&\quad\quad \text{end} \\
&\quad \text{end} \\
&\text{end}
\end{aligned}
\tag{6.11}
$$

6.4.2 Development of the MVE Formation Algorithm Using The Khachiyan Algorithm

The design of all these algorithms, i.e. calculation of 20 points from the ellipsoid and further selection of 4 points, are only meant to design a computationally efficient algorithm which may be suitable for WSNs. This subsection deals with an algorithm to find out the MVE for the 4 points, i.e. $\boldsymbol{\chi}_0$. The MVE can be obtained by using the interior point method [38, 48, 49]. However, this requires a large number of computations and the process is slow. Further, to reduce the number of computational complexity the Khachiyan algorithm which is based on Lagrange multipliers [39] is introduced. The procedure of finding the Khachiyan algorithm is given in Section D.2.

The Khachiyan algorithm estimates the parameters of the MVE (covering the core-set data) by maximizing the Lagrange multipliers of the problem defined in (6.10). The optimization problem (6.10) is called as the primary problem and the optimization problem based on the Lagrange multiplier is called as the secondary problem. The secondary problem based on the Khachiyan algorithm corresponding to the primary problem (6.10) is given as

$$\begin{aligned} &\max_{\mathbf{v}\in\Re^n} \sum_{k=1}^{n} v_k \left(\mathbf{y}_k\right)^T \prod \left(\mathbf{u}^i\right)^{-1} \mathbf{y}_k \\ &\text{s.t.} \quad \mathbf{e}^T\mathbf{v} = 1, \;\; \mathbf{v} > \mathbf{0}_{n\times 1}. \end{aligned} \tag{6.12}$$

where, $\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}^T$, $\Pi\left(u^j\right) = \sum_{k=1}^{n} u_k^j \mathbf{y}_k \mathbf{y}_k^T$, and

$$\mathbf{y} = \left[\pm \begin{bmatrix} \boldsymbol{\chi}_{0,1} \\ 1 \end{bmatrix} \pm \begin{bmatrix} \boldsymbol{\chi}_{0,2} \\ 1 \end{bmatrix} \pm \begin{bmatrix} \boldsymbol{\chi}_{0,3} \\ 1 \end{bmatrix} \pm \begin{bmatrix} \boldsymbol{\chi}_{0,4} \\ 1 \end{bmatrix} \right]$$

This formulation is given in Section D.2.

In order to achieve the objective (6.12) the following recursive method can be used.

1. $i \leftarrow 0$, $u^0 \leftarrow \left(1/8\right)\mathbf{e}$
2. While not converged
3. **loop**
4. $j := \arg \max_{k=1,\ldots,8} \mathbf{y}_k^T \prod \left(\mathbf{u}^i\right)^{-1} \mathbf{y}_k$
5. $k^j := \max_{k=1,\ldots,8} \mathbf{y}_k^T \prod \left(\mathbf{u}^i\right)^{-1} \mathbf{y}_k$
6. Taking the convex combination of $\mathbf{u}^i$ and $\mathbf{e}_j$ we get $\mathbf{u}^{i+1} := \left(1-\beta^i\right)\mathbf{u}^i + \beta^i \mathbf{e}_j$
7. The parameter β is updated as
$$\begin{aligned} \beta^i &:= \arg \max_{\beta\in[0,1]} \log\det \prod \left(\left(1-\beta\right)\mathbf{u}^i + \beta \mathbf{e}_j\right) \\ &= \left[k^i - (d+1)\right] / \left[(d+1)\left(k^j - 1\right)\right] \end{aligned}$$
8. $i \leftarrow i+1$
9. **end loop**

6.4.3 Stepwise Description of the Distributed Algorithm

Operations at the node $k+1$

Step1: Input from the previous node $\rightarrow$ token, $\mathbf{c}_k, \mathbf{Q}_k$

Step2: Calculation of the 20 points using the algorithm in Section 6.4

Step3: Find the four number of points from the 21 points.

Step4: Then find out $\mathbf{c}_{k+1}$ and $\mathbf{Q}_{k+1}$ Step5: Transmit $\mathbf{c}_{k+1}$, $\mathbf{Q}_{k+1}$ and the token to the node $k+2$

6.5 Simulation Results for the Propose Strategy-I

For the simulation based experiments 121 number of sensor nodes are spread in an environment of 120×120 square area. The distance between two nearest sensor nodes is 8 in both X- and Y-directions. The left bottom sensor node is considered as the first sensor node. The red color sensors are supposed to have measurement values above the threshold value and the green color sensor nodes have measurement value below the threshold value. The strategy I is used for finding the incremental path. Figure 6.6 shows the estimated MVE at the third red sensor node. Figure 6.7 shows the MVE formed after the completion of the estimation process. From the simulation result it is found that the proposed distributed method is able to form the MVE covering the sensor nodes having the measurement value above the threshold level.

6.6 Proposed Strategy II

The previous strategy given in Section 6.4 is based merely on the measured data value to form the MVE. The assumption for the previous technique is that the region where an abnormal condition has occurred is a convex shape. Though this strategy requires less number of communication overheads, however, it suffers from a drawback. As illustrated in Figure 6.10, when the change areas of some groups remain far away from each other at that time the estimated MVE provides more error. In order to avoid this type of error, another novel strategy is proposed. This strategy helps to form

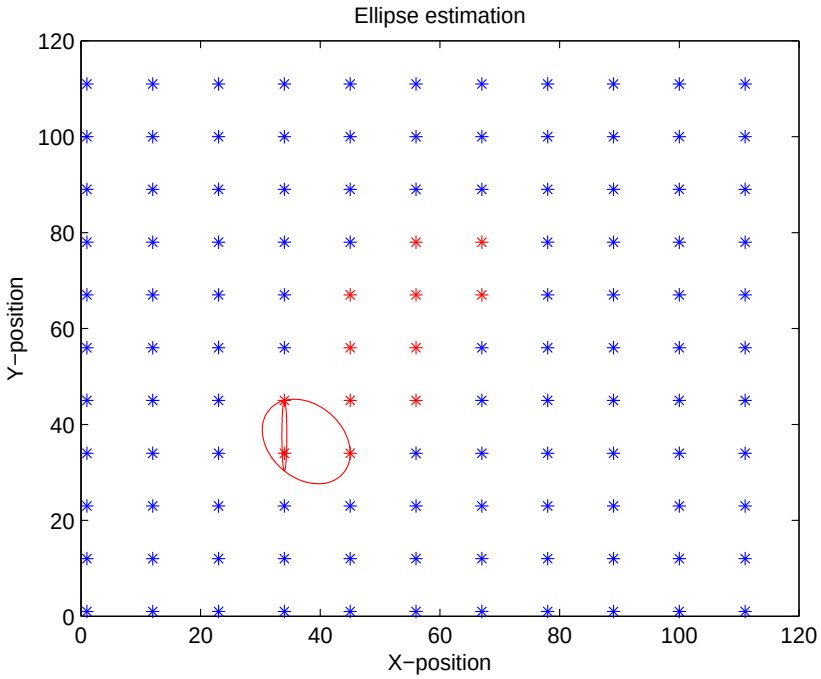

Figure 6.6: Ellipse formed by three sensors position

two or more MVEs by distributed manner. This strategy requires more information to be exchanged among the sensor nodes. The assumptions of this new strategy are: (i) each sensor knows the position of the neighbor sensor nodes,(ii) each sensor knows the shortest routing path from it to the FC and (iii) the sensors are dispersed in regular manner. This strategy is based upon three operation phases: (i) searching phase; (ii) formation phase; and (iii) breaking phase. Similar to the token passing method in the previous strategy given in Section 6.4 it also requires a method of token passing. These proposed methods start with the token at the first node. The token is embedded with an operation phase. In this strategy two another parameters are taken into account: (i) a counter value and (ii) a cluster number. In the starting of the WSN, the counter value is zero at every sensor nodes. When a sensor node receives the token, the counter value increases by one. Thus, the value of the counter at a specific node indicates the number of times the sensor node has received the token. The cluster value indicates to which cluster group the sensor node belongs.

This strategy requires, a sensor should take an action after receiving the following information from the neighbor sensor nodes:

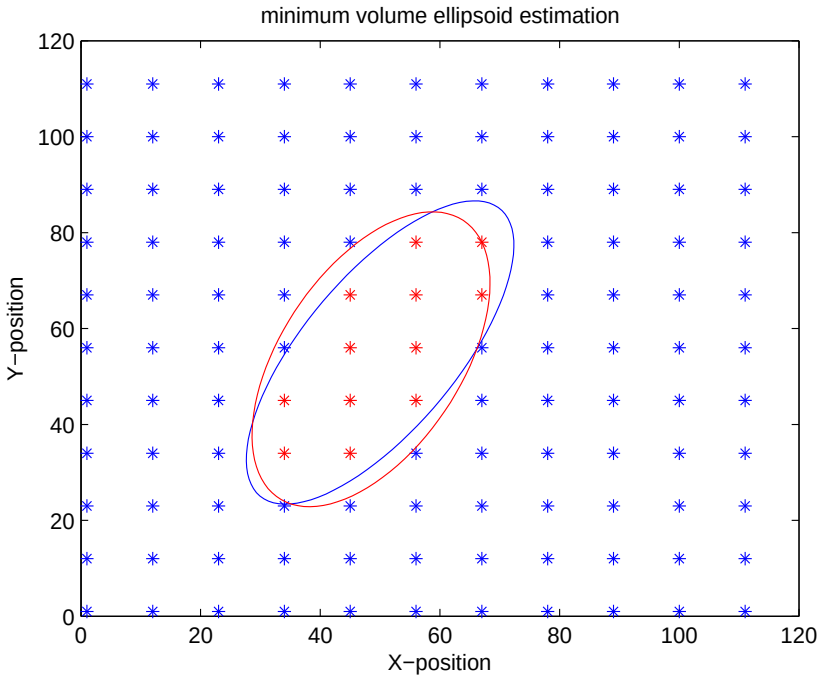

Figure 6.7: Ellipse formed by all the sensor

1. The information from the previous node are the token, the cluster number, the operation phase and prior information about the MVE;

2. The information from the other neighbor sensor nodes are the counter value and the measurement values.

The sensor node uses this information and perform some or all of the following operations:

1. The sensor updates the MVE;

2. The sensor chooses the next neighbor sensor node to which the token is to be passed;

3. The operation phase change may occur.

Corresponding to one particular set of received information a sensor node does some of the particular operations. Thus, one set of operations is associated with one particular set of received information. This strategy depends upon the relative position of the information. In order to consider the relative position of the information,

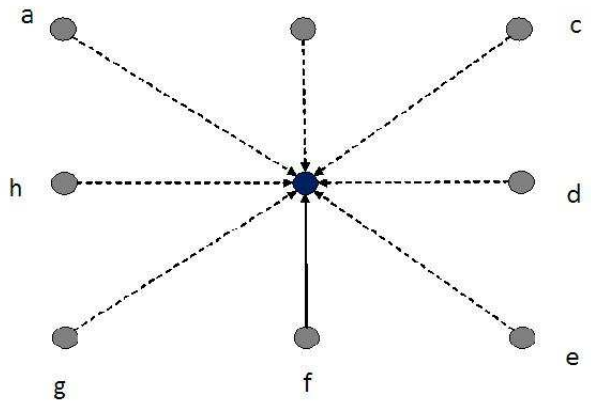

Figure 6.8: A sensor node with its neighbor sensors

the neighbor sensor nodes are divided of any sensor node k as a,b,c,d,e,f,g and h as shown in the Figure 6.8. For node k, the sensor node a is neighbor present in north-west(NW) direction. Similarly b,c,d,e,f,g and h sensor nodes are respectively present in north(NO), north-east(NE), east(EA), south-east(SE), south(SO), south-west(SW) and west(WE) directions of k^{th} sensor node. Thus, the information matrix can be denoted by

$$\mathbf{D}_k^{inf} = \begin{bmatrix} C_a & C_b & \cdots & C_h \\ T_a & T_b & \cdots & T_h \end{bmatrix} \tag{6.13}$$

The rows one and two correspond to the counter and temperature values of the sensor nodes respectively. Based on the counter values and the measured temperature values whether the temperature is above the threshold or less, 2^{16} number of different information matrices can be formed. The entire information or a portion of information in the matrix can be used to design a strategy. In order to achieve almost same performance that is available if entire information matrix is used as well as to decrease the communication overheads the strategy I and cluster number information are incorporated in the new strategy. By this way a portion of the information matrix is required, thus decreases the communication overheads. The cluster number parameter will help to assign each sensor node into one cluster group. Suppose

the token is carrying the searching phase and the temperature value is less than the threshold value then just transmit the information without any change. Otherwise if the measured value is above the threshold value then the sensor node updates the MVE corresponding to the cluster group and then sends the cluster group value to all the neighborhood sensors. If there is no cluster group value in the sensor node then it is for a new MVE and the cluster value is increased and thus a new cluster is made. The proposed operations are given for different phases. Firstly the operations for the token of searching phase are given for different information.

1. Suppose the measurement at the sensor node is above the threshold level.
   (a) A new MVE is formed and a cluster value is assigned and the counter value is increased by one.
   (b) Then the cluster value is sent to all the neighbor sensor nodes whose measurement is above the threshold level and the counter value is less than the counter value of it.
   (c) If the measurement value of no neighbor sensor node is greater than the threshold level then split the searching phase token into a searching phase token and another to a breaking phase token.
   (d) Attach the MVE parameters to the breaking phase token and transmit it through a multihop communication to the FC.
   (e) Pass the searching phase token using the strategy given in Section 6.3
2. If the measurement at the sensor node is below the threshold level, then pass the token by the same strategy given in Section 6.3.

Now the operations at a sensor node which receives a token carrying the formation phase are given.

1. Suppose the temperature value at the sensor node is above the threshold level
   (a) Then update the MVE and increase its counter value by one.

(b) Transmit the cluster value to all the neighbor sensor nodes whose temperature value is above the threshold level and the counter value is less than its counter value.

(c) If there is no neighbor sensor node whose temperature value is greater than the threshold value and the counter value is less than it then split the formation phase token into the searching phase and the breaking phase tokens.

(d) The searching phase token will be transmitted in the direction according to the incremental strategy and the breaking phase token is passed towards the direction of the FC.

2. Suppose the temperature value at the sensor node is below the threshold level. Then just pass the token without any change in the parameters of MVEs.

Suppose a node receives a token carrying a breaking phase then just route the MVE parameter to the next sensor which can forward it through multihop communication to the FC. By this way the number of MVEs increases as a trade off to the decrease of the error value. The lifetime of the network is calculated in Section 6.8. In this case the computation energy and communication energy loss increases and thus the lifetime decreases.

6.6.1 Simulation Results for Proposed Strategy-II

For the simulation based assessment of the proposed strategy 400 sensor nodes are considered in the environment of interest. These sensor nodes are placed in a regular manner. The distance between two nearest sensor node in one row or column is 10 units. As illustrated in Figure 6.9, suppose the red sensors measure the event. Thus, the objective is to cover these points by ellipsoids. As shown in Figure 6.9, the first proposed strategy leads to form the ellipsoid which covers the entire area of interest, however, it provides a large amount of error. As shown in Figure 6.10 the second proposed method leads to form 4 MVEs corresponding the 4 distinct areas so that the error value decreases substantially.

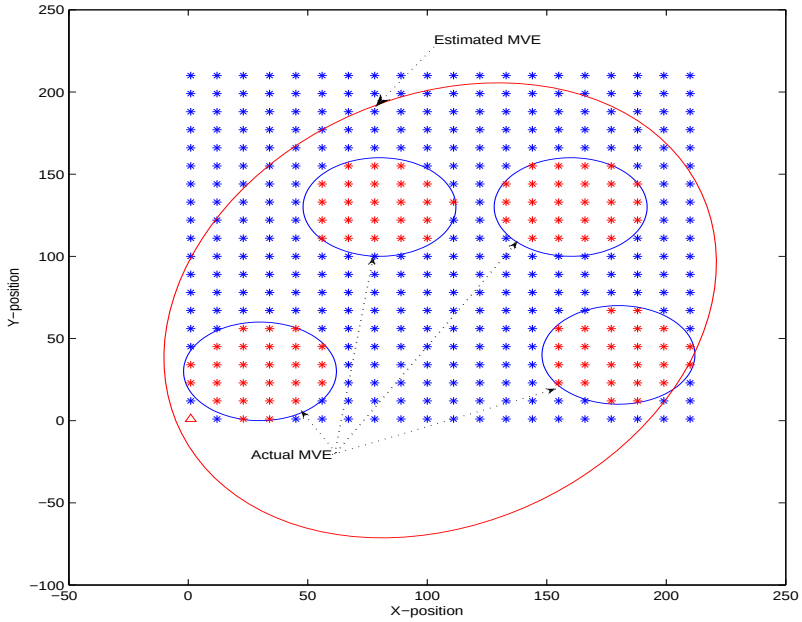

Figure 6.9: Estimated MVE obtained by using the proposed strategy-I

6.7 Incremental Distributed Strategy in Presence of Additive White Gaussian Noise

The previous sections dealt with the scenario when there is no environmental noise. However, in practice noise is present and hence the algorithm should work in the presence of environmental noise in the measured data. The proposed method for this type of scenario is the detection followed by the formation of the MVE. Therefore, the detection is to be carried out before estimation of the MVE to ascertain that the signal is present in the measured data. For the detection of the signal the likelihood ratio test(LRT) based on the Neyman-Pearson [50, 51] lemma is employed. For the design of the Neyman-Pearson lemma for the spatio-temporal data the following assumptions are considered.

1. The sensors are present in regular manner throughout the environment.

2. The noise present in the environment is the AWGN.

3. Each sensor knows the position of its neighborhood sensor nodes.

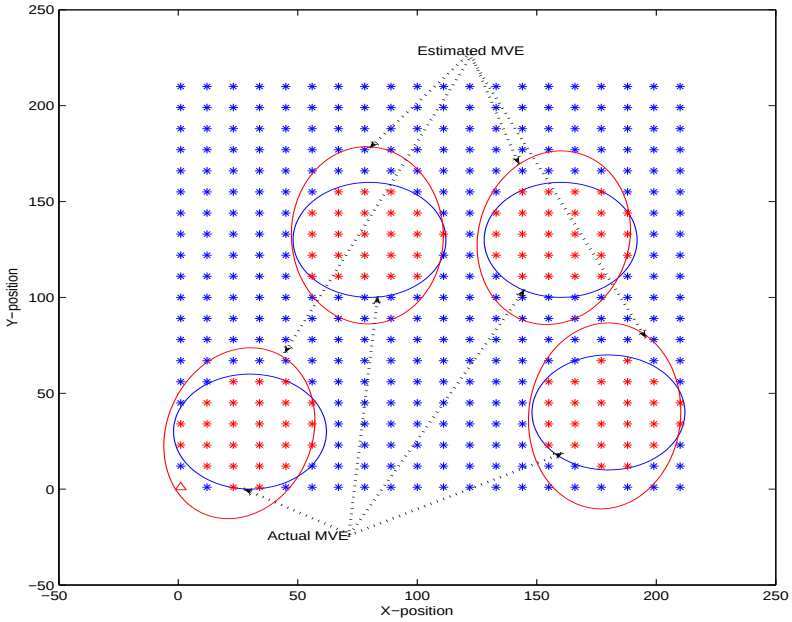

Figure 6.10: Estimated MVE obtained by using the proposed strategy-II

4. Noise present in different sensor nodes are independent of each other.

The investigation considers that the incremental path is established by the use of strategy I(the strategy I is explained in Section 6.4). Since the noise at different sensor nodes is independent of each other, the detection criteria at different nodes are independent of each other. The Neyman-Pearson lemma relies on two hypotheses: (a) null hypothesis, (b) alternative hypothesis. Null hypothesis corresponds to the situation when the measured data contains only noise (this occurs when there is no abnormality i.e. event and only the noise is present in the sensor node) where as alternative hypothesis corresponds to the situation when the measured data contains both the noise and abnormal signal. Since the required signal which is the cause of abnormality and the noise in the environmental change with time, it is appropriate to model both the signal and noise as random variables. The following assumptions are introduced for designing the detection criteria.

1. The noise present in the environment is a Gaussian random variable with mean zero and independent of the signal.

2. The signal is a Gaussian random variable with a finite mean value.

3. The signal at different sensor nodes are independent of each other.

4. The variance values of the noise and the signal are known.

5. The region where an abnormality has occurred is convex in shape.

Based on these assumptions the analysis at every sensor node is similar. The null and alternative hypotheses for any sensor node k are represented as

$$\begin{aligned} H_{0,k} &= \mathbf{w}_k \in \Re^n \\ H_{1,k} &= \mathbf{s}_k \in \Re^n + \mathbf{w}_k \in \Re^n \end{aligned} \tag{6.14}$$

where, $H_{0,k}$ and $H_{1,k}$ are the null and the alternative hypotheses.

The probability density function(PDF) for n number of measurements at any node k under the null hypothesis is

$$p\left(\mathbf{x}_k \,|H_{0,k}\right) = \frac{1}{\left(2\pi\sigma_{w_k}^2\right)^{n/2}} e^{\frac{\sum_{i=1}^{n}\left(x_{k,i}\right)^2}{2\sigma_{w_k}^2}} \tag{6.15}$$

Let the PDF of the signal at the node k be

$$p\left(\mathbf{s}_k \,|H_{0,k}\right) = \frac{1}{\left(2\pi\sigma_{\mathbf{s}_k}^2\right)^{n/2}} e^{\frac{\sum_{i=1}^{n}\left(s_{k,i}-\mu_{\mathbf{s},k}\right)^2}{2\sigma_{\mathbf{s}_k}^2}} \tag{6.16}$$

Hence, the PDF of measured data under $H_{1,k}$ is

$$p\left(\mathbf{x_k} \,|H_{1,k}\right) = \frac{1}{\left(2\pi\left(\sigma_{\mathbf{w}_k}^2 + \sigma_{\mathbf{s}_k}^2\right)\right)^{n/2}} e^{\frac{\sum_{i=1}^{n}\left(x_{k,i}-\mu_{\mathbf{s}_k}\right)^2}{2\left(\sigma_{\mathbf{w}_k}^2+\sigma_{\mathbf{s}_k}^2\right)}} \tag{6.17}$$

According to the Neyman-Pearson lemma if the LRT is greater than some threshold value then there is probability of signal. The constant term depends upon a fixed value of false alarm (FA) rate. The Neyman-Pearson lemma for node k having a constant FA rate α_k is represented as

$$l_k = \frac{p\left(\mathbf{x}_k \left| H_{1,k}\right.\right)}{p\left(\mathbf{x}_k \left| H_{0,k}\right.\right)} > \gamma_k \tag{6.18}$$

$$\int_{(\mathbf{x_k}, l_k > \gamma_k)} p\left(\mathbf{x}_k, H_{0,k}\right) = \alpha_k \tag{6.19}$$

Here the number of sensor nodes in the environment is N. The state of the environment can be denoted by an indicator function by taking the comparison with the threshold value at the sensor node. This is given by

$$\mathbf{s}_{env} = \begin{bmatrix} 1_{T_1 - T_h} & 1_{T_2 - T_h} & \cdots & 1_{T_N - T_h} \end{bmatrix} \tag{6.20}$$

Hence the number of possible states will be 2^N. According to one state there is a corresponding region. Thus $\mathbf{s}_i = A_i, i = 1 \cdots 2^N$. Therefore, there will be a 2^N number of regions can be detected. Suppose the variance is same in every node. Since, all the data present in the different sensor nodes are independent of each other then the local detection criteria do not affect the other sensor node. Accordingly the global detection criteria is defined. Collecting all the local LRT into a vector we get

$$\mathbf{l}_G = \begin{bmatrix} p\left(\mathbf{x}_1 / H_{1,1}\right) / p\left(\mathbf{x}_1 / H_{0,1}\right) \\ p\left(\mathbf{x}_2 / H_{1,2}\right) / p\left(\mathbf{x}_2 / H_{0,2}\right) \\ \vdots \\ p\left(\mathbf{x}_N / H_{1,N}\right) / p\left(\mathbf{x}_N / H_{0,N}\right) \end{bmatrix} \tag{6.21}$$

Similarly collecting all the threshold values into a vector, we get

$$\boldsymbol{\gamma}_G = \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_N \end{bmatrix}^T \tag{6.22}$$

The global Neyman-Person lemma can be defined as

$$\begin{aligned} \mathbf{l}_G &> \boldsymbol{\gamma}_G \\ \int\limits_{(\mathbf{x},\mathbf{l}_G>\boldsymbol{\gamma}_G)} & p\left(\mathbf{x}, H_0\right) = \alpha \end{aligned} \tag{6.23}$$

Using the assumptions, it can be shown that

$$\alpha = \sum_{i=1}^{N} \alpha_i \tag{6.24}$$

Thus the investigation based on LRT for one node is sufficient. Let us investigate on the LRT for any node say k. Using the PDF of the measured data at node k under the null and alternative hypotheses, the LRT for node k is

$$\frac{\frac{1}{(2\pi)^{n/2}\left(\sigma^2_{\mathbf{w}_k}+\sigma^2_{\mathbf{s}_k}\right)^{n/2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{n}\frac{\left(x_{k,i}-\mu_{\mathbf{s}_k}\right)^2}{\left(\sigma^2_{\mathbf{w}_k}+\sigma^2_{\mathbf{s}_k}\right)}\right]}{\frac{1}{(2\pi)^{n/2}\left(\sigma^2_{\mathbf{w}_k}\right)^{n/2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{n}\frac{x^2_{k,i}}{\sigma^2_{\mathbf{s}_k}}\right]} > \gamma_k \tag{6.25}$$

Simplification of (6.25) leads to

$$\left(\frac{\sigma^2_{\mathbf{w}_k}}{\sigma^2_{\mathbf{w}_k}+\sigma^2_{\mathbf{s}_k}}\right)^{n/2} \exp\left[-\frac{1}{2}\left(\sum_{i=1}^{n}\frac{\left(x_{k,i}-\mu_{s_k}\right)^2}{\sigma^2_{\mathbf{w}_k}+\sigma^2_{\mathbf{s}_k}} - \sum_{i=1}^{n}\frac{x^2_{k,i}}{\sigma^2_{\mathbf{w}_k}}\right)\right] > \gamma_k \tag{6.26}$$

Taking the logarithm on both sides of (6.26) and further manipulating, we get

$$\sum_{i=1}^{n}\left(\frac{x^2_{k,i}}{\sigma^2_{\mathbf{w}_k}} - \frac{\left(x_{k,i}-\mu_{\mathbf{s}_k}\right)^2}{\sigma^2_{\mathbf{w}_k}+\sigma^2_{\mathbf{s}_k}}\right) > 2\ln\left(\frac{\sigma^2_{\mathbf{w}_k}+\sigma^2_{\mathbf{s}_k}}{\sigma^2_{\mathbf{w}_k}}\right)^{n/2}\gamma_k \tag{6.27}$$

Simplification of (6.27) leads to

$$\sum_{i=1}^{n}\left(x^2_{k,i} - \left(1+\frac{\sigma^2_{\mathbf{s}_k}}{\sigma^2_{\mathbf{w}_k}}\right)^{-1}\left(x^2_{k,i}+\mu^2_{\mathbf{s}_k}-2x_{k,i}\mu_{\mathbf{s}_k}\right)\right) > 2\sigma^2_{\mathbf{w}_k}\ln\left(\frac{\sigma^2_{\mathbf{w}_k}+\sigma^2_{\mathbf{s}_k}}{\sigma^2_{\mathbf{w}_k}}\right)^{n/2}\gamma_k \tag{6.28}$$

In order to further manipulate (6.28), the matrix inverse lemma is introduced. The matrix inverse lemma is given as

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B\left(DA^{-1}B + C^{-1}\right)DA^{-1} \tag{6.29}$$

Taking $A = 1$, $B = 1$, $C = \sigma_s^2/\sigma_n^2$ and $D = 1$ in (6.28) and applying the matrix inverse lemma we get

$$\begin{aligned}&\sum_{i=1}^{n}\left(2x_{k,i}\mu_{\mathbf{s}_k} + \left(1+\frac{\sigma^2_{\mathbf{w}_k}}{\sigma^2_{\mathbf{s}_k}}\right)^{-1}\left(x^2_{k,i} - 2x_{k,i}\mu_{\mathbf{s}_k}\right)\right) > \\ &2\sigma^2_{\mathbf{w}_k}\ln\left(\frac{\sigma^2_{\mathbf{w}_k}+\sigma^2_{\mathbf{s}_k}}{\sigma^2_{\mathbf{w}_k}}\right)^{n/2}\gamma + n\mu^2_{\mathbf{s}_k} - n\left(1+\frac{\sigma^2_{\mathbf{w}_k}}{\sigma^2_{\mathbf{s}_k}}\right)^{-1}\mu^2_{\mathbf{s}_k}\end{aligned} \tag{6.30}$$

Further simplification of (6.30) leads to

$$\begin{aligned}&\sum_{i=1}^{n}\left(2\left(1+\frac{\sigma^2_{\mathbf{w}_k}}{\sigma^2_{\mathbf{s}_k}}\right)x_{k,i}\mu_{\mathbf{s}_k} + \left(1+\frac{\sigma^2_{\mathbf{w}_k}}{\sigma^2_{\mathbf{s}_k}}\right)^{-1}x^2_{k,i}\right) > \\ &2\sigma^2_{\mathbf{w}_k}\ln\left(\frac{\sigma^2_{\mathbf{w}_k}+\sigma^2_{\mathbf{s}_k}}{\sigma^2_{\mathbf{w}_k}}\right)^{n/2}\gamma_k + n\mu^2_{\mathbf{s}_k} - n\left(1+\frac{\sigma^2_{\mathbf{w}_k}}{\sigma^2_{\mathbf{s}_k}}\right)^{-1}\mu^2_{\mathbf{s}_k}\end{aligned} \tag{6.31}$$

In the proposed method the measured values are used to find the left portion of (6.31) and it is compared with the right portion of it, if it satisfies (6.31) then the MVE will be formed. Thus, in the proposed method when a node receives the token from the previous node it uses the measured data to find the condition (6.31). Based on the condition it updates the prior MVE or it transmits as it is.

6.7.1 Simulation Results for Noisy Measured Data

This subsection deals with the simulation based experiment to verify the proposed method. In this case the same number of sensor nodes and the same type of environment are taken into consideration. The noise at the individual node is AWGN with a zero mean and unity variance. The signal variable is white Gaussian having mean of a random value between $(10, 13)$. Total 10 number of data is used for detection whether the signal is present or not. The simulation results are given in Figures 6.11 and 6.12. Figure 6.11 corresponds to the MVE estimation based upon the detection

and the strategy-I, where as Figure 6.12 corresponds to the MVE estimation based upon the detection and the strategy-II.

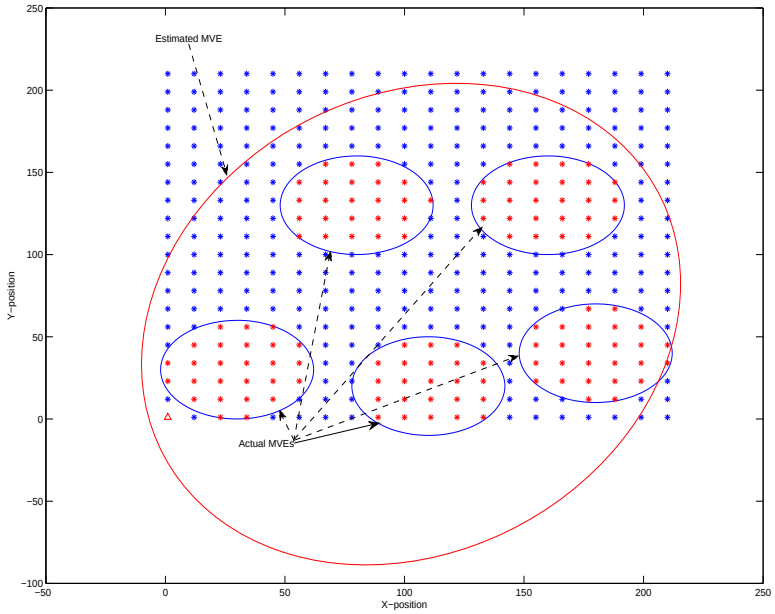

Figure 6.11: Estimation of MVE in presence of AWGN using strategy-I

6.8 Sensor Network Lifetime

The sensor network life time is defined by the duration of the deployment of the sensor nodes to the time when it would not able to solve the objective for which it was deployed. This time can be calculated by the state space approach. Suppose the energy of the i^{th} sensor node is ϵ_i and transmitting energy loss in one transmission is e_i. Let the initial energy and transmission energy be same for all the sensor nodes.

The lifetime of a sensor network depends upon three factors [52–54].

1. The architecture of the network. In this chapter the flat and fixed architecture is considered. In case of classical technique and proposed technique the routing path is to the FC and the next neighbor node defined by the incremental path given in 6.3, respectively.

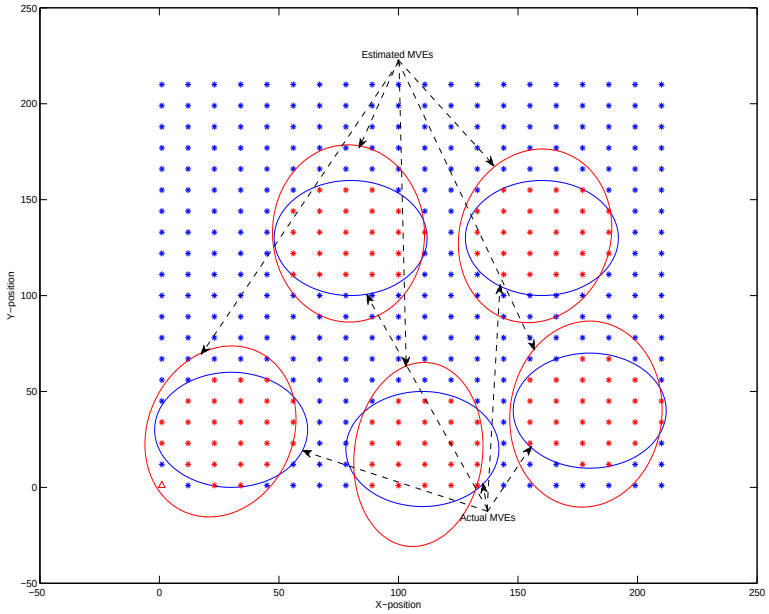

Figure 6.12: Estimation of MVE in presence of AWGN using strategy-II

2. Data collection initiation of the sensor network. There are several types of data collection modes. In this chapter the sensors are measuring the environment data at regular time interval.

3. The energy consumption model of the sensor network. There are two types energy consumption in WSNs. One is continuous energy consumption and another one is reporting energy consumption.

Let the continuous energy consumption be same for every sensor nodes and it be ϵ_c. Suppose the sensing energy is very less and hence can be neglected. Let minimum e amount of energy be required at the sensor node for transmission of one bit data in one time. Thus, if the energy of one sensor node decreases less than e then it will not be able to transmit its data. Under such situation the sensor node can be considered as dead. Similarly, the WSNs can be consider as dead when a considerable number of sensor nodes will become dead. For the mathematical formulation of the lifetime of the WSN let the energy state of the WSNs is defined by a vector

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \cdots & \varepsilon_N \end{bmatrix} \tag{6.32}$$

As time passes the energy state of the WSN changes. Energy of each sensor node decreases in every time interval by reporting energy and constantly by continuous energy. Let the reporting interval be λ. Thus, after n reporting times the energy of any sensor node say k decreases to

$$\varepsilon_i^k = \varepsilon - \lambda n \varepsilon_c - nbe \tag{6.33}$$

Therefore, the energy of any sensor node is one of the values belonging to the set.

$$\epsilon_{p,i} = \begin{bmatrix} \epsilon_i & \epsilon_i - 1 \times e_i - \lambda 1 \varepsilon_c & \epsilon_i - 2 \times e_i - \lambda 2 \varepsilon_c & \cdots & \epsilon_i - (L-1) \times e_i - \lambda (L-1) \varepsilon_c & e_r \end{bmatrix} \tag{6.34}$$

where, e_r is the residual energy of the sensor node. Thus, the entire energy space is a discrete space point of N dimensions with points corresponding to the energy state of the network. The state moves from one state to the other state with time. There are some areas in which if the state would fall then the network is of no use. But some residual energy remains unused. This is known as the dead condition of a WSN. A reward is given for every change of the state until it will fall in the dead state region. The total reward times of the interval time is the life time of the WSNs. The generalize description about the lifetime of the WSNs can be made simpler for a particular strategy.

6.8.1 For Classical Technique

In classical case every sensor transmit their measured data and positions to the FC through multihop communication. In this strategy the sensors nearer to the FC generally route all the data of the WSNs to the FC. Thus, there is a maximum probability that these sensors will lose their energy very quickly and will become dead within less time after the deployment. Further, when these sensors become dead

then the rest of the sensors cannot able to transmit their data to the FC. Now the entire WSNs is dead. By this way the life time of WSNs based on classical technique can be defined as the time duration from the deployment of the sensor nodes to the time when all sensor nodes near to the FC would become dead. This can be explained by the notion of energy state-space as follows. For classical technique the energy state vector can be rearranged as

$$\boldsymbol{\varepsilon}_{cl} = \begin{bmatrix} \boldsymbol{\varepsilon}_{fn} & \boldsymbol{\varepsilon}_{nfn} \end{bmatrix} \tag{6.35}$$

where, $\boldsymbol{\varepsilon}_{fn}$ and $\boldsymbol{\varepsilon}_{nfn}$ are the energy state vector corresponding to the sensor nodes near to the FC and the rest of the sensor nodes. Further, consider the energy value

$$e_{fne}^{total} = \left\{ \sum \mathbf{e}_i \, | i \in \varepsilon_{fn} \right\} \text{and } \; e_{nfne}^{total} = \left\{ \sum \mathbf{e}_i \, | i \in (\varepsilon_{fn}) \right\} \tag{6.36}$$

Using the parameters in (6.36) a two dimensional state vector is defined by

$$\mathbf{e} = \begin{bmatrix} e_{fne}^{total} & e_{nfne}^{total} \end{bmatrix} \tag{6.37}$$

The lifetime of the WSN based on the classical strategy can be explained in the following way. Thus, the lifetime of the WSN can be described by the time duration from the deployment of the sensor network to the time when the energy value e_{fne}^{total} would become e. It can be formulated mathematically as

$$L_{time}^{classical} = \frac{e^{nf}}{e_c^{nf} + \lambda e_r^{nf}} \tag{6.38}$$

Suppose for transmission of the data each sensor uses b number of quantization bits. In one time frame the number of data required for transmission to the FC is $3Nb$. Thus, the total energy loss per unit time is continuous energy loss which is e_c^{nf} and the reporting energy loss is $3Nbe$.

Hence, the lifetime of the sensor network based on the classical method is

$$L_{time}^{classical} = \frac{e^{nf}}{e_c^{nf} + 3Nbe} \tag{6.39}$$

Suppose the number of neighbor sensor nodes of the FC is N_c. Equation (6.39) can be changed to

$$L_{time}^{classical} = \frac{N_c e_i}{N_c e_i^{nf} + 3Nbe} \tag{6.40}$$

Equation (6.40) can be written as

$$L_{time}^{classical} = \frac{e_i}{e_i^{nf} + 3(N/N_c)be} \tag{6.41}$$

6.8.2 For Proposed Technique I

In the incremental strategy based on the proposed technique-I each sensor sends the parameter of the MVE to the next sensor node defined by the incremental path. Thus if one sensor node loses all of its energy then an incremental path cannot be established and the WSNs fails to achieve the objective. Thus, the lifetime of the WSN based on the incremental strategy is defined by the time duration of the deployment of the sensor node to the time when one sensor node loses all of its energy. The sensor node before the last sensor node transmits a large amount of data. Therefore, there is much probability of this sensor to become dead first. In this strategy-I each sensor only transmits the updated MVE parameters to next node defined by the incremental path. Thus, each sensor transmits six parameters to the next sensor node. Consider b number of quantization bits are used to transmit the data. Hence, in one interval each sensor sends $6b$ number of bits. Hence, the lifetime of the sensor network can be calculated as

$$L_T = \frac{e_i}{e_c^i + \lambda e_r^i} \tag{6.42}$$

Equation (6.42) can be written as

$$L_T = \frac{e_i}{\varepsilon_c^i + 6be/L} \tag{6.43}$$

6.8.3 For Proposed Technique II

This subsection deals with the calculation of the lifetime of the WSNs equipped with the proposed technique II given in Section 6.6. For this the worst case scenario is considered. This scenario can also be considered as the lower bound of the lifetime of the sensor network. The worst case scenario corresponds to the situation when a sensor node transmits the cluster number, counter value and the measurement value to every neighbor sensor nodes. Further, analyzing the Figure 6.9 and the strategy-II, it can be found that a sensor node needs to transmit the cluster value to the sensor nodes b, c, d, e. A sensor node needs to transmit the measurement and the counter values to the sensors a, f, g, h. In addition consider that the same sensor node transmits all the MVEs parameters to the FC. Let there be p number of MVEs are formed. Thus there are $12+6\times p$ number of communications during each iteration in the worst case for one sensor node. Thus, following (6.42) it can be found that the lifetime of the WSNs is

$$L_T = \frac{\varepsilon_i}{\varepsilon_c^i + \lambda\varepsilon_r^i} \tag{6.44}$$

Equation (D.2) can be written as

$$L_T = \frac{\varepsilon_i}{\varepsilon_c^i + \lambda 12 + 6\times p} \tag{6.45}$$

6.9 Robust Technique

In case real scenario before transmission to the next node the MVE parameters needs to be quantized and encode. Due to the quantization process the MVE parameters subject to quantization error that leads to the shifting of the mean value as well as a

change in the spreading factor and the orientation of the MVE. As shown in Figure 6.13 the sensor nodes $2, 3, 5, 9$ have detected the event. Thus, the objective is to form the MVE covering the positions of these four points. However, due to quantization of the X- and Y-coordinates, the received position values of these four points are shifted by the quantization error value from the original positions. The original positions are marked by blue color and the quantized positions are indicated by the gray color in the Figure 6.13. Therefore, the general method which are explained in Section 6.3 leads to form the MVE covering the gray color points which is given by black color ellipsoid let it be χ_Q. However, it does not cover the actual positions. The MVE formed by covering the actual positions is given by the red color and is denoted by χ_R. Since $\chi_R \not\subset \chi_Q$ so some event places are missed. If due to quantization error some event places, for the problem like monitoring of volcanic eruption or fire in the forest, some portion of the environment remains unnoticed then that small error may create more problems. Hence, there is need to design robust algorithm which will be helpful for such special scenarios. On the other hand, the objective should be to form the MVE χ_Q such that it should satisfy $\chi_R \subset \chi_Q$. In order to achieve this, the possible region of the actual position of the sensor node corresponding to the quantized position value needs to be found out. This can be obtained by using the step size value of the quantization process. If the position of the sensor node after quantization is (x, y) and the step size is μ then the possibility of the actual sensor node position may be any of the point in a square whose corner points are $(x - \mu, y - \mu)$, $(x + \mu, y - \mu)$, $(x - \mu, y + \mu)$ and $(x + \mu, y + \mu)$. However, for the case of the MVE formation from one priori MVE and a point the following procedure can be followed.

Suppose the previous sensor node sends the parameters $\mathbf{Q}^d$ and $\mathbf{c}^d$ which is quantized data of the priori MVE parameters $\mathbf{Q}$ and $\mathbf{c}$. However, it is not possible to find the possible region of the MVE from the quantized data of the mean and spreading matrix because the quantization of the spreading matrix may lead to another spreading matrix from which it is very difficult to find the original spreading matrix. In order to avoid this problem a different approach is proposed. In this case instead of transmitting the direct quantized MVE values the quantized value of the coreset data

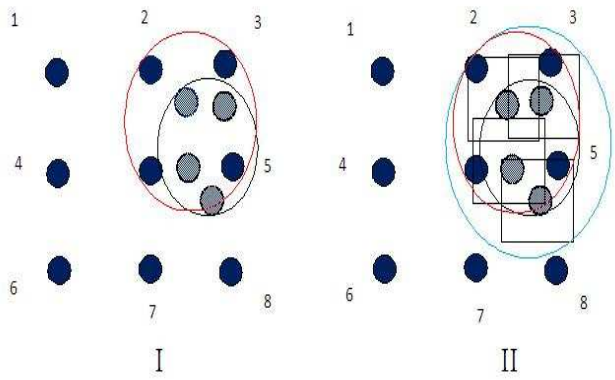

Figure 6.13: Robust method for MVE calculation

corresponding to the priori MVE is transmitted. After receiving the quantized points the sensor nodes calculate the possible region for every quantized data. Then the core set is calculated and the MVE is formed.

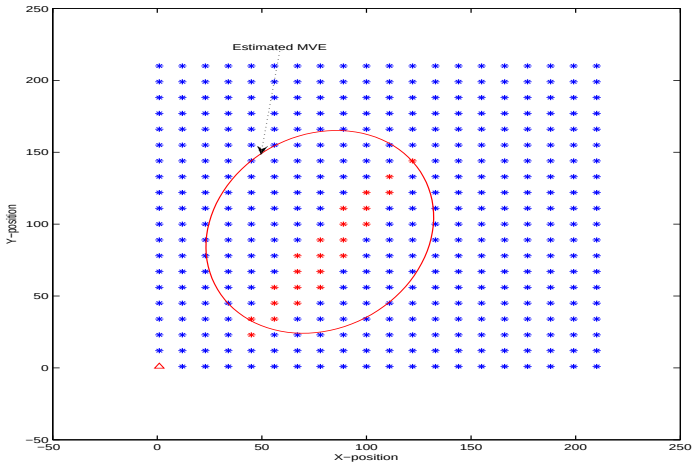

Figure 6.14: MVE Formation based on Proposed strategy-I

6.9.1 Simulation Results for Robust Method

This subsection deals with the simulation based experiment to verify the proposed method of robust MVE estimation. In this case the same number of sensor nodes and the same type of environment are taken into consideration. As illustrated in

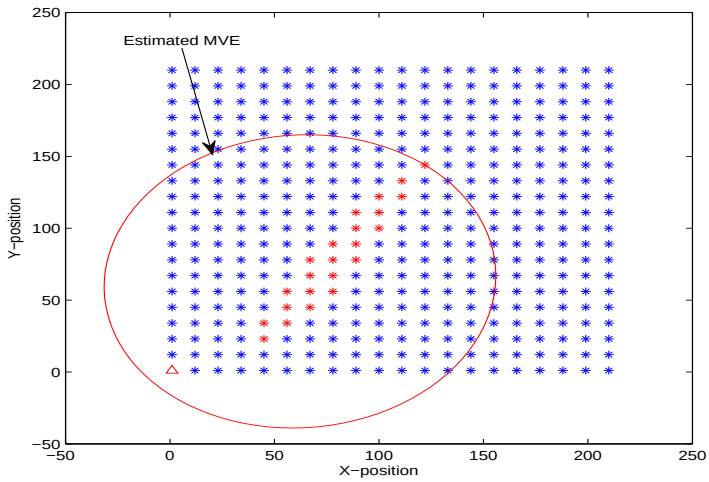

Figure 6.15: MVE formation based on robust method

Figure 6.14 the general method based MVE estimation does not cover one red point. However, as shown in Figure 6.15 the proposed robust method covers the every point with large error values.

6.10 Conclusion

The technique shown is very good for the environment where the environment change sets is very near to the convex sets. It increases the lifetime of the sensor network with large scale. A novel distributed strategy is proposed to find the incremental path. The lifetime of both the proposed methods are given and it is found that the lifetime of the proposed method is more than the case of classical method. A robust method is proposed to find the MVE in case of quantization errors. When the spreading of the sensor nodes is not regular then the problem and formulation becomes very interesting. Designing new energy efficient protocol to increase the lifetime of the sensor network can be regarded as the future extension of the present case.

Chapter 7

Conclusions and Future Work

This chapter deals with the overall contributions of the thesis. For further investigation on the same or related topics future research directions are also given.

7.1 Conclusions

In this thesis, the robust distributed adaptive estimation algorithms are proposed. These estimation algorithms are based on the Wilcoxon norm and the generalized rank norm to handle impulsive noise in the desired and/or input data. Some of these algorithms provide better estimation performance at the expense of poor convergence speed. A number of novel robust algorithms are proposed to handle outliers in the desired data. These algorithms exhibit faster convergence speed than previous existing algorithm at the expense of poor estimation accuracy. In order to handle outliers in both input and desired data a novel distributed algorithm based on GR norm is proposed. In addition a novel QR decomposition based approach is proposed which provides faster convergence speed and also exhibits better estimation accuracy. However, this algorithm requires large computational and communication complexities. Due to the development of efficient VLSI architecture and low power VLSI, this distributed algorithm can easily be implemented in WSNs. A novel environment monitoring algorithm based on distributed incremental strategy is proposed. This can be suitable for a large number of applications like precision agriculture, monitoring the fire in the forest, study of an active volcano etc. Different computational efficient methods are introduced in this environment monitoring proposed method which can be suitable for WSNs. This can be treated as a very good application of distributed strategy.

In Chapter 2, some novel robust algorithms are proposed based on the notion of the Wilcoxon norm. In addition to this a novel modified Wilcoxon norm is also proposed. Sign sign and sign regressor Wilcoxon norm and Sign sign, sign regressor modified Wilcoxon norm are also proposed. Exhaustive simulation studies have been carried out of all these algorithms and it is found that these algorithms are robust against outliers in the desired data. These algorithms provide faster convergence speed compared to the existing techniques with less performance. These algorithms

offer lesser computational complexity.

We have proposed the distributed implementation of the generalized rank norm to handle outliers in both the input and the desired data in Chapter 3. An indicator function based approach is used to analyze the norm. A novel median based approach is proposed which requires less number of computations. In addition to incremental generalized rank norm, incremental high breakdown estimator is also proposed. Similar to Chapter 2 sign regressor generalized rank estimator and sign regressor highbreak down estimator are also proposed. The new scheme is suitable for distributed implementation in WSNs. From the simulation studies it is found that the proposed techniques are robust against outliers in the desired and input data. One of the variant of this algorithm is proposed which provide faster convergence speed thus suitable for fast changing environment.

A novel QR based robust distributed strategy is proposed in Chapter 4 which exhibits faster convergence speed and provide better performance than all the other previous algorithms. These algorithms are meant to handle outliers in the desired data as well as in both the input and the desired data. A low communication complexity based QR decomposition algorithm is also proposed which is more suitable for WSNs. Exhaustive simulation studies for weak to strong outliers in both the input and desired data are done and it is found that these algorithms are robust up to 50% outliers both in input and desired data. The main drawback of this algorithm is that it requires more number of computations.

In Chapter 5 a R-APA algorithm is proposed which is suitable for colored data and it acts as a compromise between the general gradient based algorithm which is given in Chapter 2 and the QR-IMWN algorithm which is given in Chapter 4 in terms of computations and convergence. Thus this is suitable for real environment application where most of the spatial data are correlated to each other.

The novel environment monitoring method given in Chapter 6 is of great importance, since it is suitable for many real world applications. For this application we have used minimum volume ellipsoid method. A novel incremental strategy is also proposed. Some modifications are done in the algorithm for low computation. A

robust method is also proposed in order to handle quantization noise. The lifetime of the proposed method is also calculated and it is found that the proposed method requires less energy of the sensor network.

7.2 Suggestions for Future Work

Distributed signal processing for WSNs still in the infant stage. This work can be extended in different directions taking different aspects such as networking, the lifetime of the WSNs, coding, cooperative communication etc. Some of the future research directions are given here.

- All the methods given in the thesis are based only on the incremental distributed strategy of the WSNs, which requires a predefined cyclic path connecting each sensor network present in the environment. This is not possible for WSNs having large number of sensor nodes. In this scenario robust signal processing based on diffusion [11, 20, 22] and adaptive diffusion strategies are more suitable.

- In this thesis the model is a linear and lumped parameter whose value is same for every node. This work can be extended to an environment having distributed parameter system [55] with nonlinearity which frequently occur in real environment scenario where the data is spatiotemporal.

- This thesis dealt only with the signal processing part of the WSNs. This work can also be extended to study how the different layers of the wireless networking can be incorporated in the distributed estimation problem for a particular application [56].

- This thesis only covers the design of the robust algorithms to handle outliers in the measured data. Still there is more work to be done on the convergence analysis of the algorithm, finding the break down point and influence function of the algorithm. All these characteristics of an adaptive algorithm are very important to study. For these the sophisticated mathematical analysis based on the theory of rank tests [16, 42] can be considered.

- The algorithms reported in the thesis based on the assumption that the connective links between the sensor nodes are perfect and noise free. Thus this can be extended to the environment where the link is corrupted by the noise and data is non stationary [57].

Appendix A

A.1 Proof of the Function (2.33) As a Pseudo Norm

If a function satisfies three properties then it is called as a pseudo-norm [16]. These are

$$\begin{aligned} &\|\mathbf{v}_1\|_{mwn} \geq 0 \\ &\|a\mathbf{v}\|_{mwn} = a\,\|\mathbf{v}\|_{mwn} \\ &\|\mathbf{v}_1 + \mathbf{v}_2\|_{mwn} \leq \|\mathbf{v}_1\|_{mwn} + \|\mathbf{v}_2\|_{mwn} \end{aligned} \tag{A.1}$$

Proof of the property 1: The function is given by

$$\|\mathbf{v}\|_{mwn} = \sum_{i=1}^{l-1} \sum_{j=i+1}^{l} \sqrt{12} \left(\frac{\mathrm{R}\,(v_i - v_j)}{0.5 \times l(l-1)+1} - 0.5 \right) (v_i - v_j) \tag{A.2}$$

Taking the difference value as one element and substituting it in (A.2), we obtain

$$\|\mathbf{v}\|_{mwn} = \sum_{k=1}^{0.5\times l(l-1)} \sqrt{12} \left(\frac{\mathrm{R}\,(t_k)}{0.5 \times l(l-1)+1} - 0.5 \right) t_k \tag{A.3}$$

Without loss of generality, we can assume $t_1 \leq t_2 \leq \cdots \leq t_{0.5\times l(l-1)}$. Then (A.3) is modified to

$$\|\mathbf{v}\|_{mwn} = \sum_{k=1}^{0.5\times l(l-1)} \sqrt{12} \left(\frac{k}{0.5 \times l(l-1)+1} - 0.5 \right) t_k \tag{A.4}$$

Manipulating (A.4) using the median value, we get

$$\begin{aligned}\|\mathbf{v}\|_{mwn} = &\sum_{k=1}^{0.5\times l(l-1)} \sqrt{12}\left(\tfrac{k}{0.5\times l(l-1)+1} - 0.5\right)(t_k - \mathrm{Med}\,(t_k)) \\ &+ \sum_{k=1}^{0.5\times l(l-1)} \sqrt{12}\left(\tfrac{k}{0.5\times l(l-1)+1} - 0.5\right)\mathrm{Med}\,(t_k)\end{aligned} \tag{A.5}$$

The second term can be easily shown as 0. The first term contains half of the term negative and half of the term positive. From (A.5) it is found that the coefficients of the negative term is negative and the coefficients of the positive term is positive. Hence the function (A.5) is always positive. When all the elements are equal at that time the norm is of value zero. Thus the first property is proved.

Proof of the property 2:

In order to proof the second property, (A.2) is rewritten with multiplication with any positive constant, say a, as

$$\begin{aligned}\|a\mathbf{v}\|_{mwn} &= \sum_{i=1}^{l-1}\sum_{j=i+1}^{l} \sqrt{12}\left(\tfrac{R(av_i - av_j)}{0.5\times l(l-1)+1} - 0.5\right)(av_i - av_j) \\ &\overset{(i)}{=} a\sum_{i=1}^{l-1}\sum_{j=i+1}^{l} \sqrt{12}\left(\tfrac{R(av_i - av_j)}{0.5\times l(l-1)+1} - 0.5\right)(v_i - v_j) \\ &\overset{(ii)}{=} a\sum_{i=1}^{l-1}\sum_{j=i+1}^{l} \sqrt{12}\left(\tfrac{R(av_i - av_j)}{0.5\times l(l-1)+1} - 0.5\right)(v_i - v_j) \\ &= a\,\|\mathbf{v}\|_{mwn}\end{aligned} \tag{A.6}$$

By taking the constant value a common, we get (i) equal. Since $\mathrm{R}(a(v_i - v_j))$ among the elements $a\mathbf{v}$ is equal to the $\mathrm{R}(v_i - v_j)$ among the elements of $\mathbf{v}$. Therefore we get the (ii) equal. Which is nothing but $a\,\|\mathbf{v}\|_{mwn}$. Thus the second property is proved.

Proof of the property 3:

In order to prove the third property, the notion of Cauchy-Schwartz inequality is used. Therefore first objective is to show that the the function (2.34) is a convex function. A function is convex if it satisfies the following property

$$f\left(\lambda\mathbf{x} + (1-\lambda)\,\mathbf{y}\right) \leq \lambda f\left(\mathbf{x}\right) + (1-\lambda)\,f\left(\mathbf{y}\right) \tag{A.7}$$

To prove the function (2.34) as a convex function the following steps are used, in the first step the elements present in the vector $\overline{\mathbf{v}}$ given as

$$\overline{\mathbf{v}} = \text{vec}\left([v_i - v_j], \forall i < j, v_i \in \mathbf{v}, v_j \in \mathbf{v}\right) \tag{A.8}$$

are shown as a convex function. The elements are

$$v_i - v_j = y_i - y_j + (\mathbf{x}_j^T - \mathbf{x}_i^T)\mathbf{w} \tag{A.9}$$

Since it is an affine function of the parameter $\mathbf{w}$. Then by [38], (A.9) can be shown as a convex function. In the second step it is shown that the function which takes the maximum value of some number of convex functions is also a convex function. For this $\overline{v}^{max}$ is defined by

$$\overline{v}^{max} = \text{Max}\begin{bmatrix} \overline{v}_1 & \overline{v}_2 & \cdots & \overline{v}_{0.5\times l(l-1)} \end{bmatrix} \tag{A.10}$$

where $\overline{v}_i$ is the element of the random vector $\overline{\mathbf{v}}$ (which is also a convex function), then by [19,p.73] (A.10) can also be proved as a convex function. In third step a permutation function is used. Consider the the following permutation function.

$$\text{Perm}(v^1 \times v^2) = \text{vec}\left(\left[v_i^1 \times v_j^2\right], \forall i, \forall j, v_i^1 \in \mathbf{v}^1, v_j^2 \in \mathbf{v}^2\right) \tag{A.11}$$

where $\mathbf{v}^1 = \begin{bmatrix} v_1^1 & v_2^1 & \cdots & v_l^1 \end{bmatrix}$ and $\mathbf{v}^2 = \begin{bmatrix} v_1^2 & v_2^2 & \cdots & v_l^2 \end{bmatrix}$

The permutation vector contains the elements which is obtained by summation of every one by one multiplication of the elements in the vector $\mathbf{v}^1$ with the elements in $\mathbf{v}^2$. The maximum value element corresponds when the elements are in order [42]. That means $\text{R}(v_i^1) = \text{R}(v_i^2)$, where $\text{R}(v_i^1)$ and $\text{R}(v_j^2)$ are the rank order of v_i^1 in $\mathbf{v}^1$ and rank order of v_j^2 in $\mathbf{v}^2$ respectively. Therefore the function

$$y = \text{Max}\left(\text{Perm}\left(\mathbf{v}^1\mathbf{v}^2\right)\right) \tag{A.12}$$

is a convex function. Thus we have

$$\left\|\lambda \mathbf{v}^1 + (1-\lambda)\,\mathbf{v}^2\right\|_{mwn} \leq \lambda \left\|\mathbf{v}^1\right\|_{mwn} + (1-\lambda)\left\|\mathbf{v}^2\right\|_{mwn} \tag{A.13}$$

Taking the value of λ as $1/2$ and then applying the second property to the first term of (A.13), we get

$$\left\|\mathbf{v}^1 + \mathbf{v}^2\right\|_{mwn} \leq \left\|\mathbf{v}^1\right\|_{mwn} + \left\|\mathbf{v}^2\right\|_{mwn} \tag{A.14}$$

Hence the third property is proved. Thus it is proved that the function (2.33) is a pseudo norm.

Appendix B

B.1 Proof of Equation (3.4)

The Wilcoxon norm of a vector $\mathbf{v}$, as given in Section-II, is given by

$$\|\mathbf{v}\|_w = \sum_{i=1}^{L} \left(\sqrt{12} \left(\frac{\mathrm{R}(v_i)}{L+1} - 0.5 \right) v_i \right) \tag{B.1}$$

Here, $\mathrm{R}(v_i)$ is rank order of the error v_i among all elements in the vector. This implies that if all the elements in the vector are arranged in increasing order then rank order of an element is the position of the element. Without sorting operation this rank order value can also be calculated using indicator function, which is given by

$$\mathrm{I}_x = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{B.2}$$

Therefore by simple mathematical manipulation, $\mathrm{R}(v_i)$ can be described by using the indicator function given in (B.3) as

$$\mathrm{R}(v_i) = \sum_{j=1}^{L} \mathrm{I}_{(v_i - v_j)} \tag{B.3}$$

Incorporating the above indicator function based rank order value in (B.2), we get

$$\|\mathbf{v}\|_w = \sum_{i=1}^{L} \left(\sqrt{12} \left(\frac{\sum_{j=1}^{L} \mathrm{I}_{(v_i - v_j)}}{L+1} - \frac{1}{2} \right) v_i \right) \tag{B.4}$$

Modifying (B.4), we find

$$\|\mathbf{v}\|_w = \sum_{i=1}^{L} \left(\sqrt{12} \left(\frac{2\sum_{j=1}^{L} \mathrm{I}_{(v_i - v_j)} - L - 1}{2\,(L+1)} \right) v_i \right) \tag{B.5}$$

Taking the common term $\sqrt{12}/\,(L+1)$ to outside, (B.5) is changed to

$$\|\mathbf{v}\|_w = \left(\frac{\sqrt{12}}{2\,(L+1)} \right) \sum_{i=1}^{L} \left(2\sum_{j=1}^{L} \mathrm{I}_{(v_i - v_j)} - L - 1 \right) v_i \tag{B.6}$$

Since $\mathrm{I}_o = 1$ in (B.3), then $\mathrm{I}_{v_i - v_i}$ is 1 and $\mathrm{I}_{v_i - v_j} + \mathrm{I}_{v_j - v_i}$ is also equal to 1 except $v_j = v_i$, in which it is 2. Using these values in (B.6) at $l+1$ term, we get

$$\|\mathbf{v}\|_w = \left(\frac{\sqrt{12}}{2\,(L+1)} \right) \sum_{i=1}^{L} \left(2\sum_{j=1}^{L} \mathrm{I}_{(v_i - v_j)} - \left(\sum_{j=1}^{L} \left(\mathrm{I}_{(v_i - v_j)} + \mathrm{I}_{(v_j - v_i)} \right) \right) - 2\mathrm{I}_{(v_i - v_i)} \right) v_i \tag{B.7}$$

Simplification of (B.7) leads to

$$\|\mathbf{v}\|_w = \left(\frac{\sqrt{12}}{2\,(L+1)} \right) \sum_{i=1}^{L} \left(\sum_{j=1}^{L} \mathrm{I}_{(v_i - v_j)} - \sum_{j=1}^{L} \mathrm{I}_{(v_j - v_i)} - 2\mathrm{I}_{(v_i - v_j)} \right) v_i \tag{B.8}$$

Finally (B.8) is solved as

$$\|\mathbf{v}\|_w = \left(\frac{\sqrt{12}}{2\,(L+1)} \right) \sum_{i=1}^{L} \sum_{j=1, j \neq i}^{L} \left(\mathrm{I}_{(v_i - v_j)} - \mathrm{I}_{(v_j - v_i)} \right) e_i \tag{B.9}$$

Splitting the two summation terms ,$\sum_{i=1}^{L}\sum_{j=1,j\neq i}^{L}\left(\mathrm{I}_{(v_i-v_j)}+\mathrm{I}_{(v_j-v_i)}\right)e_i$, in (B.9) to two two summation terms, one for $i<j$ and the other for $i>j$, we obtain

$$\|\mathbf{v}\|_w=\left(\frac{\sqrt{12}}{2\left(L+1\right)}\right)\left(\sum_{i=2}^{L}\sum_{j=1}^{i-1}\left(\mathrm{I}_{(v_i-v_j)}-\mathrm{I}_{(v_j-v_i)}\right)v_i+\sum_{j=2}^{L}\sum_{i=1}^{j-1}\left(\mathrm{I}_{(v_i-v_j)}-\mathrm{I}_{(v_j-v_i)}\right)v_i\right) \tag{B.10}$$

Then, interchanging the j^{th} and i^{th} index in the second summation term that is after the plus sign in (B.10), we get

$$\|\mathbf{v}\|_w=\left(\frac{\sqrt{12}}{2\left(L+1\right)}\right)\left(\sum_{i=2}^{L}\sum_{j=1}^{i-1}\left(\mathrm{I}_{(v_i-v_j)}-\mathrm{I}_{(v_j-v_i)}\right)e_i+\sum_{i=2}^{L}\sum_{j=1}^{j-1}\left(\mathrm{I}_{(v_j-v_i)}-\mathrm{I}_{(v_i-v_j)}\right)v_j\right) \tag{B.11}$$

Further merging the two summations in (B.11) into one summation, we obtain

$$\|\mathbf{v}\|_w=\left(\frac{\sqrt{12}}{2\left(L+1\right)}\right)\left(\sum_{i=2}^{L}\sum_{j=1}^{i-1}\left(\left(\mathrm{I}_{(v_i-v_j)}-\mathrm{I}_{(v_j-v_i)}\right)\left(v_i-v_j\right)\right)\right) \tag{B.12}$$

Further simplification of (B.12) leads to

$$\|\mathbf{v}\|_w=\left(\frac{\sqrt{12}}{2\left(L+1\right)}\right)\left(\sum_{i=2}^{L}\sum_{j=1}^{i-1}\left(\mathrm{I}_{(v_i-v_j)}\left(v_i-v_j\right)-\mathrm{I}_{(v_j-v_i)}\left(v_i-v_j\right)\right)\right) \tag{B.13}$$

Equation (B.13) can be written as

$$\|\mathbf{v}\|_w=\left(\frac{\sqrt{12}}{2\left(L+1\right)}\right)\left(\sum_{i=2}^{L}\sum_{j=1}^{i-1}\left|v_i-v_j\right|\right) \tag{B.14}$$

which is equivalent to

$$\|\mathbf{v}\|_w=\left(\frac{\sqrt{12}}{2\left(L+1\right)}\right)\left(\sum_{j<i}\left|v_i-v_j\right|\right) \tag{B.15}$$

which is the desired form given in (3.4).

B.2 Generalize R Norm Using Indicator Function

As given in (3.5), the GR norm is

$$\|\mathbf{v}\|_{GR} = \sum_{i<j} w_{ij} \left| v_i - v_j \right| \tag{B.16}$$

Manipulation of the absolute error difference term in (B.16) using indicator function leads to

$$\|\mathbf{v}\|_{GR} = \sum_{i<j} \left(w_{ij} \mathrm{I}_{(v_i - v_j)} \left(v_i - v_j \right) + w_{ij} \mathrm{I}_{(v_j - v_i)} \left(v_j - v_i \right) \right) \tag{B.17}$$

Now replacing one summation term with condition $i < j$ with two summation terms, we obtain

$$\|\mathbf{v}\|_{GR} = \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \left(w_{ij} \mathrm{I}_{(v_i - v_j)} \left(v_i - v_j \right) + w_{ij} \mathrm{I}_{(v_j - v_i)} \left(v_j - v_i \right) \right) \tag{B.18}$$

Further solving (B.18), we get

$$\|\mathbf{v}\|_{GR} = \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \left(w_{ij} \mathrm{I}_{(v_i - v_j)} - w_{ij} \mathrm{I}_{(v_j - v_i)} \right) v_i + \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \left(w_{ij} \mathrm{I}_{(v_j - v_i)} - w_{ij} I_{(v_i - v_j)} \right) v_j \tag{B.19}$$

Interchanging the i^{th} and j^{th} index in the second summation term, which is after the plus sign, we get

$$\|\mathbf{v}\|_{GR} = \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \left(w_{ij} \mathrm{I}_{(v_i - v_j)} - w_{ij} \mathrm{I}_{(v_j - v_i)} \right) v_i + \sum_{j=1}^{L-1} \sum_{i=j+1}^{L} \left(w_{ij} \mathrm{I}_{(v_i - v_j)} - w_{ij} \mathrm{I}_{(v_j - v_i)} \right) v_i \tag{B.20}$$

Then merging the two summation terms, one before the plus sign and the other one after the plus sign, into one summation term, we get

$$\|\mathbf{v}\|_{GR} = \sum_{i=1}^{L} \sum_{j=1, j\neq i}^{L} \left(w_{ij} \mathrm{I}_{(v_i - v_j)} - w_{ij} \mathrm{I}_{(v_j - v_i)} \right) v_i \tag{B.21}$$

Equation (B.21) represents the desired form.

Appendix C

C.1 Calculation of Block Householder Transformation Matrix

Suppose $\mathbf{z} \in \Re^q$ is a vector and

$$\mathbf{T}(n) = \mathbf{I}_{q\times q} - \frac{2\mathbf{z}\mathbf{z}^T}{\|\mathbf{z}\|^2} \tag{C.1}$$

When the matrix $\mathbf{T}$ is multiplied with another vector $\mathbf{x}$, the vector $\mathbf{Tx}$ is reflected on to the vector $\mathbf{z}$. If the values $\mathbf{z}$ is suitably chosen to be $\mathbf{x} + \|\mathbf{x}\|_2 \mathbf{e}_1$ where $\mathbf{e}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \in \Re^n$ then $\mathbf{x}$ reflected onto $\mathbf{e}_1$ as $\mathbf{TX} = \pm \|\mathbf{X}\|_2 \mathbf{e}_1$. In this case all the energy of the vector $\mathbf{X}$ is reflected onto the unit vector $\mathbf{e}_1$. By changing the position of 1 in $\mathbf{e}_1$ different columns of the matrix $\mathbf{X}_{k,n}^T$ will be made zero.

For the case of block incremental RLS is reproduced as

$$\overline{\mathbf{R}}_k(n) = \begin{bmatrix} \mathbf{R}_{k-1}(n) \\ \mathbf{X}_{k,n}^T \end{bmatrix} \tag{C.2}$$

Let the block Householder transformation matrix for (C.2) be $\mathbf{H}_k(n)$. So that

$$\mathbf{H}_k(n)\,\overline{\mathbf{R}}_k(n) = \begin{bmatrix} \mathbf{R}_{k-1}(n) \\ 0 \end{bmatrix} \tag{C.3}$$

The above operation in (C.3) is split into p different operations as

$$\mathbf{H}_k^{(i)}(n)\begin{bmatrix} & \mathbf{R}_k^{(i-1)}(n) & \\ 0 \;\cdots & \mathbf{x}_{k,i}^{(i-1)} & \cdots\; \mathbf{x}_{k,p}^{(i-1)} \end{bmatrix} = \begin{bmatrix} & \mathbf{R}_k^{(i)}(n) & \\ 0 \;\cdots & \mathbf{x}_{k,i+1}^{(i)} & \cdots\; \mathbf{x}_{k,p}^{(i)} \end{bmatrix} \tag{C.4}$$

where $\mathbf{x}_{k,j}^{(0)} = \mathbf{x}_{k,j}, \mathbf{R}_{k-1}^{(0)}(n-1) = \mathbf{R}_{k-1}(n)$. and the resultant Householder transformation matrix is $\mathbf{H}_k(n) = \mathbf{H}_k^{(p)}(n)\,\mathbf{H}_k^{(p-1)}(n)\cdots\mathbf{H}_k^{(1)}(n)$. In this case each $\mathbf{H}_k^{(i)}(n)$ represents the Householder transformation which reflects the i^{th} column of updated $\mathbf{X}_{k,n}^T$ i.e. $\mathbf{x}_{n,i}^{(i-1)}$ to zero. The structure of $\mathbf{H}_k^{(i)}(n)$ is then given by

$$\mathbf{H}_k^{(i)}(n) = \begin{bmatrix} \mathbf{H}_{k,11}^{(i)}(n) & \mathbf{H}_{k,12}^{(i)}(n) \\ \mathbf{H}_{k,21}^{(i)}(n) & \mathbf{H}_{k,22}^{(i)}(n) \end{bmatrix} \tag{C.5}$$

where $\mathbf{H}_{k,11}^{(i)}(n) \in \Re^{p\times p}$ represents an identity matrix except for the i^{th} diagonal entry. In this case $\mathbf{H}_{k,11}^{(i)}(n) \in \Re^{p\times k}$ is a zero matrix except for the i^{th} row, $H_{k,12}^{(i)}(n) = \left(H_{k,21}^{(i)}(n)\right)^T$ and $H_{k,22}^{(i)}(n) = \mathbf{I} - \left(2x_{n,i}^{(i-1)}\left(x_{n,i}^{(i-1)}\right)^T \Big/ \sigma^2_{x_{n,i}^{(i-1)}}\right)$ is symmetric and $\sigma^2_{x_{n,i}^{(i-1)}} = \left\|x_{n,i}^{(i-1)}\right\|^2$. The reflection vector to calculate $\mathbf{H}_k^i(n)$ is given by

$$\left(\mathbf{z}^i\right)^T = \begin{bmatrix} \mathbf{0}_{1,i-1} & \mathbf{p}_i & \mathbf{0}_{1,p-i} & \left(\mathbf{x}_{k,i}^{i-1}\right)^T \end{bmatrix} \tag{C.6}$$

where $\mathbf{p} = \mathbf{R}_{k,i,i}^{i-1}(n) - \sqrt{\left(\mathbf{R}_{k,i,i}^{i-1}(n)\right)^2 + \left(\mathbf{x}_{k,i}^{i-1}\right)^T \mathbf{x}_{k,i}^{i-1}}$.

Here $\mathbf{H}_k^{(i)}(n)$ is obtained as

$$\mathbf{H}_k^{(i)}(n) = \mathbf{I}_{p+l} - 2\frac{\mathbf{z}^i\left(\mathbf{z}^i\right)^T}{\left(\mathbf{z}^i\right)^T\mathbf{z}^i} \tag{C.7}$$

The original BHT matrix $\mathbf{H}_k(n)$ is then obtained as

$\mathbf{H}_k(n) = \mathbf{H}_k^{(p)}(n)\,\mathbf{H}_k^{(p)}(n)\cdots\mathbf{H}_k^{(p)}(n)$.

C.2 Calculation of Block Householder Transformation Matrix for QR Based Robust Incremental Strategy

For QR based incremental proposed method the updated matrix at k^{th} node in time n is given by

$$\overline{\mathbf{R}}_k^{\mathrm{N}}(n) = \begin{bmatrix} \mathbf{R}_{k-1}^{\mathrm{N}}(n) \\ \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{X}_{k,n}^T \end{bmatrix} \tag{C.8}$$

The Householder transformation matrix $\mathbf{H}_k^{\mathrm{N}}(n)$ is such that

$$\mathbf{H}_k^{\mathrm{N}}(n)\overline{\mathbf{R}}_k^{\mathrm{N}}(n) = \begin{bmatrix} \mathbf{R}_{k-1}^{\mathrm{N}}(n) \\ \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{X}_{k,n}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k^{\mathrm{N}}(n) \\ 0 \end{bmatrix}$$

the above operation can be decomposed as in (C.4) to

$$\begin{aligned} &\mathbf{H}_k^{\mathrm{N},(i)}(n) \begin{bmatrix} \mathbf{R}_k^{\mathrm{N},(i-1)}(n) & & & & \\ 0 & \cdots & \mathbf{x}_{k,i}^{\mathrm{N},(i-1)} & \cdots & \mathbf{x}_{k,p}^{\mathrm{N},(i-1)} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_k^{\mathrm{N},(i)}(n) & & & & \\ 0 & \cdots & \mathbf{x}_{k,i+1}^{\mathrm{N},(i)} & \cdots & \mathbf{x}_{k,p}^{\mathrm{N},(i)} \end{bmatrix} \end{aligned} \tag{C.9}$$

where $\mathbf{x}_{k,j}^{(0)} = \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{x}_{k,j}, \mathbf{R}_k^{\mathrm{N},(0)}(n) = \mathbf{R}_{k-1}^{\mathrm{N}}(n)$ and the resultant Householder transformation matrix is $\mathbf{H}_k^{\mathrm{N}}(n) = \mathbf{H}_k^{\mathrm{N},(p)}(n)\mathbf{H}_k^{\mathrm{N},(p-1)}(n)\cdots\mathbf{H}_k^{\mathrm{N},(1)}(n)$. Similar steps as in (C.5)-(C.7) are carried out to get $\mathbf{H}_k^{\mathrm{N}}(n)$.

C.3 Calculation of Block Householder Transformation for QR Based Low Communication Robust Incremental Strategy

For QR based low communication scheme the updated input matrix at k^{th} node in time n is given by

$$\overline{\mathbf{R}}_k^{\mathrm{N}}(n) = \begin{bmatrix} \sqrt{\lambda}\mathbf{R}_k^{\mathrm{N}}(n-1) \\ \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{X}_{k,n}^T \end{bmatrix} \tag{C.10}$$

The Householder transformation matrix $\mathbf{H}_k^{\mathrm{N}}(n)$ is such that

$$\mathbf{H}_k^{\mathrm{N}}(n)\overline{\mathbf{R}}_k^{\mathrm{N}}(n) = \begin{bmatrix} \sqrt{\lambda}\mathbf{R}_k^{\mathrm{N}}(n-1) \\ \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{X}_{k,n}^T \end{bmatrix} = \begin{bmatrix} \mathbf{R}_k^{\mathrm{N}}(n) \\ 0 \end{bmatrix}$$

The above operation can be decomposed as in (C.4) to

$$\begin{aligned} &\mathbf{H}_k^{\mathrm{N},(i)}(n) \begin{bmatrix} & \mathbf{R}_k^{\mathrm{N},(i-1)}(n) & \\ 0 & \cdots \; \mathbf{x}_{k,i}^{\mathrm{N},(i-1)} \; \cdots & \mathbf{x}_{k,p}^{\mathrm{N},(i-1)} \end{bmatrix} \\ &\quad = \begin{bmatrix} & \mathbf{R}_k^{\mathrm{N},(i)}(n) & \\ 0 & \cdots \; \mathbf{x}_{k,i+1}^{\mathrm{N},(i)} \; \cdots & \mathbf{x}_{k,p}^{\mathrm{N},(i)} \end{bmatrix} \end{aligned} \tag{C.11}$$

where $\mathbf{x}_{k,j}^{(0)} = \sqrt{\boldsymbol{\varepsilon}_{k-1,k,n}^{\mathrm{N}}}\mathbf{x}_{k,j}, \mathbf{R}_k^{\mathrm{N},(0)}(n) = \sqrt{\lambda}\mathbf{R}_k^{\mathrm{N}}(n-1)$ and the resultant Householder transformation matrix is $\mathbf{H}_k^{\mathrm{N}}(n) = \mathbf{H}_k^{\mathrm{N},(p)}(n)\mathbf{H}_k^{\mathrm{N},(p-1)}(n)\cdots\mathbf{H}_k^{\mathrm{N},(1)}(n)$ Proceeding similarly as in (C.5)-(C.7) the expression for $\mathbf{H}_k^{\mathrm{N}}(n)$ is obtained.

Appendix D

D.1 Proof of the Proposed strategy

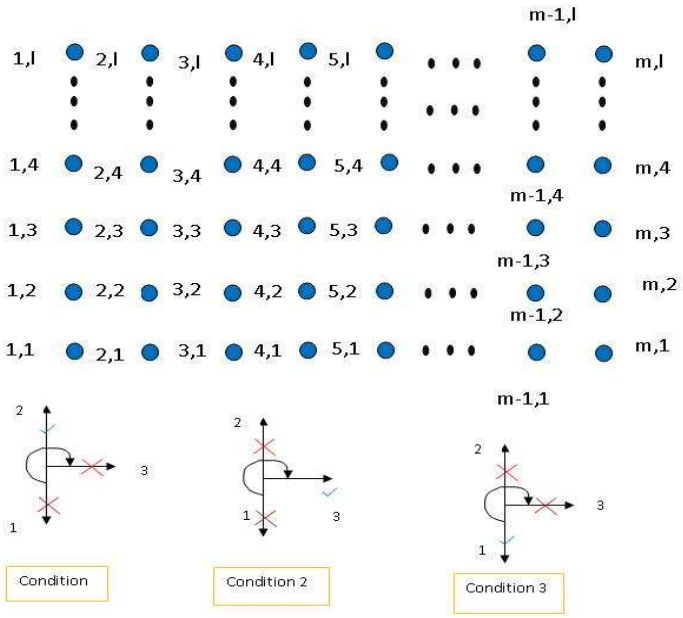

Figure D.1: Positions of the sensors present in the environment

Assumption: The sensor nodes are in regular manner. Suppose the arrangement of the sensor nodes form m and l number of rows and columns. Therefore the total number of sensor nodes are $N = m \times l$. Without loss of generality it can be assumed that the position of sensors are in unit distance apart in $x-$direction and $y-$direction. In order to analyze the routing strategy the position of the sensor node and its number is to be related. Suppose the first sensor is at $(1, 1)$ coordinate. Let for any sensor node the number of the sensor node be k and position of the sensor be (p, q). So the

number and positions of the sensor node can be related as

$$k_{number} = \begin{cases} (p-1)\,l + (l-q) & if\, p\, is\, even \\ (p-1)\,l + q & if\, p\, is\, odd \end{cases} \tag{D.1}$$

The relation between the position coordinates and the number of the sensor nodes depends upon the path of the incremental strategy. The relation between the sensor node position and number of the sensor node is shown in FigureD.1. By the strategy-1, -2 and -3 a sensor node in the position (l, m) can able to transmit the data to the sensor node in the positions $(l+1, m)$, $(l-1, m)$ and $(l, m+1)$, respectively. In order to prove that the incremental strategy shown in the Figure D.1 leads to find the global incremental path, the method of induction is used. The method of induction is based on the following three steps:

1. First, it requires to show that the proposed incremental strategy passes through the first node once in each iteration;

2. Assume that for any sensor node k the strategy is similar to the way it is defined;

3. It requires to prove that the strategy for the $(k+1)^{th}$ sensor node is the strategy for the k^{th} node with an additional strategy as defined by the proposed method.

In order to verify all these three above steps, some additional notations are introduced. Let ψ be the set which contains the number of the sensor nodes. The symbol ψ_k is set of the sensor node numbers from the first sensor node to the k^{th} sensor node. Thus ψ_N is the set which contains the entire number values corresponding to the entire WSNs. Obviously ψ_0 is the null set. Let ψ/ψ_k is the set of number corresponding to the sensors out of the set ψ_k.

Proof of Step 1: As explained in Section 6.3 by the proposed strategy the token is passed to the first node only from the N^{th} node through a predefined path. In order to prove that the first node updates the parameter once in one iteration it needs to show: (i) the first sensor receives the token only from the N^{th} node by a predefined path and (ii) it does not receive the token from any other sensor except N^{th} sensor.

Except the predefined path all other strategies passes the token to its neighbor sensor nodes. Thus if there is possibility of reception of the token by the first sensor node then it is only from its neighbor sensor nodes. The positions of neighbor sensor nodes of the first sensor are sensors present in the positions $(0,1)$, $(0,0)$, $(1,0)$, $(1,2)$, $(2,1)$ and $(2,2)$. The sensors at $(0,1)$, $(0,0)$, $(1,0)$ do not belong to the set ψ. Since there is no direction in the strategy to transmit the token from the sensor at position $(1,2)$ to the first sensor so there is no possibility of reception of token from it. Similarly there is no direction in the strategy to receive the token from the sensor at $(2,2)$. There is one strategy in the proposed method for the sensor at position $(2,1)$ to transmit the token to the first sensor. However this strategy is the third option to transmit. In the first option it may transmit to the sensor at position $(3,1)$ if not possible (in the case it has received the token from the node at $(3,1)$). Thus once the first sensor node has received the token from the N^{th} sensor node then the first sensor will not again receive the token until the token reaches the N^{th} sensor. That means in one iteration the first sensor node receives the token once in one iteration.

Let us consider for any sensor node t, the incremental path from the first sensor node to t^{th} sensor node is through the way $1, 2, \cdots, t$ in every iterations. In order to complete the proof of the proposed method by method of induction the objective is to find the incremental path of the WSN from the first node to the $(t+1)^{th}$ sensor node. The token passing starts from the sensor 1 with the strategy-2 since the sensor position in the direction of strategy-1 is $(0,1)$ which does not belong to ψ. Since the position of the first sensor is $(1,1)$ thus by the strategy-2 the first sensor chooses the sensor at the position $(2,1)$, whose number is 2, for transmission of the token. The direction of strategy-2 will be chosen by all the sensor node from position $(2,1)$ having number 2 to the sensor node at position $(m-1,1)$ having number $m-1$ because the sensor in the direction of the strategy-1 is the transmitted sensor. Sensor m chooses the strategy-3 for transmission of the token because the sensor in the direction of strategy-1 is the transmitted sensor and the sensor position in the direction of strategy-2 does not belong to ψ. For this strategy the receiving sensor is at position $(m,2)$ whose number is $m+1$. Sensors at the position $(m,2)$ whose number is $(m+1)$ to $(2,2)$

whose number is $(2m-1)$ choose the strategy-1 because the sensor in this direction is neither the transmitted sensor nor belongs to ψ. Thus the sensor at $(1,2)$ will choose the third strategy to send the data to $2m+1$ sensor. Thus the incremental path obtained by the proposed incremental strategy is $1, 2, ..., 2m+1$. Similarly the incremental path could be established for the sensor nodes having number greater than $2m+1$. Thus the incremental path for the sensor $t+1$ is $1, 2, ..., t, t+1$. That means it is the incremental path from the first sensor node to t^{th} sensor node with an additional strategy from t to $t+1$. Thus by method of induction it is proved that the proposed method helps to establish an incremental path from 1^{st} sensor node to N^{th} sensor node having one hop communication between two consecutive sensor node.

D.2 Khachiyan algorithm for the MVE formation

The Khachiyan algorithm is based on solution of the Langrage dual of the primary problem for the MVE formation. In order to make the problem simpler first the the center of the MVE to be formed is changed to origin. Then its Langrage dual is formed. In order to achieve it the following procedure is adopted.

1. The dimension of the position vector increases by one. The new position vectors and the negative of this position vectors are collected into one set. Let it be $\mathbf{S}^{'}$ which is given by

$$\mathbf{S}^{'} = \mathbf{y} \tag{D.2}$$

The original MVEE(S) can be obtained from the MVEE$S^{'}$ by

$$\mathbf{MVEE}\,(S) = \ \mathbf{MVEE}\left(S^{'}\right) \cap \mathbf{H} \tag{D.3}$$

where $\mathbf{H} = \left\{\mathbf{x} \in \Re^{d+1} : x_{d+1} = 1\right\}$

MVEE($S^{'}$) can be formed by achieving the solution of the following optimization problem.

$$\begin{aligned} \mathbf{Q}^p = \underset{\mathbf{Q}^p}{\arg\min} \; & -\log \mathrm{Det}\left(\mathbf{Q}^p\right) \\ \text{s.t.} \quad & \mathbf{y}_i^T \mathbf{Q}^p \mathbf{y}_i \leq 1,\; i = 1, \cdots, n \end{aligned} \tag{D.4}$$

2. The objective is to change the primary problem to secondary problem by using the Langrage dual. The dual form of the primary problem (D.4) is given by

$$\max_{\mathbf{u}} \; \log\det \prod(\mathbf{u}), \;\; \text{s.t.} \;\; \mathbf{e}^T\mathbf{u} = 1, \mathbf{u} \geq 0 \tag{D.5}$$

Thus the objective is to find the optimum solution for (D.5).

3. In order to achieve the solution of (D.5) with less computation, the first order approximation of the algorithm (D.5) is used. This is given by

$$\mathbf{MVEE}(S) = \left\{ \mathbf{x} \in \Re^d : [1/(d+1)] \begin{bmatrix} \mathbf{x}^T & 1 \end{bmatrix} \prod (\mathbf{u}^*)^{-1} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \geq 1 \right\} \tag{D.6}$$

This can be obtained directly from the factorization of the $\prod(\mathbf{u}^*)$. This is done as follows

$$\begin{aligned} \prod(\mathbf{u}^*) &= \begin{bmatrix} \mathbf{P}\mathbf{U}^*\mathbf{P}^T & \mathbf{P}\mathbf{u}^* \\ (\mathbf{P}\mathbf{u}^*)^T & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \mathbf{P}\mathbf{u}^* \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}\mathbf{U}^*\mathbf{P} - \mathbf{P}\mathbf{u}^*(\mathbf{P}\mathbf{u}^*)^T & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ (\mathbf{P}\mathbf{u}^*)^T & 1 \end{bmatrix} \end{aligned} \tag{D.7}$$

Taking the inverse of $\prod(\mathbf{u}^*)$ we get

$$\begin{aligned} &\prod(\mathbf{u}^*)^{-1} \\ &= \begin{bmatrix} 1 & 0 \\ -(\mathbf{P}\mathbf{u}^*)^T & 1 \end{bmatrix} \begin{bmatrix} \left(\mathbf{P}\mathbf{U}^*\mathbf{P} - \mathbf{P}\mathbf{u}^*(\mathbf{P}\mathbf{u}^*)^T\right)^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\mathbf{P}\mathbf{u}^* \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{D.8}$$

Using (D.8) the desired parameters of the center and spreading matrix are obtained as

$$\mathbf{Q}^* := (1/d)\left(\mathbf{P}\mathbf{U}^*\mathbf{P}^T - \mathbf{P}\mathbf{u}^*\left(\mathbf{P}\mathbf{u}^*\right)^T\right)^{-1}, \mathbf{c}^* = \mathbf{P}\mathbf{u}^* \tag{D.9}$$

Bibliography

[1] D. Estrin and M. Srivastav, "Instrumenting the world with wireless sensor networks," in *IEEE International conference on Acoustics, Speech, Signal Processing (ICASSP'01)*, May 2001, pp. 2033–2036.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] J. J. Xiao, A. Riberio, Z. Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Processing Magazine*, pp. 27–41, July 2006.

[4] Z. Q. Luo and J. N. Tsitsiklis, "Data fusion with minimal communication," *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1551–1563, Sep. 1994.

[5] 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, Aug. 2007.

[6] R. L. G. Cavalcante, I. Yamada, and B. Mulgrew, "An adaptive projected subgradient approach to learning in diffusion networks," *IEEE Transactions on Signal Processing*, vol. 57, pp. 2762–2774, July 2009.

[7] 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, July 2008.

[8] C. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *IEEE International conference on Acoustics, Speech, Signal Processing (ICASSP'08)*, Las Vegas, NV, April 2008, pp. 3285–3288.

[9] F. S. Cattivelli and A. H. Sayed, "Analysis of spatial and incremental LMS processing for distributed estimation," *IEEE Transaction on Signal Processing*, vol. 59, no. 4, pp. 1465–1480, April 2011.

[10] ——, "Distributed detection over adaptive networks using diffusion adaptation," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 1917–1932, May 2011.

[11] ——, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035 –1048, March 2010.

[12] P. J. Huber and E. Ronchetti, *Robust statistics*. John Wiley and Sons, 12-Feb-2009.

[13] R. Maronna, D. R. Martin, and V. J. Yohai, *Robust Statistics: Theory and Methods*. Wiley; 1 edition, 2006.

[14] S. Kassam and H. V. Poor, "Robust techniques for signal processing: A survey," *Proc. IEEE*, vol. 73, no. 3, pp. 433–481, 1985.

[15] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. Wiley-Interscience; 1 edition, 2003.

[16] T. J. W. McKean, *Robust Nonparametric Statistical Methods*. Hodder Education Publishers, January 30, 1998.

[17] J. W. McKean, "Robust analysis of liner model," *Statistical Science*, vol. 19, no. 4, pp. 562–570, 2004.

[18] J. T. Terpstra and J. W. McKean, "Rank-based analyses of linear models using r," *Journal of Statistical Software*, vol. 14, no. 7, July 2005.

[19] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," *in Proc. Adaptive Sensor Array Processing Workshop,Lexington, MA*, June 2006.

[20] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Journal on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[21] G. Mateos and G. B. Giannakis, "Distributed recursive least-squares: Stability and performance analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3740 –3754, July 2012.

[22] F. Cattivelli and A. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069 –2084, Sept. 2010.

[23] S. Kar and J. Moura, "Gossip and distributed kalman filtering: Weak consensus under weak detectability," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1766 –1784, April 2011.

[24] C. M. Ustebay, D. and M. Rabbat, "Distributed auxiliary particle filters using selected gossip," in *in Proc. IEEE Int. Conf. Acoustic, speech and Signal Proc., Prague, may 2011*, May 2011.

[25] O. M. C. M. Ustebay, D. and M. Rabbat, "Greedy gossip with eavesdropping," in *IEEE trans. on signal processing*, vol. 58, no. 7, July 2010, pp. 3765–3776.

[26] A. Bertand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2320–2330, May 2011.

[27] C. E. Davila, "An efficient recursive total east squares algorithm for fir adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 42, pp. 267–280, 1994.

[28] K. Furutsu and T. Ishiba, "On the theory of amplitude distribution of impulsive random noise and its application to the atmospheric noise," *Journal of Radio Research Laboratories, Japan*, vol. 32, no. 7, 1960.

[29] M.A.Gandhi and L.Mili, "Robust kalman filter based on a generalized maximum-likelihood-type estimator," *IEEE Transactions on Signal processing*, vol. 58, no. 5, pp. 2509–2520, May 2010.

[30] F. Cattivelli, C. Lopes, and A. Sayed, "A diffusion scheme for distributed estimation over adaptive networks," in *Signal Processing Advances in Wireless Communications, 2007. SPAWC 2007. IEEE 8th Workshop on*, June 2007, pp. 1 –5.

[31] J.-G. Hsieh, Y.-L. Lin, and J.-H. Jeng, "Preliminary study on wilcoxon learning machines," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 201 –211, Feb. 2008.

[32] B. Majhi, G. Panda, and B. Mulgrew, "Robust identification using new wilcoxon least mean square algorithm," *IEEE Electronics Letter*, vol. 45, no. 6, pp. 334–335, March 2009.

[33] U. Sahoo, G. Panda, and B. Mulgrew, "Sign-regressor wilcoxon and sign-sign wilcoxon," in *Advances in Recent Technologies in Communication and Computing (ARTCom), 2010 International Conference on*, Oct. 2010, pp. 35 –39.

[34] S. O. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2001.

[35] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Wiley-IEEE Press, 2003.

[36] B.F.Boroujency, *Adaptive Filters Theory and Applications*. Wiley, 1999.

[37] J. G. Hsieh, Y. L. Lin, and J. H. Jeng, "Preliminary study on wilcoxon learning machines," *IEEE transactions on neural networks*, vol. 19, no. 2, pp. 201–11, Feb. 2008.

[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

[39] P. Kumar and E. A. Yildirim, "Minimum-volume enclosing ellipsoids and core sets," *Journal of optimization Theory and Application*, vol. 126, no. 1, pp. 1 –21, July 2005.

[40] K. R. Liu, S. F. Hsieh, and K. Yao, "Systolic bock householder transformation for rls algorithm with two-level pipelined implementation," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 946–958, April 1992.

[41] 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, Jan 2010.

[42] J. Hajeck, Z. Sidak, and P. K. Sen, *Theory of Rank Tests*. Academic Press, 1999.

[43] N. Patwari, J. N. Ash, S. Kyperountas, A. O. H. III, R. L. Moses, and N. S. Correal, "Locating the nodes (cooperative localization in wireless sensor networks)," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[44] D. Blatt and A. O. Hero, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3614–3619, 2006.

[45] R. Shioda and L. Tuncel, "Clustering via minimum volume ellipsoids," *Journal of Computational Optimization and Application*, vol. 37, no. 3, pp. 247–295, July 2007.

[46] J. M. Jolion, P. Meer, and S. Bataouche, "Robust clustering with applications in computer vision," *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 13, no. 8, pp. 791–802, August 1991.

[47] P. Kumar, J. S. B. Mitchell, and E. A. Yildirim, "Approximate minimum enclosing balls in high dimensions using core sets," *ACM Journal of Experimental Algorithmics*, vol. 8, 2003.

[48] Y. E. Nesterov and A. S. Nemirovoski, *Interior Point Polynomial Methods in Convex programming*. Philadelphia, Pennsylvania: SIAM Publications, 1994.

[49] Y. Zhang, "An interior point algorithm for maximum volume ellipsoid problem," *Technical Report TR98-15, Department of Computational and Applied Mathematics, Rice University, Houston,Texas*, 1998.

[50] S.M.Kay, *Fundamentals of Statistical Signal Processing, volume 2: Detection Theory*. Prentice Hall, Englewood Cliffs,NJ, 1998.

[51] H.V.Poor, *An Introduction to Signal Detection and Estimation*. Springer Text in Electrical Engineering, 2nd edition Springer, New York, 1998.

[52] Y. Chen and Q. Zhao, "Impulsive noise," *On the life time of wireless sensor network*, vol. 9, no. 11, pp. 976 –978, 2005.

[53] A. Swami, Q. Zhao, Y. W. Hong, and L. Tong, *Wireless Sensor Networks: Signal processing and Communication Perspective*. Wiley, 2007.

[54] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Communication Letters*, vol. 11, no. 9, pp. 976–978, 2005.

[55] H.-X. Li and C. Qi, *Spatio-Temporal Modeling of Nonlinear Distributed Parameter Systems*. Springer, 2011.

[56] J. Zheng and A. Jamalipour, *Wireless Sensor networks: A networking perspective*. Wiley-IEEE Press, 2009.

[57] X. Zhao, S. Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, July 2012.

Dissemination

Journals

(a) **U.K.Sahoo**, G. Panda, B. Mulgrew and B. Majhi "QR-Based Incremental Minimum Wicoxon Norm Strategy for Wireless Sensor Networks," *Signal Processing, Elsevier* , Vol.92, issue 11, pp. 2657-2667, 2012 .

(b) **U.K.Sahoo**, G. Panda, B. Mulgrew, and B. Majhi, "Robust Incremental Adaptive Strategies for Distributed Networks to Handle Outliers in both Input and Desired Data," *Signal Processing, Elsevier*, Vol.96, Part B, pp. 300-309, 2014 .

(c) **U.K.Sahoo**, G. Panda, B. Mulgrew and B. Majhi, "Development of Robust Distributed Strategies for Wireless Sensor Networks using Rank Based Norms," *Signal Processing, Elsevier*, Vol.101, pp. 218-228, 2014.

(d) **U.K.Sahoo**, G. Panda and B. Mulgrew, "Energy Efficient Environment Monitoring using Minimum Volume Ellipsoid and Core Set," *Under review Signal Processing, Elsevier*,

Conferences

(a) **U.K.Sahoo**, G. Panda and B. Mulgrew, "Sign-regressor Wilcoxon and Sign-Sign Wilcoxon," in *International Conference on Advances in Recent Technologies in Communication and Computing*, at Kottayam, Kerala, India, October, 2010

(b) **U.K.Sahoo**, G. Panda and B. Mulgrew, "Energy efficient environment monitoring using minimum volume ellipsoid," in *IEEE International Conference on electronics system -2011*, NIT Rourkela, Rourkela, INDIA, Jan. 2011.

BIO-DATA

Upendra Kumar Sahoo

Date of Birth: 7^{th} June, 1982

Correspondence:
PhD Scholar, Department of Electronics and Communication Engineering,
National Institute of Technology Rourkela, India – 769 008.

Ph: +91 94373 59787 (M)

e-mail: sahooupen@nitrkl.ac.in, uksahoo01@gmail.com

Qualification

- Ph.D. (Continuing)
  National Institute of Technology Rourkela, Odisha, India
- M.Tech. (Electronics and Communication Engineering)
  National Institute of Technology Rourkela, Odisha, India [First division]
- B.E.
  University College of Engineering, Burla, Odisha, India [First class]
- +2 (Science)
  Council of Higher Secondary Education, Odisha, India [First division]
- 10th
  Board of Secondary Education, Odisha, India [First division]

Publications

- 3 Journal Articles
- 2 Conference Articles