Title of thesis	Adaptive Linear System Identification over Simulated Wireless Environment

I, MUSAB JABRALLA OMER ELAMIN

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP.
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as
   - ☐ Confidential
   - ☑ Non-confidential

If this thesis is confidential, please state the reason:

____

The contents of the thesis will remain confidential for ______________ years.

Remarks on disclosure:

____

Endorsed by

______________________ ______________________

Signature of Author Signature of Supervisor

____

Signature of Co-Supervisor

Date: ______________ Date: ______________

UNIVERSITI TEKNOLOGI PETRONAS

Approval by Supervisors

The undersigned certify that they have read, and recommend to The Postgraduate Studies Programme for acceptance, a thesis entitled **Adaptive Linear System Identification over Simulated Wireless Environment** submitted by **Musab Jabralla Omer Elamin** for the fulfillment of the requirements for the degree of Master of Science in Electrical and Electronic Engineering.

Date: ________________

Main Supervisor	:	Dr. Vijanth S. Asirvadam
Signature	:	________________
Date	:	________________
Co-Supervisor	:	Dr. Nordin Saad
Signature	:	________________
Date	:	________________

UNIVERSITI TEKNOLOGI PETRONAS

Adaptive Linear System Identification over Simulated Wireless Environment

By

Musab Jabralla Omer Elamin

A THESIS

SUBMITTED TO THE POSTGRADUATE STUDIES PROGRAMME

AS A REQUIREMENT FOR THE

DEGREE OF MASTER OF SCIENCE

IN ELECTRICAL AND ELECTRONIC ENGINEERING

Electrical and Electronic Engineering

BANDAR SERI ISKANDAR,

PERAK

February 2009

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Signature : ____________________________

Name : Musab Jabralla Omer Elamin

Date : ____________________________

ABSTRACT

Wireless technologies have become one of the basic industrial pillars, whereas system identification represents an important tool in many practical engineering circumstances and thus sooner or later both wireless technologies and system identification should be linked together in sense of having an identifier that is able to reliably identify a system over wireless links. It is well known that wireless links are considered as unreliable medium and therefore the loss of the system observations across them is unavoidable. The system observations represent the main element in the identification process since the identifier relies only on these observations in order to identify the underlying function of the system as they are the only information available to tell about the system dynamics, for this reason vast amount of literature in the context of system identification is written about the way the excitation signal is chosen to force the system to show its dynamic and also about the way the sampling process is carried out to obtain informative observations in order to construct a satisfactory model for the system. This shows that the random loss of these observations (which are vital and core element of identification process) might deter the system modeling process. Experience shows that well sampled observations over regular intervals during observations loss could not guarantee a satisfactory model for the system. This thesis looks into the concepts of system identification and the behavior of the identifier components when placing wireless links between the system and the identifier. The thesis investigates the possibility of performing system identification over wireless network for both on-line and off-line system identification approaches. This research work studies the effects of observations loss on the performance of the learning algorithms and it focuses only on first order autoregressive with exogenous input (ARX) model structure adopted on linear network. The work looks thoroughly on three forms of instantaneous learning algorithms which are: first order algorithms (e.g. least mean square (LMS)), second order algorithms (e.g. recursive least squares (RLS)) and finally high order or sliding window (SW) algorithms (e.g. moving average (MA)).

ABSTRAK

Kejuruteraan tanpa wayar buat masa kini menjadi teknologi asas di adalam industri di mana pengenalpastian sistem mewakili perkara yang penting didalam kejuruteraan harian. Maka kedua-dua teknologi, tanpa wayar dan pengenalpastian sistem harus digabungkan supaya unit pengenalpastian dapat mengenali dan membaca sistem yang dinamik jika masukan datanya dihantar melalui teknologi tanpa wayar. Teknologi tanpa wayar diketahui mempunyai sambungan media yang tidak boleh diharapkan mungkin terdapat masukan datanya yang hilang melalui sambungan tersebut. Data masukan untuk sesuatu sistem merupakan unsur utama bagi proses pengenalanpastian sistem di mana unit pengenalpastian memerlukan data masukan yang bersiri untuk membina model atau memeta fungsi dalaman sistem tersebut yang dinamik. Terdapat banyak rujukan dalam hubungan pengenalpastian sistem dimana sesuatu proses atau sistem diberi data masukan yang tidak linear untuk membolehkan (atau memaksa secara langsung) sistem tersebut untuk menonjolkan ciri–ciri dinamik yang tersirat. Maka dengan itu, data-data yang disampel digunakan untuk membina satu model yang munasabah untuk sistem tersebut. Ini menonjolkan data-data yang hilang semasa proses penghantaran melalui media tanpa wayar yang akan memberi kesan negatif kepada proses pengenalpastian sistem. Pengalaman juga dapat menunjukkan data-data yang disampel dengan menggunakan selang yang tetap tidak juga memastikan untuk memberi model yang tepat dan betul. Tesis ini mengaji konsep pengenalpastian sistem dan menilai perubahan kepada unit pengenalpastian jika terdapat sambungan dan penghantaran data melalui teknologi tanpa wayar di antara sistem dan unit pengenalpastian tersebut. Tesis ini juga mengkaji kebarangkalian untuk melakukan proses pengenalpastian sistem (melalui media tanpa wayar) secara rekursi atau dalam talian atau mengunakan sistem data-data yang terkumpul luar talian. Penyelidikkan melalui tesis ini menilai kesan data-data yang mungkin hilang kepada prestasi algoritma atau prosedur yang menumpukan kepada struktur rangkaian linear tertib pertama ‘Autoregressive Exogenous Input (ARX)’. Keseluruhan proses pengenalpastian di dalam penyelidikan ini hanya melihat kepada

algortima linear yang diadaptasikan secara spontan. Kerja penyelidikan ini mengkaji secara mendalam tiga algoritma sistem pengenalan spontan di mana: algoritma linear tertib pertama (iaitu punca kuasa dua mean minima (LMS)), algoritman linear order kedua (iaitu punca kuasa dua minima rekursi (RLS)) dan akhirnya algoritma linear order tinggi (iaitu purata bergerak (MA)).

ACKNOWLEDGEMENTS

First and foremost, all praises and thanks are due to Allah, the almighty God, the source of life and hope for giving me the strength and wisdom to succeed in life.

My very special thanks go to my supervisor Dr. Vijanth S. Asirvadam, his support, encouragement, valuable assistance, and unwavering patience rescued me from frustration and guided me throughout my research. Without his excellent guidance it would have been impossible to finish this work. His very positive influence on my professional development will be carried forward into my future career.

I would like to thank my family for their endless support and for giving me courage and strength to never give up, even though we’ve been a couple of continents apart for so long. They provided me sturdy motivation and moral support which I have needed throughout my life, they are always close to my heart.

My special thanks are extended to my colleagues at Electrical and Electronic Engineering Department. They have been very friendly and gracious which provided me an excellent study environment.

At the end, I would like to thank my friends for their consistent support, help, encouragement and all the joy that they have brought to my life.

Musab Jabralla Omer Elamin

Universiti Teknologi Petronas, February 2009

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

1.1 Introduction

The increasing availability of wireless networks and the great interest they are being given on the industrial front as well as the emergence of wireless sensor networks were the prime motivations to study the possibility of performing system identification over wireless network.

The current trend towards adopting wireless networks has its own considerable set of favorable reasons: first of all, it reduces the cost and time for network installation and maintenance. On other hand, it adds flexibility to the plant floor architecture since it would be easy to change it to meet the current requirements. It also makes the task of accessing a device into the plant network for diagnostic or programming purposes much easier. Furthermore, it is the solution when the plant environment involves vibrations, chemicals or moving parts that can damage any kind of cabling [1].

The disadvantage of replacing wired network with its wireless counterpart is the presence of significant information losses across the wireless counterpart.

1.2 Problem Statement

This thesis investigates parametric system identification over lossy network. Parametric identification sometimes called black box modeling since the identifier looks at the system as if it was a black box and uses only an observed output signal (y) and the corresponding input signal (u) to model the system (see Figure 1.1), which means the physical features of the system are ignored and only the available observations are used to describe the underlying function of the system. Thus, these observations are of great necessity as they are the only available source of information to tell about the system dynamics.

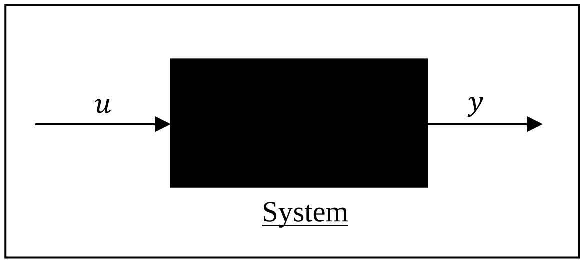


Figure 1.1: Black Box Modeling

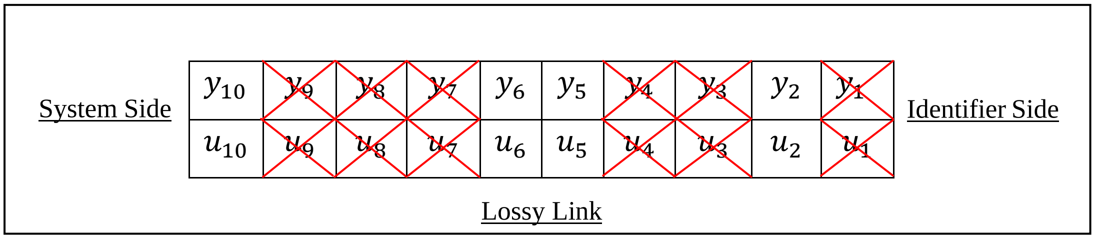
One of the main system identification processes is to sample the system information at regular intervals according to certain rules in order to assure that the obtained observations are informative. The process always ensures observations presented to the identifier in-full and at the same regular intervals that they have been sampled at as it is demonstrated in Figure 1.2(a). The figure depicts a sequence of system input/output observations sent to identifier over reliable link.

The challenge of this research work is that when sending the system observations over wireless links (which are known to be unreliable) to the identifier, the observations availability in full at the identifier side becomes hard to guarantee. Moreover, the observations structure at the identifier side might be in a random form as it is demonstrated in Figure 1.2(b). The figure depicts a sequence of system input/output observations sent to identifier over lossy link.

System Side	y_{10}	y_9	y_8	y_7	y_6	y_5	y_4	y_3	y_2	y_1	Identifier Side
	u_{10}	u_9	u_8	u_7	u_6	u_5	u_4	u_3	u_2	u_1	

Reliable Link

(a) Observations Transmission over Reliable Link



(b) Observations Transmission over Lossy Link

Figure 1.2: Observations Transmission over Reliable and Lossy Links

Observations loss during wireless transmission might deteriorate the quality of the produced model and deter the modeling process, especially when modeling the system on-line or instantaneously because only current observations will be used to obtain the model unlike off-line case where batch of observations is used to obtain the model.

1.3 Motivated Work

Networked control system with random observations loss is a subject of current research in the field of control theory and applications. The current research looks into the system controllability over communication links subject to random observation loss. For example, in [2] Kalman filter has been used to estimate the state of a controlled system using observed data provided by sensor network for control purpose. The effect of observations loss on the statistical convergence properties of Kalman filter has been studied, where observations arrival has been modeled as a Bernoulli process and it has been assumed that observation measurements are received in full or lost completely. It has been showed the existence of a critical value for observations arrival rate, beyond which a transition to unbounded error occurs and it has been given the upper and lower bounds on this expected error. In [3] again Kalman filter has been studied where it is used to estimate the state of a controlled system for control purpose, but this time Kalman filtering problem has been studied with inclusion of partial observation loss. Observations arrival has been modeled as a Bernoulli process and the statistical convergence properties of Kalman filter as a function of the network throughput has been investigated. A throughput region that guarantees the error convergence and an unstable throughput region such that the error is unbounded have been found. In [4] optimal estimation design in networked control system subject to random delay and packet loss has been studied, where the sensors measurement of sampled linear system are transmitted to the estimator side via a generic digital communication network. The network has been modeled as a module between the plant and the estimator which delivers observation measurement to the estimator with possibly random delays and allows also for packets with infinite delay which corresponds to packet loss. Two time-invariant estimator architectures have been presented and it has been showed that the stability does not depend on the packet delay but only on the packet loss probability. Algorithms to compute critical packet loss probability and estimators performance in terms of their error have been given and applied to some numerical examples. In [5] the stability of discrete-time networked control systems over a communication channel subject to packet loss has been studied, where the behavior of packets loss has been modeled as a Bernoulli process. A necessary and sufficient condition for stability has been obtained and a packet dropping margin has been introduced as a measure of stability robustness of a system then a formula for it has been derived. Finally a design method has been proposed for achieving a large packet dropping margin.

The previous work has studied the performance of control systems over lossy network which has motivated this research work on system identification over lossy network. The presented work in this thesis looks into the performance of system identification approaches over a communication channel subject to random observations loss.

1.4 Research Objectives

The objective of this research work is to model a sampled linear system of first order based on information obtained from its observations which are subject to random loss across the transfer medium. This main objective can be divided into the following sub-objectives:

- Investigating the performance of system identification approaches during observations loss.

- Highlighting the influential factors that could mitigate the observations loss effect on the modeling process.

- Optimizing the learning algorithms according to the special features and structure of each one of them to improve the performance during observations loss.

1.5 Research Scope

This research work will discuss the performance of least squares (LS) algorithm (as an example of off-line approaches) during observations loss, then it will focus on investigating and optimizing three categories of on-line linear learning algorithms when observations are subject to random loss, namely:

- First order learning algorithms (least means square (LMS) and normalized least mean square (NLMS)).

- Second order learning algorithms (recursive least squares (RLS) and recursive instrumental variable (RIV)).

- High order learning algorithms (moving average (MA) and normalized moving average (NMA)).

Throughout this research work only parametric identification and particularly first order autoregressive with exogenous input (ARX) model structure adopted on linear network will be considered.

1.6 Research Methodology

This thesis is conducted according to the following four phases:

Phase One:

- Reviewing system identification basics.
- Simulating wireless links effect on the transmitted observations.
- Discussing LS algorithm performance during observations loss.

Phase Two:

This phase starts the on-line algorithms investigation and optimzation with the algorithms of the first order.

- Investigating the sampling process role in mitigating the observations loss effect.
- Investigating the observations loss effect on the output error behavior and its reflection on the first order algorithms performance.
- Optimizing each of LMS and NLMS on the ground of the obtained results from the previous two steps.

Phase Three:

This phase continues the on-line algorithms investigation and optimzation with the algorithms of the second order.

- Investigating the observations loss effect on the covariance matrix structure and its consequences on the second order algorithms performance.

- Employing the obtained results until this point to optimize each of RLS and RIV.

Phase Four:

This phase ends the on-line algorithms investigation and optimzation with the algorithms of the high order.

- Introducing some Data Store Management (DSM) strategies to reduce the observations loss effect.

- Optimizing each of MA and NMA by adopting the introduced DSM strategies and exploiting the obtained results from the second phase.

1.7 Thesis Organization

The thesis begins at chapter 1 which describes the problem statement, motivated work, research objectives, research scope and research methodology. The chapter also lists down the thesis contributions.

Chapter 2 reviews system identification basics and simulates the wireless links effect on the transmitted observations. It also discusses the least squares (LS) algorithm performance during observations loss.

Chapter 3 starts the investigation and optimization of the instantaneous learning algorithms with the algorithms of the first order. The chapter addresses the role of sampling process in mitigating observations loss effect. It also looks into the output error behavior during observations loss and its reflection on the first order algorithms performance. The chapter then proceeds to optimize the first order algorithms on the ground of the obtained results.

Chapter 4 continues the investigation and optimization of the instantaneous learning algorithms with the algorithms of the second order. The chapter shows the effect of observations loss on the covariance matrix structure and its consequences on the second order algorithms performance. The chapter then proceeds to propose a technique for the second order algorithms to mitigate the effect of observations loss.

Chapter 5 ends the investigation and optimization of the instantaneous learning algorithms with the algorithms of the high order. The chapter addresses the role of the data store management (DSM) strategies in mitigating observations loss effect. The chapter then introduces some DSM strategies and proposes some methods to improve the performance of the high order algorithms.

Chapter 6 concludes the work and shed some light on the possible future work that can be carried over based on the foundation work laid out by this research.

1.8 Thesis Contribution

The earlier work in this thesis has tried to implement linear off-line system modeling over simulated wireless environment. Off-line learning approach during observations loss was still able to model the system which can be attributed to the fact that off-line approach obtains the model from batch of the system observations and therefore observations loss does not deteriorate the learning process drastically as it would be the case for on-line approach.

The main contribution of the thesis is optimizing instantaneous learning algorithms for instantaneous system modeling with lost observations. The thesis has adopted oversampling rule to compensate the lost observations and therefore the learning algorithms should be robust to work with correlated observations (when wireless link is strong) as well as random structure of observations (when wireless link is weak). The optimized algorithms in this thesis combine all the mentioned requirements above.

The thesis has proposed a technique known as sine function based de-correlation (SD) for first order algorithms to treat the correlation effect instantaneously at the identifier side. New version of LMS known as LMS-SD has been introduced by adopting SD technique on LMS algorithm. The thesis has also proposed another technique known as error displacement based update (EDU) to mitigate the undesired contribution of the output error in the weights update process during observations loss. Another version of LMS known as LMS-SDEDU has been proposed by adopting a combination of SD and EDU techniques on LMS algorithm.

The thesis has also proposed an optimized second order learning algorithm for system modeling over lossy network. The optimized algorithm known as RLS with weights update based on observations continuity (RLS-OC) and it uses simple weights update skipping technique to improve the performance.

The thesis has further introduced two data store management strategies known as error measurement (EM) and intelligent data store management (IDSM). New version of MA known as MA-EMSD has been introduced by adopting a combination of EM strategy and SD technique on MA algorithm. Another version of MA termed as intelligent moving average (IMA) has been proposed by adopting a combination of IDSM strategy and SDEDU method on MA algorithm.

Other contribution of the thesis will be summarized as follows:

1. The proposed SD technique and SDEDU method have been also adopted on NLMS algorithm.

2. The RIV algorithm has been tested using the proposed weights update based on observations continuity technique.

3. The NMA has been tested for both EMSD method and the intelligent approach.

4. Using Matlab/SimulinkTM environment wireless links effect on the transmitted observations has been simulated and robust code of real time s-function has been written to implement the optimized algorithms.

CHAPTER TWO: SYSTEM IDENTIFICATION

2.1 Introduction

In many engineering circumstances there is a need to model a system, such circumstances might be the need to achieve deeper knowledge about the system, predict the system output, simulate the system behavior under certain conditions or extract information about how a regulator can be designed to control the system output [6].

Sometimes models can be constructed from physical laws and principles which is known as *physical modeling*. However, when system becomes complicated it is difficult or impossible task to find the physical laws that govern its behavior. Moreover, physical modeling is considered time consuming and needs a lot of mathematical derivations. In such cases the alternative option is to model the system using *system identification* approaches.

System identification is the art of understanding the underlying function of the system by observing only its input and output. System identification deals with the problem of modeling dynamic system (see Figure 2.1) given an observed output signal (y) and the corresponding input signal (u).

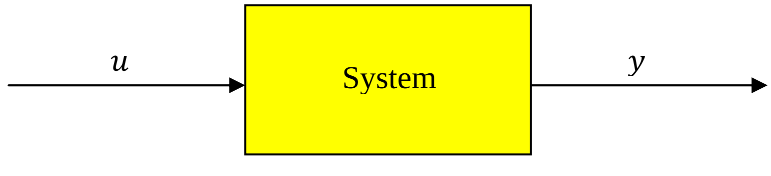


Figure 2.1: System with Input u and Output y

System identification field is divided broadly into parametric and non-parametric identification. In parametric identification models involve parameters (e.g. coefficients of difference equation) while in non-parametric identification models do not involve parameters and they are usually graphical representations (e.g. impulse response function) [7]. This work considers only parametric identification for modeling purpose.

There are many standard model structures for parametric identification; some of the popular ones are the following:

- Autoregressive (AR):

$$y_k = -a_1\, y_{k-1} - a_2\, y_{k-2} - \cdots - a_n\, y_{k-n} + e_k \tag{2.1}$$

- Autoregressive with exogenous input (ARX):

$$\begin{aligned} y_k = &- a_1\, y_{k-1} - a_2\, y_{k-2} - \cdots - a_n\, y_{k-n} \\ &+ b_1\, u_{k-1} + b_2\, u_{k-2} + \cdots + b_m\, u_{k-m} + e_k \end{aligned} \tag{2.2}$$

- Autoregressive moving average with exogenous input (ARMAX):

$$\begin{aligned} y_k = &- a_1\, y_{k-1} - a_2\, y_{k-2} - \cdots - a_n\, y_{k-n} \\ &+ b_1\, u_{k-1} + b_2\, u_{k-2} + \cdots + b_m\, u_{k-m} \\ &+ e_k + c_1\, e_{k-1} + c_2\, e_{k-2} + \cdots + c_l\, e_{k-l} \end{aligned} \tag{2.3}$$

where y is the system output, u is the system input, e is the modeling error and each of a_i, b_i and c_i are the model weights to be estimated.

Dynamic system modeling is usually carried out using ARX model structure [8], only ARX model structure will be considered throughout the thesis since all the benchmark test problems in this work are dynamic systems (see appendix A). On other hand, this research work investigates and optimizes instantaneous learning algorithms rather than model structures.

Parametric identification can be divided into two different forms of weights adjustment which are off-line and on-line identification. In off-line or batch identification system observations are gathered in batch then they are processed to adjust the model weights. In on-line or instantaneous identification system observations are processed instantaneously to adjust the model weights.

This chapter will discuss in brief off-line identification during observations loss. The further chapters of the thesis will look into the instantaneous identification which will be the scope of this research work.

2.2 Simulation of Wireless Links Effect

The assumptions that have been made in this work to simulate the effect of wireless links on the transmitted observations are the following:

- System input/output observations are to be time stamped, encapsulated into packets (each packet consists of one pair of input and output observations) and transmitted over wireless links to the identifier.

- The packet is considered lost if it has not been received after a certain prescribed time and lost packets are not retransmitted.

- The identifier receives observations in full or none and no partial loss is considered.

- The packets arrival at the identifier side is modeled via Bernoulli process (β_k) which has two possible values: 0 and 1 at any time instant, where:

$$\beta_k = \begin{cases} 1 & \text{Indicates the packet has been received successfully.} \\ 0 & \text{Indicates the packet has been lost completely during transmission.} \end{cases}$$

The probability of ($\beta_k = 0$) is invariant with time and known as packet dropping probability (PDP) [5].

Modeling observations arrival via Bernoulli process with probability of observation received correctly (1-PDP) is corresponding to the binary symmetric channel (BSC) with probability of error (PDP). The throughput of this channel (for a fixed sampling rate and a given packet size) is the product of the sampling rate, the packet size and the probability of observation received correctly [3] as it is shown below:

$$Throughput = \text{sampling rate} \times \text{packet size} \times (1 - \text{PDP}) \qquad (2.4)$$

Since the probability (1-PDP) is a scaled version of the throughput for a fixed sampling rate and a given packet size, it can be used as a reference for the channel throughput. In this research work PDP will be used to describe the rate of observations loss.

Observations arrival modeling via Bernoulli process is illustrated in Figure 2.2. In the figure y_t and u_t are the transmitted output and input respectively, y_r and u_r are the received output and input respectively and $\hat{y}$ is the estimated output.

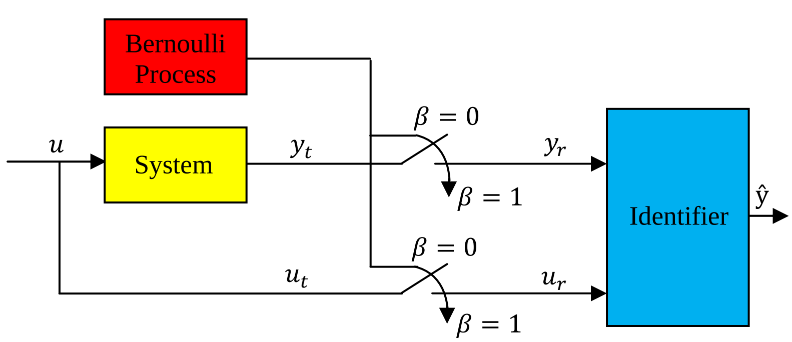
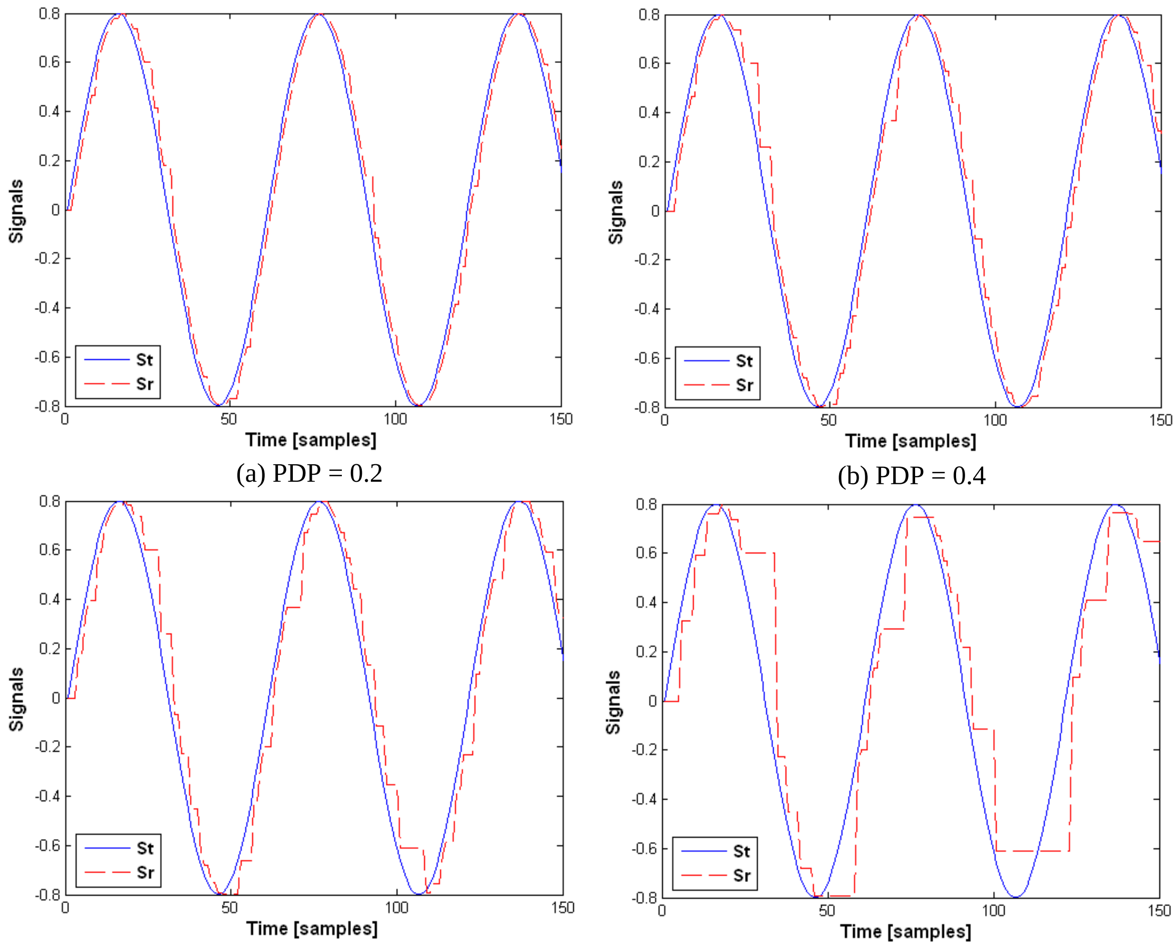


Figure 2.2: Observations Arrival Modeling via Bernoulli Process

The transmitted signal (S_t) and the received one (S_r) over the lossy link are depicted in Figure 2.3 at different settings of PDP. From the graphs it can be seen that as PDP increases the received signal deteriorates.



(a) PDP = 0.2 (b) PDP = 0.4

(c) PDP = 0.6 (d) PDP = 0.8

Figure 2.3: Transmitted and Received Signals at Different Settings of PDP

2.3 Least Squares (LS)

Least squares algorithm is an off-line approach for modeling dynamic system by processing batch of its input/output observations. Least squares estimator [6] can be derived by rewriting ARX model in equation (2.2) using the backward shift operator (q^{-1}) as follows:

$$A(q^{-1})\, y_k = B(q^{-1})\, u_k + \, e_k \tag{2.5}$$

where:

$$A(q^{-1}) = 1 + \, a_1 q^{-1} + \cdots + \, a_n q^{-n} \tag{2.6}$$

$$B(q^{-1}) = \, b_1 q^{-1} + \cdots + \, b_m q^{-m} \tag{2.7}$$

The model in equation (2.5) can be written equivalently as a linear regression as follows:

$$y_k = \varphi_k^T\, \theta + \, e_k \tag{2.8}$$

where:

$$\varphi_k^T = [-y_{k-1} \dots - y_{k-n} \quad u_{k-1} \dots \, u_{k-m}] \tag{2.9}$$

$$\theta = [a_1 \dots a_n \;\; b_1 \dots b_m]^T \tag{2.10}$$

Assuming that u_1 , y_1, … u_m , y_n are available, the lest squares estimation ($\hat{\theta}$) for the weights vector (θ) is defined as the minimization of the sum of the squared equation errors:

$$V_N(\theta) = \frac{1}{N} \sum_{k=1}^{N} e_k^2 \tag{2.11}$$

$$V_N(\theta) = \frac{1}{N} \sum_{k=1}^{N} y_k^2 - \left[\frac{2}{N} \sum_{k=1}^{N} y_k \varphi_k^T\right] \theta + \theta^T \left[\frac{1}{N} \sum_{k=1}^{N} \varphi_k \varphi_k^T\right] \theta \tag{2.12}$$

Now by setting the gradient of $V_N(\theta)$ to zero:

$$\left[\frac{1}{N} \sum_{k=1}^{N} \varphi_k \varphi_k^T\right] \hat{\theta} = \frac{1}{N} \sum_{k=1}^{N} \varphi_k\, y_k \tag{2.13}$$

Assuming the matrix in (2.13) to be nonsingular:

$$\hat{\theta} = \left[\frac{1}{N}\sum_{k=1}^{N} \varphi_k \varphi_k^T\right]^{-1} \left[\frac{1}{N}\sum_{k=1}^{N} \varphi_k \, y_k\right] \tag{2.14}$$

Equation (2.14) is the solution of the least squares weights estimation.

2.4 Off-line System Identification Setup over Wireless Network

Figure 2.4 illustrates the setup of off-line system identification over wireless network, the figure shows that the system observations are sent to the identifier over wireless network and stored upon their arrival in a store at the identifier side, after receiving the whole training set the stored observations is processed in order to obtain the model. In the figure y_t and u_t are the transmitted output and input respectively, y_r and u_r are the received output and input respectively and $\hat{y}$ is the estimated output.

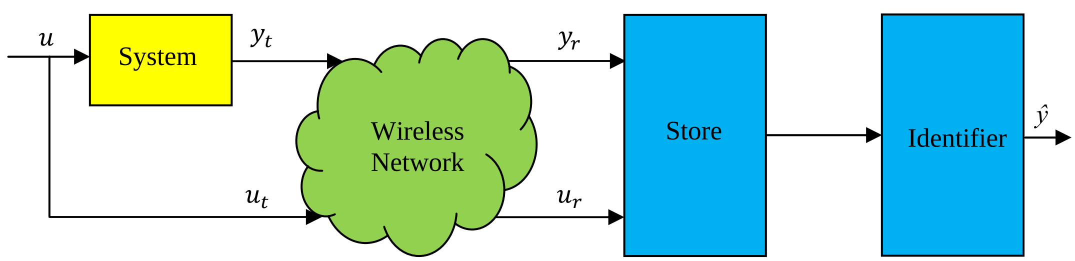
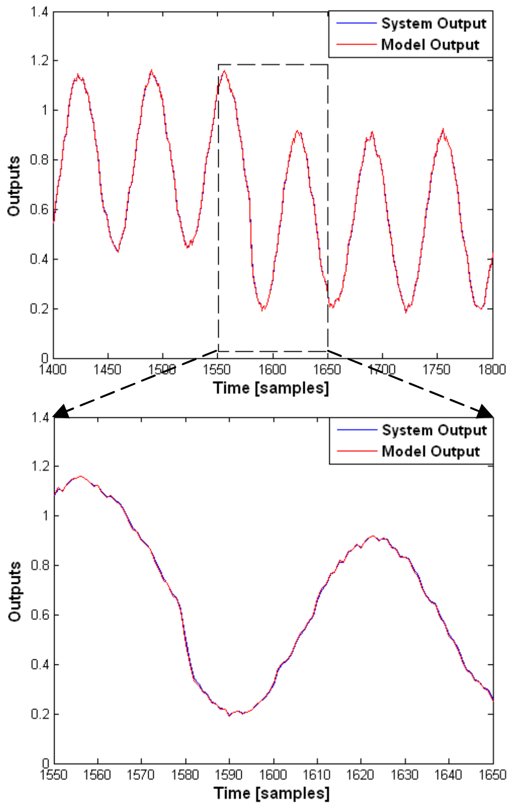


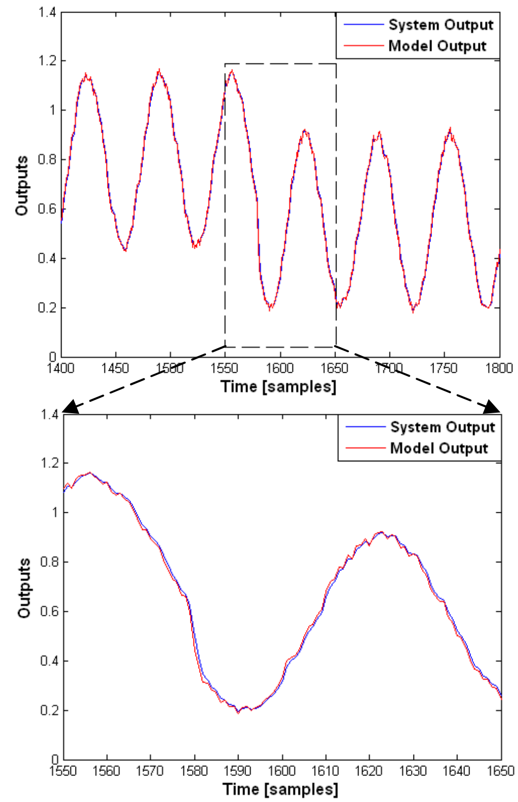
Figure 2.4: Off-line System Identification Setup over Wireless Network

2.5 Least Squares Performance over Lossy Link

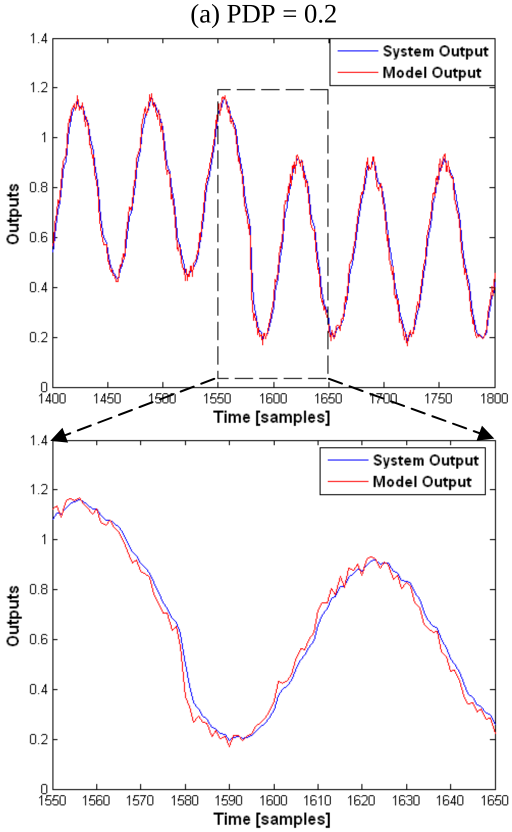
Test problem I (refer to appendix A) used to show the performance of the least squares algorithm at different settings of PDP. Figure 2.5 shows the output of the system (actual output) and the model (estimated output).



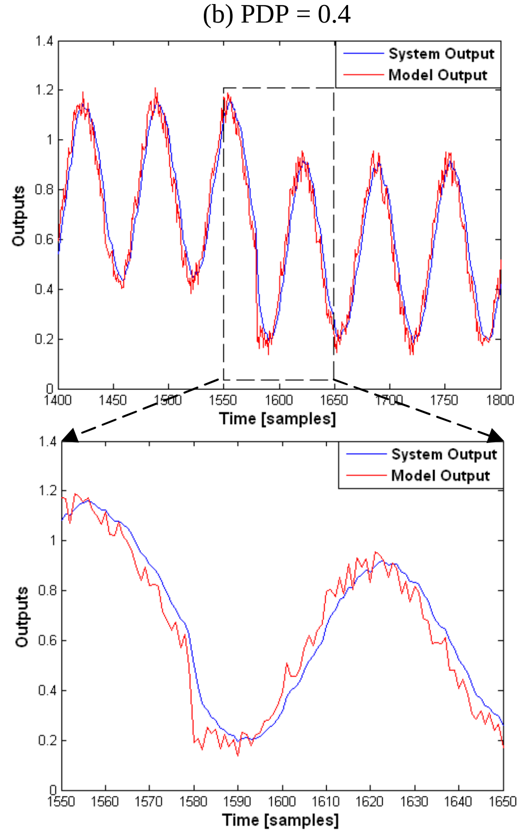
(a) PDP = 0.2



(b) PDP = 0.4



(c) PDP = 0.6



(d) PDP = 0.8

Figure 2.5: Least Squares Performance over Lossy Link at Different Settings of PDP

The period between 1550 and 1650 in the graphs shows clearly the estimation error at the sharp edges of the signals. Looking through the graphs it can be seen that as PDP increases the estimation error increases.

The performance of the least squares algorithm is acceptable to some extend since it is still able to capture the changes in the system dynamics. This capability of modeling the system off-line during observations loss can be attributed to the fact that off-line modeling is carried out by processing batch of the system observations, so even when some of these observations are lost the algorithm will be still having set of observations from which it can extract information about the system dynamics.

2.6 Conclusion

This chapter has reviewed basic concepts of system identification, it has started by giving brief introduction to system identification and showed how wireless links effect on the transmitted observations can be simulated, then it has proceeded to discuss the least squares algorithm performance (as an example of off-line identification approaches) during observations loss.

Simulation results have showed that the least squares algorithm performance during observations loss is acceptable to some extend since it is still able to capture the changes in the system dynamics. With careful choice of the excitation signal informative observations can be produced and better models can be constructed under observations loss circumstances using off-line approaches.

In order to keep focus on the thesis contribution this research report will discuss only instantaneous system modeling with lost observations from next chapter onwards.

CHAPTER THREE: FIRST ORDER INSTANTANOUS LEARNING ALGORITHMS

3.1 Introduction

Traditionally on-line system identification using linear networks has been performed by assuming a standard linear structure for the model and adjusting the structure weights based on the system observations using an instantaneous learning algorithm.

Instantaneous learning algorithms are procedures used to adjust the model structure weights based on the system observations. Thus, whenever a new observation is available the instantaneous learning algorithm measures the gradient of the performance surface for the current model and updates the weights in order to minimize the error criterion. The highly cited LMS algorithm by Widrow and its normalized version (normalized least mean square (NLMS)) are examples for first order instantaneous learning algorithms, both of them have been used in wide range of applications due to their simplicity, minimal computational complexity, minimal memory requirements and the strong theoretical basis for weights convergence [9][10].

Instantaneous learning algorithms are well studied and applied in the context of system identification when system observation are well sampled and fully received in orderly manner at the identifier side. When considering wireless network as a transfer medium, random observations loss becomes unavoidable (due to wireless impairments) which deteriorates the quality of the produced model using these algorithms.

The objective of this chapter is to investigate and optimize each of LMS and NLMS algorithms to improve their performance during observations loss. The chapter highlights the role of the sampling process and shows that oversampling improves the performance. Since oversampling could lead to correlated observations at the identifier side (when wireless link is strong), a new technique is proposed to treat the correlation effect instantaneously at the identifier side. The chapter also addresses the behavior of the output error component during observations loss where it contributes with high magnitude in the weights update process and directs the algorithm sub-optimally. Therefore, the chapter introduces another technique to reduce this undesired contribution of the output error.

3.2 On-line System Identification Setup over Wireless Network

The difference between off-line and on-line system identification setups over wireless network is shown in Figure 3.1. As it can be seen from the figure in off-line case the system observations are stored upon their arrival in a store and processed after receiving the whole training set, in on-line case there is no store and the received observations are processed instantaneously. In the figure y_t and u_t are the transmitted output and input respectively, y_r and u_r are the received output and input respectively and $\hat{y}$ is the estimated output.

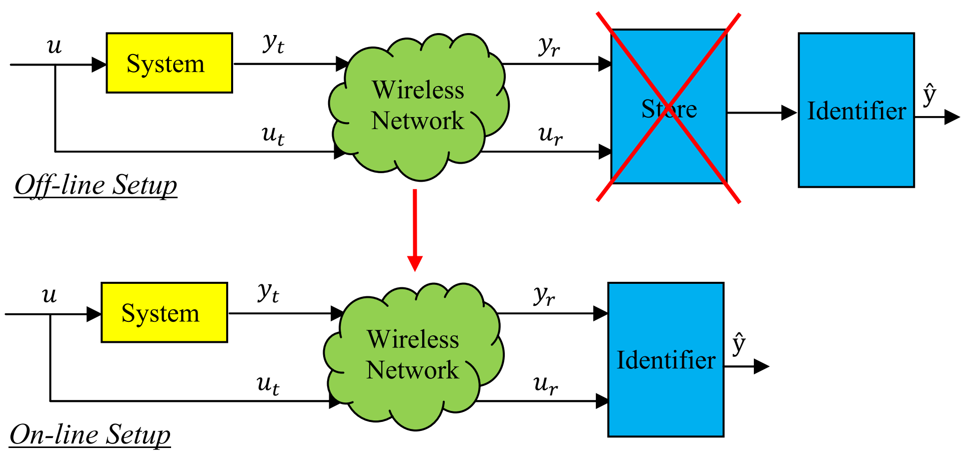


Figure 3.1: The Difference between Off-line and On-line System Identification Setups over Wireless Network

3.3 Least Mean Square (LMS)

The learning rule for LMS [11] is normally formulated as the minimization of the instantaneous mean squared output error (MSE):

$$E(\theta_k) = \frac{1}{2} e_k^2 = \frac{1}{2}(y_k - \hat{y}_k)^2 \tag{3.1}$$

where y_k is the system output and $\hat{y}_k$ is the model estimation at time k. The systems under consideration are linearly dependent on a given set of weights. Thus:

$$\hat{y}_k = \varphi_k^T \theta_{k-1} \tag{3.2}$$

where φ_k is an n-dimensional regressor vector and θ_{k-1} is an N-dimensional weights vector. The weights update rule is given as follows:

$$\theta_k = \theta_{k-1} + \lambda\, s_k \tag{3.3}$$

where λ is the learning rate (normally set to constant) and s_k is an N-dimensional search direction vector. When using the Widrow's LMS method the search direction is set as the negative of the gradient of the instantaneous MSE cost function as follows:

$$s_k = -\frac{\partial}{\partial \theta}\left(E(\theta_k)\right) = e_k \varphi_k \tag{3.4}$$

The search direction in equation (3.4) is actually parallel to the regressor vector. During weights update the actual step taken along the direction of the regressor vector is equal to the learning rate (λ) multiplied by the output error (e_k).

3.4 Observations Loss Effect

Equation (3.3) shows the classical LMS algorithm which assumes periodic weights update where the observations reach the identifier in predefined sampling intervals. Since observations arrival is modeled via Bernoulli process (β_k) during observation loss (see chapter 2), the weights update would depend on Bernoulli process behavior which leads to stochastic weights update as it is shown by the following equation:

$$\theta_k = \theta_{k-1} + \beta_k \, \lambda \, s_k \tag{3.5}$$

Test problem I (refer to appendix A) used to show the LMS performance at different settings of packets dropping probability (PDP). The green curve in Figure 3.2 shows how the error minimizes until it settles at relatively small value in absence of observations loss while the other curves show that as PDP increases the error increases.

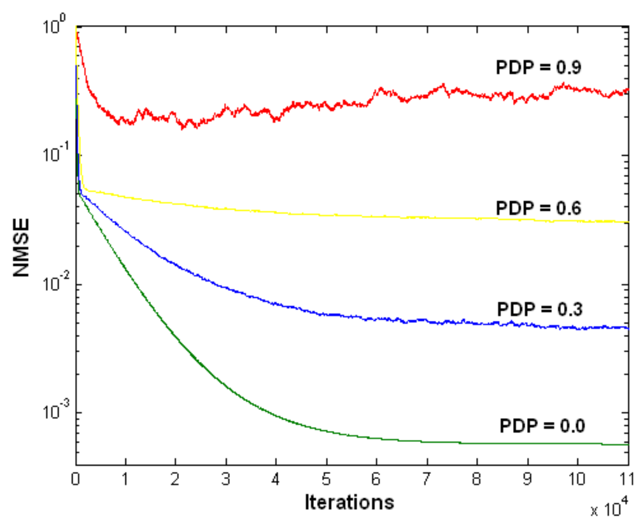


Figure 3.2: LMS Performance at Different Settings of PDP

3.5 Sampling Process Role

In system identification process system is modeled from samples of its information, such sampling process leads to information losses and therefore sampling interval should be chosen carefully to make these losses insignificant. In normal conditions the rule of thumb is to select the sampling interval so that it gives 4 to 6 samples during the rise time of the system step response [12]. When considering wireless network as a transfer medium, the observations availability at the identifier side becomes hard to guarantee. Thus, the rule of thumb mentioned above can't be applied during observations loss.

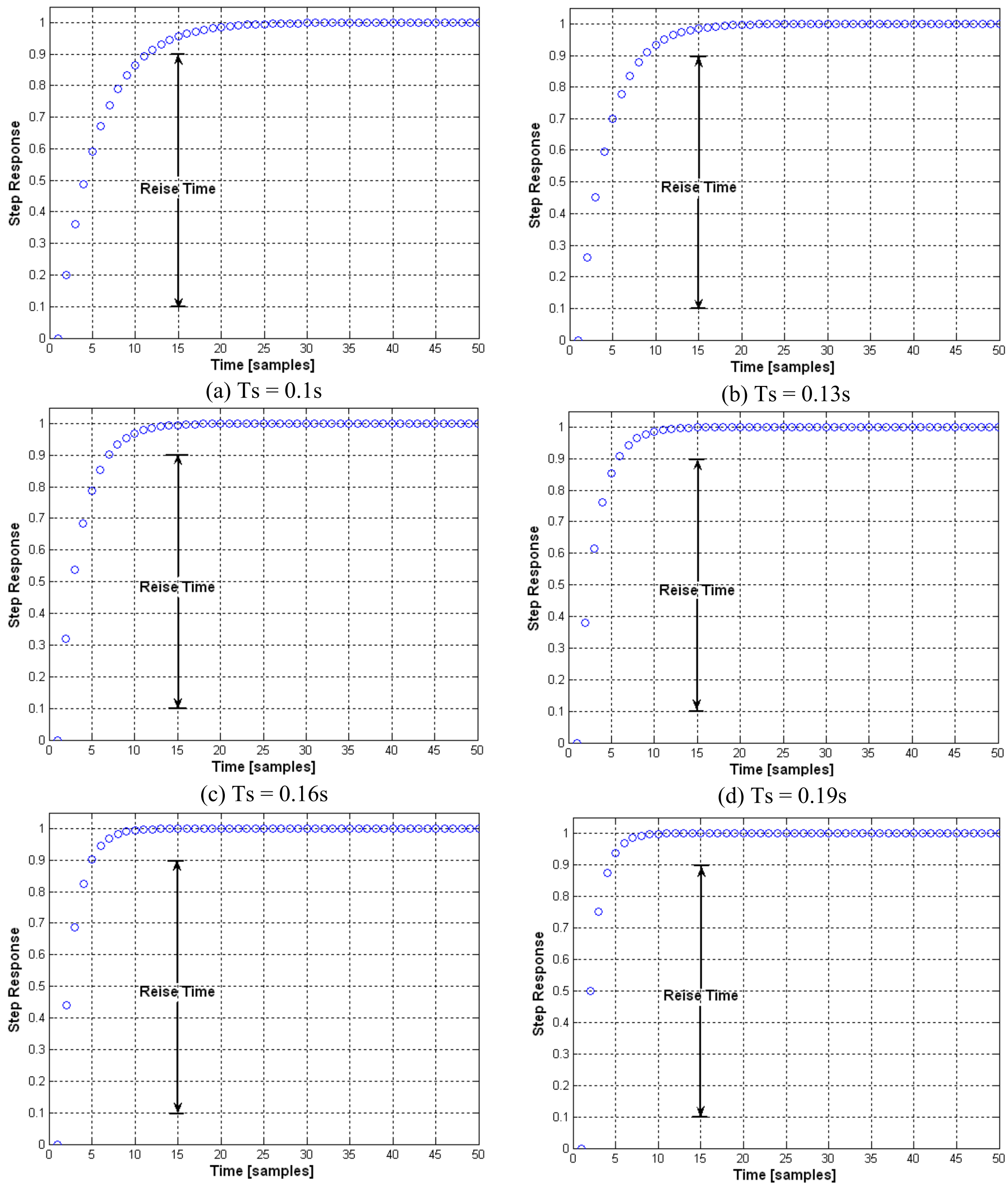
The rule adopted in this research work for wireless system identification is to oversample the system in order to compensate the lost observations during wireless transmission. Next section looks into an example shows how oversampling improves the performance of LMS algorithm during observations loss.

3.5.1 Example of Oversampling to Aid Modeling Process

In this example the step response of the system in test problem I (refer to appendix A) is plotted at different settings of sampling time (Ts), then the system is modeled at the recommended range of sampling time according to the rule of thumb and at another smaller sampling time (oversampling) to show the performance improvement.

Figure 3.3 depicts the step response of the system at different settings of Ts, according to the rule of thumb it can be seen from the graphs 3.3(a) and 3.3(b) that Ts = 0.1s and Ts = 0.13s are disqualified since they give 10 and 7 samples respectively during the rise time, Ts = 0.25s in graph 3.3(f) is also disqualified since it gives 3 samples during the rise time.

In normal conditions sampling time choice would be either of those in the graphs 3.3(c), 3.3(d) and 3.3(e) since they give 4 to 6 samples during the rise time. In the next part of this example the sampling time Ts = 0.13s will be considered in addition to those which fulfill the rule of thumb to sample the system for modeling purpose.



(a) Ts = 0.1s

(b) Ts = 0.13s

(c) Ts = 0.16s

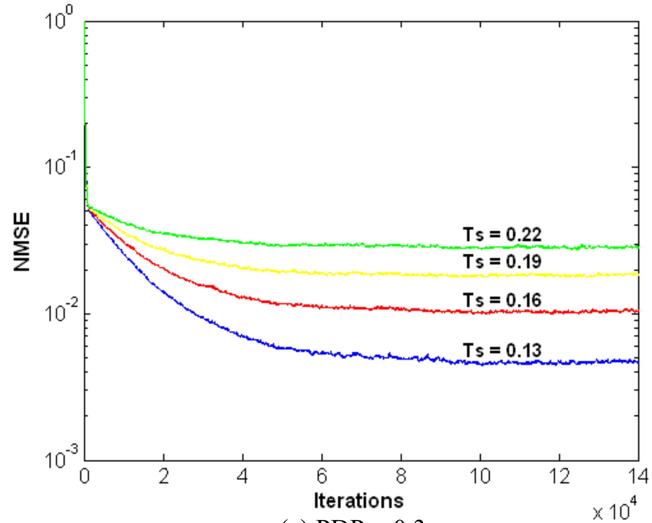
(d) Ts = 0.19s

(e) Ts = 0.22s

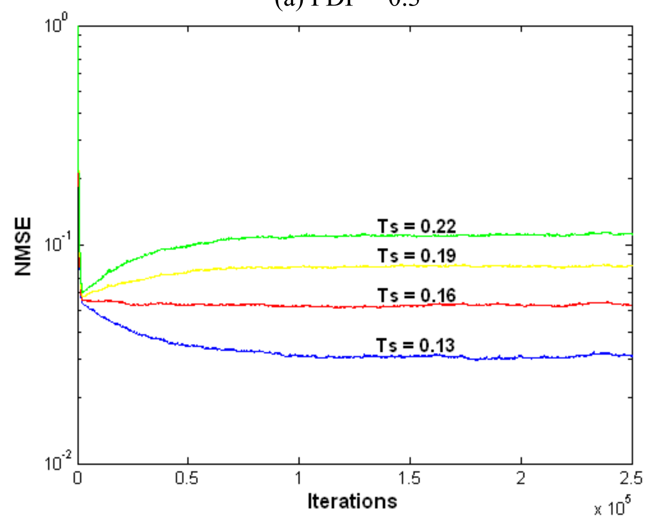
(f) Ts = 0.25s

Figure 3.3: Step Response of Test Problem I at Different Settings of Ts

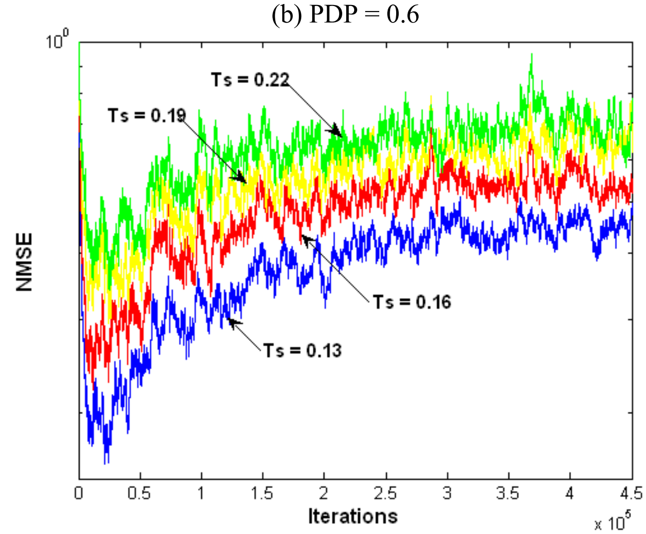
Figure 3.4 shows the LMS performance at different settings of PDP when sampling the system at Ts = 0.13s, Ts = 0.16s, Ts = 0.19s and Ts = 0.22s. It can be seen from the figure that the smallest error is at Ts = 0.13s for all the different settings of PDP.



(a) PDP = 0.3



(b) PDP = 0.6



(c) PDP = 0.9

Figure 3.4: LMS Performance at Different Settings of Ts and PDP

Both average and standard deviation of the instantaneous normalized mean square error (NMSE) over last 20000 iterations (refer to appendix A) have been recorded in Table 3.1 to evaluate the performance of LMS at different settings of Ts and PDP.

Table 3.1: Numerical Results for LMS Performance at Different Settings of Ts and PDP

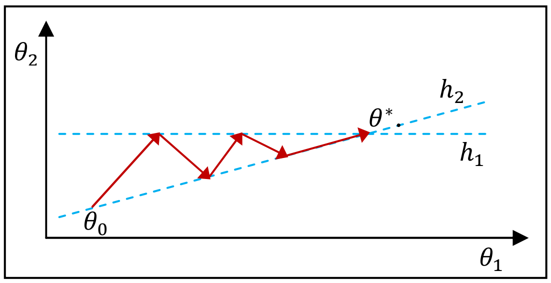
Ts \ PDP	Measure	0.3	0.6	0.9
0.13	Average	0.0047	0.0314	0.4496
	Standard Deviation	7.7480e-005	3.1168e-004	0.0136
0.16	Average	0.0104	0.0537	0.5268
	Standard Deviation	1.5060e-004	5.0707e-004	0.0160
0.19	Average	0.0184	0.0798	0.6474
	Standard Deviation	2.4294e-004	4.6440e-004	0.0325
0.22	Average	0.0284	0.1122	0.6809
	Standard Deviation	3.3444e-004	7.8923e-004	0.0505

3.6 Correlation Effect

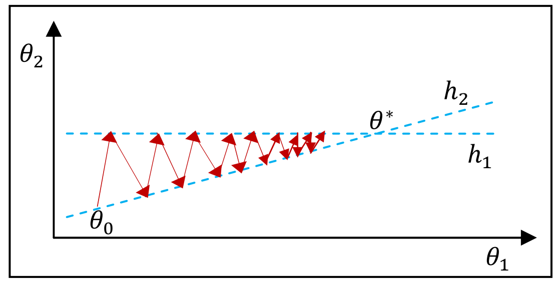
Observations loss across wireless network is of random nature which means at any time instant transmitted observations might be received or lost. In presence of observations loss oversampling helps to compensate the lost observations, but in contrary in absence of observations loss oversampling leads to correlated observations at the identifier side [13].

Since the performance of LMS is strongly dependent on the nature and quality of the presented training data it is rare for the weights to converge to the optimal weights vector (θ^*) when the presented data are correlated. Instead the weights normally drift around an area known as *minimal capture zone* rather than to their desired values [13] [14].

Figure 3.5(a) shows the weights trajectory for near orthogonal training data where the weights vector converges directly towards θ^*, graph 3.5(b) shows correlated training data where the weights vector drifts around *minimal capture zone* domain. The term h_i in the figure is an (N-1)-dimensional solution hyperplane in N-dimensional weights space and the intersection of the two hyperplanes lies θ^*.



(a) Weights Vector Converges to Optimal Value



(b) Weights Vector Converges to Minimal Capture Zone

Figure 3.5: Weights Trajectory for a Given Training Data

Correlation level between observations can be measured by computing the correlation angle (A_{corr}) between the current regressor vector (φ_c) and the previous one (φ_p), as it is demonstrated by equation (3.6) and Figure 3.6 below:

$$A_{corr} = \left| cos^{-1}\left(\frac{{\varphi_c}^T \varphi_p}{\|\varphi_c\| \, \|\varphi_p\|}\right) \right| \tag{3.6}$$

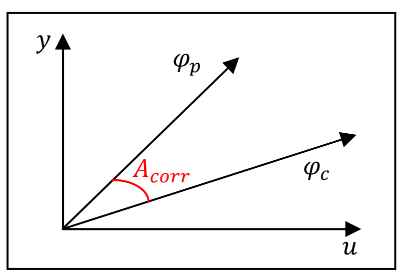
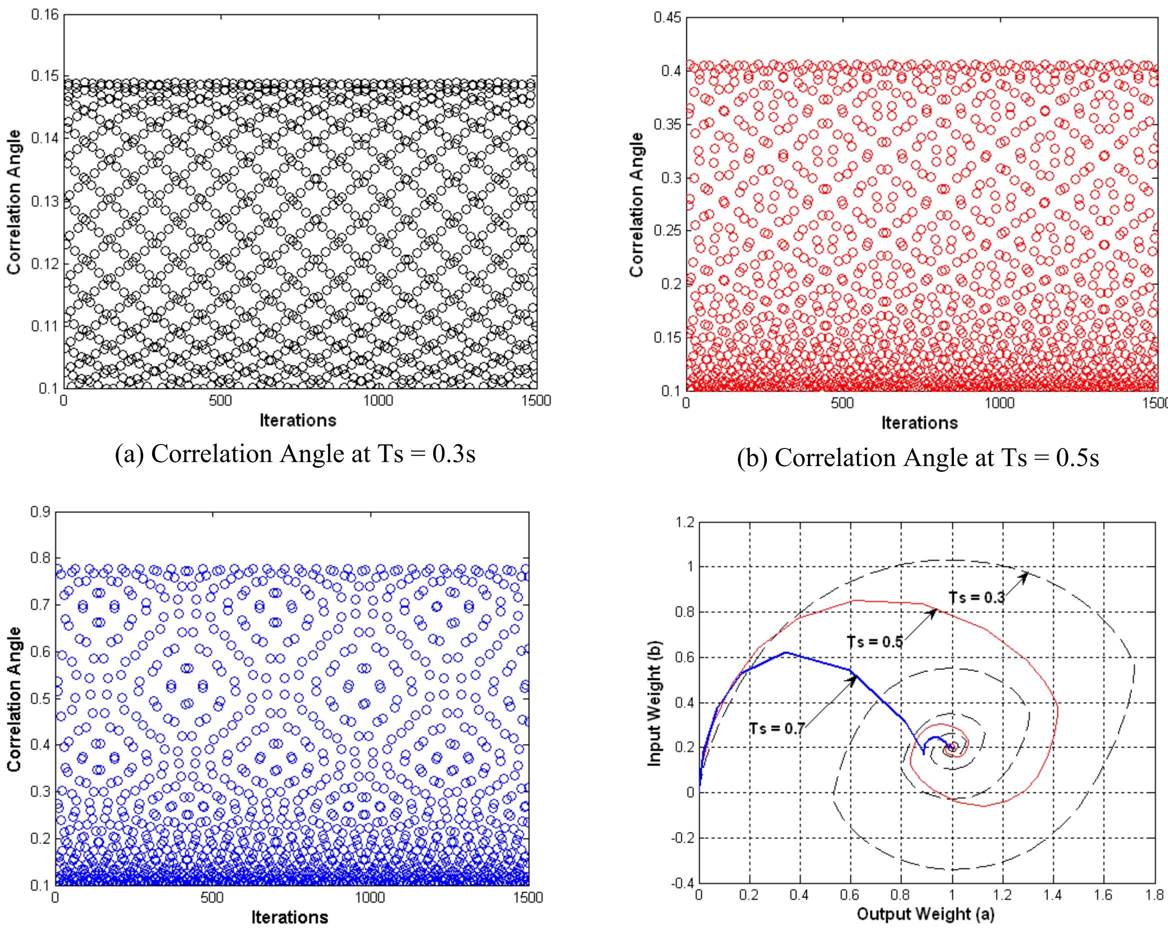


Figure 3.6: Correlation Angle between the Current and Previous Regressor Vectors

The value of A_{corr} is confined between 0 and $\pi/2$ by subtracting it from π when it is greater than $\pi/2$. The lower the correlation angle the greater the correlation level and $\pi/2$ represents orthogonal vectors while 0 represents completely correlated vectors.

Test problem II-A (refer to Appendix A) used to show the relation between each of the sampling time, correlation angle and weights convergence. The desired weights vector for the test problem is $\theta^* = [1,0.2]$. Figure 3.7 shows the instantaneous correlation angle and the weights trajectory at different settings of sampling time.



(a) Correlation Angle at Ts = 0.3s

(b) Correlation Angle at Ts = 0.5s

(c) Correlation Angle at Ts = 0.7s

(d) Weights Trajectory

Figure 3.7: Correlation Angle and Weights Trajectory at Different Settings of Ts

The above figure reveals how as the system sampled slowly the correlation angle increases and the weights attain faster to their optimal values. Graph 3.7(a) shows that at Ts = 0.3s the largest correlation angle is equal to 0.15rad while graph 3.7(b) shows that at Ts = 0.5s the largest correlation angle is equal to 0.4rad and finally graph 3.7(c) shows that at Ts = 0.7s the largest correlation angle is equal to 0.8rad, so as Ts increases (slow sampling) correlation angle increases. Now by looking at graph 3.7(d) it can be seen that at Ts = 0.3s the weights vector tends to drift around and finally converges slowly to θ^* while at Ts = 0.5s the weights vector takes shorter path to θ^* and finally at Ts = 0.7s the weights vector takes the shortest path among the others between the initial value and θ^*.

It can be concluded that oversampling leads to correlated observations at the identifier side when consecutive observations reach without loss which causes the learning algorithm to function sub-optimally. So there is a need to treat the correlated observations instantaneously at the identifier side as long as oversampling is important to compensate the lost observations. Next section looks into an instantaneous technique adopted on LMS algorithm in order to treat the correlation effect which will be likely to happen when wireless transmission is good.

3.7 Sine Function Based De-Correlation (SD)

Correlation effects can be treated during weights update by adapting the update step length taken parallel to the observations direction with the correlation level between the observations. Typically the update step length should be small as observations tend to be correlated and large as they tend to be orthogonal. Since the learning rate (λ) is one of the main factors that control the update step length (see equation (3.3)), it should be adaptive in nature with the correlation level between the observations.

This section proposes an instantaneous technique to treat the correlation effect at the identifier side known as sine function based de-correlation (SD). The SD technique in short sets the learning rate to the sine of the correlation angle and since the correlation angle ranges between 0 and $\pi/2$, the learning rate will be changing only in the range between 0 and 1. Thus:

$$\lambda_{sin} = \sin(A_{corr}) \tag{3.7}$$

Equation (3.7) shows that when observations are completely correlated ($A_{corr} = 0$) weights will be not updated as λ_{sin} will be equal to 0 while the largest step size will be taken when observations are orthogonal ($A_{corr} = \pi/2$) as λ_{sin} will be equal to 1.

Test problem II-A (refer to appendix A) used to show the weights trajectory of the traditional LMS and LMS with the adaptive learning rate for correlated training data. The desired weights vector for the test problem is $\theta^* = [1,0.2]$. Figure 3.8 shows that LMS with the adaptive learning rate takes short path to the optimal values while the traditional LMS tends to drift around and finally converges very slowly.

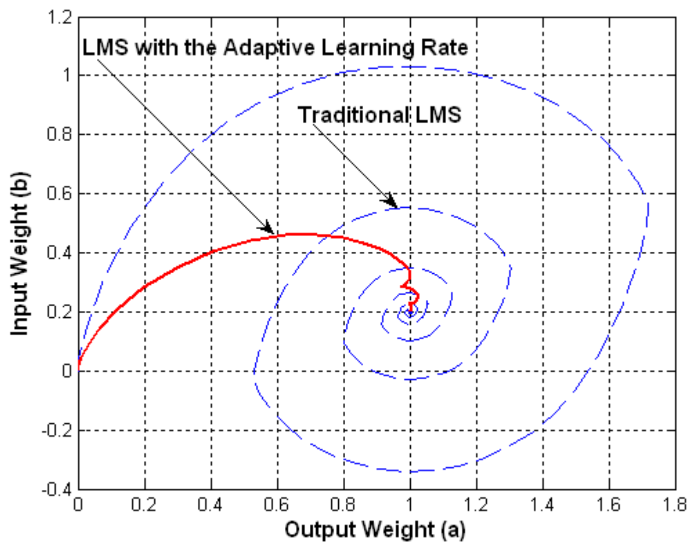


Figure 3.8: Weights Trajectory of the Traditional LMS and LMS with the Adaptive Learning Rate for Correlated Training Data

In absence of observations loss the received observations will tend to be correlated because of oversampling. Hence, the proposed adaptive learning rate (λ_{sin}) will be used to reduce the correlation effect. In presence of observations loss the loss itself will influence the sampling intervals and therefore the received observations will tend to be uncorrelated. Moreover, during observations loss the received observations will be separated in random intervals and therefore they will be not informative. Hence, instead of λ_{sin} a small constant learning rate will be considered in presence of observations loss.

The flowchart in Figure 3.9 illustrates the procedure of LMS-SD algorithm, where c is a small constant learning rate and F_l is a loss flag indicates loss event when it equals 1 and it is initially set to 0. The procedure starts by checking whether the current packet has been lost or not, where in case of loss F_l is set to 1 and the procedure starts over to check for the next packet otherwise the regressor vector (φ_k) is formed then F_l is checked to determine whether the previous packet has been lost or not, where in case of loss the small constant learning rate (c) is considered otherwise the adaptive learning rate (λ_{sin}) is considered. Once the learning rate has been determined the procedure computes the search direction (s_k) and proceeds to update the weights (θ_k) and compute the output estimation ($\hat{y}_k$). The procedure then checks whether the obtained model is accepted or not, where in case of acceptance the learning process ends otherwise F_l is set to 0 since the current packet has been received successfully and the procedure starts over.

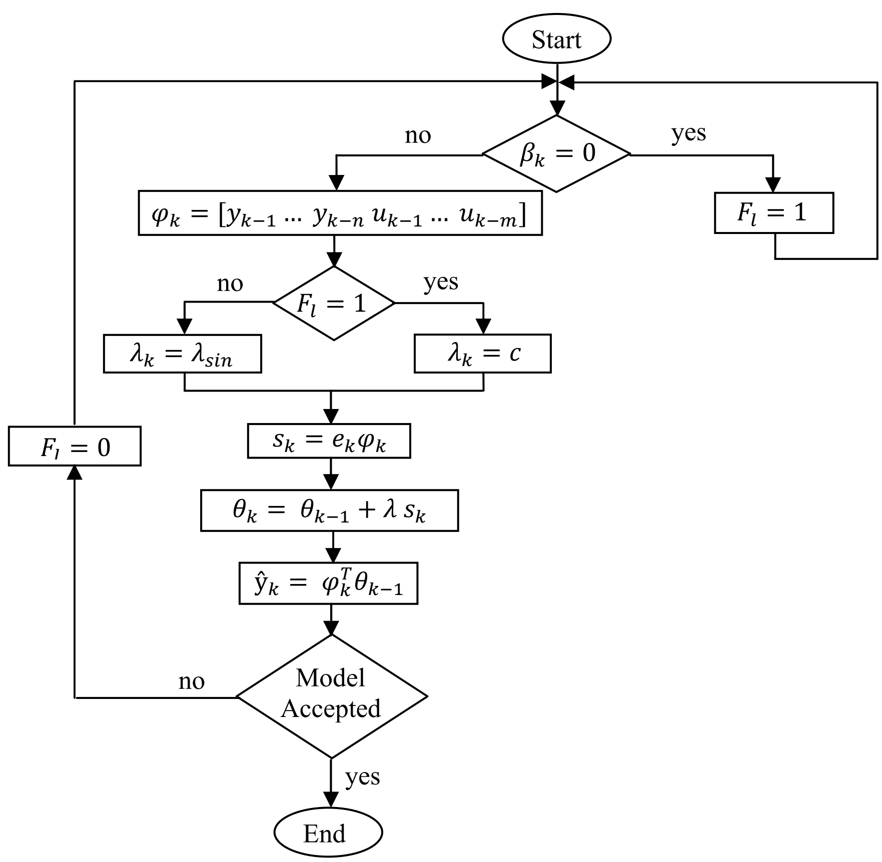
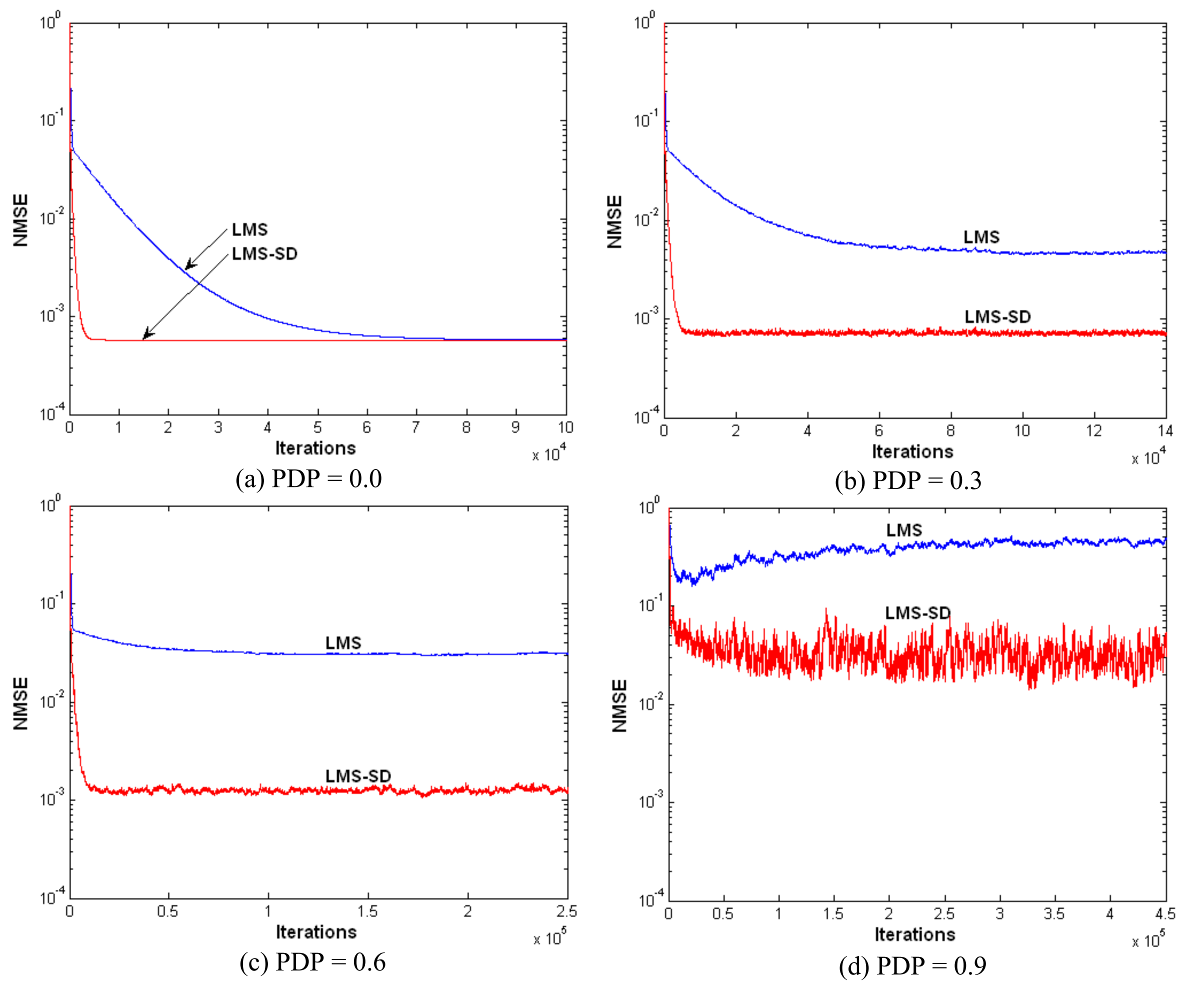


Figure 3.9: LMS-SD Algorithm Flowchart

Figure 3.10 shows the performance of LMS-SD in comparison with the traditional LMS at different settings of PDP. Graph 3.10(a) shows the algorithms performance in absence of loss where LMS-SD converges faster due to the adaptive learning rate. The rest of the graphs (3.10(b) – 3.10(d)) show that as PDP increases the performance degrades in general but LMS-SD performs better compared to the traditional LMS.

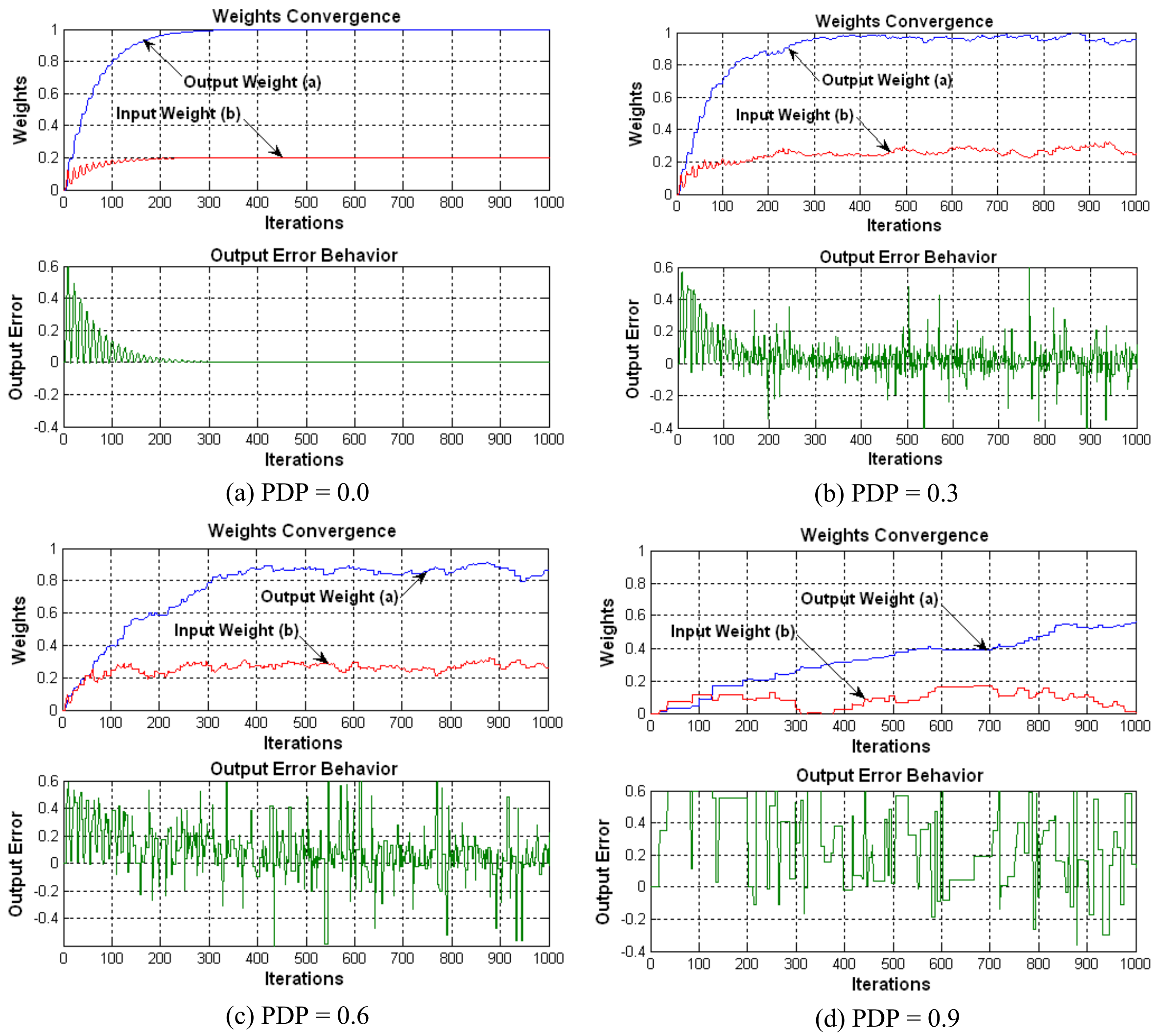


(a) PDP = 0.0 (b) PDP = 0.3 (c) PDP = 0.6 (d) PDP = 0.9

Figure 3.10: LMS-SD and the Traditional LMS Performance at Different Settings of PDP

3.8 Error Displacement based Update (EDU)

Output error component is one of the influential factors in the weights convergence process since it directs the weights update and contributes in the update step length (see equations (3.3) and (3.4)). When observations are fully received the output error minimizes until it becomes equal to 0 as the weights attain their optimal values but during observations loss the output error contributes with high magnitude in the search direction (s_k) which leads to significant changes in the weights update direction. Test problem II-A (refer to appendix A) used to show the weights and the output error behavior under loss and loss free situations. The desired weights vector for the test problem is $\theta^* = [1,0.2]$. Graph 3.11(a) shows that in absence of loss the output error minimizes until it becomes equal to 0 as the weights attain their optimal values. From the graphs (3.11(b) – 3.11(d)) the output error seems to be not directing the weights to their optimal values and contributing with high magnitude in the weights update which destroys the learning process.



(a) PDP = 0.0

(b) PDP = 0.3

(c) PDP = 0.6

(d) PDP = 0.9

Figure 3.11: The Weights and the Output Error Behavior under Loss and Loss Free Situations

This section proposes new technique known as error displacement based update (EDU) to reduce the undesired contribution of the output error in the weights update process. The technique relates the weights update to the error displacement (*Ed*) and stops temporary the weights update when *Ed* is greater than a predefined threshold (σ) (which is obtained by trail and error method) as it is shown by the following steps:

Step 1: Computing the error displacement value:

$$Ed_k = \left|\frac{y_k - \hat{y}_k}{y_k}\right| \tag{3.8}$$

Step 2: Checking whether the error displacement is greater than the threshold:

$$Ed_k > \sigma \tag{3.9}$$

Step 3: If the error displacement was greater than the threshold weights are frozen:

$$\theta_k = \theta_{k-1} \tag{3.10}$$

Otherwise weights are updated:

$$\theta_k = \theta_{k-1} + \lambda\, s_k \tag{3.11}$$

Robust algorithm for LMS known as LMS-SDEDU is proposed by adopting a combination of SD and EDU techniques on LMS algorithm. The flowchart in Figure 3.12 illustrates the procedure of LMS-SDEDU algorithm. The procedure works similar to LMS-SD until it reaches the weights update stage, at this point the procedure computes the error displacement value (Ed_k) and based on the comparison with the threshold (σ) it decides whether the weights (θ_k) will be updated or frozen. The procedure then computes the output estimation ($\hat{y}_k$) and checks whether the obtained model is accepted or not, where in case of acceptance the learning process ends otherwise F_l is set to 0 since the current packet has been received successfully and the procedure starts over.

Figure 3.13 shows the performance of LMS-SDEDU in comparison with the traditional LMS at different settings of PDP. Graph 3.13(a) shows that in absence of loss (PDP = 0) LMS-SDEDU converges faster due to SD technique. The rest of the graphs (3.13(b) – 3.13(d)) show the outperformance of LMS-SDEDU as PDP increases since EDU reduces the undesired contribution of output error in weights update process.

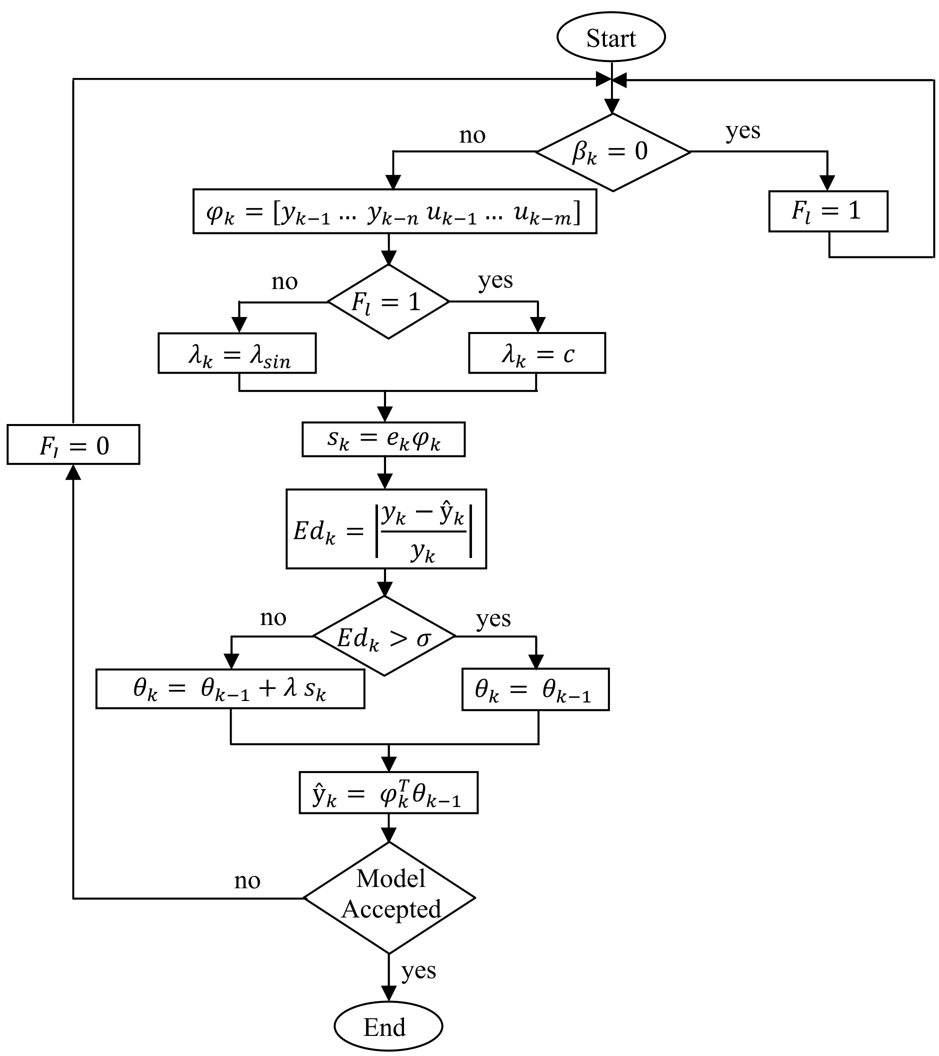


Figure 3.12: LMS-SDEDU Algorithm Flowchart

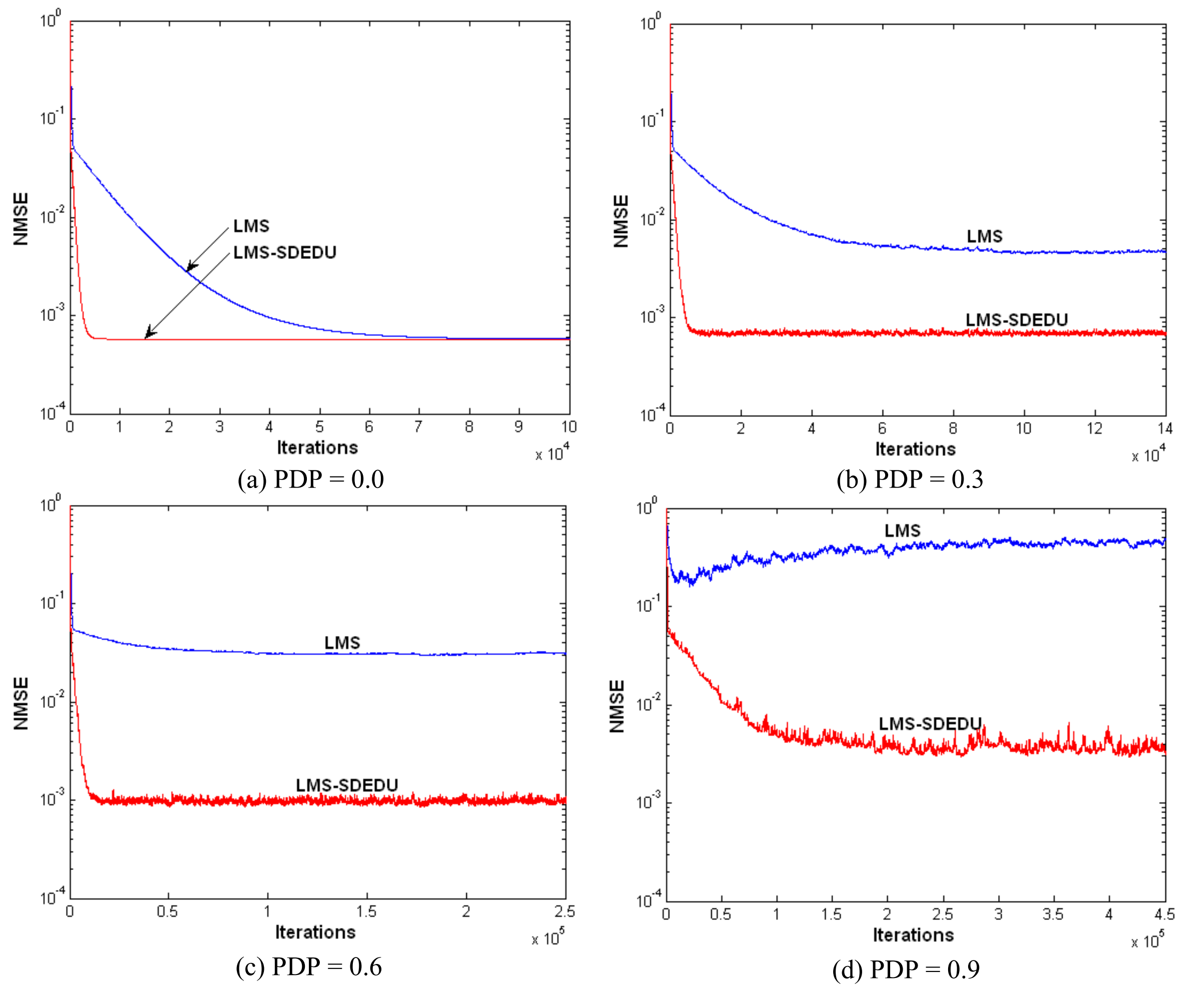


Figure 3.13: LMS-SDEDU and the Traditional LMS Performance at Different Settings of PDP

3.9 Normalized Least Mean Square (NLMS)

One disadvantage of LMS algorithm is that the reduction in the output error (e_k) depends on the size of the regressor vector (φ_k) and therefore NLMS has been introduced in literature to remedy this disadvantage. This can be shown [11] by driving the relationship between e_k and the *posteriori* output error ($\bar{e}_k$) which obtained after updating the weights as it is shown in equation (3.3). Thus:

$$\bar{e}_k = y_k - \varphi_k^T \theta_k \tag{3.12}$$

$$\bar{e}_k = y_k - \varphi_k^T (\theta_{k-1} + \lambda s_k) \tag{3.13}$$

$$\bar{e}_k = y_k - \varphi_k^T (\theta_{k-1} + \lambda e_k \varphi_k) \tag{3.14}$$

$$\bar{e}_k = y_k - \varphi_k^T \theta_{k-1} - \lambda e_k \|\varphi_k\|_2^2 \tag{3.15}$$

$$\bar{e}_k = e_k - \lambda e_k \|\varphi_k\|_2^2 \tag{3.16}$$

$$\bar{e}_k = (1 - \lambda \|\varphi_k\|_2^2) e_k \tag{3.17}$$

A stable learning rate for LMS should satisfy:

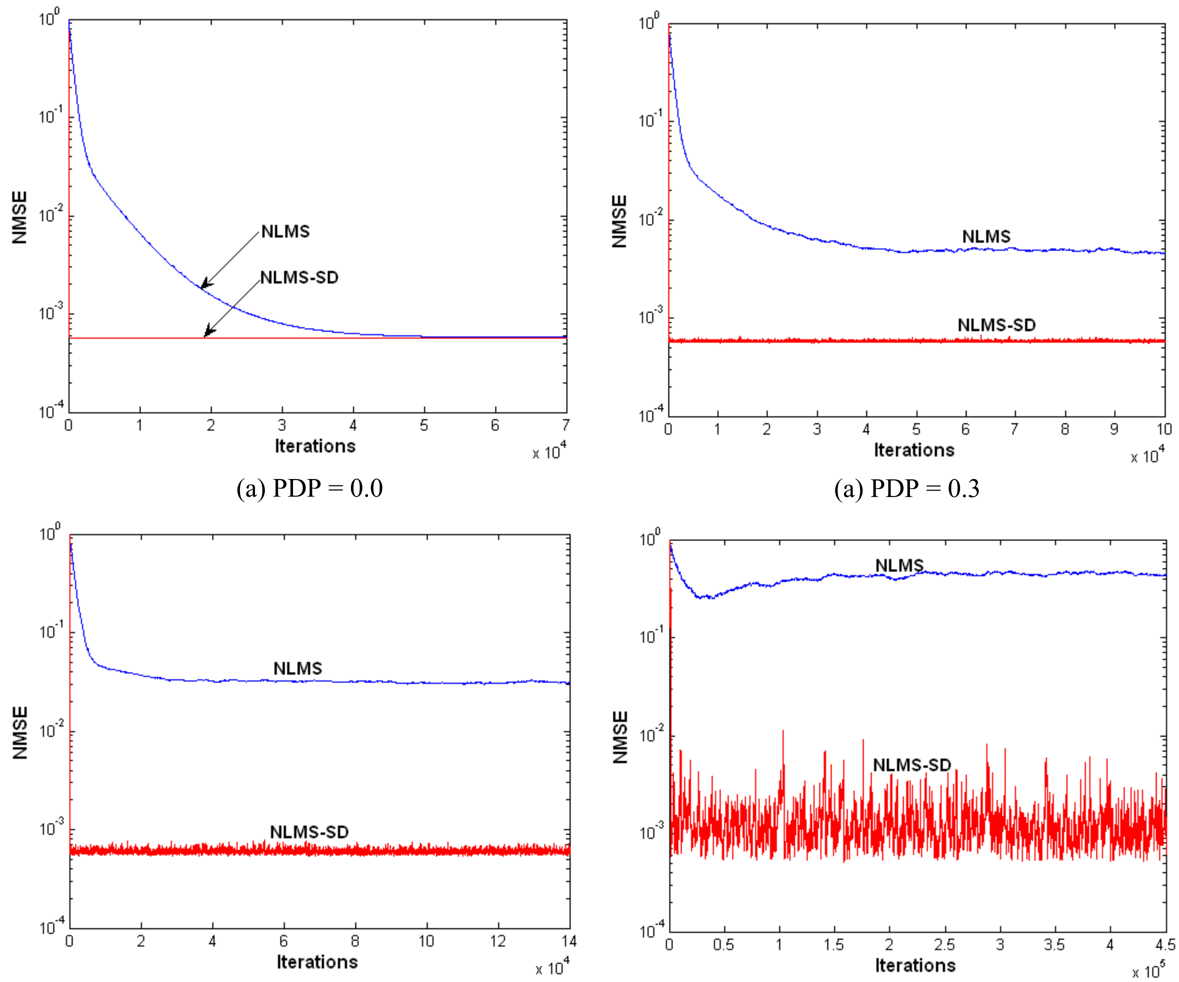
$$0 < \lambda < \frac{2}{\|\varphi_k\|_2^2} \; for \; |\bar{e}_k| < |e_k| \tag{3.18}$$

This motivates the development of NLMS search direction which is given as follows:

$$s_k = \frac{e_k \varphi_k}{\|\varphi_k\|_2^2} \tag{3.19}$$

Since the weights update is normalized, the convergence of NLMS doesn't depend on the regressor vector size.

Both of SD technique and SDEDU method are applicable to NLMS; the only difference is that the search direction will be computed according to equation (3.19) instead of equation (3.4). Figure 3.14 shows the performance of NLMS-SD in comparison with the traditional NLMS at different settings of PDP. From the figure it can be seen that NLMS-SD shows better performance and good capability of handling observations loss effect.



(a) PDP = 0.0

(a) PDP = 0.3

(c) PDP = 0.6

(d) PDP = 0.9

Figure 3.14: NLMS-SD and the Traditional NLMS Performance at Different Settings of PDP

Figure 3.15 shows the performance of NLMS-SDEDU in comparison with the traditional NLMS at different settings of PDP. The figure shows that NLMS-SDEDU outperforms the traditional NLMS in all the different settings of PDP.

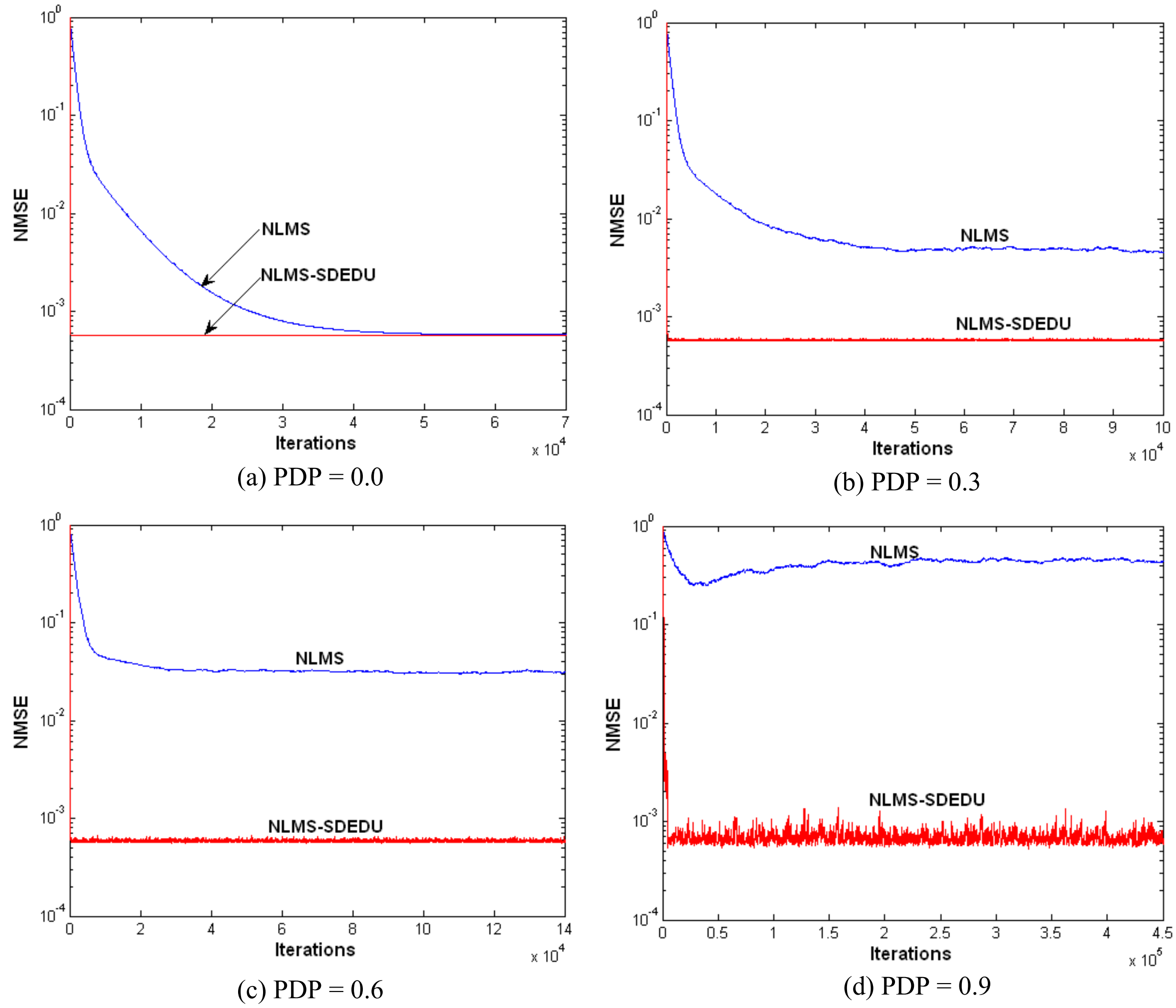


Figure 3.15: NLMS-SDEDU and the Traditional NLMS Performance at Different Settings of PDP

Table 3.2 summarizes the performance of the optimized algorithms that have been proposed in this chapter in comparison with their original versions. Both average and standard deviation of the instantaneous normalized mean square error (NMSE) over last 20000 iterations (refer to appendix A) have been recorded in the table. The results show that the optimized algorithms perform better than their original versions during observations loss. It can also be seen from the results that the optimized algorithms based on SDEDU method performs better than the optimized algorithms based on SD technique.

Table 3.2: Numerical Results for the First Order Optimized and Original Algorithms Performance at Different Settings of PDP

PDP / Algorithm	Measure	0.0	0.3	0.6	0.9
LMS	Average	5.8072e-004	0.0047	0.0314	0.4496
	Standard Deviation	2.5140e-006	7.7480e-005	3.1168e-004	0.0136
LMS-SD	Average	**5.7551e-004**	**7.1926e-004**	**0.0013**	**0.0333**
	Standard Deviation	**2.1832e-016**	**1.9043e-005**	**5.4189e-005**	**0.0092**
LMS-SDEDU	Average	**5.7551e-004**	**6.9319e-004**	**9.8657e-004**	**0.0037**
	Standard Deviation	**2.1832e-016**	**1.8433e-005**	**4.7552e-005**	**3.2900e-004**
NLMS	Average	5.8089e-004	0.0048	0.0311	0.4518
	Standard Deviation	3.8430e-006	1.6100e-004	5.9911e-004	0.0127
NLMS-SD	Average	**5.7551e-004**	**5.8328e-004**	**6.0607e-004**	**0.0011**
	Standard Deviation	**1.6695e-016**	**8.1490e-006**	**2.5888e-005**	**5.2776e-004**
NLMS-SDEDU	Average	**5.7551e-004**	**5.7970e-004**	**5.7983e-004**	**6.9815e-004**
	Standard Deviation	**1.6695e-016**	**4.6835e-006**	**4.4862e-006**	**9.9496e-005**

3.10 Conclusion

This chapter has investigated the performance of the well known LMS algorithm and its normalized version (NLMS) when observations are sent to the identifier over lossy links. The chapter has showed the difference between the off-line and on-line system identification setups over wireless network then it has proceeded to address the role of the sampling process where it has showed that oversampling mitigates the effect of random observations loss. Oversampling could correlate the observations at the identifier side (when wireless link is strong) which causes the algorithm to function sub-optimally, therefore sine based de-correlation (SD) technique has been introduced in this chapter to treat the correlation effect instantaneously at the identifier side. By adopting SD technique on LMS algorithm a new version of LMS known as LMS-SD has been proposed. The performance of LMS-SD has been evaluated in comparison with the traditional LMS at different settings of PDP; the comparison has showed that LMS-SD performs better than the traditional LMS.

During observations loss the output error contributes with high magnitude in the search direction which causes significant changes in the weights update direction and destroys the learning process, therefore error displacement based update (EDU) technique has been introduced in this chapter to reduce the undesired contribution of the output error in the weights update process. By adopting a combination of SD and EDU techniques on LMS another version of LMS known as LMS-SDEDU has been proposed. LMS-SDEDU sets the learning rate according to SD technique and updates the weights according to EDU technique. The performance of LMS-SDEDU has been evaluated in comparison with the traditional LMS at different settings of PDP; the comparison has showed that LMS-SDEDU is capable of reducing the effect of random observations loss and improving the performance.

Finally both of SD technique and SDEDU method have been applied to NLMS. The performances of NLMS-SD and NLMS-SDEDU have been evaluated in comparison with the traditional NLMS at different settings of PDP; the comparison has showed the superior performance for each of NLMS-SD and NLMS-SDEDU compared to the traditional NLMS.

Both of LMS and NLMS are based on first order weights update as there is no prior knowledge used for weights update. Next chapter will look into the second order algorithms which use prior knowledge for their weights update in form of covariance matrix.

CHAPTER FOUR: SECOND ORDER INSTANTANOUS LEARNING ALGORITHMS

4.1 Introduction

There are various forms of off-line algorithms reported in literature for both linear and non-linear systems, the drawback of these algorithms is that their formulation is not suitable for on-line or real time applications such as supervised learning, tracking of time varying parameters and time series prediction. Therefore, off-line algorithms are reformulated to on-line algorithms in order to be applicable to on-line applications [11], two examples of this reformulation are the recursive least squares (RLS) and the recursive instrumental variable (RIV).

This chapter investigates the performance of RLS and RIV during random observations loss. The chapter starts by reviewing the reformulation of the off-line least squares (LS) to the on-line RLS then it highlights the capability of RLS in dealing with the correlated observations in contrast with the first order algorithms such as LMS and NLMS. The chapter then proceeds to discuss the observations discontinuity effect on the structure of the covariance matrix (P) and shows how this discontinuity in the covariance matrix structure deteriorates the performance of the algorithm. Thus, the chapter proposes new algorithm to improve the performance using simple update skipping technique.

The chapter also reviews the reformulation of the off-line instrumental variable (IV) to the on-line RIV then it applies the proposed update skipping technique to RIV since it shares with RLS the same weights update principles.

4.2 Recursive Least Squares (RLS)

RLS algorithm [15] [16] can be derived from the ordinary off-line least squares according to the following derivation starting with the weights estimation of the ordinary least squares which is given as:

$$\theta_k = \left(\sum_{i=1}^{k} \varphi_i \varphi_i^T\right)^{-1} \left(\sum_{i=1}^{k} \varphi_i y_i\right) \tag{4.1}$$

where φ_i is the regressor vector and y_i is the system output. Let:

$$P_k = \left(\sum_{i=1}^{k} \varphi_i \varphi_i^T\right)^{-1} \tag{4.2}$$

where P_k is the current covariance matrix. The previous covariance matrix P_{k-1} is given as:

$$P_{k-1} = \left(\sum_{i=1}^{k-1} \varphi_i \varphi_i^T\right)^{-1} \tag{4.3}$$

The recursive covariance matrix update is then given as:

$$P_k = P_{k-1} + (\varphi_k \varphi_k^T)^{-1} \tag{4.4}$$

$$P_k = (P_{k-1}^{-1} + \varphi_k \varphi_k^T)^{-1} \tag{4.5}$$

Using the matrix inversion lemma:

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1} \tag{4.6}$$

$$P_k = (P_{k-1}^{-1} + \varphi_k \varphi_k^T)^{-1} = P_{k-1} - P_{k-1}\varphi_k(I + \varphi_k^T P_{k-1}\varphi_k)^{-1}\varphi_k^T P_{k-1} \tag{4.7}$$

Hence, RLS algorithm can be expressed as:

$$\theta_k = \theta_{k-1} + P_k \varphi_k e_k \tag{4.8}$$

$$e_k = y_k - \varphi_k^T \theta_{k-1} \tag{4.9}$$

$$P_k = P_{k-1} - \frac{P_{k-1}\varphi_k \varphi_k^T P_{k-1}}{1 + \varphi_k^T P_{k-1}\varphi_k} \tag{4.10}$$

where P_o is initially set to an identity matrix (I_N).

The recursive least squares estimation gives equal weights to the old and new observations vectors by accumulating observations vectors information into its covariance matrix. In real time identification it is normal to suppress old observations vectors in order to track the weight which is time varying. This can be achieved by introducing the forgetting factor (γ) in the covariance matrix update as follows:

$$P_k = \frac{1}{\gamma}\left(P_{k-1} - \frac{P_{k-1}\varphi_k \varphi_k^T P_{k-1}}{\gamma + \varphi_k^T P_{k-1}\varphi_k}\right) \quad where \;\; 0 < \gamma \leq 1 \tag{4.11}$$

As the value of γ decreases the speed of weights convergence increases but the estimation becomes more sensitive towards noise and when the value of γ increases the speed of the weights convergence decreases but the estimation becomes robust towards noise. Hence, the choice of the forgetting factor is a trade off between the weights convergence speed and the estimation sensitivity towards noise.

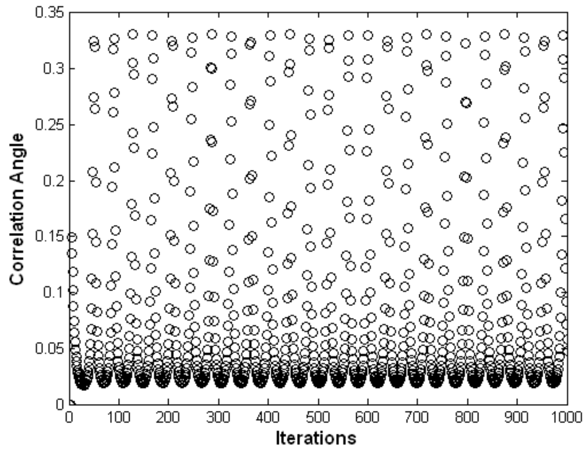
4.3 RLS and Correlated Observations

When the received observations are highly correlated RLS performs better than LMS and NLMS [17]. This can be attributed to the fact that RLS does not depend only on the current presented observations as it is the case in LMS and NLMS, instead it employs information from old observations which are accumulated in the covariance matrix.

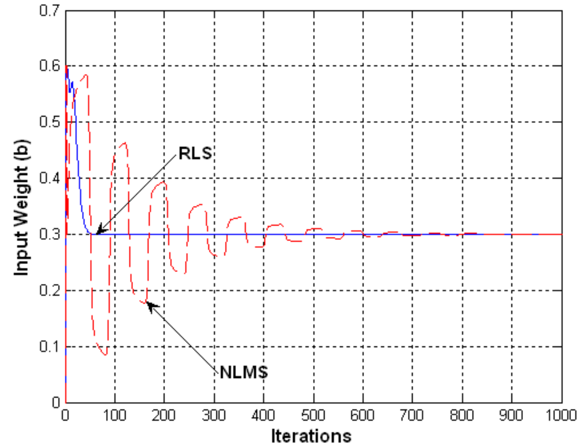
Test problem II-B (refer to appendix A) used to show the capability of RLS in dealing with highly correlated observations in comparison with each of LMS and NLMS. The desired weights vector for the test problem is $\theta^* = [0.9, 0.3]$. Figure 4.1 shows the instantaneous correlation angle and the weight convergence of the three algorithms.

Graph 4.1(a) shows the instantaneous correlation angle between the subsequence observations in which the largest correlation angle is shown to be 0.35rad (correlated observations). Graph 4.1(b) summarizes the performance of RLS and NLMS, from the graph it can be seen that NLMS oscillates around and converges very slowly to the single optimal weight while RLS converges directly to it. Finally graph 4.1(c) shows clearly that LMS diverges with correlated observation and becomes not able to converge to the optimal weight at all.

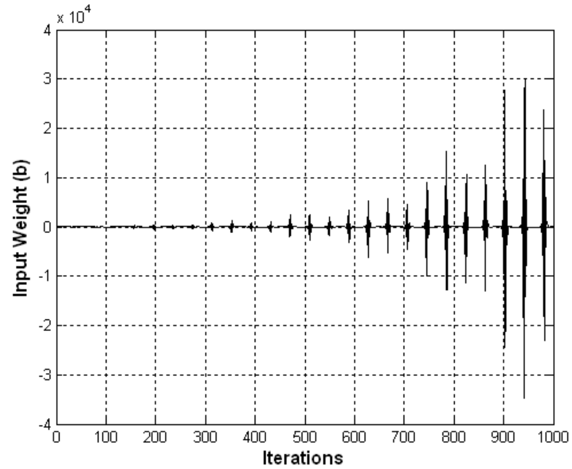
Figure 4.2 shows the weights trajectory of RLS, NLMS and LMS for the same test problem mentioned above. Graph 4.2(a) shows that RLS takes short path to the optimal weights while NLMS drifts around and converges very slowly. Graph 4.2(b) shows that LMS doesn’t converge to the optimal weights at all.



(a) Correlation Angle

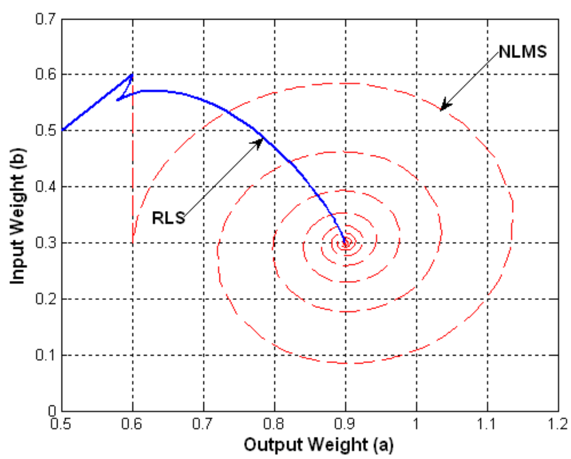


(b) Weight Convergence of RLS and NLMS

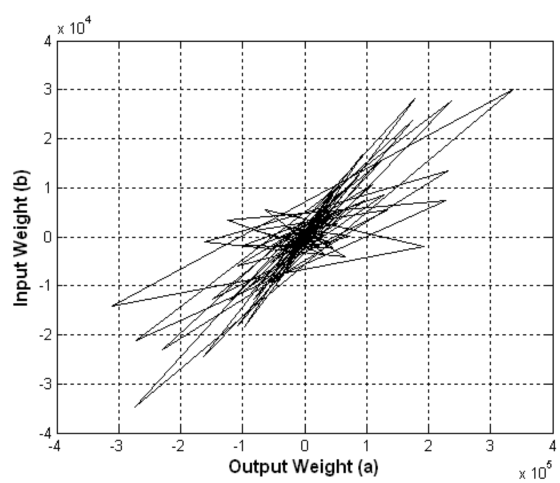


(c) Weight Convergence of LMS

Figure 4.1: Correlation Angle and Weight Convergence of RLS, NLMS and LMS



(a) Weights Trajectory of RLS and NLMS



(b) Weights Trajectory of LMS

Figure 4.2: Weights Trajectory of RLS, NLMS and LMS

It can be concluded that when the system is oversampled and consecutive observations reach the identifier without loss (when wireless link is strong), RLS functions well and the resulted correlated observations at the identifier side because of oversampling doesn't lead to serious deterioration in the performance. In contrary these circumstances normally show poor performance for both LMS and NLMS.

4.4 Covariance Matrix and Observations Discontinuity

Test problem I (refer to appendix A) used to show the RLS performance over the lossy link. Figure 4.3 depicts the effect of observations loss on the RLS performance at different settings of PDP. The green curve in the figure shows the performance of RLS in absence of loss where the error minimizes until it settles at relatively small value. The other curves show that as PDP increases the error increases.

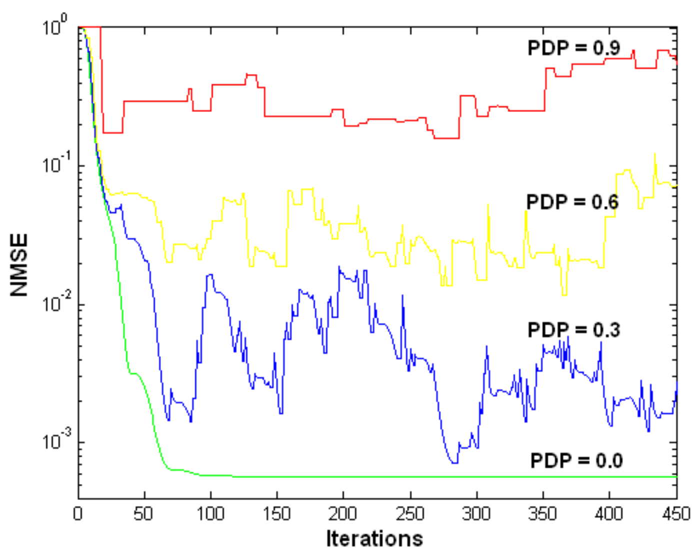


Figure 4.3: RLS Performance at Different Settings of PDP

After each training iteration RLS produces near optimal estimation for the weights vector using information about the previous observations in form of covariance matrix. This can be seen from equation (4.4) where the current covariance matrix (P_k)

equals the sum of the previous covariance matrix (P_{k-1}) (which holds information about previous observations in a sequenced form) and the inverse correlation of the current observations vector (φ_k). During random observations loss the sequence of the received observations tend to be irregular which creates discontinuity in the covariance matrix and affects its structure since there is no guarantee that φ_k is the observations vector which is right after last one has been included in P_{k-1} at the identifier side.

In order to illustrate this concept both of the equations (4.3) and (4.4) are merged in one equation as follows:

$$P_k = \left(\sum_{i=1}^{k-1} \varphi_i \varphi_i^T\right)^{-1} + (\varphi_k \varphi_k^T)^{-1} \tag{4.12}$$

Figure 4.4 illustrates the observations discontinuity concept in pictorial form. The figure shows a sequence of observations transmitted from the system to the identifier across wireless link. Assume P_4 is a covariance matrix without suffering from observations discontinuity, according to equation (4.12) it should be containing information about the current observations vector (φ_4) and all the observations vectors from φ_1 to φ_3 (which have been included in the previous covariance matrix (P_3)). Now assume that the observations from φ_5 to φ_9 have been completely lost during transmission as it can be seen from the corresponding output of Bernoulli process (from β_5 up to β_9), this loss leads to discontinuity in P_{10} since the current observations vector is φ_{10} and only the observations vectors from φ_1 to φ_4 are included in the previous covariance matrix which actually represents the previous covariance matrix for P_5 not for P_{10} as it is supposed to be.

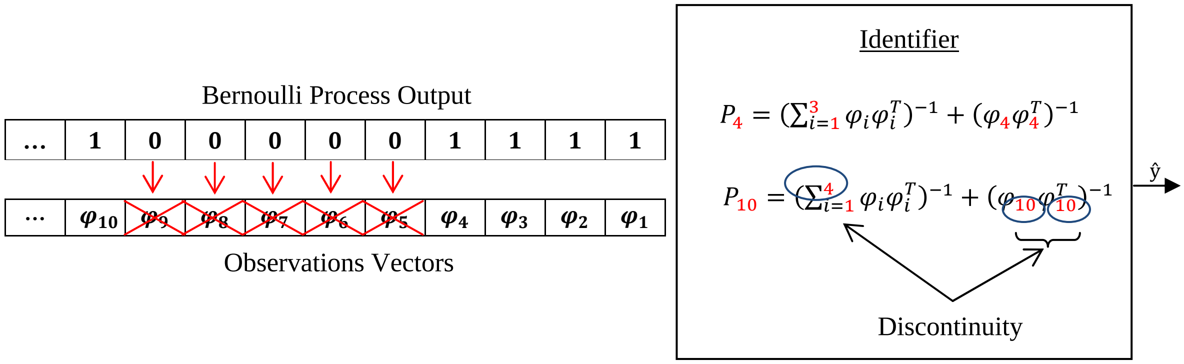
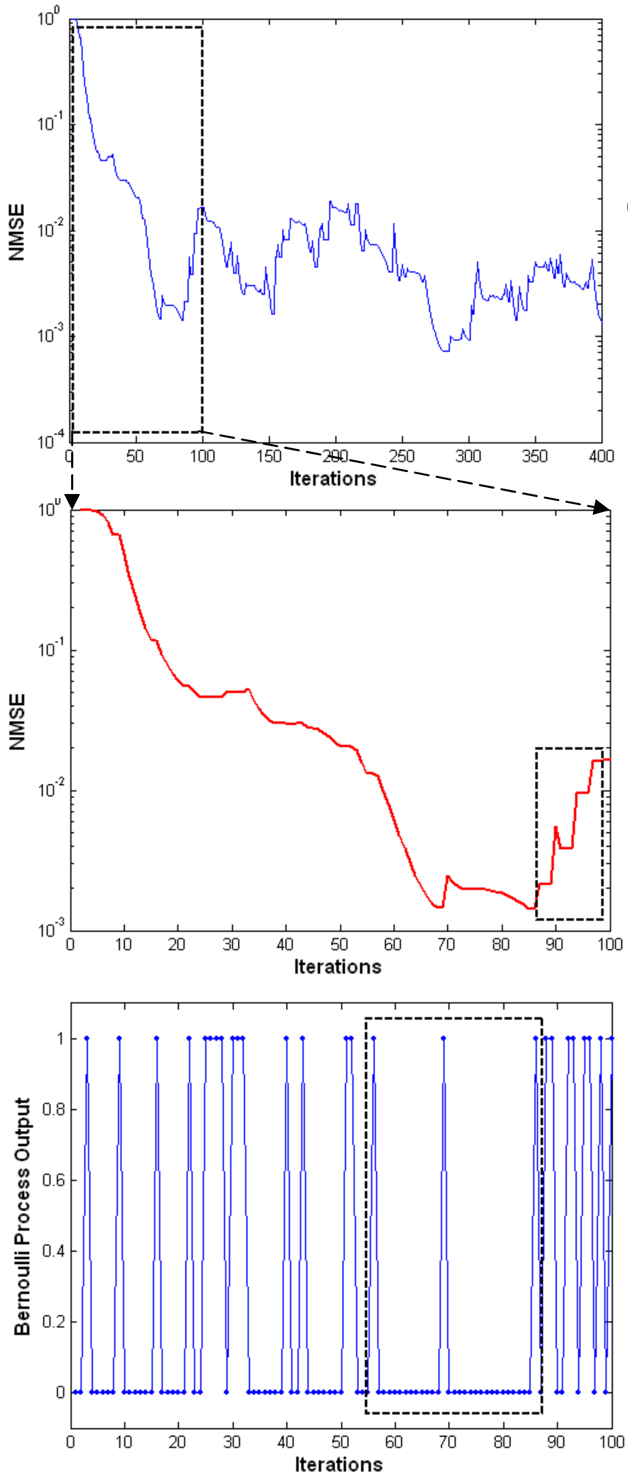


Figure 4.4: Observations Discontinuity Effect on the Covariance Matrix Structure

Figure 4.5(a) shows the learning curve of RLS where PDP was set to 0.3 and γ to 0.9. By taking a closer look at the period from 0 to 100 (graph 4.5(b)) and the corresponding output of Bernoulli process (graph 4.5(c)), it can be seen that observations discontinuity leads to serious deterioration in RLS performance.

In graph 4.5(c) there was no significant loss during the period from 0 to 50 and therefore the error was minimizing at this period as it can be seen from the corresponding learning curve in graph 4.5(b) which indicates that RLS was performing well and learning the system dynamics in this region.

In contrary during the period from 60 to 80 there was a significant loss (almost no reception of observations) as it can be seen from graph 4.5(c). The effect of this observations loss on RLS performance appears clearly in the later stage during the period from 90 to 100 as it can be seen from Graph 4.5(b) where the learning curve starts to go up between 80 and 90 which indicates that the algorithm was losing its ability to learn the system dynamics.



(a) Original Curve

(b) Zoomed Curve

(c) Corresponding Output of Bernoulli Process

Figure 4.5: Observations Discontinuity Effect on RLS Performance

Next section proposes a novel procedure to improve the performance of RLS algorithm during observations loss by taking into account the undesired effect of observations discontinuity on the covariance matrix structure and its consequences on the algorithm performance.

4.5 Weights Update Based on Observations Continuity

This section proposes a new version of RLS known as RLS with weights update based on observation continuity (RLS-OC). RLS-OC updates the covariance matrix (which holds information about the previous observations) whenever a new observation is available while it updates the weights vector only when there is a sort of observations continuity in the current covariance matrix.

The flowchart in Figure 4.6 illustrates the procedure of RLS-OC algorithm, where F_l is a loss flag indicates loss event when it equals 1 and it is initially set to 0. The procedure starts by checking whether the current packet has been lost or no, where in case of loss F_l is set to 1 and the procedure starts over to check for the next packet otherwise the regressor vector (φ_k) is formed and the covariance matrix is updated, then F_l is checked to test the observations continuity in the current covariance matrix (P_k), where in case of F_l being equal to 0 the weights vector (θ_k) is updated otherwise it is frozen. The procedure then computes the output estimation ($\hat{y}_k$) and checks whether the obtained model is accepted or not, where in case of acceptance the learning process ends otherwise F_l is set to 0 since the current packet has been received successfully and the procedure starts over.

Figure 4.7 shows the performance of RLS-OC in comparison with the traditional RLS at different settings of PDP. RLS-OC significantly outperforms the traditional RLS for all the different settings of PDP. The graphs in the figure show that as PDP increases the performance of the traditional RLS degrades while RLS-OC gives good performance, but as PDP value increases RLS-OC takes longer time to learn the system which can be contributed to the fact that when PDP increases the probability of not updating the weights increases (due to the contribution of the loss flag) which delays the learning process.

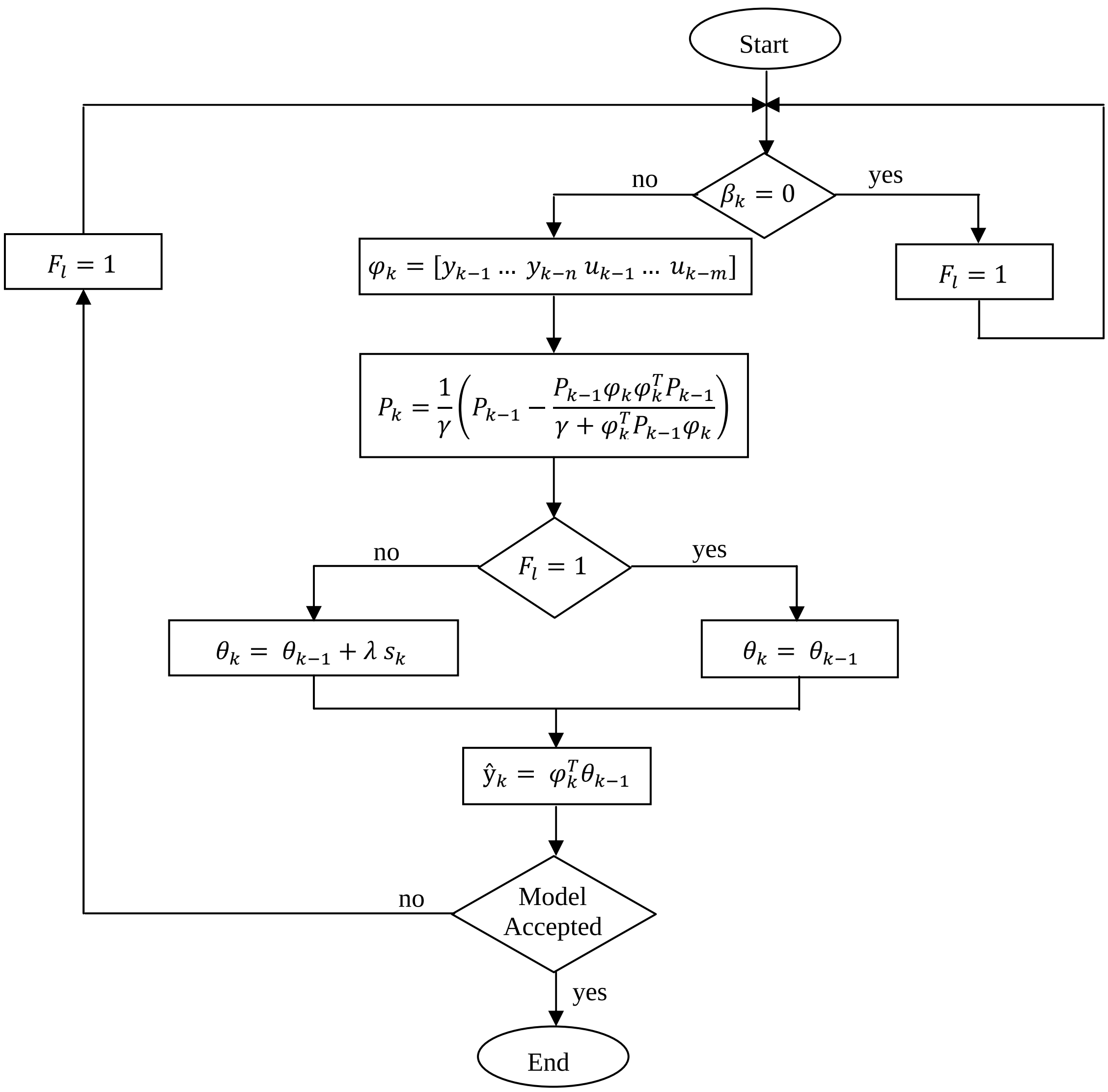
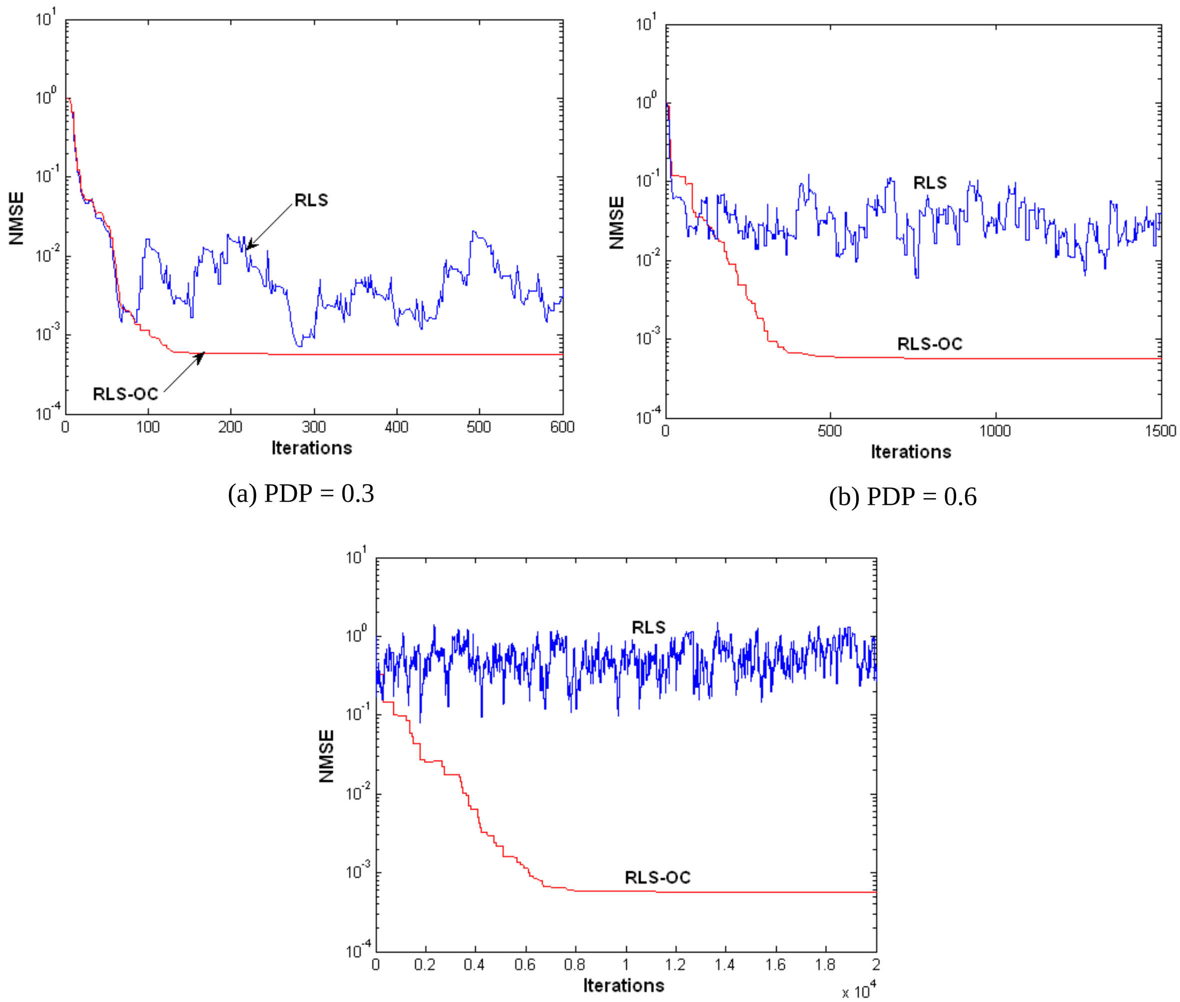


Figure 4.6: RLS-OC Algorithm Flowchart



(a) PDP = 0.3

(b) PDP = 0.6

(c) PDP = 0.9

Figure 4.7: RLS-OC and the Traditional RLS Performance at Different Settings of PDP

4.6 Recursive Instrumental Variable (RIV)

Recursive instrumental variable algorithm can be considered as a variant of RLS algorithm as it introduces an instrumental variables vector to RLS procedure. RIV algorithm can be derived by introducing the instrumental variables vector to equation (4.1) as follows:

$$\theta_k = \left(\sum_{i=1}^{k} z_i \varphi_i^T\right)^{-1} \left(\sum_{i=1}^{k} z_i y_i\right) \tag{4.13}$$

where:

$$z_i^T = [\hat{y}_{i-1} \ \dots \ \hat{y}_{i-n} \quad u_{i-1} \ \dots \ u_{i-m}] \tag{4.14}$$

where z_i^T is the instrumental variables vector, $\hat{y}_i$ is the estimated output and u_i is the system input.

The current covariance matrix (P_k) is given as:

$$P_k = \left(\sum_{i=1}^{k} z_i \varphi_i^T\right)^{-1} \tag{4.15}$$

The previous covariance matrix (P_{k-1}) is given as:

$$P_{k-1} = \left(\sum_{i=1}^{k-1} z_i \varphi_i^T\right)^{-1} \tag{4.16}$$

The recursive covariance matrix update for RIV is given as:

$$P_k = P_{k-1} + (z_k \varphi_k^T)^{-1} \tag{4.17}$$

$$P_k = (P_{k-1}^{-1} + z_k \varphi_k^T)^{-1} \tag{4.18}$$

Using the matrix inversion lemma:

$$P_k = (P_{k-1}^{-1} + z_k \varphi_k^T)^{-1} = P_{k-1} - P_{k-1} z_k (I + \varphi_k^T P_{k-1} z_k)^{-1} \varphi_k^T P_{k-1} \quad (4.19)$$

Hence, RIV algorithm can be expressed as:

$$\theta_k = \theta_{k-1} + P_k z_k e_k \quad (4.20)$$

$$e_k = y_k - \varphi_k^T \theta_{k-1} \quad (4.21)$$

$$P_k = P_{k-1} - \frac{P_{k-1} z_k \varphi_k^T P_{k-1}}{1 + \varphi_k^T P_{k-1} z_k} \quad (4.22)$$

By introducing the forgetting factor (γ) to equation (4.22):

$$P_k = \frac{1}{\gamma}\left(P_{k-1} - \frac{P_{k-1} z_k \varphi_k^T P_{k-1}}{\gamma + \varphi_k^T P_{k-1} z_k}\right) \quad where\ \ 0 < \gamma \leq 1 \quad (4.23)$$

The weights update based on observation continuity technique is also applicable to RIV since it shares with RLS the same weights update principle. Figure 4.8 shows RIV-OC performance in comparison with the traditional RIV at different settings of PDP. The graphs show that as PDP increases the traditional RIV gives poor performance while RIV-OC shows satisfactory performance for all the different settings of PDP. Similar to RLS when PDP value increases RIV-OC takes longer time to learn the system dynamics which can be attributed to the fact that as PDP increases the probability of not updating the weights increases (due to the contribution of the loss flag) which delays the learning process.

Table 3.2 summarizes the performance of the optimized algorithms that have been proposed in this chapter in comparison with their original versions. Both average and standard deviation of the instantaneous normalized mean square error (NMSE) over last 500 iterations (refer to appendix A) have been recorded in the table.

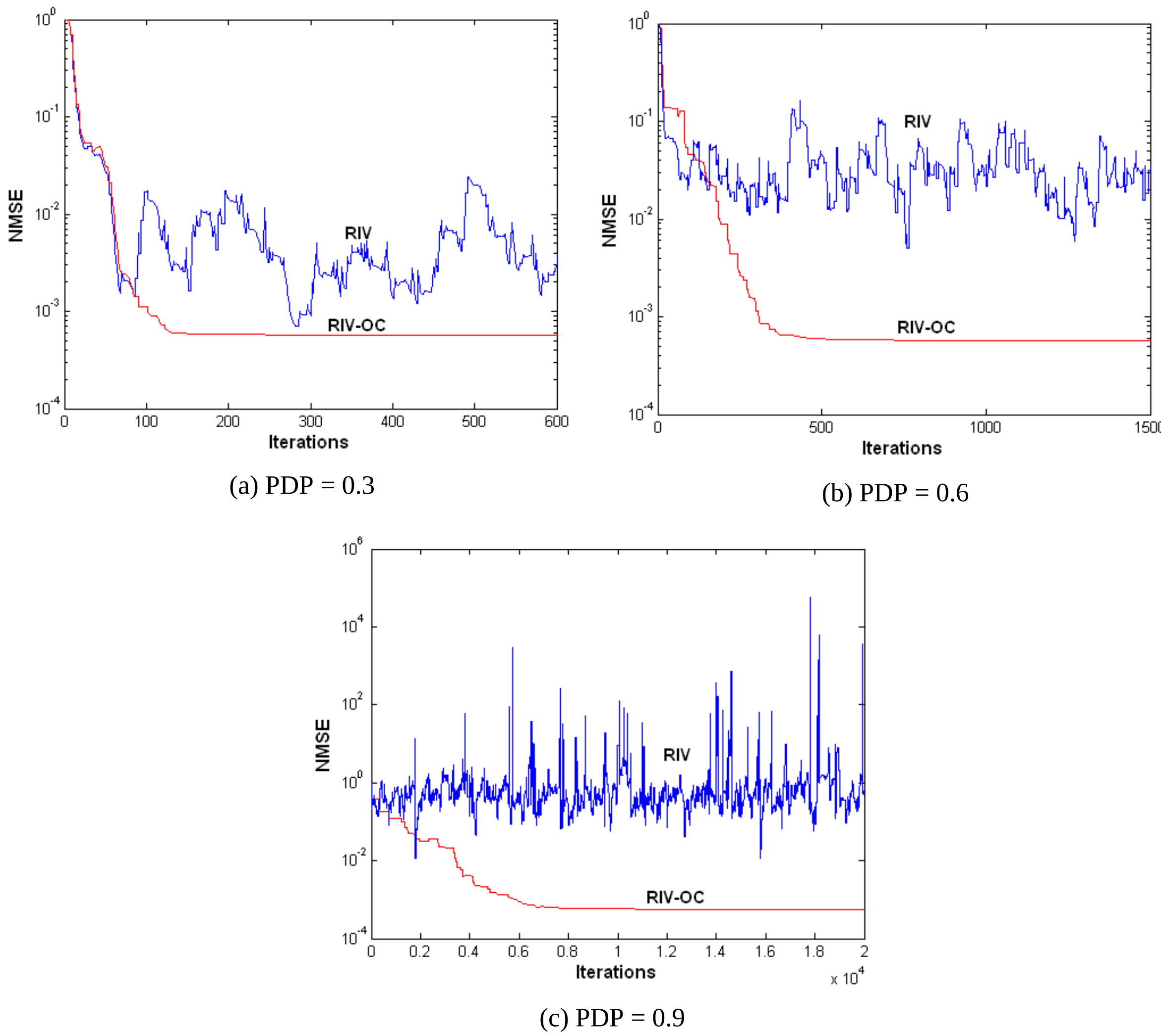


Figure 4.8: RIV-OC and the Traditional RIV Performance at Different Settings of PDP

Table 4.1: Numerical Results for the Second Order Optimized and Original Algorithms Performance at Different Settings of PDP

Algorithm \ PDP	Measure	0.3	0.6	0.9
RLS	Average	0.0054	0.0254	0.6110
	Standard Deviation	0.0048	0.0119	0.2317
RLS-OC	Average	**5.7551e-004**	**5.7551e-004**	**5.7551e-004**
	Standard Deviation	**1.2015e-012**	**8.7553e-013**	**7.8127e-012**
RIV	Average	0.0058	0.0278	128.8472
	Standard Deviation	0.0056	0.0139	675.4070
RIV-OC	Average	**5.7551e-004**	**5.7551e-004**	**5.7551e-004**
	Standard Deviation	**1.1504e-012**	**6.9510e-013**	**1.0661e-011**

4.7 Conclusion

This chapter has investigated the performance of RLS and RIV during observations loss. It has started by reviewing the derivation of RLS algorithm from the ordinary least squares algorithm. The chapter then has proceeded to highlight the capability of RLS algorithm in dealing with the correlated observations compared to the first order learning algorithms such as LMS and NLMS, the simulation results have confirmed the outperformance of RLS compared to both LMS and NLMS in dealing with the correlated observations.

The chapter has also addressed the effect of observations discontinuity on the covariance matrix structure and its consequences on the algorithm performance. Thus, a novel technique has been introduced known as weights update based on observations continuity for both RLS (RLS-OC) and RIV (RIV-OC) algorithms. The performances of RLS-OC and RIV-OC have been evaluated in comparison with the traditional RLS and RIV respectively at different settings of PDP, the comparison has showed that RLS-OC and RIV-OC have the upper hand compared to their original versions.

Next chapter will look into the high order algorithms which utilize information from data store of previous observations to perform instantaneous system identification.

CHAPTER FIVE: HIGH ORDER INSTANTANOUS LEARNING ALGORITHMS

5.1 Introduction

The major deficiency of first order instantaneous learning algorithms such as LMS and NLMS is that their performance is strongly dependent on the nature and quality of the presented training data. In particular when successive data points are highly correlated, weights convergence occurs very slowly if at all [13]. On other hand the superior convergence properties of more advanced second order instantaneous learning algorithms such as RLS and RIV is at the expense of significantly greater computational complexity and memory requirements arising from storing and updating an (n x n) covariance matrix at each iteration. In many practical applications these methods cannot be utilized due to real-time and hardware constraints [18].

This chapter investigates and optimizes an alternative class of training algorithms during observations loss, this class is referred to as sliding window (SW) training algorithms. SW algorithms utilize information from sliding window or store of previous data points to improve the performance while maintaining the simplicity of the first order algorithms in terms of computational complexity and use of memory.

The chapter shows the formulation of SW and outlines moving average (MA) algorithm as an example of SW training algorithms. The chapter then proceeds to discuss some data store management (DSM) strategies as they are a key consideration when using SW training algorithms. In this chapter two DSM strategies are proposed to manage the store in presence of random observations loss and based on these strategies two SW based algorithms are proposed to improve the performance during observations loss.

5.2 Sliding Window (SW)

Sliding window training algorithms (also known as high order training algorithms [19]) use sliding window of system observations to perform instantaneous system identification. In sliding window training the model weights are updated using information obtained from store of L previous training pairs. The regressor vector (φ_k), weights vector (θ_k) and data store (Sd_k) are defined as follows:

$$\varphi_k = \begin{bmatrix} y_{k-1} \\ \cdots \\ y_{k-n} \\ u_{k-1} \\ \cdots \\ u_{k-m} \end{bmatrix} \quad \theta_k = \begin{bmatrix} \theta_1 \\ \cdots \\ \theta_n \\ \cdots \\ \theta_{n+m} \end{bmatrix} \quad Sd_k = \begin{bmatrix} \varphi_1 & \varphi_2 & \cdots & \varphi_i & \cdots & \varphi_L \\ y_1 & y_2 & \cdots & y_i & \cdots & y_L \end{bmatrix} \quad (5.1)$$

The pair (φ_i, y_i) refers to the i^{th} training vector at time k. The output error for this vector with respect to the current set of weights is defined as follows:

$$e_i = y_i - \varphi_i^T \theta_k \quad (5.2)$$

5.2.1 Moving Average (MA) and Normalized Moving Average (NMA)

Given L data store vectors and the current data points φ_k and y_k, MA algorithm computes moving average search direction (s_k^{MA}) for LMS as follows:

$$s_k^{MA} = \frac{1}{L} \sum_{i=1}^{L} e_i \, \varphi_i \quad (5.3)$$

For the same given data NMA algorithm computes normalized moving average search direction (s_k^{NMA}) for NLMS as follows:

$$s_k^{NMA} = \frac{1}{L}\sum_{i=1}^{L}\frac{e_i\varphi_i}{\|\varphi_i\|} \tag{5.4}$$

5.3 Data Store Management (DSM)

Managing or maintaining the data store for sliding window is as important as choosing a suitable training algorithm and should not be overlooked. One of the simplest strategies for data store management is to adopt first in first out (FIFO) strategy. This strategy discards the oldest training pair in the store and admits the current one as follows:

$$\varphi_i = \varphi_{k-L-1+i} \qquad y_i = y_{k-L-1+i} \tag{5.5}$$

FIFO strategy has an advantage that the store information represents the current state of the system.

The received observations at the identifier side might be correlated because of oversampling and therefore DSM strategies should take into account the correlation effect. Next section outlines one of the DSM strategies known as total correlation measurement (TCM) strategy which reduces the correlation effect.

5.3.1 Total Correlation Measurement (TCM)

Total correlation measurement strategy [18] manages the store based on a measure of the total correlation between the training pairs, it computes the total correlation angle (Tc) between the store members as well as the current training pair (φ_k,y_k), then the training pair with the smallest total correlation angle (φ_a,y_a) is replaced with the current input pair (φ_k, y_k) as follows:

$$Tc_i = \sum_{\substack{j=1 \\ j\neq i}}^{L+1} \left| cos^{-1}\left(\frac{{\varphi_i}^T \varphi_j}{\|\varphi_i\|\|\varphi_j\|}\right)\right| \quad a = arg\left\{\min_i(Tc_i)\right\} \begin{cases} \underline{a \neq L+1:} \\ \begin{bmatrix}\varphi_a \\ y_a\end{bmatrix} = \begin{bmatrix}\varphi_k \\ y_k\end{bmatrix} \\ \\ \underline{a = L+1:} \\ (\varphi_{k,} y_k)\ discarded \end{cases} \tag{5.6}$$

Since TCM computes the total correlation angle for the store members as well as the current training pair it is possible to discard the current training pair in the process.

Figure 5.1 shows store members of (L=2) and the current training pair, based on TCM concept the training pair (φ_2,y_2) could possibly be discarded since it has the smallest total correlation angle.

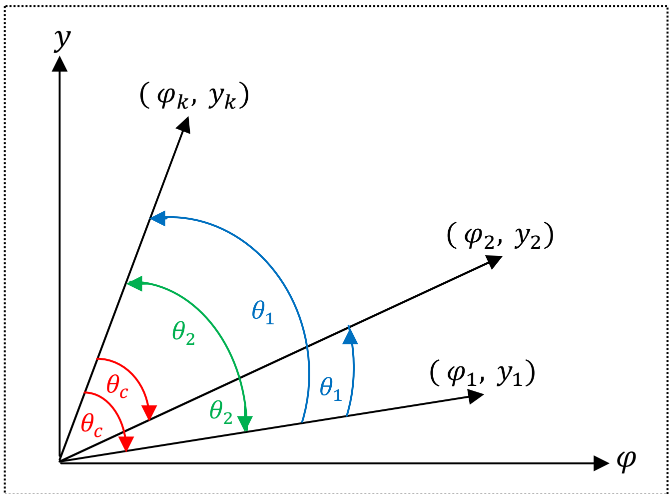
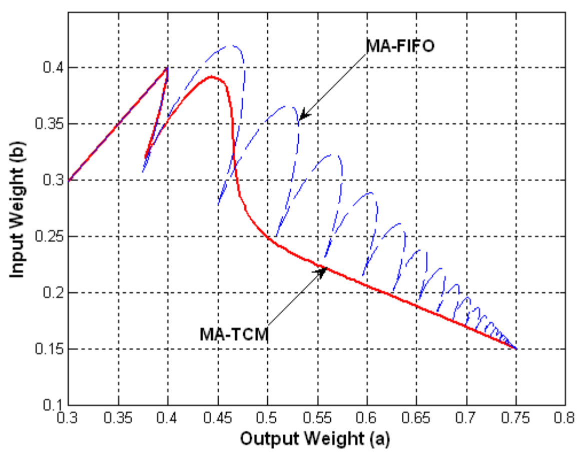
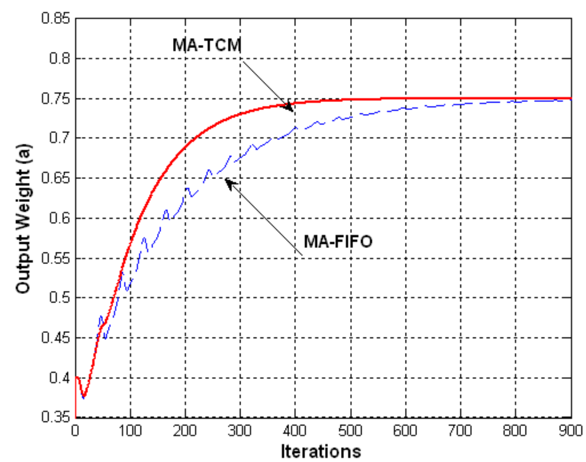


Figure 5.1: Total Correlation Measurement Concept

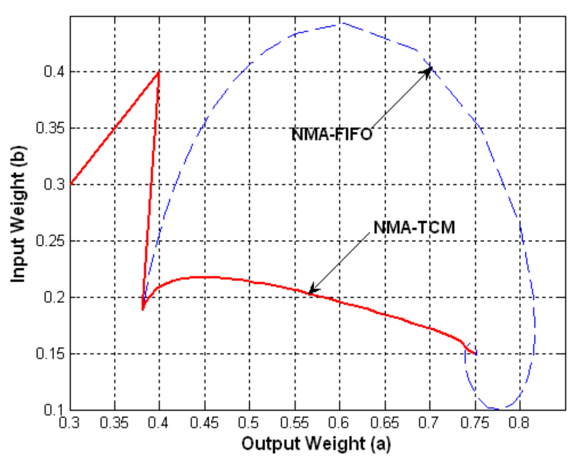
Test problem II-C (refer to appendix A) used to evaluate the performance of TCM strategy in comparison with FIFO strategy for MA and NMA. The desired weights vector for the test problem is $\theta^* = [0.75,0.15]$. Figure 5.2 shows that both MA and NMA converge faster when using TCM in comparison with FIFO.



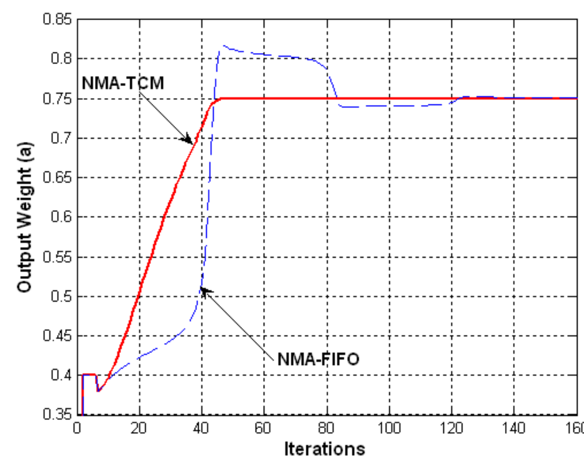
(a) Weights Trajectory for MA



(b) Weight Convergence for MA



(c) Weights Trajectory for NMA



(d) Weight Convergence for NMA

Figure 5.2: Weights Trajectory and Convergence for TCM and FIFO Strategies

During observations loss the output error contributes with high magnitude in the weights update process which destroys the learning process as it has been explained in chapter 3. Thus, DSM strategies should take into account the undesired contribution of the output error in the weights update process. Next section proposes DSM strategy known as error measurement (EM) strategy to reduce this undesired contribution of the output error.

5.3.2 Error Measurement (EM)

Error measurement strategy manages the data store based on the output error value. Typically when the training pairs number exceeds the predefined window size for the store, EM strategy computes the output error for each training pair then it discards the training pair with the largest output error (φ_a, y_a) and admits the current one (φ_k, y_k) as follows:

$$a = arg\{\max_i |e_i|\} \qquad \begin{bmatrix} \varphi_a \\ y_a \end{bmatrix} = \begin{bmatrix} \varphi_k \\ y_k \end{bmatrix} \tag{5.7}$$

One advantage of using EM strategy is that during random observations loss the undesired contribution of the output error in the weights update process will be reduced since the store will always contain the training pairs with the smallest output error.

5.3.3 Intelligent Data Store Management (IDSM)

Intelligent data store management strategy keeps the store information representative for the current state of the system and also reduces the undesired effects of both the correlation and the output error. It achieves this by combining the three strategies mentioned previously (FIFO, TCM and EM), FIFO keeps the store information representative for the current state of the system and each of TCM and EM reduces the undesired effects of the correlation and the output error respectively. Thus, the insignificant training pair from IDSM point of view is the training pair which is old, highly correlated with others and has high output error value. The following steps explain the strategy:

- Step 1: Weighting the store training pairs and the current one from 1 to ($L+1$) according to the length of time they have been members of the store. Typically the oldest training pair is given the smallest weight (1) and the newest training pairs is given the biggest weight ($L+1$).

- Step 2: Weighting the store training pairs and the current one from 1 to ($L+1$) according to the total correlation angle. Typically the training pair with the smallest angle is given the smallest weight (1) and the training pair with the biggest angle is given the biggest weight ($L+1$).

- Step 3: Weighting the store training pairs and the current one from 1 to ($L+1$) according to the output error value. Typically the training pair with the biggest output error is given the smallest weight (1) and the training pair with the smallest output error is given the biggest weight ($L+1$).

- Step 4: Adding the weights that have been obtained from the three steps and discarding the training pair with the smallest score.

5.4 Moving Average Based on Combination of EM and SD (MA-EMSD)

Moving average based on combination of error management (EM) strategy and sine function based de-correlation (SD) technique (MA-EMSD) manages the store using EM strategy while sets the learning rate according to SD technique (see chapter 3). The procedure of MA-EMSD is illustrated by the flowchart in Figure 5.3, where c is a small constant learning rate and F_l is a loss flag indicates loss event when it equals 1 and it is initially set to 0. The procedure starts by checking whether the current packet has been lost or not, where in case of loss the procedure sets F_l to 1 and starts over to check for the next packet otherwise the current observations vector is admitted into the store and the number of the store members is checked to see whether it has exceeded the predefined window size or not, where in case of not exceeding the predefined window size F_l is set to 0 since the current packet been received successfully and the procedure starts over to check for the next packet otherwise the store member with the largest output error will be replaced with the current observations vector according to EM strategy. The procedure then proceeds to check F_l in order to determine whether the previous packet has been lost or not, where in case of loss the small constant learning rate (c) will be considered otherwise the adaptive learning rate (λ_{sin}) will be considered which is computed in SW case as follows:

$$\lambda_{sin} = \frac{1}{L}\sum_{i=1}^{L} \sin(Tc_i) \tag{5.8}$$

Once the learning rate has been determined the procedure computes the moving average search direction (s_k^{MA}) then it proceeds to update the weights and compute the estimated output ($\hat{y}_k$). The procedure then checks whether the obtained model is accepted or not, where in case of acceptance the learning process ends otherwise F_l is set to 0 since the current packet has been received successfully and the procedure starts over.

EMSD method is also applicable to the normalized moving average (NMA) algorithm; the only difference is that the search direction will be normalized as it is shown in equation (5.4).

Test problem I (refer to appendix A) used to evaluate the performance of MA-EMSD in comparison with MA-FIFO at different settings of PDP. Figure 5.4(a) shows that in absence of loss (PDP = 0) MA-EMSD settles faster than MA-FIFO since it sets the learning rate according to SD technique. The rest of the graphs (5.4(b)-5.4(d)) shows that during observations loss MA-EMSD performs better than MA-FIFO which can be attributed to the fact that MA-EMSD reduces the undesired contribution of the output error using EM strategy which always ensures the existence of training pairs with smallest output error in the store.

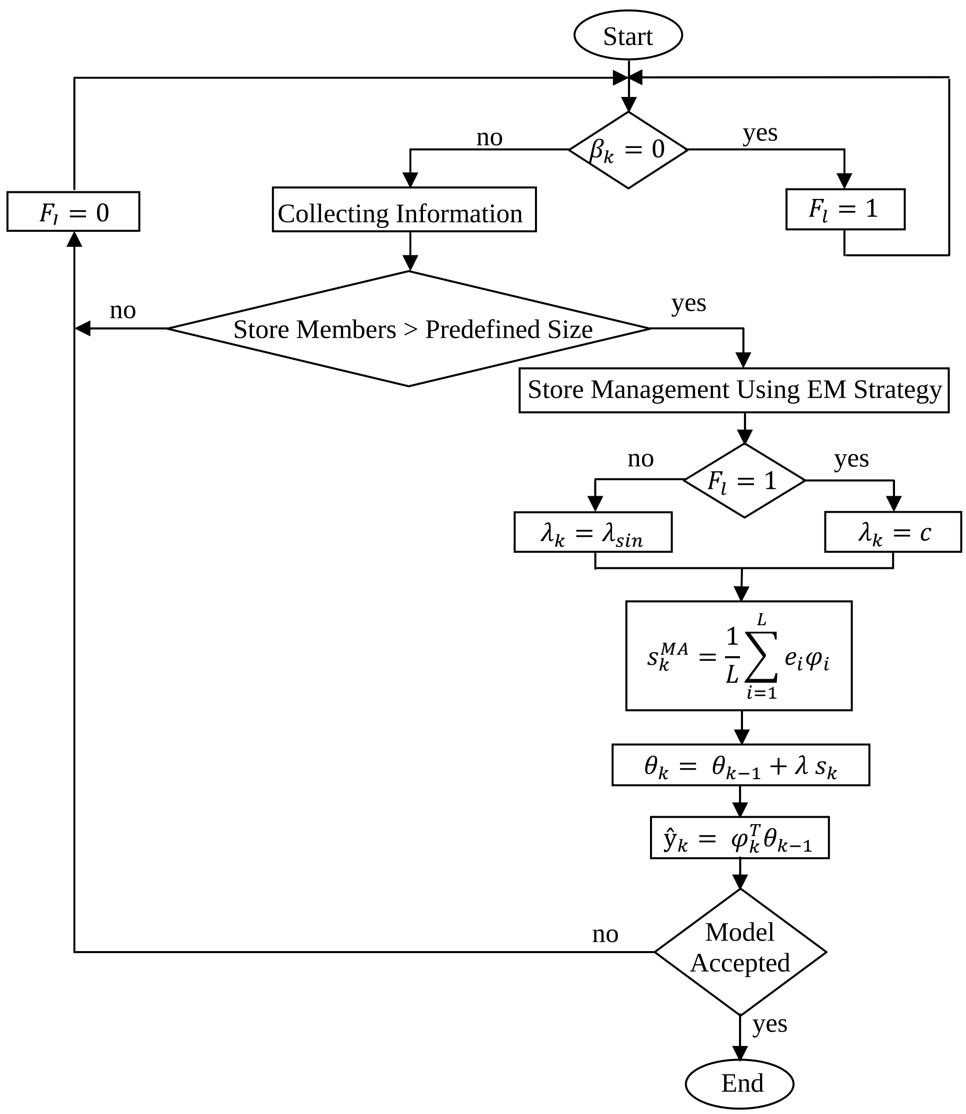
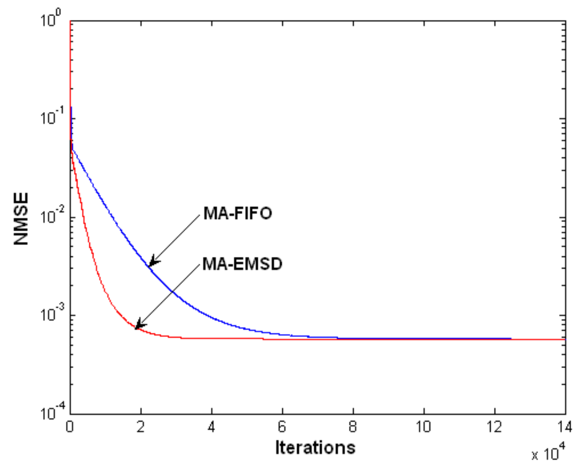
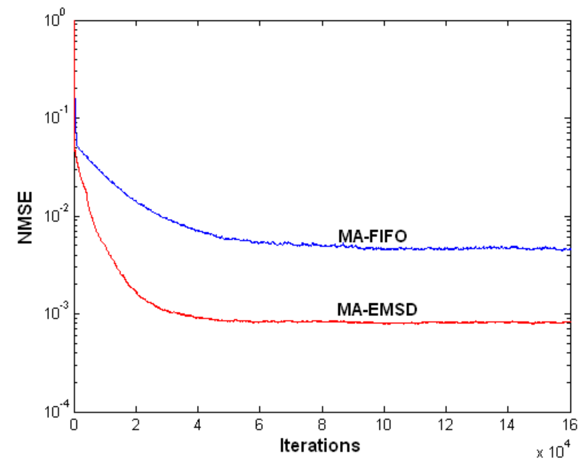


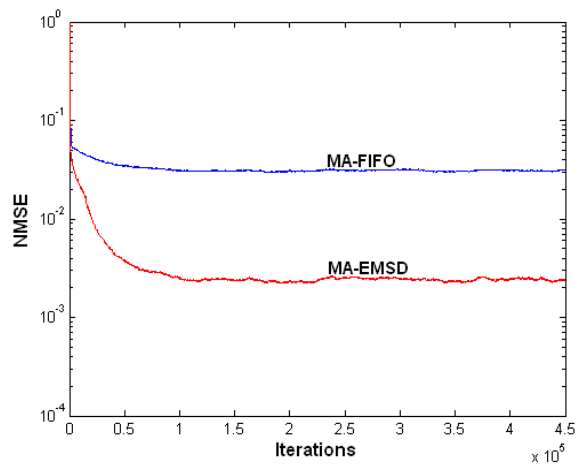
Figure 5.3: MA-EMSD Algorithm Flowchart



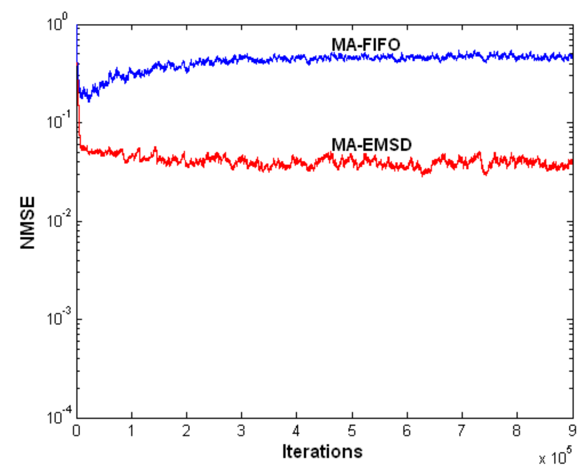
(a) PDP = 0.0



(b) PDP = 0.3



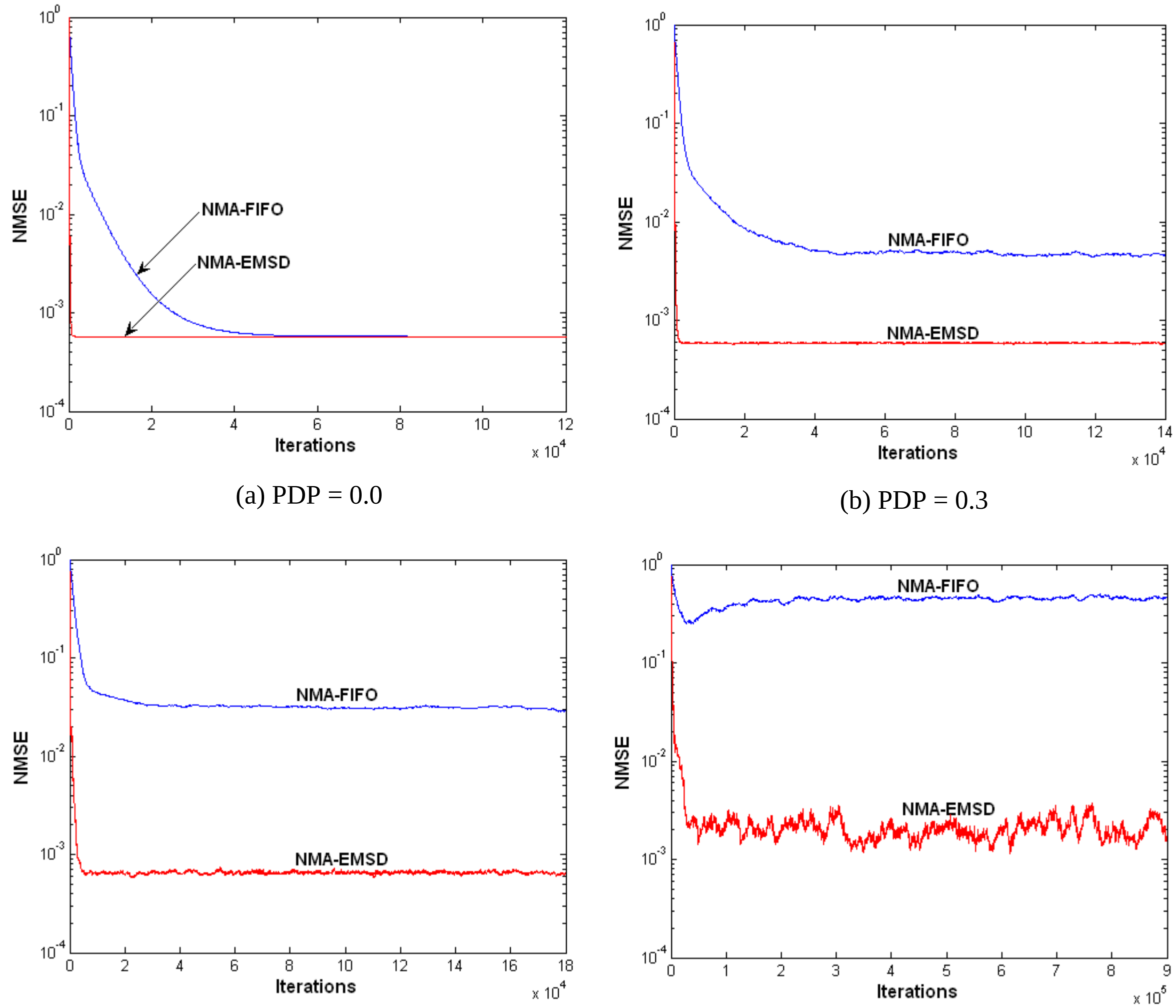
(c) PDP = 0.6



(d) PDP = 0.9

Figure 5.4: MA-EMSD and MA-FIFO Performance at Different Settings of PDP

Figure 5.5 shows similar comparison between NMA-EMSD and NMA-FIFO, similarly the comparison shows that NMA-EMSD gives better performance compared to NMA-FIFO for all the different settings of PDP.



(a) PDP = 0.0

(b) PDP = 0.3

(c) PDP = 0.6

(d) PDP = 0.9

Figure 5.5: NMA-EMSD and NMA-FIFO Performance at Different Settings of PDP

5.5 Intelligent Moving Average (IMA)

This chapter proposes another SW based algorithm termed as intelligent moving average (IMA) algorithm, the prefix term 'intelligent' is given because it uses IDSM strategy in addition to SDEDU method (see chapter 3).

The procedure of IMA algorithm is illustrated by the flowchart in Figure 5.6, where c is a small constant learning rate and F_l is a loss flag indicates loss event when it equals 1 and it is initially set to 0. The procedure works similar to MA-EMSD until it reaches the store management stage, at this point the procedure manages the store according to IDSM then it checks F_l to determine whether the previous packet has been lost or not, where in case of loss the small constant learning rate (c) is considered and both of the error displacement (Ed_k) and the output error (e_k) are computed from the store training pairs as moving average (so the search direction will computed using the moving average output error) otherwise the adaptive learning rate (λ_{sin}) is considered and both of the error displacement (Ed_k) and the output error (e_k) are computed using only the current training pair (so the search direction will be computed using only the current output error). Once the learning rate, output error, search direction and error displacement have been determined the algorithm proceeds to the weights (θ_k) update stage. The weights are updated only if the error displacement is not greater than the predefine threshold (σ) otherwise the weights are frozen. The procedure then proceeds to compute the output estimation ($\hat{y}_k$) and checks whether the obtained model is accepted or not, where in case of acceptance the learning process ends otherwise F_l is set to 0 since the current packet has been received successfully and the procedure starts over.

The intelligent approach is also applicable to the normalized moving average (NMA); the only difference is that the search direction will be normalized as it is shown in equation (5.4).

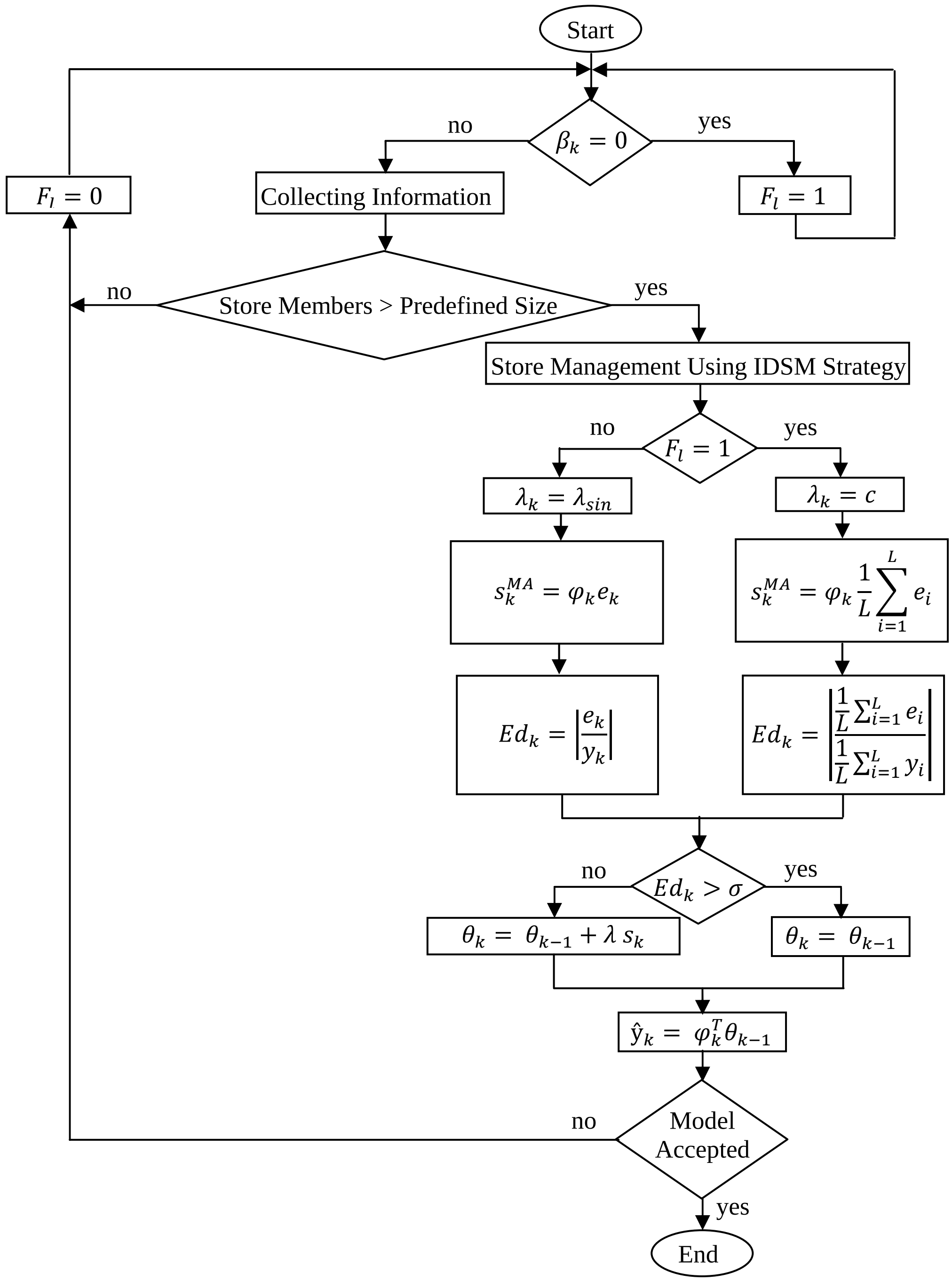


Figure 5.6: IMA Algorithm Flowchart

Figure 5.7 shows the performance of IMA in comparison with MA-FIFO at different settings of PDP. Graph 5.7(a) shows the learning curves in absence of loss (PDP

= 0) where IMA settles faster since TCM strategy contributes in deciding which vector is discarded as the number of the store members exceeds the predefined window size, furthermore the procedure has been made more robust by setting the learning rate according to SD technique as observations become correlated. The rest of the graphs (5.7(b) - 5.7(d)) show that IMA outperforms MA-FIFO during observations loss which can be attributed to the fact that EM strategy ensures the existence of training pairs with smallest output error in the store and therefore the contribution of the output error in the weights update process is reduced. Moreover, EDU technique allows weights update only when the output error contribution is acceptable.

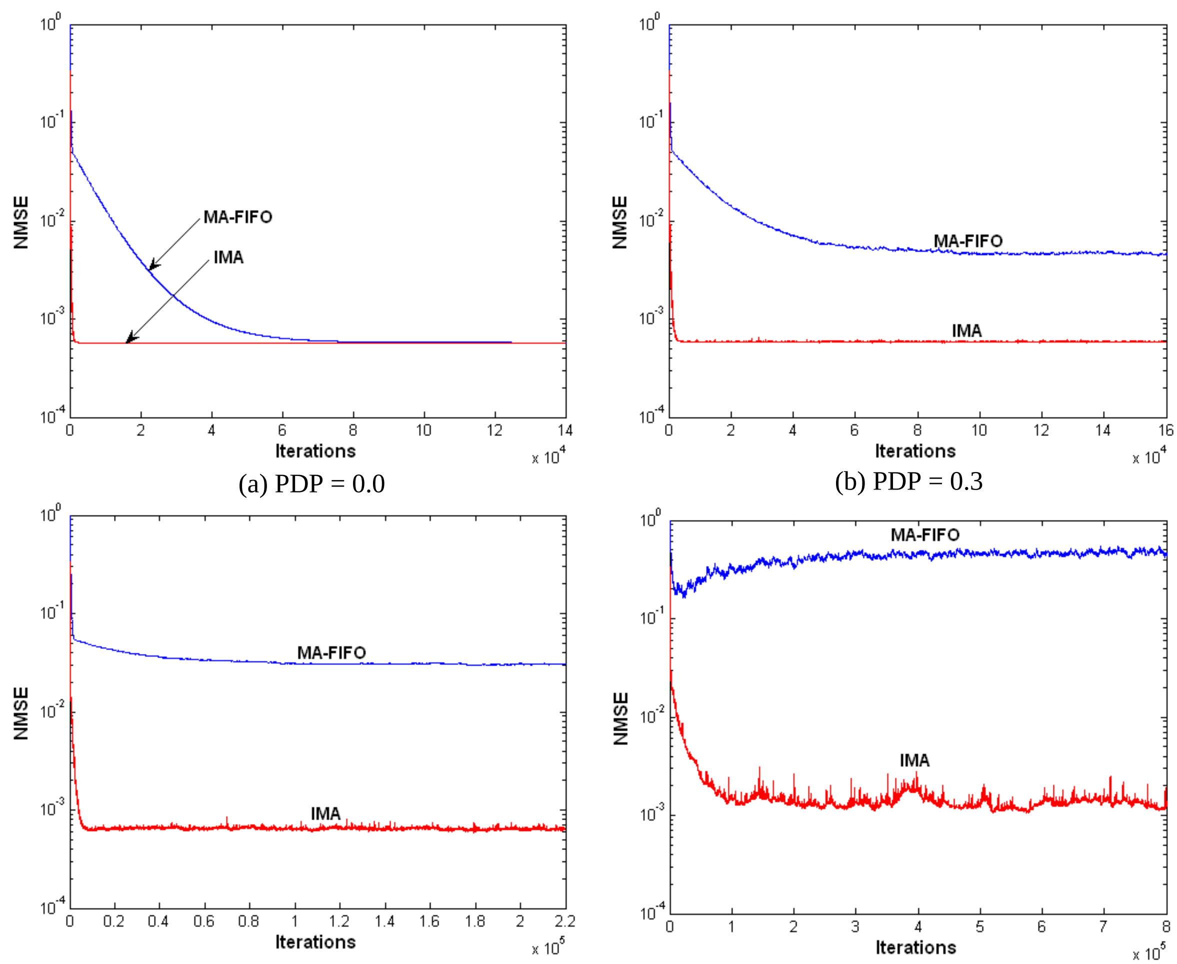
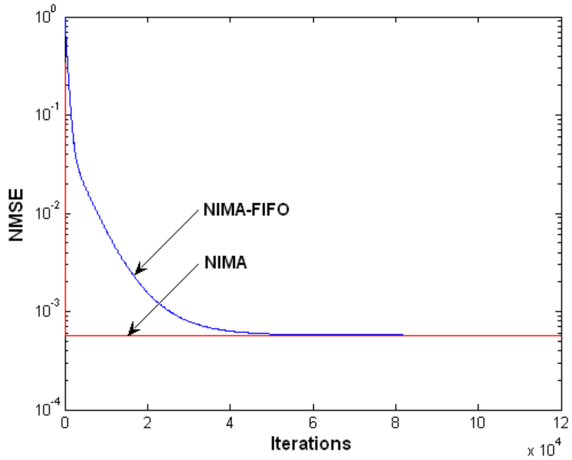
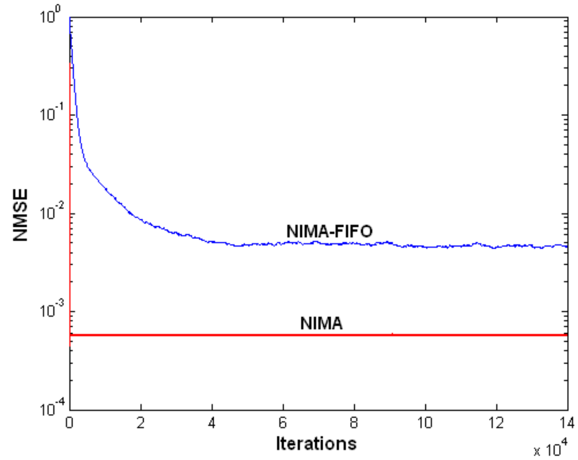


Figure 5.7: IMA and MA-FIFO Performance at Different Settings of PDP

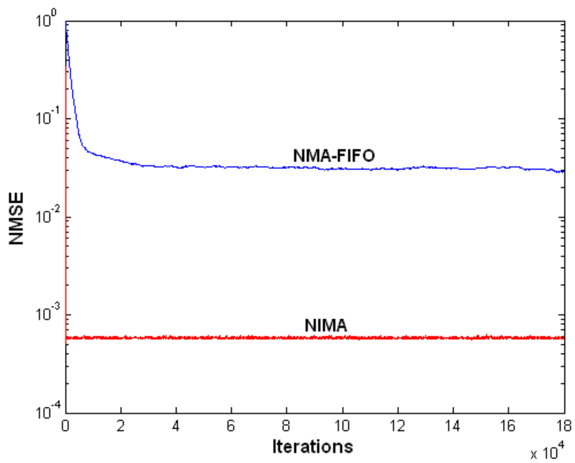
Figure 5.8 shows similar comparison between NIMA and NMA-FIFO. The comparison shows similar pattern of performance gain.



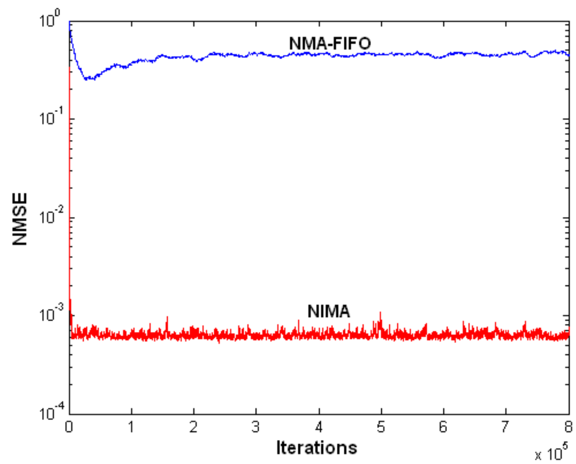
(a) PDP = 0.0



(b) PDP = 0.3



(c) PDP = 0.6



(d) PDP = 0.9

Figure 5.8: NIMA and NMA-FIFO Performance at Different Settings of PDP

Table 5.1 summarizes the performance of the optimized algorithms that have been proposed in this chapter in comparison with their original versions. Both average and standard deviation of the instantaneous normalized mean square error (NMSE) over last 20000 iterations (refer to appendix A) have been recorded in the table. The results show that the optimized algorithms perform better than their original versions during observations loss. It can also be seen from the results that the optimized algorithms based on the intelligent approach performs better than the optimized algorithms based on EMSD method.

Table 5-1: Numerical Results for the High Order Optimized and Original Algorithms Performance at Different Settings of PDP

Algorithm \ PDP	Measure	0.0	0.3	0.6	0.9
MA-FIFO	Average	5.7568e-004	0.0046	0.0308	0.4594
	Standard Deviation	8.2558e-008	9.0230e-005	3.2208e-004	0.0189
MA-EMSD	Average	**5.7551e-004**	**8.1726e-004**	**0.0024**	**0.0374**
	Standard Deviation	**4.0216e-013**	**9.5268e-006**	**4.6127e-005**	**0.0023**
IMA	Average	**5.7551e-004**	**5.8626e-004**	**6.3919e-004**	**0.0012**
	Standard Deviation	**2.0420e-016**	**2.8622e-006**	**1.2090e-005**	**9.3302e-005**
NMA-FIFO	Average	5.7552e-004	0.0046	0.0304	0.4585
	Standard Deviation	5.7917e-009	9.5090e-005	0.0011	0.0065
NMA-EMSD	Average	**5.7551e-004**	**5.8809e-004**	**6.4976e-004**	**0.0023**
	Standard Deviation	**1.6871e-016**	**4.8600e-006**	**1.9513e-005**	**4.3538e-004**
NIMA	Average	**5.7551e-004**	**5.7676e-004**	**5.8166e-004**	**6.3372e-004**
	Standard Deviation	**1.6730e-016**	**1.4497e-006**	**4.9847e-006**	**5.0457e-005**

5.6 Conclusion

This chapter has investigated the performance of sliding window (SW) training algorithms in presence of random observations loss. It has started by showing the formulation of SW and outlining moving average (MA) algorithm, then it has proceeded to discuss various data store management (DSM) strategies as they form a key ingredient of SW training algorithms.

With inclusion of observations loss raises other considerations, new DSM strategy known as error measurement (EM) strategy has been introduced. EM strategy manages the store based on the output error value where it always ensures the existence of training pairs with smallest output error in the store. New version of MA algorithm known as MA-EMSD has been proposed by adopting a combination of EM strategy and SD technique on MA algorithm. The proposed algorithm has improved the performance in comparison with MA-FIFO during observations loss for all the different settings of PDP.

The chapter has proposed another DSM strategy known as IDSM which combines each of FIFO, TCM and EM strategies using weighting rules. In the environment where random observation loss is imminent, an adopted combination of SDEDU method and IDSM strategy on MA algorithm has formed another version of MA algorithm that is the intelligent moving average (IMA). The new algorithm has showed satisfactory performance and good capability of handling each of the correlation and the output error effects in comparison with MA-FIFO during observations loss for all the different settings of PDP.

CHAPTER SIX: CONCLUSION AND FUTURE WORK

This thesis has focused the research scope on investigating and optimizing various instantaneous learning algorithms to allow of performing instantaneous system identification over wireless network. The instantaneous learning algorithms in question can be divided into three categories based on the order of complexity, which are:

- First order algorithms (least mean square (LMS) and normalized least mean square (NLMS)).
- Second order algorithms (recursive least squares (RLS) and recursive instrumental variable (RIV)).
- High order or sliding window (SW) algorithms (moving average (MA) and normalized moving average (NMA)).

The main contribution of the thesis is optimizing instantaneous learning algorithms for instantaneous system modeling with lost observations. The thesis has adopted oversampling rule to compensate the lost observations and therefore the learning algorithms should be robust to work with correlated observations (when wireless link is strong) as well as random structure of observations (when wireless link is weak). The optimized algorithms in this thesis combine all the mentioned requirements above and they have showed good performance and high capability of treating the effect of observations loss.

6.1 Chapters Revisiting

The thesis has started by discussing the performance of off-line system identification approaches over lossy link in Chapter 2 where it has been found that the performance deterioration due to observations loss is not that critical compared to the on-line counterpart. This capability of modeling the system off-line during observations loss can be attributed to the fact that off-line approaches model the system by processing batch of its observations which allows of capturing the system dynamics even when some of these observations are lost. The effect of observations loss can be reduced for off-line learning by exciting the system with an appropriate stimulated signal to produce informative observations.

Chapter 3 has started the optimization of the instantaneous learning algorithms with the algorithms of the first order. The chapter has looked into the sampling process role in mitigating the effect of observations loss where it has been found that oversampling improves the performance since it compensates the lost observations. Since oversampling could lead to correlated observations at the identifier side (when wireless link is strong), new technique known as sine function based de-correlation (SD) has been proposed to treat the correlation effect instantaneously at the identifier side using adaptive learning rate. New version of LMS known as LMS-SD has been introduced by adopting the technique on the classical LMS algorithm. During observations loss the output error tends to contribute with high magnitude in the search direction which leads to significant changes in the weights update direction and destroys the learning process. Therefore, a new technique known as error displacement based update (EDU) has been introduced, the technique reduces the undesired contribution of the output error in the weights update process by allowing the weights update only when the output error contribution is acceptable. By adopting a combination of SD and EDU techniques on LMS another version of LMS known as LMS-SDEDU has been proposed. Both of the proposed SD technique and SDEDU method have been applied to the NLMS. The optimized algorithms in general have showed good improvement in the performance compared to their original versions. The optimized algorithms based on SDEDU method have shown better performance than the optimized algorithms based on SD technique.

Chapter 4 has continued the optimization of the instantaneous learning algorithms with the algorithms of the second order. The chapter has started by highlighting the capability of the second order algorithms in dealing with the correlated observations, then it has focused on the covariance matrix behavior which represents an essential component of the second order algorithms. It has been found that the observation loss creates discontinuity in the covariance matrix structure which deteriorates the algorithm performance. Thus, a new version of RLS has been proposed known as RLS with weights update based on observations continuity (RLS-OC). The proposed algorithm updates the covariance matrix whenever a new observation is available while it updates the weights only when there is a sort of observations continuity in the current covariance matrix. The weights update based on observations continuity technique has been also applied to RIV which shares with RLS the same weights update principle. The optimized algorithms have shown quite satisfactory performance and high capability of reducing the effect of observations loss in comparison with their original versions.

Chapter 5 has ended the optimization of the instantaneous learning algorithms with the algorithm of the high order where the main focus was on the data store management (DSM) strategies which form a key ingredient when using the high order algorithms. Two data store management strategies known as error measurement (EM) and intelligent data store management (IDSM) have been introduced to improve the performance of the high order algorithms. EM strategy manages the store based on the output error value where it always ensures the existence of training pairs with smallest output error in the store. IDSM strategy manages the store by combining three strategies which are first in first out (FIFO), total correlation measure (TCM) and EM using weighting rules. IDSM strategy keeps the store information representative to the current state of the system and also reduces the undesired effects of both the correlation and the output error. New version of MA algorithm known as MA-EMSD has been proposed by adopting a combination of EM strategy and SD technique on MA algorithm. Another version of MA algorithm known as IMA has been proposed by adopting a combination of IDSM strategy and SDEDU method on MA algorithm. Both of the EMSD method and the intelligent approach have been applied to NMA algorithm. The optimized algorithms in general have shown good improvement in the performance compared to their original versions. The optimized algorithms based on the intelligent approach have shown better performance than the optimized algorithms based on EMSD method.

6.2 Future Work Directions

During random observations loss linear system tends to show nonlinearity and could be represented as though nonlinear system. Therefore, the work can be extended using nonlinear networks such as multilayer perceptron (MLP) or radial basis function (RBF) to represent the nonlinearity due to random observation loss.

The initial step towards progressing from this research work is to consider hybrid neural network architecture (see in Figure 6.1) where the hidden nodes can be used for nonlinear projection [14] and the linear weights can be adjusted using the optimized algorithms that have been discussed in this thesis.

The work can be extended in similar way by adopting the extreme learning machine (ELM) [20] which updates only the linear weights to increase the speed of the learning process. Using ELM observations vectors can be projected to form nonlinear transformation through the hidden nodes then the network output can be optimized using the optimized algorithms that have been discussed in this thesis.

The work can be further extended using hybrid learning policies network where both nonlinear and linear weights are updated concurrently [21] or using sliding window structure [22]. One big huddle in nonlinear weights update for lost observations will be the formulation of the partial weights derivatives for k step ahead prediction, where k now can be translated as the number of the lost observations. One possible remedy will be to employ the finite difference rule to estimate the k step ahead prediction which requires the partial derivates of the weights to be computed using central differentiation operations.

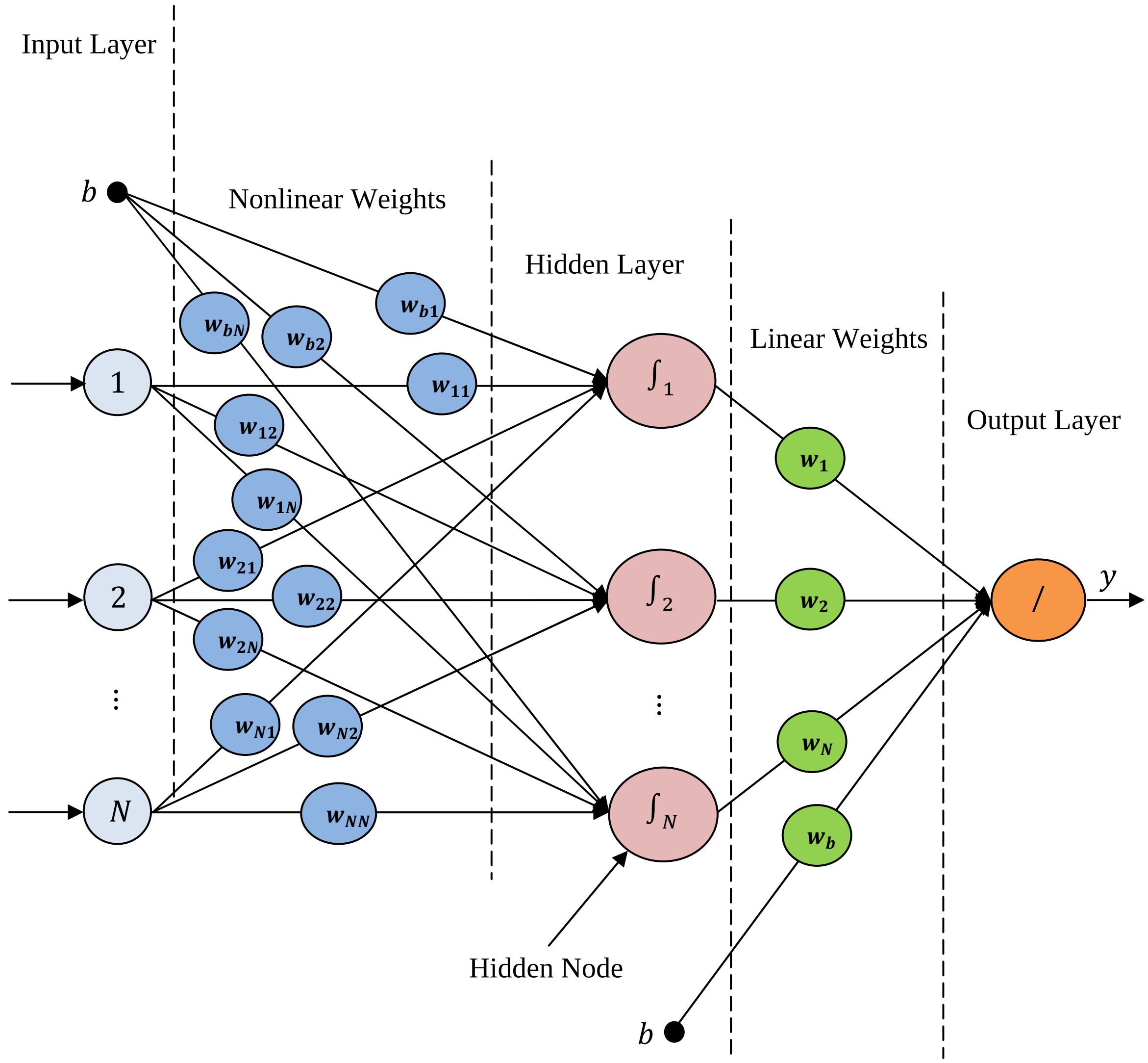


Figure 6.1: MLP Network Structure

REFERENCES

[1] A. Willig, K. Matheus and A. Wolisz, “Wireless technology in industrial networks,” *Proceedings of the IEEE on Special Issue on Industrial Communication Systems*, vol. 93, no. 6, pp. 1130–1151, June 2005.

[2] B. Sinopoli, L. Schenato, M. Franceschetti , K. Poolla, M.I. Jordan and S. S. Sastry, “Kalman filtering with intermittent observations,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, September 2004.

[3] X. Liu and A. Goldsmith, “Kalman filtering with partial observation losses,” *Proceedings of the IEEE conference on Decision Control*, vol. 4, pp. 4180 - 4186, 2004.

[4] L. Schenato, “Optimal estimation in networked control systems subject to random delay and packet loss,” *Proceedings of the IEEE Conference on Decision and Control*, pp. 5615 – 5620, December 2006.

[5] S. Hu and W. Yan, “Stability Robustness of Networked Control Systems with Respect to Packet loss,” *Automatica*, vol. 43, pp. 1243–1248, 2007.

[6] T. Soderstrom and P. G. Stoica, “Instrumental Variable methods for System Identification,” *Circuits Systems Signal Processing*, vol. 21, no. 1, pp. 1-9, 2002.

[7] P. N. Paraskevopoulos, Modern Control Engineering, Marcel Dekker, 2002.

[8] F. Gustafsson, Adaptive Filtering and Change Detection, John Wiley & Sons, 2000.

[9] B. Widrow and S. D. Stearns, Adaptive Signal Processing, Prentice Hall, Englewood Cliffs, 1993, 1985, NJ.

[10] B. Widrow and M. A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415-1442, September 1990.

[11] V. S. Asirvadam, "Online Learning and Construction for Neural Network," PhD thesis, School of Electrical and Electronics Engineering, Queen's University of Belfast, 2002.

[12] L. Ljung, System identification: theory for the user, Prentice-Hall, Second edition, 1999.

[13] P.E. An, M. Brown and C.J. Harris, "Aspects of Instantaneous On-line Learning Rules," *International Conference on Control, IEE Part*, vo1. 1, pp. 646 – 651, March 1994.

[14] P.C Parks and J. Militzer, "A comparison of five algorithm for the training CMAC memories for learning control systems," *Automatica*, vol. 28, No. 5, pp. 1027 - 1035, 1992.

[15] J. Rolf, System Modeling Identification, Prentice Hall Information and System Science Series, New Jersey, 1993.

[16] L. Ljung and T. Soderstrom, Theory and Practice of Recursive Identification, MIT Press, Cambridge, MA, 1983.

[17] S. Werner and P. S. R. Diniz, "Set-Membership Affine Projection Algorithm," *IEEE Signal Processing Letters*, vol. 8, no. 8, pp. 231-235, August 2001.

[18] V. S. Asirvadam, S. F. McLoone and G. W. Irwin, "Date store management in sliding window based on-line identification algorithms," *Proceedings of the Irish Signals and Systems Conference,* pp.123-128, June 2001.

[19] P.E. An, M. Brown, D. Mills and C.J. Harris, “High Order Instantaneous Learning Algorithms for On-line Training,” *International Symposium on Signal Processing, Robotics and Neural Networks*, pp. 525 – 529, 1994.

[20] G. B. Huang, Q. Y. Zhu, C. K. Siew, “Extreme learning machine*:* theory and applications*,”* *Neurocomputing*, vol. 70, pp. 489-501, 2006.

[21] V. S. Asirvadam, S. F. McLoone and G. W. Irwin, “Separable Recursive Training Algorithms for Feedforward Neural Networks,” *Proceedings of the International Joint Conference on Neural Network, World Congress on Computational Intelligence, Honolulu, Hawaii*, vol. 2, pp. 1212-1217, May 2002.

[22] V. S. Asirvadam, S. F. McLoone and G. W. Irwin, “ Recursive Sliding Window Based Neural Network Training with Data Store Management”, *Proceedings of the Irish Signals and Systems Conference, Cork*, pp. 149-154, June 2002.

[23] C. Brosilow and B. Joseph, Techniques of Model-Based Control, Prentice-Hall, 2002.

BIBLIOGRAPHY

A. Bernieri, G.Betta and A.Pietrosanto, “Optimum Sampling Frequency Choice for Accurate Identification of Electrical Systems,” *Proceedings of the IEEE Conference on Instrumentation and Measurement Technology*, pp. 10-13, May 1992.

A. Cervin, D. Henriksson, B.Lincoln, J. Eker and K. E Arzen, “How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime,” *The IEEE Control Systems Magazine*, vol. 23, no. 3, pp. 16-30, June 2003.

A. Feuer and R. Middleton, “Conditioning of LMS Algorithms with Fast Sampling,” IEEE Transactions on Signal Processing, vol. 43, no 8, pp. 1978-1981, August 1995.

C.F.N. Cowan, “An Introduction to Adaptive Filters,” *The IEE Younger Members Tutorial Seminar on DSP: Theory, Applications and Implementation*, pp. 2/1 - 2/6, Jan 1996.

C. L. Robinson and P. R. Kumar, “Control over Networks of Unreliable Links: Controller Location and Performance Bounds,” *5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops,* pp. 1-8, April 2007.

D. Alexander and M. Zayed, Adaptive Filtering Primer with Matlab, CRC Press Taylor & Francis Group, 2006.

D. j. G. Morrell and B. W. Hogg, “Identification and validation of turbogenerator models,” *Automatica Journal of IFAC*, vol. 26, no. 1, pp. 135-156 , January 1990.

E. Eweda, “Transient Performance Degradation of the LMS, RLS, Sign, Signed Regressor, and Sign–Sign Algorithms with Data Correlation,” *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. 46, no. 8, pp. 1055 – 1062, August 1999.

G. Long, F.Ling and J.G. Proakis, “The LMS Algorithm with Delayed Coefficient Adaptation,” *The IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 9, pp.1397 – 1405, September 1989.

I. Eker, “Open-loop and Closed-loop Experimental On-line Identification of a Three-mass Electromechanical System,” *Mechatronics*, pp. 549-565, 2004.

J. A. B. Wills, “Data sampling and data recovery,” *Experiments in Fluids*, vol. 12, pp. 23-28, December 1991.

J. Colandairaj, G.W. Irwin and W.G. Scanlon, “Understanding wireless networked control systems through simulation,” *the IEEE Computing and Control Journal*, vol. 16, no. 2, pp. 476–481, December 2005.

J. Eker, A. Cervin and A. Horjel, “Distributed Wireless Control Using Bluetooth,” *IFAC Conference on New Technologies for Computer Control*, November 2001.

J. P. Corriou, Process Control: Theory and Applications, Springer-Verlag, 2004.

J. R. Treichler, JR. C. Richard Johnson and M. G. Larimore, Theory and Design of Adaptive Filters, Prentice Hall, 2001.

J. Xin, H. Tsuji and A. Sano, “Optimal sampling interval for system identification based on decimation and interpolation,” *Proceedings of the IEEE Conference on Control Theory and Applications*, vol. 142, no. 1, pp. 15 – 22, January 1995.

L. Ljung, “Issues in system identification,” *The IEEE Control Systems Magazine*, vol. 11, no. 1, pp. 25 - 29, Jan 1991.

L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla and S. Sastry, “Foundations of control and estimation over lossy networks,” *Proceedings of the IEEE on Special issue on emerging technologies of networked control systems*, vol. 95, no. 1, pp.163-187, January 2007.

M. Karlsson and M. Covell, “Dynamic Black-Box Performance Model Estimation for Self-Tuning Regulators,” *Proceedings of the Second International Conference on Autonomic Computing*, pp. 172-182, June 2005.

O. C. Imer, S. Yuksel, and T. Basar, “Optimal control of LTI systems over unreliable communication links,” *Automatica*, vol. 42, pp. 1429-1440, September 2006.

P. A. Kawka, and A. G. Alleyne, “Wireless Servo Control for Electro-Hydraulic Positioning,” *Proceedings of the 2004 Bath Workshop on Power Transmission and Motion Control*, pp. 159-172, September 2004.

P.C. Young, Recursive Estimation and Time Series Analysis, Springer-Verlag, Berlin, 1984.

P. Lindskog and L. Ljung. “Tools for semi physical modeling”, *Proceedings of the 10th IFAC symposium on Identification (SYSID'94)*, vol. 3, pp. 237-242, July 1994.

R.D. Poltmann, “Conversion of the delayed LMS algorithm into the LMS algorithm,” *The IEEE Signal Processing Letters*, vol. 2, no. 12, pp. 223,December 1995.

S. Haykin, Adaptive Filter Theory, Prentice Hall, Fourth Edition, 2002.

APPENDIX A: BENCHMARK TEST PROBLEMS

This section provides a brief description of various benchmark modeling problems used to evaluate the performance of the optimized algorithms in this thesis. In all cases the performance is measured in terms of the instantaneous normalized mean squared error (NMSE) computed over a representative validation data set (φ_{n}, y_n) as follows:

$$NMSE = J(\theta_k) = \frac{\sum_{n=1}^{N_v} \left(f\left(\varphi_{n}, \theta_k\right) - y_n\right)^2}{\sum_{n=1}^{N_v} (y_n)^2} \tag{A.1}$$

where N_v is the size of the validation data set. In addition to plotting the instantaneous NMSE evolution over time another two measures are frequently used to provide summary of the performance information, namely the average and the standard deviation of the instantaneous NMSE over the last p iterations. These are defined as follows:

$$E_p\big(J(\theta)\big) = \frac{1}{p} \sum_{k=N_v-p}^{N_v-1} J(\theta_{k+1}) \tag{A.2}$$

$$Std_p\big(J(\theta_k)\big) = \sqrt{\frac{1}{p} \sum_{k=N_v-p}^{N_v-1} \left[J(\theta_{k+1}) - E_p\big(J(\theta_k)\big)\right]^2} \tag{A.3}$$

where N_v is the total number of iteration performed and $p \leq N_v$.

A.1 Test Problem I: Single Tank System

Modeling the dynamics of single tank system is used as a benchmark to show the performance of the optimized algorithms in this thesis during random observations loss. The system [22] is shown in Figure A.1, where the inlet flow $\left(f_{in}(t)\right)$ is regulated by a valve while the outlet flow is proportional to the height given by $f_{out}(t) = Kh(t)$, k depends on the outlet valve opening flow (here k was set equal to 2 m^3/s/m). The identification problem considered here is the prediction of h given f_{in} .

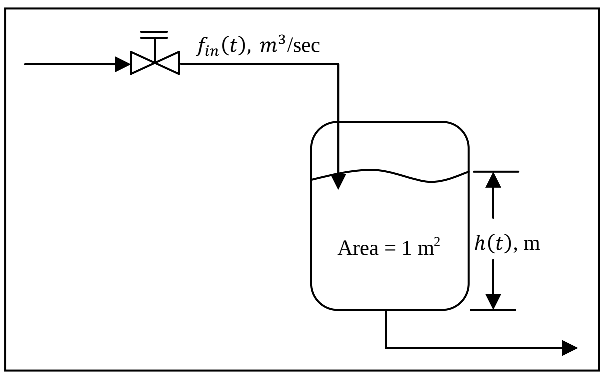


Figure A.1: Single Tank System

This system is governed by the following equation:

$$\frac{dh}{dt} = f_{in}(t) - Kh(t) \tag{A.4}$$

or

$$h(t) = \int [\, f_{in}(t) - Kh(t)]\, dt \tag{A.5}$$

A.2 Test Problem II: Skeleton Linear Systems

Since the weights convergence behavior of instantaneous learning algorithms is best understood when considering systems with known weights, three first order discrete-time systems are included in test problem II to view their weights trajectories graphically at different locations in the thesis. The systems are defined by the following equations:

- System A:

$$y_k = 1\, y_{k-1} + 0.2\, u_{k-1} \tag{A.6}$$

- System B:

$$y_k = 0.9\, y_{k-1} + 0.3\, u_{k-1} \tag{A.7}$$

- System C:

$$y_k = 0.75\, y_{k-1} + 0.15\, u_{k-1} \tag{A.8}$$

This benchmark problem is used mainly to show the weights trajectories of the instantaneous learning algorithms when the presented training data is correlated.

APPENDIX B: PUBLICATIONS

This section lists down the papers that have been published form this work.

1. M. Jabralla, V.S. Asirvadam and N. Saad, "Systems Modeling with Lost Information Packets Using ARMAX Model," *Proceedings of the International Conference on Intelligent & Advanced Systems, Kuala Lumpur,* 25th-28th November 2007.

2. M. Jabralla, V.S. Asirvadam and N. Saad, "Instantaneous Modeling of Linear Network on Simulated Wireless Environment," *Proceedings of the 2nd International Conference on Computer and Communication Engineering, Kuala Lumpur,* 13th-15^{h} May 2008.

3. Musab Jabralla Elamin and Vijanth S. Asirvadam, "Instantaneous System Modeling over Lossy Network," *Proceedings of Student Conference on Research and Development 2008, Johor, Malaysia,* 26th – 27th November 2008.

4. Musab Jabralla Elamin and Vijanth S. Asirvadam, "RLS based on Observations Continuity for Instantaneous Wireless System Modeling," *Proceedings of Student Conference on Research and Development 2008, Johor, Malaysia,* 26th – 27th November 2008.