ADAPTIVE ESTIMATION USING STATE-SPACE METHODS

By

MUHAMMAD SALMAN

A DISSERTATION

Submitted to

National University of Sciences and Technology

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Supervised by

DR MOHAMMAD BILAL MALIK

College of Electrical and Mechanical Engineering

National University of Sciences and Technology, Pakistan

2009

بسم الله الرحمن الرحيم

In the name of Allah, the most Merciful and the most Beneficent

ABSTRACT

ADAPTIVE ESTIMATION USING STATE-SPACE METHODS

By

MUHAMMAD SALMAN

This thesis focuses on the development of new variants of adaptive filters. Built around state-space framework, the proposed filters are especially suitable for applications like tracking, output feedback control and recursive spectrum estimation. They operate without prior knowledge of process and observation noise statistics and exhibit good stability properties. The development in this thesis can broadly be classified into state-space least mean square (SSLMS) and finite memory least-squares filters.

SSLMS is a generalization of the well-known least mean square (LMS) filter. Incorporating linear time-varying state-space model of the underlying environment, SSLMS exhibits marked improvement in its tracking performance over the standard LMS. An extension of SSLMS is SSLMS with adaptive memory (SSLMSWAM). SSLMSWAM iteratively tunes the step-size parameter by stochastic gradient method in an attempt to yield its most appropriate value. This filter is useful for situations where a suitable value of step-size parameter is difficult to obtain beforehand.

Recursive nature of an adaptive filter brings with it stability issues. The concept of finite memory (or receding horizon) for an adaptive filter is appealing because it ensures stability. This motivates the development of finite memory filters, both for

unforced and forced systems. Finite impulse response (FIR) adaptive filter, built around structure of an unforced system, uses weighted observations on a finite interval. Uniform weighting of the observations results in rectangular RLS (RRLS). Additional flexibility is achieved by developing an adaptive memory variant of FIR adaptive filter. Similar to SSLMSWAM, the data window size is iteratively tuned so as to minimize the prediction error. For the forced system case, a useful solution in the form of receding horizon state observer is obtained. It finds utility in output feedback control of linear time-varying systems. An insight into convergence properties of finite memory based filters is provided by the convergence analyses.

Spectrum update with the arrival of new data is a desirable feature in real-time spectrum estimation applications. The mathematical equivalence of RRLS resonator bank and recursive discrete Fourier transform (DFT) gives the rationale for using the newly developed filters for recursive spectrum estimation. A symmetric windowed variant of RRLS called 'truncated exponential RLS (TERLS)' is useful for reducing spectral leakage. Same is true for an SSLMS resonator, which has an attractive feature that spectral side levels and main lobe width may be reduced simultaneously by reducing the step-size parameter. The higher order resonator (HOR), constructed from several SSLMS resonators, exhibits close resemblance to an ideal (rectangular) frequency bin, thus minimizing spectral leakage and increasing resolution.

To

My parents and my wife

ACKNOWLEDGEMENTS

I would like to acknowledge the guidance and cooperation of all my teachers whose earnest efforts have made possible the successful completion of this work. The knowledge I gained from them proved helpful throughout the course of my studies. The person to whom I would like to address very special thanks is my supervisor Dr Mohammad Bilal Malik, both for accepting me as a PhD student and for providing valuable guidance, advice and support over the years resulting in the completion of this thesis.

I would like to express my sincere gratitude to Dr Mohammad Ali Maud, Dr Ejaz Muhammad and Dr Khalid Munawar for their precious time, attention and the inspiration I have received from them. I am very thankful to Dr Ziauddin Ahmad, Dr Khurram Waheed and Dr Ijaz Mansoor Qureshi for evaluating this work and providing valuable comments for its improvement. I am grateful to Anees Ahmed with whose help I was able to join the PhD programme. I thankfully acknowledge the support I received from Engr Najeeb Tariq and Engr Muhammad Shahid. I am thankful to Dr Abdul Ghafoor who has been very helpful all the way along. I gratefully appreciate the cooperation I received from Dr Nawar Khan and Dr Muhammad Almas Anjum.

I address special thanks to Muhammad Asim Ajaz and Tahir Naseer Qureshi who have been very supportive during the course of my studies. I am grateful to my colleagues Salman Muhammad Aslam, Arbab Waleed Ahmad, Aleem Mushtaq, Muhammad Omer Bin Saeed, Fahad Mumtaz Malik, Aamir Hussain and Usman Ali for their cooperation and beneficial academic discussions I have had with them.

I am grateful to all my family members for their support. My parents, aunt Khaleda Begum, uncle Maqsood Ali (late) and my mother in law have been sources of inspiration for me and their prayers have accompanied me throughout my life. I am indebted to my wife who has been very supportive and has shown extraordinary patience all the way along.

I would also like to gratefully acknowledge Higher Education Commission (HEC), Pakistan for the financial support they provided for my studies and research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Estimation

The problem of estimation of an unknown quantity of interest is frequently encountered in various engineering disciplines. Based on a *measured* dataset (possibly corrupted by observation noise), an estimate of the unknown quantity is obtained using some estimation technique. The quality of an estimate is judged by how close it is to the actual value, in some statistical sense. The three basic variations on the notion of estimation are filtering, smoothing and prediction [13]. *Filtering* uses present and past observations to yield *present* estimate of the unknown quantity; a causal operation. *Smoothing* uses present and past observations to obtain a *past* estimate; a non-causal operation. *Prediction* uses present and past observations to yield a *future* estimate; a causal operation.

Among various estimation methods that have historically been used, least squares methods have received considerable attention in the past and have been used to solve the problem of estimation and tracking. The method of least squares was first used by Gauss in 1795 to study the motion of heavenly bodies, though it was first published by Legendre in 1805 who independently invented the method ([13], [19]). The first studies of least squares estimation in stochastic processes were made by Kolmogorov, Krein and Wiener [19]. The work of Wiener was independent from the work of Kolmogorov and Krein. Wiener worked on continuous-time linear prediction problem and gave explicit formula for the optimum predictor. The problem of estimating a process corrupted by additive noise was also considered by Wiener [13]. Thereafter, Levinson gave discrete-time counterpart of the Wiener filtering problem [13].

Kalman gave solution to linear filtering problem in state-space framework ([3], [19], [20]). Kalman filter has been successfully used in various applications. A practical difficulty faced in the implementation of optimal filters is the requirement of *a priori* information about the statistics of the data ([3], [20]). This *a priori* information may not be available in certain cases. Least mean square (LMS) and recursive least squares (RLS) are the two fundamental adaptive filtering algorithms [13] that work satisfactorily in the absence of this *a priori* information. These algorithms find common application areas, despite having different weight update structures. LMS has lower convergence rate and larger steady state mean square error, as compared to RLS. Nevertheless, LMS is more robust than RLS and is H_∞ optimal [11].

1.2 Tracking

Adaptive filters have played a vital role in the development of a wide variety of systems, for the last three decades. The philosophy of adaptive filters revolves around recursive least squares (RLS) and least mean square (LMS) [13]. One of the major applications of LMS and RLS has been in tracking time variations in a signal. Tracking performance of LMS and RLS has received considerable interest in the literature ([2], [8], [9], [13]). Model dependent nature of tracking problem requires that designer be given freedom to select a model that closely matches the underlying environment. Operating in nonstationary environment, LMS (being model independent) provides some advantage over RLS, that assumes multiple linear regression model [13]. In case of model mismatch, the tracking performance of RLS may degrade, whereas LMS shows robust behavior. Model dependent nature of the tracking problem makes it especially sensitive to this limitation. Consequently, tracking performance of LMS and RLS have been thoroughly explored ([1], [2], [8], [9] and [13] etc.).

If the mechanism responsible for generating a signal is known *a priori*, it may be used to advantage in tracking the signal closely in the presence of observation noise. In case of LMS and RLS, there is no direct way to incorporate such *a priori* information into their mathematical formulations. On the other hand, the filters that are built around state-space framework can incorporate such *a priori* information. Some examples are Kalman filter [13] and state-space RLS (SSRLS) [35]. SSRLS takes into consideration the state-space model of the system, thus resulting in a very useful generalization of the standard RLS. SSRLS shows considerable improvement in tracking performance over standard RLS and LMS ([35], [36], [42]). Development of SSRLS with adaptive memory (SSRLSWAM) [39] adds a level of versatility to this philosophy. SSRLSWAM proves to be an effective tracker even in difficult scenarios [39]. However, SSRLS and its variants achieve this performance boost at the expense of increased computational effort. Sayed and Kailath have also given a state-space formulation for RLS which is, in fact, a representation of the standard RLS [55].

1.2.1 The Signal Model

Many signals encountered in physical applications can be modeled as output of unforced neutrally stable linear systems e.g. sinusoidal signal [34]. The development of adaptive filtering algorithms in this thesis is based on following unforced discrete-time state-space model [35]

$$\begin{aligned} x[k+1] &= A[k]\,x[k] \\ y[k] &= C[k]\,x[k] + v[k] \end{aligned} \tag{1-2.1}$$

where $x[k] \in \mathbb{R}^n$ is the state vector at time *k,* the elements of which are called state variables, which may also be referred to as the process states. $y[k] \in \mathbb{R}^m$ is the observation vector at time *k.* We assume that the maximum number of outputs of the system is less than or equal to the states i.e. $m \le n$. This is a logical assumption as a system with $m > n$ can be simplified to the one with $m \le n$ without the loss of any information about the states [52].

The system matrix $A[k]$ and the output matrix $C[k]$ may be stochastic or deterministic depending on the nature of the problem. $C[k]$ is assumed to be full rank, which is a reasonable assumption. To see this, consider the case when $C[k]$ is deterministic and rank deficient. We can reduce the number of outputs to make a new full rank output matrix, without the loss of any information about the states. On the other hand a stochastic $C[k]$ is full rank by virtue of unavoidable presence of white noise. Pair ($A[k], C[k]$) is assumed to be *l*-step observable [39]. Observation noise is represented by $v[k]$. We will have more to say about the model state-space model (1-2.1) in Section 2.1.

1.2.2 The Tracking Problem

We want the estimated output $\hat{y}[k] = C\,\hat{x}[k]$ to asymptotically track the uncorrupted output of the system. Mathematically stating

$$\lim_{k\to\infty}\left[\hat{y}[k]-Cx[k]\right]=0 \tag{1-2.2}$$

1.3 Feedback Control Systems

Feedback control problems can be broadly classified into two categories viz state feedback and output feedback. The former one assumes complete knowledge of the system states, whereas in the latter case only the plant outputs are available for measurement. State feedback control design is relatively a simpler and more effective proposition of the two categories. In case of an autonomous plant, the problem is further simplified and the feedback is static in nature. In practical situations, however, all or some system states may be either inaccessible or not feasible for measurement due to economic/practical constraints. Moreover, sensors used to measure system states introduce their own errors. In such cases, output feedback approach is an appealing alternative.

1.3.1 State Observation

The philosophy is to design a state estimator (observer) based on information about the system dynamics, its inputs and outputs. This breakthrough concept was initially proposed by Luenberger ([30] – [32]). For the case when system parameters are not known, an adaptive approach to state observation may be used wherein system parameters are estimated on-line (see for example [4], [14], [25] and [57]). Some robust state observation schemes that work well in the presence of model uncertainties, plant disturbances have been reported (see for example [16], [46] and [58]). The same concept has been proposed for a class of nonlinear systems [22]. In case of presence of process and/or sensor (observation) noise, Kalman filter enjoys a celebrated status. In fact, there is a history of research in this area [19]. Whereas there have been numerous efforts in observer design, all these approaches have associated stability issues due to their recursive nature that must be addressed when applying them to a specific problem. The stability is however guaranteed if the observer is based on a finite set of system inputs and outputs; the so called *receding horizon*.

1.3.2 Receding Horizon Approach

The concept of finite observations based state observers was discussed by Ling et al. [29]. They have derived a couple of recursive solutions for an unforced system in the absence of process noise and have suggested the conditions under which the observer remains stable. The point that the observer is *inherently* stable in view of finite observation length was later appreciated by Kwon et al. [27]. They also suggested a receding horizon unbiased FIR (RHUF) filter for the state-space framework. Their formulation however requires *a priori* knowledge of second order statistical characterization of process noise and observation noise in order to start the recursion. Both Ling et al. [29] and Kwon et al. [27] have considered time-invariant case only. An earlier paper by Kwon et al. [28] describes Kalman FIR filter for time-invariant case that requires *a priori* knowledge of second order statistics of process and observation noises. Using maximum likelihood criteria, a very early framework for optimal filtering based on finite observations was proposed by Jazwinski [18]. However, his approach also requires *a priori* knowledge of second order statistical characterization of observation noise.

1.3.3 The System Model

The development of receding horizon state observer in this thesis is based on following discrete-time state-space model

$$\begin{aligned} \boldsymbol{x}[k+1] &= A[k]\boldsymbol{x}[k] + B[k]\boldsymbol{u}[k] \\ \boldsymbol{y}[k] &= C[k]\,\boldsymbol{x}[k] + \boldsymbol{v}[k] \end{aligned} \tag{1-3.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state, $\boldsymbol{y} \in \mathbb{R}^m$ is the output and $\boldsymbol{u} \in \mathbb{R}^s$ is the input. The system matrices, which are $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times s}$ and $C \in \mathbb{R}^{m\times n}$, are assumed to be known and deterministic. k is the discrete time index with $k \in \mathbb{Z}$. The output is corrupted by an inaccessible observation noise $\boldsymbol{v}[k]$ such that $\boldsymbol{v} \in \mathbb{R}^m$. We will have more to say about the model (1-3.1) in Section 5.1.

1.3.4 Receding Horizon State Observation Problem

Given uncorrupted inputs and outputs of the system based on (1-3.1), we want the estimated states $\hat{\boldsymbol{x}}[k]$ equal system states $\boldsymbol{x}[k]$ for time index k exceeding or equaling horizon depth p. Mathematically stating

$$\hat{\boldsymbol{x}}[k] = \boldsymbol{x}[k] \qquad \text{for } k \geq p \tag{1-3.2}$$

1.4 Spectrum Estimation

Spectrum estimation is a fundamental tool in signal analysis. Frequency domain representation helps better understand the nature of the signal. A signal characteristic that may not be evident in time domain sometimes becomes clear in frequency domain. Classical methods of spectrum estimation are essentially based on Fourier analysis that make no assumption about how the data was generated and hence are called non-parametric. Bartlett, Blackman & Tukey, and Welch may be regarded as pioneers of discrete Fourier analysis [34]. These methods require long data records in order to obtain necessary frequency resolution, which is required in many application. Moreover, windowing of time domain data results in spectral leakage that may obscure weak signals present in the data. Several window functions have been suggested to minimize spectral leakage [48]. Picket fence effect is yet another issue often encountered in spectrum estimation, whose *ideal* solution is to construct rectangular frequency bins [33].

Some of the modern schemes that rely on parametric modeling of the signal produce high resolution estimates. The modeling approach eliminates the need for window functions and hence avoids spectral leakage. Details can be found in [21] and the references therein. In many applications, an updated estimate of the spectrum is required with every new data sample observed. To meet this requirement, some recursive solutions have been suggested by Hostetter [15], Varkonyi-Koczy [23] and G. Peceli [50].

1.5 The Need for New Developments

In view of the discussion in preceding sections, new developments are considered necessary as discussed below:

- A new tracking algorithm built around state-space framework that is comparable in performance to existing well-known tracking algorithms, with reduced computational effort. The development of state-space least mean square (SSLMS), which is a generalization of standard LMS, achieves this aim. Moreover, the adaptive memory variant of SSLMS finds usefulness in the applications where most suitable value of step-size parameter is difficult to obtain *a priori.* This is especially true in non-stationary operating environment.

- For satisfactory operation in the applications where standard LMS, RLS may fail or perform poorly due to stability issues, a filter that guarantees stability is an attractive choice. Motivated by receding horizon philosophy, a finite impulse response (FIR) adaptive filter is developed, that is inherently stable. FIR adaptive filter is built around state-space framework of an unforced system and works in the absence of process and observation noises.
- State observation of forced linear *time-varying* systems, while ensuring stability of the observer, is a challenging task. A receding horizon state observer is developed that is inherently stable by virtue of finite horizon of inputs and outputs. It operates on a finite set of system inputs and outputs to estimate unknown states of the system.
- The existing non-parametric spectrum estimation techniques e.g. Fourier transform based methods suffer from spectral leakage and picket fence effect [33]. Based on recursive least squares approach, some new methods are developed that yield improved results. The development of truncated exponential RLS (TERLS) resonator helps reduce spectral leakage by implementing a recursive exponential window. The SSRLS/SSLMS resonators achieve simultaneous reduction of spectral main lobe width and side-levels, an attract feature for high resolution spectrum estimation. The development of higher order resonator (HOR) achieves almost rectangular frequency bins, thus minimizing spectral leakage and picket fence effect.

1.6 Overview of the Thesis

The thesis is organized in eight chapters including the Introduction. State-space least mean square (SSLMS) is developed in CHAPTER 2. The discussion on the signal model (1-2.1) is extended to highlight the properties of system matrix $A[k]$. The model (1-2.1) forms the basis of further development. The derivation of SSLMS is based on the minimum norm solution of an underdetermined system of linear equations. The solution with normalization factor is called state-space normalized LMS (SSNLMS), where omitting this factor gives simply SSLMS. The choice of step-size parameter μ is important in defining performance of SSLMS. It is shown that the linear regression model is a special case of the general state-space model used in derivation of SSLMS. Steady-state solution of SSLMS, which has reduced computational effort, is developed next. A noisy sinusoid tracking example concludes the chapter, where a performance comparison of SSLMS with SSRLS, RLS and LMS is drawn.

CHAPTER 3 develops adaptive memory variant of SSLMS i.e. SSLMS with adaptive memory (SSLMSWAM). Since the most appropriate value of step-size parameter μ may not be known beforehand, the idea is to iteratively tune μ so as to minimize the mean square value of the prediction error. SSLMSWAM uses updated value of μ in its recursion. This is particularly useful in a time-varying setting. This results in overall performance improvement, though at the cost of increased computational complexity. The examples of tracking noisy chirp signal and Van der Pol oscillations follow. The performance of SSLMSWAM is also compared with that of

SSLMS, SSRLS, SSRLSWAM, RLS and LMS. The paper concludes with a detailed discussion of computational complexities of SSLMS family of algorithms.

CHAPTER 4 develops finite impulse response (FIR) adaptive filter which is built around state-space framework of an *unforced* system (1-2.1). The filter is inherently stable by virtue of its FIR 'observations to state estimate' mapping. This is also confirmed by its transfer function representation, which contains *zeros* only. We derive both time-varying and time-invariant solutions. We also give alternate forms of the filter i.e. current estimator and predictor form, which are useful for specific applications. Under a given set of conditions, the FIR adaptive filter and SSRLS are shown to be equivalent. First and second order converge analysis follows. The FIR adaptive filter performs satisfactorily in applications where standard LMS, RLS may fail or perform poorly. Performance of the filter is demonstrated using an example considering tracking problem of Van der Pol oscillations. A performance comparison with LMS and RLS is also given.

Motivated by the development of FIR adaptive filter, the receding horizon state observer for linear time-varying systems is presented in CHAPTER 5. Key benefits of the observer are its inherent stability and ability to operate in time-varying and time-invariant settings alike. The inherent stability of the observer is attributed to finite horizon of inputs and outputs used to generate state estimate. The development of the observer is based on the system model (1-3.1). Using method of least squares, both batch-processed and recursive solutions are derived. Time-invariant results are easily obtained from the derivations for time-varying case. The observer does not require *a priori* knowledge of properties of process and/or observation noise. First and second order convergence analyses of the observer follow that give insight into its convergence properties. The performance of the observer is demonstrated with the help of an example considering regulation problem of a gyroscope, a linear time-varying system.

In CHAPTER 6, we address non-parametric spectrum estimation problem using recursive least squares methods. While we develop various estimation techniques, we look for the desirable properties like reduced spectral leakage, simultaneous reduction of spectral main lobe & side-levels and reduced picket fence effect. The rationale for using these techniques for spectrum estimation is provided by the rectangular RLS (RRLS) (FIR adaptive filter with uniform weighting of observations) resonator that is shown to coincide with discrete Fourier transform (DFT). The spectral leakage that may occur due to rectangular windowing in RRLS is addressed by deriving a recursive exponential window based algorithm, called truncated exponential RLS (TERLS). We also use resonator banks based on SSLMS and SSRLS. With SSRLS and SSLMS, it is possible to simultaneously reduce spectral side-levels and main lobe width. This contrasts with standard Fourier transform based methods where increase in spectral resolution does not yield simultaneous reduction in side-levels and main lobe width. We use this property of SSRLS/SSLMS to develop higher order resonator (HOR) in which a number of closely spaced SSRLS (or SSLMS) based resonators contribute to a single frequency bin. This provides the key benefits like reduced spectral leakage and picket fence effect. An example considering spectrum estimation of a tone at non-orthogonal frequency demonstrates the capabilities of recursive spectrum estimation methods presented.

The last chapter on conclusions and a few suggestions for future considerations concludes the thesis.

CHAPTER 2

STATE-SPACE LEAST MEAN SQUARE

In this chapter, we develop state-space least mean square (SSLMS) ([37], [45]) by incorporating the linear state-space model of the environment. This offers two plus points. Firstly any causal linear system can be represented by a state-space model, thus a designer is not restricted to the linear regression model. Secondly, the multiple-input multiple-output (MIMO) nature of state-space model allows handling vector observations. The standard RLS and LMS, on the other hand, only deal with scalar observations. Appropriate to the nature of generalization, we use the term state-space least mean square (SSLMS).

The development begins with a discussion of the state-space model of an unforced linear time-varying discrete system. The output of the system that may have been corrupted by observation noise is assumed to be available for measurements. This model forms the basis of further development. The derivation of SSLMS, based on the minimum norm solution of an underdetermined system of linear equations, uses only latest observation in the recursive algorithm. The solution with normalization factor is called state-space normalized LMS (SSNLMS), where omitting this factor gives us simply SSLMS. It is shown that the linear regression model is a special case of the general state-space model used in derivation of SSLMS. This in fact, shows that SSLMS is a true generalization of the standard LMS. State-space formulation of SSLMS also makes it possible to use it in state estimation in control systems.

Steady-state solution of SSLMS is also developed. The resultant time-invariant filter is computationally less intensive as compared to the time-varying solution, since some of the computations can be done offline. The transfer function of the solution has also been derived.

A noisy sinusoid tracking example concludes the chapter, where a performance comparison of SSLMS with SSRLS, RLS and LMS is drawn.

2.1 State-Space Model

State-space model (1-2.1) of an unforced discrete-time system was presented in Section 1.2.1. In this section, we extend the discussion on the model (1-2.1). It is customary to assume $v[k]$ to be a zero-mean white process, although these assumptions do not effect the derivations and development done in this thesis. Such assumptions on the observation noise are important for analyzing the performance of the filters presented here. The state-transition matrix for the system (1-2.1) is given as follows [52]

$$A[k,j]=\begin{cases} A[k-1]A[k-2]\cdots A[j], & k>j \\ I, & k=j \end{cases} \tag{2-1.1}$$

The system matrix $A[k]$ is assumed to be invertible for all k which results in the following properties [52]

$$\begin{aligned} A^{-1}[k,j] &= A[j,k], && \forall\, j,k \\ A[k,i] &= A[k,j]A[j,i], && i\le j\le k \\ A[k+1,k] &= A[k] \end{aligned} \tag{2-1.2}$$

The absence of 'process noise' or any other deterministic inputs poses a question about the usefulness of (1-2.1). However, this framework is general enough to address adaptive filtering applications as would be shown later in this chapter. Moreover, this concept is a familiar one in the context of exosystems used in servomechanism problems [7]. The reference signals (to be tracked) and disturbances (to be rejected) are modeled using systems similar to (1-2.1). In all these cases, the system matrix $A[k]$ is neutrally stable. For a time-invariant system matrix $A = A[k]$, neutral stability implies that all of the eigenvalues of A are strictly on the unit circle. An extension of this philosophy to nonlinear systems can be found in the regulation problem, where once again the references/disturbances are modeled by neutrally stable exosystems [17]. Neutral stability of $A[k]$ rules out existence of unstable or exponentially stable states and hence the concern about applicability of (1-2.1) to physical systems. Having said that, the theory developed in this and next chapter however, does not make any assumptions about the neutral stability of $A[k]$. This is only a matter of usefulness of models like (1-2.1) in practical applications.

2.2 State Estimator

Suppose that the observations $y[k]$ start appearing at time $k=1$. The initial state vector is $x[0] = x_o$ and is not known. The observability assumption allows us to design a state estimator. The idea is to generate the estimated state vector $\hat{x}[k]$ making use of the observations $y[1], y[2], \cdots, y[k]$.

The system equation (1-2.1) enables us to compute the predicted state estimate at time k (using observations up to time $k-1$) as follows

$$\bar{x}[k] = A[k-1]\hat{x}[k-1] \tag{2-2.1}$$

The prediction error can now be defined as

$$\varepsilon[k] = y[k] - \bar{y}[k] \tag{2-2.2}$$

with

$$\bar{y}[k] = C[k]\bar{x}[k] \tag{2-2.3}$$

as the predicted output. The prediction error is also referred to as innovations in the realm of Kalman filtering [13]. We can also define the estimation error as

$$e[k] = y[k] - \hat{y}[k] \tag{2-2.4}$$

where $\hat{y}[k] = C[k]\hat{x}[k]$ is the estimated output. One of the well-known estimator forms is [52]

$$\hat{x}[k] = \bar{x}[k] + K[k]\varepsilon[k] \tag{2-2.5}$$

where $K[k]$ is the observer gain, which is to be determined by different methods presented in this chapter.

2.3 Observer Gain

In this section we derive a generalized version of LMS viz SSLMS which incorporates model dynamics as given in (1-2.1). The discussion begins with relating the prediction error (2-2.2) and estimation error (2-2.4) as follows (2-3.1)

$$e[k] = \varepsilon[k] - C[k]\delta[k] \tag{2-3.1}$$

where $\delta[k]$ is defined as

$$\delta[k] = \hat{x}[k] - \bar{x}[k] \tag{2-3.2}$$

The assumption that $C[k]$ is full rank makes it possible to choose $\hat{x}[k]$ such that

$$e[k] = 0 \tag{2-3.3}$$

which gives

$$\varepsilon[k] = C[k]\delta[k] \tag{2-3.4}$$

If $m < n$ then there are infinitely many choices of $\hat{x}[k]$ that satisfy (2-3.3). We resort to minimum norm solution of (2-3.4), which minimizes $\delta[k]$ in (2-3.2) subject to the constraint $e[k] = 0$ [13]. We get

$$\delta[k] = C^T[k](C[k]C^T[k])^{-1}\varepsilon[k] \tag{2-3.5}$$

From (2-3.2) and (2-3.5)

$$\hat{x}[k] = \overline{x}[k] + C^T[k](C[k]C^T[k])^{-1}\varepsilon[k] \tag{2-3.6}$$

Comparing (2-3.6) with (2-2.5), the observer gain $K[k]$ according to the method of minimum norm solution comes out to be

$$K[k] = C^T[k](C[k]C^T[k])^{-1} \tag{2-3.7}$$

As apparent from its form, the gain in (2-3.7) has a limited scope. In order for a state estimator to be valid, the map from output of the system (which is the input of the estimator) to state estimates should be controllable. Alternately stating, $(A[k-1] - K[k]C[k]A[k-1], K[k])$ pair should be controllable. The choice of gain as given in (2-3.7) does not guarantee that this requirement will always be satisfied. Furthermore, a designer prefers to have a control of the rate of convergence, which is done through step-size parameter in the standard LMS [13]. In view of these considerations, we introduce a step-size parameter μ and matrix G to arrive at a more useful expression as follows

$$K[k] = \mu G C^T[k](C[k]C^T[k])^{-1} \tag{2-3.8}$$

The matrix G is chosen so as to have a valid estimator (i.e. controllable pair $(A[k-1] - K[k]C[k]A[k-1], K[k])$, whereas rate of convergence is controlled through μ.

In certain cases like sinusoidal model ([1], [35]), the controllability condition exists without the matrix G in (2-3.8). On the other hand, constant velocity and constant acceleration models ([1], [35]) do not fulfill this requirement and hence a designer has to choose a G for the estimator to be valid. The actual choice of this matrix depends on the nature of the problem. One simple approach is illustrated in Section 3.4, where the first column of G consists of non-zero entries. The rest are all zeroes.

Finally, for the cases where invertibility of $C[k]C^T[k]$ can not be ensured, we may use a small number γ, which modifies (2-3.8) into

$$K[k] = \mu G C^T[k](\gamma I + C[k]C^T[k])^{-1} \tag{2-3.9}$$

This arrangement allows us to handle problems where $C[k]$ may become rank deficient for a short interval. $\gamma = 0$ is used in situations where $C[k]$ is guaranteed to be full rank. Defining

$$\gamma I + C[k]C^T[k] \tag{2-3.10}$$

as a normalization factor, the algorithm comprising (2-2.1) − (2-2.3), (2-2.5), (2-3.9) is termed as state-space normalized LMS (SSNLMS). It is apparent from (2-3.9) that an $m \times m$ matrix is required to be inverted. A simplification in this algorithm results by removing the normalization factor, which reduces the observer gain to

$$K[k] = \mu G C^T[k] \tag{2-3.11}$$

The algorithm (2-2.1) − (2-2.3), (2-2.5), (2-3.11) is accordingly called state-space LMS (SSLMS). An analogy of SSLMS with standard LMS would become clear in Section 2.6.

2.4 Steady-State Solution

If the following limit exists

$$\lim_{k\to\infty} C[k] = C \tag{2-4.1}$$

then by (2-3.8)

$$\lim_{k\to\infty} K[k] = K = \mu G C^T (CC^T)^{-1}$$

and by (2-3.9)

$$\lim_{k\to\infty} K[k] = K = \mu G C^T (\gamma I + CC^T)^{-1}$$

and by (2-3.11)

$$\lim_{k\to\infty} K[k] = K = \mu G C^T$$

Any definition of K that we choose, the observer gain asymptotically settles down to a steady-state value. Moreover, if the following limit also exists

$$\lim_{k\to\infty} A[k] = A \tag{2-4.2}$$

then (2-2.5) approaches the following steady state solution

$$\hat{x}[k] = A\hat{x}[k-1] + K(y[k] - CA\hat{x}[k-1]) \tag{2-4.3}$$

As (2-4.3) is an LTI system, it is possible to write an expression of its transfer function

$$H(z) = z\left(zI - A + KCA\right)^{-1} K \tag{2-4.4}$$

The steady-state SSLMS is numerically efficient as compared to SSLMS. However, its use is limited to time-invariant systems.

2.5 Initialization

In order to start the recursion, SSLMS requires knowledge of $\hat{x}[0]$. If no other estimate is available $\hat{x}[0] = \mathbf{0}$ is a good choice. One way to obtain this estimate is to use the method of delayed recursion [34]. Before starting the recursion some samples of the observations are batch processed to obtain estimate of $\hat{x}[0]$, hence the name delayed recursion. These samples are given negative time indices to ensure our recursion always starts at $k=1$.

2.6 Analogy with the Standard LMS

The classical adaptive transversal filter [13] is illustrated in Figure 2-1. In order to show analogy of SSLMS with standard LMS, let

$$\mathbf{u}[k]=\left[u[k],u[k-1],\cdots u[k-n+1]\right]^T \tag{2-6.1}$$

be input vector for a system with tap weights

$$\mathbf{w_o}[k]=\left[w_{o1}[k],w_{o2}[k],\cdots w_{on}[k]\right]^T \tag{2-6.2}$$

The output of this filter is corrupted by additive observation noise $v[k]$. The signal $d[k]$ is called the desired signal. The tap-weights of an adaptive transversal filter

$$\hat{\mathbf{w}}[k]=\left[\hat{w}_1[k],\hat{w}_2[k],\cdots \hat{w}_n[k]\right]^T \tag{2-6.3}$$

can be thought of as an estimate of the unknown tap-weights $\mathbf{w_o}$. The output of this filter is $s[k]$. The problem is to adjust the estimated tap-weights so as to minimize the error $e[k]$ in some sense. We choose the minimum norm least square error as our optimization criterion. If the problem is overdetermined then the filter becomes the standard RLS. If it is underdetermined then we get the normalized LMS. Finally dropping the normalization factor gives us the standard LMS. The relevant details can be found in [13]. It is not difficult to see that if the following condition holds

$$
\begin{aligned}
m &= 1 \\
A &= I \\
C[k] &= \mathbf{u}[k] \\
\bar{x}[k] &= \hat{\mathbf{w}}[k] \\
\varepsilon[k] &= e[k] \\
y[k] &= d[k] \\
\bar{y}[k] &= s[k]
\end{aligned}
\tag{2-6.4}
$$

then the filter developed in this chapter turns into standard LMS. Due to this reason, we assert that SSLMS and its variants are in fact generalization of the conventional LMS filters.

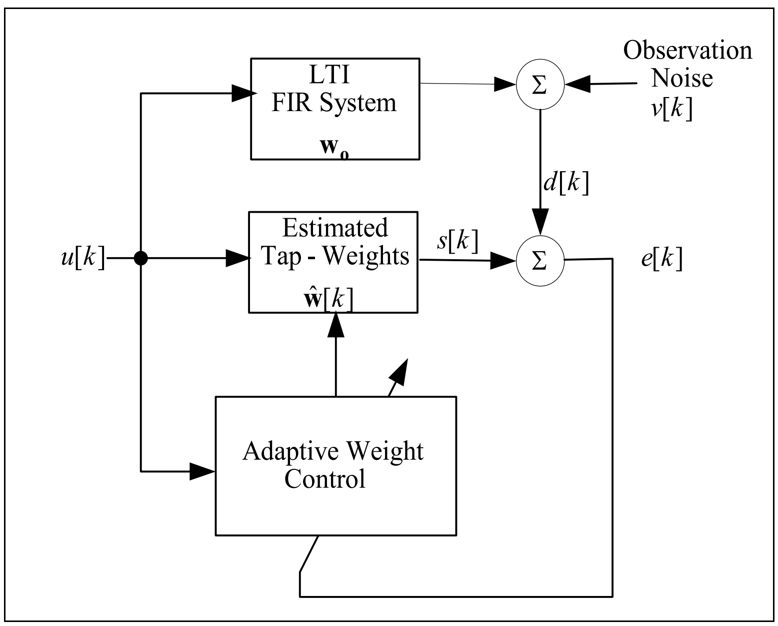

Figure 2-1. Adaptive Transversal Filter

2.7 Example (Sinusoid Tracking)

2.7.1 Sinusoid Tracking Problem

The problem of tracking a noisy sinusoid/chirp is of historical significance and has received considerable attention in the literature ([12], [41], [53]). The problem arises naturally in the context of an interfering signal of known frequency. Widrow et al. considered the problem of cancellation of 60 Hz interference in electrocardiography, using LMS [59]. In this section we give a brief account of how SSLMS can be used in such cases.

The phase and amplitude of the interfering signal are assumed to be unknown. Apparently, *a priori* knowledge of frequency simplifies the problem into a trivial one. However in the case of standard RLS and LMS, the designer has no direct way to incorporate this information. By virtue of state-space formulation of SSLMS, this information can be incorporated in a straightforward manner. For a sinusoid represented in discrete time as

$$y[k] = \sigma_s \cos(\omega_o kT + \phi) + v[kT] \tag{2-7.1}$$

the system matrices are given by [35]

$$\begin{aligned} A &= \begin{bmatrix} \cos(\omega_o T) & \sin(\omega_o T) \\ -\sin(\omega_o T) & \cos(\omega_o T) \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 \end{bmatrix} \end{aligned} \tag{2-7.2}$$

where

$$\begin{aligned} \sigma_s^2 &= \text{Signal power} \\ \omega_o &= \text{Signal frequency} \\ \phi &= \text{Phase of signal} \\ \mathrm{T} &= \text{Sampling time} \\ v &= \text{Observation noise} \end{aligned}$$

SSLMS may then be used track the interfering signal.

2.7.2 Computer Experiment

SSLMS is selected as the estimator. In order to demonstrate the net algorithm, the following parameters are assumed

$$\begin{aligned} a &= 1.5 \\ \omega_o &= 0.1 \\ \phi &= \pi / 4 \\ \mu &= 0.05 \\ x[0] &= \begin{bmatrix} 1 & 0 \end{bmatrix}^T \\ G &= I \end{aligned} \tag{3}$$

where I is the identity matrix. Observation noise is taken as a zero mean white sequence of variance 0.1. We observe the signal in (2-7.1) in discrete domain after sampling with sampling time $T = 1s$. The simulation results as illustrated in Figure 2-2 demonstrate the performance of the net algorithm. In order to draw a comparison with other adaptive algorithms, tracking results for SSRLS [35] with forgetting factor $\lambda = 0.99$, 3-tap standard LMS filter with step size parameter $\mu = 0.005$ and 3-tap standard RLS filter with forgetting factor $\lambda = 0.99$ are also reported.

2.7.3 Remarks

The tracking performance of SSLMS is much better than that of LMS and RLS, whereas the performance of SSLMS is comparable to SSRLS in steady-state. After the expiry of transient period, the SSLMS catches up quite well with the signal to be tracked.

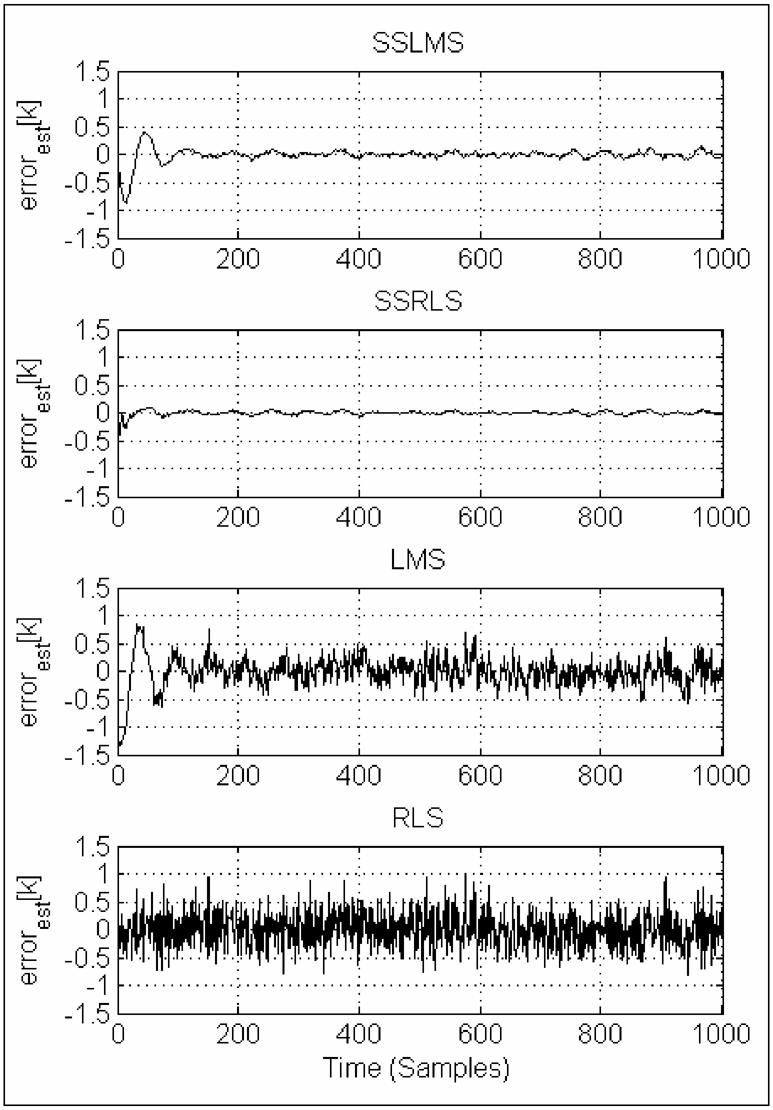

Figure 2-2. Sinusoid Estimation Error for SSLMS, SSRLS, LMS and RLS

CHAPTER 3

STATE-SPACE LEAST MEAN SQUARE WITH ADAPTIVE MEMORY

This chapter extends the discussion of CHAPTER 2 by developing adaptive memory variant of SSLMS i.e. SSLMS with adaptive memory (SSLMSWAM) ([40], [45]). The motivation for this development comes from the need to choose the most suitable value of step-size parameter μ for a particular application. In a time-varying setting the suitable choice of μ also varies with time. It turns out that it is difficult to choose *a priori* the most suitable value of μ.

The idea is to iteratively tune μ by stochastic gradient method so as to minimize the mean square value of the prediction error. SSLMSWAM then uses updated value of μ in its recursion. This results in overall performance improvement, though at the cost of added computational cost. To highlight the usefulness of SSLMSWAM, the applications of tracking noisy chirp signal and Van der Pol oscillations are discussed. The performance of SSLMSWAM is also compared with that of SSLMS, SSRLS, SSRLSWAM, RLS and LMS.

A detailed discussion of computational complexities of SSLMS family of algorithms concludes the chapter.

3.1 Memory Length

The memory of SSLMS is the amount of data from the past it uses to go through an iteration. The memory is linked with the choice μ. Higher the value of μ lower will be the memory size, and vice versa. The length of filter memory could be approximated by the following expression [13]

$$\text{Filter Memory} = \frac{1}{\mu} \tag{3-1.1}$$

3.2 Adaptation of Step-Size Parameter

When the model of the underlying environment is completely/partially unknown, a presumed model results in a model mismatch almost obviously. A partial compensation for this mismatch may be achieved if we adaptively tune the step-size parameter, so as to minimize a cost function. SSLMS with adaptive memory builds upon the framework of SSLMS to achieve iterative tuning of step-size parameter. Our objective is to tune the step size parameter μ so as to minimize the following cost function

$$J[k] = \frac{1}{2} E\left[\varepsilon^T[k] \varepsilon[k] \right] \tag{3-2.1}$$

where $E[\cdot]$ is the expectation operator and $\varepsilon[k]$ is the prediction error defined in (2-2.2). Differentiating $J[k]$ with respect to μ gives

$$\begin{aligned} \nabla_\mu[k] &= \frac{\partial J[k]}{\partial \mu} \\ &= E\left[\frac{\partial \varepsilon^T[k]}{\partial \mu} \varepsilon[k] \right] \end{aligned} \tag{3-2.2}$$

where $\frac{\partial \varepsilon^T[k]}{\partial \mu}$ is a row vector. Defining

$$\psi[k] = \frac{\partial \hat{x}[k]}{\partial \mu} \tag{3-2.3}$$

we get

$$\begin{aligned}\frac{\partial \varepsilon[k]}{\partial \mu} &= \frac{\partial}{\partial \mu}\left[y[k] - C[k]A[k-1]\hat{x}[k-1]\right] \\ &= -C[k]A[k-1]\psi[k-1]\end{aligned} \tag{3-2.4}$$

which implies that

$$\nabla_{\mu}[k] = -E\left[\psi^{T}[k-1]A^{T}[k-1]C^{T}[k]\varepsilon[k]\right] \tag{3-2.5}$$

Differentiating (2-2.5) with respect to μ and using (2-2.1), (2-3.11), (3-2.3) and (3-2.4) we get

$$\psi[k] = (A[k-1] - K[k]C[k]A[k-1])\psi[k-1] + GC^{T}[k]\varepsilon[k] \tag{3-2.6}$$

Now we are in a position to formulate SSLMS with adaptive memory. The stochastic gradient method that updates $\mu[k]$, which in turn is a function of time, is [13]

$$\mu[k] = \mu[k-1] - \alpha\nabla_{\mu}[k] \tag{3-2.7}$$

where α is a small positive learning rate parameter. Based on (3-2.5), an instantaneous estimate for the scalar gradient $\nabla_{\mu}[k]$ can be taken as

$$\hat{\nabla}_{\mu}[k] = -\psi^{T}[k-1]A^{T}[k-1]C^{T}[k]\varepsilon[k] \tag{3-2.8}$$

which modifies (3-2.7) into

$$\mu[k] = \left[\mu[k-1] + \alpha\psi^{T}[k-1]A^{T}[k-1]C^{T}[k]\varepsilon[k]\right]_{\mu_{-}}^{\mu_{+}} \tag{3-2.9}$$

For this algorithm to be meaningful we require $\mu > 0$. The bracket followed by μ_{-} and μ_{+} in equation indicates truncation that restricts step size parameter to $\left[\mu_{-}, \mu_{+}\right]$. The lower limit is generally set close to zero, whereas the upper limit depends on the nature of the problem. Its value is determined through experimentation. A simplification in the algorithm results if we ignore the normalization factor. The normalization factor improves convergence properties of the SSLMS by normalizing the step-size parameter μ. However, when we have a complete scheme to adapt step-size parameter μ, then normalization is somewhat unnecessary. Replacing μ by $\mu[k]$ in (2-3.8), the complete SSLMS algorithm with adaptive memory is summarized below in (3-2.10). First three equations constitute SSLMS, whereas the last two address the update of step-size parameter μ.

$$
\begin{aligned}
K[k] &= \mu[k]GC^T[k] \\
\varepsilon[k] &= y[k] - C[k]A[k-1]\hat{x}[k-1] \\
\hat{x}[k] &= A[k-1]\hat{x}[k-1] + K[k]\varepsilon[k] \\
\mu[k] &= \Big[\mu[k-1] + \alpha\psi^T[k-1]A^T[k-1]C^T[k]\varepsilon[k]\Big]_{\mu_-}^{\mu_+} \\
\psi[k] &= (A[k-1] - K[k]C[k]A[k-1])\psi[k-1] \\
&\quad + GC^T[k]\varepsilon[k]
\end{aligned}
\tag{3-2.10}
$$

3.3 Initialization

In order to start the recursion, SSLMSWAM requires knowledge of $\hat{x}[0]$, $\mu[0]$ and $\psi[0]$. Recursion can be started with any reasonable estimate of $\mu[0]$, whereas $\psi[0] = \mathbf{0}$ is a good choice. For the choice of $\hat{x}[0]$, refer to the discussion in Section 2.5.

3.4 Example (Tracking Van der Pol Oscillations)

3.4.1 Model Selection

We illustrate the performance of SSLMSWAM by tracking Van der Pol oscillations [22]. An electronic circuit with vacuum tubes, that acts like a resistor for high current through it, can be modeled by a Van der Pol oscillator. The circuit exhibits negative resistance behavior for low values of current through it. Small oscillations are amplified whereas larger ones are suppressed. Equations for Van der Pol oscillator are [22]

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + \varepsilon(1 - x_1^2)x_2 \end{aligned} \tag{3-4.1}$$

Working with the assumption that actual (Van der Pol) signal model is completely unknown, the constant acceleration model [34] given below is a good choice.

$$\begin{aligned} A &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{3-4.2}$$

3.4.2 Computer Experiment

SSLMSWAM is selected as the estimator. We observe the signal x_2 in (3-4.1) in discrete domain after sampling with sampling time $T = 0.01s$. Zero mean white Gaussian noise with variance 0.1 corrupts the observations. The learning rate parameter is chosen to be $\alpha = 0.01$. The system is started with zero initial conditions except $\mu[0] = 0.1$. Matrix G is chosen to be

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 0 & 0 \\ 0.3 & 0 & 0 \end{bmatrix} \tag{3-4.3}$$

The noisy Van der Pol oscillations to be tracked are shown in Figure 3-1. The simulation results as illustrated in Figure 3-2 demonstrate the performance of the net algorithm. In order to draw a comparison with other adaptive algorithms, tracking results for SSLMS with step size parameter $\mu = 0.1$, SSRLS [35] with forgetting factor $\lambda = 0.97$, SSRLSWAM [39] with initial forgetting factor $\lambda[0] = 0.93$, 3-tap standard LMS filter with $\mu = 0.001$ and 3-tap standard RLS filter with forgetting factor $\lambda = 0.99$ are also given. Adaptation of step size parameter for SSLMSWAM is given in Figure 3-3. Results for the case of uniformly distributed observation noise are given in Figure 3-4, where LMS operates with $\mu = 0.1$ and RLS operates with $\lambda = 0.8$.

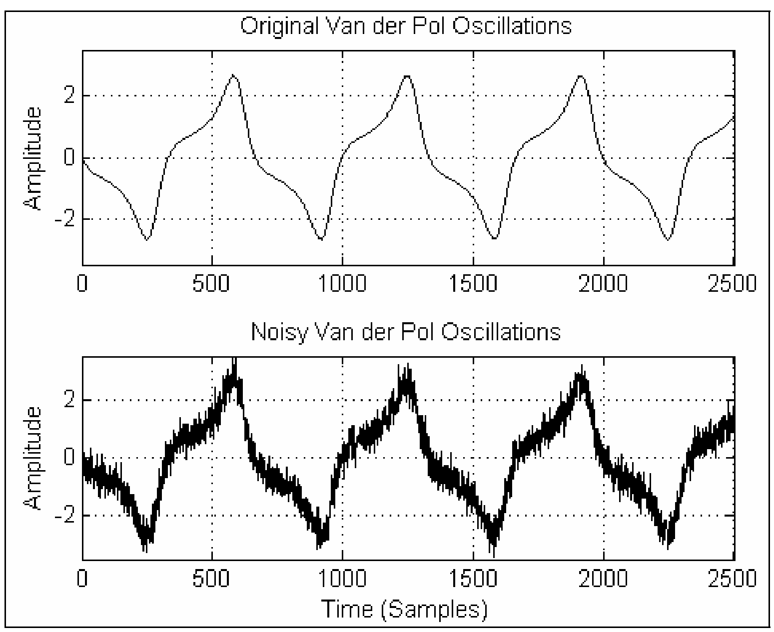

Figure 3-1. Van der Pol Oscillations

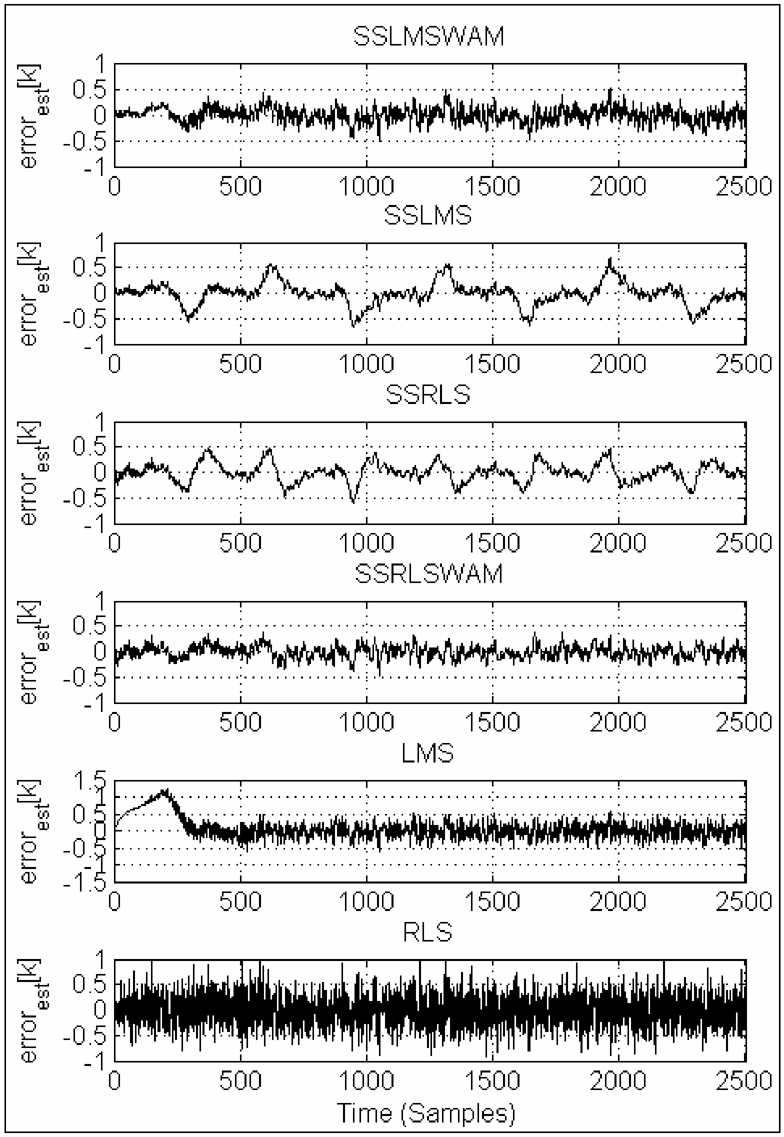

Figure 3-2. Van der Pol Oscillations Estimation Error for SSLMSWAM, SSLMS, SSRLS, SSRLSWAM, LMS AND RLS (Gaussian Observation Noise)

3.4.3 Remarks

The constant acceleration model (3-4.2) used in these simulations provides an approximation by fitting second order polynomial on various segments of Van der Pol oscillations. For the segments containing lower frequency components, the fit is good and

SSLMS & SSRLS perform well. On the other hand for the segments having higher frequency components the fit is not good enough. SSLMSWAM and SSRLSWAM compensate partially for this *misfit* by tuning step-size parameter and forgetting factor, respectively. Consequently, the estimation error for SSLMSWAM & SSRLSWAM is less than SSLMS & SSRLS for these segments. This effect is easier to notice in case of uniformly distributed observation noise (Figure 3-4).

In Figure 3-4, periodic sharp rise and fall in estimation error for LMS is noticeable whereas performance of RLS for the selected value of forgetting factor is satisfactory. It is possible to improve the performance of SSLMS, LMS, SSRLS and RLS by changing values of step-size parameter and forgetting factor, respectively. However, we assume that the designer does not know beforehand the most suitable values of these parameters.

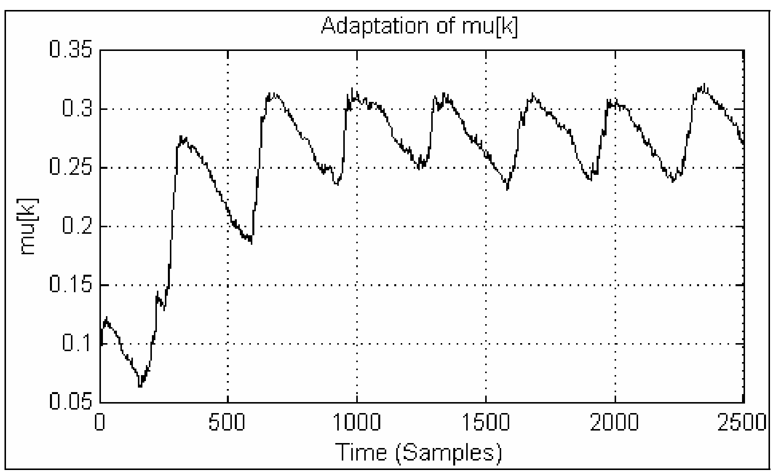

Figure 3-3. Adaptation of Step Size Parameter for SSLMSWAM – Van der Pol Oscillations Tracking

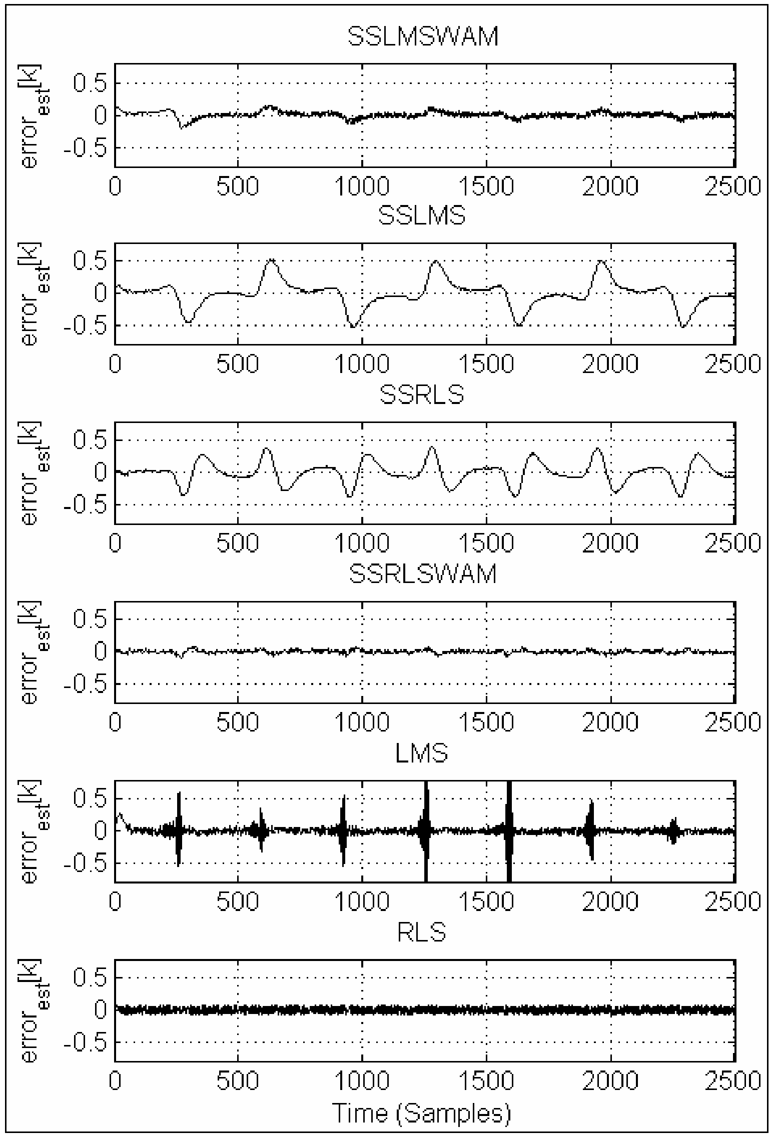

Figure 3-4. Van der Pol Oscillations Estimation Error for SSLMSWAM, SSLMS, SSRLS, SSRLSWAM, LMS AND RLS (Uniformly Distributed Observation Noise)

3.5 Computational Complexity

Computational complexity of an algorithm is usually of significant importance particularly in real-time applications. In this section, we discuss this aspect of SSLMS. The complexities of equations (2-2.1)−(2-2.3), (2-2.5) are given in Table 3-1. These equations are common to all the variants of SSLMS. Table 3-2 furnishes complexity of

SSLMS that comprises equations (2-2.1)−(2-2.3), (2-2.5) and (2-3.11). It can be seen that complexity of SSLMS is $O(n^2)$.

Table 3-1. Complexity of State Estimator (Equations (2-2.1)−(2-2.3), (2-2.5))

Eq #	Operation	Multiplications	Additions	Divisions
(2-2.1)	$\bar{x}[k]_{n\times 1} = A_{n\times n}\hat{x}[k-1]_{n\times 1}$	n^2	$n^2 - n$	–
(2-2.2)	$\varepsilon[k]_{m\times 1} = y[k]_{m\times 1} - \bar{y}[k]_{m\times 1}$	–	m	–
(2-2.3)	$\bar{y}[k]_{m\times 1} = C_{m\times n}\bar{x}[k]_{n\times 1}$	mn	$mn - m$	–
(2-2.5)	$\hat{x}[k]_{n\times 1} = \bar{x}[k]_{n\times 1} + K_{n\times m}\varepsilon[k]_{m\times 1}$	mn	mn	–
	Total	$n^2 + 2mn$	$n^2 + 2mn$ $-n$	–

Table 3-2. Complexity of SSLMS State Observer (Equations (2-2.1)−(2-2.3), (2-2.5), (2-3.11))

Eq #	Operation	Multiplications	Additions	Divisions
	Total from Table 3-1	$n^2 + 2mn$	$n^2 + 2mn$ $-n$	–
(2-3.11) [a]	$K[k]_{n\times m} = \mu_{1\times 1}G_{n\times n}C^T[k]_{n\times m}$	mn^2	$mn^2 - mn$	–
	Total	$n^2 + mn^2 + 2mn$	$n^2 + mn^2 + mn$ $-n$	–

[a] $\mu_{1\times 1}G_{n\times n}$ computed offline

Table 3-3. Complexity of SSNLMS State Observer (Equations (2-2.1)−(2-2.3), (2-2.5), (2-3.9))

Eq #	Operation	Multiplications	Additions	Divisions
	Total from Table 3-1	n^2+2mn	n^2+2mn $-n$	–
[a]	$P_{n\times m}=\mu_{1\times 1}G_{n\times n}C^T[k]_{n\times m}$	mn^2	mn^2-mn	–
	$Q_{m\times m}=C[k]_{m\times n}C^T[k]_{n\times m}$	m^2n	m^2n-m^2	–
[b]	$R_{m\times m}=\gamma_{1\times 1}I_{m\times m}+Q_{m\times m}$	–	m^2	–
	Gauss-Jordan Inversion [26] $S_{m\times m}=R^{-1}_{m\times m}$	$\frac{m^3}{3}+\frac{m^2}{2}+\frac{m}{6}$	$\frac{m^3}{3}+\frac{m^2}{2}+\frac{m}{6}$	$\frac{m^2}{2}+\frac{m}{2}$
(2-3.9)	$K[k]_{n\times m}=P_{n\times m}S_{m\times m}$	m^2n	m^2n-mn	–
	Total	$n^2+\frac{m^3}{3}+2m^2n$ $+mn^2+\frac{m^2}{2}+2mn$ $+\frac{m}{6}$	$n^2+\frac{m^3}{3}+2m^2n+mn^2$ $+\frac{m^2}{2}-n+\frac{m}{6}$	$\frac{m^2}{2}+\frac{m}{2}$

[a] $\mu_{1\times 1}G_{n\times n}$ computed offline

[b] $\gamma_{1\times 1}I_{m\times m}$ computed offline

Table 3-3 provides complexity of SSNLMS that comprises equations (2-2.1) −(2-2.3), (2-2.5) and (2-3.9). Although the complexity of SSNLMS is also $O(n^2)$, it is computationally more intensive of the two due to its requirement of mth order matrix inversion. SSLMSWAM (equation (3-2.10)) is $O(n^2)$ but computationally more intensive (Table 3-4) than SSNLMS and SSLMS, due to adaptation of step-size parameter.

Table 3-4. Complexity of SSLMSWAM State Observer (Equation (3-2.10))

Eq #	Operation	Multiplications	Additions	Divisions
	$H_{n\times m} = G_{n\times n} C^T[k]_{n\times m}$	mn^2	$mn^2 - mn$	–
	$K[k]_{n\times m} = \mu[k]_{1\times 1} H_{n\times m}$	mn	–	–
	Total from Table 3-1	$n^2 + 2mn$	$n^2 + 2mn$ $-n$	–
	$P_{n\times 1} = C^T[k]_{n\times m} \varepsilon[k]_{m\times 1}$	mn	$mn - n$	–
	$Q_{n\times 1} = A^T[k-1]_{n\times n} P_{n\times 1}$	n^2	$n^2 - n$	–
	$R_{1\times 1} = \alpha_{1\times 1} \psi^T[k-1]_{1\times n} Q_{n\times 1}$	$n+1$	$n-1$	–
	$\mu[k]_{1\times 1} = \mu[k-1]_{1\times 1} + R_{1\times 1}$	–	1	–
	$J_{n\times 1} = H_{n\times m} \varepsilon[k]_{m\times 1}$	mn	$mn - n$	–
	$M_{m\times n} = C[k]_{m\times n} A[k-1]_{n\times n}$	mn^2	$mn^2 - mn$	–
	$N_{n\times n} = K[k]_{n\times m} M_{m\times n}$	mn^2	$mn^2 - n^2$	–
	$O_{n\times n} = A[k-1]_{n\times n} - N_{n\times n}$	–	n^2	–
	$S_{n\times 1} = O_{n\times n} \psi[k-1]_{n\times 1}$	n^2	$n^2 - n$	–
(3-2.10)	$\psi[k]_{n\times 1} = S_{n\times 1} + J_{n\times 1}$	–	n	–
	Total	$3n^2 + 3mn^2 + 5mn$ $+n+1$	$3n^2 + 3mn^2 + 2mn - 3n$	–

We summarize the complexity of algorithms presented in this chapter and the previous one, and compare them with SSRLS, SSLMSWAM, RLS, LMS and NLMS in Table 3-5, for $m=1$ ([13], [35] and [39]).

Table 3-5. Summary of Complexities of Algorithms for $m=1$

Algorithm	Multiplications	Additions	Divisions
SSLMS (Table 3-2)	$2n^2+2n$	$2n^2$	–
SSNLMS (Table 3-3)	$2n^2+4n+1$	$2n^2+n+1$	1
SSLMSWAM (Table 3-4)	$6n^2+6n+1$	$6n^2-n$	–
SSRLS [35]	$2n^3+3n^2+4n$	$2n^3+n^2+n$	1
SSRLSWAM [39]	$8n^3+\frac{19n^2}{2}+\frac{13n}{2}+1$	$8n^3-n^2+4n$	3
RLS [13]	$4n^2+O(n)$		1
LMS [13]	$3n+2$		–
NLMS [13]	$5n+2$		1

CHAPTER 4

FINITE IMPULSE RESPONSE ADAPTIVE FILTER

In CHAPTER 2 and CHAPTER 3, we developed SSLMS and its variants. The ideas presented were demonstrated with the help of a couple of examples considering tracking of noise corrupted signals. Due to recursive nature of these algorithms, the choice of step-size parameter is crucial in determining their stability. However, if a filter is based on observations on a finite interval, the stability is guaranteed [48]. This leads to the main idea of this chapter.

In this chapter, we develop finite impulse response (FIR) adaptive filter which is built around state-space framework of an unforced system. The filter is inherently stable by virtue of its FIR 'observations to state estimate' mapping, which is also confirmed by its transfer function representation. State-space formulation of the filter allows seamless integration of model of the underlying environment. Moreover, multiple-input multiple-output (MIMO) systems can be handled conveniently. The filter does not require *a priori* knowledge of properties of process and/or observation noise and hence it works in presence/absence of both or any one of the noises. The performance of the filter is however dependent on various factors like time-varying nature of the observed signal, model uncertainty, and/or nonstationary behavior of the observation noise.

We start with batch processed least squares solution to the state estimate. The discussion is then extended to a recursive solution. The derivations for time-varying solutions are naturally extended to the time-invariant case. Alternate forms of the filter i.e. current estimator and predictor form, are also derived. These forms are useful for specific applications. Under a given set of conditions, the FIR adaptive filter and state-space RLS (SSRLS; exponential forgetting of the past observations) [35] are shown to be equivalent. We also carry out first and second order convergence analysis of the filter for time-varying case. Similar to the development of SSLMSWAM in CHAPTER 3, adaptive memory variant of FIR adaptive filter is formulated next. The idea is to recursively tune the observation window size so as to minimize prediction error.

We demonstrate the performance of FIR adaptive filter using an example considering tracking problem of Van der Pol oscillations. A performance comparison with LMS and RLS [13] is also given.

4.1 Batch Processed Solution

Consider the discrete-time unforced system (1-2.1), with deterministic matrices $A[k]$ and $C[k]$. Let the observations $\boldsymbol{y}[k]$ start appearing at time $k=1$. Data pre-windowing is assumed i.e. $\boldsymbol{y}[k]=0$ for $k \le 0$. We begin our discussion by batch processing the observations. From equation (1-2.1), we may write

$$\begin{aligned}\boldsymbol{y}[k-1] &= C[k-1]\boldsymbol{x}[k-1]+\boldsymbol{v}[k-1]\\ &= C[k-1]A[k-1,k]\boldsymbol{x}[k]+\boldsymbol{v}[k-1]\end{aligned}$$

where we have used equations (2-1.1) and (2-1.2). Similarly, we can write p different equations from (1-2.1) as follows

$$\boldsymbol{\mathcal{Y}}[k]=\begin{bmatrix}\boldsymbol{y}[k-p+1]\\ \boldsymbol{y}[k-p+2]\\ \vdots\\ \boldsymbol{y}[k-2]\\ \boldsymbol{y}[k-1]\\ \boldsymbol{y}[k]\end{bmatrix}=\begin{bmatrix}C[k-p+1]A[k-p+1,k]\\ C[k-p+2]A[k-p+2,k]\\ \vdots\\ C[k-2]A[k-2,k]\\ C[k-1]A[k-1,k]\\ C[k]\end{bmatrix}\boldsymbol{x}[k]+\boldsymbol{\mathcal{V}}[k] \qquad (4\text{-}1.1)$$

where $p \geq l$ such that both p and l are positive integers, $k \geq p$ and observation noise vector $\boldsymbol{\mathcal{V}}[k]$ is given by

$$\boldsymbol{\mathcal{V}}[k] = \left[\boldsymbol{v}^T[k-p+1] \quad \boldsymbol{v}^T[k-p+2] \quad \cdots \quad \boldsymbol{v}^T[k-1] \quad \boldsymbol{v}^T[k] \right]^T, \tag{4-1.2}$$

Define

$$H[k] = \begin{bmatrix} C[k-p+1]A[k-p+1,k] \\ C[k-p+2]A[k-p+2,k] \\ \vdots \\ C[k-2]A[k-2,k] \\ C[k-1]A[k-1,k] \\ C[k] \end{bmatrix} \tag{4-1.3}$$

where $H[k] \in \mathbb{R}^{mp \times n}$. Using (4-1.2) and (4-1.3), we may write (4-1.1) as

$$\boldsymbol{\mathcal{Y}}[k] = H[k]\boldsymbol{x}[k] + \boldsymbol{\mathcal{V}}[k] \tag{4-1.4}$$

The solution of system (4-1.4) in terms of least squares is given as follows [13]

$$\hat{\boldsymbol{x}}[k] = (H^T[k]H[k])^{-1}H^T[k]\boldsymbol{\mathcal{Y}}[k] \tag{4-1.5}$$

The condition $p \geq l$ along with the observability assumption ensures the invertibility of $H^T[k]H[k]$. The dimension of $(H^T[k]H[k])^{-1}H^T[k]$ is $n \times mp$.

We introduce a weighting factor [13] such that the observations on the interval $\left[k-p+1,k\right]$ are given lesser weight as they become old; the most recent being with the largest weight. Defining $0 < \beta \leq 1$ as the weighting factor, the corresponding weighting matrix is given as

$$W = \begin{bmatrix} \beta^{p-1} I_m & 0 & \cdots & 0 & 0 \\ 0 & \beta^{p-2} I_m & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & \beta I_m & 0 \\ 0 & 0 & \cdots & 0 & I_m \end{bmatrix} \tag{4-1.6}$$

where I_m is identity matrix of dimension $m \times m$ and the dimension of weighting matrix W is $mp \times mp$. Taking equation (4-1.6) into account, the solution of system (4-1.4) in terms of weighted least squares is given as follows [56]

$$\hat{\boldsymbol{x}}[k] = (H^T[k] W H[k])^{-1} H^T[k] W \boldsymbol{\mathcal{Y}}[k] \tag{4-1.7}$$

For the special case $\beta = 1$, we get $W = I$ i.e. a rectangular window. Consequently, the solution (4-1.7) and (4-1.5) become equivalent. In the next section, we derive rectangular RLS (RRLS).

4.2 Recursive Solution

Let us define the following symbols

$$\begin{aligned} \Phi[k] &= H^T[k] W H[k] \\ \boldsymbol{\zeta}[k] &= H^T[k] W \boldsymbol{\mathcal{Y}}[k] \end{aligned} \tag{4-2.1}$$

From (4-1.7) and (4-2.1), we get

$$\hat{\boldsymbol{x}}[k] = \Phi^{-1}[k] \boldsymbol{\zeta}[k] \tag{4-2.2}$$

We want the second equation in (4-2.2) to be computed recursively. To realize this goal, we find recursive solutions of $\Phi^{-1}[k]$ and $\boldsymbol{\zeta}[k]$.

4.2.1 Computation of $\Phi[k]$

From equation (4-1.4) and (4-2.1), we may write

$$\Phi[k]=\Big[\beta^{p-1}A^T[k-p+1,k]C^T[k-p+1]C[k-p+1]A[k-p+1,k]+ \\ \beta^{p-2}A^T[k-p+2,k]C^T[k-p+2]C[k-p+2]A[k-p+2,k]+ \\ \cdots\cdots+\beta A^T[k-1,k]C^T[k-1]C[k-1]A[k-1,k]+C^T[k]C[k]\Big] \tag{4-2.3}$$

Replacing time index k by $k-1$ in (4-2.3), we get

$$\Phi[k-1]=\Big[\beta^{p-1}A^T[k-p,k-1]C^T[k-p]C[k-p]A[k-p,k-1]+ \\ \beta^{p-2}A^T[k-p+1,k-1]C^T[k-p+1]C[k-p+1]A[k-p+1,k-1]+ \\ \cdots\cdots+\beta A^T[k-2,k-1]C^T[k-2]C[k-2]A[k-2,k-1]+C^T[k-1]C[k-1]\Big] \tag{4-2.4}$$

Pre-multiplying both sides of equation (4-2.4) by $\beta A^T[k-1,k]$ and post-multiplying by $A[k-1,k]$, we get

$$\beta A^T[k-1,k]\Phi[k-1]A[k-1,k]=\beta^p A^T[k-p,k]C^T[k-p]C[k-p]A[k-p,k]+ \\ \beta^{p-1}A^T[k-p+1,k]C^T[k-p+1]C[k-p+1]\times \\ A[k-p+1,k]+\beta^{p-2}A^T[k-p+2,k]C^T[k-p+2]\times \\ C[k-p+2]A[k-p+2,k]+\cdots\cdots+ \\ \beta A^T[k-1,k]C^T[k-1]C[k-1]A[k-1,k] \tag{4-2.5}$$

Adding $C^T[k]C[k]$ and subtracting $\beta^p A^T[k-p,k]C^T[k-p]C[k-p]A[k-p,k]$ on both sides of equation (4-2.5), we find that right hand side of resulting equation is same as right hand side of (4-2.3). Hence we may write

$$\Phi[k]=\beta A^T[k-1,k]\Phi[k-1]A[k-1,k]+C^T[k]C[k]- \\ \beta^p A^T[k-p,k]C^T[k-p]C[k-p]A[k-p,k] \tag{4-2.6}$$

The $n\times n$ matrix $\Phi[k]$ computed recursively using (4-2.6) must be inverted so that we obtain state estimate using second equation of (4-2.1). To alleviate the computational complexity associated with matrix inversion of (4-2.6), we suggest recursive computation of $\Phi^{-1}[k]$. Before we give recursive computation of $\Phi^{-1}[k]$, we suggest recursive computation of $A[k-p,k]$ (a product of p different matrices) that appears on right hand side of equation (4-2.6).

4.2.2 Computation of $A[k-p,k]$

Define

$$\begin{aligned}\theta_p[k] &= A[k-p,k] \\ &= A^{-1}[k,k-p] \\ &= A^{-1}[k-p]\cdots\cdots A^{-1}[k-2]A^{-1}[k-1]\end{aligned} \tag{4-2.7}$$

Replacing time index k by $k-1$ in (4-2.7), we get

$$\begin{aligned}\theta_p[k-1] &= A[k-p-1,k-1] \\ &= A^{-1}[k-p-1]\cdots\cdots A^{-1}[k-3]A^{-1}[k-2]\end{aligned} \tag{4-2.8}$$

Pre-multiplying both sides of equation (4-2.8) by $A[k-p-1]$ and post-multiplying by $A^{-1}[k-1]$, we find that the right hand side of the resulting equation is same as the right hand side of (4-2.7). Hence we may write

$$\theta_p[k] = A[k-p-1]\theta_p[k-1]A^{-1}[k-1] \tag{4-2.9}$$

4.2.3 Computation of $\Phi^{-1}[k]$

Define

$$\Psi[k] = C[k-p]\theta_p[k] \tag{4-2.10}$$

From (4-2.6) and (4-2.9), we get

$$\Phi[k] = \beta A^T[k-1,k]\Phi[k-1]A[k-1,k] + C^T[k]C[k] - \beta^p \Psi^T[k]\Psi[k] \tag{4-2.11}$$

which may be written as

$$A^T[k,k-1]\Phi[k]A[k,k-1] = \beta\Phi[k-1] + \left(C[k]A[k,k-1]\right)^T C[k]A[k,k-1] - \beta^p \left(\Psi[k]A[k,k-1]\right)^T \Psi[k]A[k,k-1] \tag{4-2.12}$$

Define

$$\Lambda[k] = \beta\Phi[k-1] - \beta^p \left(\Psi[k]A[k,k-1]\right)^T \Psi[k]A[k,k-1] \tag{4-2.13}$$

From (4-2.12) and (4-2.13), we get

$$A^T[k,k-1]\Phi[k]A[k,k-1] = \Lambda[k] + \left(C[k]A[k,k-1]\right)^T C[k]A[k,k-1] \tag{4-2.14}$$

Making use of Matrix Inversion Lemma [13], equation (4-2.14) gives the following Riccati equation after some algebraic manipulations

$$\begin{aligned}\Phi^{-1}[k] = {} & A[k,k-1]\Lambda^{-1}[k]A^T[k,k-1] - A[k,k-1]\Lambda^{-1}[k]A^T[k,k-1]C^T[k] \times \\ & \left(I + C[k]A[k,k-1]\Lambda^{-1}[k]A^T[k,k-1]C^T[k]\right)^{-1} \times \\ & C[k]A[k,k-1]\Lambda^{-1}[k]A^T[k,k-1]\end{aligned} \tag{4-2.15}$$

where I is an identity matrix of appropriate dimension. $\Lambda^{-1}[k]$ in (4-2.15) is given by

$$\begin{aligned}\Lambda^{-1}[k] = {} & \beta^{-1}\Phi^{-1}[k-1] + \beta^{p-2}\Phi^{-1}[k-1]A^T[k,k-1]\Psi^T[k] \times \\ & \left(I - \beta^{p-1}\Psi[k]A[k,k-1]\Phi^{-1}[k-1]A^T[k,k-1]\Psi^T[k]\right)^{-1} \times \\ & \Psi[k]A[k,k-1]\Phi^{-1}[k-1]\end{aligned} \tag{4-2.16}$$

which is obtained from (4-2.13) using Matrix Inversion Lemma [13].

4.2.4 Computation of $\zeta[k]$

From equation (4-1.4) and (4-2.1)

$$\begin{aligned}\zeta[k] = \Big[& \beta^{p-1}A^T[k-p+1,k]C^T[k-p+1]\boldsymbol{y}[k-p+1] + \cdots\cdots + \\ & \beta A^T[k-1,k]C^T[k-1]\boldsymbol{y}[k-1] + C^T[k]\boldsymbol{y}[k]\Big]\end{aligned} \tag{4-2.17}$$

Following a similar approach as in Section 4.2.1, equation (4-2.17) leads to

$$\boldsymbol{\zeta}[k] = \beta A^T[k-1,k]\boldsymbol{\zeta}[k-1] + C^T[k]\boldsymbol{y}[k] - \beta^p A^T[k-p,k]C^T[k-p]\boldsymbol{y}[k-p] \quad (4\text{-}2.18)$$

4.2.5 State Update

The relation (4-2.2) can be construed as a change of variables [52], which transforms (4-2.18) into

$$\begin{aligned}\hat{\boldsymbol{x}}[k] = {} & \beta\Phi^{-1}[k]A^T[k-1,k]\Phi[k-1]\hat{\boldsymbol{x}}[k-1] + \Phi^{-1}[k]C^T[k]\boldsymbol{y}[k] - \\ & \beta^p\Phi^{-1}[k]\Psi^T[k]\boldsymbol{y}[k-p]\end{aligned} \quad (4\text{-}2.19)$$

Equation (4-2.19), which is equivalent to (4-2.2), completely specifies the map from observations $\boldsymbol{y}[k]$ to the state estimate $\hat{\boldsymbol{x}}[k]$. The recursion (4-2.19) is based on finite number of observations. As the recursion makes use of new observation $\boldsymbol{y}[k]$, the effect of the oldest observation $\boldsymbol{y}[k-p]$ is removed, thus making the filter FIR.

4.3 Alternate Forms of Estimator

There are two fundamental types of state estimates [10]. The first is *current* estimate $\hat{\boldsymbol{x}}[k]$, which is based on measurements of observations on the interval $[k-p+1,k]$. Equation (4-2.19) gives the current estimate, an alternate form of which is discussed in sub section 4.3.1. The second estimate type is *predictor* estimate $\overline{\boldsymbol{x}}[k]$, which is based on measurements on the interval $[k-p,k-1]$.

4.3.1 Current Estimator

This sub section discusses an alternate form of the current estimator in terms of predicted state plus the correction terms [10]. This alternate form is given by

$$\hat{\boldsymbol{x}}[k] = \overline{\boldsymbol{x}}[k] + K[k]\left(\boldsymbol{y}[k] - \overline{\boldsymbol{y}}[k]\right) - G[k]\left(\boldsymbol{y}[k-p] - \varsigma[k]\overline{\boldsymbol{y}}[k-p]\right) \quad (4\text{-}3.1)$$

where $\bar{\boldsymbol{x}}[k] = A[k-1]\hat{\boldsymbol{x}}[k-1]$ is the predicted state, $\bar{y}[k] = C[k]\bar{\boldsymbol{x}}[k]$ and $\bar{y}[k-p] = \Xi[k]\bar{\boldsymbol{x}}[k]$ are the predicted outputs and $\Xi[k] = \Psi[k]A^{-1}[k]$. The last two terms on the right hand side of (4-3.1) are the correction terms. Equation (4-3.1) recursively computes state estimate $\hat{\boldsymbol{x}}[k]$ at *kth* time instant based on observations on the interval $\left[k-p+1, k\right]$. The last term on the right hand side of (4-3.1) makes the recursion FIR. The latest observation occurs at *kth* time instant, hence the name *current* estimator. Equation (4-3.1) is equivalent to (4-2.2) as will be established in sub section 4.3.2. The definitions of $K[k], G[k]$ and $\varsigma[k]$ used in (4-3.1) are given by

$$\begin{aligned} \varsigma[k] &= \Psi[k]\Xi^{+}[k] \\ G[k] &= \beta^{p}\Phi^{-1}[k]\Psi^{T}[k] \\ K[k] &= \Phi^{-1}[k]C^{T}[k] \end{aligned} \tag{4-3.2}$$

where $\Xi^{+}[k] = \Xi^{T}[k]\left(\Xi[k]\Xi^{T}[k]\right)^{-1}$ [13]. The definitions of $\Psi[k]$ and $\Phi[k]$ are same as in (4-2.10) and (4-2.1), respectively. We next sub section establish the equivalence of equations (4-2.2) and (4-3.1).

4.3.2 Equivalence of Current Estimator (4-2.2) and Batch-Processed Estimator (4-3.1)

Using (4-3.2), equation (4-3.1) can be expanded as below

$$
\begin{aligned}
\hat{\boldsymbol{x}}[k] &= A[k-1]\hat{\boldsymbol{x}}[k-1] + \Phi^{-1}[k]C^T[k]\big(\boldsymbol{y}[k] - C[k]A[k-1]\hat{\boldsymbol{x}}[k-1]\big) \\
&\quad -\beta^p\Phi^{-1}[k]\Psi^T[k]\big(\boldsymbol{y}[k-p] - \Psi[k]\Xi^+[k]\Xi[k]\overline{\boldsymbol{x}}[k]\big) \\
&= A[k-1]\hat{\boldsymbol{x}}[k-1] + \Phi^{-1}[k]C^T[k]\boldsymbol{y}[k] - \Phi^{-1}[k]C^T[k]C[k]\times \\
&\quad A[k-1]\hat{\boldsymbol{x}}[k-1] - \beta^p\Phi^{-1}[k]\Psi^T[k]\boldsymbol{y}[k-p] + \beta^p\Phi^{-1}[k]\Psi^T[k]\times \\
&\quad \Psi[k]A[k-1]\hat{\boldsymbol{x}}[k-1] \\
&= \Phi^{-1}[k]\Big[\Phi[k]A[k-1]\hat{\boldsymbol{x}}[k-1] + C^T[k]\boldsymbol{y}[k] - C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}[k-1] \\
&\quad -\beta^p\Psi^T[k]\boldsymbol{y}[k-p] + \beta^p\Psi^T[k]\Psi[k]A[k-1]\hat{\boldsymbol{x}}[k-1]\Big]
\end{aligned}
\tag{4-3.3}
$$

Using (4-2.6), equation (4-3.3) may be written as

$$
\begin{aligned}
\hat{\boldsymbol{x}}[k] &= \Phi^{-1}[k]\Big[\big(\beta A^{-T}[k-1]\Phi[k-1]A^{-1}[k-1] + C^T[k]C[k] - \beta^p\Psi^T[k]\Psi[k]\big)\times \\
&\quad A[k-1]\hat{\boldsymbol{x}}[k-1] + C^T[k]\boldsymbol{y}[k] - C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}[k-1] - \\
&\quad \beta^p\Psi^T[k]\boldsymbol{y}[k-p] + \beta^p\Psi^T[k]\Psi[k]A[k-1]\hat{\boldsymbol{x}}[k-1]\Big] \\
&= \Phi^{-1}[k]\Big[\beta A^{-T}[k-1]\Phi[k-1]\hat{\boldsymbol{x}}[k-1] + C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}[k-1] - \\
&\quad \beta^p\Psi^T[k]\Psi[k]A[k-1]\hat{\boldsymbol{x}}[k-1] + C^T[k]\boldsymbol{y}[k] - C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}[k-1] \\
&\quad -\beta^p\Psi^T[k]\boldsymbol{y}[k-p] + \beta^p\Psi^T[k]\Psi[k]A[k-1]\hat{\boldsymbol{x}}[k-1]\Big] \\
&= \Phi^{-1}[k]\Big[\beta A^{-T}[k-1]\Phi[k-1]\hat{\boldsymbol{x}}[k-1] + C^T[k]\boldsymbol{y}[k] - \beta^p\Psi^T[k]\boldsymbol{y}[k-p]\Big]
\end{aligned}
\tag{4-3.4}
$$

Using (4-2.2) and (4-2.18), equation (4-3.4) may be written as

$$
\begin{aligned}
\hat{\boldsymbol{x}}[k] &= \Phi^{-1}[k]\Big[\beta A^{-T}[k-1]\zeta[k-1] + C^T[k]\boldsymbol{y}[k] - \beta^p\Psi^T[k]\boldsymbol{y}[k-p]\Big] \\
&= \Phi^{-1}[k]\zeta[k]
\end{aligned}
\tag{4-3.5}
$$

which is same as (4-2.2).

4.3.3 Predictor Form

Substituting equation (4-3.1) into $\overline{\boldsymbol{x}}[k] = A[k-1]\hat{\boldsymbol{x}}[k-1]$, we get

$$
\overline{\boldsymbol{x}}[k+1] = A[k]\overline{\boldsymbol{x}}[k] + A[k]K[k]\big(\boldsymbol{y}[k] - \overline{\boldsymbol{y}}[k]\big) - A[k]G[k]\big(\boldsymbol{y}[k-p] - \varsigma[k]\overline{\boldsymbol{y}}[k-p]\big) \tag{4-3.6}
$$

Define

$$\xi[k] = A[k]K[k]$$
$$\varphi[k] = A[k]G[k] \tag{4-3.7}$$

Substituting (4-3.7) into (4-3.6), we get

$$\bar{\boldsymbol{x}}[k+1] = A[k]\bar{\boldsymbol{x}}[k] + \xi[k]\left(\boldsymbol{y}[k] - \bar{\boldsymbol{y}}[k]\right) - \varphi[k]\left(\boldsymbol{y}[k-p] - \varsigma[k]\bar{\boldsymbol{y}}[k-p]\right) \tag{4-3.8}$$

Equation (4-3.8) gives the predictor form of FIR adaptive filter. The predicted estimate $\bar{\boldsymbol{x}}[k]$ is based on the system output on the interval $\left[k-p, k-1\right]$. Predictor form of the estimator finds utility in applications that need future state estimate of the system. For time-invariant case, equation (4-3.8) simplifies to

$$\bar{\boldsymbol{x}}[k+1] = A\bar{\boldsymbol{x}}[k] + \xi\left(\boldsymbol{y}[k] - \bar{\boldsymbol{y}}[k]\right) - \varphi\left(\boldsymbol{y}[k-p] - \varsigma\bar{\boldsymbol{y}}[k-p]\right) \tag{4-3.9}$$

Note that ξ, φ and ς in (4-3.9) can be computed offline.

4.4 Transfer Function Representation

For the case when both A and C matrices in equation (1-2.1) are constant, equation (4-2.19) simplifies to

$$\hat{\boldsymbol{x}}[k] = \beta\Phi^{-1}A^{-T}\Phi\,\hat{\boldsymbol{x}}[k-1] + \Phi^{-1}C^{T}\boldsymbol{y}[k] - \beta^{p}\Phi^{-1}\left(A^{T}\right)^{-p}C^{T}\boldsymbol{y}[k-p] \tag{4-4.1}$$

Note that Φ^{-1} and Φ in (4-4.1) can be computed offline. Taking z-transform of (4-4.1), we obtain the corresponding transfer function as,

$$H(z) = \Phi^{-1}\left(I - A^{-T}z^{-1}\right)^{-1}\left(I - A^{-pT}z^{-p}\right)C^{T} \tag{4-4.2}$$

Using the algebraic formula

$$(1-a^{n}) = (1-a)(1+a+a^{2}+\cdots+a^{n-1})$$

equation (4-4.2) may be written as

$$H(z) = \Phi^{-1}\left(I - \beta A^{-T} z^{-1}\right)^{-1}\left(I - \beta^{p} A^{-T} z^{-1}\right)\times$$
$$\left(I + \beta A^{-T} z^{-1} + \beta^{2} A^{-2T} z^{-2} + \cdots + \beta^{p-1} A^{-(p-1)T} z^{-(p-1)}\right) C^{T}$$

which may be simplified into [52]

$$H(z) = \Phi^{-1}\left(\sum_{i=0}^{p-1} \beta^{i} A^{-iT} z^{-i}\right) C^{T} \tag{4-4.3}$$

4.5 Stability

Provided each observation value is finite in magnitude, the solution (4-2.19) suggests that the filter is inherently stable as it is based on finite number of observations, an FIR filter [48]. Further insight into stability of the filter is obtained when we look at the transfer function representation (4-4.3) which shows that it is actually an FIR filter. The result is intuitively satisfying due to state estimate based on finite number of the past observations.

4.6 Initialization

Two different methods are suggested for initializing the recursive algorithm (4-2.19). These are same as those used for state-space RLS (SSRLS), with exponential forgetting [35].

The first method employs regularization term. Since $\Phi[0], \Phi[1], \cdots, \Phi[l-2]$ are rank-deficient due to l-step observability of (1-3.1), we may use either $\Phi[0] = \delta I$ or more appropriately $\Phi[0] = \delta I + C[0]^{T} C[0]$ if $C[0]$ is available where δ is a positive real number. The choice of δ depends on signal to noise ratio (SNR) [13]. Moreover, to start the recursion we may take $\hat{\boldsymbol{x}}[0] = \mathbf{0}$ if no other estimates are available.

The second initialization method uses delayed recursion [35]. This method proved to be a logical and efficient method of initializing SSRLS [35]. The same is preferred for the FIR adaptive filter also.

4.7 Equivalence of FIR Adaptive Filter and State-Space RLS

State-space RLS (SSRLS) [35] takes into consideration the state-space model of the system and is, therefore, a very useful generalization of the standard RLS. With forgetting factor $0 < \lambda < 1$, SSRLS *forgets* the past observations exponentially. This contrasts with the FIR adaptive filter which is based on finite number of observations of equal weightage. For $\lambda = 1$, SSRLS is equivalent to Kalman filter [13] which is optimal in the sense of *mean square error*.

In this section, we show that FIR adaptive filter is equivalent to SSRLS [35] if certain conditions hold. This equivalence gives some insight into the optimality properties associated with the FIR adaptive filter.

Proposition 4-1

If $p \to \infty$ and $\beta = \lambda$, then FIR adaptive filter (4-2.19) *and SSRLS are equivalent.*

Proof

For $p \to \infty$ and $\beta = \lambda$, equation (4-2.11) is reduced to

$$\Phi[k] = \lambda A^T[k-1,k]\Phi[k-1]A[k-1,k] + C^T[k]C[k] \tag{4-7.1}$$

Making use of Matrix Inversion Lemma [13], equation (4-7.1) gives the following Riccati equation after some algebraic manipulations

$$
\begin{aligned}
\Phi^{-1}[k] = {} & \lambda^{-1} A[k,k-1]\Phi^{-1}[k-1]A^T[k,k-1] - \lambda^{-2} A[k,k-1]\Phi^{-1}[k-1] \times \\
& A^T[k,k-1]C^T[k]\left(I + \lambda^{-1} C[k]A[k,k-1]\Phi^{-1}[k-1]A^T[k,k-1]C^T[k]\right)^{-1} \times \\
& C[k]A[k,k-1]\Phi^{-1}[k-1]A^T[k,k-1]
\end{aligned} \tag{4-7.2}
$$

Equation (4-7.2) is same as that for SSRLS [35]. The estimated state for SSRLS may be written as [35]

$$
\hat{\boldsymbol{x}}[k] = \left(I - \Phi^{-1}[k]C^T[k]C[k]\right) A[k,k-1]\hat{\boldsymbol{x}}[k-1] + \Phi^{-1}[k]C^T[k]\boldsymbol{y}[k] \tag{4-7.3}
$$

On the other hand, for $p \to \infty$ and $\beta = \lambda$ equation (4-2.19) is reduced to

$$
\hat{\boldsymbol{x}}[k] = \lambda \Phi^{-1}[k]A^T[k-1,k]\Phi[k-1]\,\hat{\boldsymbol{x}}[k-1] + \Phi^{-1}[k]C^T[k]\boldsymbol{y}[k] \tag{4-7.4}
$$

From (4-7.1) and (4-7.4), we have

$$
\begin{aligned}
\hat{\boldsymbol{x}}[k] &= \lambda \Phi^{-1}[k]A^T[k-1,k]\Big(\lambda^{-1} A^T[k,k-1]\Phi[k]A[k,k-1] - \\
&\qquad \lambda^{-1} A^T[k,k-1]C^T[k]C[k]A[k,k-1]\Big)\hat{\boldsymbol{x}}[k-1] + \Phi^{-1}[k]C^T[k]\boldsymbol{y}[k] \\
&= \left(I - \Phi^{-1}[k]C^T[k]C[k]\right)A[k,k-1]\hat{\boldsymbol{x}}[k-1] + \Phi^{-1}[k]C^T[k]\boldsymbol{y}[k]
\end{aligned} \tag{4-7.5}
$$

Equation (4-7.3) and second equation of (4-7.5) are same, hence confirming the equivalence.

4.8 Convergence Analysis

In this section we analyze convergence behavior of the FIR adaptive filter (4-2.19). After making certain assumptions, we show that the proposed filter converges in mean in finite time. A discussion of mean square deviation follows.

4.8.1 Assumptions

Certain assumptions were made about the system (1-2.1) in Section 1.2.1. In order to proceed with a statistical analysis we have to make additional assumptions about the nature of the observation noise $\boldsymbol{v}[k]$.

4.8.1.1 Assumption I

Observation noise $\boldsymbol{v}[k]$ is assumed to be zero mean i.e.

$$E\left[\boldsymbol{v}[k]\right] = \mathbf{0} \tag{4-8.1}$$

where $E[\cdot]$ is the expectation operator.

4.8.1.2 Assumption II

Observation noise $\boldsymbol{v}[k]$ is assumed to be white with

$$E\left[\boldsymbol{v}[k]\boldsymbol{v}^T[j]\right] = \begin{cases} \sigma_v^2 \mathbf{I} & if\ k = j \\ \mathbf{0} & otherwise \end{cases} \tag{4-8.2}$$

where σ_v^2 is variance of observation noise $\boldsymbol{v}[k]$.

4.8.2 Convergence in Mean

Lemma 4-1

When the pair $(A[k], C[k])$ *is l-step observable,* $A[k]$ *is invertible on the interval* $\left[k-p+1, k\right]$ *and* $k \geq p$, *the FIR adaptive filter* (4-2.19) *converges in mean i.e.* $E\left[\hat{\boldsymbol{x}}[k]\right] = \boldsymbol{x}[k]$ *on the interval* $\left[k-p+1, k\right]$.

Proof

From equation (4-2.3), $\Phi[k]$ may be written as

$$\Phi[k]=\sum_{i=0}^{p-1}\beta^i A^T[k-i,k]C^T[k-i]C[k-i]A[k-i,k] \tag{4-8.3}$$

Similarly from equation (4-2.17), $\boldsymbol{\zeta}[k]$ may be written as

$$\zeta[k]=\sum_{i=0}^{a}\beta^i A^T[k-i,k]C^T[k-i]\boldsymbol{y}[k-i], \qquad a=\begin{cases} p-1 & \text{if } k\geq p \\ k-1 & \text{otherwise}\end{cases} \tag{4-8.4}$$

The invertibility of the system matrix $A[k]$ enables us to write the second equation of (1-2.1) as

$$\boldsymbol{y}[i]=C[i]A[i,k]\boldsymbol{x}[k]+\boldsymbol{v}[i] \tag{4-8.5}$$

where $i=0,1,...,k$. We replace $\boldsymbol{y}[i]$ from (4-8.5) in (4-8.4) to get

$$\begin{aligned}\zeta[k]=&\left[\sum_{i=0}^{a}\beta^i A^T[k-i,k]C^T[k-i]C[k-i]A[k-i,k]\right]\boldsymbol{x}[k]+\\ &\sum_{i=0}^{a}\beta^i A^T[k-i,k]C^T[k-i]\boldsymbol{v}[k-i]\end{aligned} \tag{4-8.6}$$

Define

$$\begin{aligned}\Omega[k]=&\sum_{i=0}^{p-1}\beta^i A^T[k-i,k]C^T[k-i]C[k-i]A[k-i,k]-\\ &\sum_{i=0}^{a}\beta^i A^T[k-i,k]C^T[k-i]C[k-i]A[k-i,k]\\ =&\,\Phi[k]-\sum_{i=0}^{a}\beta^i A^T[k-i,k]C^T[k-i]C[k-i]A[k-i,k]\end{aligned} \tag{4-8.7}$$

where we have also used equation (4-8.3). Using second equation of (4-8.7) in (4-8.6), we get

$$\zeta[k]=(\Phi[k]-\Omega[k])\,\boldsymbol{x}[k]+\sum_{i=0}^{a}\beta^i A^T[k-i,k]C^T[k-i]\boldsymbol{v}[k-i] \tag{4-8.8}$$

From equation (4-2.2) and (4-8.8)

$$
\begin{aligned}
\hat{\boldsymbol{x}}[k] &= \Phi^{-1}[k]\zeta[k] \\
&= \boldsymbol{x}[k] - \Phi^{-1}[k]\Omega[k]\,\boldsymbol{x}[k] + \Phi^{-1}[k]\sum_{i=0}^{a}\beta^{i} A^{T}[k-i,k]C^{T}[k-i]\boldsymbol{v}[k-i]
\end{aligned}
\tag{4-8.9}
$$

Taking expectation of both sides and invoking assumption (4-8.1)

$$
\begin{aligned}
E\left[\hat{\boldsymbol{x}}[k]\right] &= \boldsymbol{x}[k] - \Phi^{-1}[k]\Omega[k]\,\boldsymbol{x}[k] + \Phi^{-1}[k]\sum_{i=0}^{a}\beta^{i} A^{T}[k-i,k]C^{T}[k-i]E\left[\boldsymbol{v}[k-i]\right] \\
&= \boldsymbol{x}[k] - \Phi^{-1}[k]\Omega[k]\boldsymbol{x}[k]
\end{aligned}
\tag{4-8.10}
$$

For $k < p$, we get a biased state estimate. Since $\Omega[k] = \mathbf{0}$ for $k \geq p$, the second term on the right side of second equation of (4-8.10) decays to zero as *k* approaches *p*. This shows that the initial conditions do not effect the foregoing analysis. Hence we have

$$
E\left[\hat{\boldsymbol{x}}[k]\right] = \boldsymbol{x}[k], \qquad k \geq p \tag{4-8.11}
$$

This proves convergence in mean in finite time.

4.8.3 Mean Square Deviation

Lemma 4-2

When the pair $(A[k], C[k])$ *is l-step observable,* $A[k]$ *is invertible on the interval* $\left[k-p+1, k\right]$ *and* $k \geq p$*, the mean square deviation for FIR adaptive filter* (4-2.19) *is given by* $D[k] = Tr\left[E\left\{\Phi^{-1}[k]\right\}W\right]\sigma_v^2$.

Proof

The estimation error is defined as the difference between the actual states and their estimates as follows.

$$
\boldsymbol{e}[k] = \boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k] \tag{4-8.12}
$$

The mean-square deviation is defined as

$$D[k] = E\left[\boldsymbol{e}^T[k]\boldsymbol{e}[k] \right] = Tr\left(E\left[\boldsymbol{e}^T[k]\boldsymbol{e}[k] \right] \right) = Tr\left(E\left[\boldsymbol{e}[k]\boldsymbol{e}^{\boldsymbol{T}}[k] \right] \right) \tag{4-8.13}$$

where we have used the fact $Tr(AB) = Tr(BA)$. For $k \geq p$, using second equation of (4-8.9) in (4-8.12) we get

$$\boldsymbol{e}[k] = -\Phi^{-1}[k]\sum_{i=0}^{p-1} \beta^i A^T[k-i,k]C^T[k-i]\boldsymbol{v}[k-i] \tag{4-8.14}$$

Replacing $\boldsymbol{e}[k]$ from (4-8.14) in (4-8.13) and recalling symmetric nature of $\Phi[k]$ from (4-2.1)

$$\begin{aligned} D[k] &= E\left[Tr\left\{ \Phi^{-1}[k]\left(\sum_{i=0}^{p-1}\sum_{j=0}^{p-1} \beta^i \beta^j A^T[k-i,k]C^T[k-i] \times \right.\right.\right. \\ &\qquad \left.\left.\left. \boldsymbol{v}[k-i]\boldsymbol{v}^T[k-j]C[k-j]A[k-j,k] \right) \Phi^{-1}[k] \right\} \right] \\ &= Tr\left[E\left\{ \Phi^{-1}[k]\left(\sum_{i=0}^{p-1}\sum_{j=0}^{p-1} \beta^i \beta^j A^T[k-i,k]C^T[k-i] \times \right.\right.\right. \\ &\qquad \left.\left.\left. \boldsymbol{v}[k-i]\boldsymbol{v}^T[k-j]C[k-j]A[k-j,k] \right) \Phi^{-1}[k] \right\} \right] \end{aligned}$$

Invoking Assumption (4-8.2)

$$D[k] = Tr\left[E\left\{ \Phi^{-1}[k]\left(\sum_{i=0}^{p-1} \beta^{2i} A^T[k-i,k]C^T[k-i]C[k-i]A[k-i,k] \right) \Phi^{-1}[k] \right\} \right] \sigma_v^2 \tag{4-8.15}$$

In equation (4-8.15), identifying the term in round brackets as $\Phi[k]$ of equation (4-8.3) post multiplied by W of (4-1.6)

$$\begin{aligned} D[k] &= Tr\left[E\left\{ \Phi^{-1}[k]\Phi[k]W\,\Phi^{-1}[k] \right\} \right] \sigma_v^2 \\ &= Tr\left[E\left\{ \Phi^{-1}[k] \right\} W \right] \sigma_v^2 \end{aligned} \tag{4-8.16}$$

4.9 Example (Tracking Van der Pol Oscillations)

We demonstrate the performance of FIR adaptive filter by tracking Van der Pol oscillations [22]. This example was previously taken up in Section 3.4 to evaluate performance of SSLMSWAM. We have used equations for Van der Pol oscillator (3-4.1) to generate the oscillations.

Using constant acceleration state-space model (3-4.2) , we observe the signal x_2 in (3-4.1) in discrete domain after sampling with sampling time $T = 0.01s$. Zero mean white Gaussian noise with variance 0.1 corrupts the observations. We have taken $p = 70$, step-size parameter $\mu = 0.45$ for LMS and the forgetting factor $\lambda = 0.4$ for RLS. LMS and RLS are taken to be 3-tap filters. All filters are assumed to be initially at rest. The simulation results as illustrated in Figure 4-1 demonstrate the performance of FIR adaptive filter, LMS and RLS.

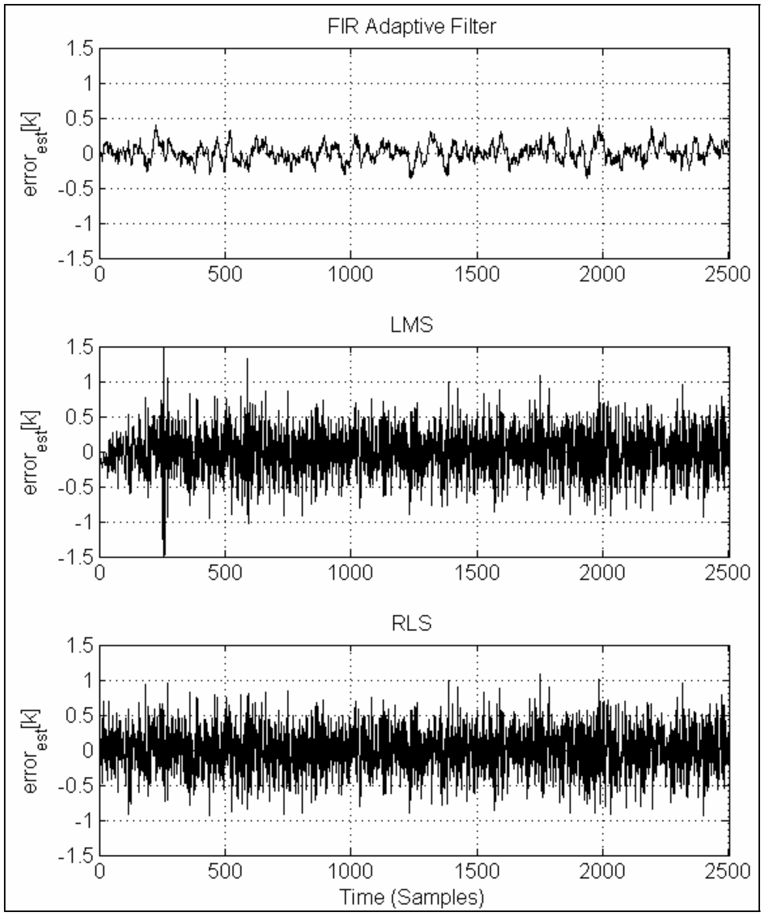

Figure 4-1. Error for FIR Adaptive Filter, LMS and RLS – Van der Pol Oscillations

4.9.1 Comments

We have used constant acceleration model as an approximation to actual Van der Pol oscillator model. This approximation is achieved by the constant acceleration model by fitting 3rd order polynomial on various segments of Van der Pol oscillations. Due to time varying Van der Pol signal, extent of model match/mismatch varies with time. Despite this model mismatch, performance of FIR adaptive filter is better than LMS and RLS.

4.10 Adaptive Memory Variant of FIR Adaptive Filter

For FIR adaptive filter to perform well, the model incorporated into its state-space formulation must match the model of the underlying environment, as closely as possible. In case model of the underlying environment is completely/partially unknown, one may work with a presumed model. Model mismatch may result if the presumed model is different from the underlying environment.

Partial compensation for model mismatch may be achieved if we adaptively tune the filter length *p* (the filter memory), so as to minimize a cost function. The approach is similar to that adopted for developing SSLMSWAM in CHAPTER 3 where step-size parameter μ was tuned iteratively. The foregoing development differs from SSLMSWAM in the sense that the filter length *p* is a positive *integer* whereas μ is a positive *real* number. Adaptive memory variant of FIR adaptive filter iteratively tunes the filter length *p* so as to minimize the following cost function

$$J[k] = E\left[\varepsilon^T[k]\varepsilon[k]\right] \tag{4-10.1}$$

where $E\left[\cdot\right]$ is the expectation operator and $\varepsilon[k]$ is the prediction error defined in (2-2.2). Using finite forward difference method defined as [13]

$$\Delta a[n] \equiv a[n+1] - a[n]$$

we may obtain finite forward difference of $J[k]$ (4-10.1) with respect to *p* as follows

$$
\begin{aligned}
\Delta_p[k] &= \Delta_p E\left[\varepsilon^T[k]\varepsilon[k]\right] \\
&= E\left[\Delta_p\left\{\left(\boldsymbol{y}[k]-C[k]A[k-1]\hat{\boldsymbol{x}}_{\boldsymbol{p}}[k-1]\right)^T\left(\boldsymbol{y}[k]-C[k]A[k-1]\hat{\boldsymbol{x}}_{\boldsymbol{p}}[k-1]\right)\right\}\right] \\
&= E\Big[\Delta_p\Big\{\left(\boldsymbol{y}^{\boldsymbol{T}}[k]-\hat{\boldsymbol{x}}_{\boldsymbol{p}}^{\boldsymbol{T}}[k-1]A^T[k-1]C^T[k]\right)\times \\
&\qquad\qquad \left(\boldsymbol{y}[k]-C[k]A[k-1]\hat{\boldsymbol{x}}_{\boldsymbol{p}}[k-1]\right)\Big\}\Big] \\
&= E\Big[\Delta_p\Big\{\boldsymbol{y}^{\boldsymbol{T}}[k]\boldsymbol{y}[k]-\boldsymbol{y}^{\boldsymbol{T}}[k]C[k]A[k-1]\hat{\boldsymbol{x}}_{\boldsymbol{p}}[k-1]- \\
&\qquad\qquad \hat{\boldsymbol{x}}_{\boldsymbol{p}}^{\boldsymbol{T}}[k-1]A^T[k-1]C^T[k]\boldsymbol{y}[k]+ \\
&\qquad\qquad \hat{\boldsymbol{x}}_{\boldsymbol{p}}^{\boldsymbol{T}}[k-1]A^T[k-1]C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}_{\boldsymbol{p}}[k-1]\Big\}\Big] \qquad (4\text{-}10.2) \\
&= E\Big[-\boldsymbol{y}^{\boldsymbol{T}}[k]C[k]A[k-1]\Delta_p\hat{\boldsymbol{x}}_{\boldsymbol{p}}[k-1]-\Delta_p\hat{\boldsymbol{x}}_{\boldsymbol{p}}^{\boldsymbol{T}}[k-1]A^T[k-1]C^T[k]\boldsymbol{y}[k]+ \\
&\qquad \Delta_p\left(\hat{\boldsymbol{x}}_p^{\boldsymbol{T}}[k-1]A^T[k-1]C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}_{\boldsymbol{p}}[k-1]\right)\Big\}\Big]
\end{aligned}
$$

where we have made use of equations (2-2.1) through (2-2.3). The dependence of $\hat{\boldsymbol{x}}[k]$ on filter length p is shown explicity in (4-10.2) using subscript notation. We may write last equation of (4-10.2) as

$$
\begin{aligned}
\Delta_p[k] = E\Big[&-\boldsymbol{y}^T[k]C[k]A[k-1]\left(\hat{\boldsymbol{x}}_{p+1}[k-1]-\hat{\boldsymbol{x}}_p[k-1]\right)- \\
&\left(\hat{\boldsymbol{x}}_{p+1}^T[k-1]-\hat{\boldsymbol{x}}_p^T[k-1]\right)A^T[k-1]C^T[k]\boldsymbol{y}[k]+ \\
&\hat{\boldsymbol{x}}_{p+1}^T[k-1]A^T[k-1]C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}_{p+1}[k-1]- \\
&\hat{\boldsymbol{x}}_p^T[k-1]A^T[k-1]C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}_p[k-1]\Big]
\end{aligned}
\qquad (4\text{-}10.3)
$$

Based on (4-10.3), an instantaneous estimate for the scalar gradient $\Delta_p[k]$ can be taken as

$$
\begin{aligned}
\Delta_p[k] = \Big[&-\boldsymbol{y}^T[k]C[k]A[k-1]\left(\hat{\boldsymbol{x}}_{p+1}[k-1]-\hat{\boldsymbol{x}}_p[k-1]\right)- \\
&\left(\hat{\boldsymbol{x}}_{p+1}^T[k-1]-\hat{\boldsymbol{x}}_p^T[k-1]\right)A^T[k-1]C^T[k]\boldsymbol{y}[k]+ \\
&\hat{\boldsymbol{x}}_{p+1}^T[k-1]A^T[k-1]C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}_{p+1}[k-1]- \\
&\hat{\boldsymbol{x}}_p^T[k-1]A^T[k-1]C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}_p[k-1]\Big]
\end{aligned}
\qquad (4\text{-}10.4)
$$

Now we are in a position to formulate adaptive memory variant of FIR adaptive filter. The stochastic gradient method that updates $p[k]$, which in turn is a function of time, is given by [13]

$$p[k] = \left[round\left(p[k-1] - \alpha \Delta_p [k] \right) \right]_{p-}^{p+} \tag{4-10.5}$$

where α is a small positive learning rate parameter. The *rounding* operation in (4-10.5) is necessary to ensure the updated filter length $p[k]$ remains an integer. For the algorithm (4-10.5) to be meaningful we require $p \geq l$. The bracket followed by p_- and p_+ in equation (4-10.5) indicates saturation that restricts the filter length to $\left[p_-, p_+ \right]$, assuming $p_+ > p_-$. The lower limit is generally set close to l, whereas the upper limit depends on the nature of the problem. Its value is determined through experimentation. Expressing the dependence of various quantities on updated filter length $p[k]$ using subscript notation, the adaptive memory variant of FIR adaptive filter is summarized below

$$\begin{aligned}
\Delta_{p[k-1]}[k] = \Big[& -\boldsymbol{y}^T[k]C[k]A[k-1]\left(\hat{\boldsymbol{x}}_{p[k-1]+1}[k-1] - \hat{\boldsymbol{x}}_{p[k-1]}[k-1] \right) - \\
& \left(\hat{\boldsymbol{x}}^T_{p[k-1]+1}[k-1] - \hat{\boldsymbol{x}}^T_{p[k-1]}[k-1] \right) A^T[k-1]C^T[k]\boldsymbol{y}[k] + \\
& \hat{\boldsymbol{x}}^T_{p[k-1]+1}[k-1]A^T[k-1]C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}_{p[k-1]+1}[k-1] - \\
& \hat{\boldsymbol{x}}^T_{p[k-1]}[k-1]A^T[k-1]C^T[k]C[k]A[k-1]\hat{\boldsymbol{x}}_{p[k-1]}[k-1] \Big] \\
p[k] = & \left[round\left(p[k-1] - \alpha \Delta_{p[k-1]}[k] \right) \right]_{p-}^{p+} \\
\hat{\boldsymbol{x}}_{p[k]}[k] = & \beta \Phi^{-1}_{p[k]}[k] A^T[k-1,k] \Phi_{p[k]}[k-1] \hat{\boldsymbol{x}}_{p[k]}[k-1] + \Phi^{-1}_{p[k]}[k] C^T[k] \boldsymbol{y}[k] - \\
& \beta^p \Phi^{-1}_{p[k]}[k] \Psi^T_{p[k]}[k] \boldsymbol{y}\left[k - p[k] \right]
\end{aligned} \tag{4-10.6}$$

In order to implement the filter (4-10.6), a total of $p_+ - p_- + 2$ estimators $\hat{\boldsymbol{x}}_i[k]$ (differing in filter length) work in parallel, where

$i = p_-, p_- + 1, p_- + 2, \ldots\ldots, p_+ - 1, p_+, p_+ + 1$. In each recursion, the filter (4-10.6) uses one of these estimators i.e. $\hat{\boldsymbol{x}}_{p[k]}[k]$ in its third equation, whereas the first equation computes $\Delta_{p[k-1]}[k]$ using two of these estimators $\hat{\boldsymbol{x}}_{p[k-1]}[k-1]$ and $\hat{\boldsymbol{x}}_{p[k-1]+1}[k-1]$.

CHAPTER 5

RECEDING HORIZON STATE OBSERVER FOR LINEAR TIME-VARYING SYSTEMS

In CHAPTER 4 we developed finite impulse response (FIR) adaptive filter built around state-space framework of an unforced system. The inherent stability of FIR adaptive filter is a desirable feature that makes it suitable for a variety of applications. This feature, which is due to finite-horizon based estimation, also finds its usefulness when the system is forced and the requirement is to estimate (observe) the state of an unknown system using its inputs and outputs on a finite interval.

Motivated by the idea of FIR adaptive filter, this chapter develops receding horizon state observer for time-varying forced linear discrete systems [54]. The observer is inherently stable by virtue of inputs and outputs on a finite interval. The observer does not require *a priori* knowledge of properties of process and/or observation noise and hence it works in presence/absence of both or any one of the noises. The performance of the observer is however dependent on various factors like time-varying nature of the input and output signals, model uncertainty, and/or nonstationary behavior of the observation noise.

The development starts with stating the state observation problem for a forced system. Based on state-space framework, we give batch processed least squares solution to the state estimate. The discussion is then extended to a recursive solution. The derivations for time-varying solutions are naturally extended to the time-invariant case.

First and second order convergence analyses of the observer follow that give insight into stochastic behavior of the observer.

We demonstrate the performance of the observer using a computer simulation where we consider regulation problem of a linear time-varying system.

5.1 Problem Statement

Consider the discrete-time system (1-3.1). At this point, we make no assumptions about the nature of $\boldsymbol{v}[k]$. However, attributing certain statistical properties to $\boldsymbol{v}[k]$ will assist us to carry out convergence analysis later in Section 5.7. We assume that the pair $(A[k], C[k])$ is *l*-step observable [52] and $A[k]$ is invertible for $k > 0$. The state-transition matrix for the system (1-3.1) is given by equation (2-1.1) and its properties given by (2-1.2).

The initial state of the system, $\boldsymbol{x}[0]$ (and hence the initial output $\boldsymbol{y}[0]$) is assumed to be unavailable for measurement. Let the inputs and outputs start to appear at time $k=1$. We assume that $\boldsymbol{y}[k]$ and $\boldsymbol{u}[k]$ will remain available for $k \geq 1$. Furthermore, data pre-windowing is assumed i.e. $\boldsymbol{y}[k]=0$ and $\boldsymbol{u}[k]=0$ for $k \leq 0$.

The goal of an observer is to design an $n \times 1$ vector function $\hat{\boldsymbol{x}}[k]$ that is the estimate of $\boldsymbol{x}[k]$ such that

$$\lim_{k \to \infty} \left[\boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k] \right] = 0 \tag{5-1.1}$$

If the observation noise is statistical in nature, it is more appropriate to consider convergence in mean i.e.

$$\lim_{k \to \infty} E\left[\boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k] \right] = 0 \tag{5-1.2}$$

where $E[\bullet]$ is the expectation operator. In the classical observer design, the input $\boldsymbol{u}[k]$ and output $\boldsymbol{y}[k]$ for all $k \geq 1$ are utilized for calculating the state estimate governed by linear state equation of the form [52]

$$\begin{aligned} \hat{\boldsymbol{x}}[k+1] &= A[k]\hat{\boldsymbol{x}}[k] + B[k]\boldsymbol{u}[k] + Q[k]\left(\boldsymbol{y}[k] - \hat{\boldsymbol{y}}[k]\right) \\ \hat{\boldsymbol{y}}[k] &= C[k]\,\hat{\boldsymbol{x}}[k] \end{aligned} \tag{5-1.3}$$

where $Q[k]$ is the *observer gain* matrix. On the hand, in the receding horizon philosophy, an additional restriction is imposed; only a finite number of the last (say p) measurements of the input and output are used. Alternately speaking, the observations are purposefully windowed. Formal mathematical treatment follows in the next sections.

5.2 Batch Processed Least Squares State Estimate

We begin our development with the design of a state estimator that is based on batch processed least squares approach [13]. The batch of measurements consists of the inputs $\boldsymbol{u}[k]$ and outputs $\boldsymbol{y}[k]$ on the interval $\left[k-p+1, k\right]$, where p is a positive integer such that $p \geq l$. From equation (1-3.1), we may write

$$\begin{aligned} \boldsymbol{y}[k-1] &= C[k-1]\boldsymbol{x}[k-1] + \boldsymbol{v}[k-1] \\ &= C[k-1]A[k-1,k]\boldsymbol{x}[k] - C[k-1]A[k-1,k]\Upsilon[k-1] + \boldsymbol{v}[k-1] \end{aligned}$$

where we have defined $\Upsilon[k] = B[k]\boldsymbol{u}[k]$ for notational convenience and used equations (2-1.1) and (2-1.2). Similarly, we can write p different equations from (1-3.1) as follows

$$\boldsymbol{\mathcal{Y}}[k] = \begin{bmatrix} \boldsymbol{y}[k-p+1] \\ \boldsymbol{y}[k-p+2] \\ \vdots \\ \boldsymbol{y}[k-2] \\ \boldsymbol{y}[k-1] \\ \boldsymbol{y}[k] \end{bmatrix} = H[k]\boldsymbol{x}[k] - \Delta[k] + \boldsymbol{\mathcal{V}}[k], \tag{5-2.1}$$

where the observation noise vector is given by

$$\boldsymbol{\mathcal{V}}[k]=\left[\boldsymbol{v}^T[k-p+1] \quad \boldsymbol{v}^T[k-p+2] \quad \cdots \quad \boldsymbol{v}^T[k-1] \quad \boldsymbol{v}^T[k]\right]^T, \tag{5-2.2}$$

$H[k]\in\mathbb{R}^{mp\times n}$ is defined as

$$H[k]=\begin{bmatrix} C[k-p+1]A[k-p+1,k] \\ C[k-p+2]A[k-p+2,k] \\ \vdots \\ C[k-2]A[k-2,k] \\ C[k-1]A[k-1,k] \\ C[k] \end{bmatrix} \tag{5-2.3}$$

and $\Delta[k]\in\mathbb{R}^{mp\times 1}$ is defined as

$$\Delta[k]=\begin{bmatrix} C[k-p+1]A[k-p+1,k]\big(\Upsilon[k-1]+A[k,k-1]\Upsilon[k-2]+\cdots+A[k,k-p+2]\Upsilon[k-p+1]\big) \\ C[k-p+2]A[k-p+2,k]\big(\Upsilon[k-1]+A[k,k-1]\Upsilon[k-2]+\cdots+A[k,k-p+3]\Upsilon[k-p+2]\big) \\ \vdots \\ C[k-2]A[k-2,k]\big(\Upsilon[k-1]+A[k,k-1]\Upsilon[k-2]\big) \\ C[k-1]A[k-1,k]\Upsilon[k-1] \\ \mathbf{0} \end{bmatrix}. \tag{5-2.4}$$

The solution of system (5-2.1) in terms of least squares is given as follows [13]

$$\hat{\boldsymbol{x}}[k]=(H^T[k]H[k])^{-1}H^T[k]\big(\boldsymbol{\mathcal{Y}}[k]+\Delta[k]\big) \tag{5-2.5}$$

The condition $p\geq l$ along with the observability assumption ensures the invertibility of $H^T[k]H[k]$. The dimension of $(H^T[k]H[k])^{-1}H^T[k]$ is $n\times mp$.

We introduce a weighting factor [13] such that the inputs and outputs on the interval $\left[k-p+1,k\right]$ are given lesser weight as they become old; the most recent being with the largest weight. With $0<\beta\leq 1$ as the weighting factor, the corresponding weighting matrix is given by (4-1.6). Taking (4-1.6) into account, the solution of system (5-2.1) in terms of weighted least squares is given as follows [56]

$$\hat{\boldsymbol{x}}[k] = (H^T[k]WH[k])^{-1}H^T[k]W\left(\boldsymbol{\mathcal{Y}}[k] + \Delta[k]\right) \tag{5-2.6}$$

From equation (4-1.6), for the special case $\beta = 1$ we get $W = I$ i.e. a rectangular window. Consequently, the solution (5-2.6) and (5-2.5) become equivalent.

5.3 Recursive Solution

Let us define the following symbols

$$\begin{aligned} \Phi[k] &= H^T[k]WH[k] \\ \boldsymbol{\zeta}[k] &= H^T[k]W\boldsymbol{\mathcal{Y}}[k] \\ \Pi[k] &= H^T[k]W\Delta[k] \end{aligned} \tag{5-2.7}$$

From (5-2.6) and (5-2.7), we get

$$\hat{\boldsymbol{x}}[k] = \Phi^{-1}[k]\left(\boldsymbol{\zeta}[k] + \Pi[k]\right) \tag{5-2.8}$$

where $\Phi[k]$, $\Phi^{-1}[k]$ and $\zeta[k]$ may be computed recursively using equations (4-2.6), (4-2.15) and (4-2.18) respectively. We next find recursive solution of $\Pi[k]$.

5.3.1 Computation of $\Pi[k]$

From equation (5-2.1) and (5-2.7)

$$\begin{aligned} \Pi[k] = \Big[& \beta^{p-1}A^T[k-p+1,k]C^T[k-p+1]C[k-p+1]A[k-p+1,k]\times \\ & \left(\Upsilon[k-1] + A[k,k-1]\Upsilon[k-2] + \cdots + A[k,k-p+2]\Upsilon[k-p+1]\right) + \\ & \beta^{p-2}A^T[k-p+2,k]C^T[k-p+2]C[k-p+2]A[k-p+2,k]\times \\ & \left(\Upsilon[k-1] + A[k,k-1]\Upsilon[k-2] + \cdots + A[k,k-p+3]\Upsilon[k-p+2]\right) + \cdots + \\ & \beta^2 A^T[k-2,k]C^T[k-2]C[k-2]A[k-2,k]\left(\Upsilon[k-1] + A[k,k-1]\Upsilon[k-2]\right) + \\ & \beta A^T[k-1,k]C^T[k-1]C[k-1]A[k-1,k]\Upsilon[k-1] \Big] \end{aligned} \tag{5-2.9}$$

Following a similar approach as in Section 4.2.3, equation (5-2.9) leads to following recursion

$$\Pi[k] = \beta A^T[k-1,k]\Pi[k-1] + \left(\Phi[k] - C^T[k]C[k]\right)\Upsilon[k-1] - \beta^p A^T[k-p,k]C^T[k-p]C[k-p]A[k-p,k-1]\times \left(\Upsilon[k-2] + A[k-1,k-2]\Upsilon[k-3] + \cdots + A[k-1,k-p+1]\Upsilon[k-p]\right) \quad (5\text{-}2.10)$$

Define

$$\eta[k] = \Upsilon[k-2] + A[k-1,k-2]\Upsilon[k-3] + \cdots + A[k-1,k-p+1]\Upsilon[k-p] \quad (5\text{-}2.11)$$

which also gives us

$$\eta[k-1] = \Upsilon[k-3] + A[k-2,k-3]\Upsilon[k-4] + \cdots + A[k-2,k-p]\Upsilon[k-p-1] \quad (5\text{-}2.12)$$

From (5-2.11) and (5-2.12), we get following recursion

$$\eta[k] = A[k-1,k-2]\eta[k-1] + \Upsilon[k-2] - A[k-1,k-p]\Upsilon[k-p-1] \quad (5\text{-}2.13)$$

Making use of (4-2.7), (4-2.10) and (5-2.13), we get following recursion from (5-2.10)

$$\Pi[k] = \beta A^T[k-1,k]\Pi[k-1] + \left(\Phi[k] - C^T[k]C[k]\right)\Upsilon[k-1] - \beta^p \Psi^T[k]\Psi[k]A[k,k-1]\eta[k] \quad (5\text{-}2.14)$$

5.3.2 Complete Algorithm

Equations (5-2.8), (4-2.15), (4-2.18) and (5-2.14) give the state estimate in its general form, as summarized below

$$\begin{aligned}
\hat{\boldsymbol{x}}[k] &= \Phi^{-1}[k]\left(\boldsymbol{\zeta}[k] + \Pi[k]\right)\\
\Phi^{-1}[k] &= A[k,k-1]\Lambda^{-1}[k]A^T[k,k-1] - A[k,k-1]\Lambda^{-1}[k]A^T[k,k-1]C^T[k]\times\\
&\quad \left(I + C[k]A[k,k-1]\Lambda^{-1}[k]A^T[k,k-1]C^T[k]\right)^{-1}\times\\
&\quad C[k]A[k,k-1]\Lambda^{-1}[k]A^T[k,k-1]\\
\boldsymbol{\zeta}[k] &= \beta A^T[k-1,k]\boldsymbol{\zeta}[k-1] + C^T[k]\boldsymbol{y}[k] - \beta^p A^T[k-p,k]C^T[k-p]\boldsymbol{y}[k-p]\\
\Pi[k] &= \beta A^T[k-1,k]\Pi[k-1] + \left(\Phi[k] - C^T[k]C[k]\right)\Upsilon[k-1] -\\
&\quad \beta^p \Psi^T[k]\Psi[k]A[k,k-1]\eta[k]
\end{aligned} \quad (5\text{-}2.15)$$

The recursive formulae in the third and fourth equations of (5-2.15) are based on finite number of inputs and outputs, respectively. The nature of recursion is such that as the latest input $\boldsymbol{u}[k]$ and output $\boldsymbol{y}[k]$ are incorporated to obtain updated state estimate, the effect of the input $\boldsymbol{u}[k-p]$ and output $\boldsymbol{y}[k-p]$ is removed. Thus the observer (5-2.15) has a finite impulse response (FIR). This behavior has been termed as receding horizon by some authors (see for example [27] – [29]). The result is also intuitively satisfying due to windowing of the inputs and outputs on the interval $\left[k-p+1,k\right]$. This is in contrast with the exponential forgetting of the past observations as in the case of RLS [13] and SSRLS filters [35], where it takes infinitely long time to completely forget a past observation. Thus, the standard RLS or SSRLS are infinite impulse response (IIR) filters. Accordingly, these filters always have one or more poles if considered in the linear time invariant (LTI) case. On the other hand, the proposed observer is an all-zero system [48], when applied in a time-invariant framework. This would be discussed in Section 5.5.

5.3.3 Alternate Form

The relation (5-2.8) can be construed as a change of variables [52], which transforms (4-2.18) into

$$\begin{aligned}\hat{\boldsymbol{x}}[k] &= \beta\Phi^{-1}[k]A^{T}[k-1,k]\Phi[k-1]\hat{\boldsymbol{x}}[k-1]+\Phi^{-1}[k]C^{T}[k]\boldsymbol{y}[k]-\\ &\beta^{p}\Phi^{-1}[k]\Psi^{T}[k]\boldsymbol{y}[k-p]+\Phi^{-1}[k]\Pi[k]-\beta\Phi^{-1}[k]A^{T}[k-1,k]\Pi[k-1]\end{aligned} \tag{5-3.1}$$

Equation (5-3.1) completely specifies the map from inputs and outputs to the state estimate $\hat{\boldsymbol{x}}[k]$ on the interval $\left[k-p+1,k\right]$.

5.4 Time Invariant Form

For the case when both A and C matrices in equation (1-3.1) are constant, equation (5-3.1) simplifies to

$$\begin{aligned}\hat{\boldsymbol{x}}[k] = \beta\Phi^{-1}A^{-T}\Phi\,\hat{\boldsymbol{x}}[k-1] + \Phi^{-1}C^{T}\boldsymbol{y}[k] - \beta^{p}\Phi^{-1}\left(A^{T}\right)^{-p}C^{T}\boldsymbol{y}[k-p] + \\ \Phi^{-1}\Pi[k] - \beta\Phi^{-1}A^{-T}\Pi[k-1]\end{aligned} \tag{5-4.1}$$

Note that Φ^{-1} and Φ in (5-4.1) can be computed offline.

5.5 Stability

The FIR nature of the receding horizon state observer suggests that it is inherently stable. This stability includes internal as well as bounded input bounded output (BIBO) stability ([48], [52]). This feature is more illustrative if we consider the linear time invariant case. Taking z-Transform of (5-4.1), we get

$$\hat{X}(z) = \Phi^{-1}\left(\sum_{i=0}^{p-1}\beta^{i}A^{-iT}z^{-i}\right)C^{T}Y(z) + \Phi^{-1}\left(\sum_{i=1}^{p-1}\beta^{i}A^{-iT}C^{T}CA^{-i}\sum_{j=0}^{i-1}A^{j}B\,z^{-j-1}\right)U(z) \tag{5-5.1}$$

Equation (5-5.1) shows that the proposed observer comprises of *zeros* only.

5.6 Initialization

The discussion on initialization done in Section 4.6 for FIR adaptive filter case is equally valid for the receding horizon state observer. One addition to this discussion is the choice of $\Pi[0]$. We take $\Pi[0] = \mathbf{0}$ due to data pre-windowing.

5.7 Convergence Analysis

In this section we analyze convergence behavior of the FIR adaptive filter. The assumptions that we use for this purpose are same as those specified in Section 4.8.1. We show that the receding horizon state observer converges in mean in finite time. A discussion of mean square deviation follows.

5.7.1 Convergence in Mean

Lemma 5-1

When the pair $(A[k], C[k])$ *is l-step observable,* $A[k]$ *is invertible on the interval* $\left[k-p+1, k\right]$ *and* $k \geq p$*, the receding horizon state observer (5-3.1) converges in mean i.e.* $E\left[\hat{\boldsymbol{x}}[k]\right] = \boldsymbol{x}[k]$ *on the interval* $\left[k-p+1, k\right]$.

Proof

From equation (5-2.9), $\Pi[k]$ may be written as

$$\Pi[k] = \sum_{i=1}^{a}\left(\beta^{i} A^{T}[k-i,k] C^{T}[k-i] C[k-i] A[k-i,k] \sum_{j=0}^{i-1} A[k,k-j] \Upsilon[k-j-1]\right), \quad (5\text{-}7.1)$$

where the definition of a is same as in equation (4-8.4). The invertibility of the system matrix $A[k]$ enables us to write the second equation of (1-3.1) as

$$\boldsymbol{y}[i] = \begin{cases} C[i]A[i,k]\boldsymbol{x}[k] - C[i]A[i,k]\left(\sum_{j=0}^{k-i-1} A[k,k-j]\Upsilon[k-j-1]\right) + \boldsymbol{v}[i] & i = 0,1,\ldots,k-1 \\ C[i]A[i,k]\boldsymbol{x}[k] + \boldsymbol{v}[i] & i = k \end{cases} \quad (5\text{-}7.2)$$

We replace $\boldsymbol{y}[i]$ from (5-7.2) in (4-8.4), and make use of (5-7.1) to get

$$
\begin{aligned}
\zeta[k] = &\left[\sum_{i=0}^{a} \beta^i A^T[k-i,k] C^T[k-i] C[k-i] A[k-i,k]\right] \boldsymbol{x}[k] - \\
&\sum_{i=1}^{a}\left(\beta^i A^T[k-i,k] C^T[k-i] C[k-i] A[k-i,k] \sum_{j=0}^{i-1} A[k,k-j] \Upsilon[k-j-1]\right) + \\
&\sum_{i=0}^{a} \beta^i A^T[k-i,k] C^T[k-i] \boldsymbol{v}[k-i] \\
= &\left[\sum_{i=0}^{a} \beta^i A^T[k-i,k] C^T[k-i] C[k-i] A[k-i,k]\right] \boldsymbol{x}[k] - \Pi[k] + \\
&\sum_{i=0}^{a} \beta^i A^T[k-i,k] C^T[k-i] \boldsymbol{v}[k-i]
\end{aligned} \tag{5-7.3}
$$

Define

$$
\begin{aligned}
\Omega[k] = &\sum_{i=0}^{p-1} \beta^i A^T[k-i,k] C^T[k-i] C[k-i] A[k-i,k] - \\
&\sum_{i=0}^{a} \beta^i A^T[k-i,k] C^T[k-i] C[k-i] A[k-i,k] \\
= &\ \Phi[k] - \sum_{i=0}^{a} \beta^i A^T[k-i,k] C^T[k-i] C[k-i] A[k-i,k]
\end{aligned} \tag{5-7.4}
$$

where we have also used equation (4-8.3). Using second equation of (5-7.4) in (5-7.3), we get

$$
\zeta[k] = (\Phi[k] - \Omega[k])\, \boldsymbol{x}[k] + \sum_{i=0}^{a} \beta^i A^T[k-i,k] C^T[k-i] \boldsymbol{v}[k-i] - \Pi[k] \tag{5-7.5}
$$

From second equation of (5-2.8) and (5-7.5)

$$
\begin{aligned}
\hat{\boldsymbol{x}}[k] &= \Phi^{-1}[k]\left(\zeta[k] + \Pi[k]\right) \\
&= \boldsymbol{x}[k] - \Phi^{-1}[k] \Omega[k]\, \boldsymbol{x}[k] + \Phi^{-1}[k] \sum_{i=0}^{a} \beta^i A^T[k-i,k] C^T[k-i] \boldsymbol{v}[k-i]
\end{aligned} \tag{5-7.6}
$$

Taking expectation of both sides and invoking assumption (4-8.1)

$$E\left[\hat{\boldsymbol{x}}[k]\right] = \boldsymbol{x}[k] - \Phi^{-1}[k]\Omega[k]\boldsymbol{x}[k] + \Phi^{-1}[k]\sum_{i=0}^{a}\beta^{i}A^{T}[k-i,k]C^{T}[k-i]E\left[\boldsymbol{v}[k-i]\right] \\ = \boldsymbol{x}[k] - \Phi^{-1}[k]\Omega[k]\boldsymbol{x}[k] \tag{5-7.7}$$

For $k < p$, we get a biased state estimate. Since $\Omega[k] = \mathbf{0}$ for $k \geq p$, the second term on the right side of second equation of (5-7.7) decays to zero as *k* approaches *p*. This shows that the initial conditions do not effect the foregoing analysis. Hence we have

$$E\left[\hat{\boldsymbol{x}}[k]\right] = \boldsymbol{x}[k], \qquad k \geq p \tag{5-7.8}$$

This proves convergence in mean in finite time.

5.7.2 Mean Square Deviation

Lemma 5-2

When the pair $(A[k], C[k])$ is l-step observable, $A[k]$ is invertible on the interval $\left[k-p+1, k\right]$ and $k \geq p$, the mean square deviation for the receding horizon state observer (5-3.1) is given by $D[k] = Tr\left[E\left\{\Phi^{-1}[k]\right\}W\right]\sigma_v^2$.

Proof

Same as for Lemma 4-2.

5.8 Example (Gyro Motion Control)

5.8.1 Sampled-Data Control of a Linear Time-Varying System

In this section, we demonstrate the receding horizon state observer in feedback control of a linear time-varying system. Consider a gyroscope described by the following state-space system

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \\ \boldsymbol{y}_m(t) &= C(t)\,\boldsymbol{x}(t) \\ \boldsymbol{y}(t) &= C'\,\boldsymbol{x}(t) \end{aligned} \tag{5-7.9}$$

where $\boldsymbol{x} \in \mathbb{R}^4$ is the state, $\boldsymbol{u} \in \mathbb{R}^2$ is the input, $\boldsymbol{y}_m \in \mathbb{R}$ is the output available for measurement and $\boldsymbol{y} \in \mathbb{R}^2$ is the output of the system that is required to follow a constant reference $\boldsymbol{r} = \begin{bmatrix} r_x & r_y \end{bmatrix}^T$. Our goal is to design an output feedback control that would achieve asymptotic regulation i.e.

$$\lim_{t\to\infty} \boldsymbol{y}(t) = \boldsymbol{r}$$

The system matrices corresponding to (5-7.9) are

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -H/J \\ 0 & 0 & 0 & 1 \\ 0 & H/J & 0 & 0 \end{bmatrix} \\ B &= \begin{bmatrix} 0 & 0 \\ K_i/J & 0 \\ 0 & 0 \\ 0 & K_i/J \end{bmatrix} \\ C(t) &= K_o \begin{bmatrix} \cos(\omega t) & 0 & \sin(\omega t) & 0 \end{bmatrix} \\ C' &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \tag{5-7.10}$$

It can be shown that the system (5-7.9) is controllable and observable. The observability of the system (5-7.9) facilitates state-observation, whereas controllability helps achieve successful regulation. The various parameters in equation (5-7.10) and their numerical values chosen for simulation are given below

H = angular momentum $= \left(J_z - J\right)\omega$

J = moment of inertia = 0.01 Kg-m^2

J_z = moment of inertia about z-axis = 0.017 Kg-m^2

ω = spin frequency of gyroscope about z-axis = 200π rad/sec

K_o = scaling factor that depends on mechanism of generating $\boldsymbol{y}_m(t)$ = 0.1

K_i = torque constant = 0.01

The gyroscope is assumed to be spinning **z** axis with a constant angular speed ω. The first and third states viz x_1 and x_3 represent the angular positions in radians of the gyroscope about **x** and **y** axes. The second and fourth states viz x_2 and x_4 (that correspond to spin frequencies about **x** and **y** axes, respectively) of the system (5-7.9) are the derivatives of first and third states, respectively. We are required to lock the gyroscope **x** and **y** positions onto a reference $\boldsymbol{r} = \begin{bmatrix}0.3 & 0.2\end{bmatrix}^T$. The plant (gyroscope) is assumed to be initially at $\boldsymbol{x}[0] = \begin{bmatrix}0.01 & 0.02 & 0.01 & 0.01\end{bmatrix}^T$. This information is not available to the observer and hence it is taken to be initially at rest. Continuous-time plant (5-7.9) is discretized using zero-order hold approach suggested by Nešić and Teel [47], with sampling time $T_s = 0.001\,\text{sec}$. The resultant discrete-time model, which is used by the observer (5-3.1) to estimate states of the system (5-7.9), is the exact discrete equivalent [47]. The estimated states are then used to implement state feedback control to regulate the gyroscope motion. The controller is designed by placing controller eigenvalues at $0.9 \pm \text{j}0.01$ and $0.9 \pm j0.02$. We have taken $p = 20$ and data pre-windowing is assumed i.e. $\boldsymbol{u}[k] = 0,\ \boldsymbol{y}_m[k] = 0$ for $k \leq 0$.

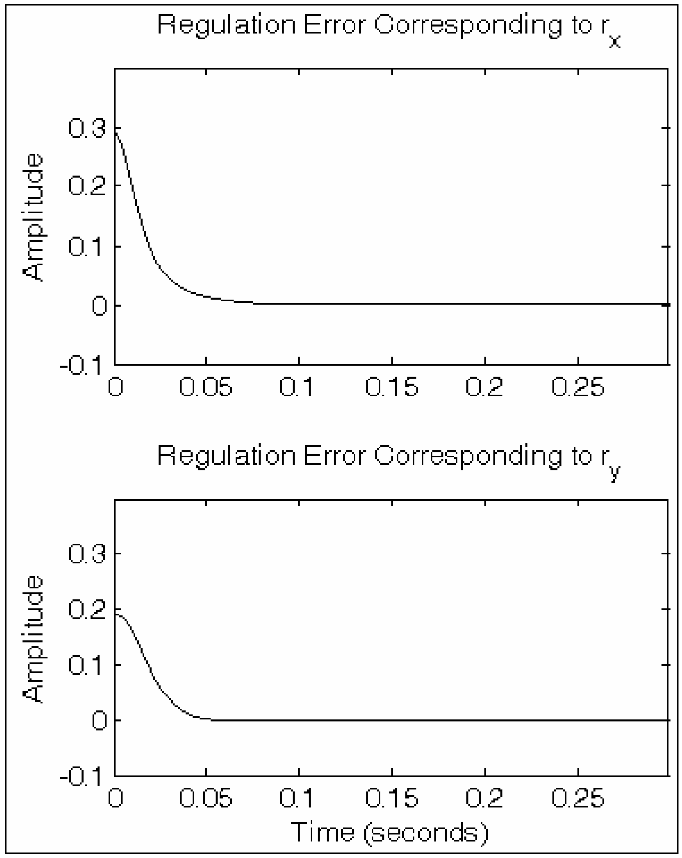

Figure 5-1. Regulation Error – Gyroscope in Motion

5.8.2 Remarks

Figure 5-1 shows the regulation error, which is defined as

$$\text{Regulation Error} = \boldsymbol{y}[k] - \boldsymbol{r}$$

Figure 5-2 shows the estimation error, which is defined as

$$\text{Estimation Error} = \boldsymbol{x}[k] - \hat{\boldsymbol{x}}[k]$$

Both types of errors are close to zero, thereby indicating good regulation achieved with receding horizon state observer.

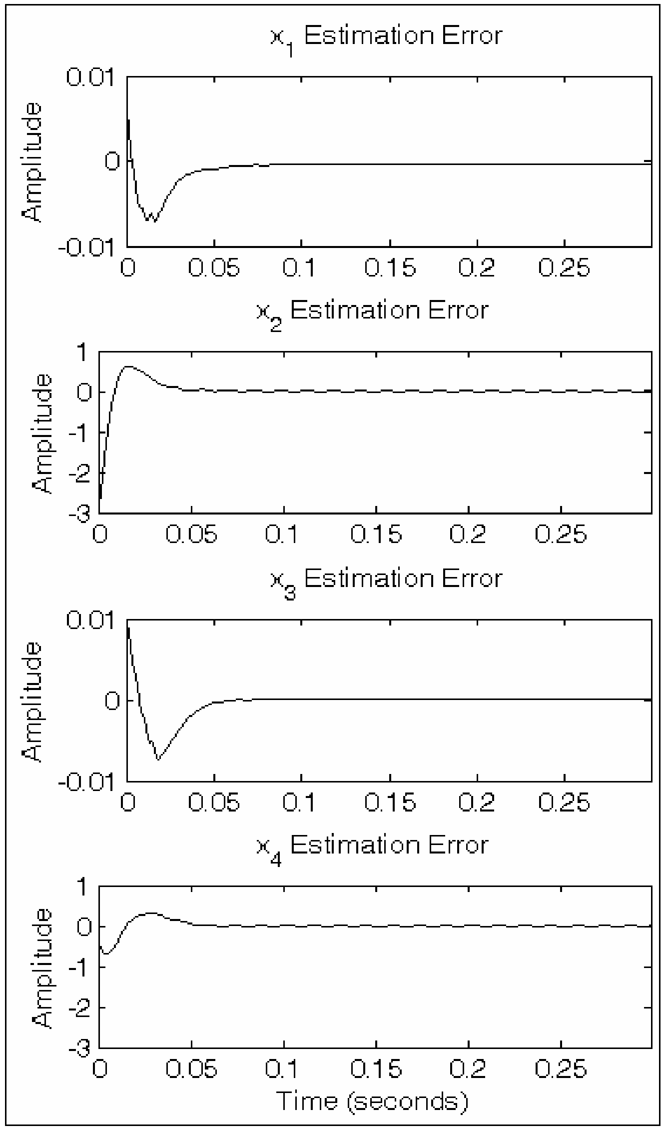

Figure 5-2. Estimation Error – Gyroscope in Motion

CHAPTER 6

NON-PARAMETRIC RECURSIVE LEAST SQUARES SPECTRUM ESTIMATION

This chapter develops recursive least squares based non-parametric spectrum estimation methods ([43], [44]). The development begins with a discussion on rectangular RLS (RRLS) (FIR adaptive filter with $\beta = 1$) developed in CHAPTER 4, as a resonator i.e. a tone tracker. The transfer function representation of the estimator is followed by generalization of the analysis to multiple sinusoids, which provides a unifying basis of discrete Fourier transform (DFT) and RRLS resonator. In the process we derive recursive DFT, which is a familiar concept ([24], [15]). The link thus established between RRLS and DFT provides logical basis for use of other state-space filters for recursive non-parametric spectrum estimation. Specifically, we use truncated exponential RLS (TERLS) developed in this chapter, SSLMS developed in CHAPTER 2 and SSRLS [35].

The spectral leakage that may occur due to rectangular windowing in RRLS is addressed next by deriving a recursive exponential window based algorithm, called truncated exponential RLS (TERLS). The observations in the middle of the window are given largest weightage, whereas the weightage decays exponentially towards the sides of the window. TERLS exhibits some improvement over RRLS by suppressing sidelobes.

The idea of resonator banks for non-parametric spectrum estimation is well-known ([23], [38], [49]-[51]). We extend this idea by using SSLMS as a resonator.

Moreover, the relation between SSRLS and RRLS, as established in Section 4.7, provides a mathematical rationale for using resonator banks. With SSLMS and SSRLS, it is possible to simultaneously reduce side levels and spectral width of main lobe. This contrasts with standard Fourier transform based methods where increase in spectral resolution does not yield simultaneous reduction in side levels and main lobe width. We use this property of SSLMS/SSRLS to develop higher order resonator (HOR) where a number of closely spaced SSLMS (or SSRLS) resonators contribute to a single frequency bin. Computer simulations demonstrate the developed recursive spectrum estimation methods.

6.1 RRLS Resonator

6.1.1 Batch Processed State Estimate

Consider a sinusoidal signal

$$s[k] = a\cos\left(\omega k\right) + b\sin\left(\omega k\right) \tag{6-1.1}$$

which can be represented by the following state-space matrices

$$\begin{aligned} A &= \begin{bmatrix} \cos(\omega) & \sin(\omega) \\ -\sin(\omega) & \cos(\omega) \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 \end{bmatrix} \end{aligned} \tag{6-1.2}$$

It can be shown that the states and coefficients are related by

$$\boldsymbol{x}[k] = A^k \begin{bmatrix} a \\ b \end{bmatrix} \tag{6-1.3}$$

If it is assumed that frequency ω is known but the amplitude and phase are unknown, then RRLS can be used to estimate the signal $s[k]$ by incorporating model (6-1.2). RRLS hence acts as a resonator, capable of tracking a single tone. In order to discretize the frequency domain, we impose additional constraints on the choice of ω as follows

$$\omega = \frac{2\pi q}{p}, \quad q = \begin{bmatrix} 0,1,\cdots,p/2-1 & p\ even \\ 0,1,\cdots,(p-1)/2 & p\ odd \end{bmatrix} \tag{6-1.4}$$

The condition (6-1.4) simplifies the problem as

$$\Phi = \frac{p}{2} I \tag{6-1.5}$$

where I is the identity matrix and the definition of Φ remains same as in (4-2.1) with $W = I$. In order to derive batch-processed least-squares solution for such case, we use (4-1.7), (4-2.1) and (6-1.5) to get

$$\begin{aligned} \hat{\boldsymbol{x}}[k] &= \Phi^{-1} H^T \boldsymbol{\mathcal{Y}}[k] \\ &= \frac{2}{p} \begin{bmatrix} CA^{-p+1} \\ CA^{-p+2} \\ \vdots \\ CA^{-2} \\ CA^{-1} \\ C \end{bmatrix}^T \begin{bmatrix} \boldsymbol{y}[k-p+1] \\ \boldsymbol{y}[k-p+2] \\ \vdots \\ \boldsymbol{y}[k-2] \\ \boldsymbol{y}[k-1] \\ \boldsymbol{y}[k] \end{bmatrix} \\ &= \frac{2}{p} \Big\{ (A^{-p+1})^T C^T \boldsymbol{y}[k-p+1] + (A^{-p+2})^T C^T \boldsymbol{y}[k-p+2] + \cdots + \\ &\qquad (A^{-2})^T C^T \boldsymbol{y}[k-2] + (A^{-1})^T C^T \boldsymbol{y}[k-1] + C^T \boldsymbol{y}[k] \Big\} \end{aligned} \tag{6-1.6}$$

Using (6-1.2), we may write third equation in (6-1.6) as

$$\hat{\boldsymbol{x}}[k] = \frac{2}{p}\left\{\begin{bmatrix} \cos\left((p-1)\omega\right) & \sin\left((p-1)\omega\right) \\ -\sin\left((p-1)\omega\right) & \cos\left((p-1)\omega\right) \end{bmatrix}\begin{bmatrix} \boldsymbol{y}[k-p+1] \\ 0 \end{bmatrix} + \right.$$
$$\begin{bmatrix} \cos\left((p-2)\omega\right) & \sin\left((p-2)\omega\right) \\ -\sin\left((p-2)\omega\right) & \cos\left((p-2)\omega\right) \end{bmatrix}\begin{bmatrix} \boldsymbol{y}[k-p+2] \\ 0 \end{bmatrix} + \cdots +$$
$$\begin{bmatrix} \cos\left(2\omega\right) & \sin\left(2\omega\right) \\ -\sin\left(2\omega\right) & \cos\left(2\omega\right) \end{bmatrix}\begin{bmatrix} \boldsymbol{y}[k-2] \\ 0 \end{bmatrix} +$$
$$\left.\begin{bmatrix} \cos\left(\omega\right) & \sin\left(\omega\right) \\ -\sin\left(\omega\right) & \cos\left(\omega\right) \end{bmatrix}\begin{bmatrix} \boldsymbol{y}[k-1] \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{y}[k] \\ 0 \end{bmatrix}\right\} \tag{6-1.7}$$
$$= \frac{2}{p}\begin{bmatrix} \boldsymbol{y}[k-p+1]\cos\left((p-1)\omega\right) + \boldsymbol{y}[k-p+2]\cos\left((p-2)\omega\right) + \cdots + \\ -\boldsymbol{y}[k-p+1]\sin\left((p-1)\omega\right) - \boldsymbol{y}[k-p+2]\sin\left((p-2)\omega\right) - \cdots - \end{bmatrix}$$
$$\begin{bmatrix} \boldsymbol{y}[k-2]\cos\left(2\omega\right) + \boldsymbol{y}[k-1]\cos\left(\omega\right) + \boldsymbol{y}[k] \\ \boldsymbol{y}[k-2]\sin\left(2\omega\right) - \boldsymbol{y}[k-1]\sin\left(\omega\right) - \boldsymbol{y}[k] \end{bmatrix}$$

Second equation in (6-1.7) may be written in compact form as

$$\hat{\boldsymbol{x}}[k] = \frac{2}{p}\begin{bmatrix} \sum_{i=k-p+1}^{k} y[i]\cos\left((k-i)\omega\right) \\ -\sum_{i=k-p+1}^{k} y[i]\sin\left((k-i)\omega\right) \end{bmatrix} \tag{6-1.8}$$

Equation (6-1.8) is the batch processed state estimate for RRLS resonator.

6.1.2 DFT Coefficients

From (6-1.3), we may write

$$\begin{bmatrix} a \\ b \end{bmatrix} = A^{-k}\boldsymbol{x}[k] \tag{6-1.9}$$

Using (6-1.8), coefficients a and b in (6-1.9) may be estimated as

$$
\begin{bmatrix} \hat{a} \\ \\ \hat{b} \end{bmatrix} = \frac{2}{p} A^{-k} \begin{bmatrix} \sum_{i=k-p+1}^{k} y[i]\cos\left((k-i)\omega\right) \\ -\sum_{i=k-p+1}^{k} y[i]\sin\left((k-i)\omega\right) \end{bmatrix} \tag{6-1.10}
$$

An alternate (and equivalent) form of (6-1.10) may be obtained if we write first equation of (6-1.7) as

$$
\begin{aligned}
\hat{\boldsymbol{x}}[k] &= \frac{2}{p} A^{k} \left\{ A^{-k} A^{p-1} \begin{bmatrix} \boldsymbol{y}[k-p+1] \\ 0 \end{bmatrix} + A^{-k} A^{p-2} \begin{bmatrix} \boldsymbol{y}[k-p+2] \\ 0 \end{bmatrix} + \cdots + \right. \\
&\qquad \left. A^{-k} A^{-2} \begin{bmatrix} \boldsymbol{y}[k-2] \\ 0 \end{bmatrix} + A^{-k} A^{-1} \begin{bmatrix} \boldsymbol{y}[k-1] \\ 0 \end{bmatrix} + A^{-k} \begin{bmatrix} \boldsymbol{y}[k] \\ 0 \end{bmatrix} \right\} \\
&= \frac{2}{p} A^{k} \left\{ A^{-(k-p+1)} \begin{bmatrix} \boldsymbol{y}[k-p+1] \\ 0 \end{bmatrix} + A^{-(k-p+2)} \begin{bmatrix} \boldsymbol{y}[k-p+2] \\ 0 \end{bmatrix} + \cdots + \right. \\
&\qquad \left. A^{-(k-2)} \begin{bmatrix} \boldsymbol{y}[k-2] \\ 0 \end{bmatrix} + A^{-(k-1)} \begin{bmatrix} \boldsymbol{y}[k-1] \\ 0 \end{bmatrix} + A^{-k} \begin{bmatrix} \boldsymbol{y}[k] \\ 0 \end{bmatrix} \right\}
\end{aligned} \tag{6-1.11}
$$

Using (6-1.2), we may write second equation in (6-1.11) as

$$
\begin{aligned}
\hat{\boldsymbol{x}}[k] &= \frac{2}{p} A^{k} \left\{ \begin{bmatrix} \cos\left((k-p+1)\omega\right) & -\sin\left((k-p+1)\omega\right) \\ \sin\left((k-p+1)\omega\right) & \cos\left((k-p+1)\omega\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{y}[k-p+1] \\ 0 \end{bmatrix} + \right. \\
&\qquad \begin{bmatrix} \cos\left((k-p+2)\omega\right) & -\sin\left((k-p+2)\omega\right) \\ \sin\left((k-p+2)\omega\right) & \cos\left((k-p+2)\omega\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{y}[k-p+2] \\ 0 \end{bmatrix} + \cdots + \\
&\qquad \begin{bmatrix} \cos\left((k-2)\omega\right) & -\sin\left((k-2)\omega\right) \\ \sin\left((k-2)\omega\right) & \cos\left((k-2)\omega\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{y}[k-2] \\ 0 \end{bmatrix} + \\
&\qquad \begin{bmatrix} \cos\left((k-1)\omega\right) & -\sin\left((k-1)\omega\right) \\ \sin\left((k-1)\omega\right) & \cos\left((k-1)\omega\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{y}[k-1] \\ 0 \end{bmatrix} + \\
&\qquad \left. \begin{bmatrix} \cos\left(k\omega\right) & -\sin\left(k\omega\right) \\ \sin\left(k\omega\right) & \cos\left(k\omega\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{y}[k] \\ 0 \end{bmatrix} \right\} \\
&= \frac{2}{p} A^{k} \left[\begin{matrix} \boldsymbol{y}[k-p+1]\cos\left((k-p+1)\omega\right) + \boldsymbol{y}[k-p+2]\cos\left((k-p+2)\omega\right) + \cdots + \\ \boldsymbol{y}[k-p+1]\sin\left((k-p+1)\omega\right) + \boldsymbol{y}[k-p+2]\sin\left((k-p+2)\omega\right) + \cdots + \end{matrix} \right. \\
&\qquad \left. \begin{matrix} \boldsymbol{y}[k-2]\cos\left((k-2)\omega\right) + \boldsymbol{y}[k-1]\cos\left((k-1)\omega\right) + \boldsymbol{y}[k]\cos\left(k\omega\right) \\ \boldsymbol{y}[k-2]\sin\left((k-2)\omega\right) + \boldsymbol{y}[k-1]\sin\left((k-1)\omega\right) + \boldsymbol{y}[k]\sin\left(k\omega\right) \end{matrix} \right]
\end{aligned} \tag{6-1.12}
$$

Second equation in (6-1.12) may be written in compact form as

$$\hat{\boldsymbol{x}}[k] = \frac{2}{p} A^k \begin{bmatrix} \sum_{i=k-p+1}^{k} \boldsymbol{y}[i]\cos(\omega i) \\ \sum_{i=k-p+1}^{k} \boldsymbol{y}[i]\sin(\omega i) \end{bmatrix} \tag{6-1.13}$$

Using (6-1.13), coefficients a and b in (6-1.9) may be estimated as

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \frac{2}{p} \begin{bmatrix} \sum_{i=k-p+1}^{k} \boldsymbol{y}[i]\cos(\omega i) \\ \sum_{i=k-p+1}^{k} \boldsymbol{y}[i]\sin(\omega i) \end{bmatrix} \tag{6-1.14}$$

With (6-1.4), the estimates $\hat{a}$ and $\hat{b}$ in (6-1.14) can be recognized as DFT coefficients ([15], [48]).

6.1.3 Recursive Solution

Using equations (4-2.19) and (6-1.5), we get the following recursive filter for this particular case

$$\hat{\boldsymbol{x}}[k] = A^{-T}\,\hat{\boldsymbol{x}}[k-1] + \frac{2}{p} C^T \boldsymbol{y}[k] - \frac{2}{p}\left(A^T\right)^{-p} C^T \boldsymbol{y}[k-p] \tag{6-1.15}$$

6.1.4 Transfer Function Representation

The transfer function corresponding to (6-1.15) is given as [9]

$$H(z) = \frac{2}{p} z^{-p+1} \begin{bmatrix} \dfrac{-z\cos(2\pi q) + z^p\left(z - \cos(\omega)\right) + \cos\left(\omega(p-1)\right)}{z^2 - 2z\cos(\omega) + 1} \\ \dfrac{z\sin(2\pi q) - z^p \sin(\omega) - \sin\left(\omega(p-1)\right)}{z^2 - 2z\cos(\omega) + 1} \end{bmatrix} \tag{6-1.16}$$

where (6-1.4) holds. Note that the pole zero cancellation makes (6-1.16) an FIR filter, as was done in (4-4.3). First magnitude versus frequency plot in Figure 6-1 illustrates the performance of RRLS resonator for $q = 7$, $p = 32$ and $T = 1\sec$.

6.2 Equivalence of RRLS Resonator and DFT

In this section we show that RRLS resonator is equivalent to DFT. Some insight into this equivalence was obtained from (6-1.14). Generalizing the results of Section 6.1, consider a signal consisting of N sinusoids plus a DC component as follows

$$s[k] = a_o + \sum_{q=1}^{N} a_q \cos\left(\omega_q k\right) + \sum_{q=1}^{N} b_q \sin\left(\omega_q k\right) \tag{6-2.1}$$

where $\omega_q = \dfrac{2\pi q}{p}$ and the sense of q remains the same as in (6-1.4). The frequencies are multiples of the fundamental $f = 1/p$. Define

$$A_q = \begin{bmatrix} \cos\left(\omega_q\right) & \sin\left(\omega_q\right) \\ -\sin\left(\omega_q\right) & \cos\left(\omega_q\right) \end{bmatrix} \tag{6-2.2}$$

The following system matrices model (6-2.1) in state-space representation.

$$\begin{aligned} A &= diag\{1, A_1, \cdots, A_N\} \\ C &= [1 \quad 1 \quad 0 \quad \cdots \quad 1 \quad 0] \end{aligned} \tag{6-2.3}$$

We get

$$\Phi = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & p/2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & p/2 \end{bmatrix} \tag{6-2.4}$$

by use of the following identities [21]

$$\sum_{k=0}^{p-1}\cos\left(\frac{2\pi ik}{p}\right)\cos\left(\frac{2\pi jk}{p}\right)=\frac{p}{2}\delta_{ij}$$
$$\sum_{k=0}^{p-1}\sin\left(\frac{2\pi ik}{p}\right)\sin\left(\frac{2\pi jk}{p}\right)=\frac{p}{2}\delta_{ij} \qquad (6\text{-}2.5)$$
$$\sum_{k=0}^{p-1}\cos\left(\frac{2\pi ik}{p}\right)\sin\left(\frac{2\pi jk}{p}\right)=0 \quad \textit{for all } i,j$$

where δ_{ij} is Kronecker delta [13]. Using RRLS batch processed solution (equation (4-1.7)) and equation (6-2.4), we may write

$$\hat{\boldsymbol{x}}[k]=\frac{2}{p}H^T\boldsymbol{\mathcal{Y}}[k] \qquad (6\text{-}2.6)$$

With results of this and previous section, we get

$$\begin{bmatrix}\hat{a}_o\\ \hat{a}_q\\ \hat{b}_q\end{bmatrix}=\begin{bmatrix}\dfrac{1}{p}\displaystyle\sum_{i=k-p+1}^{k}y[i]\\ \dfrac{2}{p}\displaystyle\sum_{i=k-p+1}^{k}y[i]\cos\left(\omega_q i\right)\\ \dfrac{2}{p}\displaystyle\sum_{i=k-p+1}^{k}y[i]\sin\left(\omega_q i\right)\end{bmatrix} \qquad (6\text{-}2.7)$$

which is precisely the definition of DFT coefficients ([21], [48]) on a moving horizon $\left[k-p+1,k\right]$. With the definition of system matrices given in (6-2.2) and (6-2.3), we can make use of the RRLS to arrive at recursive DFT [15].

6.3 Truncated Exponential RLS

In DFT, uniform weighting of the data samples in the observation window may lead to frequency leakage [33]. Since DFT and RRLS resonator are equivalent, as was shown in Sections 6.1 and 6.2, spectral leakage effect is found in RRLS as well. To reduce spectral leakage, the batch of time-domain data may be *windowed* [48] before obtaining DFT.

This approach is known to reduce spectral leakage [48]. Various window functions are suitable for obtaining *batch-processed* DFT. In this section, we develop a *recursive* window as weighting pattern for the observations. Specifically, we use *two-sided* exponential weighting of the data samples that modifies RRLS into Truncated Exponential RLS (TERLS).

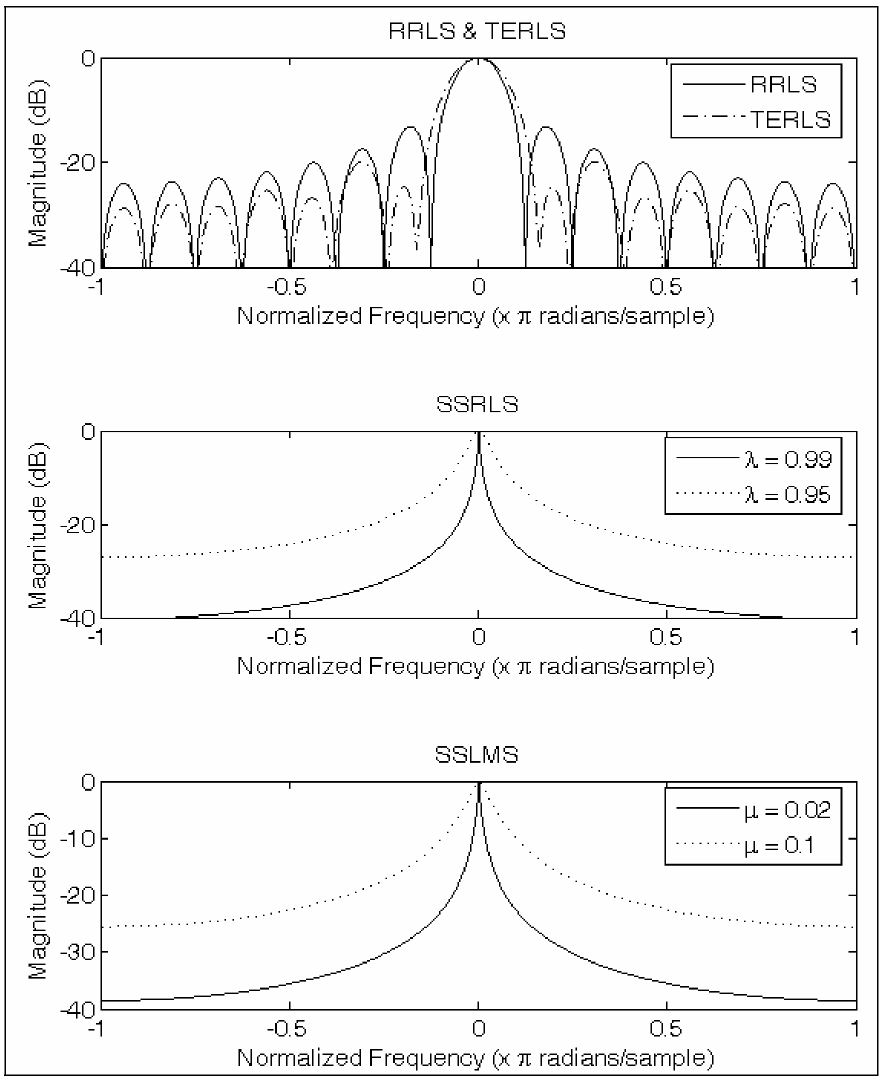

Figure 6-1. Magnitude versus Frequency Response – RRLS, TERLS, SSRLS, SSLMS Resonators

6.3.1 Batch Processed Solution

Consider discrete time state-space model (1-2.1). To simplify the exposition we consider here the case of constant system matrices A and C. The formulations based on time-varying $A[k]$ and $C[k]$ may also be derived easily. Let the observations $\boldsymbol{y}[k]$ start appearing at time $k=1$. Data pre-windowing is assumed i.e. $\boldsymbol{y}[k]=0$ for $k \le 0$. From (1-2.1), we can write p different equations as follows

$$\boldsymbol{\mathcal{Y}}[k]=\begin{bmatrix} \boldsymbol{y}[k-p+1] \\ \boldsymbol{y}[k-p+2] \\ \vdots \\ \boldsymbol{y}[k-2] \\ \boldsymbol{y}[k-1] \\ \boldsymbol{y}[k] \end{bmatrix}=\begin{bmatrix} CA^{-p+1} \\ CA^{-p+2} \\ \vdots \\ CA^{-2} \\ CA^{-1} \\ C \end{bmatrix}\boldsymbol{x}[k]+\boldsymbol{\mathcal{V}}[k] \tag{6-3.1}$$

where $p \ge l$ and observation noise vector is given by

$$\boldsymbol{\mathcal{V}}[k]=\begin{bmatrix} \boldsymbol{v}^T[k-p+1] & \boldsymbol{v}^T[k-p+2] & \cdots & \boldsymbol{v}^T[k-1] & \boldsymbol{v}^T[k] \end{bmatrix}^T.$$

We may write (6-3.1) as

$$\boldsymbol{\mathcal{Y}}[k]=H\boldsymbol{x}[k]+\boldsymbol{\mathcal{V}}[k], \tag{6-3.2}$$

where H is defined as

$$H=\begin{bmatrix} \left(A^{-p+1}\right)^T C^T & \left(A^{-p+2}\right)^T C^T & \dots & A^{-T}C^T & C^T \end{bmatrix}^T \tag{6-3.3}$$

We introduce a weighting factor such that the observation(s) in the center of the window is (are) given largest weight, with p chosen to be an odd (even) positive integer. The weighting of the data samples decays exponentially and symmetrically on both sides as we move away from the center of the observation window. Defining $0<\chi<1$ as the weighting factor, the corresponding weighting matrix (diagonal) for p odd is given as

$$\begin{aligned} W = diag\Big[& \chi^{\frac{p-1}{2}} I_m \quad \chi^{\frac{p-3}{2}} I_m \quad \cdots \quad \chi I_m \quad I_m \\ & \chi I_m \quad \cdots \quad \chi^{\frac{p-3}{2}} I_m \quad \chi^{\frac{p-1}{2}} I_m \Big] \end{aligned} \tag{6-3.4}$$

where $diag[\cdot]$ represents a diagonal matrix, with diagonal entries being the terms inside the brackets of (6-3.4). I_m is identity matrix of dimension $m \times m$. For p even, the weighting matrix is given as

$$\begin{aligned} W = diag\Big[& \chi^{\frac{p}{2}-1} I_m \quad \chi^{\frac{p}{2}-2} I_m \quad \cdots \quad \chi I_m \quad I_m \\ & I_m \quad \chi I_m \quad \cdots \quad \chi^{\frac{p}{2}-2} I_m \quad \chi^{\frac{p}{2}-1} I_m \Big] \end{aligned} \tag{6-3.5}$$

Taking weighting matrix (6-3.4) or (6-3.5) (depending on choice of p) into account, the solution of system (6-3.2) in terms of weighted least squares is given as follows [13]

$$\hat{\boldsymbol{x}}[k] = (H^T W H)^{-1} H^T W \boldsymbol{\mathcal{Y}}[k] \tag{6-3.6}$$

The condition $p \geq l$ along with the observability assumption ensures the invertibility of $H^T W H$.

6.3.2 Recursive Solution

Define following symbols

$$\begin{aligned} \Phi &= H^T W H \\ \zeta[k] &= H^T W \boldsymbol{\mathcal{Y}}[k] \end{aligned} \tag{6-3.7}$$

From (6-3.6) and (6-3.7), we get

$$\Phi\hat{\boldsymbol{x}}[k]=\zeta[k] \quad \text{or} \quad \hat{\boldsymbol{x}}[k]=\Phi^{-1}\zeta[k] \tag{6-3.8}$$

In equation (6-3.8), Φ^{-1} can be computed offline.

6.3.2.1 Recursive Computation of $\zeta[k]$

To facilitate recursive computation of $\zeta[k]$ we express it as a sum of two matrices of compatible dimensions

$$\zeta[k]=\zeta_1[k]+\zeta_2[k] \tag{6-3.9}$$

Both $\zeta_1[k]$ and $\zeta_2[k]$ may be computed recursively. There is slight difference in the recursive formulations of $\zeta_1[k]$ and $\zeta_2[k]$ depending on whether *p* is even or odd, as will be shown subsequently.

6.3.2.1.1 Even p

For the case when *p is* even, $\zeta_1[k]$ is given by

$$\begin{aligned}\zeta_1[k]=\Bigg[&\chi^{\frac{p}{2}-1}\left(A^T\right)^{-p+1}C^T\boldsymbol{y}[k-p+1]+\\&\chi^{\frac{p}{2}-2}\left(A^T\right)^{-p+2}C^T\boldsymbol{y}[k-p+2]+\cdots\\&+\chi(A^T)^{-\frac{p}{2}-1}C^T\boldsymbol{y}[k-\frac{p}{2}-1]+(A^T)^{-\frac{p}{2}}C^T\boldsymbol{y}[k-\frac{p}{2}]\Bigg]\end{aligned} \tag{6-3.10}$$

and $\zeta_2[k]$ is given by

$$\zeta_2[k] = \Bigg[\left(A^T\right)^{-\frac{p}{2}+1} C^T \boldsymbol{y}[k-\frac{p}{2}+1] +$$

$$\chi \left(A^T\right)^{-\frac{p}{2}+2} C^T \boldsymbol{y}[k-\frac{p}{2}+2] + \cdots \tag{6-3.11}$$

$$+\chi^{\frac{p}{2}-2} A^{-T} C^T \boldsymbol{y}[k-1] + \chi^{\frac{p}{2}-1} C^T \boldsymbol{y}[k] \Bigg]$$

which also gives us

$$\zeta_1[k-1] = \Bigg[\chi^{\frac{p}{2}-1} \left(A^T\right)^{-p+1} C^T \boldsymbol{y}[k-p] +$$

$$\chi^{\frac{p}{2}-2} \left(A^T\right)^{-p+2} C^T \boldsymbol{y}[k-p+1] + \cdots \tag{6-3.12}$$

$$+\chi (A^T)^{-\frac{p}{2}-1} C^T \boldsymbol{y}[k-\frac{p}{2}-2] + (A^T)^{-\frac{p}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}-1] \Bigg]$$

and

$$\zeta_2[k-1] = \Bigg[\left(A^T\right)^{-\frac{p}{2}+1} C^T \boldsymbol{y}[k-\frac{p}{2}] +$$

$$\chi \left(A^T\right)^{-\frac{p}{2}+2} C^T \boldsymbol{y}[k-\frac{p}{2}+1] + \cdots \tag{6-3.13}$$

$$+\chi^{\frac{p}{2}-2} A^{-T} C^T \boldsymbol{y}[k-2] + \chi^{\frac{p}{2}-1} C^T \boldsymbol{y}[k-1] \Bigg]$$

From (6-3.10) and (6-3.12), we get following recursion

$$\begin{aligned} \zeta_1[k] &= \chi A^{-T} \zeta_1[k-1] + (A^T)^{-\frac{p}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}] \\ &\quad - \chi^{\frac{p}{2}} \left(A^T\right)^{-p} C^T \boldsymbol{y}[k-p] \end{aligned} \tag{6-3.14}$$

and from (6-3.11) and (6-3.13), we get

$$\zeta_2[k] = \chi^{-1} A^{-T} \zeta_2[k-1] + \chi^{\frac{p}{2}-1} C^T \boldsymbol{y}[k] - \chi \left(A^T\right)^{-\frac{p}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}] \tag{6-3.15}$$

Equations (6-3.8), (6-3.9), (6-3.14) and (6-3.15) constitute TERLS for the case when p is even.

6.3.2.1.2 Odd p

For the case when p is odd, $\zeta_1[k]$ is given by

$$\begin{aligned}\zeta_1[k] = \Bigg[& \chi^{\frac{p-1}{2}} \left(A^T\right)^{-p+1} C^T \boldsymbol{y}[k-p+1] + \\ & \chi^{\frac{p-3}{2}} \left(A^T\right)^{-p+2} C^T \boldsymbol{y}[k-p+2] + \cdots \\ & + \chi (A^T)^{-\frac{p}{2}-\frac{1}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}-\frac{1}{2}] + \\ & (A^T)^{-\frac{p}{2}+\frac{1}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}+\frac{1}{2}] \Bigg]\end{aligned} \tag{6-3.16}$$

and $\zeta_2[k]$ is given by

$$\begin{aligned}\zeta_2[k] = \Bigg[& \chi \left(A^T\right)^{-\frac{p}{2}+\frac{3}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}+\frac{3}{2}] + \\ & \chi^2 \left(A^T\right)^{-\frac{p}{2}+\frac{5}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}+\frac{5}{2}] + \cdots \\ & + \chi^{\frac{p-3}{2}} A^{-T} C^T \boldsymbol{y}[k-1] + \chi^{\frac{p-1}{2}} C^T \boldsymbol{y}[k] \Bigg]\end{aligned} \tag{6-3.17}$$

which also gives us

$$\zeta_1[k-1] = \Bigg[\chi^{\frac{p-1}{2}} \left(A^T\right)^{-p+1} C^T \boldsymbol{y}[k-p] + \chi^{\frac{p-3}{2}} \left(A^T\right)^{-p+2} C^T \boldsymbol{y}[k-p+1] + \cdots + \chi (A^T)^{-\frac{p}{2}-\frac{1}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}-\frac{3}{2}] + (A^T)^{-\frac{p}{2}+\frac{1}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}-\frac{1}{2}] \Bigg] \tag{6-3.18}$$

and

$$\zeta_2[k-1] = \Bigg[\chi \left(A^T\right)^{-\frac{p}{2}+\frac{3}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}+\frac{1}{2}] + \chi^2 \left(A^T\right)^{-\frac{p}{2}+\frac{5}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}+\frac{3}{2}] + \cdots + \chi^{\frac{p-3}{2}} A^{-T} C^T \boldsymbol{y}[k-2] + \chi^{\frac{p-1}{2}} C^T \boldsymbol{y}[k-1] \Bigg] \tag{6-3.19}$$

From (6-3.16) and (6-3.18), we get following recursion

$$\zeta_1[k] = \chi A^{-T} \zeta_1[k-1] + (A^T)^{-\frac{p}{2}+\frac{1}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}+\frac{1}{2}] - \chi^{\frac{p}{2}+\frac{1}{2}} \left(A^T\right)^{-p} C^T \boldsymbol{y}[k-p] \tag{6-3.20}$$

and from (6-3.17) and (6-3.19), we get

$$\zeta_2[k] = \chi^{-1} (A^T)^{-1} \zeta_2[k-1] + \chi^{\frac{p-1}{2}} C^T \boldsymbol{y}[k] - \left(A^T\right)^{-\frac{p}{2}+\frac{1}{2}} C^T \boldsymbol{y}[k-\frac{p}{2}+\frac{1}{2}] \tag{6-3.21}$$

Equations (6-3.8), (6-3.9), (6-3.20) and (6-3.21) constitute TERLS for the case when p is odd. Stability and initialization discussion in Sections 4.5 and 4.6 applies to TERLS as well, for p even and odd.

6.3.3 Magnitude versus Frequency Response

Incorporating system matrices (6-1.2) into its state-space formulation, TERLS acts as a resonator capable of tracking a single tone. First magnitude versus frequency plot (dotted) in Figure 6-1 illustrates the performance of TERLS resonator for $\chi = 0.93$, $q = 7$, $p = 32$ and $T = 1\sec$. Some improvement, as compared to RRLS resonator, in side-lobe suppression is noticeable at the expense of slight increase in main lobe width. This improvement is attributed to non-rectangular windowing of observations used in TERLS.

6.4 SSLMS and SSRLS Based Resonators

In this section we discuss how other non-rectangular windowing recursive algorithms, SSLMS and SSRLS, can give a designer freedom to choose the desired main lobe width and side levels.

6.4.1 SSLMS Resonator

6.4.1.1 Transfer Function Representation

Since state-space LMS (SSLMS) developed in CHAPTER 2 has the ability to track signals, it may be used as a resonator in recursive spectrum estimation. The transfer function for SSLMS resonator is given as

$$H(z) = \begin{bmatrix} \dfrac{z\mu\left(z-\cos(\omega)\right)}{z^2 + z(\mu-2)\cos(\omega)+1-\mu} \\ \dfrac{-z\mu\sin(\omega)}{z^2 + z(\mu-2)\cos(\omega)+1-\mu} \end{bmatrix} \tag{6-4.1}$$

where μ is the step-size parameter.

6.4.1.2 Bandwidth as a Function of Step-Size

The SSLMS estimate for the sinusoidal case is a narrow band pass filter with center frequency ω. Its bandwidth depends on the step-size in the sense that increasing the step-size increases the bandwidth. SSLMS offers good side lobe suppression and narrower bandwidth for lower values of step-size parameter. Third plot in Figure 6-1 illustrates the performance of SSLMS. The effect of step-size on bandwidth of the filter is evident in the plot.

6.4.2 SSRLS Resonator

6.4.2.1 Transfer Function Representation

The concept of recursive spectrum estimation using the standard SSRLS (with exponential forgetting) was introduced in [38]. The corresponding transfer function for SSRLS resonator is given as [9]

$$H(z)=\begin{bmatrix} \dfrac{(1-\lambda)z\left(z(1+\lambda)-2\lambda\cos(\omega)\right)}{z^2-2z\lambda\cos(\omega)+\lambda^2} \\ \dfrac{(1-\lambda)z\left(z(1-\lambda)\cos(\omega)+\lambda\cos(2\omega)-1\right)\operatorname{cosec}(\omega)}{z^2-2z\lambda\cos(\omega)+\lambda^2} \end{bmatrix} \tag{6-4.2}$$

where λ is the forgetting factor and the filter memory length is $1/(1-\lambda)$ [35].

6.4.2.2 Bandwidth as a Function of Filter Memory

The SSRLS estimate for the sinusoidal case is a narrow band pass filter with center frequency ω. Its bandwidth depends on the memory in the sense that increasing the memory decreases the bandwidth. To keep the memory length the same for SSRLS and

RRLS, we take $p = 1/(1-\lambda)$. Second plot in Figure 6-1 illustrates the performance of SSRLS. The effect of λ on the bandwidth of the filter, that was studied in [38], is also evident in the plot. Interestingly SSRLS offers better side lobe suppression and narrower bandwidth, as compared to RRLS and TERLS.

6.4.3 Remarks

The performance tuning parameters available with SSRLS and SSLMS help a designer in choosing the desired spectral width of the main lobe and the side levels. As shown in second and third plots of Figure 6-1, respectively, higher value of forgetting factor for SSRLS and lower value of step-size parameter for SSLMS cause main lobe to shrink and side levels to reduce. This is achieved at no additional computational cost. On the other hand, with standard Fourier transform based methods the increase in spectral resolution does not yield simultaneous reduction in side levels.

6.5 Higher Order Resonator (HOR)

We have seen in Section 6.4 that, using SSLMS and SSRLS resonators, it is possible to simultaneously reduce side levels and spectral width of main lobe. We use this property to develop higher order resonator (HOR) in which a number of closely spaced SSLMS (or SSRLS) based resonators contribute to a single frequency bin. With right choice of parameters, HOR achieves almost flat top (rectangular) frequency bin with high side-level suppression, thus minimizing the *picket fence effect* [33].

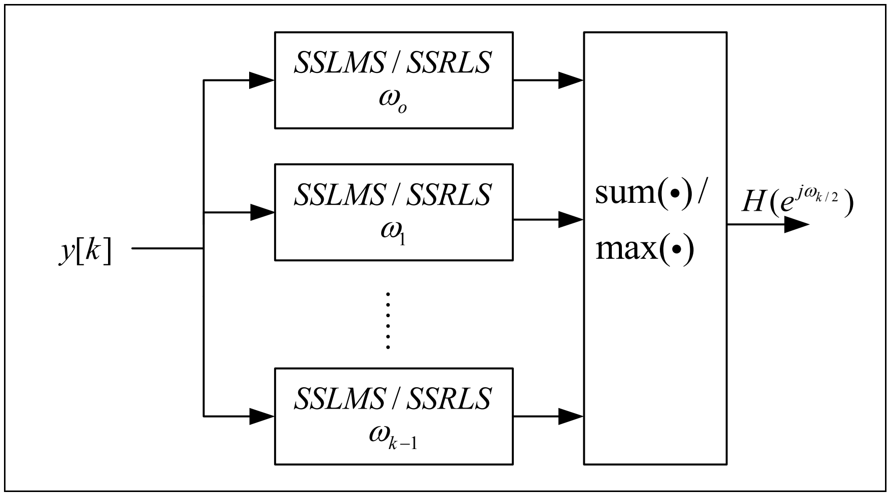

Figure 6-2. Construction of Higher Order Resonator (HOR)

6.5.1 HOR Construction

6.5.1.1 The Concept

The idea is to construct a frequency bin comprising a number of closely spaced individual SSLMS/SSRLS resonators. Each constituent resonator is tuned to a different frequency. The response of a HOR frequency bin is based on either *summing* the outputs of the individual resonators, or choosing the *maximum* of them. Preference for a particular choice is based on the application.

6.5.1.2 HOR with Summing Option

In case multiple signal frequencies lie within a single HOR bin, summing option is preferred to capture contribution of each component. This is illustrated in Figure 6-2. p number of such frequency bins cover the complete discrete spectrum, where the sense of p remains the same as in (6-1.4). First plot in Figure 6-3 shows magnitude versus frequency response for a HOR with summing option, based on six SSLMS resonators.

Decreasing value of μ results in increased pass-band ripple with reduced side levels. Second plot in Figure 6-3 shows HOR performance for fixed μ and variable k, the number of individual SSRLS resonators. Increasing value of k results in decreased pass-band ripple. Response of an ideal frequency bin is also shown. SSRLS based HOR may also be built in a similar way.

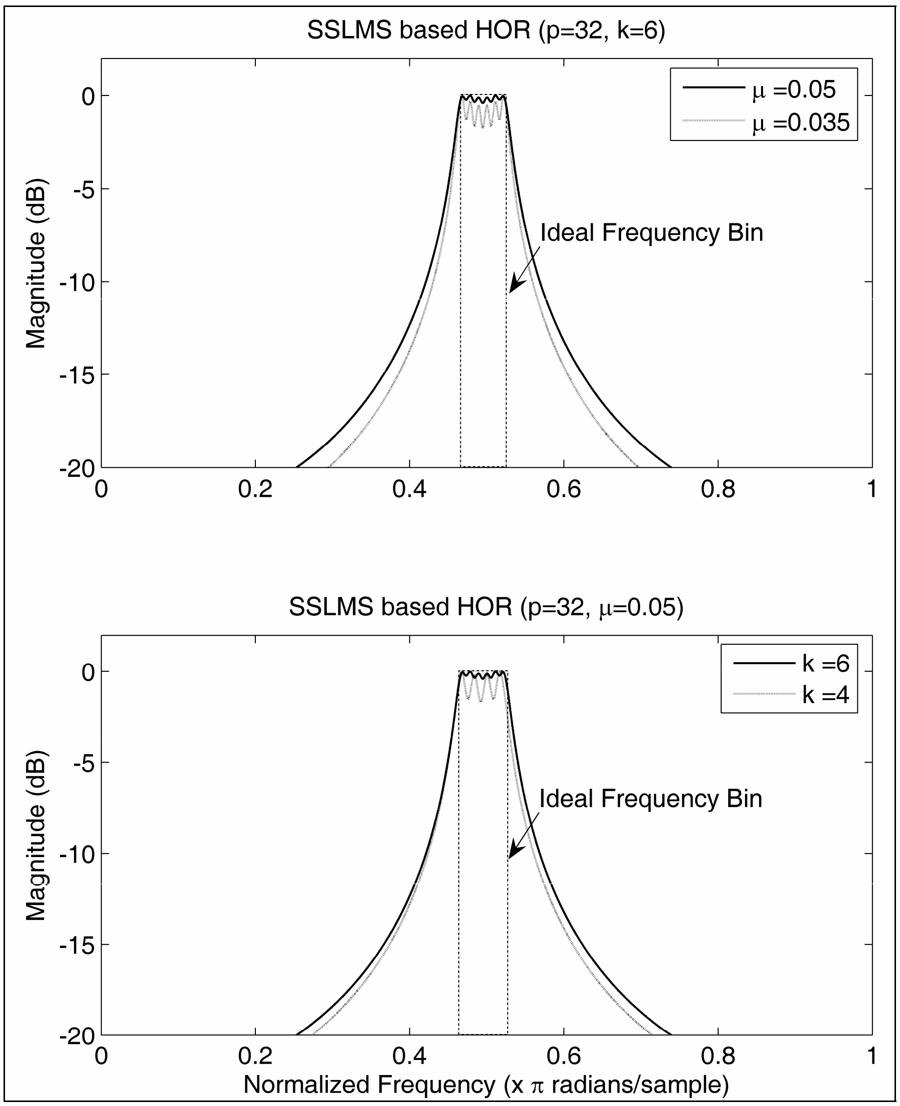

Figure 6-3. HOR with *Summing* Option – Magnitude versus Frequency Response

6.5.1.3 HOR with Maximum Option

For the case of spectral estimation of a single narrow band signal, HOR may be based on the *maximum* option. This approach yields reduced main lobe width and spectral side levels as compared to the summing option. As illustrated in Figure 6-4, six SSLMS resonators are shown to constitute an HOR. Based on signal frequency, only one SSLMS resonator is selected at a time (shown with bold plot). The other resonators are shown with thin lines.

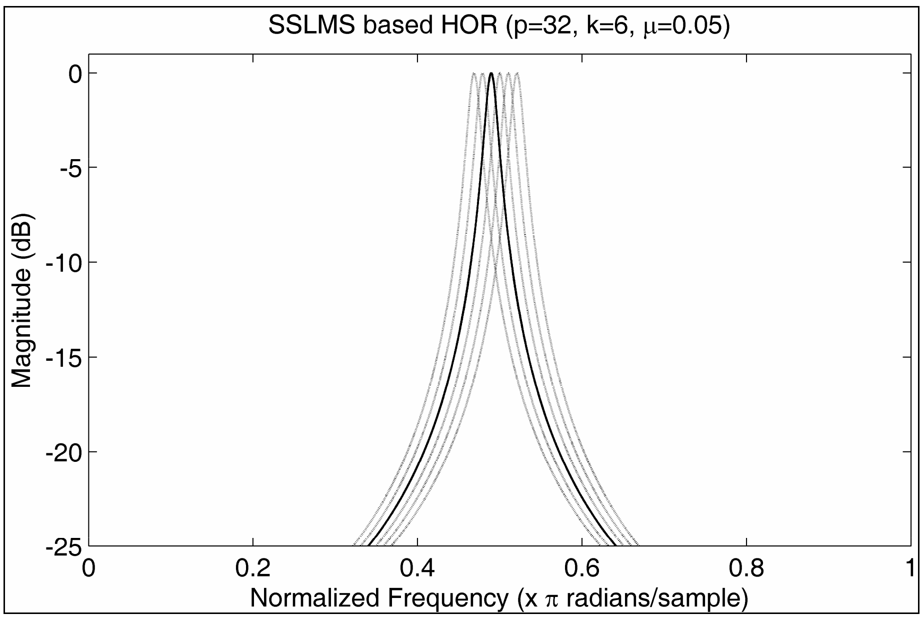

Figure 6-4. HOR with *Maximum* Option – Magnitude versus Frequency Response

6.5.2 Remarks

As shown in Figure 6-1, the side lobes in RRLS and TERLS may cause frequency leakage. On the other hand, a signal component *miss* may result due to narrow main lobe width of SSLMS and SSRLS. A comparison of Figure 6-3 with Figure 6-1 shows that the response of HOR is closer to the ideal frequency bin. Due to this reason HOR better avoids frequency leakage and chances of signal component miss.

6.5.3 HOR Parameters

The amount of pass band ripples r in HOR (with summing option) is dependent on λ, total number of bins p and number of individual SSRLS resonators k in a HOR frequency bin. This dependence is depicted by following proportionality

$$r \propto \frac{\lambda}{pk}$$

Using following formula, a suitable value of λ may be obtained by specifying values of r (dB), k and p

$$\lambda = \frac{\theta_1 \theta_4 p}{1 + \theta_1 p + \theta_2 k + \theta_3 10^{r/20}} \tag{6-5.1}$$

where $\theta_1 = 8.18647$, $\theta_2 = -0.134125$, $\theta_3 = -0.017176$ and $\theta_4 = 0.98965$. Equation (6-5.1) was obtained empirically using nonlinear least squares approach [20] by specifying a nominal model $\theta_1 \theta_4 p / (1 + \theta_1 p + \theta_2 k + \theta_3 10^{r/20})$. A training set comprising 200 different combinations of r (dB), k, p and λ was used to obtain the estimates of parameters $\theta_1, \theta_2, \theta_3$ and θ_4.

6.6 Example (Spectrum Estimation of Non-Orthogonal Tone)

6.6.1.1 Simulation Setup

In this section, we demonstrate the recursive spectrum estimation performance of RRLS, TERLS, SSRLS, SSLMS and HOR (with summing option) arranged in resonator banks configuration ([49]-[51], [23]). We consider a single tone. With following parameters

$$
\begin{aligned}
p &= 32 \\
q &= 7.5 \\
T &= 1 \text{ sec} \\
\chi &= 0.97 \\
\lambda &= 0.99, 0.9 \\
\mu &= 0.02, 0.15 \\
x[0] &= \begin{bmatrix} 1 & 0 \end{bmatrix}^T
\end{aligned}
\tag{6-6.1}
$$

RRLS, TERLS, SSRLS, SSLMS and HOR recursively estimated the frequency spectrum of an input sinusoid with peak amplitude = 1.5. Observation noise is taken as a zero mean white sequence of variance 0.01. We observe the signal in discrete domain after sampling with sampling time T. The frequency of input sinusoid does not exactly match any frequency bin, rather it lies between 7th and 8th frequency bins on the spectrum containing total p frequency bins, where sense of p and q remains same as in equation (6-1.4).

The simulation results as illustrated in Figure 6-5 demonstrate the performance of the recursive algorithms. First plot in top-left corner shows performance of RRLS and TERLS. Second plot in top-right corner reflects performance of SSRLS with forgetting factor λ, where third plot in bottom-left corner shows spectrum estimated by SSLMS with step-size parameter μ. In bottom-right corner, performance of HOR is shown.

6.6.1.2 Remarks

The effect of spectral leakage is evident in all the plots of Figure 6-5; the minimum being for HOR. Parameter χ is used to tune performance of TERLS, which is shown to perform slightly better than RRLS. Although parameters λ in SSRLS and μ in SSLMS may be tuned to reduce spectral leakage effect, attempting to make SSRLS and SSLMS very narrow band filters may obscure the non-orthogonal input signal altogether, as obvious in Figure 6-4. Due to its almost flat passband HOR captures the signal well, while minimizing the chances of spectral leakage.

The example discussed above considers evenly spaced frequency bins only. In situations where *a priori* information is available about concentration of spectral components of a signal to a specific band, it is possible for a designer to choose unevenly spaced frequency bins using the proposed methods. For example, one could choose closely spaced bins in the middle of spectrum and thinly populated ones towards sides.

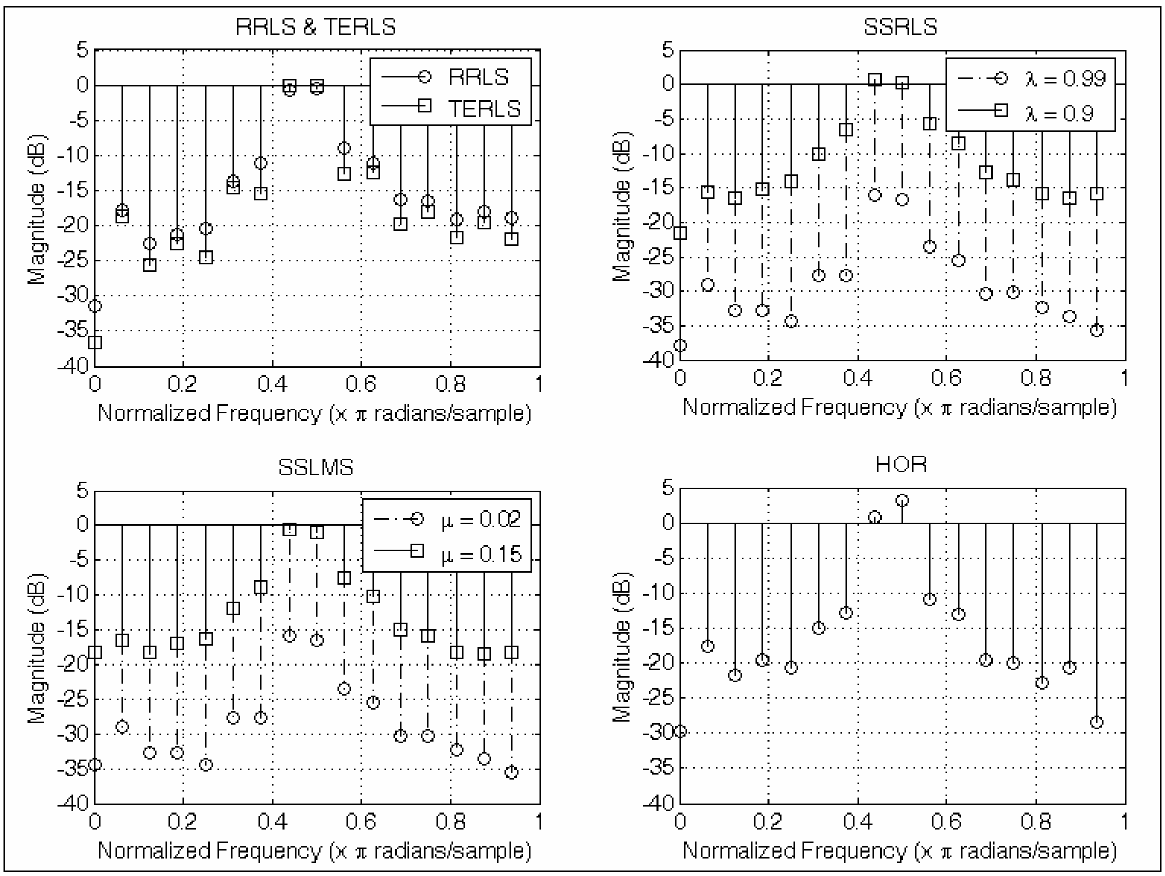

Figure 6-5. Estimated Spectrum – Tone at Non-Orthogonal Frequency

CHAPTER 7

CONCLUSIONS AND FUTURE SUGGESTIONS

7.1 Conclusions

The emphasis of this thesis has been on the development of new variants of adaptive filters. Examples demonstrating the performance of these algorithms are kept at a minimum whereas larger part of the work deals with algorithm formulations, and their stochastic analyses in case of finite memory filters. A key feature of these algorithms is their ability to operate without prior knowledge of process and observation noise statistics.

The development of state-space LMS (SSLMS) is a useful addition to adaptive filtering tools. SSLMS is a generalization of standard LMS. SSLMS exhibits superior tracking performance due to its ability to incorporate model specific information into its state-space formulation. The concept of adaptive memory enhances the capability of SSLMS when it comes to dealing with uncertain time-varying systems. Stochastic gradient approach that has been used to adapt step-size parameter successfully achieves the desired results. Some applications presented give insight into the potential usefulness of the new algorithm. A detailed account of computational complexity presented is valuable for a designer.

FIR adaptive filter, which is built around state-space framework of an unforced system, is especially suitable for use in the situations where LMS and RLS may fail or perform poorly. The filter is inherently stable by virtue of its FIR 'observations to state estimate' mapping. The transfer function of the filter, which consists of *zeros* only, gives insight into its stability properties. The derivations for time-varying case are easily extended to the time-invariant case. We have also formulated the alternate forms of the filter i.e. current estimator and predictor form, which are useful for specific applications. Under a given set of conditions, the FIR adaptive filter and SSRLS (exponential forgetting of past observations) [35] are shown to be equivalent. The convergence analysis gives insight into stochastic behavior of the filter.

Inspired by the property of inherent stability of FIR adaptive filter, we have developed receding horizon state observer for discrete linear forced time-varying systems. The new development augments the repertoire of existing state-observation techniques. The inherent stability of the observer is attributed to finite horizon of system inputs and outputs. Both batch processed and recursive solutions are derived. The observer does not require *a priori* knowledge of second order statistics of process and observation noises. The first and second order convergence analyses for time-varying case give insight into its convergence properties. The performance of the observer is demonstrated using an example considering regulation problem of a gyroscope, which is a time-varying system.

Based on recursive least squares methods, we have developed a set of tools for non-parametric recursive spectrum estimation. Rectangular RLS (FIR adaptive filter with uniform weighting of observations) resonator provides unification of Fourier analysis/spectrum estimation and adaptive filters. To reduce spectral leakage found with RRLS resonator, its *windowed* variant 'truncated exponential RLS (TERLS)' is developed. Furthermore, using state-space RLS (SSRLS) and SSLMS resonators, it is possible to simultaneously reduce spectral main lobe width and side levels. This interesting property of SSRLS and SSLMS forms the basis of development of higher order resonator (HOR). A number of SSRLS/SSLMS resonators contribute to a single HOR. With right choice of parameters, HOR gives response that is close to the ideal frequency bin. HOR reduces spectral leakage and picket fence effect, thus achieving better spectral estimates.

7.2 Future Suggestions

The developments in this thesis give rise to a number of new ideas, thus setting future directions. The usefulness of SSLMS and its variants developed in this thesis, as shown with some examples, invites exploring its stability and convergence properties. It is suggested that stability analysis of SSLMS be carried out to place bounds on the choice of step-size parameter to achieve stable performance in a given application. Furthermore, convergence analysis be carried out to obtain a deeper insight into the potentials of the filter. This will aid a designer in making the best use of the filter in advanced applications and analyses. The next suggestion is to analyze the robustness of SSLMS under model uncertainty using H_∞ methods [13].

The structure of receding horizon state observer, which is built around a linear state-space framework, may be extended to address nonlinear state-observation problem. This effort is expected to be a formidable task. Working with receding horizon state observer or FIR adaptive filter, the most suitable value of weighting factor β is not known *a priori.* The suggestion is to make the weighting factor β function of time *k*; i.e. *adaptive weighting,* that minimizes certain cost function. This exercise is expected to be similar to that carried out for SSLMS to achieve its adaptive memory variant (see

CHAPTER 3). Furthermore, it is also suggested that the computational effort of receding horizon state observer and FIR adaptive filter be worked out.

The tools presented for non-parametric recursive spectrum estimation may be supplemented by developing other types of recursive windows, in addition to exponential weighting developed in this thesis for truncated exponential RLS (TERLS). An important variation on the theme of spectrum estimation is *parametric* spectrum estimation. It is suggested that the presented *non-parametric* methods be extended to address parametric spectrum estimation problem. Moreover, utilization of orthogonal frequencies to achieve computationally efficient results may be taken up in the future.

REFERENCES

[1] Yaakov Bar-Shalom, Xiao-Rong Li and Thiagalingam Kirubarajan, *Estimation with applications to tracking and navigation*, John Wiley & Sons, 2001.

[2] A. Benveniste, "Design of adaptive algorithms for the tracking of time-varying systems", *International Journal of Adaptive Control and Signal Processing*, Vol. 1, pp. 3-29, 1987.

[3] Robert G. Brown, PYC Hwang, *Introduction to random signals and applied Kalman filtering*, 2nd edition, John Wiley & Sons, 1997.

[4] Robert L. Carroll and David P. Lindorff, "An adaptive observer for single-input single-output linear systems", *IEEE Trans. on Automat. Contr.*, Vol. AC-18, No. 5, pp. 428-435, Oct. 1973.

[5] B. Chun, B. Kim and Y. H. Lee, "Generalization of exponentially weighted RLS algorithm based on a state-space model", Vol. V, *IEEE International Symposium on Circuits and System,* pp. 198-201, 1998.

[6] J. M. Cioffi and Thomas Kailath, "Fast recursive least squares transversal filters for adaptive filtering", *IEEE Trans. Acoust., Speech, Signal Processing,* Vol. ASSP-32, pp. 304-337, 1984.

[7] C. A. Desoer and Y. T. Wang, "Linear time-invariant robust servomechanism problem: a self-contained exposition", *Control and Dynamic Systems*, C. T. Leondes, ed., Vol. 16, pp. 81-129, 1980.

[8] E. Eleftheriou and D. D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP- 34, pp. 1097-1110, 1986.

[9] E. Ewada, "Comparison of RLS, LMS and sign algorithms for tracking randomly time-varying channels", *IEEE Trans. on Signal Processing*, Vol. 42, pp. 2937-2944, 1994.

[10] Gene F. Franklin, J. David Powell and Michael L. Workman, *Digital Control of Dynamic Systems*, 3rd edition, Addison-Wesley, 1998.

[11] Babak Hassibi, Ali H. Sayed and Thomas Kailath, "H_∞ *optimality of the LMS algorithm*", *IEEE Trans. Signal Process.*, Vol. 44, pp. 267-280, 1996.

[12] Simon Haykin, A. H. Sayed, J. Zeidler, P. Yee, P. Wei , “Adaptive tracking of linear time-variant systems by extended RLS algorithms”, *IEEE Trans. on Signal Processing*, Vol. 45, No. 6, pp. 1118-1128, May 1997.

[13] Simon Haykin, *Adaptive filter theory*, 4th edition, Prentice Hall, 2001.

[14] N. Hori, P.N. Nikiforuk, K. Kanai and S. Uchikado, “On the improvement of an adaptive observer for multi-output systems”, *IEE Proceedings on Control Theory and Applications*, Vol. 135, pp. 67-71, Jan. 1988.

[15] Gene H. Hostetter, "Recursive discrete Fourier transform", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-28, no. 2, pp. 182-190, Apr. 1990.

[16] Kunsoo Huh and Byungkil Kwak, “Discrete-time well-conditioned observer design for two-output systems”, *Proceedings of the IEEE International Conference on Control Applications*, pp. 441-444, 1997.

[17] Alberto Isidori, *Nonlinear control systems*, 3rd edition, Springer-Verlag, 2001.

[18] Andrew H. Jazwinski, “Limited memory optimal filtering”, *IEEE Trans. on Automat. Contr.*, Vol. 13, pp. 558-563, 1968.

[19] Thomas Kailath, “A view of three decades of linear filtering theory”, *IEEE Trans. Inf. Theory*, IT-20(2), pp. 146-181, 1974.

[20] Steven M. Kay, *Fundamentals of statistical signal processing: estimation theory*, Prentice Hall, 1993.

[21] Steven M. Kay, *Modern Spectral Estimation: Theory & Applications*, Prentice Hall, 1993.

[22] Hassan K. Khalil, *Nonlinear systems*, 3rd edition, Prentice Hall, 2002.

[23] A. R. Varkonyi-Koczy, G. Simon, L. Sujbret and M. Fek, “A fast filter-bank for adaptive Fourier analysis”, *IEEE Trans. Instrumentation and Measurement*, Vol. 47, No. 5, pp. 1124-1128, Oct. 1998.

[24] A. R. Varkonyi-Koczy “A recursive fast Fourier transformation algorithm”, *IEEE Trans. Circuits Syst.*, Vol. 42, No. 9, pp. 614-616, Sept. 1995.

[25] Gerhard Kreisselmeier, “Adaptive observers with exponential rate of convergence”, *IEEE Trans. on Automat. Contr.*, Vol. 22, pp. 2-8, 1977.

[26] Erwin Kreyszig, *Advanced engineering mathematics*, 8th edition, John-Wiley and Sons, 1983.

[27] Wook H. Kwon, Pyung S. Kim, Soo H. Han, "A receding horizon unbiased FIR (RHUF) filter for discrete-time state-space models", Automatica, Vol. 38, pp. 545-551, 2002.

[28] Wook H. Kwon, Pyung S. Kim, PooGyeon Park, " A receding horizon Kalman FIR filter for discrete time-invariant systems", *IEEE Trans. on Automat. Contr.*, Vol. 44, No. 9, pp. 1787-1791, 1999.

[29] K. V. Ling and K. W. Lim, “Receding horizon recursive state estimation”, *IEEE Trans. on Automat. Contr.*, Vol. 44, No. 9, pp. 1750-1753, Sept. 1999.

[30] David G. Luenberger, “Observing the state of a linear system”, *IEEE Trans. on Mil. Electron.*, Vol. MIL-8, No. 6, pp. 74-80, Apr. 1964.

[31] David G. Luenberger, “Observers for multivariable systems”, *IEEE Trans. on Automat. Contr.*, Vol. AC-11, pp. 190-197, Apr. 1966.

[32] David G. Luenberger, “An introduction to observers”, *IEEE Trans. on Automat. Contr.*, Vol. AC-16, No. 6, pp. 596-602, Dec. 1971.

[33] Richard G. Lyons, *Understanding Digital Signal Processing*, 2nd edition, Pearson Education, 2004.

[34] Mohammad Bilal Malik, “State-space recursive least-squares”, *PhD Dissertation*, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan, 2004.

[35] Mohammad Bilal Malik, “State-space recursive least-squares: part I & II”, *Signal Processing Journal,* Vol. 84, pp. 1709-1728, 2004.

[36] Mohammad B. Malik, Hafsa Qureshi, Rashid A. Bhatti, “Tracking of linear time-varying systems by state-space recursive least squares”, *IEEE International Symposium on Circuits and System,* Vol. III, pp. 305-308, 2004.

[37] Mohammad Bilal Malik and Rashid Bhatti, “Tracking of linear time-varying systems using state-space least mean square”, *IEEE International Symposium on Communications and Information Technologies*, pp. 582-585, 2004.

[38] Mohammad Bilal Malik, Ata-ul-Basit Hassan, Imran Ghazi and Mohammad U. Hakeem, “On spectrum estimation using SSRLS”, *IEEE International Multitopic Conference*, 2004.

[39] Mohammad Bilal Malik, “State-space recursive least squares with adaptive memory”, *Signal Processing Journal,* Vol. 86, pp. 1365-1374, 2006.

[40] Mohammad Bilal Malik and Muhammad Salman, "State-space least mean square with adaptive memory", *IEEE Region 10 Conference,* pp. 1-6, 2005.

[41] Mohammad Bilal Malik and Muhammad Salman, "Comparative tracking performance of SSRLS and SSLMS algorithms for chirped signal recovery", *IEEE International Multi Topic Conference*, pp. 1-6, 2005.

[42] Mohammad Bilal Malik and Muhammad Salman, "Adaptive tracking of a noisy sinusoid/chirp with unknown parameters", *IEEE International Symposium on Industrial Electronics,* Vol. 1, pp. 593-598, 2006.

[43] Mohammad Bilal Malik, Mohammad Umer Hakeem, Imran Ghazi and Ata-ul-Basit Hassan, "Recursive least squares spectrum estimation", *IEEE International Symposium on Industrial Electronics*, Vol. 1, pp. 599-602, 2006.

[44] Mohammad Bilal Malik and Muhammad Salman, "Non-parametric recursive least squares spectrum estimation", *IEEE Trans. on Circuits and System I,* (under revision), 2009.

[45] Mohammad Bilal Malik and Muhammad Salman, "State-space least mean square", *Digital Signal Processing Journal,* Vol. 18, pp. 334-345, 2008.

[46] Horacio J. Marquez, "Observing the state of a linear time-invariant system: a frequency domain approach", *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Vol. 2, pp. 429-431, 2001.

[47] D. Nešić and Andrew R. Teel, "A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models", *IEEE Trans. on Automat. Contr.*, Vol. 49, No. 7, pp. 1103-1122, 2004.

[48] Alan V. Oppenheim, Ronald W. Schafer and John R. Buck, *Discrete-Time Signal Processing*, 2nd edition, Prentice Hall, 1999.

[49] M. Padmanabhan and K. Martin "Resonator-based filter-banks for frequency-domain applications", *IEEE Trans. Circuits Syst.*, vol. 38, pp. 1145-1159, Oct. 1991.

[50] G. Peceli, "Resonator-based digital filters", *IEEE Trans. Circuits Syst.*, vol. 36, pp 156-159, 1989.

[51] G. Peceli, "A common structure for recursive discrete transforms", *IEEE. Trans. Circuits Syst.*, Vol. CAS-33, pp. 1035-1036, Oct. 1986.

[52] Wilson J. Rugh, *Linear system theory*, Prentice Hall, 1996.

[53] Muhammad Salman and Mohammad Bilal Malik, "Adaptive recovery of a noisy chirp: performance of the SSLMS algorithm", *IEEE International Symposium on Signal Processing and its Applications*, Vol. 2, pp. 763-766, 2005.

[54] Muhammad Salman, Mohammad Bilal Malik and Khalid Munawar, “A receding horizon state observer for linear time-varying systems”, *International Journal of Control*, (submitted), 2009.

[55] Ali H. Sayed and Thomas Kailath , “A state-space approach to adaptive RLS filtering”, *IEEE Signal Processing Magazine*, Vol. 11 No. 3, pp. 18-60, 1994.

[56] Ali H. Sayed, Adaptive Filters, John Wiley & Sons, 2008.

[57] B. Shafai, C.T. Pi, S. Nork and S.P. Linder, “Proportional integral adaptive observer for parameter and disturbance estimations”, *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 4694-4699, 2002.

[58] May-Win L. Thein, “A variable structure parallel observer system for robust state estimation of multirate systems with noise”, *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 1303-1308, 2003.

[59] B. Widrow, J.R. Glover Jr., J.M. McCool, J. Caunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, E. Dong Jr., R.C. Goodlin, “Adaptive noise cancelling: principles and applications", *Proceedings IEEE*, Vol. 63, No. 12, pp. 1692-1716, 1975.

APPENDIX A

PhD-5 FORM (DOCTORAL DEFENCE)

Form PhD-5
DOCTORAL PROGRAMME OF STUDY
(Must be type written)

National University of Sciences & Technology, Islamabad

DOCTORAL DEFENCE

We hereby recommend that the student **Muhammad Salman**

Regn No **2004-NUST-TfrPhD-Elec-25** may be accepted for Doctor of Philosophy Degree.

DOCTORAL DEFENCE COMMITTEE

Held on **27 March 2009**

GEC Member 1: **Dr. Ejaz Muhammad** Signature : Ejaz Muhammad

GEC Member 2: **Dr. Khalid Munawar** Signature : Khalid Munawar

GEC Member (External) 3: **Dr. Mohammad Ali Maud** Signature : M. A. Maud.

Supervisor: **Dr. Mohammad Bilal Malik** Signature : BMalik

Co-Supervisor: **Nil** Signature :
(if appointed)

External Evaluator 1: **Dr. Ijaz Mansoor Qureshi** Signature : IM Qureshi
(Local Expert)

External Evaluator 2: **Dr. Ziauddin Ahmad** Signature : See the attached form
(Foreign Expert) **(USA)**

External Evaluator 3: **Dr. Khurram Waheed** Signature : See the attached form
(Foreign Expert) **(USA)**

COUNTERSIGNED

Dated: 23 Apr, 2009

Dean/Principal

Distribution:

- 1 x copy each to Registrar, D(R&D), Dy Controller (Exam) at HQ NUST and HoD, Supervisor, Co-Supervisor (if appointed), in student's dossier and each member of GEC.

Form PhD-5
DOCTORAL PROGRAMME OF STUDY
(Must be type written)

National University of Sciences & Technology, Islamabad

DOCTORAL DEFENCE

We hereby recommend that the student **Muhammad Salman**

Regn No **2004-NUST-TfrPhD-Elec-25** may be accepted for Doctor of Philosophy Degree.

DOCTORAL DEFENCE COMMITTEE

Held on **27 March 2009**

GEC Member 1: **Dr. Ejaz Muhammad**	Signature :
GEC Member 2: **Dr. Khalid Munawar**	Signature :
GEC Member (External) 3: **Dr. Mohammad Ali Maud**	Signature :
Supervisor: **Dr. Mohammad Bilal Malik**	Signature :
Co-Supervisor: **Nil** (if appointed)	Signature :
External Evaluator 1: **Dr. Ijaz Mansoor Qureshi** (Local Expert)	Signature :
External Evaluator 2: **Dr. Ziauddin Ahmad** (Foreign Expert) **(USA)**	Signature :
External Evaluator 3: **Dr. Khurram Waheed** (Foreign Expert) **(USA)**	Signature :

COUNTERSIGNED

Dated: ______________

Dean/Principal

Distribution:

- 1 x copy each to Registrar, D(R&D), Dy Controller (Exam) at HQ NUST and HoD, Supervisor, Co-Supervisor (if appointed), in student's dossier and each member of GEC.

Form PhD-5
DOCTORAL PROGRAMME OF STUDY
(Must be type written)

National University of Sciences & Technology, Islamabad

DOCTORAL DEFENCE

We hereby recommend that the student **Muhammad Salman**

Regn No **2004-NUST-TfrPhD-Elec-25** may be accepted for Doctor of Philosophy Degree.

DOCTORAL DEFENCE COMMITTEE

Held on **27 March 2009**

GEC Member 1: **Dr. Ejaz Muhammad**	Signature : Ejaz Muhammad
GEC Member 2: **Dr. Khalid Munawar**	Signature : Khalid Munawar
GEC Member (External) 3: **Dr. Mohammad Ali Maud**	Signature : M. A. Maud
Supervisor: **Dr. Mohammad Bilal Malik**	Signature : BMalik
Co-Supervisor: **Nil** (if appointed)	Signature :
External Evaluator 1: **Dr. Ijaz Mansoor Qureshi** (Local Expert)	Signature : IMQureshi
External Evaluator 2: **Dr. Ziauddin Ahmad** (Foreign Expert) **(USA)**	Signature :
External Evaluator 3: **Dr. Khurram Waheed** (Foreign Expert) **(USA)**	Signature :

COUNTERSIGNED

Dated: ______________

Dean/Principal

Distribution:

- 1 x copy each to Registrar, D(R&D), Dy Controller (Exam) at HQ NUST and HoD, Supervisor, Co-Supervisor (if appointed), in student's dossier and each member of GEC.

APPENDIX B

PhD-4 FORM (DOCTORAL THESIS WORK)

Form PhD-4
DOCTORAL PROGRAMME
OF STUDY
(Must be type written)

National University of Sciences & Technology, Rawalpindi
DOCTORAL THESIS WORK

We hereby recommend that the dissertation prepared under our supervision

by Muhammad Salman **Regn No: 2004-NUST-TfrPhD-Elec-25**

Entitled: Adaptive Estimation Using State-Space Methods

be accepted as fulfilling in part of Doctor of Philosophy Degree.

THESIS EVALUATION COMMITTEE

GEC Member 1: Dr. Ejaz Muhammad	**Signature :**
GEC Member 2: Dr. Khalid Munawar	**Signature :**
GEC Member (External) 3: Dr. Mohammad Ali Maud	**Signature :**
Supervisor: Dr. Mohammad Bilal Malik	**Signature :**
Co-Supervisor (if appointed): Nil	**Signature :**
External Evaluator 1: Dr. Ziauddin Ahmad **(Foreign Expert) (USA)**	**Signature :** See the attached form
External Evaluator 2: Dr. Khurram Waheed **(Foreign Expert) (USA)**	**Signature :** See the attached form
External Evaluator 3: Dr. Ijaz Mansoor Qureshi	**Signature :**

APPROVED

Dated: 19/2/9

Head of the Department

COUNTERSIGNED

Dated: 22 Feb, 2009

Dean/Commandant/Principal/DG

Distribution:

- **1 x copy each to Registrar, D(R&D), D(E&A) at HQ NUST and HoD, Supervisor, Co-Supervisor (if appointed), in student's dossier and each member of GEC.**

Form PhD-4
DOCTORAL PROGRAMME
OF STUDY
(Must be type written)

National University of Sciences & Technology, Rawalpindi
DOCTORAL THESIS WORK

We hereby recommend that the dissertation prepared under our supervision

by Muhammad Salman **Regn No: 2004-NUST-TfrPhD-Elec-25**

Entitled: Adaptive Estimation Using State-Space Methods

be accepted as fulfilling in part of Doctor of Philosophy Degree.

THESIS EVALUATION COMMITTEE

GEC Member 1: Dr. Ejaz Muhammad **Signature :** Ejaz Muhammad

GEC Member 2: Dr. Khalid Munawar **Signature :** Khalid Munawar

GEC Member (External) 3: Dr. Mohammad Ali Maud **Signature :** M. A. Maud.

Supervisor: Dr. Mohammad Bilal Malik **Signature :** BMalik

Co-Supervisor (if appointed): Nil **Signature :** ______

External Evaluator 1: Dr. Ziauddin Ahmad **Signature :** Z. Ahmad
(Foreign Expert) (USA)

External Evaluator 2: Dr. Khurram Waheed **Signature :** ______
(Foreign Expert) (USA)

External Evaluator 3: Dr. Ijaz Mansoor Qureshi **Signature :** I M Qureshi

APPROVED

______
Dated: ______ **Head of the Department**

COUNTERSIGNED

______
Dated: ______ **Dean/Commandant/Principal/DG**

Distribution:

- **1 x copy each to Registrar, D(R&D), D(E&A) at HQ NUST and HoD, Supervisor, Co-Supervisor (if appointed), in student's dossier and each member of GEC.**

Form PhD-4
DOCTORAL PROGRAMME
OF STUDY
(Must be type written)

National University of Sciences & Technology, Rawalpindi
DOCTORAL THESIS WORK

We hereby recommend that the dissertation prepared under our supervision

by Muhammad Salman **Regn No: 2004-NUST-TfrPhD-Elec-25**

Entitled: Adaptive Estimation Using State-Space Methods

be accepted as fulfilling in part of Doctor of Philosophy Degree.

THESIS EVALUATION COMMITTEE

GEC Member 1: Dr. Ejaz Muhammad	**Signature :**
GEC Member 2: Dr. Khalid Munawar	**Signature :**
GEC Member (External) 3: Dr. Mohammad Ali Maud	**Signature :**
Supervisor: Dr. Mohammad Bilal Malik	**Signature :**
Co-Supervisor (if appointed): Nil	**Signature :**
External Evaluator 1: Dr. Ziauddin Ahmad **(Foreign Expert) (USA)**	**Signature :**
External Evaluator 2: Dr. Khurram Waheed **(Foreign Expert) (USA)**	**Signature :**
External Evaluator 3: Dr. Ijaz Mansoor Qureshi	**Signature :**

APPROVED

Dated: ______________ **Head of the Department**

COUNTERSIGNED

Dated: ______________ **Dean/Commandant/Principal/DG**

Distribution:

- 1 x copy each to Registrar, D(R&D), D(E&A) at HQ NUST and HoD, Supervisor, Co-Supervisor (if appointed), in student's dossier and each member of GEC.