Retrospective Cost Adaptive Unknown Input Observers with Application to State and Driver Estimation in the Ionosphere-Thermosphere

by

Asad A. Ali

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2013

Doctoral Committee:

Professor Dennis S. Bernstein, Chair
Professor Brent Gillespie
Professor Daniel J. Inman
Professor Ilya Vladimir Kolmanovsky

2013

To my parents

Acknowledgements

First and foremost, I would like to thank my advisor Dr. Dennis Bernstein for his guidance and advice. I would also like to thank Professor Aaron Ridley and my committee members: Professors Ilya Kolmanovsky, Daniel Inman, and Brent Gillespie. I would like to especially thank Jesse Hoagg for his valuable insight and guidance. I am also thankful to the professors at Michigan and Buffalo who have inspired me throughout my graduate school years.

I would like to thank my family for their perpetual encouragement and support, especially during times of uncertainty and doubt.

Finally, I would like to thank my friends and mentors, without whom life would have been very difficult indeed: Alexey Morozov, Gerardo Cruz, Amy Wu, Matthew Robertson, Jin Yan, Kenny Mitchell, Marco Ceze, Matthew Holzel, Anthony D'Amato, Khaled Aljanaideh, Ravi Kumar, and others.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The classical Kalman filter is the optimal state estimator for linear systems under white process and sensor noise with zero mean and finite second moments. In addition, the Kalman filter accommodates the presence of a known, deterministic input. In practice, however, the deterministic input may not be known exactly, and this error can be viewed as a component of the process noise. However, this approach may be too conservative and can lead to bias when the unknown input has a nonzero "mean" value. Consequently, a more direct approach is to extend the estimator to include an estimate of the unknown input.

In this work, we consider an unknown input observer based on retrospective cost optimization, where the unknown input is estimated by first minimizing a retrospective cost function, and then updating an adaptive feedback system using recursive least squares. The retrospective cost method is a minimal modeling approach that is applicable to both minimum- and nonminimum-phase systems.

Since the retrospective cost observer relies on recursive least squares to update an adaptive feedback system, a novel sliding window, variable regularization recursive least squares algorithm is developed and investigated. In contrast to classical recursive least squares algorithms, the sliding window recursive least squares algorithm does not lose its ability to adapt, and does not become unstable when the data lose persistency.

Finally, we use the retrospective-cost-based observer to estimate states and input in the ionosphere-thermosphere using real satellite data and the Global Ionosphere-Thermosphere model.

CHAPTER 1

Introduction

State estimation of dynamical systems is a well studied problem in control theory. After the development of the Kalman filter [2–4] in the 1960s and its successful application to various commercial and national projects, a considerable amount of effort has been devoted to developing filters for nonlinear systems with Gaussian output and process noise, such as the extended Kalman Filter [5, 6] and the Unscented Kalman filter [7, 8], as well as nonlinear systems with non-Gaussian output and process noise, such as the Gaussian sum filter [9], moving horizon estimators [10], and particle filters [11].

In many applications, such as fault detection, parameter estimation, and model correction, the objective is to obtain input estimates in addition to state estimates. Recently, considerable attention has been given to developing state estimators that are either insensitive to the lack of knowledge of the deterministic input or attempt to estimate this signal along with the states. These techniques are referred to as unbiased Kalman Filters, unknown input observers, and state estimators with input reconstruction [1, 12–20].

A further challenge in state estimation is the case of uncertain modelling information or the unavailability of explicit dynamical equations that describe the model. Such models occur in space weather applications, where the models are a combination of partial differential equations, empirical models of poorly understood physical

phenomena, and lookup tables that approximate parameters in the model based on initial conditions or the current state of the system. In these cases, a Jacobian of the dynamics cannot be obtained, and thus methods such as the Extended Kalman Filter cannot be applied. For these applications, ensemble-based data-assimilation methods, known as particle filters, are typically used, such as the Unscented Kalman Filter [21], Ensemble Kalman Filters [22, 23], and the Ensemble Adjustment Kalman Filter [24]. In order to apply these methods in the presence of unknown inputs, the state vector must be augmented with the unknown inputs as state variables, and models of the unknown inputs must be provided.

To illustrate these challenges, we begin with the classical Luenberger observer. Consider the discrete-time linear system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k), \end{aligned}$$

where $x(k) \in \mathbb{R}^n$ is the unknown state, $y(k) \in \mathbb{R}^q$ is the measured output, $u(k) \in \mathbb{R}^p$ is the measured input, and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, and $C \in \mathbb{R}^{q \times n}$ are system matrices, which are assumed to be known. Furthermore, we assume that the pair (A, C) is observable. The Luenberger observer is given by [25]

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + L(y(k) - \hat{y}(k)), \\ \hat{y}(k) &= C\hat{x}(k), \end{aligned}$$

where $\hat{x}(k) \in \mathbb{R}^n$ is the estimated state, $\hat{y}(k) \in \mathbb{R}^q$ is the estimator output, and $L \in \mathbb{R}^{n \times p}$ is the observer gain matrix. A schematic of the Luenberger observer is shown in Figure 1.1. Since (A, C) is observable, there exists $L \in \mathbb{R}^{n \times q}$ such that the

error system

$$e(k+1) = (A - LC)e(k),$$

is asymptotically stable. In other words, when (A, C) is observable, we can place the eigenvalues of $(A - LC)$ at any desired location. However, in the case where A, and C, or $u(k)$ are unknown, then choosing L to stabilize $(A-LC)$ is not straightforward. Furthermore, the term $L(y(k) - \hat{y}(k))$ does not account for the unknown input.

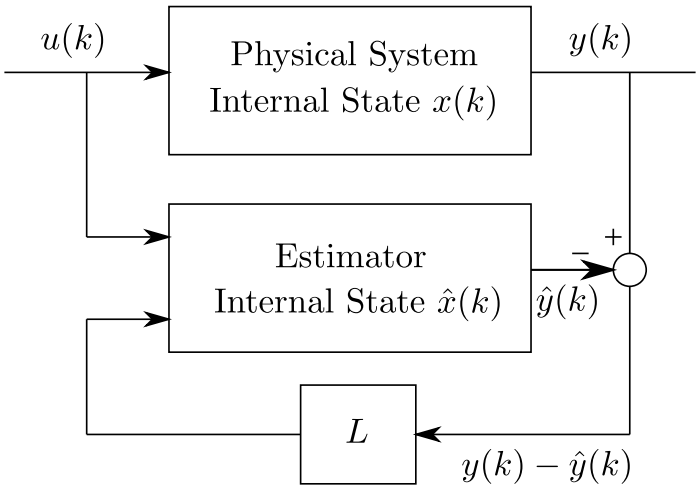


Figure 1.1: Architecture of the Luenberger Observer. The residual $\hat{y}(k) - y(k)$ is multiplied by the observer gain L, which is fed back into the estimator to drive the residual to zero.

For discrete-time systems, one of the earlier approaches to unknown-input state-estimation was presented by Kitanidis [12], in which a recursive unbiased minimum-variance filter was developed in the presence of an arbitrary unknown input. The approach taken was to solve an optimization problem in the presence of an algebraic constraint that guaranteed that the filter was unbiased. This filter is only robust to the presence of an unknown input, and does not provide an estimate of the input itself. In [14], Darouach and Zasadzinski solved the optimization problem proposed in [12] under more generalized conditions, from which the Kitanidis filter can be

obtained as a special case. Kerwin and Prince [26] showed that the recursive form of the filter considered in [12] and [14] was indeed the optimal form for the unbiased minimum-variance estimator.

The results in [12] and [14] were extended by [1], which provides minimum-variance unbiased estimates of both the states and the unknown inputs. The filters in [12] and [14] can be obtained as special cases of the filter in [1]. These methods were then extended to the case where the unknown input or disturbance affects not only the state but also the measurements [27, 28]. The above filters are also subject to an observer matching condition, which was relaxed in [18] by designing a delayed observer. The drawback of these methods is that they cannot be applied to nonminimum-phase systems. A method for simultaneous estimation of states and inputs for nonminimum-phase systems was proposed in [29], where it is assumed that the input is slowly time varying, and remains approximately constant over a certain time interval.

In this thesis, we consider the Retrospective Cost Unknown Input Observer (RCUIO), which is based on Retrospective Cost Adaptive Control (RCAC) [30,31] shown in Figure 1.2. RCAC uses the residual $\hat{y}(k) - y(k)$ to update an adaptive feedback system which then drives the residual $\hat{y}(k) - y(k)$ to zero. The implementation of RCAC to unknown input state estimation, known as RCUIO, is shown in Figure 1.3. This ensemble-free method has the following features:

1. RCUIO is a minimal modeling approach that only needs a limited number of Markov parameters. For some systems, we have shown that RCUIO can be successfully applied using only one Markov parameter. Due to this, RCUIO can be applied to systems where explicit equations of the dynamics of the system are not known, such as large scale computational fluid dynamics (CFD) codes.

2. Using small modifications to RCUIO, it can be applied to systems with non-minimum phase zeros.

3. Because RCUIO uses only a limited number of Markov parameters, it turns out to be effective for some nonlinear systems.

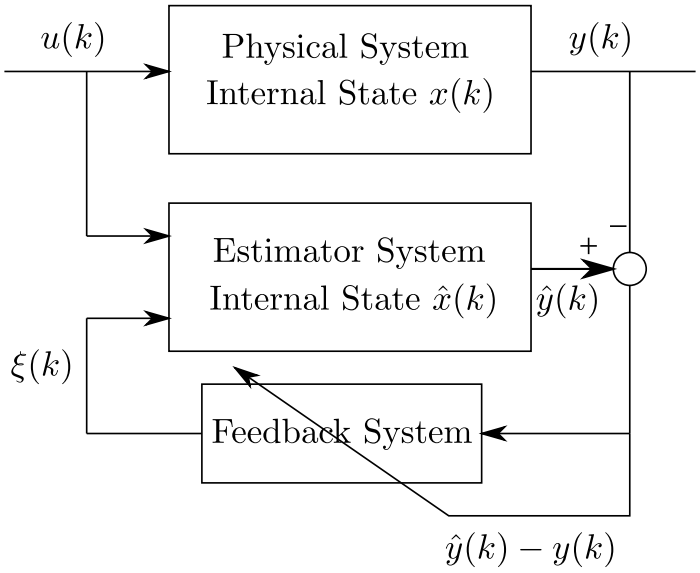


Figure 1.2: Architecture of RCAC. The residual $\hat{y}(k) - y(k)$ is used to update an adaptive feedback system, which then drives the residual to zero.

1.1 Contributions

- RCUIO relies on recursive least squares. In this thesis, we develop a novel variable regularization sliding window recursive least squares algorithm, and analyze its convergence properties, numerical stability, and computational complexity. We also incorporate this algorithm into RCUIO, and compare it with RCUIO with standard least squares.

- We successfully use RCUIO and the Global Ionosphere-Thermosphere Model [32] to obtain estimates of the states and input in the ionosphere-thermosphere using simulated and real satellite data.

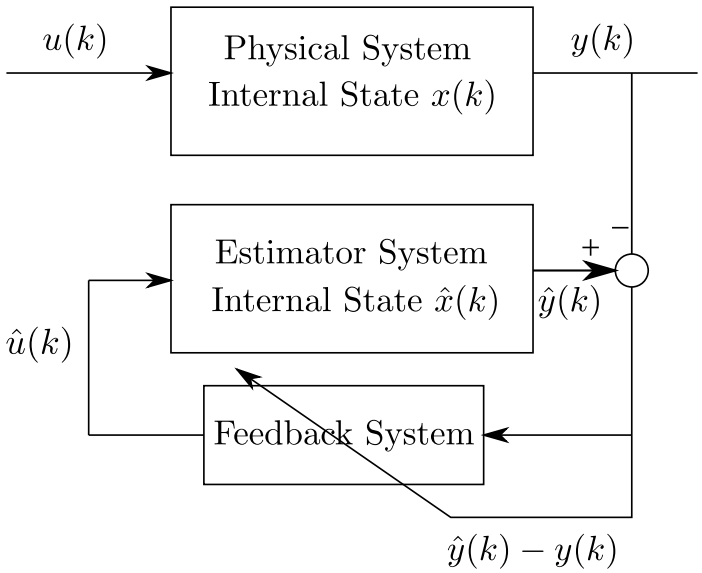


Figure 1.3: Architecture of RCUIO. The residual $\hat{y}(k) - y(k)$ is used to update an adaptive feedback system, which then drives the residual to zero and, as a consequence, yields an estimate $\hat{u}(k)$ of $u(k)$.

1.2 Chapter Outlines

This dissertation is organized in the following chapters.

1.2.1 Chapter II

The classical Kalman filter is the optimal state estimator for linear systems under white process and sensor noise with zero mean and finite second moments. Implementation of the optimal estimator under these idealized conditions depends on knowledge of the linear dynamics and noise covariances. When these assumptions are not satisfied, the accuracy of the Kalman filter can be degraded [33–35].

If the transfer function from the process noise to the measurements is minimum phase, the number of outputs equals the number of disturbances, and there is no sensor noise, then the minimum achievable estimation error is zero [36]. On the other hand, the presence of nonminimum-phase zeros increases the minimum achievable

estimation error and thus, for harmonic disturbances, the Kalman filter does not give perfect state estimates [37, 38].

A more proactive approach is to implement an adaptive state estimator, where the goal is to identify the dynamics and noise statistics during system operation and use this information to tune the estimator on-line [39].

In addition to compensating for white process noise, the Kalman filter accommodates the presence of a known, deterministic input. By injecting this signal into the estimator, the estimator experiences no loss of estimation accuracy relative to the case in which no deterministic input is present. This feature is essential when the Kalman filter is used in conjunction with the linear-quadratic regulator for constructing the full-order dynamic LQG controller.

In practice, however, the deterministic input may not be known exactly, and this error can viewed as a component of the process noise. However, this approach may be conservative and can lead to bias when the unknown input has a nonzero "mean" value. Consequently, a more direct approach is to extend the estimator to include an estimate of the unknown input [13, 20, 40, 41]. Yet another approach is to constrain the gains of the estimator in order to guarantee that the state estimates are unbiased [1, 12, 14, 42].

In this chapter we consider state estimation for minimum- or nonminimum-phase systems in the presence of an unknown harmonic input. To address this problem we consider the estimator structure shown in Figure 2.13 with an auxiliary input $\hat{u}$, which is the output of an adaptive feedback system that is updated on-line. The signal $\hat{u}$ is estimated using a retrospective-cost-based input-reconstruction technique. In this way, the adaptive feedback system uses knowledge of the estimator residual to improve the accuracy of the state estimator by reconstructing the harmonic disturbance, thereby achieving perfect estimates in the minimum and nonminimum-phase cases. A related technique is used in

1.2.2 Chapter III

Recursive-least-squares (RLS) and gradient-based algorithms are widely used in signal processing, estimation, identification, and control [43–51]. Under ideal conditions, that is, noiseless measurements and persistency of the data, these techniques are guaranteed to converge to the minimizer of a quadratic function [44, 47]. In practice, the accuracy of the estimates depends on the level of noise and the persistency of the data.

The standard RLS algorithm operates on a growing window of data, where new data are added to the RLS cost function as they become available and past data are progressively discounted through the use of a forgetting factor. In contrast, sliding-window RLS algorithms [52–56] require no forgetting factor since they operate on a finite data window of fixed length, where new data replace past data in the RLS cost function. Sliding-window least-squares techniques are available in both batch and recursive formulations. As shown in [53], sliding-window RLS algorithms have enhanced tracking performance compared to standard RLS algorithms in the presence of time-varying parameters.

In standard RLS, the positive-definite initialization of the covariance matrix is the inverse of the weighting on a regularization term in a quadratic cost function. This regularization term compensates for the potential lack of persistency, ensuring that the cost function has a unique minimizer at each step. Traditionally, the regularization term is fixed for all steps of the recursion. Additionally, an optimally regularized adaptive filtering algorithm with constant regularization is presented in [57]. However, variants of RLS with time-varying regularization have been developed in the context of adaptive filtering, echo cancellation, and affine projection [58–63].

In the present work, we derive a novel sliding-window variable-regularization RLS (SW-VR-RLS) algorithm, where the weighting on the regularization term can change at each step. An additional extension presented in this chapter also involves the

regularization term. Specifically, the regularization term in standard RLS weights the difference between the next estimate and the initial estimate, while the regularization term in sliding-window RLS weights the difference between the next estimate and the estimate at the beginning of the sliding window. In this work, the regularization term weights the difference between the next estimate and an arbitrarily chosen time-varying vector. As a special case, the time-varying vector can be the current estimate or a recent estimate. These variable-regularization extensions of sliding-window RLS can facilitate tradeoffs among transient error, rate of convergence, and steady-state error.

In this work, we derive the SW-VR-RLS equations and analyze their convergence properties in the absence of noise. While standard RLS entails the update of the estimate and the covariance matrix, sliding-window RLS involves the update of an additional symmetric matrix of size $n \times n$, where n is the dimension of the estimate. Furthermore, SW-VR-RLS requires updating of one more symmetric matrix of size $n \times n$ to account for the time-varying regularization.

The SW-VR-RLS algorithm was first presented in [64] together with a preliminary numerical study and without convergence analysis. In addition, a growing-window RLS algorithm with time-varying regularization appears in [65]. The goal of this work is to provide a more complete development of the SW-VR-RLS algorithm, including an analysis of convergence and numerical stability.

1.2.3 Chapter V

In this chapter, we derive a growing-window variable-regularization RLS (GW-VR-RLS) algorithm, where the weighting of the regularization term changes at each step. As a special case, the regularization can be decreased in magnitude or rank as the rank of the covariance matrix increases, and can be removed entirely when no longer needed. This ability is not available in standard RLS where the regularization term

is weighted by the inverse of the initial covariance at every step.

A second extension presented in this work also involves the regularization term. Specifically, the regularization term in standard RLS weights the difference between the next state estimate and the initial state. In GW-VR-RLS, the regularization term weights the difference between the next state estimate and an arbitrarily chosen time-varying vector of parameters. As a special case, the time-varying vector can be the current state estimate, and thus the regularization term weights the difference between the next state estimate and the current state estimate. This formulation allows us to modulate the rate at which the current estimate changes from step to step.

For these extensions, we derive GW-VR-RLS update equations. While standard RLS entails the update of the state estimate and the covariance matrix, GW-VR-RLS entails the update of an additional symmetric matrix of dimension $n \times n$ to allow for the variable regularization. Thus, GW-VR-RLS entails some additional computational burden relative to classical RLS.

1.2.4 Chapter VI

This chapter is concerned with modeling and prediction of space weather effects. In the near-Earth environment, the effects of space weather are primarily manifested by the properties of the ionosphere and thermosphere, which influence radio propagation and satellite drag. The sun is one of the primary drivers of the ionosphere and thermosphere. In particular, extreme ultraviolet (EUV) and X-ray radiation produce photo-ionization, which, in turn, through chemistry and heating, drives the formation of the ionosphere and shapes the thermosphere. In addition, the effect of the EUV and X-ray radiation is sufficient to render the ionosphere-thermosphere a strongly driven system.

Since a significant portion of EUV and X-ray radiation is absorbed by the at-

mosphere, it is not possible to measure these quantities from the ground. Instead, a proxy is used. The most common proxy for EUV and X-ray radiation is the flux solar irradiance at a wavelength of 10.7 cm ($F_{10.7}$), which is measured (in units of 10^{-22} W Hz^{-1} m^{-2} = 1 solar flux unit (SFU)) by the Dominion radio observatory in Penticton, Canada [66]. A shortcoming of this technique is that $F_{10.7}$ does not have a one-to-one correlation with each of the wavelengths in the EUV and X-ray bands, and thus the measured $F_{10.7}$ is often a misrepresentation of the true solar spectrum.

Although our ultimate goal is to estimate the true flux in multiple EUV and X-ray wavelength bins, a more attainable intermediate goal is to estimate the value of $F_{10.7}$ that best characterizes the ionosphere and thermosphere. The ability to estimate $F_{10.7}$ from alternative measurements can provide a cross check on the available measurements, while also providing an illustrative proof-of-concept demonstration of the adaptive state estimation algorithm described below as a first step toward estimating X-ray and EUV in multiple bands. Furthermore, current models do not fully capture the dynamics of the ionosphere-thermosphere, in which case $F_{10.7}$ can be used as an input to the model for the purpose of eliminating the errors between real measurements and simulated measurements. This study thus attempts to specify $F_{10.7}$ based on simulated measurements of the atmosphere as well as with real satellite data. The specified $F_{10.7}$ can then be used to obtain improved estimates of the state of the ionosphere and thermosphere globally and possibly predict its future evolution. This is a problem of state and input estimation.

To estimate $F_{10.7}$, we use the Global Ionosphere Thermosphere Model (GITM) [32]. GITM simulates the density, temperature, and winds in the thermosphere and ionosphere across the globe from 100 km to 600 km altitude, depending on the solar conditions at the time. The main inputs to GITM are the high-latitude electrodynamics (i.e., the aurora and the associated electric fields), tides from the lower atmosphere, and the brightness of the sun at various wavelengths, which can be proxied through

the use of $F_{10.7}$. GITM solves for the chemistry, dynamics, and thermodynamics of the upper atmosphere self-consistently by accounting for interactions among various species of ions and neutrals.

In this work, we use the retrospective cost adaptive unknown input observer (RCUIO) technique given in Chapter 2 to estimate the unknown solar driver $F_{10.7}$ using the Global Ionosphere-Thermosphere Model and satellite measurements. RCUIO assumes that the input to the system is unknown, and uses retrospective optimization to construct an input to the adaptive estimator that minimizes the retrospective cost function given in Chapter 2. The retrospectively optimized input is then used to asymptotically drive the error between the measured output and the estimator output to zero. In this way, RCUIO asymptotically estimates the unknown input to the system and the unknown states of the system. A useful feature of RCUIO is that an explicit nonlinear or linearized model is not required. In addition, unlike ensemble-based data-assimilation algorithms [23, 24, 67], RCUIO uses only one copy of the system model and thus is ensemble-free.

The derivation of the RCUIO algorithm given in Chapter 2 is based on a linear dynamics model, and the modeling information needed to implement the algorithm consists of components of the impulse response. However, since RCUIO does not require an explicit model of the dynamics (which may, for example, be in the form of a computer code as in the case of GITM), RCUIO can be applied to nonlinear systems. In this case, ersatz modeling information can be chosen based on the qualitative behavior of the system, or can be determined by trial and error. Numerical examples show that RCUIO is effective on nonlinear plants, which is presumably due to the fact that extremely limited modeling information is required by the algorithm.

In [68], RCUIO was used to estimate a constant $F_{10.7}$ in 3D GITM using simulated measurement data, where the measurement was assumed to be at a fixed position in the terrestrial atmosphere. The goal of this work is to extend this application to the

case where the measurements are obtained from real or simulated satellites. Furthermore, the ability of the method to estimate a time-varying $F_{10.7}$ using simulated data is demonstrated.

CHAPTER 2

Retrospective-Cost Adaptive Unknown Input Observers

In this chapter, we formulate the unknown input estimation problem and develop the Retrospective-Cost Adaptive Unknown Input Observer (RCUIO). In Section 2.1, we define input observability for linear systems and, in Section 2.2, we consider RCUIO for a special case where the first Markov parameter is non-zero and only one measurement is used. In Section 2.3, we derive RCUIO for general systems, and in Section 2.4, we analyze the computational complexity of RCUIO. Finally, in Section 2.5, we give illustrative examples.

2.1 Input Observability

Consider the linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k), \tag{2.1}$$

$$y(k) = Cx(k), \tag{2.2}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $y \in \mathbb{R}^q$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, and $C \in \mathbb{R}^{q\times n}$. The input $u(k)$ is assumed to be unknown. Next, let ℓ be a positive integer, and define

$$\mathcal{Y}_\ell \triangleq \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_\ell \end{bmatrix}, \quad \mathcal{U}_\ell \triangleq \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_\ell \end{bmatrix}, \quad \Gamma_\ell \triangleq \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{\ell-1} \end{bmatrix},$$

$$M_\ell = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{\ell-1}B & CA^{\ell-2}B & \cdots & CB \end{bmatrix}, \quad \Psi_\ell \triangleq \begin{bmatrix} \Gamma_\ell & M_\ell \end{bmatrix}.$$

Definition 2.1.1 ([20])**.** Let $\ell \geq 1$. Then the input and state unobservable subspace of (2.1) and (2.2) is given by $\left\{ \begin{bmatrix} x_0 \\ \mathcal{U}_\ell \end{bmatrix} \in \mathbb{R}^{n+\ell p} : \mathcal{Y}_\ell = 0 \right\}$.

Next, define

$$\ell_0 \triangleq \begin{cases} \max\{\lceil \frac{n-q}{q-p} \rceil, 1\}, & p < q, \\ 1, & p = q, \end{cases}$$

where $\lceil a \rceil$ is the smallest integer greater than or equal to a. We now give sufficient conditions for input and state observability.

Theorem 1 ([20])**.** The following statements are equivalent:

(i) (2.1) and (2.2) are input and state observable.

(*ii*) For all $\ell \geq \ell_0$, $\mathcal{Y}_\ell = 0$ if and only if $\begin{bmatrix} x_0 \\ \mathcal{U}_{\ell-1} \end{bmatrix} = 0$.

(*iii*) For all $\ell \geq \ell_0$, $\operatorname{rank}(\Psi_\ell) = n + \ell p$.

(*iv*) $\operatorname{rank}(\Psi_{n-1}) = n + (n-1)p$ and, for all $\ell \geq \ell_0$, $\operatorname{rank}(C_\ell B_{\ell-1})$.

2.2 Problem Formulation : Full Column Rank First Markov Parameter

In this section, we derive RCUIO for linear systems where the first Markov parameter has full column rank. The purpose of this chapter is to give insight into RCUIO and comparisons to other unknown input state estimators. A more general development of RCUIO is done in Section 2.3. Consider the linear time-invariant system

$$x(k) = Ax(k-1) + Bu(k-1), \tag{2.3}$$

$$y(k) = Cx(k), \tag{2.4}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $y \in \mathbb{R}^q$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, $C \in \mathbb{R}^{q \times n}$. Furthermore, we assume that $u(k)$ is unknown and we consider the following assumption:

Assumptions 2.2.1. CB has full column rank.

Assumption 2.2.1 is used to simplify the analysis, and will not be used in Section 2.3. The goal is to obtain $u(k)$ and $x(k)$ using $y(k)$. First, we develop an unknown input and state estimator based on direct input reconstruction.

2.2.1 Unknown Input and State Estimation Using Direct Input Reconstruction

Let $\hat{x}(k-1)$ be the estimate of $x(k-1)$. For $k > 0$, define

$$x^-(k) = A\hat{x}(k-1) + Bu^-(k-1), \tag{2.5}$$

$$y^-(k) = Cx^-(k), \tag{2.6}$$

where $x^-(k)$, $y^-(k)$, $u^-(k-1) \in \mathbb{R}$ are prior estimates. Note that $u^-(k-1)$ can be chosen to be any value. Next, define the error

$$z(k) = y^-(k) - y(k). \tag{2.7}$$

Using (2.5) and (2.6), it follows from (2.7) that

$$z(k) = CA\hat{x}(k-1) + CBu^-(k-1) - y(k).$$

We now wish to replace $u^-(k-1)$ with an input $u^*(k-1)$ that minimizes $z(k)$. Define

$$\hat{z}(k) \triangleq CA\hat{x}(k-1) + CBu^*(k-1) - y(k),$$

and the optimization problem

$$\begin{aligned}
&\min_{u^*(k-1)\in\mathbb{R}} \left(\hat{z}^{\mathrm{T}}(k)\hat{z}(k)\right) \\
&= \min_{u^*(k-1)\in\mathbb{R}} \left[(CA\hat{x}(k-1) + CBu^*(k-1) - y(k))^{\mathrm{T}}(CA\hat{x}(k-1) + CBu^*(k-1) - y(k))\right], \\
&= \min_{u^*(k-1)\in\mathbb{R}} \left[(CBu^*(k-1) - CBu^-(k-1) + z(k))^{\mathrm{T}}(CBu^*(k-1) - CBu^-(k-1) + z(k))\right].
\end{aligned} \tag{2.8}$$

Note that (2.8) resembles the least squares minimization problem

$$\min_{x\in\mathbb{R}} \left[(b-Ax)^{\mathrm{T}}(b-Ax)\right]$$

with $x = u^*(k-1)$, $b = CBu^-(k-1) - z(k)$, and $A = CB$. Solving (2.8) yields

$$u^*(k-1) = [(CB)^{\mathrm{T}}CB]^{-1}(CB)^{\mathrm{T}}\left(CBu^-(k-1) - z(k)\right). \tag{2.9}$$

Next, we set $\hat{u}(k-1) = u^*(k-1)$, and obtain the posterior state and output estimates using

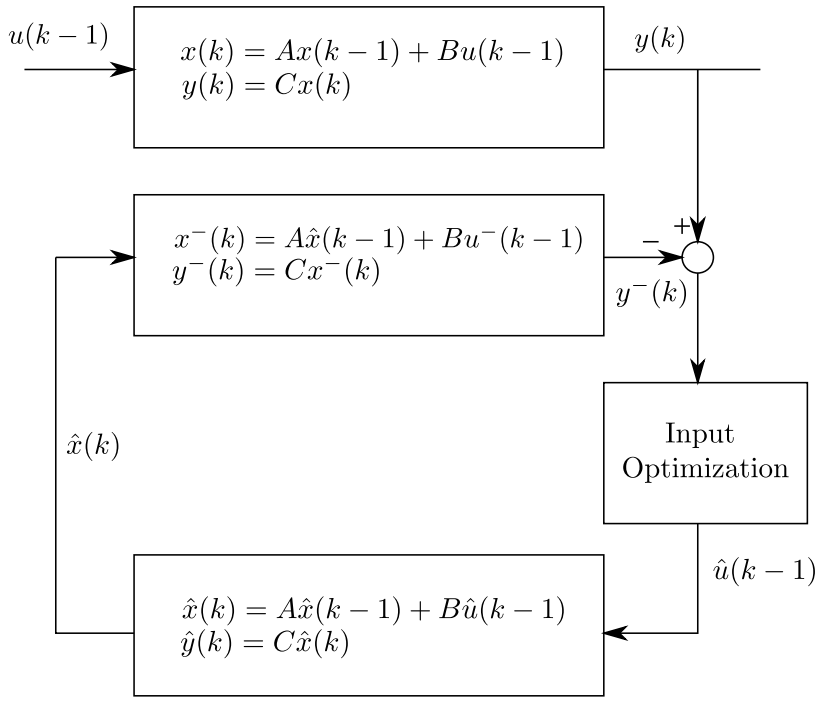
$$\hat{x}(k) = A\hat{x}(k-1) + B\hat{u}(k-1), \tag{2.10}$$

$$\hat{y}(k) = \hat{x}(k). \tag{2.11}$$

The schematic of the direct input reconstruction based estimator is shown in Figure 2.1. Next, define the error $e(k) = \hat{x}(k) - x(k)$. It follows from (2.10), (2.11), and (2.9) that

$$\begin{aligned}
e(k) &= \hat{x}(k) - x(k) \\
&= A\hat{x}(k-1) + B\hat{u}(k-1) - Ax(k-1) - Bu(k-1) \\
&= Ae(k-1) - Bu(k) - B[(CB)^{\mathrm{T}}CB]^{-1}(CB)^{\mathrm{T}}\left(CBu^-(k-1) - z(k)\right) \\
&= Ae(k-1) - Bu(k) - B[(CB)^{\mathrm{T}}CB]^{-1}(CB)^{\mathrm{T}}\left(CA\hat{x}(k-1) - y(k)\right) \\
&= Ae(k-1) - Bu(k-1) - B[(CB)^{\mathrm{T}}CB]^{-1}(CB)^{\mathrm{T}}\left(CAe(k-1) - CBu(k-1)\right) \\
&= \left(A - B[(CB)^{\mathrm{T}}CB]^{-1}(CB)^{\mathrm{T}}CA\right)e(k-1).
\end{aligned}$$

It can be shown that the eigenvalues of $A - B[(CB)^{\mathrm{T}}CB]^{-1}(CB)^{\mathrm{T}}CA$ are the zeros of (2.10) and (2.11). Therefore, if the system (2.3), (2.4) is input observable and minimum-phase, then the error system is asymptotically stable. This method is



At step k:
Known: $\hat{x}(k-1)$, $y(k)$ up to step k.
1) Calculate $x^-(k) = A\hat{x}(k-1) + Bu^-(k-1)$, $u^-(k-1)$ arbitrary.
2) Calculate $y^-(k) = Cx^-(k)$.
3) Obtain $\hat{u}(k-1)$ using input optimization.
4) Calculate $\hat{x}(k) = A\hat{x}(k-1) + B\hat{u}(k-1)$.

Figure 2.1: Schematic of the direct input reconstruction based estimator.

demonstrated in the following example

Example 2.2.1. Consider the system

$$x(k+1) = \begin{bmatrix} 0.2 & 2 \\ 0 & 0.4 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(k),$$
$$y(k) = \begin{bmatrix} 0.1 & 0.5 \end{bmatrix} x(k), \tag{2.12}$$

where $u(k) = 5\sin(0.1k)$ and is unknown. Furthermore, $\hat{x}(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}$. The poles and zero of (2.12) are at 0.2, 0.4 and -0.2, respectively, and thus (2.12) is minimum phase. Figures 2.2 and 2.3 shows $x(k)$, $\hat{x}(k)$ and $u(k)$, $\hat{u}(k)$, respectively. For this

example, $\hat{x}(k)$ converges to $x(k)$, and $\hat{u}(k)$ converges to $u(k)$.

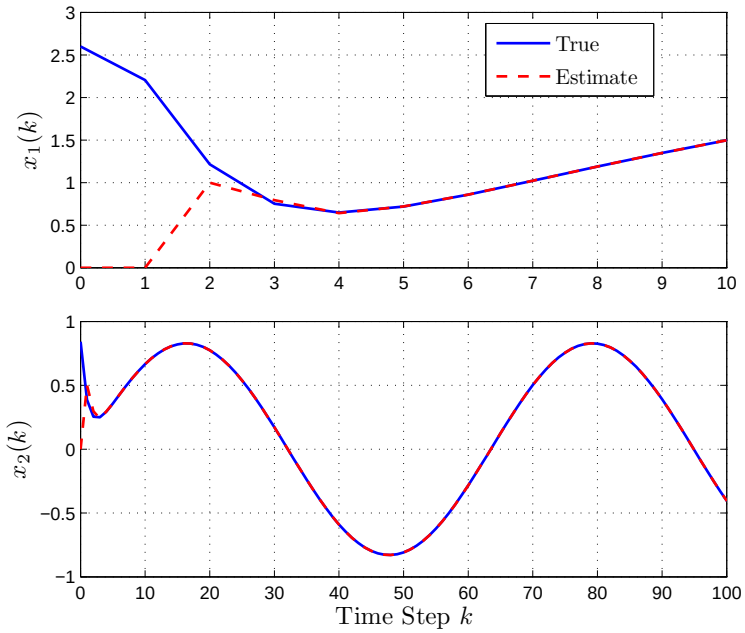


Figure 2.2: State estimates for the minimum-phase system given by (2.12) using direct input reconstruction. For this example, $\hat{x}(k)$ converges to $x(k)$.

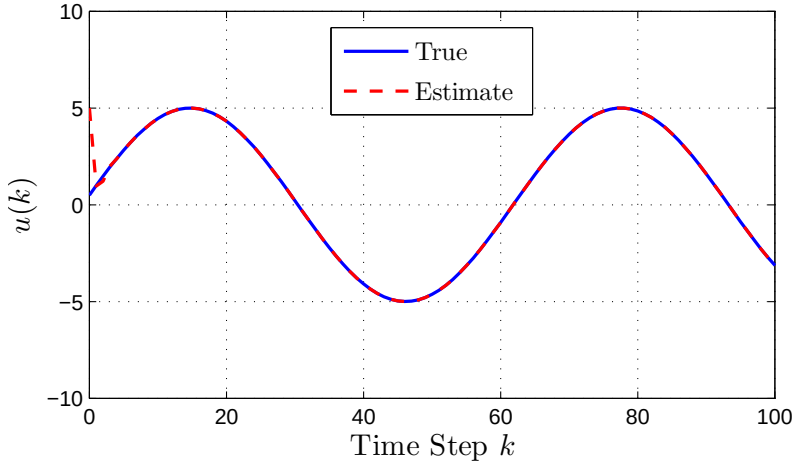


Figure 2.3: Input estimates for the minimum-phase system given by (2.12) using direct input reconstruction. For this example, $\hat{u}(k)$ converges to $u(k)$.

Example 2.2.2. Consider the system

$$x(k+1) = \begin{bmatrix} 0.2 & 2 \\ 0 & 0.4 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(k),$$
$$y(k) = \begin{bmatrix} 1.2 & 0.5 \end{bmatrix} x(k), \tag{2.13}$$

where $u(k) = 5\sin(0.1k)$ and is unknown. Furthermore, $\hat{x}(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}$. The poles and zero of (2.13) are at 0.2, 0.4 and -4.6, respectively, and thus (2.13) is nonminimum phase. Figures 2.4 and 2.5 shows $x(k)$, $\hat{x}(k)$ and $u(k)$, $\hat{u}(k)$, respectively. For this example, $\hat{x}(k)$ and $\hat{u}(k)$ are unbounded.

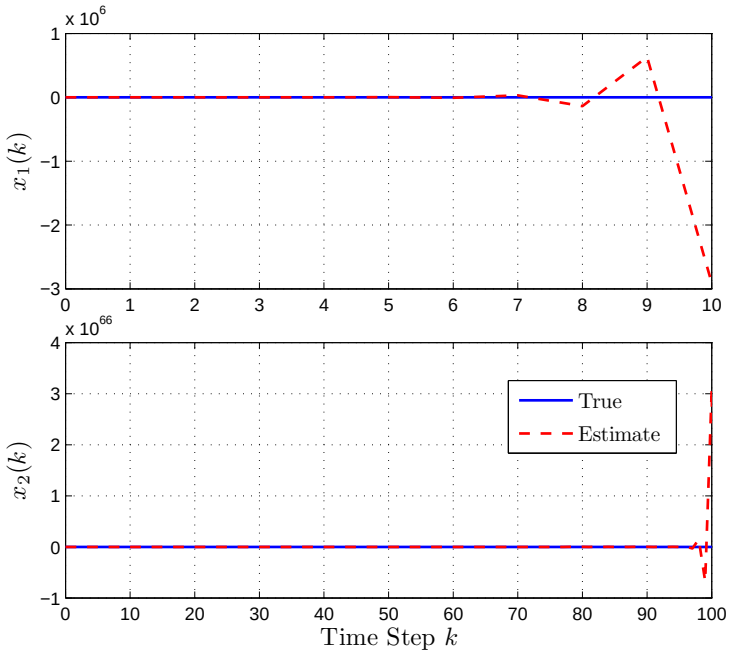


Figure 2.4: State estimates for the nonminimum-phase system given by (2.13) using direct input reconstruction. For this example, $\hat{x}(k)$ is unbounded.

In [1], a weighted least squares cost function is used to obtain a minimum variance unbiased input estimate. However, when the output and process noise covariances are zero, the weighted least squares cost function in [1] reduces to (2.8). In the following

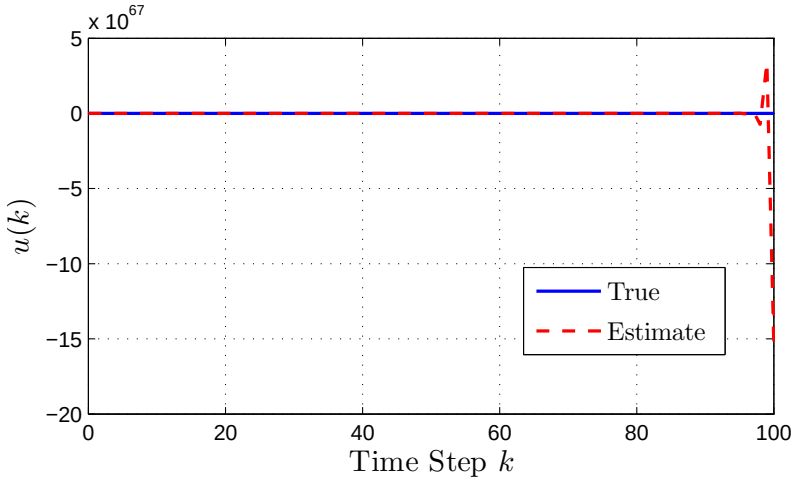


Figure 2.5: Input estimates for the nonminimum-phase system given by (2.13) using direct input reconstruction. For this example, $\hat{u}(k)$ is unbounded.

examples we show that, for nonminimum phase MIMO systems, even in the presence of process and output noise, the filter in [1] can be unstable.

Example 2.2.3. Consider the MIMO system

$$x(k+1) = \begin{bmatrix} 0.5 & -0.34 & 0.208 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.125 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & -0.34 & 0.208 \\ 0 & 0 & 0 & 05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.125 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 16 & 00 \\ 0 & 0 \\ 0 & 0 \\ 0 & 16 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} u(k) + w(k),$$

$$y(k) = \begin{bmatrix} 0.0625 & 0.1250 & 2 & 0.0625 & 0 & -12 \\ 0.0625 & 0.1250 & -4 & 0.1250 & -0.1250 & -8 \\ 0.0625 & 0 & -4 & 0.1250 & 0 & -8 \end{bmatrix} x(k) + \nu(k), \tag{2.14}$$

where $u(k)$, $w(k) \in \mathbb{R}^6$, $v(k) \in \mathbb{R}^3$, and $w(k)$ and $v(k)$ are sampled from white noise

processes with zero-mean Gaussian distributions with variances

$$Q = \begin{bmatrix} 0.0245 & -0.0157 & -0.0277 & 0.0041 & 0.0019 & -0.0151 \\ -0.0157 & 0.0194 & 0.0219 & -0.0032 & -0.0101 & 0.0124 \\ -0.0277 & 0.0219 & 0.1220 & -0.0131 & 0.0541 & -0.0479 \\ 0.0041 & -0.0032 & -0.0131 & 0.0095 & 0.0167 & -0.0085 \\ 0.0019 & -0.0101 & 0.0541 & 0.0167 & 0.1122 & -0.0670 \\ -0.0151 & 0.0124 & -0.0479 & -0.0085 & -0.0670 & 0.1145 \end{bmatrix},$$

and

$$R = \begin{bmatrix} 0.0650 & -0.0133 & -0.0228 \\ -0.0133 & 0.1123 & 0.0034 \\ -0.0228 & 0.0034 & 0.0144 \end{bmatrix},$$

respectively, and the unknown input is $u(k) = \begin{bmatrix} 10\sin(k) & 0.1\cos(k) \end{bmatrix}^{\mathrm{T}}$. The poles of (2.14) are $0.2 \pm 0.3i$, $0.2 \pm 0.3i$, 0.1, and 0.1, and the zero of (2.14) is 2, and thus (2.14) is nonminimum-phase. The matrices Q and R are assumed to be known. We use the filter equations given in [1]. Figure 2.6 shows the first four elements of $x(k)$ and $\hat{x}(k)$, and Figure 2.7 shows $u(k)$ and $\hat{u}(k)$. These figures show that, for this example, the filter given in [1] is unstable.

Since the direct input reconstruction method yields an unbounded input estimate, we test the effect of saturating the estimated input $\hat{u}(k)$ obtained from direct input reconstruction.

Example 2.2.4. Consider the same setup as in Example 2.2.2. Let $u^*_{\max} = 5$, $u^*_{\min} =$

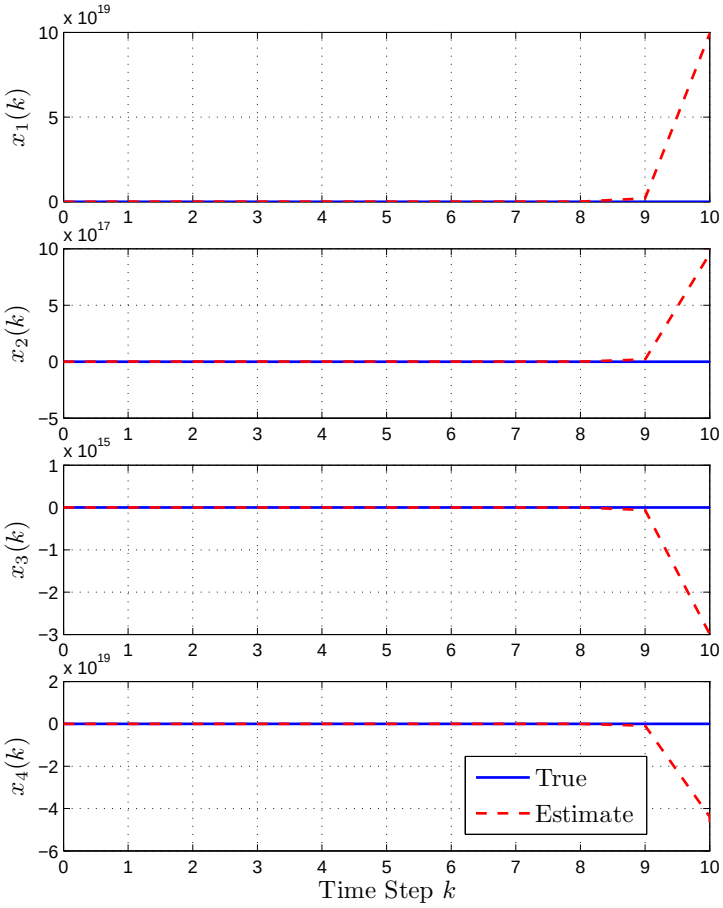


Figure 2.6: State estimates for system given by (2.14) using the filter given in [1]. For this example, the state estimates are unbounded.

-5, and

$$u^*(k-1) = \begin{cases} u^*_{max}, & \text{if } u^*(k-1) > u_{max}, \\ u^*_{min}, & \text{if } u^*(k-1) < u_{min}, \\ u^*(k-1), & \text{otherwise.} \end{cases}$$

Figure 2.8 shows $\hat{x}(k)$ and $x(k)$. For this example, although $\hat{x}(k)$ is bounded in contrast to the $\hat{x}(k)$ in Example 2.2.2, $\hat{x}(k)$ does not converge to x_k. Figure 2.9 shows $\hat{u}(k)$ and $u(k)$, and shows that $\hat{u}(k)$ does not converge to $u(k)$.

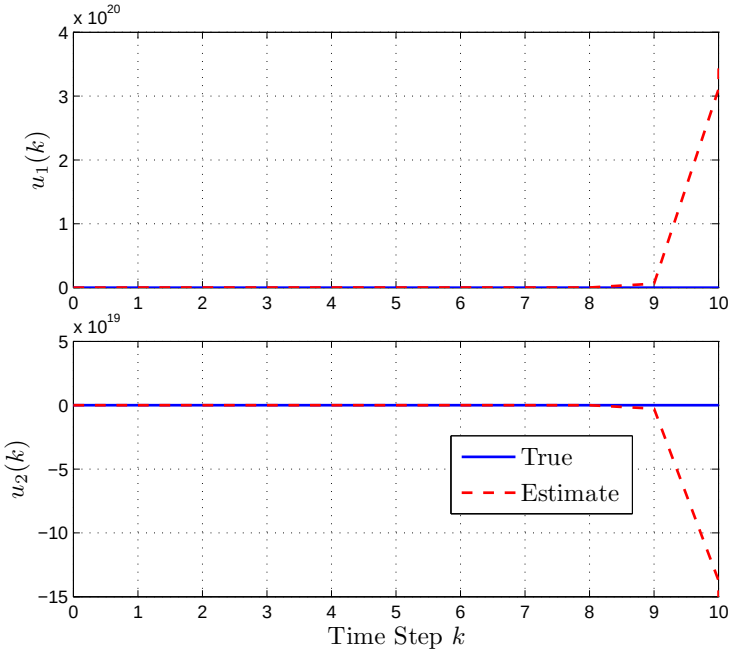


Figure 2.7: Input estimates for system given by (2.14) using the filter given in [1]. For this Example, the input estimates are unbounded.

2.2.2 RCUIO: Combining direct input reconstruction with the adaptive feedback system

Instead of setting $\hat{u}(k-1) = u^*(k-1)$ and using it in (2.10), as was done in the direct input reconstruction method, where $u^*(k-1)$ is obtained from equation (2.9), we use $u^*(k-1)$ to update the coefficients $m_i(k)$ and $n_i(k)$ of the adaptive feedback system

$$\hat{u}(k) = \sum_{i=1}^{n_c} m_i(k)\hat{u}(k-i) + \sum_{i=0}^{n_c} n_i(k)z(k-i),$$

where n_c is the order of the adaptive feedback system. The adaptive feedback system is then used to obtain $\hat{u}(k)$, which is consequently used to obtain $\hat{x}(k+1)$. The combination of input reconstruction and adaptive feedback system update yields RCUIO. Note that, unlike the direct input reconstruction based estimator, RCUIO does not

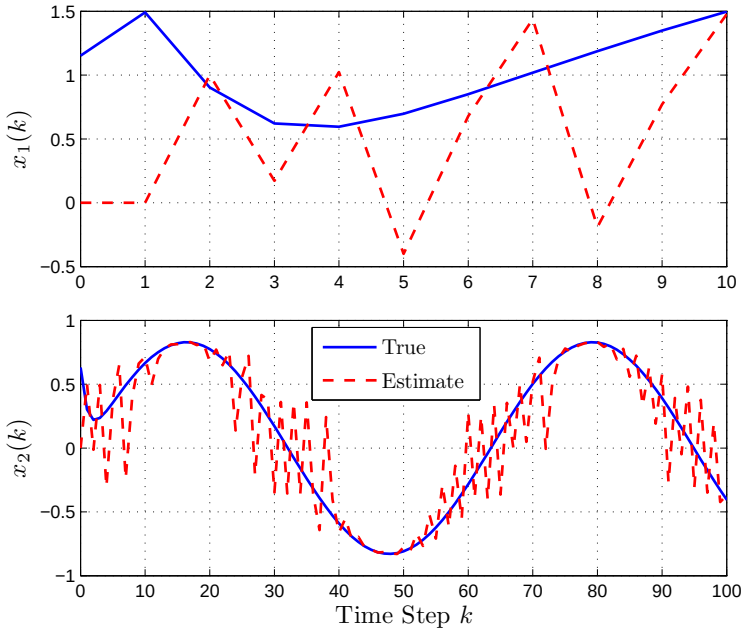


Figure 2.8: State estimates for Example 2.2.4 for the nonminimum phase system (2.13) using direct input reconstruction with saturated $u^*(k-1)$. For this example, the state estimates $\hat{x}(k)$ are bounded, but do not converge to the true states $x(k)$.

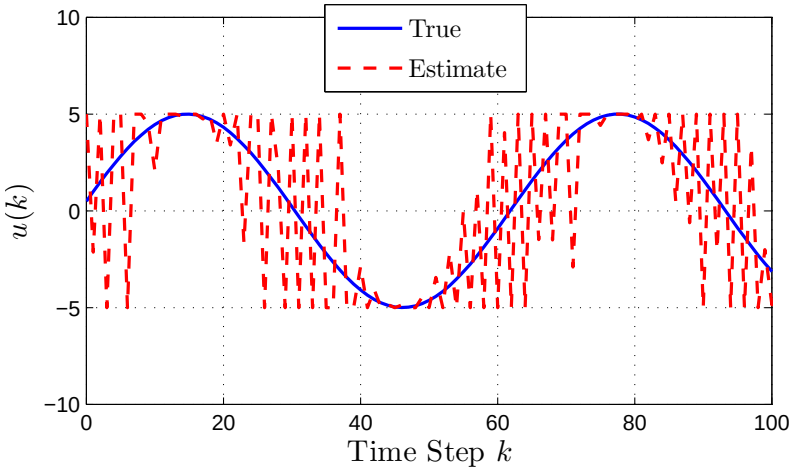


Figure 2.9: Input estimates for Example 2.2.4 for the nonminimum phase system (2.13) using direct input reconstruction with saturated $u^*(k-1)$. For this example, $\hat{u}(k)$ does not converge to $u(k)$.

use the optimized input $u^*(k-1)$ to update $\hat{x}(k)$, but instead obtains $\hat{x}(k+1)$. In this respect, RCUIO resembles the Luenberger observer, which uses $y(k)$ to obtain $\hat{x}(k+1)$, and does not use it to update $\hat{x}(k)$.

Since $\hat{u}(k)$ is computed at step k, it follows that $\hat{u}(k-1)$ is available at step $k+1$. Therefore, in (2.5), if we set $u^-(k-1) = \hat{u}(k-1)$, then it follows from (2.10) that $x^-(k) = \hat{x}(k)$, and since $\hat{x}(k)$ is known as it was computed at step $k-1$, we do not need to first compute $x^-(k-1)$ and then $\hat{x}(k-1)$. This is beneficial from a computational point of view. In this dissertation, we focus on this implementation of RCUIO. The schematic of RCUIO is given in 2.10.

We now derive RCUIO. Define the estimator equations

$$\hat{x}(k) = A\hat{x}(k-1) + B\hat{u}(k-1), \tag{2.15}$$

$$\hat{y}(k) = C\hat{x}(k), \tag{2.16}$$

$$\hat{u}(k) = \sum_{i=1}^{n_c} m_i(k)\hat{u}(k-i) + \sum_{i=0}^{n_c} n_i(k)z(k-i). \tag{2.17}$$

As in the case of the direct input reconstruction based estimator, we define

$$z(k) \stackrel{\triangle}{=} \hat{y}(k) - y(k). \tag{2.18}$$

Using (2.15) and (2.16), it follows from (2.18) that

$$z(k) \stackrel{\triangle}{=} CA\hat{x}(k-1) + CB\hat{u}(k-1) - y(k).$$

Next, define

$$\hat{z}(k) \stackrel{\triangle}{=} CA\hat{x}(k-1) + CBu^*(k-1) - y(k),$$

and pose the optimization problem

$$\begin{aligned}
&\min_{u^*(k-1)\in\mathbb{R}} \left(\hat{z}^{\mathrm{T}}(k)\hat{z}(k)\right) \\
&= \min_{u^*(k-1)\in\mathbb{R}} \left[(CA\hat{x}(k-1)+CBu^*(k-1)-y(k))^{\mathrm{T}}(CA\hat{x}(k-1)+CBu^*(k-1)-y(k))\right]. \\
&= \min_{u^*(k-1)\in\mathbb{R}} \left[(CBu^*(k-1)-CB\hat{u}(k-1)+z(k))^{\mathrm{T}}(CBu^*(k-1)-CB\hat{u}(k-1)+z(k))\right].
\end{aligned} \tag{2.19}$$

Solving (2.19) yields

$$u^*(k-1) = [(CB)^{\mathrm{T}}CB]^{-1}(CB)^{\mathrm{T}}\left(CBu^-(k-1)-z(k)\right).$$

If the system (2.15) and (2.16) is nonminimum phase, we restrict $u^*(k-1)$ by saturating it or adding an adaptive regularization term in (2.9). In this development, we saturate $u^*(k-1)$. More specifically, let $u^*_{max}, u^*_{min} \in \mathbb{R}$, such that $u^*_{min} < u^*_{max}$, and define

$$u^*(k-1) = \begin{cases} u^*_{max}, & \text{if } u^*(k-1) > u_{max}, \\ u^*_{min}, & \text{if } u^*(k-1) < u_{min}, \\ u^*(k-1), & \text{otherwise.} \end{cases}$$

Now, we update the coefficients $m_i(k)$ and $n_i(k)$ using least squares. Let (2.17) be expressed as

$$\hat{u}(k) = \theta(k)\phi(k-1), \tag{2.20}$$

where

$$\theta(k) \triangleq [m_1(k)\ \cdots\ m_{n_c}(k)\ n_0(k)\ \cdots\ n_{n_c}(k)],$$

and

$$\phi(k-1) \triangleq \begin{bmatrix} \hat{u}(k-1) \\ \vdots \\ \hat{u}(k-n_{\mathrm{c}}) \\ z(k) \\ \vdots \\ z(k-n_{\mathrm{c}}) \end{bmatrix}.$$

Now, define the least squares cost function

$$J_{\mathrm{R}}(\theta(k)) \triangleq \sum_{i=1}^{k} \lambda^{k-i} \|\phi^{\mathrm{T}}(i-2)\theta^{\mathrm{T}}(k-1) - u^*(i-1)\|^2, \tag{2.21}$$

where $\|\cdot\|$ is the Euclidean norm, and $\lambda(k) \in (0,1]$ is the forgetting factor. In this cost function, we use the optimized input $u^*(k-1)$ to update the coefficients $\theta(k)$. The recursive minimizer of (2.21) is given by

$$\begin{aligned} P(k) &= \lambda^{-1}(k)P(k-1) - \lambda^{-1}(k)P(k-1)\phi(k-d-1) \\ &\quad \times [\phi^{\mathrm{T}}(k-d-1)P(k-1)\phi(k-d-1) + \lambda(k)]^{-1} \\ &\quad \times \phi^{\mathrm{T}}(k-d-1)P(k-1), \end{aligned} \tag{2.22}$$

$$\begin{aligned} \theta^{\mathrm{T}}(k) &= \theta^{\mathrm{T}}(k-1) + P(k)\phi(k-d-1) \\ &\quad \times [\phi^{\mathrm{T}}(k-d-1)\theta^{\mathrm{T}}(k-1) - u^{*\mathrm{T}}(k-d)], \end{aligned} \tag{2.23}$$

with the initial conditions $\theta^{\mathrm{T}}(0) \in \mathbb{R}^{2n_c+1}$ and $P(0) = \gamma I$, where $\gamma > 0$.

To summarize, at each step k RCUIO involves the following steps:

1. Use $\hat{x}(k)$ and $\hat{u}(k-1)$ to compute $u^*(k-1)$.

2. Use (2.22) and (2.23) to obtain $\theta(k)$.

3. Use $\theta(k)$ in (2.20) to obtain $\hat{u}(k)$.

4. Use $\hat{u}(k)$ in (2.15) to obtain $\hat{x}(k+1)$.

Example 2.2.5. We consider the same setup in Example 2.2.2, in which a second order, nonminimum-phase system is considered. For this example, we let $n_c = 3$, $u^*_{\min} = -10$, and $u^*_{\max} = 10$. Figure 2.11 shows the true and estimated states, and shows that the estimated states converge to the true states. Figure 2.12 shows the true and estimated inputs for both RCUIO and the direct input reconstruction based estimator with saturation. For this example, the input estimate from the direct input reconstruction based method does not converge to the true input, whereas the input estimate from RCUIO converges to the true input.

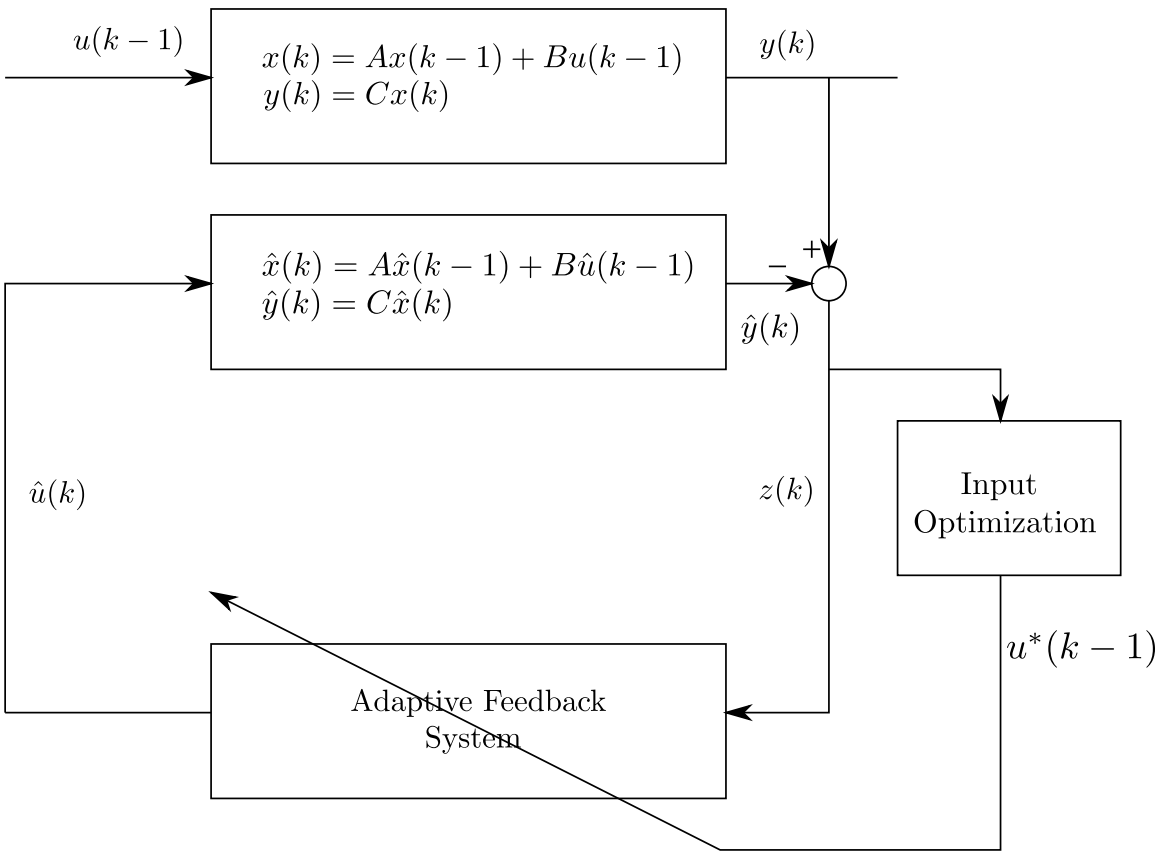
2.3 Problem Formulation : General Linear Systems

In this chapter, we derive RCUIO for linear MIMO systems. Consider the linear-time-invariant system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k), \end{aligned} \tag{2.24}$$

where $x(k) \in \mathbb{R}^n$ is the unknown state, $u(k) \in \mathbb{R}^m$ is an unknown input, and $y(k) \in \mathbb{R}^p$ is the measured output, which is assumed to be bounded. The matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$ are known, and (A, C) is observable. Furthermore, we assume that $u(k)$ is the output of a Lyapunov-stable, linear system, which means that $u(k)$ is a sum of constant and sinusoidal signals.

In order to obtain an estimate $\hat{x}(k) \in \mathbb{R}^n$ of the state $x(k)$, we construct an



At step k:
Known: $\hat{x}(k)$, $y(k)$ up to step k.
1) Obtain $u^*(k-1)$ using input optimization.
2) Use $u^*(k-1)$ to update the adaptive feedback system.
3) Use the adaptive feedback system to obtain $\hat{u}(k)$.
4) Calculate $\hat{x}(k+1) = A\hat{x}(k) + B\hat{u}(k)$.

Figure 2.10: Schematic of RCUIO.

adaptive state estimator of the form

$$\hat{x}(k+1) = A\hat{x}(k) + B\hat{u}(k), \tag{2.25}$$

$$\hat{y}(k) = C\hat{x}(k), \tag{2.26}$$

$$z(k) = \hat{y}(k) - y(k), \tag{2.27}$$

where $\hat{x}(k) \in \mathbb{R}^n$ is the estimated state, $\hat{y}(k) \in \mathbb{R}^p$ is the estimated output, $\hat{u}(k) \in \mathbb{R}^m$

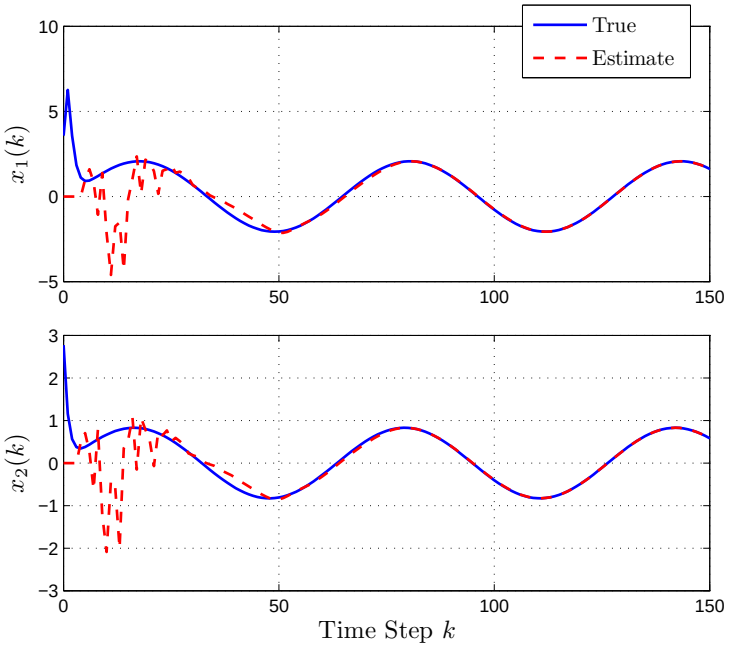


Figure 2.11: State estimates for Example 2.2.5 and the nonminimum phase system (2.13) using RCUIO. This figure shows that $\hat{x}(k)$ converges to $x(k)$.

is the estimator input, and $z(k) \in \mathbb{R}^p$ is the output error. The signal $\hat{u}(k)$ is the output of the strictly proper adaptive subsystem of order n_c, with input $z(k)$, given by

$$\hat{u}(k) = \sum_{i=1}^{n_c} M_i(k)\hat{u}(k-i) + \sum_{i=0}^{n_c} N_i(k)z(k-i), \tag{2.28}$$

where $M_i(k) \in \mathbb{R}^{m\times m}$, $i = 1, \ldots, n_c$, and $N_i(k) \in \mathbb{R}^{m\times p}$, $i = 0, \ldots, n_c$. The goal is to use $z(k)$ to update $M_i(k)$ and $N_i(k)$. Figure 2.13 shows the structure of the adaptive estimator.

For all $i \geq 1$, define the Markov parameter H_i of (A, B, C) by $H_i \triangleq CA^{i-1}B$. Let r be a nonnegative integer. Then, substituting (2.25) into itself $r-1$ times yields, for all $k \geq r$,

$$\hat{x}(k) = A^r\hat{x}(k-r) + \sum_{i=1}^{r} A^{i-1}B\hat{u}(k-i). \tag{2.29}$$

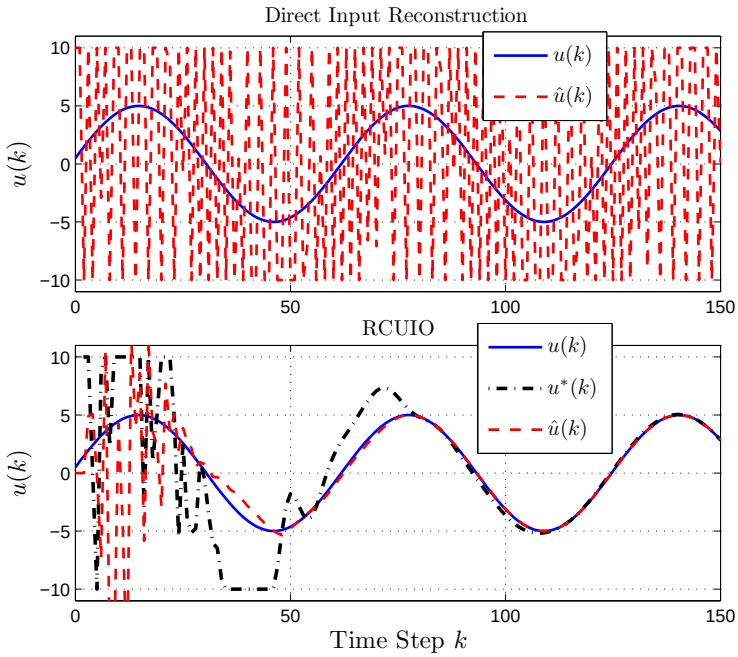


Figure 2.12: State estimates for Example 2.2.5 and the nonminimum phase system (2.13) using RCUIO. This figure shows that $\hat{u}(k)$ converges to $u(k)$.

It follows from (2.24), (2.26), (2.27), and (2.29) that

$$z(k) = CA^r\hat{x}(k-r) - y(k) + \bar{H}\hat{\bar{U}}(k-1), \tag{2.30}$$

where

$$\bar{H} \triangleq \left[\begin{array}{ccc} H_1 & \cdots & H_r \end{array} \right] \in \mathbb{R}^{p\times rm}$$

and

$$\hat{\bar{U}}(k-1) \triangleq \left[\begin{array}{ccc} \hat{u}(k-1) & \cdots & \hat{u}(k-r) \end{array} \right]^{\mathrm{T}}.$$

Next, we rearrange the columns of $\bar{H}$ and the components of $\hat{\bar{U}}(k-1)$ and partition

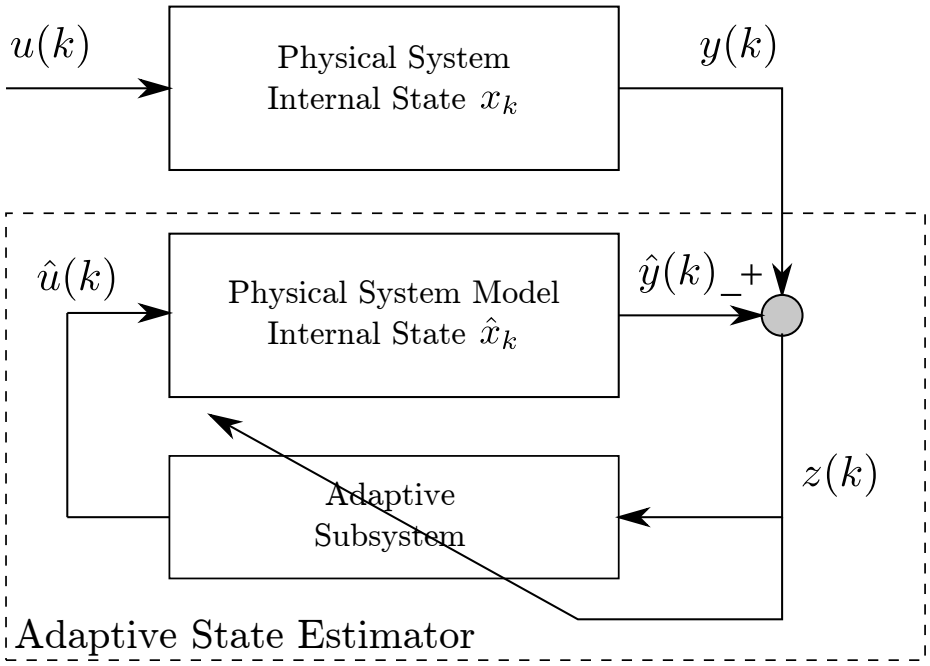


Figure 2.13: Adaptive state estimator structure. The adaptive state estimator uses the error $z(k)$ between the measured output $y(k)$ and the computed output $\hat{y}(k)$ to update the adaptive subsystem in order to obtain an estimate $\hat{u}(k)$ of $u(k)$ that minimizes the retrospective cost $\bar{J}$ defined by (2.43). The estimate $\hat{u}(k)$ asymptotically drives the error between $y(k)$ and $\hat{y}(k)$ to zero. Consequently, the states of the physical system model $\hat{x}(k)$ converge to the physical system states $x(k)$.

the resulting matrix and vector so that

$$\bar{H}\hat{\bar{U}}(k-1) = \mathcal{H}'\hat{U}'(k-1) + \mathcal{H}\hat{U}(k-1), \tag{2.31}$$

where $\mathcal{H}' \in \mathbb{R}^{p\times(rm-l)}$, $\mathcal{H} \in \mathbb{R}^{p\times l}$, $\hat{U}'(k-1) \in \mathbb{R}^{rm-l}$, and $\hat{U}(k-1) \in \mathbb{R}^{l}$. For example, if $\bar{H} = \left[\begin{array}{ccccc} H_1 & H_2 & H_3 & H_4 & H_5 \end{array}\right]$, then $\bar{H}$ can be partitioned as (2.31) with

$$\mathcal{H}' = \left[\begin{array}{ccc} H_1 & H_2 & H_4 \end{array}\right], \quad U'(k-1) = \left[\begin{array}{c} \hat{u}(k-1) \\ \hat{u}(k-2) \\ \hat{u}(k-4) \end{array}\right],$$

and

$$\mathcal{H} = \begin{bmatrix} H_3 & H_5 \end{bmatrix}, \quad \hat{U}(k-1) = \begin{bmatrix} \hat{u}(k-3) \\ \hat{u}(k-5) \end{bmatrix}.$$

Next, using (2.31), we can rewrite (2.30) as

$$z(k) = \mathcal{S}(k) + \mathcal{H}\hat{U}(k-1), \tag{2.32}$$

where

$$\mathcal{S}(k) \triangleq CA^r\hat{x}(k-r) - y(k) + \mathcal{H}'\hat{U}'(k-1). \tag{2.33}$$

Next, for $j = 1, \ldots, s$, we rewrite (2.32) with a delay of k_j time steps, where $0 \leq k_1 \leq k_2 \leq \cdots \leq k_s$, in the form

$$z_j(k-k_j) = \mathcal{S}_j(k-k_j) + \mathcal{H}_j\hat{U}_j(k-k_j-1), \tag{2.34}$$

where (2.33) becomes

$$\mathcal{S}_j(k-k_j) \triangleq CA^r x(k-k_j-r) - y(k-k_j) + \mathcal{H}_j'\hat{U}_j'(k-k_j-1)$$

and (2.31) becomes

$$\bar{H}\hat{\bar{U}}(k-k_j-1) = \mathcal{H}_j'\hat{U}_j'(k-k_j-1) + \mathcal{H}_j\hat{U}_j(k-k_j-1), \tag{2.35}$$

where $\mathcal{H}_j' \in \mathbb{R}^{p\times(rm-l_j)}$, $\mathcal{H}_j \in \mathbb{R}^{p\times l_j}$, $\hat{U}_j'(k-k_j-1) \in \mathbb{R}^{rm-l_j}$, and $\hat{U}_j(k-k_j-1) \in \mathbb{R}^{l_j}$. The subscript j in $\mathcal{H}_j'$, $\mathcal{H}_j$, $\hat{U}_j'$, and $\hat{U}_j$ indicates that the portioning of $\bar{H}\hat{\bar{U}}(k-k_j-1)$ can be different for each delay. Now, by stacking $z(k-k_1), \ldots, z(k-k_s)$, we define

the *extended performance*

$$Z(k) \triangleq \begin{bmatrix} z_1(k-k_1) \\ \vdots \\ z_j(k-k_s) \end{bmatrix} \in \mathbb{R}^{sp}. \tag{2.36}$$

Therefore,

$$Z(k) \triangleq \tilde{\mathcal{S}}(k) + \tilde{\mathcal{H}}\hat{\tilde{U}}(k-1), \tag{2.37}$$

where

$$\tilde{\mathcal{S}}(k) \triangleq \begin{bmatrix} \mathcal{S}_1(k-k_1) \\ \vdots \\ \mathcal{S}_s(k-k_s) \end{bmatrix} \in \mathbb{R}^{sp}$$

and

$$\hat{\tilde{U}}(k-1) \triangleq \begin{bmatrix} \hat{u}(k-q_1) \\ \vdots \\ \hat{u}(k-q_g) \end{bmatrix} \in \mathbb{R}^{mg}, \tag{2.38}$$

where, for $i = 1, \ldots, g$, $k_1 \leq q_i \leq k_s + r$, and $\tilde{\mathcal{H}} \in \mathbb{R}^{sp \times mg}$ is constructed according to the structure of $\hat{\tilde{U}}(k-1)$. The vector $\hat{\tilde{U}}(k-1)$ is formed by stacking $\hat{U}_1(k-k_1-1), \ldots, \hat{U}_s(k-k_s-1)$ and removing copies of repeated components.

For example, with $k_1 = 0$ and $k_2 = 1$, stacking $\hat{U}_1(k-1) = \begin{bmatrix} \hat{u}(k-1) & \hat{u}(k-2) \end{bmatrix}^{\mathrm{T}}$ and $\hat{U}_2(k-2) = \hat{u}(k-2)$ results in $\hat{\tilde{U}}(k-1) = \begin{bmatrix} \hat{u}(k-1) & \hat{u}(k-2) \end{bmatrix}^{\mathrm{T}}$. The coefficient matrix $\tilde{\mathcal{H}}$ consists of the entries of $\mathcal{H}_1, \ldots, \mathcal{H}_s$ arranged according to the structure of $\hat{\tilde{U}}(k-1)$.

Next, we define the *retrospective performance*

$$\hat{z}_j(k-k_j) \triangleq \mathcal{S}_j(k-k_j) + \mathcal{H}_j U_j^*(k-k_j-1), \tag{2.39}$$

where the past inputs $\hat{U}_j(k-k_j-1)$ in (2.34) are replaced by the retrospectively optimized inputs $U_j^*(k-k_j-1)$, which are obtained below using retrospective cost optimization. In analogy with (2.36), the *extended retrospective performance* for (2.39) is defined as

$$\hat{Z}(k) \triangleq \begin{bmatrix} \hat{z}_1(k-k_1) \\ \vdots \\ \hat{z}_s(k-k_s) \end{bmatrix} \in \mathbb{R}^{sp},$$

and thus is given by

$$\hat{Z}(k) = \tilde{\mathcal{S}}(k) + \tilde{\mathcal{H}}\tilde{U}^*(k-1), \tag{2.40}$$

where the components of $\tilde{U}^*(k-1) \in \mathbb{R}^g$ are the components of $U_1^*(k-k_1-1), \ldots, U_s^*(k-k_s-1)$ ordered in the same way as the components of $\hat{\tilde{U}}(k-1)$. Subtracting (2.37) from (2.40) yields

$$\hat{Z}(k) = Z(k) - \tilde{\mathcal{H}}\hat{\tilde{U}}(k-1) + \tilde{\mathcal{H}}\tilde{U}^*(k-1). \tag{2.41}$$

Finally, we define the *retrospective cost function*

$$J(\tilde{U}^*(k-1), k) \triangleq \hat{Z}^{\mathrm{T}}(k) R(k) \hat{Z}(k), \tag{2.42}$$

where $R(k) \in \mathbb{R}^{ps \times ps}$ is a positive-definite performance weighting. The goal is to determine retrospectively optimized inputs $\tilde{U}^*(k-1)$ that would have provided better

performance than the inputs $\hat{\tilde{U}}(k-1)$ that were applied to the estimator system.

2.3.1 Cost Function Optimization with Adaptive Regularization

To ensure that (2.42) has a global minimizer, we consider the regularized cost

$$\bar{J}(\tilde{U}^*(k-1),k) \triangleq \hat{Z}^{\mathrm{T}}(k)R(k)\hat{Z}(k) + \eta(k)\tilde{U}^{*\mathrm{T}}(k-1)\tilde{U}^*(k-1), \tag{2.43}$$

where $\eta(k) \geq 0$. Substituting (2.41) into (2.43) yields

$$\bar{J}(\tilde{U}^*(k-1),k) = \tilde{U}^*(k-1)^{\mathrm{T}}\mathcal{A}(k)\tilde{U}^*(k-1) + \mathcal{B}(k)\tilde{U}^*(k-1) + \mathcal{C}(k),$$

where

$$\begin{aligned}
\mathcal{A}(k) &\triangleq \tilde{\mathcal{H}}^{\mathrm{T}}R(k)\tilde{\mathcal{H}} + \eta(k)I_g,\\
\mathcal{B}(k) &\triangleq 2\tilde{\mathcal{H}}^{\mathrm{T}}R(k)[Z(k) - \tilde{\mathcal{H}}\hat{\tilde{U}}(k-1)],\\
\mathcal{C}(k) &\triangleq Z^{\mathrm{T}}(k)R(k)Z(k) - 2Z^{\mathrm{T}}(k)R(k)\tilde{\mathcal{H}}\hat{\tilde{U}}(k-1) +\\
&\quad \hat{\tilde{U}}^{\mathrm{T}}(k-1)\tilde{\mathcal{H}}^{\mathrm{T}}R(k)\tilde{\mathcal{H}}\hat{\tilde{U}}(k-1).
\end{aligned}$$

If either $\tilde{\mathcal{H}}$ has full column rank or $\eta(k) > 0$, then $\mathcal{A}(k)$ is positive definite. In this case, $\bar{J}(\hat{\tilde{U}}(k-1),k)$ has the unique global minimizer

$$\tilde{U}^*(k-1) = -\frac{1}{2}\mathcal{A}^{-1}(k)\mathcal{B}(k). \tag{2.44}$$

The optimized inputs $\tilde{U}^*(k-1)$ are subsequently used to update the adaptive subsystem (2.28). Note that $\tilde{\mathcal{H}}$ is the only modeling information required by RCUIO.

2.3.2 Adaptive Subsystem Update

The subsystem model (2.28) can be expressed as

$$\hat{u}(k) = \theta(k)\phi(k-1),$$

where

$$\theta(k) \triangleq [M_1(k) \ \cdots \ M_{n_c}(k) \ N_0(k) \ \cdots \ N_{n_c}(k)]$$

and

$$\phi(k-1) \triangleq \begin{bmatrix} \hat{u}(k-1) \\ \vdots \\ \hat{u}(k-n_c) \\ z(k) \\ \vdots \\ z(k-n_c) \end{bmatrix}.$$

Next, let d be a positive integer such that $\tilde{U}^*(k-1)$ contains $u^*(k-d)$. Then, we define the cumulative cost function

$$J_R(\theta(k)) \triangleq \sum_{i=1}^{k} \lambda^{k-i} \|\phi^T(i-d-1)\theta^T(k-1) - u^*(i-d)\|^2, \tag{2.45}$$

where $\|\cdot\|$ is the Euclidean norm, and $\lambda(k) \in (0,1]$ is the forgetting factor. In this cost function, we aim to update the controller gains $\theta(k)$ using the retrospectively optimized input $u^{*T}(k-d)$ and the product $\phi^T(k-d-1)\theta^T(k-1)$. The recursive

minimizer of (2.45) is given by

$$\begin{aligned} P(k) &= \lambda^{-1}(k)P(k-1) - \lambda^{-1}(k)P(k-1)\phi(k-d-1) \\ &\quad \times [\phi^{\mathrm{T}}(k-d-1)P(k-1)\phi(k-d-1) + \lambda(k)]^{-1} \\ &\quad \times \phi^{\mathrm{T}}(k-d-1)P(k-1), \qquad (2.46) \\ \theta^{\mathrm{T}}(k) &= \theta^{\mathrm{T}}(k-1) + P(k)\phi(k-d-1) \\ &\quad \times [\phi^{\mathrm{T}}(k-d-1)\theta^{\mathrm{T}}(k-1) - u^{*\mathrm{T}}(k-d)], \qquad (2.47) \end{aligned}$$

with the initial conditions $\theta^{\mathrm{T}}(0) \in \mathbb{R}^{[n_c(m+p)+1]\times m}$ and $P(0) = \gamma I$, where $\gamma > 0$.

2.4 Computational Complexity

Since RCUIO is an ensemble-free algorithm, it has significantly lower computational complexity than ensemble-based data assimilation algorithms. For implementation, RCUIO requires the computation of equations (2.28), (2.44), (2.46), and (2.47). The computational complexity of the matrix product in (2.28) is $\mathcal{O}(m[n_c m + (n_c+1)p])$, and the computational complexity of the matrix products in (2.46) and (2.47) is $\mathcal{O}((n_c m + (n_c+1)p)^2)$. Finally, the computational complexity of the matrix products and inverse in (2.44) is $\max(\mathcal{O}((mg)^2 sp), \mathcal{O}((sp)^2 mg))$ and $\mathcal{O}((mg)^3)$, respectively.

2.5 Illustrative Examples

The above derivation of RCUIO is based on a linear dynamics model, and the implementation of this algorithm requires the matrix $\tilde{\mathcal{H}}$, whose entries are components of the impulse response. In this section we apply RCUIO to illustrative linear and nonlinear examples, where, in the latter case, we choose $\tilde{\mathcal{H}}$ based on trial and error. For all examples in this section, $x(0) = 0$, and RCUIO is switched on after 80 time steps with $\hat{x}(80) = 0$, where, for all $k < 80$, $\hat{u}(k) = 0$.

Example 2.5.1 (mass-spring-damper system)**.** Consider the discretized linear mass-spring-damper system

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k), \\ x_2(k+1) &= \left(1 - \frac{T_s c}{m}\right) x_2(k) - \frac{T_s k}{m} x_1(k) + \frac{T_s}{m} u(k), \\ y(k) &= x_1(k), \end{aligned}$$

where $T_s = 0.1$ s, $m = 1$ kg, $c = 1$ N s m^{-1}, $k = 1$ N m^{-1}, and the input $u(k) = 10\sin(0.01k) + \sin(0.1k)$ N is assumed to be unknown. The adaptive estimator has the form

$$\begin{aligned} \hat{x}_1(k+1) &= \hat{x}_1(k) + T_s \hat{x}_2(k), \\ \hat{x}_2(k+1) &= \left(1 - \frac{T_s c}{m}\right) \hat{x}_2(k) - \frac{T_s k}{m} \hat{x}_1(k) + \frac{T_s}{m} \hat{u}(k), \\ \hat{y}(k) &= \hat{x}_1(k). \end{aligned}$$

Since $H_1 = 0$, we use

$$\tilde{\mathcal{H}} = \begin{bmatrix} H_3 \\ H_2 \end{bmatrix} = \begin{bmatrix} 0.019 \\ 0.01 \end{bmatrix}.$$

Then, using (2.37), we have

$$Z(k) = \tilde{\mathcal{S}}(k) + \tilde{\mathcal{H}} \hat{\tilde{U}}(k-1),$$

where

$$Z(k) = \begin{bmatrix} z(k) \\ z(k-1) \end{bmatrix},$$

$$\mathcal{S}(k) = \begin{bmatrix} \mathcal{S}_1(k) \\ \mathcal{S}_2(k-1) \end{bmatrix},$$

and

$$\hat{\hat{U}}(k-1) = \begin{bmatrix} \hat{u}(k-3) \end{bmatrix}.$$

We use (2.44) to obtain the retrospectively optimized input $\hat{u}^*(k-3)$, which is then used in (2.46) and (2.47) to update the coefficients of the adaptive subsystem (2.28). The updated adaptive subsystem (2.28) is then used to obtain the next input $\hat{u}(k)$. For this example, $R(k) \equiv 1$, $\eta = 0$, $n_c = 10$, and $\gamma = 100$. Figure 2.14 shows the true and estimated states and input, and Figure 2.15 shows $|z(k)|$. Figure 2.14 shows that, after RCUIO is turned on at $k = 80$, RCUIO reduces the error in $\hat{x}_1(k)$, $\hat{x}_2(k)$, and $\hat{u}(k)$ by approximately two orders of magnitude. Finally, Figure 2.15 shows that, after $k = 80$, RCUIO reduces the error in $|z(k)|$ by approximately two orders of magnitude.

Example 2.5.2 (Van der Pol oscillator)**.** Consider the nonlinear discretized Van der Pol oscillator

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k), \\ x_2(k+1) &= x_2(k) + T_s \left(1 - x_1(k)^2\right) x_2(k) - T_s x_1(k) + T_s u(k), \\ y(k) &= x_1(k) + 0.2 x_2(k), \end{aligned}$$

where $T_s = 0.1$ s and the unknown input $u(k) = \sin(0.01k)$. Let $R(k) \equiv 1$, $\tilde{\mathcal{H}} = 0.03$, $\eta = 0.001$, $n_c = 4$, $\gamma = 200$, where $\tilde{\mathcal{H}}$ is chosen by trial and error. Figure 2.16 shows the true and estimated states and input. Figure 2.16 shows that, after RCUIO is turned on at $k = 80$, RCUIO reduces the error in $\hat{x}_1(k)$, $\hat{x}_2(k)$, $\hat{u}(k)$ by approximately three orders of magnitude.

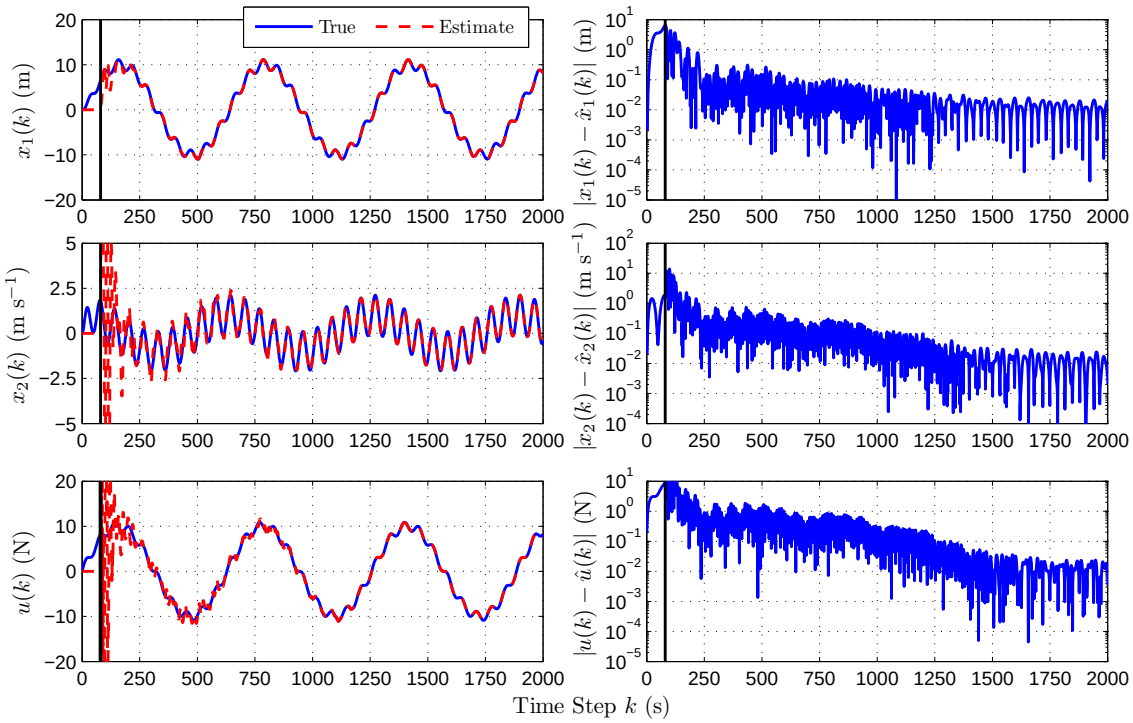


Figure 2.14: True and estimated position $x_1(k)$, velocity $x_2(k)$, and input $u(k)$ for the mass-spring-damper system. The vertical black line indicates that RCUIO is turned on at $k = 80$. After $k = 80$, RCUIO reduces the error in $\hat{x}_1(k)$, $\hat{x}_2(k)$, and $\hat{u}(k)$ by approximately two orders of magnitude.

Example 2.5.3 (Van der Pol oscillator with matched unmodeled dynamics)**.** Consider the modified discretized Van der Pol oscillator

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k), \\ x_2(k+1) &= x_2(k) + T_s \left(1 - x_1(k)^2\right) x_2(k) - T_s x_1(k) + \\ &\qquad \sin(x_2(k)) + T_s u(k), \\ y(k) &= x_1(k) + 0.2 x_2(k), \end{aligned}$$

where $\sin(x_2(k))$ is an unmodeled feature of the dynamics, $T_s = 0.1$ s, and the input $u(k) = \sin(0.01k)$ is unknown. Note that the unmodeled term $\sin(x_2(k))$ is matched to the unknown input in the sense that there exists an input signal that can account

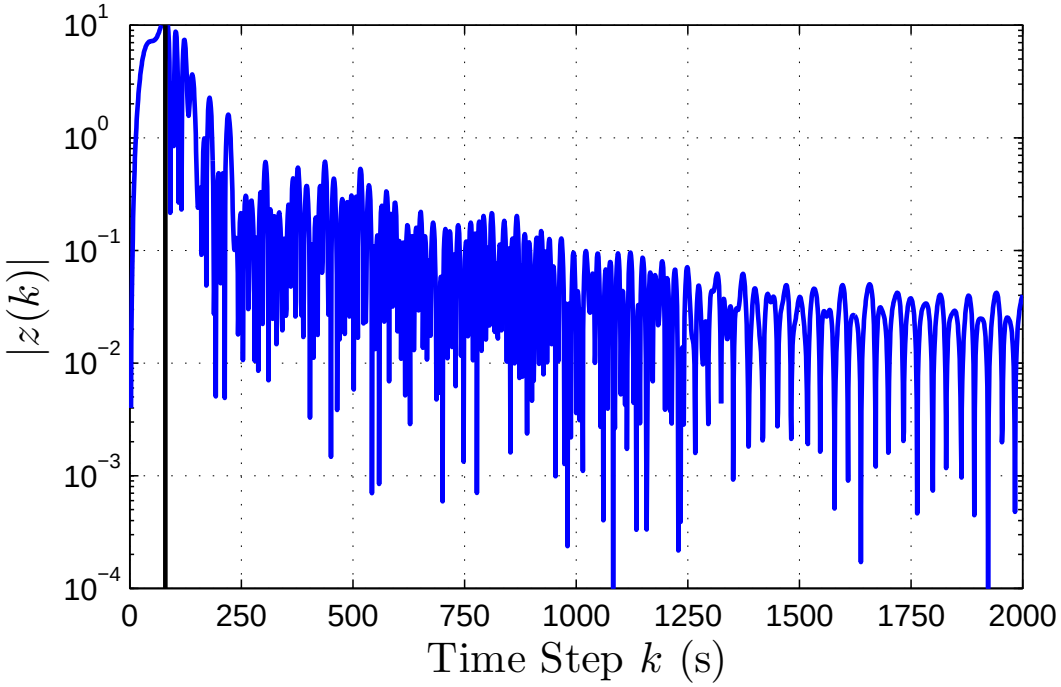


Figure 2.15: Error $|z(k)|$ for the mass-spring-damper example. After RCUIO is turned on at $k = 80$, RCUIO reduces the error in $|z(k)|$ by approximately two orders of magnitude.

for the presence of this term. In particular, replacing $u(k)$ by $u(k) - \sin(x_2(k))/T_s$ effectively removes the unmodeled term. Since the term $\sin(x_2(k))$ is unmodeled, the estimator has the form

$$\begin{aligned}
\hat{x}_1(k+1) &= \hat{x}_1(k) + T_s\hat{x}_2(k), \\
\hat{x}_2(k+1) &= \hat{x}_2(k) + T_s\left(1 - \hat{x}_1(k)^2\right)\hat{x}_2(k) - T_s\hat{x}_1(k) + T_s\hat{u}(k), \\
\hat{y}(k) &= \hat{x}_1(k) + 0.2\hat{x}_2(k).
\end{aligned}$$

Let $R(k) \equiv 1$, $\tilde{\mathcal{H}} = 0.008$, $\eta = 0.0008$, $n_c = 4$, and $\gamma = 1000$. Figure 2.17 shows the true and estimated states and input. Figure 2.17 shows that, after RCUIO is turned on at $k = 80$, RCUIO reduces the error in $\hat{x}_1(k)$ and $\hat{x}_2(k)$ by one order of magnitude. For this example, the estimated input does not converge to the true input due to the

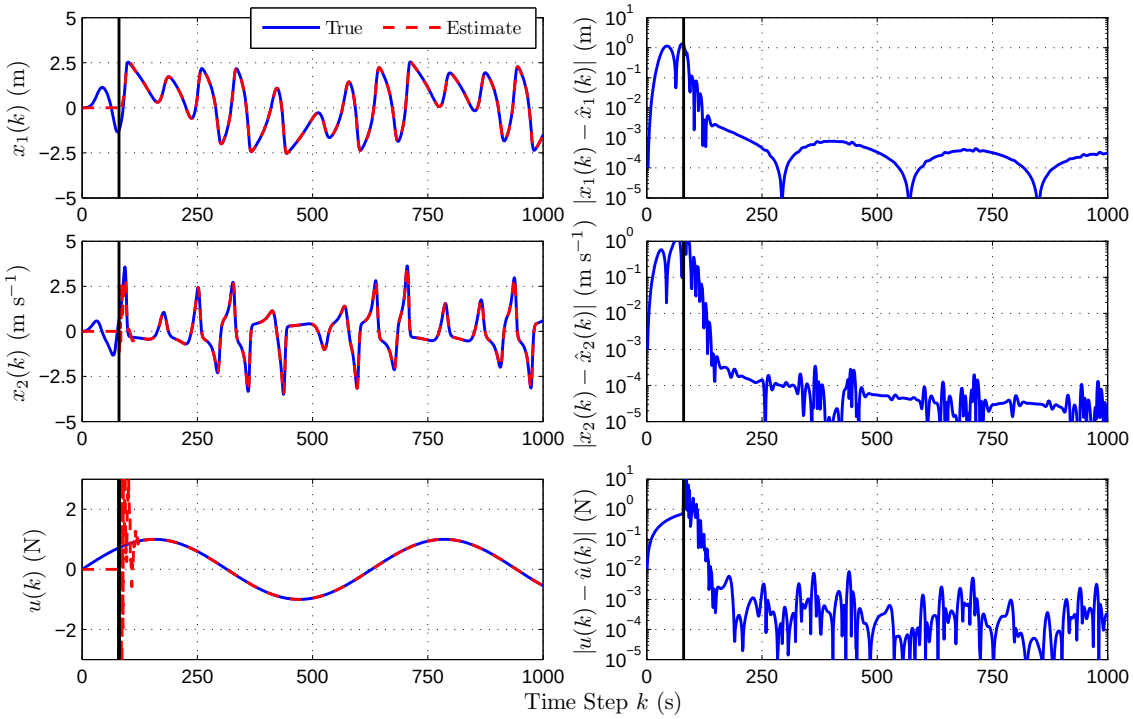


Figure 2.16: True and estimated position $x_1(k)$, velocity $x_2(k)$, and input $u(k)$ for the Van der Pol oscillator. The vertical black line indicates that RCUIO is turned on at $k = 80$. After $k = 80$, RCUIO reduces the error in $\hat{x}_1(k)$, $\hat{x}_2(k)$, and $\hat{u}(k)$ by approximately three orders of magnitude.

matched unmodeled term $\sin(x_2(k))$ in the dynamics of the oscillator. In fact, the estimated input $\hat{u}(k)$ converges to $u(k) - \sin(x_2(k))/T_s$, which shows that RCUIO reconstructs not only the unknown input but also the matched unmodeled feature of the dynamics.

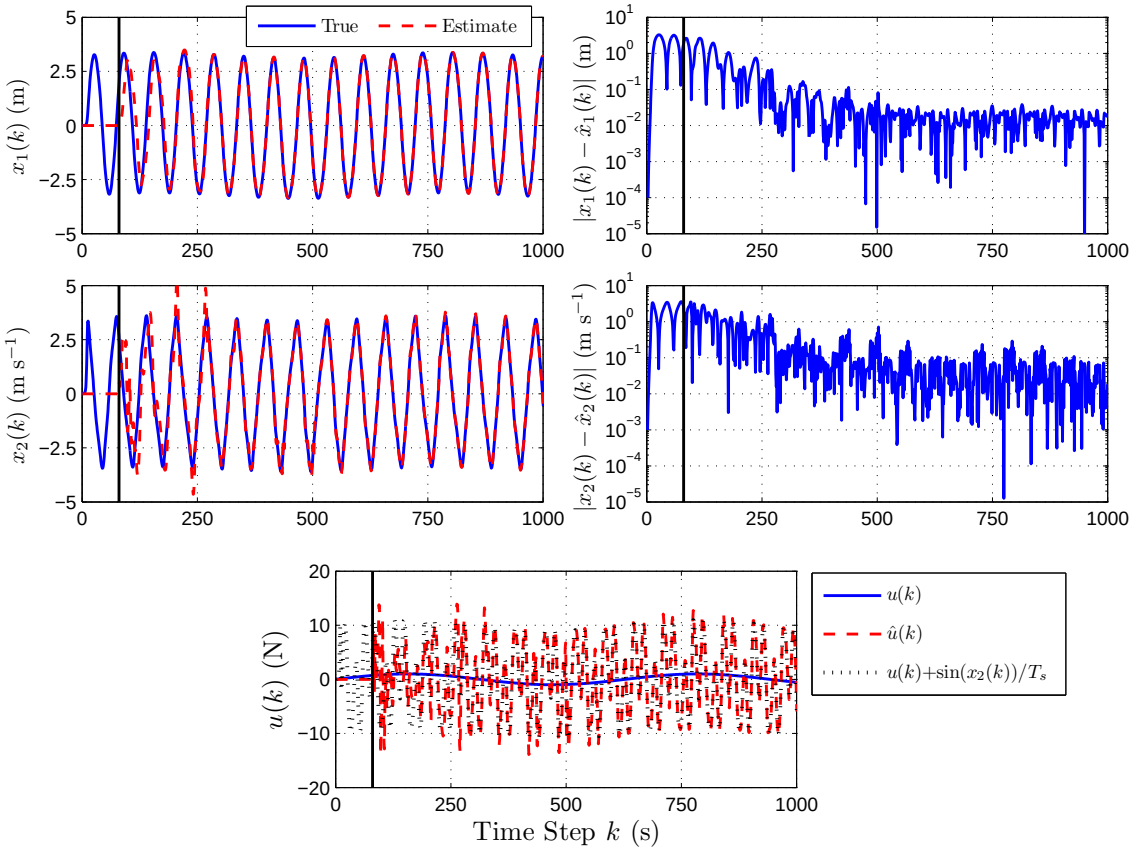


Figure 2.17: True and estimated position $x_1(k)$, velocity $x_2(k)$, and input $u(k)$ for the Van der Pol oscillator with matched unmodeled dynamics. The vertical black line indicates that RCUIO is turned on at $k = 80$. After $k = 80$, RCUIO reduces the errors in $\hat{x}_1(k)$ and $\hat{x}_2(k)$ by approximately one order of magnitude. For this example, the estimated input does not converge to the true input because of the unknown term $\sin(x_2(k))$ in the dynamics of the oscillator. However, $\hat{u}(k)$ does give an estimate of $u(k) - \sin(x_2(k))/T_s$, which shows that RCUIO reconstructs not only the unknown input but also the matched unmodeled feature of the dynamics.

Example 2.5.4 (Van der Pol oscillator with unmatched unmodeled dynamics)**.** Consider the modified discretized Van der Pol oscillator

$$\begin{aligned} x_1(k+1) &= x_1(k) + T_s x_2(k) + 0.1\sin(x_2(k)), \\ x_2(k+1) &= x_2(k) + T_s\left(1 - x_1(k)^2\right)x_2(k) - T_s x_1(k) + T_s u(k), \\ y(k) &= x_1(k) + 0.2x_2(k), \end{aligned}$$

where $0.1\sin(x_2(k))$ is an unmodeled feature of the dynamics, $T_s = 0.1$ s, and the input $u(k) = \sin(0.01k)$ is unknown. Note that, unlike Example 3.3, the unmodeled term $0.1\sin(x_2(k))$ is not matched to the unknown input, and thus there does not exist an input signal that can account for the presence of this term. Since the term $0.1\sin(x_2(k))$ is unmodeled, the estimator has the form

$$\begin{aligned} \hat{x}_1(k+1) &= \hat{x}_1(k) + T_s\hat{x}_2(k), \\ \hat{x}_2(k+1) &= \hat{x}_2(k) + T_s\left(1 - \hat{x}_1(k)^2\right)\hat{x}_2(k) - T_s\hat{x}_1(k) + T_s\hat{u}(k), \\ \hat{y}(k) &= \hat{x}_1(k) + 0.2\hat{x}_2(k). \end{aligned}$$

Let $R(k) \equiv 1$, $\tilde{\mathcal{H}} = 0.008$, $\eta = 0.0008$, $n_c = 4$, and $\gamma = 1000$. Figure 2.18 shows that there is a persistent error in the estimated states and that $\hat{u}(k)$ cannot estimate $u(k)$ due to the unmatched unmodeled feature of the dynamics.

2.6 Conclusions

In this chapter, we developed RCUIO for the special case where the first Markov parameter is nonzero and a single past input is reconstructed, and the general case. RCUIO was demonstrated on minimum-phase, nonminimum-phase, and nonlinear systems. For the special case where the first Markov parameter is non-zero, we showed

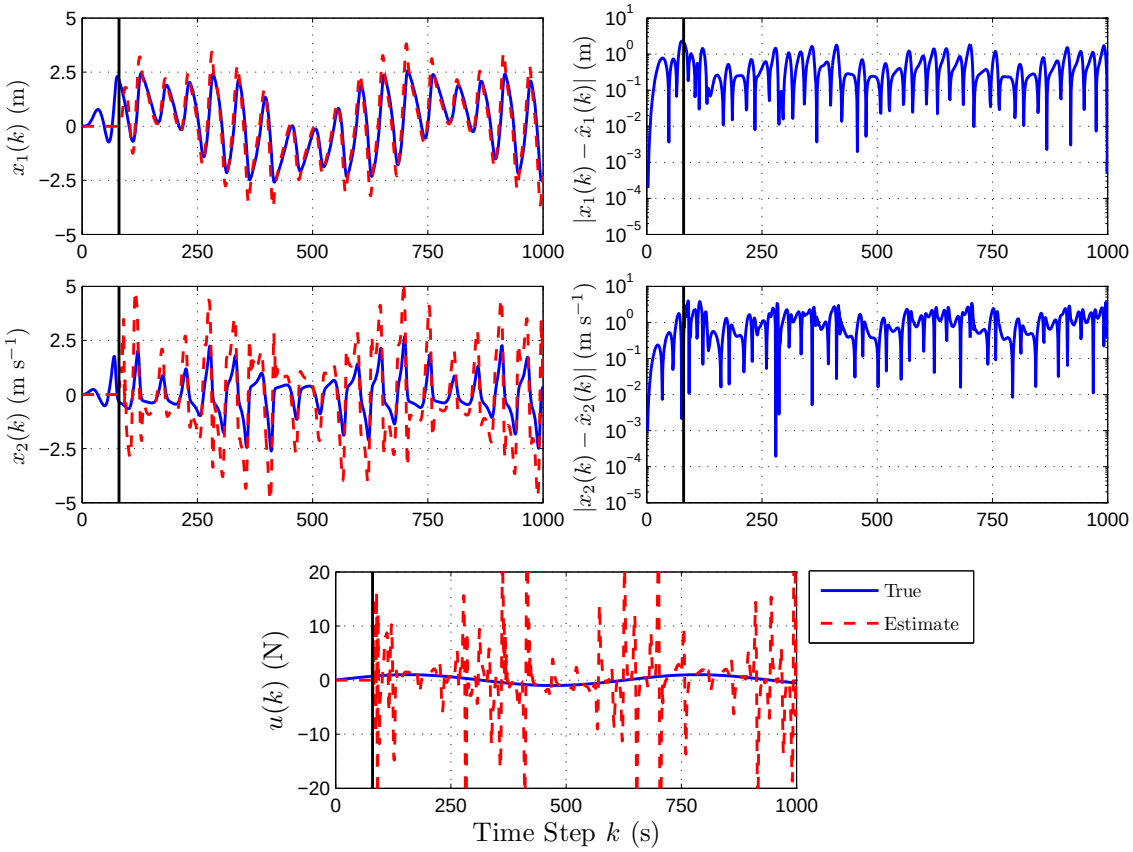


Figure 2.18: True and estimated position $x_1(k)$, velocity $x_2(k)$, and input $u(k)$ for the Van der Pol oscillator with unmatched unmodeled dynamics. The persistent errors in the estimates of $x_1(k)$ and $x_2(k)$ are due to the unmatched unmodeled term $0.1\sin(x_2(k))$, which cannot be estimated by the adaptive estimator. Furthermore, the estimated input does not converge to the true input because of the unmatched unmodeled term $0.1\sin(x_2(k))$ in the dynamics of the oscillator.

the connection between RCUIO and direct input reconstruction based estimators, and the use of adaptive feedback systems to handle nonminimum-phase systems.

CHAPTER 3

Sliding Window Variable Regularization Recursive Least Squares

In this chapter, we develop the sliding window, variable regularization recursive least squares (SW-VR-RLS) algorithm and analyze its convergence properties, numerical stability, and computational complexity. Finally, we compare SW-VR-RLS with the Proportionate Affine Projection algorithm and the Proportionate Normalized Least Mean Squares algorithm. In Section 3.2, we give the cost function and the non-recursive solution. In Section 3.3, we derive the Sliding Window Variable Regularization Recursive Least Squares Algorithms and, in Section 3.4, we give their computational complexity. Section gives convergence properties of SW-VR-RLS, and Section 3.6 gives simulation results and a numerical stability analysis.

3.1 Introduction

Recursive-least-squares (RLS) and gradient-based algorithms are widely used in signal processing, estimation, identification, and control [43–51]. Under ideal conditions, that is, noiseless measurements and persistency of the data, these techniques are guaranteed to converge to the minimizer of a quadratic function [44, 47]. In practice, the accuracy of the estimates depends on the level of noise and the persistency of the data.

The standard RLS algorithm operates on a growing window of data, where new data are added to the RLS cost function as they become available and past data are progressively discounted through the use of a forgetting factor. In contrast, sliding-window RLS algorithms [52–56] require no forgetting factor since they operate on a finite data window of fixed length, where new data replace past data in the RLS cost function. Sliding-window least-squares techniques are available in both batch and recursive formulations. As shown in [53], sliding-window RLS algorithms have enhanced tracking performance compared to standard RLS algorithms in the presence of time-varying parameters.

In standard RLS, the positive-definite initialization of the covariance matrix is the inverse of the weighting on a regularization term in a quadratic cost function. This regularization term compensates for the potential lack of persistency, ensuring that the cost function has a unique minimizer at each step. Traditionally, the regularization term is fixed for all steps of the recursion. Additionally, an optimally regularized adaptive filtering algorithm with constant regularization is presented in [57]. However, variants of RLS with time-varying regularization have been developed in the context of adaptive filtering, echo cancellation, and affine projection [58–63].

In the present work, we derive a novel sliding-window variable-regularization RLS (SW-VR-RLS) algorithm, where the weighting on the regularization term can change at each step. An additional extension presented in this chapter also involves the regularization term. Specifically, the regularization term in standard RLS weights the difference between the next estimate and the initial estimate, while the regularization term in sliding-window RLS weights the difference between the next estimate and the estimate at the beginning of the sliding window. In this work, the regularization term weights the difference between the next estimate and an arbitrarily chosen time-varying vector. As a special case, the time-varying vector can be the current estimate or a recent estimate. These variable-regularization extensions of sliding-window RLS

can facilitate tradeoffs among transient error, rate of convergence, and steady-state error.

In this work, we derive the SW-VR-RLS equations and analyze their convergence properties in the absence of noise. While standard RLS entails the update of the estimate and the covariance matrix, sliding-window RLS involves the update of an additional symmetric matrix of size $n \times n$, where n is the dimension of the estimate. Furthermore, SW-VR-RLS requires updating of one more symmetric matrix of size $n \times n$ to account for the time-varying regularization.

The SW-VR-RLS algorithm was first presented in [64] together with a preliminary numerical study and without convergence analysis. In addition, a growing-window RLS algorithm with time-varying regularization appears in [65]. The goal of this work is to provide a more complete development of the SW-VR-RLS algorithm, including an analysis of convergence and numerical stability.

In this chapter, a matrix $A \in \mathbb{R}^{n \times n}$ is positive semidefinite ($A \geq 0$) if it is symmetric and has nonnegative eigenvalues. Furthermore, $A \in \mathbb{R}^{n \times n}$ is positive definite ($A > 0$) if it is symmetric and has positive eigenvalues.

3.2 The Non-Recursive Solution

Let r be a nonnegative integer. For all integers $i \geq -r$, let α_i, $b_i \in \mathbb{R}^n$ and $A_i \in \mathbb{R}^{n \times n}$, where A_i is positive semidefinite. For all $i \geq 0$, let $R_i \in \mathbb{R}^{n \times n}$, where R_i is positive semidefinite. Assume that, for all $k \geq 0$, $\sum_{i=k-r}^{k-1} A_i + R_k$ is positive definite. In practice, the matrix A_k depends on data, whereas R_k is chosen by the user. For all $k \geq 0$, the sliding-window, variable-regularization quadratic cost is defined by

$$J_k(x) \triangleq \sum_{i=k-r}^{k} \left(x^{\mathrm{T}} A_i x + b_i^{\mathrm{T}} x \right) + (x - \alpha_k)^{\mathrm{T}} R_k (x - \alpha_k), \tag{3.1}$$

where $x \in \mathbb{R}^n$ and $x_0 = -\frac{1}{2}\left(\sum_{i=-r}^{0} A_i + R_0\right)^{-1}\left(\sum_{i=-r}^{0} b_i - 2R_0\alpha_0\right)$ is the minimizer of $J_0(x)$. For all $k \geq 0$, the unique minimizer x_k of (3.1) is

$$x_k = -\tfrac{1}{2}\left(\sum_{i=k-r}^{k} A_i + R_k\right)^{-1}\left(\sum_{i=k-r}^{k} b_i - 2R_k\alpha_k\right). \tag{3.2}$$

Example 3.2.1. Consider the weighted regularized least-squares cost function

$$\mathcal{J}_k(x) \triangleq \sum_{i=k-r}^{k} (y_i - F_i^{\mathrm{T}}x)^{\mathrm{T}} W_i (y_i - F_i^{\mathrm{T}}x) + (x - \alpha_k)^{\mathrm{T}} R_k (x - \alpha_k),$$

where $x \in \mathbb{R}^n$. Let r be a nonnegative integer, and, for all $i \geq -r$, let $y_i \in \mathbb{R}^l$, $\alpha_i \in \mathbb{R}^n$, $F_i \in \mathbb{R}^{n\times l}$, $R_i \in \mathbb{R}^{n\times n}$, and $W_i \in \mathbb{R}^{l\times l}$, where W_i is positive definite. Furthermore, for all $i \geq -r$, define $A_i \triangleq F_i W_i F_i^{\mathrm{T}}$ and $b_i \triangleq -2F_i W_i y_i$. Then, for all $k \geq 0$ and $x \in \mathbb{R}^n$, $\mathcal{J}_k(x) = J_k(x) + \sum_{i=k-r}^{k} y_i^{\mathrm{T}} W_i y_i$. Thus, the minimizer of $J_k(x)$ is also the minimizer of $\mathcal{J}_k(x)$. Moreover, it follows from (4.2) that the minimizer of $\mathcal{J}_k(x)$ is given by

$$x_k = \left(\sum_{i=k-r}^{k} F_i W_i F_i^{\mathrm{T}} + R_k\right)^{-1}\left(\sum_{i=k-r}^{k} F_i W_i y_i + R_k\alpha_k\right).$$

Example 3.2.2. Let n and r be positive integers, for $i \in \{1, \ldots, n\}$, let a_i, $c_i \in \mathbb{R}$, and, for all $i \geq -r - n$, let u_i, $y_i \in \mathbb{R}$. Furthermore, for all $k \geq 0$, let y_k satisfy the infinite impulse response model

$$y_k = \sum_{i=1}^{n} a_i y_{k-i} + \sum_{i=1}^{n} c_i u_{k-i}. \tag{3.3}$$

Next, for all $i \geq -r$, define $\psi_i \triangleq \begin{bmatrix} u_{i-1} & \cdots & u_{i-n} & y_{i-1} & \cdots & y_{i-n} \end{bmatrix}^{\mathrm{T}}$. Furthermore, for all $i \geq -r$, define $A_i \triangleq \psi_i \psi_i^{\mathrm{T}}$, $b_i \triangleq -2y_i\psi_i$. Finally, for all $k \geq 0$, let $R_k \in \mathbb{R}^{2n\times 2n}$ and $\alpha_k \in \mathbb{R}^{2n}$, and define $x_* \triangleq \begin{bmatrix} a_1 & \cdots & a_n & c_1 & \cdots & c_n \end{bmatrix}^{\mathrm{T}}$. The objective is to

choose R_k and α_k such that the sequence of minimizers $\{x_k\}_{k=0}^{\infty}$ of (3.1) converges to x_*. Note that, for all $k \geq -r$, $\mathrm{rank}(A_k) \leq 1$. As shown in Section IV, the rank of A_k affects the computational complexity of the recursive formulation of (4.2).

3.3 The SW-VR-RLS Solution

Defining

$$P_k \triangleq \left(\sum_{i=k-r}^{k} A_i + R_k \right)^{-1}, \tag{3.4}$$

(4.2) can be written as

$$x_k = -\tfrac{1}{2} P_k \left(\sum_{i=k-r}^{k} b_i - 2R_k \alpha_k \right). \tag{3.5}$$

To express P_k recursively, consider the decomposition

$$A_k = \psi_k \psi_k^{\mathrm{T}}, \tag{3.6}$$

where $\psi_k \in \mathbb{R}^{n \times n_k}$ and $n_k \triangleq \mathrm{rank}(A_k)$. Next, for all $k \geq 1$, define

$$Q_k \triangleq \left(\sum_{i=k-r}^{k-1} A_i + R_k \right)^{-1} = \left(P_k^{-1} - A_k \right)^{-1}. \tag{3.7}$$

It follows from (3.6) and (3.7) that

$$P_k = \left(Q_k^{-1} + \psi_k \psi_k^{\mathrm{T}} \right)^{-1}.$$

Using the matrix inversion lemma

$$(X + UCV)^{-1} = X^{-1} - X^{-1} U \left(C^{-1} + V X^{-1} U \right)^{-1} V X^{-1}, \tag{3.8}$$

with $X \triangleq Q_k^{-1}$, $U \triangleq \psi_k$, $C \triangleq I_{n_k}$, where I_{n_k} is the $n_k \times n_k$ identity matrix, and $V \triangleq \psi_k^{\mathrm{T}}$, it follows that

$$P_k = Q_k - Q_k\psi_k \left(I_{n_k} + \psi_k^{\mathrm{T}} Q_k \psi_k\right)^{-1} \psi_k^{\mathrm{T}} Q_k.$$

To express Q_k recursively, for all $k \geq 1$, define

$$L_k \triangleq \left(\sum_{i=k-r-1}^{k-1} A_i + R_k\right)^{-1} = \left(Q_k^{-1} - A_{k-r-1}\right)^{-1} = \left(Q_k^{-1} - \psi_{k-r-1}\psi_{k-r-1}^{\mathrm{T}}\right)^{-1}. \quad (3.9)$$

Using (3.8) with $X \triangleq L_k^{-1}$, $U \triangleq \psi_{k-r-1}$, $C \triangleq -I_{n_{k-r-1}}$, and $V \triangleq \psi_{k-r-1}^{\mathrm{T}}$, it follows that

$$Q_k = L_k - L_k\psi_{k-r-1} \left(-I_{n_{k-r-1}} + \psi_{k-r-1}^{\mathrm{T}} L_k \psi_{k-r-1}\right)^{-1} \psi_{k-r-1}^{\mathrm{T}} L_k.$$

To express L_k recursively, we substitute (3.4) into itself to obtain

$$P_k^{-1} = \sum_{i=k-r}^{k} A_i + R_k = P_{k-1}^{-1} + A_k - A_{k-r-1} + R_k - R_{k-1}. \quad (3.10)$$

Thus, it follows from (3.7), (3.9), and (3.10) that

$$L_k = \left(P_{k-1}^{-1} + R_k - R_{k-1}\right)^{-1}. \quad (3.11)$$

Next, we factor $R_k - R_{k-1}$ as

$$R_k - R_{k-1} = \phi_k S_k \phi_k^{\mathrm{T}}, \quad (3.12)$$

where $\phi_k \in \mathbb{R}^{n \times m_k}$, $m_k \triangleq \operatorname{rank}(R_k - R_{k-1})$, and $S_k \in \mathbb{R}^{m_k \times m_k}$ has the form $S_k \triangleq \operatorname{diag}(\pm 1, \ldots, \pm 1)$. Using (3.8) with $X \triangleq P_{k-1}^{-1}$, $U \triangleq \phi_k$, $C \triangleq S_k$, and $V \triangleq \phi_k^{\mathrm{T}}$, it

follows from (4.7) that

$$L_k = P_{k-1} - P_{k-1}\phi_k \left(S_k + \phi_k^{\mathrm{T}} P_{k-1}\phi_k\right)^{-1} \phi_k^{\mathrm{T}} P_{k-1}.$$

We now summarize the SW-VR-RLS algorithm

Algorithm 1. For each $k \geq 1$, the unique minimizer x_k of (3.1) is given by

$$L_k = P_{k-1} - P_{k-1}\phi_k \left(S_k + \phi_k^{\mathrm{T}} P_{k-1}\phi_k\right)^{-1} \phi_k^{\mathrm{T}} P_{k-1}, \tag{3.13}$$

$$Q_k = L_k - L_k\psi_{k-r-1} \left(-I_{n_{k-r-1}} + \psi_{k-r-1}^{\mathrm{T}} L_k \psi_{k-r-1}\right)^{-1} \psi_{k-r-1}^{\mathrm{T}} L_k, \tag{3.14}$$

$$P_k = Q_k - Q_k\psi_k \left(I_{n_k} + \psi_k^{\mathrm{T}} Q_k \psi_k\right)^{-1} \psi_k^{\mathrm{T}} Q_k, \tag{3.15}$$

$$x_k = -\tfrac{1}{2} P_k \left(\sum_{i=k-r}^{k} b_i - 2R_k\alpha_k\right), \tag{3.16}$$

where $P_0 = \left(\sum_{i=-r}^{0} A_i + R_0\right)^{-1}$ and $x_0 = -\frac{1}{2} P_0 \left(\sum_{i=-r}^{0} b_i - 2R_0\alpha_0\right)$.

As an alternative to Algorithm 1, the equation for x_k can be expressed using the recursion matrix P_k. First, it follows from (3.16) that

$$\sum_{i=k-r-1}^{k-1} b_i = -2P_{k-1}^{-1} x_{k-1} + 2R_{k-1}\alpha_{k-1}. \tag{3.17}$$

Using (3.10) and (4.10), it follows that (3.16) can be written as

$$\begin{aligned}
x_k &= -\tfrac{1}{2} P_k \left(\sum_{i=k-r-1}^{k-1} b_i + b_k - b_{k-r-1} - 2R_k\alpha_k\right) \\
&= -\tfrac{1}{2} P_k \left(-2P_{k-1}^{-1} x_{k-1} + 2R_{k-1}\alpha_{k-1} + b_k - b_{k-r-1} - 2R_k\alpha_k\right) \\
&= -\tfrac{1}{2} P_k \left[-2(P_k^{-1} - A_k + A_{k-r-1} - R_k + R_{k-1})x_{k-1} + 2R_{k-1}\alpha_{k-1} + b_k - b_{k-r-1} - 2R_k\alpha_k\right] \\
&= x_{k-1} - P_k \left[(A_k - A_{k-r-1})x_{k-1} + (R_k - R_{k-1})x_{k-1} + R_{k-1}\alpha_{k-1} + \tfrac{1}{2}(b_k - b_{k-r-1}) - R_k\alpha_k\right].
\end{aligned}$$

We now summarize the alternative SW-VR-RLS algorithm.

Algorithm 2. For each $k \geq 1$, the unique minimizer x_k of (3.1) is given by

$$L_k = P_{k-1} - P_{k-1}\phi_k \left(S_k + \phi_k^{\mathrm{T}} P_{k-1}\phi_k\right)^{-1} \phi_k^{\mathrm{T}} P_{k-1}, \tag{3.18}$$

$$Q_k = L_k - L_k\psi_{k-r-1} \left(-I_{n_{k-r-1}} + \psi_{k-r-1}^{\mathrm{T}} L_k \psi_{k-r-1}\right)^{-1} \psi_{k-r-1}^{\mathrm{T}} L_k, \tag{3.19}$$

$$P_k = Q_k - Q_k\psi_k \left(I_{n_k} + \psi_k^{\mathrm{T}} Q_k \psi_k\right)^{-1} \psi_k^{\mathrm{T}} Q_k, \tag{3.20}$$

$$\begin{aligned} x_k = x_{k-1} &- P_k \left[(A_k - A_{k-r-1})x_{k-1} + \tfrac{1}{2}(b_k - b_{k-r-1})\right] \\ &- P_k \left[(R_k - R_{k-1})x_{k-1} + R_{k-1}\alpha_{k-1} - R_k\alpha_k\right], \end{aligned} \tag{3.21}$$

where $P_0 = \left(\sum_{i=-r}^{0} A_i + R_0\right)^{-1}$ and $x_0 = -\frac{1}{2}P_0 \left(\sum_{i=-r}^{0} b_i - 2R_0\alpha_0\right)$.

The theoretical properties and computational complexity of Algorithm 1 and 2 are identical, but their numerical properties are different, which will be discussed in Section 3.7.

If, for all $i \in \{-r, \ldots, 0\}$, $A_i = 0$ and $b_i = 0$, then $x_0 = \alpha_0$ and $P_0 = R_0^{-1}$. Furthermore, if the regularization weighting R_k is constant, that is, for all $k \geq 0$, $R_k = R_0 > 0$, then (3.12) implies that $\phi_k = 0$ and (3.18) simplifies to $L_k = P_{k-1}$, and thus computation of L_k is not required.

3.4 Computational Complexity

First, consider Algorithm 1. The computational complexity of the matrix products and inverse in (3.13) are $\mathcal{O}(n^2 m_k)$ and $\mathcal{O}(m_k^3)$, respectively, where $m_k = \operatorname{rank}(R_k - R_{k-1}) \leq n$. Hence, (3.13) is $\mathcal{O}(n^2 m_k)$. In particular, if, for all $k \geq 0$, $m_k = 1$, then the inverse in (3.13) is a scalar inverse, and (3.13) is $\mathcal{O}(n^2)$.

The matrix products and inverse in (3.15) are $\mathcal{O}(n^2 n_k)$ and $\mathcal{O}(n_k^3)$, respectively, where $n_k = \operatorname{rank}(A_k) \leq n$. Hence, (3.15) is $\mathcal{O}(n^2 n_k)$. Similarly, (3.14) is $\mathcal{O}(n^2 n_{k-r-1})$. In particular, if, for all $k \geq 0$, $n_k = 1$, then the inverses in (3.14) and (3.15) are scalar inverses, and (3.14) and (3.15) are $\mathcal{O}(n^2)$.

Finally, note that (3.16) is $\mathcal{O}(n^2)$. Therefore, if for all $k \geq 0$, $\text{rank}(R_k - R_{k-1}) = 1$ and $\text{rank}(A_k) = 1$, then the computational complexity of Algorithm 1 is $\mathcal{O}(n^2)$.

Now, consider Algorithm 2. Since (3.18), (3.19), and (3.20) are identical to (3.13), (3.14), and (3.15), respectively, and (3.21) is $\mathcal{O}(n^2)$, it follows that the computational complexity of Algorithm 2 is identical to the computational complexity of Algorithm 1.

3.5 Convergence Analysis of SW-VR-RLS

Definition 3.5.1 ([69])**.** Let $x_{\text{eq}} \in \mathbb{R}^n$. Consider the system

$$x_{k+1} = f(x_k, k), \tag{3.22}$$

where $f : \mathbb{R}^n \times \{0, 1, 2, \ldots\} \to \mathbb{R}^n$ is a continuous function such that, for all $k \geq 0$, $f(x_{\text{eq}}, k) = x_{\text{eq}}$. The equilibrium solution $x_k \equiv x_{\text{eq}}$ of (3.22) is *Lyapunov stable* if, for every $\varepsilon > 0$ and $k_0 \geq 0$, there exists $\delta(\varepsilon, k_0) > 0$ such that $||x_{k_0} - x_{\text{eq}}|| < \delta$ implies that, for all $k \geq k_0$, $||x_k - x_{\text{eq}}|| < \varepsilon$. The equilibrium solution $x_k \equiv x_{\text{eq}}$ of (3.22) is *uniformly Lyapunov stable* if, for every $\varepsilon > 0$, there exists $\delta = \delta(\varepsilon) > 0$ such that, for all $k_0 \geq 0$, $||x_{k_0} - x_{\text{eq}}|| < \delta$ implies that, for all $k \geq k_0$, $||x_k - x_{\text{eq}}|| < \varepsilon$. The equilibrium solution $x_k \equiv x_{\text{eq}}$ of (3.22) is *globally asymptotically stable* if it is Lyapunov stable and, for all $k_0 \geq 0$ and $x_{k_0} \in \mathbb{R}^n$, $\lim_{k\to\infty} x_k = x_{\text{eq}}$.

The following result provides boundedness properties of the SW-VR-RLS algorithm. This result applies to both SW-VR-RLS implementations, specifically, Algorithm 1 and Algorithm 2.

Theorem 2. For all $k \geq 0$, let $T_k \in \mathbb{R}^{n\times n}$ be positive definite, and assume there exist $\varepsilon_1, \varepsilon_2 \in (0, \infty)$ such that, for all $k \geq 0$,

$$\varepsilon_1 I_n \leq T_{k+1} \leq T_k \leq \varepsilon_2 I_n. \tag{3.23}$$

Furthermore, for all $k \geq 0$, let $\xi_k \in \mathbb{R}$, assume that $0 < \inf_{k\geq 0} \xi_k \leq \sup_{k\geq 0} \xi_k < \infty$, and define $R_k \stackrel{\triangle}{=} \xi_k T_k$. Then the following statements hold:

(i) $\{L_k\}_{k=1}^{\infty}$, $\{Q_k\}_{k=1}^{\infty}$, and $\{P_k\}_{k=0}^{\infty}$ are bounded.

(ii) Assume that $\{\alpha_k\}_{k=0}^{\infty}$ and $\{b_k\}_{k=0}^{\infty}$ are bounded. Then $\{x_k\}_{k=0}^{\infty}$ is bounded.

Proof. (i) It follows from the first inequality in (3.23) that, for all $k \geq 0$, $R_k \geq c_1 I_n$, where $c_1 \stackrel{\triangle}{=} \varepsilon_1 \inf_{k\geq 0} \xi_k > 0$. Since, for all $k \geq 0$, A_k is positive semidefinite, it follows from (3.4) that $P_k^{-1} \geq c_1 I_n$, which implies that $0 \leq P_k \leq \frac{1}{c_1} I_n$. Thus, $\{P_k\}_{k=0}^{\infty}$ is bounded. Similarly, it follows from (3.7) and (3.9) that, for all $k \geq 1$, $Q_k^{-1} \geq c_1 I_n$ and $L_k^{-1} \geq c_1 I_n$, which imply that $0 \leq Q_k \leq \frac{1}{c_1} I_n$ and $0 \leq L_k \leq \frac{1}{c_1} I_n$. Thus, $\{Q_k\}_{k=1}^{\infty}$ and $\{L_k\}_{k=1}^{\infty}$ are bounded.

(ii) Since $\{b_k\}_{k=0}^{\infty}$ is bounded, it follows that $\kappa_1 \stackrel{\triangle}{=} \sup_k ||b_k|| < \infty$. Additionally, since $\{\alpha_k\}_{k=0}^{\infty}$ is bounded, it follows that $\kappa_2 \stackrel{\triangle}{=} \sup_k ||\alpha_k|| < \infty$. Furthermore, it follows from the last inequality in (3.23) that, for all $k \geq 0$, $R_k \leq c_2 I_n$, where $c_2 \stackrel{\triangle}{=} \varepsilon_2 \sup_{k\geq 0} \xi_k < \infty$. Hence, it follows from (3.5) that, for all $k \geq 0$,

$$\begin{aligned} ||x_k|| &= \left|\left| \tfrac{1}{2} P_k \left(\sum_{i=k-r}^{k} b_i - 2R_k \alpha_k \right) \right|\right| \\ &\leq \tfrac{1}{2} ||P_k|| \left(\left|\left| \sum_{i=k-r}^{k} b_i \right|\right| + 2||R_k||||\alpha_k|| \right) \\ &\leq \frac{1}{c_1} \left((r+1)\kappa_1 + 2c_2\kappa_2 \right). \end{aligned}$$

Therefore, $\{x_k\}_{k=0}^{\infty}$ is bounded. □

For all $k \geq 0$, define $\Phi_k \stackrel{\triangle}{=} [\psi_k \quad \cdots \quad \psi_{k-r}] \in \mathbb{R}^{n\times q_k}$, where $q_k \stackrel{\triangle}{=} \sum_{i=0}^{r} n_{k-i}$, so that $\sum_{i=k-r}^{k} A_i = \Phi_k \Phi_k^{\mathrm{T}}$. Furthermore, using the matrix inversion lemma, it follows from (3.4) that

$$P_k = R_k^{-1} - R_k^{-1}\Phi_k \left(I_{q_k} + \Phi_k^{\mathrm{T}} R_k^{-1} \Phi_k \right)^{-1} \Phi_k^{\mathrm{T}} R_k^{-1}. \tag{3.24}$$

Next, let ν be a positive integer, for all $k \geq \nu$, let $\alpha_k = x_{k-\nu}$, for all $k > \nu - 1$, define $\chi_k \triangleq \left[\begin{array}{cccc} x_k^{\mathrm{T}} & x_{k-1}^{\mathrm{T}} & \cdots & x_{k-\nu+1}^{\mathrm{T}} \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^{n\nu}$, and, for all $i \in \{1, \ldots, \nu\}$, let $\chi_{k,i} \triangleq x_{k-i+1}$. Then it follows from (3.16) that, for all $k > \nu - 2$,

$$\left[\begin{array}{c} \chi_{k+1,1} \\ \chi_{k+1,2} \\ \vdots \\ \chi_{k+1,\nu} \end{array} \right] = \left[\begin{array}{c} -P_{k+1} \left(\displaystyle\sum_{i=k+1-r}^{k+1} \tfrac{1}{2} b_i - R_{k+1} \chi_{k,\nu} \right) \\ \chi_{k,1} \\ \vdots \\ \chi_{k,\nu-1} \end{array} \right]. \tag{3.25}$$

Theorem 3. For all $k \geq 0$, let $T_k \in \mathbb{R}^{n \times n}$ be positive definite, and assume there exist $\varepsilon_1, \varepsilon_2 \in (0, \infty)$ such that, for all $k \geq 0$, (3.23) holds. Furthermore, for all $k \geq 0$, let $\xi_k \in \mathbb{R}$, assume that $0 < \inf\limits_{k \geq 0} \xi_k \leq \sup\limits_{k \geq 0} \xi_k < \infty$, and define $R_k \triangleq \xi_k T_k$ and $\Omega_k \triangleq \xi_k I_{q_k} + \Phi_k^{\mathrm{T}} T_k^{-1} \Phi_k$. Let ν be a positive integer and let $\eta \in \mathbb{R}^n$, for all $0 \leq k \leq \nu - 1$, define $\alpha_k \triangleq \eta$, and, for all $k \geq \nu$, define $\alpha_k \triangleq x_{k-\nu}$, where $x_{k-\nu}$ satisfies (3.5). Furthermore, let $P_0 = \left(\sum_{i=-r}^{0} A_i + R_0 \right)^{-1}$ and $x_0 = -\frac{1}{2} P_0 \left(\sum_{i=-r}^{0} b_i - 2R_0 \eta \right)$, assume there exists a unique $x_* \in \mathbb{R}^n$ such that, for all $k \geq 0$,

$$A_k x_* + \tfrac{1}{2} b_k = 0, \tag{3.26}$$

and define $\chi_* \triangleq \left[\begin{array}{cccc} x_*^{\mathrm{T}} & x_*^{\mathrm{T}} & \cdots & x_*^{\mathrm{T}} \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^{n\nu}$. Then the following statements hold:

(i) $\chi_k \equiv \chi_*$ is an equilibrium solution of (3.25)

(ii) The equilibrium solution $\chi_k \equiv \chi_*$ of (3.25) is uniformly Lyapunov stable, and, for all $x_0 \in \mathbb{R}^n$, $\{x_k\}_{k=0}^{\infty}$ is bounded.

(iii) $\sum_{j=\nu}^{\infty} (x_{j-\nu} - x_*)^{\mathrm{T}} \Phi_j \Omega_j^{-1} \Phi_j^{\mathrm{T}} (x_{j-\nu} - x_*)$ and $\sum_{j=\nu}^{\infty} ||x_j - x_{j-\nu}||^2$ exist.

(iv) Assume that $\{A_k\}_{k=0}^{\infty}$ is bounded. Then $\lim\limits_{k \to \infty} \left(A_k x_k + \frac{1}{2} b_k \right) = 0$, and $\lim\limits_{k \to \infty} \psi_k^{\mathrm{T}} (x_{k-\nu} - x^*) = 0$.

(*v*) Assume that $\{A_k\}_{k=0}^{\infty}$ is bounded and there exists $c > 0$ and a positive integer l such that, for all $k \geq \nu(l-1) - r$, $cI_n \leq \sum_{i=0}^{l-1} A_{k-\nu i}$. Then $\lim_{k\to\infty} x_k = x_*$, $\chi_k \equiv \chi_*$ is the unique equilibrium solution of (3.25), and, $\chi_k \equiv \chi_*$ is globally asymptotically stable.

Proof. (*i*) Let $\chi_{\nu-1} = \chi_*$. Then it follows from (3.25) and (3.26) that $\chi_{\nu,2} = \chi_{\nu,3} = \chi_{\nu,\nu} = \cdots = x_*$, and

$$\chi_{\nu,1} = -P_\nu \left(\sum_{i=\nu-r}^{\nu} \tfrac{1}{2} b_i - R_\nu x_* \right) = -P_\nu \left(-\sum_{i=\nu-r}^{\nu} \tfrac{1}{2} A_i - R_\nu \right) x_* = x_*,$$

and thus $\chi_\nu = \chi_*$. Similarly, for $k = \nu$, it follows from (3.25) and (3.26) that $\chi_{\nu+1,2} = \chi_{\nu+1,3} = \chi_{\nu+1,\nu} = \cdots = x_*$, and

$$\chi_{\nu+1,1} = -P_{\nu+1} \left(\sum_{i=\nu+1-r}^{\nu+1} \tfrac{1}{2} b_i - R_{\nu+1} x_* \right) = -P_{\nu+1} \left(-\sum_{i=\nu+1-r}^{\nu+1} \tfrac{1}{2} A_i - R_{\nu+1} \right) x_* = x_*,$$

and thus $\chi_{\nu+1} = \chi_*$. It follows that, for all $k > \nu - 2$, $\chi_k = \chi_*$, and thus $\chi_k \equiv \chi_*$ is an equilibrium solution of (3.25).

(*ii*) Since, for all $k \geq \nu$, $\alpha_k = x_{k-\nu}$, it follows from (3.5) that, for all $k \geq \nu$,

$$\begin{aligned} x_k &= -P_k \left(\sum_{i=k-r}^{k} \tfrac{1}{2} b_i - R_k x_{k-\nu} \right) \\ &= P_k \left(\sum_{i=k-r}^{k} A_i + R_k \right) x_{k-\nu} - P_k \sum_{i=k-r}^{k} \left(A_i x_{k-\nu} + \tfrac{1}{2} b_i \right) \\ &= x_{k-\nu} - P_k \sum_{i=k-r}^{k} \left(A_i x_{k-\nu} + \tfrac{1}{2} b_i \right). \end{aligned} \tag{3.27}$$

Define $\tilde{x}_k \stackrel{\triangle}{=} x_k - x_*$. Subtracting x_* from (3.27), and using (3.24) and (3.26) yields,

for all $k \geq \nu$,

$$\begin{aligned}
\tilde{x}_k &= \tilde{x}_{k-\nu} - P_k \sum_{i=k-r}^{k} A_i \tilde{x}_{k-\nu} \\
&= \tilde{x}_{k-\nu} - P_k \Phi_k \Phi_k^{\mathrm{T}} \tilde{x}_{k-\nu} \\
&= P_k \left(P_k^{-1} - \Phi_k \Phi_k^{\mathrm{T}} \right) \tilde{x}_{k-\nu} \\
&= P_k R_k \tilde{x}_{k-\nu} \\
&= \left[R_k^{-1} - R_k^{-1} \Phi_k \left(I_{q_k} + \Phi_k^{\mathrm{T}} R_k^{-1} \Phi_k \right)^{-1} \Phi_k^{\mathrm{T}} R_k^{-1} \right] R_k \tilde{x}_{k-\nu} \\
&= \tilde{x}_{k-\nu} - T_k^{-1} \Phi_k \left(\xi_k I_{q_k} + \Phi_k^{\mathrm{T}} T_k^{-1} \Phi_k \right)^{-1} \Phi_k^{\mathrm{T}} \tilde{x}_{k-\nu} \\
&= \tilde{x}_{k-\nu} - T_k^{-1} \Phi_k \Omega_k^{-1} \Phi_k^{\mathrm{T}} \tilde{x}_{k-\nu}. \qquad (3.28)
\end{aligned}$$

Define $\tilde{\chi}_k \stackrel{\triangle}{=} \chi_k - \chi_*$, and, for all $i \in \{1, \ldots, \nu\}$, let $\tilde{\chi}_{k,i} \stackrel{\triangle}{=} \tilde{x}_{k-i+1}$. Then it follows from (3.25) and (3.28) that, for all $k > \nu - 2$,

$$\begin{bmatrix} \tilde{\chi}_{k+1,1} \\ \tilde{\chi}_{k+1,2} \\ \vdots \\ \tilde{\chi}_{k+1,\nu} \end{bmatrix} = \begin{bmatrix} \left(I - T_{k+1}^{-1} \Phi_{k+1} \Omega_{k+1}^{-1} \Phi_{k+1}^{\mathrm{T}} \right) \tilde{\chi}_{k,\nu} \\ \tilde{\chi}_{k,1} \\ \vdots \\ \tilde{\chi}_{k,\nu-1} \end{bmatrix}. \qquad (3.29)$$

Note that $\tilde{\chi}_k \equiv 0$ is an equilibrium solution of (3.29). For all $z \in \mathbb{R}$, define the strictly increasing functions $\alpha(z) \stackrel{\triangle}{=} \varepsilon_1 z$ and $\beta(z) \stackrel{\triangle}{=} \varepsilon_2 z$, and, for all $k \geq \nu - 1$, define the function

$$V(\tilde{\chi}_k, k) \stackrel{\triangle}{=} \sum_{i=1}^{\nu} \tilde{\chi}_{k,i}^{\mathrm{T}} T_{k+1-i} \tilde{\chi}_{k,i}.$$

The difference $\Delta V_k \triangleq V(\tilde{\chi}_k, k) - V(\tilde{\chi}_{k-1}, k-1)$ is given by

$$\begin{aligned}
\Delta V_k &= \tilde{\chi}_{k-1,\nu}^{\mathrm{T}} \left(T_k - T_{k-\nu}\right) \tilde{\chi}_{k-1,\nu} - 2\tilde{\chi}_{k-1,\nu}^{\mathrm{T}} \Phi_k \Omega_k^{-1} \Phi_k^{\mathrm{T}} \tilde{\chi}_{k-1,\nu} \\
&\quad + \tilde{\chi}_{k-1,\nu}^{\mathrm{T}} \Phi_k \Omega_k^{-1} \Phi_k^{\mathrm{T}} T_k^{-1} \Phi_k \Omega_k^{-1} \Phi_k^{\mathrm{T}} \tilde{\chi}_{k-1,\nu} \\
&\leq -2\tilde{\chi}_{k-1,\nu}^{\mathrm{T}} \Phi_k \Omega_k^{-1} \Phi_k^{\mathrm{T}} \tilde{\chi}_{k-1,\nu} + \tilde{\chi}_{k-1,\nu}^{\mathrm{T}} \Phi_k \Omega_k^{-1} \Phi_k^{\mathrm{T}} T_k^{-1} \Phi_k \Omega_k^{-1} \Phi_k^{\mathrm{T}} \tilde{\chi}_{k-1,\nu} \\
&\leq -\tilde{\chi}_{k-1,\nu}^{\mathrm{T}} \Phi_k \Omega_k^{-1} \left(I_{q_k} + I_{q_k} - \Phi_k^{\mathrm{T}} T_k^{-1} \Phi_k \Omega_k^{-1}\right) \Phi_k^{\mathrm{T}} \tilde{\chi}_{k-1,\nu} \\
&\leq -\tilde{\chi}_{k-1,\nu}^{\mathrm{T}} \Phi_k \Omega_k^{-1} \left[I_{q_k} + \left(\Omega_k - \Phi_k^{\mathrm{T}} T_k^{-1} \Phi_k\right) \Omega_k^{-1}\right] \Phi_k^{\mathrm{T}} \tilde{\chi}_{k-1,\nu} \\
&\leq -\tilde{\chi}_{k-1,\nu}^{\mathrm{T}} \Phi_k \Omega_k^{-1} \left(I_{q_k} + \xi_k \Omega_k^{-1}\right) \Phi_k^{\mathrm{T}} \tilde{\chi}_{k-1,\nu} \\
&\leq -\tilde{\chi}_{k-1,\nu}^{\mathrm{T}} \Phi_k \Omega_k^{-1} \Phi_k^{\mathrm{T}} \tilde{\chi}_{k-1,\nu}. \qquad (3.30)
\end{aligned}$$

Since, for all $k \geq \nu - 1$ and $\tilde{\chi}_k \in \mathbb{R}^{n\nu}$, $\alpha(||\tilde{\chi}_k||) \leq V(\tilde{\chi}_k, k) \leq \beta(||\tilde{\chi}_k||)$ and $\Delta V_k \leq 0$, it follows from [69, Theorem 13.11] that the equilibrium solution $\tilde{\chi}_k \equiv 0$ of (3.29) is uniformly Lyapunov stable. Furthermore, since $\alpha(z) \to \infty$ as $z \to \infty$, it follows from [69, Corollary 13.4] that, for each $\tilde{\chi}_{\nu-1} \in \mathbb{R}^{n\nu}$, the sequence $\{\tilde{\chi}_k\}_{k=\nu-1}^{\infty}$ is bounded. Hence, for each $x_0 \in \mathbb{R}^n$, $\{\tilde{x}_k\}_{k=0}^{\infty}$ is bounded, and thus $\{x_k\}_{k=0}^{\infty}$ is bounded.

(*iii*) It follows from (3.30) that

$$0 \leq \sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}} \Phi_j \Omega_j^{-1} \Phi_j^{\mathrm{T}} \tilde{x}_{j-\nu} \leq -\sum_{j=\nu}^{k} \Delta V_j = V(\tilde{\chi}_{\nu-1}, \nu-1) - V(\tilde{\chi}_k, k) \leq V(\tilde{\chi}_{\nu-1}, \nu-1).$$

Hence, the nondecreasing sequence $\{\sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}} \Phi_j \Omega_j^{-1} \Phi_j^{\mathrm{T}} \tilde{x}_{j-\nu}\}_{k=\nu}^{\infty}$ is bounded, and thus $\sum_{j=\nu}^{\infty} \tilde{x}_{j-\nu}^{\mathrm{T}} \Phi_j \Omega_j^{-1} \Phi_j^{\mathrm{T}} \tilde{x}_{j-\nu}$ exists.

Next, for all $k \geq \nu$, define $\mathcal{M}_k \triangleq \sum_{j=\nu}^{k} ||x_j - x_{j-\nu}||^2$. It follows from (3.28) that

$$\begin{aligned}
\mathcal{M}_k &= \sum_{j=\nu}^{k} ||T_j^{-1}\Phi_j\Omega_j^{-1}\Phi_j^{\mathrm{T}}\tilde{x}_{j-\nu}||^2 \\
&= \sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}}\Phi_j\Omega_j^{-1}\Phi_j^{\mathrm{T}}T_j^{-2}\Phi_j\Omega_j^{-1}\Phi_j^{\mathrm{T}}\tilde{x}_{j-\nu} \\
&\leq \sum_{j=\nu}^{k} \|T_j^{-1}\|\tilde{x}_{j-\nu}^{\mathrm{T}}\Phi_j\Omega_j^{-1}\Phi_j^{\mathrm{T}}T_j^{-1}\Phi_j\Omega_j^{-1}\Phi_j^{\mathrm{T}}\tilde{x}_{j-\nu}.
\end{aligned}$$

Note that, for all $k \geq \nu$, $\|T_k^{-1}\| \leq \|\frac{1}{\varepsilon_1}I_n\| = \frac{1}{\varepsilon_1}$. Therefore,

$$\begin{aligned}
\mathcal{M}_k &\leq \frac{1}{\varepsilon_1}\sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}}\Phi_j\Omega_j^{-1}\left(\xi_j I_{q_j} + \Phi_j^{\mathrm{T}}T_j^{-1}\Phi_j - \xi_j I_{q_j}\right)\Omega_j^{-1}\Phi_j^{\mathrm{T}}\tilde{x}_{j-\nu} \\
&= \frac{1}{\varepsilon_1}\sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}}\Phi_j\Omega_j^{-1}\left(\Omega_j - \xi_j I_{q_j}\right)\Omega_j^{-1}\Phi_j^{\mathrm{T}}\tilde{x}_{j-\nu} \\
&= \frac{1}{\varepsilon_1}\sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{T}\Phi_j\Omega_j^{-1}\Phi_j^{\mathrm{T}}\tilde{x}_{j-\nu} - \frac{1}{\varepsilon_1}\sum_{j=\nu}^{k} \xi_j\tilde{x}_{j-\nu}^{\mathrm{T}}\Phi_j\Omega_j^{-2}\Phi_j^{\mathrm{T}}\tilde{x}_{j-\nu} \\
&\leq \frac{1}{\varepsilon_1}\sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}}\Phi_j\Omega_j^{-1}\Phi_j^{\mathrm{T}}\tilde{x}_{j-\nu}.
\end{aligned}$$

Since $\sum_{j=\nu}^{\infty} \tilde{x}_{j-\nu}^{\mathrm{T}}\Phi_j\Omega_j^{-1}\Phi_j^{\mathrm{T}}\tilde{x}_{j-\nu}$ exists, it follows that the nondecreasing sequence $\{\mathcal{M}_k\}_{k=\nu}^{\infty}$ is bounded, and thus $\lim_{k\to\infty}\mathcal{M}_k$ exists, which verifies (iii).

(iv) It follows from (iii) that $\lim_{k\to\infty}\Omega_k^{-1/2}\Phi_k^{\mathrm{T}}\tilde{x}_{k-\nu} = 0$. Next, since $\{A_k\}_{k=0}^{\infty}$ is bounded, it follows that $\{\Phi_k\}_{k=0}^{\infty}$ is bounded. Since, in addition, $\{\xi_k\}_{k=0}^{\infty}$ and $\{T_k^{-1}\}_{k=0}^{\infty}$ are bounded, it follows that there exists $c_3 > 0$ such that, for all $k \geq 0$, $c_3 I_{q_k} \leq \sigma_{\min}\left(\Omega_k^{-1/2}\right) I_{q_k} \leq \Omega_k^{-1/2}$. Therefore,

$$0 \leq c_3||\Phi_k^{\mathrm{T}}\tilde{x}_{k-\nu}|| \leq \sigma_{\min}\left(\Omega_k^{-1/2}\right)||\Phi_k^{\mathrm{T}}\tilde{x}_{k-\nu}|| \leq ||\Omega_k^{-1/2}\Phi_k^{\mathrm{T}}\tilde{x}_{k-\nu}|| \to 0 \text{ as } k \to \infty.$$

Therefore, $\lim_{k\to\infty}\Phi_k^{\mathrm{T}}\tilde{x}_{k-\nu} = 0$, which implies that $\lim_{k\to\infty}\psi_k^{\mathrm{T}}\tilde{x}_{k-\nu} = 0$. Next, since

$\{A_k\}_{k=0}^{\infty}$ is bounded, it follows that $\kappa \stackrel{\triangle}{=} \sup_{k\geq 0} \sigma_{\max}(\psi_k) < \infty$. It follows from (iii) that $||\tilde{x}_k - \tilde{x}_{k-\nu}|| \to 0$ as $k \to \infty$, and thus

$$\begin{aligned}
||A_k x_k + \tfrac{1}{2} b_k|| &= ||A_k x_k - A_k x_*|| \\
&= ||\psi_k \psi_k^{\mathrm{T}} \tilde{x}_k|| \\
&\leq \kappa ||\psi_k^{\mathrm{T}} \tilde{x}_k|| \\
&= \kappa ||\psi_k^{\mathrm{T}} \tilde{x}_{k-\nu} + \psi_k^{\mathrm{T}} \tilde{x}_k - \psi_k^{\mathrm{T}} \tilde{x}_{k-\nu}|| \\
&\leq \kappa \Big(||\psi_k^{\mathrm{T}} \tilde{x}_{k-\nu}|| + ||\psi_k|| ||\tilde{x}_k - \tilde{x}_{k-\nu}|| \Big) \\
&\leq \kappa \Big(||\psi_k^{\mathrm{T}} \tilde{x}_{k-\nu}|| + \kappa ||\tilde{x}_k - \tilde{x}_{k-\nu}|| \Big) \\
&= \kappa ||\psi_k^{\mathrm{T}} \tilde{x}_{k-\nu}|| + \kappa^2 ||\tilde{x}_k - \tilde{x}_{k-\nu}|| \\
&\to 0 \text{ as } k \to \infty,
\end{aligned} \tag{3.31}$$

which proves (iv).

(v) First, we show that $\chi_k = \chi_*$ is the unique equilibrium solution of (3.25). Let $\bar{x} \in \mathbb{R}^n$, and assume $\bar{\chi} \stackrel{\triangle}{=} \left[\begin{array}{ccc} \bar{x}^{\mathrm{T}} & \cdots & \bar{x}^{\mathrm{T}} \end{array} \right]^{\mathrm{T}}$ is an equilibrium solution of (3.25). Let $\chi_{\nu-1} = \bar{\chi}$. It follows that $\chi_{\nu,2} = \chi_{\nu,3} = \cdots = \chi_{\nu,\nu} = \bar{x}$, and

$$\chi_{\nu,1} = \bar{x} = -\left(\sum_{i=\nu-r}^{\nu} A_i + R_\nu \right)^{-1} \left(\sum_{i=\nu-r}^{\nu} \tfrac{1}{2} b_i - R_\nu \bar{x} \right). \tag{3.32}$$

Multiplying both sides of (3.32) by $\left(\sum_{i=\nu-r}^{\nu} A_i + R_\nu \right)$ yields $\sum_{i=\nu-r}^{\nu} A_i \bar{x} = -\sum_{i=\nu-r}^{\nu} \frac{1}{2} b_i$. Similarly, for $k = \nu$, it follows from (3.25) that $\chi_{\nu+1,2} = \chi_{\nu+1,3} = \cdots = \chi_{\nu+1,\nu} = \bar{x}$ and $\sum_{i=\nu+1-r}^{\nu+1} A_i \bar{x} = -\sum_{i=\nu+1-r}^{\nu+1} \frac{1}{2} b_i$. Repeating this process for $k = \nu+1, \nu+2, \ldots$ yields, for all $k \geq \nu$,

$$\sum_{i=k-r}^{k} A_i \bar{x} = -\sum_{i=k-r}^{k} \tfrac{1}{2} b_i. \tag{3.33}$$

Summing (3.33) over $k \geq \nu l$ yields

$$\left(\sum_{k-\nu(l-1)}^{k-\nu(l-1)+r} A_i + \sum_{k-\nu(l-1)+1}^{k-\nu(l-1)+1+r} A_i + \cdots + \sum_{k-r}^{k} A_i \right) \bar{x} \\ = \tfrac{1}{2} \left(\sum_{k-\nu(l-1)}^{k-\nu(l-1)+r} b_i + \sum_{k-\nu(l-1)+1}^{k-\nu(l-1)+1+r} b_i + \cdots + \sum_{k-r}^{k} b_i \right). \tag{3.34}$$

Since for all $k \geq \nu(l-1) - r$, $cI_n \leq \sum_{i=0}^{l-1} A_{k-\nu i}$, the coefficient of $\bar{x}$ in equation (3.34) is positive definite, and thus solution of (3.34) is unique. Since for all $k \geq 0$, $A_k x_* + \frac{1}{2} b_k = 0$, it follows that x_* is the unique solution of (3.34), and thus $\bar{x} = x_*$, and $\chi_k \equiv \chi_*$ is the unique equilibrium solution of (3.25).

Now, since $\{A_k\}_{k=0}^{\infty}$ is bounded, it follows that $\kappa \triangleq \sup_{k \geq 0} \sigma_{\max}(\psi_k) < \infty$. Using (iii) and arguments similar to prove (3.31) in (iv), it follows that

$$\begin{aligned} ||A_{k-\nu} \tilde{x}_k|| &\leq \kappa ||\psi_{k-\nu}^{\mathrm{T}} \tilde{x}_k|| \\ &= \kappa ||\psi_{k-\nu}^{\mathrm{T}} \tilde{x}_{k-\nu} + \psi_{k-\nu}^{\mathrm{T}} \tilde{x}_k - \psi_{k-\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}|| \\ &\leq \kappa ||\psi_{k-\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}|| + \kappa^2 ||\tilde{x}_k - \tilde{x}_{k-\nu}|| \\ &\to 0 \text{ as } k \to \infty. \end{aligned} \tag{3.35}$$

Note that (3.35) implies that $\lim_{k \to \infty} ||\psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}|| = 0$. It follows that

$$\begin{aligned} ||A_{k-2\nu} \tilde{x}_k|| &\leq \kappa ||\psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_k|| \\ &= \kappa ||\psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_{k-\nu} + \psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_k - \psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}|| \\ &\leq \kappa ||\psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}|| + \kappa^2 ||\tilde{x}_k - \tilde{x}_{k-\nu}|| \\ &\to 0 \text{ as } k \to \infty. \end{aligned}$$

Repeating this argument shows that, for all $i \in \{k-(l-1)\nu,\, k-(l-2)\nu,\, \ldots,\, k-\nu,\, k\}$,

$\lim_{k\to\infty} ||A_i\tilde{x}_k|| = 0$. Since, for all $k \geq \nu(l-1) - r$, $cI_n \leq \sum_{i=0}^{l-1} A_{k-\nu i}$, it follows that

$$\begin{aligned} ||\tilde{x}_k|| &\leq \frac{1}{c}\left|\left|\sum_{i=0}^{l-1} A_{k-\nu i}\tilde{x}_k\right|\right| \\ &\leq \frac{1}{c}\sum_{i=0}^{l-1} ||A_{k-\nu i}\tilde{x}_k|| \\ &\to 0 \text{ as } k \to \infty. \end{aligned}$$

Hence, $\lim_{k\to\infty} \tilde{x}_k = 0$. Thus the unique equilibrium solution $\chi_k \equiv \chi_*$ of (3.25) is globally asymptotically stable.

□

3.6 Simulations

In this section, we study the effect of R_k and r on SW-VR-RLS, and compare SW-VR-RLS with the proportionate affine projection algorithm (PAPA) [70] and the proportionate normalized least mean squares (PNLMS) algorithm [71] for systems where x_* changes abruptly.

Let ℓ be the number of data points, and, for $\{p_k\}_{k=1}^{\ell} \subset \mathbb{R}$, define

$$\sigma_p \triangleq \sqrt{\frac{1}{\ell}\sum_{k=1}^{\ell} p_k^2}.$$

Let n be a nonnegative integer. For $i \in \{0, \ldots, n-1\}$, let $h_i \in \mathbb{R}$, and define

$$x_* \triangleq \begin{bmatrix} h_0 & h_1 \cdots & h_{n-1} \end{bmatrix}^{\mathrm{T}}.$$

For all $k \geq 1$, let u_k, $y_k \in \mathbb{R}$, and, for all $-r-n+1 \leq k \leq 0$, let $u_k = 0$ and $y_k = 0$.

Furthermore, for all $k \geq 1$, let y_k satisfy the finite impulse response

$$y_k = \sum_{i=0}^{n-1} h_i u_{k-i}. \tag{3.36}$$

Next, for all $k \geq -r-n+1$, define the noisy output $\bar{y}_k \triangleq y_k + w_k$, where, for all $-r-n+1 \leq k \leq 0$, $w_k = 0$, and, for all $k \geq 1$, $w_k \in \mathbb{R}$ is sampled from a white noise process with a zero-mean Gaussian distribution with variance σ_w^2. Define the signal to noise ratio $\mathrm{SNR} \triangleq \sigma_y/\sigma_w$.

Let $x \in \mathbb{R}^n$, for all $k \geq -r$, define $\psi_k \triangleq \begin{bmatrix} u_k & \cdots & u_{k-n+1} \end{bmatrix}^{\mathrm{T}}$, and, for all $k \geq 0$, define the cost function

$$\mathcal{J}_k(x) \triangleq \sum_{i=k-r}^{k} \left(\bar{y}_k - \psi_k^{\mathrm{T}} x\right)^{\mathrm{T}} \left(\bar{y}_k - \psi_k^{\mathrm{T}} x\right) + (x-\alpha_k)^{\mathrm{T}} R_k (x-\alpha_k). \tag{3.37}$$

For all $k \geq -r$, define $A_k \triangleq \psi_k \psi_k^{\mathrm{T}}$ and $b_k \triangleq -2\bar{y}_k \psi_k$. It follows from (3.37) that

$$\mathcal{J}_k(x) = \sum_{i=k-r}^{k} \left(x^{\mathrm{T}} A_k x + b_k x_k\right) + (x-\alpha_k)^{\mathrm{T}} R_k (x-\alpha_k) + \sum_{i=k-r}^{k} \bar{y}_k^{\mathrm{T}} \bar{y}_k. \tag{3.38}$$

Then, for all $k \geq 0$, $\mathcal{J}_k(x) = J_k(x) + \sum_{i=k-r}^{k} \bar{y}_i^{\mathrm{T}} \bar{y}_i$, and thus the minimizer x_k of (3.38) is given by the minimizer (4.2) of the SW-VR-RLS cost function (3.1).

Next, define the performance $\varepsilon_k \triangleq 20 \log_{10} (\|x_* - x_k\|/\|x_*\|)$. Define

$$z_1 \triangleq \begin{bmatrix} -1.0667 & 0.9337 & 0.3503 & -0.0290 & 0.1825 & -1.5651 & -0.0845 \\ 1.6039 & 0.0983 & 0.0414 & -0.7342 & -0.0308 & 0.2323 & 0.4264 & -0.3728 \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{15},$$

$$z_2 \triangleq \begin{bmatrix} -0.0835 & 0.8205 & -1.3594 & 1.4417 & 0.8726 & 0.4442 & -0.2222 \\ -0.8215 & 0.5131 & -0.6638 & 0.1265 & -0.0155 & -0.1581 & 0.6957 & -0.8379 \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{15}.$$

For all examples, $\alpha_0 = x_0$, for all $k \geq 1$, $\alpha_k = x_{k-1}$, $n = 15$, and

$$x_* = \begin{cases} z_1, & \text{if } 0 \leq k \leq 999, \\ z_2, & \text{if } k \geq 1000, \end{cases} \tag{3.39}$$

unless otherwise specified. Furthermore, for all examples, we use Algorithm 1. We compute the ensemble average of ε_k based on 100 simulations with independent realizations of u_k and w_k.

3.6.1 Effect of R_k

First, we examine the effect of R_k on the performance of SW-VR-RLS, where $R_k \equiv R$ is constant and the coefficients of (3.36) change abruptly at $k = 1000$. Let $r = 60$, for all $k \geq 0$, let u_k be sampled from a white noise process with a zero-mean Gaussian distribution with variance 10, and let x_* be given by (3.39).

We test SW-VR-RLS for three values of R_k and three values of SNR. Specifically, $R = 1,000 I_{n\times n}$, $R = 10,000 I_{n\times n}$, and $R = 30,000 I_{n\times n}$. Figure 3.1 shows that, for this example, a smaller value of R yields faster convergence of ε_k but also a larger asymptotic mean value of ε_k. Furthermore, for each R, a larger value of SNR yields a smaller asymptotic mean value of ε_k.

To understand why a smaller value of R yields a larger asymptotic mean value of ε_k in the case of noisy data, first note that a smaller R makes the regularization term $(x_k - x_{k-1})^{\mathrm{T}} R (x_k - x_{k-1})$ of (3.1) smaller. Since the regularization term has the effect of opposing movement of the estimate x_k away from x_{k-1}, smaller R makes x_k more sensitive to noise. Furthermore, as k increases, $||x_k - x_{k-1}||$ tends to decrease to its asymptotic mean value, and thus the regularization term $(x_k - x_{k-1})^{\mathrm{T}} R (x_k - x_{k-1})$ decreases. Thus, a larger value of R means that the regularization term contributes more asymptotically to the cost function (3.1). Thus, more regularization (i.e., larger

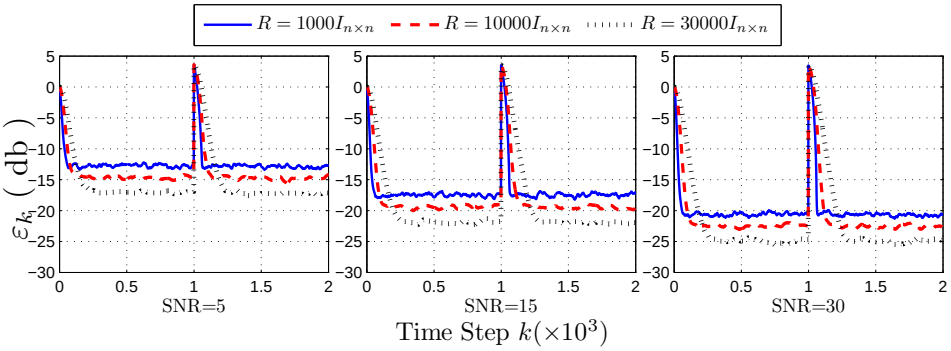


Figure 3.1: Effect of R_k on convergence of ε_k to its asymptotic mean value, where $R_k \equiv R$ is a constant. For this example, a smaller value of R yields faster convergence of ε_k to its asymptotic mean value but a larger asymptotic mean value of ε_k. Furthermore, for each value of R, a larger value of SNR yields a smaller asymptotic mean value of ε_k.

R) can make the estimate x_k asymptotically less sensitive to noise in y_k, which in turn can yield smaller asymptotic mean values of ε_k.

Next, we consider a time-varying R_k. First, define the residual $v_k \triangleq ||\bar{y}_k - x_k\psi_k||$ and the filtered residual

$$\bar{v}_k = \gamma\bar{v}_{k-1} + (1-\gamma)v_k,$$

where $\gamma \in (0,1)$ is a smoothing factor. Furthermore, let

$$R_k = \begin{cases} R_{\min}I_{n\times n}, & \bar{v}_k \leq \rho, \\ R_{\max}I_{n\times n}, & \bar{v}_k > \rho. \end{cases} \tag{3.40}$$

For this example, $\gamma = 0.05$, $R_{\min} = 10,000$, $R_{\max} = 50,000$, $\rho = 2.5$, and SNR= 20. Note that we allow only rank-1 modifications in R_k so that the computational complexity of SW-VR-RLS is $\mathcal{O}(n^2)$. Therefore, in order to modify R_k from $R_{\min}I_{n\times n}$ to $R_{\max}I_{n\times n}$, we modify the first diagonal entry of R_k at the current time step, and change the next diagonal entry at the next time step and so on. Figure 3.2 shows

that (3.40) yields a smaller asymptotic mean value of ε_k than $R_k \equiv 10,000 I_{n\times n}$, and faster convergence of ε_k to its asymptotic mean value than $R_k \equiv 50,000 I_{n\times n}$.

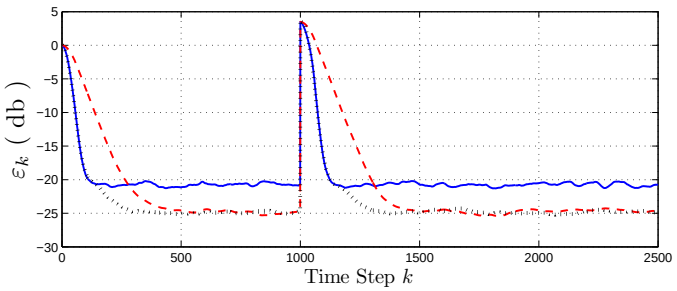


Figure 3.2: Effect of R_k on convergence of ε_k to its asymptotic mean value when R_k is time-varying. The solid line, dashed line, and dotted line indicate SW-VR-RLS with $R_k \equiv 10,000 I_{n\times n}$, $R_k \equiv 50,000 I_{n\times n}$, and R_k given by (3.40), respectively. For this example, R_k given by (3.40) yields a smaller asymptotic mean value of ε_k than $R_k \equiv 10,000 I_{n\times n}$, and yields faster convergence of ε_k to its asymptotic mean value than $R_k \equiv 50,000 I_{n\times n}$.

3.6.2 Effect of window size

For all $k \geq 0$, let u_k be sampled from a zero-mean Gaussian white noise process with variance 10, let SNR= 20, let x_* be given by (3.39), and, for all $k \geq 0$, let $R_k = 1,000 I_{n\times n}$. We test SW-VR-RLS with $r = 0$, $r = 50$, $r = 100$, and $r = 200$. Figure 3.3 shows that, as r is increased from 0, the asymptotic mean value of ε_k and the speed of convergence of ε_k to its asymptotic mean value initially increase and then decrease.

To gain further insight into how to choose r, we fix $r = 200$, and test SW-VR-RLS when $R_k \equiv R$ is constant. We test five different values of R, specifically, $R = I_{n\times n}$, $R = 10 I_{n\times n}$, $R = 100 I_{n\times n}$, $R = 1,000 I_{n\times n}$, and $R = 10,000 I_{n\times n}$. For this simulation, Figure 3.4 shows that decreasing the value of R from $1,000 I_{n\times n}$ to $I_{n\times n}$ does not increase the speed of convergence of ε_k to its asymptotic mean value. This suggests that, as R is decreased beyond a certain value, it no longer affects the speed of

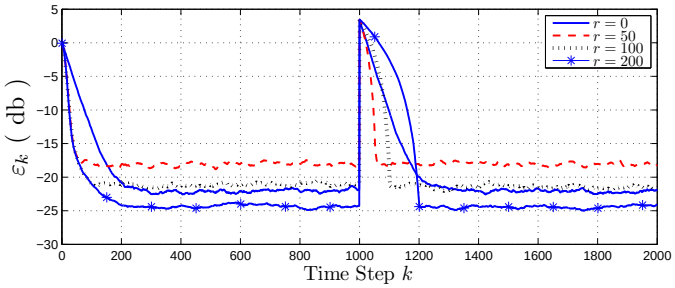


Figure 3.3: Effect of r on convergence of ε_k to its asymptotic mean value. This plot shows that, as r is increased from 0, the asymptotic mean value of ε_k and the speed of convergence of ε_k to its asymptotic mean value first increase and then decrease.

convergence or asymptotic mean value of ε_k, and r must be decreased in order to increase the speed of convergence of ε_k.

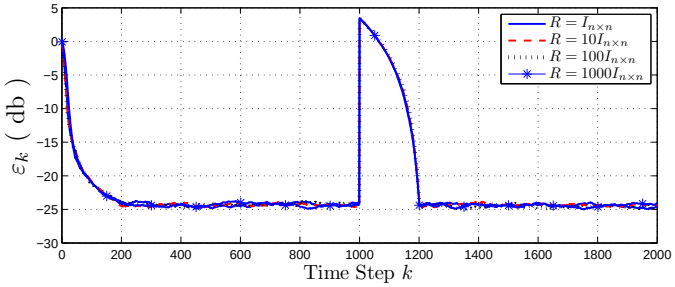


Figure 3.4: Effect of constant R on convergence of ε_k to its asymptotic mean value when $r = 200$. This plot shows that decreasing the value of R from $1,000I_{n\times n}$ to $I_{n\times n}$ does not increase either the speed of convergence or the asymptotic mean value of ε_k.

3.6.3 Comparison with PAPA and PNLMS

To compare SW-VR-RLS with PAPA and PNLMS, for all $k \geq 0$, let u_k be sampled from a white noise process with a zero-mean Gaussian distribution with variance 10, let x_* be given by (3.39), and let SNR = 20. For SW-VR-RLS, we use $r = 60$ and R_k specified by (3.40) with $R_{\min} = 6,000$, $R_{\max} = 25,000$, $\rho = 2.5$, and $\gamma = 0.1$. For PNLMS [71], we set δ(PNLMS)= 0.01, ρ(PNLMS)= $15/(n+1)$, μ(PNLMS)= 0.2, and, for the PAPA [70], we set δ_ρ(PAPA)= 0.01, ρ(PAPA)= $15/n$, μ(PAPA)= 0.2,

and $\delta(\text{PAPA}) = 100/n$. Note that for these parameters all three algorithms have approximately the same mean steady-state error. Figure 3.5 shows that, for $k \leq 999$, SW-VR-RLS yields faster convergence of ε_k to its asymptotic mean value than PNLMS and PAPA. Furthermore, at $k = 1000$, $x_* \neq z_1$, and SW-VR-RLS yields faster convergence of ε_k to its new asymptotic mean value than PNLMS and PAPA.

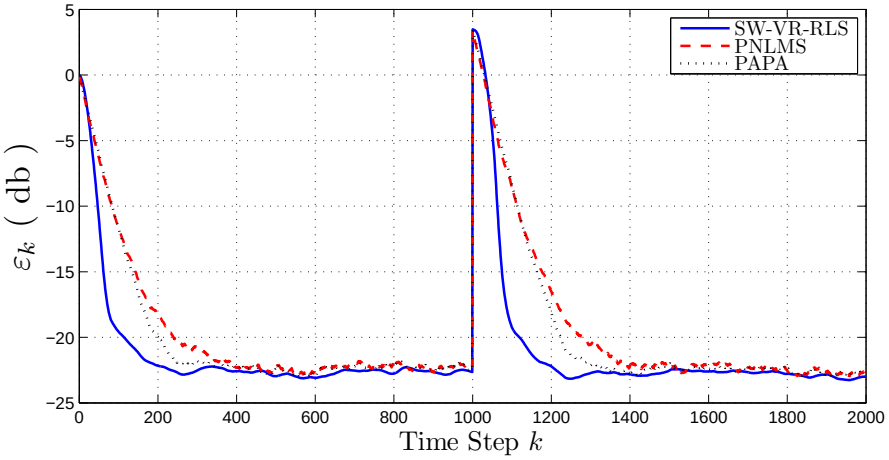


Figure 3.5: This plot compares SW-VR-RLS with PAPA and PNLMS when the input signal is white. For $k \leq 1000$, SW-VR-RLS yields faster convergence of ε_k to its asymptotic mean value than PNLMS and PAPA. Furthermore, at $k = 1000$, $x_* \neq z_1$, and SW-VR-RLS yields faster convergence of ε_k to its new asymptotic mean value than PNLMS and PAPA.

Next, we consider the case where u_k is colored. Since convergence of SW-VR-RLS, PAPA, and PNLMS are slower in the presence of colored inputs as compared to white inputs, we consider

$$x_* = \begin{cases} z_1, & \text{if } 0 \leq k \leq 3999, \\ z_2, & \text{if } k \geq 4000. \end{cases}$$

Let SNR$= 20$, $\bar{u}_k$ be sampled from a white noise process with a zero-mean Gaussian

distribution with variance 10, and let

$$u_k = 0.9u_{k-1} + \bar{u}_k.$$

For SW-VR-RLS, we use $r = 800$ and R_k specified by (3.40) with $R_{\min} = 5 \times 10^4$, $R_{\max} = 35 \times 10^4$, $\rho = 3.5$, and $\gamma = 0.01$. For PNLMS [71], we set δ(PNLMS)$= 0.05$, ρ(PNLMS)$= 15/(n+1)$, μ(PNLMS)$= 0.085$, and, for PAPA [70], we set δ_ρ(PAPA)$=$ 0.01, ρ(PAPA)$=$ $15/n$, μ(PAPA)$=$ 0.02, and δ(PAPA)$=$ $5/n$. Note that we have chosen these parameters such that all three algorithms have approximately the steady-state mean error. Figure 3.6 shows that, for this example, and for $k \leq 3999$, SW-VR-RLS yields faster convergence of ε_k to its asymptotic mean value than PNLMS and PAPA. Furthermore, at $k = 4000$, $x_* \neq z_1$, and SW-VR-RLS yields faster convergence of ε_k to its asymptotic mean value than PNLMS and PAPA.

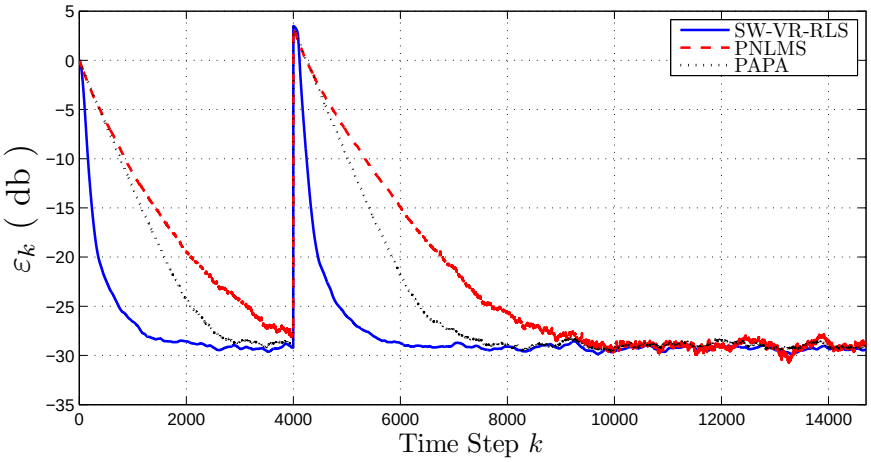


Figure 3.6: This plot compares SW-VR-RLS with PAPA and PNLMS when the input signal u_k is colored. For $k \leq 1000$, SW-VR-RLS yields faster convergence of ε_k to its asymptotic mean value than PNLMS and PAPA. Furthermore, at $k = 4000$, $x_* \neq z_1$, and SW-VR-RLS yields faster convergence of ε_k to its new asymptotic mean value than PNLMS and PAPA.

3.7 Numerical Stability

In this section, we investigate the numerical stability of SW-VR-RLS to account for the effects of roundoff and quantization errors in x_k and P_k. Throughout this section, we assume that, for all $0 \leq k \leq \nu - 1$, $\alpha_k \stackrel{\triangle}{=} x_0$, and, for all $k \geq \nu$, $\alpha_k \stackrel{\triangle}{=} x_{k-\nu}$, where ν is a positive integer.

3.7.1 Numerical Errors in x_k

To examine the numerical stability of Algorithm 2, we perturb x_{k_0} at step k_0, and analyze the propagation of this error, assuming all subsequent calculations are performed with infinite-precision arithmetic. Let $\gamma \in \mathbb{R}^n$. For all $k > k_0$, let $\bar{x}_k$ denote the SW-VR-RLS minimizer given by Algorithm 2, where the initial condition is $\bar{x}_{k_0} \stackrel{\triangle}{=} x_{k_0} + \gamma$, where x_{k_0} is the SW-VR-RLS minimizer given by Algorithm 2 at step k_0. Thus, it follows from (3.21) that, for all $k \geq k_0$, $\bar{x}_k$ satisfies

$$\begin{aligned} \bar{x}_k = {} & [I_n - P_k(A_k - A_{k-r-1} + R_k - R_{k-1})]\,\bar{x}_{k-1} - \tfrac{1}{2}P_k(b_k - b_{k-r-1}) \\ & + P_k R_k \bar{\alpha}_k - P_k R_{k-1}\bar{\alpha}_{k-1}, \end{aligned} \tag{3.41}$$

where, for all $k_0 \leq k \leq k_0 + \nu - 1$, $\bar{\alpha}_k \stackrel{\triangle}{=} \alpha_k$, and, for all $k \geq k_0 + \nu$, $\bar{\alpha}_k \stackrel{\triangle}{=} \bar{x}_{k-\nu}$. For all $k \geq k_0$, define $\delta_k \stackrel{\triangle}{=} \bar{x}_k - x_k$ and note that $\delta_{k_0} = \gamma$. Subtracting (3.21) from (3.41), and using (3.10), it follows that, for all $k > k_0$,

$$\begin{aligned} \delta_k = {} & [I_n - P_k(A_k - A_{k-r-1} + R_k - R_{k-1})]\,(\bar{x}_{k-1} - x_{k-1}) \\ & + P_k R_k\,(\bar{\alpha}_k - \alpha_k) - P_k R_{k-1}\,(\bar{\alpha}_{k-1} - \alpha_{k-1}) \\ = {} & (I_n - P_k(P_k^{-1} - P_{k-1}^{-1}))\,(\bar{x}_{k-1} - x_{k-1}) + P_k R_k\,(\bar{\alpha}_k - \alpha_k) - P_k R_{k-1}\,(\bar{\alpha}_{k-1} - \alpha_{k-1}) \\ = {} & P_k P_{k-1}^{-1}\delta_{k-1} + P_k R_k \delta_{k-\nu} - P_k R_{k-1}\delta_{k-\nu-1}, \end{aligned} \tag{3.42}$$

where, for all $k_0 - \nu \leq k \leq k_0 - 1$, we define $\delta_k \triangleq 0$. We show numerically that there exists $\delta_{k_0} \in \mathbb{R}^n$ such that δ_k given by (3.42) does not decay to zero.

We now analyze the numerical stability of Algorithm 1, that is, we analyze the propagation of a perturbation in x_{k_0} at step k_0 assuming that, for all $k > k_0$, x_k is updated using (3.16). For all $k > k_0$, let $\bar{x}_k$ denote the SW-VR-RLS minimizer given by Algorithm 1, where the initial condition is $\bar{x}_{k_0} \triangleq x_{k_0} + \gamma$, where x_{k_0} is the SW-VR-RLS minimizer given by Algorithm 1 at step k_0. Thus, it follows from (3.16) that, for all $k \geq k_0$, $\bar{x}_k$ satisfies

$$\bar{x}_k = -\tfrac{1}{2} P_k \left(\sum_{i=k-r}^{k} b_i - 2R_k \bar{\alpha}_k \right), \tag{3.43}$$

where, for all $k_0 \leq k \leq k_0 + \nu - 1$, $\bar{\alpha}_k \triangleq \alpha_k$, and, for all $k \geq k_0 + \nu$, $\bar{\alpha}_k \triangleq \bar{x}_{k-\nu}$. For all $k \geq k_0$, define $\delta_k \triangleq \bar{x}_k - x_k$ and note that $\delta_{k_0} = \gamma$. It follows from (3.43) and (3.16) that, for all $k > k_0$,

$$\delta_k = P_k R_k \left(\bar{\alpha}_k - \alpha_k \right) = P_k R_k \delta_{k-\nu}, \tag{3.44}$$

where, for all $k_0 - \nu + 1 \leq k \leq k_0 - 1$, we define $\delta_k \triangleq 0$. For all $k > k_0 + \nu - 1$, define $\Delta_k \triangleq \left[\begin{array}{cccc} \delta_k^{\mathrm{T}} & \delta_{k-1}^{\mathrm{T}} & \cdots & \delta_{k-\nu+1}^{\mathrm{T}} \end{array} \right]^{\mathrm{T}} \in \mathbb{R}^{n\nu}$ and, for all $i \in \{1, \ldots, \nu\}$, let $\Delta_{k,i} \triangleq \delta_{k-i+1}$. Then it follows from (3.44) that, for all $k > k_0 + \nu - 2$,

$$\left[\begin{array}{c} \Delta_{k+1,1} \\ \Delta_{k+1,2} \\ \vdots \\ \Delta_{k+1,\nu} \end{array} \right] = \left[\begin{array}{c} P_{k+1} R_{k+1} \Delta_{k,\nu} \\ \Delta_{k,1} \\ \vdots \\ \Delta_{k,\nu-1} \end{array} \right]. \tag{3.45}$$

Note that $\Delta_k \equiv 0$ is an equilibrium solution of (3.45). The following result shows that, under the assumptions of Theorem 2, the equilibrium solution $\Delta_k \equiv 0$ of (3.45)

is globally asymptotically stable.

Theorem 4. Consider the error system (3.18), (3.19), (3.20), and (3.44). For all $k \geq k_0$, let $T_k \in \mathbb{R}^{n\times n}$ be positive definite, and assume there exist $\varepsilon_1, \varepsilon_2 \in (0,\infty)$ such that, for all $k \geq k_0$, (3.23) holds. Furthermore, for all $k \geq k_0$, let $\xi_k \in \mathbb{R}$, assume that $0 < \inf_{k\geq k_0} \xi_k \leq \sup_{k\geq k_0} \xi_k < \infty$, and define $R_k \triangleq \xi_k T_k$. Then the following statements hold:

(i) $\{L_k\}_{k=k_0+1}^{\infty}$, $\{Q_k\}_{k=k_0+1}^{\infty}$, and $\{P_k\}_{k=k_0}^{\infty}$ are bounded.

(ii) The equilibrium solution $\Delta_k \equiv 0$ of (3.45) is uniformly Lyapunov stable, and, for all $\delta_{k_0} \in \mathbb{R}^n$, $\{\delta_k\}_{k=k_0}^{\infty}$ is bounded.

(iii) Assume that $\{A_k\}_{k=k_0}^{\infty}$ is bounded and there exists $c > 0$ and a positive integer l such that, for all $k \geq k_0 + \nu(l-1) - r$, $cI_n \leq \sum_{i=0}^{l-1} A_{k-\nu i}$. Then, for all $\delta_{k_0} \in \mathbb{R}^n$, $\lim_{k\to\infty} \delta_k = 0$, $\Delta_k \equiv 0$ is the unique equilibrium solution of (3.25), and, furthermore, $\Delta_k \equiv 0$ is globally asymptotically stable.

Proof. (i) Since the update equations for L_k, Q_k, and P_k are identical to those in SW-VR-RLS, (i) follows directly from Theorem 1.

(ii) It follows from (3.44) and (3.24) that, for all $k \geq k_0 + \nu$,

$$\begin{aligned}
\delta_k &= P_k R_k \delta_{k-\nu} \\
&= \left[R_k^{-1} - R_k^{-1}\Phi_k \left(I_{q_k} + \Phi_k^{\mathrm{T}} R_k^{-1}\Phi_k\right)^{-1} \Phi_k^{\mathrm{T}} R_k^{-1}\right] R_k \delta_{k-\nu} \\
&= \delta_{k-\nu} - T_k^{-1}\Phi_k \left(\xi_k I_{q_k} + \Phi_k^{\mathrm{T}} T_k^{-1}\Phi_k\right)^{-1} \Phi_k^{\mathrm{T}} \delta_{k-\nu}.
\end{aligned}$$

The remainder of the proof is analogous to the proof of Theorem 2 from (3.28) onwards with x_k replaced by δ_k, $\tilde{\chi}_k$ replaced by Δ_k, $x_* = 0$, and $\tilde{x}_k$ replaced by δ_k. □

We now numerically test the stability of the single error propagation dynamics for x_k given by (3.42) and (3.44). Let $n = 10$, $r = 5$, $\nu = 1$, and, for all $k \geq -r$,

let the entries of ψ_k be generated from a zero mean Gaussian distribution with unit variance. Furthermore, for all $k \geq -r$, let $A_k = \psi_k \psi_k^{\mathrm{T}}$, and, for all $k \geq 0$, let $R_k = I_{n \times n}$. Moreover, let $\delta_{-1} = 0$, and let δ_0 be generated from a zero mean Gaussian distribution with unit variance. Finally, for all $k \geq 0$, let P_k be given by (3.4). For all $k \geq 1$, Figure 3.7 shows δ_k for (3.42) and (3.44), and shows that, for this example, δ_k given by (3.42) does not decay to zero, whereas δ_k given by (3.44) decays to zero.

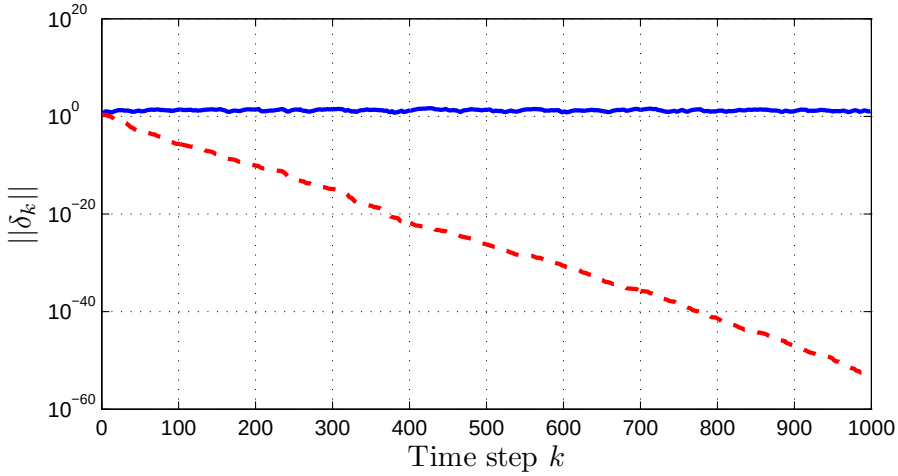


Figure 3.7: This plot shows the solution δ_k of the error-propagation systems for x_k given by (3.42) and (3.44). The solid line indicates the solution to (3.42), whereas the dashed line indicates the solution to (3.44). This plot shows that δ_k given by (3.42) does not decay to zero, whereas δ_k given by (3.44) decays to zero.

Next, we test Algorithm 1 and Algorithm 2 using the same setup as in Section 3.6.1 but with no noise, $x_* = z_1$, and a perturbation in x_k at step $k = 500$. Figure 3.8 shows ε_k for Algorithm 1 and Algorithm 2 with perturbation (dashed line) and without perturbation (solid line) in x_k and shows that, after $k = 500$, for Algorithm 1 with perturbation, ε_k converges to the unperturbed value of ε_k, but for Algorithm 2 with perturbation, ε_k does not converge the unperturbed value of ε_k.

Since the x_k update for Algorithm 2 is derived from the x_k update for Algorithm 1, Figure 3.8 suggests that the derivation of the x_k update for Algorithm 2 introduces the equivalent of a pole on the unit circle at 1 of a linear time-invariant discrete-time

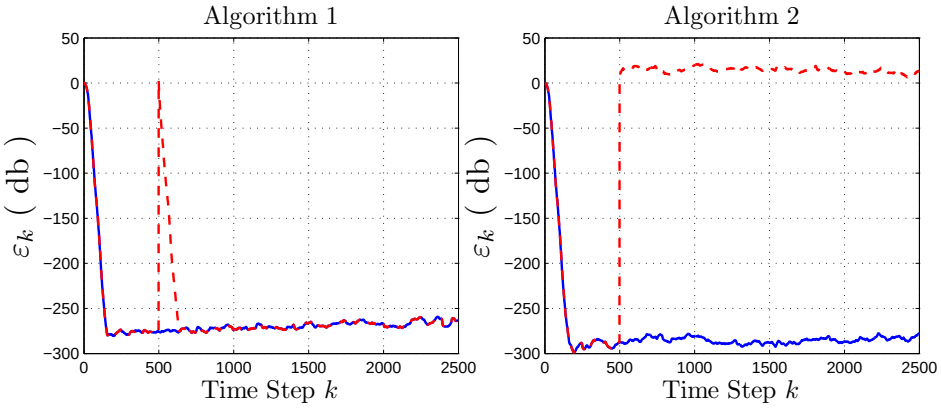


Figure 3.8: This plot shows ε_k for Algorithm 1 and Algorithm 2 with perturbation (dashed line) and without perturbation (solid line) in x_k and shows that, after $k = 500$, for Algorithm 1 with perturbation, ε_k converges to the unperturbed value of ε_k, but for Algorithm 2 with perturbation, ε_k does not converge the unperturbed value of ε_k.

system, due to which a perturbation in x_k does not decay. To illustrate this, let $\kappa \in \mathbb{R}$, for all $k \geq 0$, let $a_k \in \mathbb{R}$ be sampled from a white noise process with a zero-mean Gaussian distribution and variance 0.0025, let $b_k = a_k + 0.5\sin(0.01k)$, and, for all $k \geq 0$, define the stable linear system

$$x_{k+1} = 0.5x_k + b_{k+1} + b_k, \tag{3.46}$$

with the initial condition $x_0 = \kappa$. It follows from (3.46) that $x_k = 0.5x_{k-1} + b_k + b_{k-1}$, and thus

$$b_k = x_k - 0.5x_{k-1} - b_{k-1}. \tag{3.47}$$

Using (3.47) in (3.46) yields, for all $k \geq 0$,

$$x_{k+2} = 1.5x_{k+1} - 0.5x_k + b_{k+2} - b_k, \tag{3.48}$$

with the initial conditions $x_0 = \kappa$ and $x_1 = 0.5\kappa + b_1 + b_0$. Note that (3.48) has a pole at 1. Note that using (3.47) in (3.46) is similar to using (3.10) and (4.10) in (3.16) to obtain

$$\begin{aligned} x_k &= -\tfrac{1}{2}P_k \left(\sum_{i=k-r-1}^{k-1} b_i + b_k - b_{k-r-1} - 2R_k\alpha_k \right) \\ &= -\tfrac{1}{2}P_k \left(-2P_{k-1}^{-1}x_{k-1} + 2R_{k-1}\alpha_{k-1} + b_k - b_{k-r-1} - 2R_k\alpha_k \right), \end{aligned}$$

which is one of the steps in deriving Algorithm 2 from Algorithm 1.

Figure 3.9 shows x_k given by (3.46) and (3.48) with a perturbation at step $k = 200$ (dashed line) and without perturbation (solid line) and shows that, after $k = 200$, for (3.46) with perturbation, x_k converges to the unperturbed value of x_k, but for (3.48) with perturbation, x_k does not converge the unperturbed value of x_k.

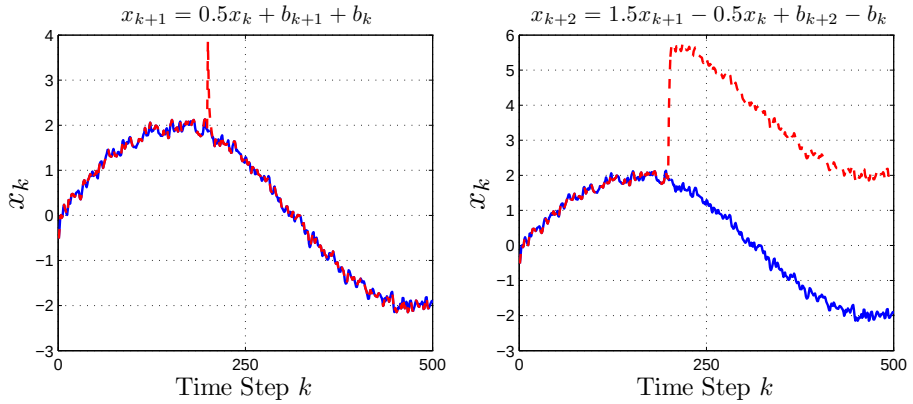


Figure 3.9: This plot shows x_k given by (3.46) and (3.48) with perturbation at step $k = 200$ (dashed line) and without perturbation (solid line) and shows that, after $k = 200$, for (3.46) with perturbation, x_k converges to the unperturbed value of x_k, but for (3.48) with perturbation, x_k does not converge the unperturbed value of x_k.

3.7.2 Numerical Errors in P_k

We now consider the effect of roundoff and quantization errors in P_k. As in the case of x_k, we perturb P_{k_0} at step k_0, and analyze the propagation of this error, assuming all subsequent calculations are performed with infinite-precision arithmetic. Let $\Gamma \in \mathbb{R}^{n\times n}$. For all $k > k_0$, let $\bar{P}_k$ be given by Algorithm 1, where the initial conditions are $\bar{P}_{k_0} = P_{k_0} + \Gamma$, $\bar{Q}_{k_0} = Q_{k_0}$, and $\bar{L}_{k_0} = L_{k_0}$, where P_{k_0}, Q_{k_0}, and L_{k_0} are given by Algorithm 1 at step k_0. Thus, it follows that, for all $k \geq k_0$, $\bar{P}_k$, $\bar{Q}_k$, and $\bar{L}_k$ satisfy

$$\begin{aligned}
\bar{L}_k &= \bar{P}_{k-1} - \bar{P}_{k-1}\phi_k \left(S_k + \phi_k^{\mathrm{T}} \bar{P}_{k-1}\phi_k\right)^{-1} \phi_k^{\mathrm{T}} \bar{P}_{k-1}, \\
\bar{Q}_k &= \bar{L}_k - \bar{L}_k \psi_{k-r-1} \left(-I_{n_{k-r-1}} + \psi_{k-r-1}^{\mathrm{T}} \bar{L}_k \psi_{k-r-1}\right)^{-1} \psi_{k-r-1}^{\mathrm{T}} \bar{L}_k, \\
\bar{P}_k &= \bar{Q}_k - \bar{Q}_k \psi_k \left(I_{n_k} + \psi_k^{\mathrm{T}} \bar{Q}_k \psi_k\right)^{-1} \psi_k^{\mathrm{T}} \bar{Q}_k.
\end{aligned}$$

For all $k \geq k_0$, define $\delta P_k \triangleq \bar{P}_k - P_k$ and note that $\delta P_{k_0} = \Gamma$. We now show numerically that δP_k does not decay to zero. In this work, we mitigate this by resetting SW-VR-RLS at regular intervals.

We consider the same setup as in Example 3.6.3, where the input is white except, for all $k \geq 0$, $R_k = 3 \times 10^3 I_{n\times n}$ and $w_k = 0$. We compare SW-VR-RLS with P_{400} perturbed by a positive definite matrix $\Gamma = \delta P_{400}$ and SW-VR-RLS with no perturbation. Figure 3.10 shows that the error δP_k does not decay.

We now numerically investigate the effect of resetting SW-VR-RLS at regular intervals. The following procedure resets SW-VR-RLS at time step k:

1. x_k is unchanged.
2. For all $i < k$, set $x_i = 0$.
3. Set $\alpha_k = x_k$.
4. For all $i \leq k$, set $A_i = 0$ and $b_i = 0$.

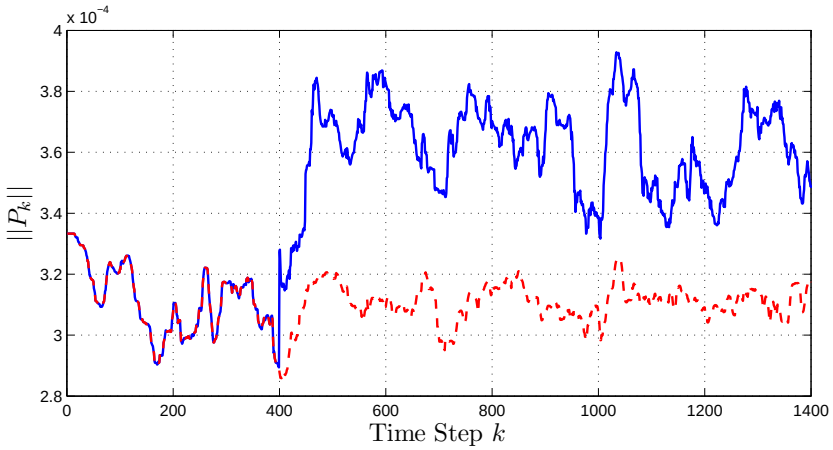


Figure 3.10: This Figure shows $||P_k||$ for SW-VR-RLS with P_k perturbed at $k = 400$ (solid line) and SW-VR-RLS with unperturbed P_k (dashed line). This figure shows that, after P_k is perturbed at $k = 400$, the error between SW-VR-RLS with perturbed P_k and SW-VR-RLS with unperturbed P_k does not decay.

5. Set $P_k = R_k^{-1}$.

Note that the resetting procedure is the same for Algorithm 1 and Algorithm 2 as the Q_k, L_k, and P_k update equations are identical for both algorithms. Furthermore, note that if R_k is a diagonal matrix, then the inverse in Step 5 is $\mathcal{O}(n)$. We now investigate the effect of periodically resetting SW-VR-RLS after k_s steps. For this example, we consider the same setup as in Example 3.6.3 where the input is white. We compare SW-VR-RLS without resetting, and SW-VR-RLS with $k_s = 60$, $k_s = 120$ steps, and $k_s = 300$ steps. We show ε_k for a single trial. Figure 3.11 shows that, if ε_k reaches its asymptotic value and $R_k = R_{\max}$, then ε_k for SW-VR-RLS with covariance resetting does not deviate significantly from SW-VR-RLS without resetting. However, resetting SW-VR-RLS when $R_k = R_{\min}$ and ε_k is adapting quickly yields slower convergence of ε_k to its asymptotic value as compared to SW-VR-RLS without resetting. Note that in all cases, resetting SW-VR-RLS does not introduce large transients in ε_k.

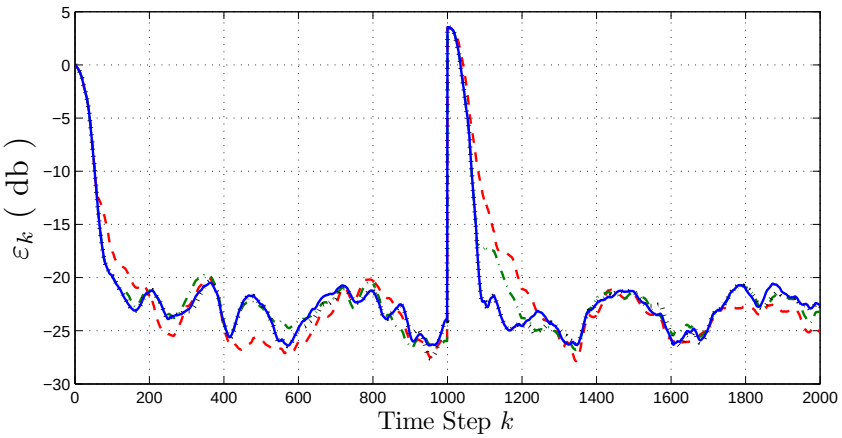


Figure 3.11: Effect of resetting on SW-VR-RLS for $k_s = 60$ (dashed line), $k_s = 120$ (dash-dotted line), $k_s = 300$ (dotted line), and no resetting (solid line). This plot shows that, after ε_k reaches its asymptotic value and $R_k = R_{\max}$, then ε_k for SW-VR-RLS with covariance resetting does not deviate significantly from SW-VR-RLS without resetting.

3.8 RCUIO with SW-VR-RLS based adaptive subsystem update

In this example, we compare RCUIO with the adaptive feedback subsystem updated using standard RLS, given in Section 4.4.1, and RCUIO with the adaptive feedback subsystem updated using SW-VR-RLS. Consider the truth system

$$x(k+1) = Ax(k) + Bu(k),$$

$$y(k) = Cx(k) + v(k),$$

where

$$A = \begin{bmatrix} 0.2 & 2 \\ 0 & 0.4 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix},$$

$$C = \begin{cases} \begin{bmatrix} 0.1 & 0.5 \end{bmatrix}, & k < 2500, \\ \begin{bmatrix} 1.2 & 0.5 \end{bmatrix}, & k \geq 2500, \end{cases}$$

and the unknown input is

$$u(k) = \begin{cases} 5\sin(0.01\pi k), & k < 2500, \\ 6\sin(0.005\pi k), & k \geq 2500. \end{cases}$$

Furthermore, let $v(k)$ be sampled from a white noise process with a zero-mean Gaussian distribution with variance 0.04. For RCUIO, we use an adaptive feedback system of order 50, we set $\tilde{\mathcal{H}} = CB$, and saturate the optimized input $u^*(k-1)$ between ± 10. For RCUIO with standard RLS, we set $\gamma = 0.01$, and for RCUIO with SW-VR-RLS, we set $\alpha_k = x_{k-1}$, $R_k \equiv 30000$, and $r = 0$. Note that at step $k = 2500$, not only does the input change, but also the truth system changes from a minimum-phase system to a nonminimum-phase system. Figures 3.14 and 3.15 show the estimated states and estimated input, respectively, for RCUIO with standard RLS. These figures show that, for $k < 2500$, $\hat{x}(k)$ and $\hat{u}(k)$ converge to $x(k)$ and $u(k)$, respectively, and for $k \geq 2500$, $\hat{x}(k)$ and $\hat{u}(k)$ do not converge to $x(k)$ and $u(k)$, respectively.

Figures 3.12 and 3.13 show the estimated states and estimated input, respectively, for RCUIO with SW-VR-RLS. These figures show that, for $k < 2500$, $\hat{x}(k)$ and $\hat{u}(k)$ converge to $x(k)$ and $u(k)$, respectively, and for $k \geq 2500$, RCUIO with SW-VR-RLS yields smaller steady state mean value of $\hat{x}(k) - x(k)$ and $\hat{u}(k) - u(k)$ than RCUIO with standard RLS.

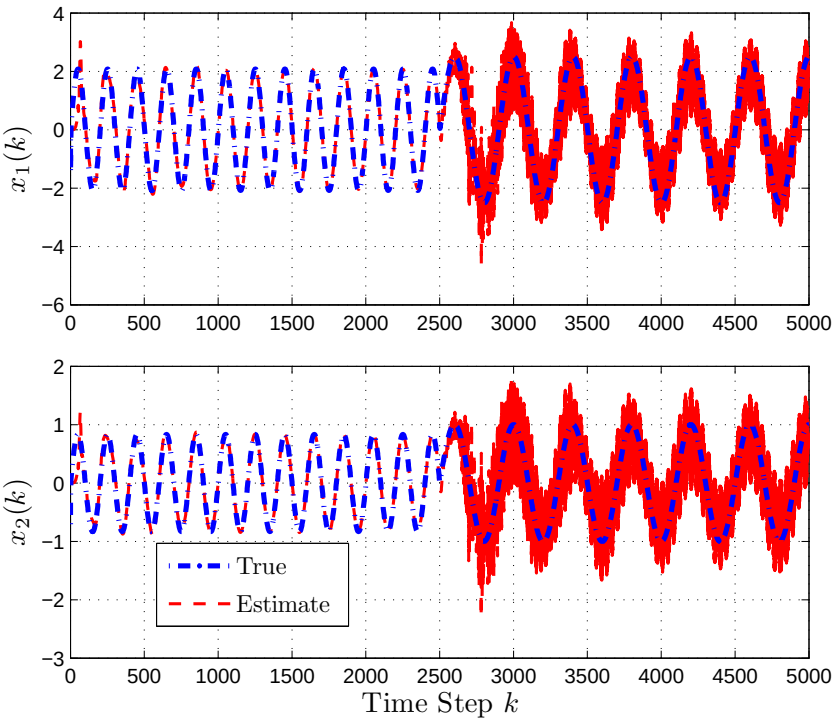


Figure 3.12: True and estimated states for RCUIO with standard RLS. This figure shows that, for $k < 2500$, $\hat{x}(k)$ converges to $x(k)$, and for $k \geq 2500$, $\hat{x}(k)$ does not converge to $x(k)$.

3.9 Conclusions

A sliding-window variable-regularization recursive-least-squares algorithm has been presented. This algorithm allows for a cost function that has a time-varying regularization term, which provides the ability to vary the weighting in the regularization as well as what is being weighted. The convergence properties of the algorithm in the absence of noise were proved, and the effects of window size and regularization were investigated numerically. Furthermore SW-VR-RLS was numerically compared to PAPA and PNLMS for white and colored input noise. Numerical examples demonstrated that time-varying regularization can have a positive impact on the convergence properties. The numerical stability of the algorithm was analyzed analytically and numerically, and it was proved that numerical errors in x_k decay to zero. Furthermore,

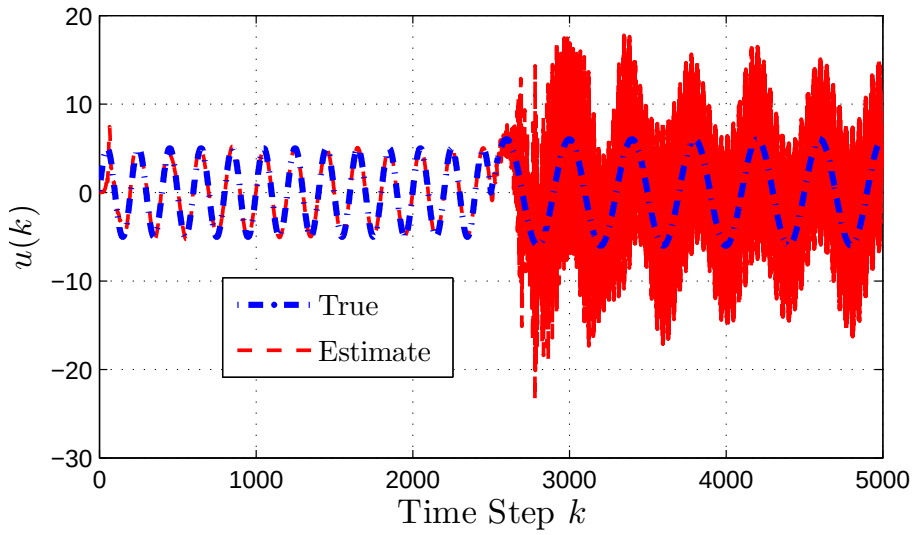


Figure 3.13: True and estimated input for RCUIO with standard RLS. This figure shows that, for $k < 2500$, $\hat{u}(k)$ converges to $u(k)$, and for $k \geq 2500$, $\hat{u}(k)$ does not converge to $u(k)$.

the numerical errors in P_k were mitigated using resetting, and the effect of resetting on SW-VR-RLS was investigated numerically. Future work will concentrate on the use of alternative methods for obtaining P_k recursively to improve the numerical stability of the algorithm, such as QR decomposition, and a computationally efficient method for obtaining an optimal variable R_k.

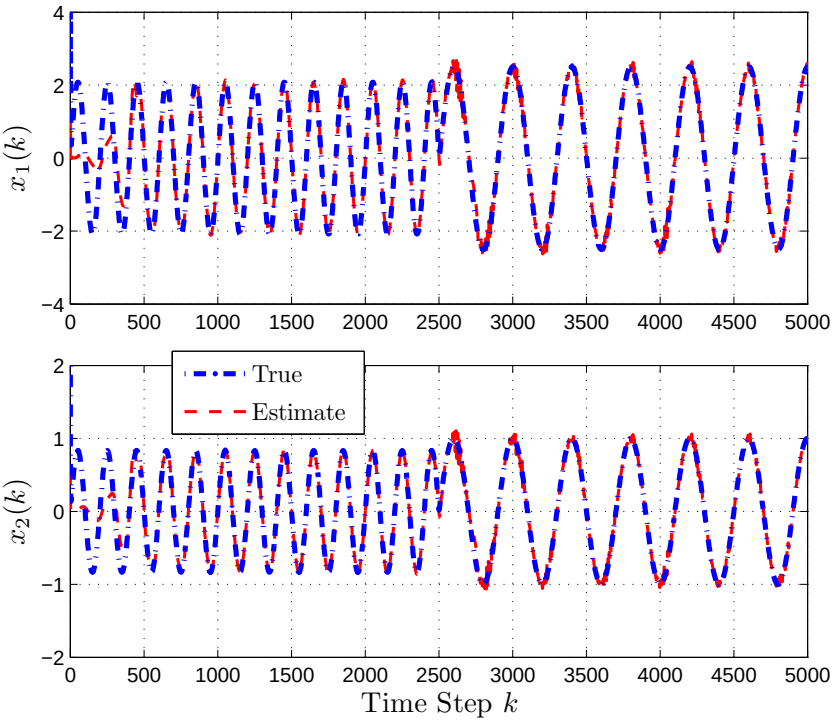


Figure 3.14: True and estimated states for RCUIO with SW-VR-RLS. This figure shows that, for $k < 2500$, $\hat{x}(k)$ converges to $x(k)$, and for $k \geq 2500$, RCUIO with SW-VR-RLS yields smaller steady state mean value of $\hat{x}(k) - x(k)$ than RCUIO with standard RLS.

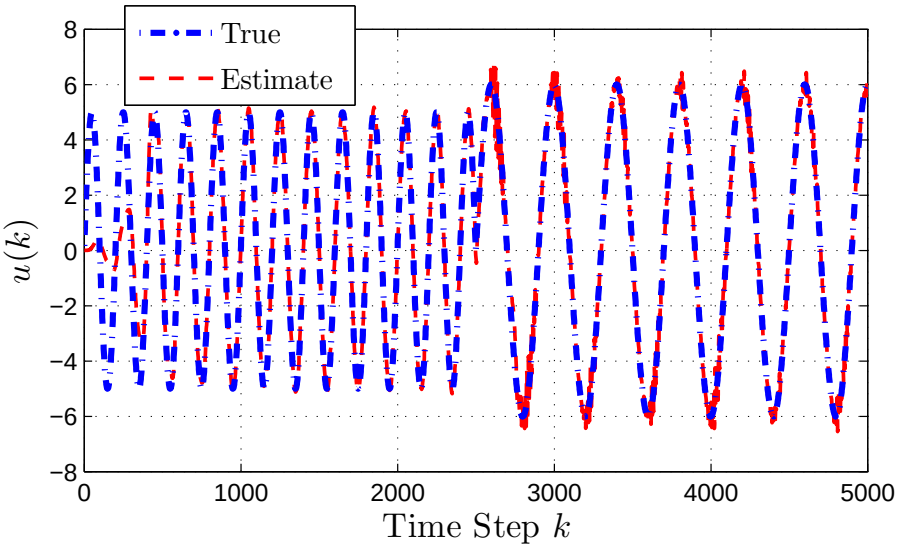


Figure 3.15: True and estimated input for RCUIO with SW-VR-RLS. This figure shows that, for $k < 2500$, $\hat{u}(k)$ converges to $u(k)$, and for $k \geq 2500$, RCUIO with SW-VR-RLS yields smaller steady state mean value of $\hat{u}(k) - u(k)$ than RCUIO with standard RLS.

CHAPTER 4

Growing Window Variable Regularization Recursive Least Squares

In this chapter, we present a growing-window variable-regularization recursive least squares (GW-VR-RLS) algorithm. Standard recursive least squares (RLS) uses a time-invariant regularization. More specifically, the inverse of the initial covariance matrix in classical RLS can be viewed as a regularization term, which weights the difference between the next state estimate and the initial state estimate. The present work allows for time-varying in the weighting as well as what is being weighted. This extension can be used to modulate the speed of convergence of the estimates versus the magnitude of transient estimation errors. Furthermore, the regularization term can weight the difference between the next state estimate and a time-varying vector of parameters rather than the initial state estimate as is required in standard RLS.

4.1 Introduction

Recursive least squares (RLS) is widely used in signal processing, identification, estimation, and control [43, 44, 46–51]. Under ideal conditions, that is, nonnoisy measurements and persistency of the data, RLS is guaranteed to converge to the minimizer of a quadratic function [44, 47]. In practice, the accuracy of the estimates and the rate of convergence depend on the level of noise and persistency of the data. The

goal of this chapter is to extend standard RLS in two ways. First, in standard RLS, the positive-definite initialization of the covariance matrix serves as the weighting of a regularization term within the context of a quadratic optimization. Until at least n measurements are available, this regularization term compensates for the lack of persistency in order to obtain a unique minimizer. Traditionally, the regularization weighting is fixed for all steps of the recursion. In the present work, we derive a growing-window variable-regularization RLS (GW-VR-RLS) algorithm, where the weighting of the regularization term changes at each step. As a special case, the regularization can be decreased in magnitude or rank as the rank of the covariance matrix increases, and can be removed entirely when no longer needed. This ability is not available in standard RLS where the regularization term is weighted by the inverse of the initial covariance at every step.

A second extension presented in this work also involves the regularization term. Specifically, the regularization term in standard RLS weights the difference between the next state estimate and the initial state. In the present work, the regularization term weights the difference between the next state estimate and an arbitrarily chosen time-varying vector of parameters. As a special case, the time-varying vector can be the current state estimate, and thus the regularization term weights the difference between the next state estimate and the current state estimate. This formulation allows us to modulate the rate at which the current estimate changes from step to step.

For these extensions, we derive GW-VR-RLS update equations. While standard RLS entails the update of the state estimate and the covariance matrix, GW-VR-RLS entails the update of an additional symmetric matrix of dimension $n \times n$ to allow for the variable regularization. Thus, GW-VR-RLS entails some additional computational burden relative to classical RLS.

4.2 The Non-Recursive Solution

For all integers i, let $\alpha_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{n\times n}$, and $R_i \in \mathbb{R}^{n\times n}$, where A_i and R_i are positive semidefinite; and, for $i \leq 0$, let $A_i = 0$ and $b_i = 0$. Furthermore, assume that, for all $k \geq 0$, $\sum_{i=0}^{k-1} A_i + R_k$ is positive definite. Thus, R_0 and R_1 are positive definite.

Define the growing-window variable-regularization quadratic cost

$$J_k(x) \triangleq \sum_{i=0}^{k} \left(x^{\mathrm{T}} A_i x + b_i^{\mathrm{T}} x\right) + (x-\alpha_k)^{\mathrm{T}} R_k (x-\alpha_k), \tag{4.1}$$

where $x \in \mathbb{R}^n$ and $x_0 = \alpha_0$ is the minimizer of $J_0(x)$. For all $k \geq 0$, the minimizer x_k of (4.1) is

$$x_k = -\tfrac{1}{2} \left(\sum_{i=0}^{k} A_i + R_k\right)^{-1} \left(\sum_{i=0}^{k} b_i - 2R_k\alpha_k\right), \tag{4.2}$$

where the inverse exists because $\sum_{i=0}^{k} A_i + R_k$ is positive definite.

4.3 The GW-VR-RLS Solution

The equations for the GW-VR-RLS algorithm are derived in this section. Define

$$P_k \triangleq \left(\sum_{i=0}^{k} A_i + R_k\right)^{-1}, \tag{4.3}$$

which is the matrix inverse that appears in the solution (4.2). To write P_k recursively, define

$$Q_k \triangleq \left(\sum_{i=0}^{k-1} A_i + R_k\right)^{-1}, \tag{4.4}$$

where the inverse exists because $\sum_{i=0}^{k-1} A_i + R_k$ is positive definite. Next, consider the decomposition

$$A_k = \psi_k \psi_k^{\mathrm{T}}, \tag{4.5}$$

where $\psi_k \in \mathbb{R}^{n \times n_k}$ and $n_k \triangleq \operatorname{rank}(A_k)$. It follows from (4.3)-(4.5) that $P_k = \left(Q_k^{-1} + \psi_k \psi_k^{\mathrm{T}}\right)^{-1}$. Using the matrix inversion lemma

$$(X + UCV)^{-1} = X^{-1} - X^{-1}U\left(C^{-1} + VX^{-1}U\right)^{-1} VX^{-1}, \tag{4.6}$$

with $X \triangleq Q_k^{-1}$, $U \triangleq \psi_k$, $C \triangleq I_{n_k}$, where I_{n_k} is the identity matrix of dimension n_k, and $V \triangleq \psi_k^{\mathrm{T}}$, it follows that

$$P_k = Q_k\Big(I_n - \psi_k \left(I_{n_k} + \psi_k^{\mathrm{T}} Q_k \psi_k\right)^{-1} \psi_k^{\mathrm{T}} Q_k\Big).$$

To write Q_k recursively, it follows from (4.3) and (4.4) that

$$Q_k = \left(P_{k-1}^{-1} + R_k - R_{k-1}\right)^{-1} = \left(P_{k-1}^{-1} + \phi_k S_k \phi_k^{\mathrm{T}}\right)^{-1}. \tag{4.7}$$

where $R_k - R_{k-1}$ has the decomposition

$$R_k - R_{k-1} = \phi_k S_k \phi_k^{\mathrm{T}}, \tag{4.8}$$

where $\phi_k \in \mathbb{R}^{n \times m_k}$, $m_k \triangleq \operatorname{rank}(R_k - R_{k-1})$, and $S_k \in \mathbb{R}^{m_k \times m_k}$ is a matrix of the form $S_k \triangleq \operatorname{diag}(\pm 1, \ldots, \pm 1)$. Using (4.6) with $X \triangleq P_{k-1}^{-1}$, $U \triangleq \phi_k$, $C \triangleq S_k$, and $V \triangleq \phi_k^{\mathrm{T}}$, it follows from (4.7) that

$$Q_k = P_{k-1}\left(I_n - \phi_k \left(S_k + \phi_k^{\mathrm{T}} P_{k-1} \phi_k\right)^{-1} \phi_k^{\mathrm{T}} P_{k-1}\right).$$

Finally, to write x_k recursively, it follows from (4.2) and (4.3) that

$$x_k = -\tfrac{1}{2} P_k \left(\sum_{i=0}^{k} b_i - 2R_k \alpha_k \right), \tag{4.9}$$

which implies that

$$\sum_{i=0}^{k-1} b_i = -2P_{k-1}^{-1} x_{k-1} + 2R_{k-1}\alpha_{k-1}. \tag{4.10}$$

Furthermore, note that (4.3) and (4.4) imply

$$P_k^{-1} = \sum_{i=0}^{k} A_i + R_k = P_{k-1}^{-1} + A_k + R_k - R_{k-1}. \tag{4.11}$$

Using (4.10) and (4.11), it follows that (4.9) can be written as

$$\begin{aligned}
x_k &= -\tfrac{1}{2} P_k \left(\sum_{i=0}^{k-1} b_i + b_k - 2R_k \alpha_k \right) \\
&= -\tfrac{1}{2} P_k \Big(-2P_{k-1}^{-1} x_{k-1} + 2R_{k-1}\alpha_{k-1} + b_k - 2R_k \alpha_k \Big) \\
&= -\tfrac{1}{2} P_k \Big(-2(P_k^{-1} - A_k - R_k + R_{k-1}) x_{k-1} + 2R_{k-1}\alpha_{k-1} + b_k - 2R_k \alpha_k \Big) \\
&= x_{k-1} - P_k \Big(A_k x_{k-1} + \tfrac{1}{2} b_k + (R_k - R_{k-1}) x_{k-1} + R_{k-1}\alpha_{k-1} - R_k \alpha_k \Big).
\end{aligned}$$

The following result summarizes the GW-VR-RLS algorithm.

Algorithm 3. For each $k > 0$, the unique global minimizer of (4.1) is given by

$$Q_k = P_{k-1} - P_{k-1}\phi_k \left(S_k + \phi_k^{\mathrm{T}} P_{k-1} \phi_k \right)^{-1} \phi_k^{\mathrm{T}} P_{k-1}, \tag{4.12}$$

$$P_k = Q_k - Q_k \psi_k \left(I_{n_k} + \psi_k^{\mathrm{T}} Q_k \psi_k \right)^{-1} \psi_k^{\mathrm{T}} Q_k, \tag{4.13}$$

$$x_k = x_{k-1} - P_k \Big(A_k x_{k-1} + \tfrac{1}{2} b_k + (R_k - R_{k-1}) x_{k-1} + R_{k-1}\alpha_{k-1} - R_k \alpha_k \Big), \tag{4.14}$$

where $x_0 = \alpha_0$ and $P_0 = R_0^{-1}$, ψ_k is given by (4.5), and ϕ_k is given by (4.8).

Note that in the case where the regularization weighting is constant, that is, for all $k \geq 0, R_k = R_0 > 0$, (4.12) simplifies to $Q_k = P_{k-1}$, and thus propagation of Q_k is not required.

4.4 Specializations

4.4.1 Standard RLS

Consider the special case $R_k \equiv R_0$ and $\alpha_k \equiv x_0$. Then the quadratic cost

$$J_k(x) \stackrel{\triangle}{=} \sum_{i=0}^{k} \left(x^{\mathrm{T}} A_i x + b_i^{\mathrm{T}} x\right) + (x - x_0)^{\mathrm{T}} R_0 (x - x_0) \tag{4.15}$$

is minimized by

$$x_k = x_{k-1} - P_k \Big(A_k x_{k-1} + \tfrac{1}{2} b_k\Big), \tag{4.16}$$

$$P_k = P_{k-1} \Big(I_n - \psi_k (I_{n_k} + \psi_k^{\mathrm{T}} P_{k-1} \psi_k)^{-1} \psi_k^{\mathrm{T}} P_{k-1}\Big), \tag{4.17}$$

where $P_0 = R_0^{-1}$. Since the recursive update for Q_k given by (4.13) simplifies to $Q_k = P_k$, standard RLS does not require the update of Q_k.

4.4.2 Standard RLS with $\alpha_k = x_{k-1}$ and $R_k \equiv R_0$

Consider the special case $R_k \equiv R_0$ and $\alpha_k = x_{k-1}$. Then the quadratic cost

$$J_k(x) \stackrel{\triangle}{=} \sum_{i=0}^{k} \left(x^{\mathrm{T}} A_i x + b_i^{\mathrm{T}} x\right) + (x - x_{k-1})^{\mathrm{T}} R_0 (x - x_{k-1})$$

is minimized by

$$P_k = P_{k-1} \Big(I_{n_k} - \psi_k (I_{n_k} + \psi_k^{\mathrm{T}} P_{k-1} \psi_k)^{-1} \psi_k^{\mathrm{T}} P_{k-1}\Big),$$

$$x_k = x_{k-1} - P_k \Big(A_k x_{k-1} + P_0^{-1} (x_{k-2} - x_{k-1}) + \tfrac{1}{2} b_k\Big),$$

where $P_0 = R_0^{-1}$. Note that the update for P_k does not require Q_k.

4.4.3 Standard RLS with forgetting factor

Let $0 < \lambda \leq 1$, and consider the modified cost

$$\bar{J}_k(x) \stackrel{\triangle}{=} \sum_{i=0}^{k} \lambda^{k-i} \big(x^{\mathrm{T}} \bar{A}_i x + \bar{b}_i^{\mathrm{T}} x \big) + (x - x_0)^{\mathrm{T}} \lambda^k \bar{R}_0 (x - x_0),$$

where for $i \geq 0$, $\bar{A}_i = \bar{\psi}_i \bar{\psi}_i^{\mathrm{T}}$. Next, it follows that

$$\begin{aligned} \bar{J}_k(x) &= \lambda^k \sum_{i=0}^{k} \lambda^{-i} \left(x^{\mathrm{T}} \bar{A}_i x + \bar{b}_i^{\mathrm{T}} x \right) + (x - x_0)^{\mathrm{T}} \bar{R}_0 (x - x_0) \\ &= \lambda^k \sum_{i=0}^{k} \left(x^{\mathrm{T}} A_i x + b_i^{\mathrm{T}} x \right) + (x - x_0)^{\mathrm{T}} R_0 (x - x_0), \end{aligned}$$

where $A_i \stackrel{\triangle}{=} \lambda^{-i} \bar{A}_i$, $b_i \stackrel{\triangle}{=} \lambda^{-i} \bar{b}_i$, and $R_0 \stackrel{\triangle}{=} \bar{R}_0$. Therefore, $\bar{J}_k(x) = \lambda^k J_k(x)$, where $J_k(x)$ is given by the traditional RLS quadratic cost (4.15). Minimizing $\bar{J}_k(x)$ is equivalent to minimizing $J_k(x)$. In this case, the minimizer of J_k is given by (4.16) and (4.17); however, the minimizer x_k is expressed in terms of A_k and b_k rather than $\bar{A}_k$ and $\bar{b}_k$. Substituting $A_k = \lambda^{-k} \bar{A}_k$, $b_k = \lambda^{-k} \bar{b}_k$, and $\psi_k = \lambda^{-k/2} \bar{\psi}_k$ into (4.16) and (4.17) yields

$$\begin{aligned} P_k &= P_{k-1} \Big(I_n - \lambda^{-k} \bar{\psi}_k (I_{n_k} + \lambda^{-k} \bar{\psi}_k^{\mathrm{T}} P_{k-1} \bar{\psi}_k)^{-1} \bar{\psi}_k^{\mathrm{T}} P_{k-1} \Big), \\ x_k &= x_{k-1} - P_k \Big(\lambda^{-k} \bar{A}_k x_{k-1} + \tfrac{1}{2} \lambda^{-k} \bar{b}_k \Big). \end{aligned}$$

Next, for $i \geq 0$, define $\bar{P}_i \stackrel{\triangle}{=} \lambda^{-i} P_i$, and it follows that the minimizer of $\bar{J}_k$ is given by

$$\begin{aligned} \bar{P}_k &= \lambda^{-1} \bar{P}_{k-1} \Big(I_n - \bar{\psi}_k (\lambda I_{n_k} + \bar{\psi}_k^{\mathrm{T}} \bar{P}_{k-1} \bar{\psi}_k)^{-1} \bar{\psi}_k^{\mathrm{T}} \bar{P}_{k-1} \Big), \\ x_k &= x_{k-1} - \bar{P}_k \Big(\bar{A}_k x_{k-1} + \tfrac{1}{2} \bar{b}_k \Big), \end{aligned}$$

where $\bar{P}_0 = R_0^{-1}$.

4.4.4 Standard RLS with $\alpha_k = x_{k-1}$ and forgetting factor

Let $0 < \lambda \leq 1$, and consider the modified cost

$$\bar{J}_k(x) \stackrel{\triangle}{=} \sum_{i=0}^{k} \lambda^{k-i} \left(x^{\mathrm{T}} \bar{A}_i x + \bar{b}_i^{\mathrm{T}} x \right) + (x - x_{k-1})^{\mathrm{T}} \lambda^k \bar{R}_0 (x - x_{k-1}),$$

where for $i \geq 0$, $\bar{A}_i = \bar{\psi}_i \bar{\psi}_i^{\mathrm{T}}$. Next, it follows that

$$\begin{aligned} \bar{J}_k(x) &= \lambda^k \sum_{i=0}^{k} \lambda^{-i} \left(x^{\mathrm{T}} \bar{A}_i x + \bar{b}_i^{\mathrm{T}} x \right) + (x - x_{k-1})^{\mathrm{T}} \bar{R}_0 (x - x_{k-1}) \\ &= \lambda^k \sum_{i=0}^{k} \left(x^{\mathrm{T}} A_i x + b_i^{\mathrm{T}} x \right) + (x - x_{k-1})^{\mathrm{T}} R_0 (x - x_{k-1}), \end{aligned} \tag{4.18}$$

where $A_i \stackrel{\triangle}{=} \lambda^{-i} \bar{A}_i$, $b_i \stackrel{\triangle}{=} \lambda^{-i} \bar{b}_i$, and $R_0 \stackrel{\triangle}{=} \bar{R}_0$. Combining the steps in Section 4.4.3 and Section 4.4.4, it follows that the minimizer of $\bar{J}_k$ is given by

$$\begin{aligned} \bar{P}_k &= \lambda^{-1} \bar{P}_{k-1} \Big(I_n - \bar{\psi}_k (\lambda I_{n_k} + \bar{\psi}_k^{\mathrm{T}} \bar{P}_{k-1} \bar{\psi}_k)^{-1} \bar{\psi}_k^{\mathrm{T}} \bar{P}_{k-1} \Big), \\ x_k &= x_{k-1} - \bar{P}_k \Big(\bar{A}_k x_{k-1} + \lambda^{k+1} P_0^{-1} (x_{k-2} - x_{k-1}) + \tfrac{1}{2} \bar{b}_k \Big), \end{aligned}$$

where $P_0 = R_0^{-1}$.

4.5 Convergence Analysis of GW-VR-RLS

In this section, we analyze the GW-VR-RLS algorithm.

Theorem 5. For all $k \geq 0$, let $T_k \in \mathbb{R}^{n \times n}$ be positive definite, and assume there exist $\varepsilon_1, \varepsilon_2 \in (0, \infty)$ such that, for all $k \geq 0$,

$$\varepsilon_1 I_n \leq T_{k+1} \leq T_k \leq \varepsilon_2 I_n. \tag{4.19}$$

Furthermore, for all $k \geq 0$, let $\xi_k \in \mathbb{R}$, assume that $0 < \inf_{k\geq 0} \xi_k \leq \sup_{k\geq 0} \xi_k < \infty$, and define $R_k \triangleq \xi_k T_k$, $\Psi_k \triangleq \left(\sum_{i=0}^{k} A_i\right)^{1/2}$, and $\Omega_k \triangleq \xi_k I_{q_k} + \Psi_k^{\mathrm{T}} T_k^{-1} \Psi_k$. Let $\delta \in \mathbb{R}^n$, let ν be a positive integer and, for all $0 \leq k \leq \nu - 1$, define $\alpha_k \triangleq \delta$, and, for all $k \geq \nu$, define $\alpha_k \triangleq x_{k-\nu}$, where $x_{k-\nu}$ satisfies (4.2). Furthermore, assume there exists a unique $x_* \in \mathbb{R}^n$ such that, for all $k \geq 0$,

$$A_k x_* + \tfrac{1}{2} b_k = 0. \tag{4.20}$$

Then the following statements hold:

(i) Q_k, P_k, and x_k are bounded.

(ii) $\sum_{j=\nu}^{\infty} (x_{j-\nu} - x_*)^{\mathrm{T}} \Psi_j \Omega_j^{-1} \Psi_j^{\mathrm{T}} (x_{j-\nu} - x_*)$ and $\sum_{j=\nu}^{\infty} ||x_j - x_{j-\nu}||^2$ exist.

(iii) $\lim_{k\to\infty} \left(A_k x_k + \frac{1}{2} b_k\right) = 0$, and $\lim_{k\to\infty} \psi_k^{\mathrm{T}} (x_{k-\nu} - x^*) = 0$.

(iv) Assume that $\{A_k\}_{k=0}^{\infty}$ is bounded and there exists $c > 0$ and a positive integer l such that, for all $k \geq \nu(l-1) - r$, $cI_n \leq \sum_{i=0}^{l-1} A_{k-\nu i}$. Then, for all $x_0 \in \mathbb{R}^n$, $\lim_{k\to\infty} x_k = x_*$.

Proof. (i) It follows from the first inequality in (4.19) that, for all $k \geq 0$, $R_k \geq c_1 I_n$, where $c_1 \triangleq \varepsilon_1 \inf_{k\geq 0} \xi_k > 0$. Since, for all $k \geq 0$, A_k is positive semidefinite, it follows from (4.3) that $P_k^{-1} \geq c_1 I_n$, which implies that $0 \leq P_k \leq \frac{1}{c_1} I_n$. Thus, $\{P_k\}_{k=0}^{\infty}$ is bounded. Similarly, it follows from (4.4), for all $k \geq 1$, $Q_k^{-1} \geq c_1 I_n$, which implies that $0 \leq Q_k \leq \frac{1}{c_1} I_n$. Thus, $\{Q_k\}_{k=1}^{\infty}$ is bounded.

Next, we show that x_k is bounded. Since, for all $k \geq \nu$, $\alpha_k = x_{k-\nu}$, it follows from

(4.3), (4.9), and (4.20) that

$$\begin{aligned} x_k &= P_k \left(R_k x_{k-\nu} - \sum_{i=0}^{k} \frac{1}{2} b_i \right) \\ &= P_k \left(P_k^{-1} x_{k-\nu} - \sum_{i=0}^{k} A_i x_{k-\nu} - \frac{1}{2} \sum_{i=0}^{k} b_i \right) \\ &= x_{k-\nu} - P_k \left(\sum_{i=0}^{k} A_i x_{k-\nu} - \sum_{i=0}^{k} A_i x_* \right). \end{aligned} \tag{4.21}$$

Define $\tilde{x}_k \stackrel{\triangle}{=} x_k - x_*$ and $\Psi_k \stackrel{\triangle}{=} \left(\sum_{i=0}^{k} A_i \right)^{1/2}$. Subtracting x_* from (4.21) yields

$$\tilde{x}_k = \tilde{x}_{k-\nu} - P_k \Psi_k \Psi_k \tilde{x}_{k-\nu}. \tag{4.22}$$

Since $P_k = (\Psi_k \Psi_k + R_k)^{-1}$ and $R_k = \xi_k T_k$, it follows from (4.6) that

$$\begin{aligned} P_k &= R_k^{-1} - R_k^{-1} \Psi_k \left(I_n + \Psi_k R_k^{-1} \Psi_k \right)^{-1} \Psi_k R_k^{-1} \\ &= \frac{1}{\xi_k} \left(T_k^{-1} - T_k^{-1} \Psi_k \Omega_k^{-1} \Psi_k T_k^{-1} \right). \end{aligned} \tag{4.23}$$

Using (4.23), it follows from (4.21) that

$$\begin{aligned} \tilde{x}_k &= \tilde{x}_{k-\nu} - \frac{1}{\xi_k} T_k^{-1} \Psi_k \Psi_k \tilde{x}_{k-\nu} + \frac{1}{\xi_k} T_k^{-1} \Psi_k \Omega_k^{-1} \Psi_k T_k^{-1} \Psi_k \Psi_k \tilde{x}_{k-\nu} \\ &= \tilde{x}_{k-\nu} - \frac{1}{\xi_k} T_k^{-1} \Psi_k \Omega_k^{-1} \left(\xi_k I_n + \Psi_k T_k^{-1} \Psi_k \right) \Psi_k \tilde{x}_{k-\nu} + \frac{1}{\xi_k} T_k^{-1} \Psi_k \Omega_k^{-1} \Psi_k T_k^{-1} \Psi_k \Psi_k \tilde{x}_{k-\nu} \\ &= \tilde{x}_{k-\nu} - T_k^{-1} \Psi_k \Omega_k^{-1} \Psi_k \tilde{x}_{k-\nu}. \end{aligned} \tag{4.24}$$

Next, note that

$$\begin{aligned} P_k^{-1}T_k^{-1}\Psi_k\Omega_k^{-1} &= \left(\Psi_k^2 + T_k\xi_k\right)T_k^{-1}\Psi_k\Omega_k^{-1} \\ &= \Psi_k^2T_k^{-1}\Psi_k\Omega_k^{-1} + \xi_k\Psi_k\Omega_k^{-1} \\ &= \Psi_k\left(\Psi_kT_k^{-1}\Psi_k + \xi_kI_n\right)\Omega_k^{-1} \\ &= \Psi_k. \end{aligned} \tag{4.25}$$

Next, define the function

$$V(\tilde{x}_k, k) \stackrel{\triangle}{=} \sum_{i=k-\nu+1}^{k} \xi_k^{-1}\tilde{x}_i^{\mathrm{T}}P_i^{-1}\tilde{x}_i.$$

Using (4.25), the difference $\Delta V_k \stackrel{\triangle}{=} V(\tilde{x}_k, k) - V(\tilde{x}_{k-1}, k-1)$ is given by

$$\begin{aligned} \Delta V_k &= \xi_k^{-1}\tilde{x}_k^{\mathrm{T}}P_k^{-1}\tilde{x}_k - \xi_{k-\nu}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}P_{k-\nu}^{-1}\tilde{x}_{k-\nu}, \\ &= \xi_k^{-1}\left(\tilde{x}_{k-\nu}^{\mathrm{T}} - \tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_k\Omega_k^{-1}\Psi_kT_k^{-1}\right)P_k^{-1}\left(\tilde{x}_{k-\nu} - T_k^{-1}\Psi_k\Omega_k^{-1}\Psi_k\tilde{x}_{k-\nu}\right) \\ &\quad - \xi_{k-\nu}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}P_k^{-1}\tilde{x}_{k-\nu} \\ &= \xi_k^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}P_k^{-1}\tilde{x}_{k-\nu} - \xi_{k-\nu}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}P_{k-\nu}^{-1}\tilde{x}_{k-\nu} - 2\xi_k^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}P_k^{-1}T_k^{-1}\Psi_k\Omega_k^{-1}\Psi_k\tilde{x}_{k-\nu} \\ &\quad + \xi_k^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_k\Omega_k^{-1}\Psi_kT_k^{-1}P_k^{-1}T_k^{-1}\Psi_k\Omega_k^{-1}\Psi_k\tilde{x}_{k-\nu} \\ &= \xi_k^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_k^2\tilde{x}_{k-\nu} - \xi_{k-\nu}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k-\nu}^2\tilde{x}_{k-\nu} \\ &\quad + \tilde{x}_{k-\nu}^{\mathrm{T}}\left(T_k - T_{k-\nu}\right)\tilde{x}_{k-\nu} - 2\xi_k^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_k^2\tilde{x}_{k-\nu} \\ &\quad + \xi_k^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_k\Omega_k^{-1}\Psi_kT_k^{-1}P_k^{-1}T_k^{-1}\Psi_k\Omega_k^{-1}\Psi_k\tilde{x}_{k-\nu} \\ &\leq -\xi_k^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_k^2\tilde{x}_{k-\nu} - \xi_{k-\nu}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k-\nu}^2\tilde{x}_{k-\nu} \\ &\quad + \xi_k^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_k\Omega_k^{-1}\Psi_kT_k^{-1}P_k^{-1}T_k^{-1}\Psi_k\Omega_k^{-1}\Psi_k\tilde{x}_{k-\nu}. \end{aligned}$$

$$
\begin{aligned}
&\leq -\xi_{k-\nu}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k-\nu}^{2}\tilde{x}_{k-\nu} - \xi_{k}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k}^{2}\tilde{x}_{k-\nu} + \xi_{k}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k}\Omega_{k}^{-1}\Psi_{k}T_{k}^{-1}\Psi_{k}^{2}\tilde{x}_{k-\nu} \\
&= -\xi_{k-\nu}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k-\nu}^{2}\tilde{x}_{k-\nu} - \xi_{k}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k}^{2}\tilde{x}_{k-\nu} \\
&\quad + \xi_{k}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k}\Omega_{k}^{-1}\left(\xi_{k}I_{n} - \xi_{k}I_{n} + \Psi_{k}T_{k}^{-1}\Psi_{k}\right)\Psi_{k}\tilde{x}_{k-\nu} \\
&= -\xi_{k-\nu}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k-\nu}^{2}\tilde{x}_{k-\nu} - \xi_{k}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k}^{2}\tilde{x}_{k-\nu} + \xi_{k}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k}\Omega_{k}^{-1}\left(-\xi_{k}I_{n} + \Omega_{k}\right)\Psi_{k}\tilde{x}_{k-\nu} \\
&= -\xi_{k-\nu}^{-1}\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k-\nu}^{2}\tilde{x}_{k-\nu} - \tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k}\Omega_{k}^{-1}\Psi_{k}\tilde{x}_{k-\nu} \qquad (4.26) \\
&\leq -\tilde{x}_{k-\nu}^{\mathrm{T}}\Psi_{k}\Omega_{k}^{-1}\Psi_{k}\tilde{x}_{k-\nu}. \qquad (4.27)
\end{aligned}
$$

Since P_k^{-1} is positive definite, and ξ_k^{-1} is positive, it follows that $V(\tilde{x}_k, k)$ is a positive-definite function of $\tilde{x}_k, \ldots, \tilde{x}_{k-\nu+1}$. Furthermore, since for all $k \geq 0$, $P_k^{-1} \geq c_1 I_n$ and $\inf_{k\geq 0}\left\{\xi_k^{-1}\right\} \geq 0$, it follows that $\tilde{x}_k$ is bounded, and thus x_k is bounded.

(ii) It follows from (4.27) that

$$
0 \leq \sum_{j=\nu}^{k}\tilde{x}_{j-\nu}^{\mathrm{T}}\Psi_{j}\Omega_{j}^{-1}\Psi_{j}^{\mathrm{T}}\tilde{x}_{j-\nu} \leq -\sum_{j=\nu}^{k}\Delta V_{j} = V(\tilde{x}_{\nu-1}, \nu-1) - V(\tilde{x}_{k}, k) \leq V(\tilde{x}_{\nu-1}, \nu-1).
$$

Hence, the nondecreasing sequence $\{\sum_{j=\nu}^{k}\tilde{x}_{j-\nu}^{\mathrm{T}}\Psi_{j}\Omega_{j}^{-1}\Psi_{j}^{\mathrm{T}}\tilde{x}_{j-\nu}\}_{k=\nu}^{\infty}$ is bounded, and thus $\sum_{j=\nu}^{\infty}\tilde{x}_{j-\nu}^{\mathrm{T}}\Psi_{j}\Omega_{j}^{-1}\Psi_{j}^{\mathrm{T}}\tilde{x}_{j-\nu}$ exists.

Next, for all $k \geq \nu$, define $\mathcal{M}_k \triangleq \sum_{j=\nu}^{k}||x_j - x_{j-\nu}||^2$. It follows from (4.24) that

$$
\begin{aligned}
\mathcal{M}_k &= \sum_{j=\nu}^{k}||T_j^{-1}\Psi_j\Omega_j^{-1}\Psi_j^{\mathrm{T}}\tilde{x}_{j-\nu}||^2 \\
&= \sum_{j=\nu}^{k}\tilde{x}_{j-\nu}^{\mathrm{T}}\Psi_j\Omega_j^{-1}\Psi_j^{\mathrm{T}}T_j^{-2}\Psi_j\Omega_j^{-1}\Psi_j^{\mathrm{T}}\tilde{x}_{j-\nu} \\
&\leq \sum_{j=\nu}^{k}\|T_j^{-1}\|\tilde{x}_{j-\nu}^{\mathrm{T}}\Psi_j\Omega_j^{-1}\Psi_j^{\mathrm{T}}T_j^{-1}\Psi_j\Omega_j^{-1}\Psi_j^{\mathrm{T}}\tilde{x}_{j-\nu}.
\end{aligned}
$$

Note that, for all $k \geq \nu$, $\|T_k^{-1}\| \leq \|\frac{1}{\varepsilon_1} I_n\| = \frac{1}{\varepsilon_1}$. Therefore,

$$\begin{aligned}
\mathcal{M}_k &\leq \frac{1}{\varepsilon_1} \sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}} \Psi_j \Omega_j^{-1} \left(\xi_j I_{q_j} + \Psi_j^{\mathrm{T}} T_j^{-1} \Psi_j - \xi_j I_{q_j} \right) \Omega_j^{-1} \Psi_j^{\mathrm{T}} \tilde{x}_{j-\nu} \\
&= \frac{1}{\varepsilon_1} \sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}} \Psi_j \Omega_j^{-1} \left(\Omega_j - \xi_j I_{q_j} \right) \Omega_j^{-1} \Psi_j^{\mathrm{T}} \tilde{x}_{j-\nu} \\
&= \frac{1}{\varepsilon_1} \sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{T} \Psi_j \Omega_j^{-1} \Psi_j^{\mathrm{T}} \tilde{x}_{j-\nu} - \frac{1}{\varepsilon_1} \sum_{j=\nu}^{k} \xi_j \tilde{x}_{j-\nu}^{\mathrm{T}} \Psi_j \Omega_j^{-2} \Psi_j^{\mathrm{T}} \tilde{x}_{j-\nu} \\
&\leq \frac{1}{\varepsilon_1} \sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}} \Psi_j \Omega_j^{-1} \Psi_j^{\mathrm{T}} \tilde{x}_{j-\nu}.
\end{aligned}$$

Since $\sum_{j=\nu}^{\infty} \tilde{x}_{j-\nu}^{\mathrm{T}} \Psi_j \Omega_j^{-1} \Psi_j^{\mathrm{T}} \tilde{x}_{j-\nu}$ exists, it follows that the nondecreasing sequence $\{\mathcal{M}_k\}_{k=\nu}^{\infty}$ is bounded, and thus $\lim_{k\to\infty} \mathcal{M}_k$ exists, which verifies (ii).

(iii) Since $\inf_{k\geq 0} \xi_k^{-1} > 0$, it follows that there exists $\kappa > 0$ such that, for all $k \geq 0$, $\xi_k \geq \kappa$. It follows from (4.26) that

$$\begin{aligned}
\Delta V_k &\leq -\xi_k^{-1} \tilde{x}_{k-\nu}^{\mathrm{T}} \Psi_{k-\nu}^2 \tilde{x}_{k-\nu} \\
&= -\xi_k^{-1} \tilde{x}_{k-\nu}^{\mathrm{T}} \left(\sum_{i=0}^{k-\nu} A_i \right) \tilde{x}_{k-\nu} \\
&\leq -\kappa \tilde{x}_{k-\nu}^{\mathrm{T}} \psi_{k-\nu} \psi_{k-\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}. \qquad (4.28)
\end{aligned}$$

It follows from (i) that

$$0 \leq \sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}} \psi_j \psi_j^{\mathrm{T}} \tilde{x}_{j-\nu} \leq -\sum_{j=\nu}^{k} \Delta V_j = V(\tilde{x}_{\nu-1}, \nu-1) - V(\tilde{x}_k, k) \leq V(\tilde{x}_{\nu-1}, \nu-1),$$

Hence, the nondecreasing sequence $\{\sum_{j=\nu}^{k} \tilde{x}_{j-\nu}^{\mathrm{T}} \psi_j \psi_j^{\mathrm{T}} \tilde{x}_{j-\nu}\}_{k=\nu}^{\infty}$ is bounded, and thus $\sum_{j=\nu}^{\infty} \tilde{x}_{j-\nu}^{\mathrm{T}} \psi_j \psi_j^{\mathrm{T}} \tilde{x}_{j-\nu}$ exists. It follows that $\lim_{k\to\infty} ||\psi_k^{\mathrm{T}} \tilde{x}_{k-\nu}|| = 0$. It follows from (ii)

that $||\tilde{x}_k - \tilde{x}_{k-\nu}|| \to 0$ as $k \to \infty$, and thus

$$\begin{aligned}
||A_k x_k + \tfrac{1}{2} b_k|| &= ||A_k x_k - A_k x_*|| \\
&= ||\psi_k \psi_k^{\mathrm{T}} \tilde{x}_k|| \\
&\leq \kappa ||\psi_k^{\mathrm{T}} \tilde{x}_k|| \\
&= \kappa ||\psi_k^{\mathrm{T}} \tilde{x}_{k-\nu} + \psi_k^{\mathrm{T}} \tilde{x}_k - \psi_k^{\mathrm{T}} \tilde{x}_{k-\nu}|| \\
&\leq \kappa \Big(||\psi_k^{\mathrm{T}} \tilde{x}_{k-\nu}|| + ||\psi_k|| ||\tilde{x}_k - \tilde{x}_{k-\nu}|| \Big) \\
&\leq \kappa \Big(||\psi_k^{\mathrm{T}} \tilde{x}_{k-\nu}|| + \kappa ||\tilde{x}_k - \tilde{x}_{k-\nu}|| \Big) \\
&= \kappa ||\psi_k^{\mathrm{T}} \tilde{x}_{k-\nu}|| + \kappa^2 ||\tilde{x}_k - \tilde{x}_{k-\nu}|| \\
&\to 0 \text{ as } k \to \infty,
\end{aligned}$$

which proves (iii).

(iv) Now, since $\{A_k\}_{k=0}^{\infty}$ is bounded, it follows that $\kappa \stackrel{\triangle}{=} \sup_{k \geq 0} \sigma_{\max}(\psi_k) < \infty$. It thus follows from (ii) and (iii) that

$$\begin{aligned}
||A_{k-\nu} \tilde{x}_k|| &\leq \kappa ||\psi_{k-\nu}^{\mathrm{T}} \tilde{x}_k|| \\
&= \kappa ||\psi_{k-\nu}^{\mathrm{T}} \tilde{x}_{k-\nu} + \psi_{k-\nu}^{\mathrm{T}} \tilde{x}_k - \psi_{k-\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}|| \\
&\leq \kappa ||\psi_{k-\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}|| + \kappa^2 ||\tilde{x}_k - \tilde{x}_{k-\nu}|| \\
&\to 0 \text{ as } k \to \infty. \qquad (4.29)
\end{aligned}$$

Similarly, It follows from (ii) and (4.29) that

$$\begin{aligned}
||A_{k-2\nu} \tilde{x}_k|| &\leq \kappa ||\psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_k|| \\
&= \kappa ||\psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_{k-\nu} + \psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_k - \psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}|| \\
&\leq \kappa ||\psi_{k-2\nu}^{\mathrm{T}} \tilde{x}_{k-\nu}|| + \kappa^2 ||\tilde{x}_k - \tilde{x}_{k-\nu}||. \\
&\to 0 \text{ as } k \to \infty.
\end{aligned}$$

Repeating this argument shows that, for all $i \in \{k-(l-1)\nu,\, k-(l-2)\nu,\, \ldots,\, k-\nu,\, k\}$, $\lim_{k\to\infty} ||A_i\tilde{x}_k|| = 0$. Since, for all $k \geq \nu(l-1) - r$, $cI_n \leq \sum_{i=0}^{l-1} A_{k-\nu i}$, it follows that

$$\begin{aligned} ||\tilde{x}_k|| &\leq \frac{1}{c}\left|\left|\sum_{i=0}^{l-1} A_{k-\nu i}\tilde{x}_k\right|\right| \\ &\leq \frac{1}{c}\sum_{i=0}^{l-1} ||A_{k-\nu i}\tilde{x}_k|| \\ &\to 0 \text{ as } k \to \infty. \end{aligned}$$

Hence, $\lim\limits_{k\to\infty} \tilde{x}_k = 0$. □

4.6 Setup for Numerical Simulations

For all $k \geq 0$, let $x_{k,\mathrm{opt}} \in \mathbb{R}^n$, let $\psi_k \in \mathbb{R}^n$ be generated from a zero mean, unit variance Gaussian distribution, and define

$$\beta_k \triangleq \psi_k^{\mathrm{T}} x_{k,\mathrm{opt}}.$$

Let K be the number of data points. Define

$$\begin{aligned} \sigma_{\psi,i} &\triangleq \sqrt{\frac{1}{K}\sum_{k=1}^{K} \psi_{k,i}^2} \xrightarrow{K\to\infty} 1, \\ \sigma_{\beta} &\triangleq \sqrt{\frac{1}{K}\sum_{k=1}^{K} \beta_k^2} \xrightarrow{K\to\infty} 1, \end{aligned}$$

where $\psi_{k,i}$ is the i^{th} entry of ψ_k. Next, for $i = 1, \ldots, n$, let $N_{k,i} \in \mathbb{R}$, and $M_k \in \mathbb{R}$ be generated from zero mean Gaussian distributions with standard deviations $\sigma_{N,i}$ and σ_M, respectively, where $\sigma_{N,i}$ and σ_M are determined from the signal-to-noise ratio

(SNR). More specifically, for $i = 1, \ldots, n$,

$$\mathrm{SNR}_{\psi,i} \stackrel{\triangle}{=} \frac{\sigma_{\psi,i}}{\sigma_{N,i}}, \quad \text{and}$$
$$\mathrm{SNR}_{\beta} \stackrel{\triangle}{=} \frac{\sigma_{\beta}}{\sigma_{M}},$$

where, for $i = 1, \ldots, n$, $\sigma_{N,i} = \sqrt{\frac{1}{K}\sum_{k=1}^{K} N_{k,i}^2}$ and $\sigma_M = \sqrt{\frac{1}{K}\sum_{k=1}^{K} M_k^2}$.

Finally, for $k \geq 0$, define $A_k \stackrel{\triangle}{=} (\psi_k + N_k)(\psi_k + N_k)^{\mathrm{T}}$ and $b_k \stackrel{\triangle}{=} -2(\beta_k + M_k)(\psi_k + N_k)$, where $N_k \stackrel{\triangle}{=} [N_{k,1} \quad N_{k,2} \quad \cdots \quad N_{k,7}]^{\mathrm{T}}$ is the noise in ψ_k and M_k is the noise in β_k.

Define

$$z_1 \stackrel{\triangle}{=} \begin{bmatrix} 0.08 & -1.12 & 1.55 & 1.47 & -2.22 & -2.07 & 0.32 \end{bmatrix},$$
$$z_2 \stackrel{\triangle}{=} \begin{bmatrix} -1.11 & -0.19 & 1.09 & -0.21 & 0.45 & 0.23 & -2.48 \end{bmatrix}.$$

Unless otherwise specified, for all $k \geq 0$, $x_{k,\mathrm{opt}} = z_1$, $\alpha_k = x_0$ and $x_0 = 0_{7\times 1}$.

Define the performance

$$\varepsilon_k \stackrel{\triangle}{=} \frac{\|x_{k,\mathrm{opt}} - x_k\|}{\|x_{k,\mathrm{opt}}\|}.$$

4.7 Numerical Simulations of VR-RLS with Noiseless Data

In this section, we investigate the effect of R_k, α_k and λ on Variable Regularization RLS(VR-RLS). Furthermore, in this section, A_k and b_k contain no noise, specifically, for all $k \geq 0$, $N_k = 0_{7\times 1}$ and $M_k = 0$.

4.7.1 Effect of R_k

First, we begin by testing the effect of R_k on convergence of ε_k when R_k is constant. In the following example, we test VR-RLS for three different values of R_k. Specifically,

for all $k \geq 0$, $R_k = I_{7\times 7}$, $R_k = 0.1I_{7\times 7}$ or $R_k = 0.01I_{7\times 7}$. In all three cases, for all $k \geq 0$, A_k and b_k are the same. For this example, Figure 4.1 shows that smaller values of R_k yield faster convergence of ε_k to zero. Note that this effect occurs because decreasing R_k reduces the magnitude of the regularization term in the cost function (4.1).

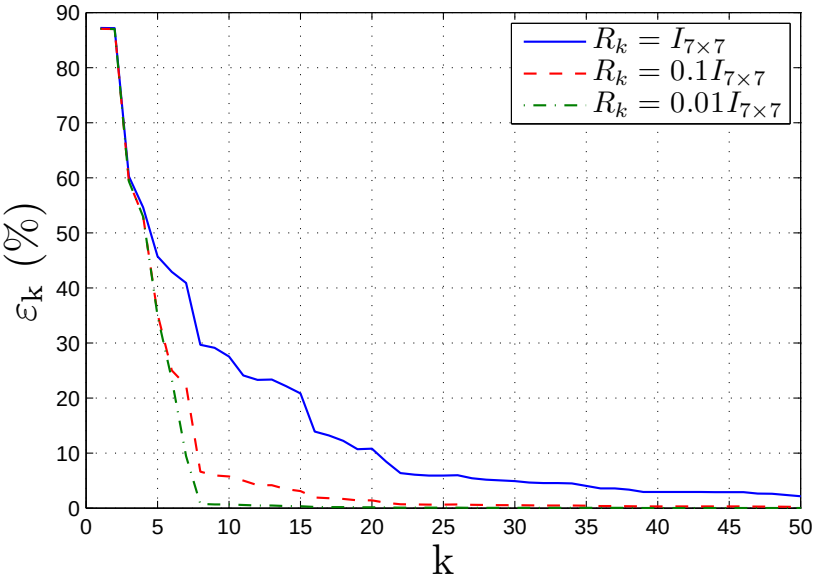


Figure 4.1: Effect of R_k on convergence of x_k to $x_{k,\text{opt}}$. For this example, smaller values of R_k yield faster convergence of ε_k to zero.

Next, we let R_k be constant and positive definite until $\sum_{i=0}^{k-1} A_i$ has full rank, then we let $R_k = 0$. More specifically,

$$R_k = \begin{cases} 0.1I_{7\times 7}, & \text{if rank } \sum_{i=0}^{k-1} A_i < n, \\ 0, & \text{if rank } \sum_{i=0}^{k-1} A_i = n. \end{cases} \tag{4.30}$$

For R_k given by (4.30), if there is no noise in the data, then x_k can converge to $x_{k,\text{opt}}$ in finite time. In particular, if there exists a positive integer N such that $\sum_{i=0}^{N-1} A_i$ has full rank, then, for all $k \geq N$, $x_k = x_{k,\text{opt}}$. Figure 4.2 shows that ε_k converges to zero in finite time when R_k is given by (4.30). In this case for all $k \geq 7$, $\sum_{i=0}^{k-1} A_i$ has full rank. Thus, for all $k \geq 8$, $x_k = x_{k,\text{opt}}$.

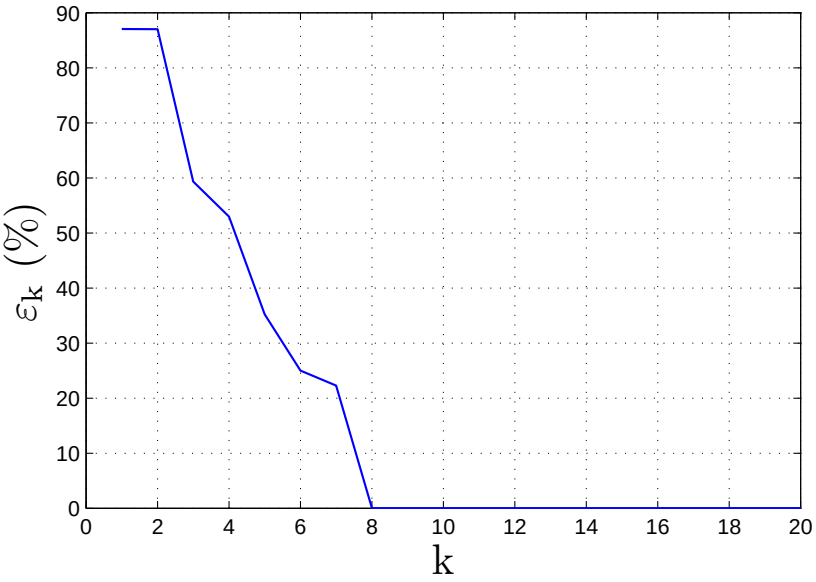


Figure 4.2: Effect of R_k on convergence of x_k to $x_{k,\text{opt}}$. In this example, $\sum_{i=0}^{7} A_i$ has full rank. Therefore, for $k \geq 8$, $R_k = 0$ and $x_k = x_{k,\text{opt}}$.

Next, we pick the smallest R_k such that $\sum_{i=0}^{k} A_i$ is positive definite. More specifically, we conduct the singular value decomposition $USU^{\mathrm{T}} = \sum_{i=0}^{k-1} A_i$, where $U \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{n\times n}$ and has the form

$$S \triangleq \begin{bmatrix} \Gamma_{m\times m} & 0_{m\times(n-m)} \\ 0_{(n-m)\times m} & 0_{(n-m)\times(n-m)} \end{bmatrix},$$

where $\Gamma \in \mathbb{R}^{m\times m}$ contains the m non-zero singular values of $\sum_{i=0}^{k-1} A_i$. Note that the singular value decomposition has the form USU^{T} because $\sum_{i=0}^{k-1} A_i$ is symmetric [72, Corollary 5.4.5]. Next, define

$$\hat{S} \triangleq \begin{bmatrix} 0_{m\times m} & 0_{m\times(n-m)} \\ 0_{(n-m)\times m} & \epsilon I_{(n-m)\times(n-m)} \end{bmatrix},$$

where $\epsilon \in \mathbb{R}^+$. Finally,

$$R_k \stackrel{\triangle}{=} \begin{cases} R_0, & k = 0 \\ U\hat{S}U^{\mathrm{T}}, & \text{if rank } \sum_{i=0}^{k-1} A_i < n, \\ 0, & \text{if rank } \sum_{i=0}^{k-1} A_i = n, \end{cases} \tag{4.31}$$

In the following example we compare VR-RLS with $R_k = I_{3\times 3}$ and R_k given by (4.31) with $\epsilon = 1$. In both cases, for all $k \geq 0$, A_k and b_k are the same. For this example, Figure 4.3 shows that setting R_k given by (4.31) with $\epsilon = 1$ yields faster convergence of ε_k to zero than setting $R_k = I_{7\times 7}$

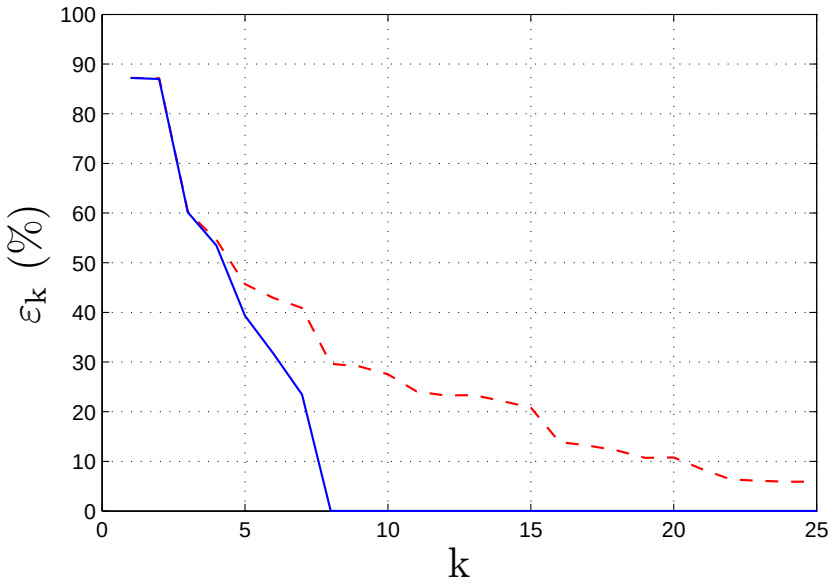


Figure 4.3: Effect of R_k on convergence of x_k to $x_{k,\mathrm{opt}}$. The solid line denotes ε_k with R_k given by (4.31) and the dashed line denotes ε_k with $R_k = I_{7\times 7}$. For this example, setting R_k given by (4.31) with $\epsilon = 1$ yields faster convergence of ε_k to zero than setting $R_k = I_{7\times 7}$

4.7.2 Effect of α_k

Figure 4.4 compares VR-RLS with $\alpha_k = x_{k-1}$ and $\alpha_k = x_0$, where, for all $k \geq 0$, $R_k = I_{7\times 7}$. For this example, setting $\alpha_k = x_{k-1}$ yields faster convergence of ε_k to zero than setting $\alpha_k = x_0$.

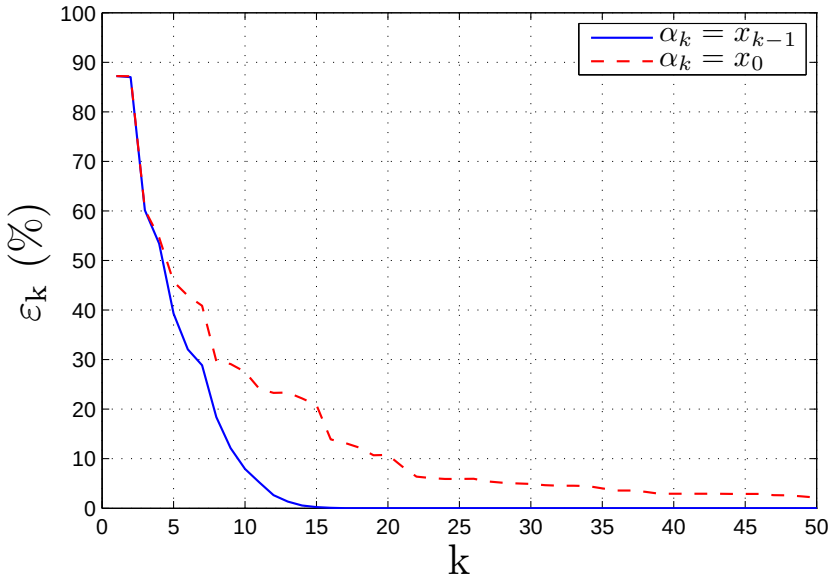


Figure 4.4: Effect of one step regularization on convergence of x_k to $x_{k,\text{opt}}$. For this example, setting $\alpha_k = x_{k-1}$ yields faster convergence of ε_k to zero than setting $\alpha_k = x_0$

4.7.3 Effect of Forgetting Factor

In this section, we examine standard RLS with forgetting factor (as described in Section 4.4.3). Using a forgetting factor allows x_k to approximate $x_{k,\text{opt}}$ when $x_{k,\text{opt}}$ varies with time.

In the following example, we test RLS for three values of λ, specifically λ=1, λ=0.995 or λ=0.9. For all $k \geq 0$, $R_k = 0.1 I_{7\times 7}$ and

$$x_{k,\text{opt}} = \begin{cases} z_1, & 0 \leq k \leq 200 \\ z_2, & k > 200 \end{cases}$$

For this example, Figure 4.5 shows that, for $k \leq 200$, the forgetting factor has negligible impact on the behavior of ε_k. For $k > 200$, smaller values of λ yield faster convergence of ε_k to zero.

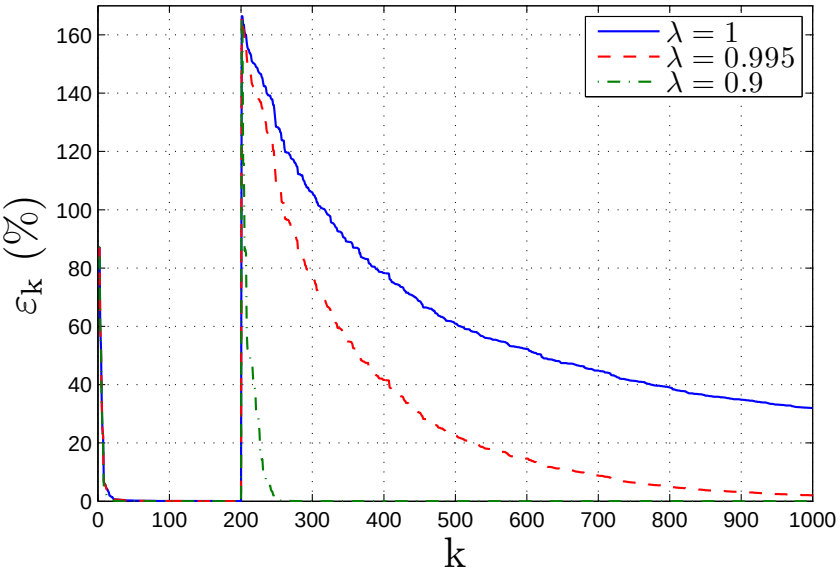


Figure 4.5: Effect of forgetting factor on convergence of x_k to $x_{k,\text{opt}}$. For $k \leq 200$, the forgetting factor has negligible impact on the behavior of ε_k. For $k > 200$, $x_{k,\text{opt}} \neq x_{200,\text{opt}}$, and a smaller value of λ yields faster convergence of x_k to $x_{k,\text{opt}}$.

4.7.4 Loss of Persistency

In this example, we study the effect of loss of persistency on standard RLS with forgetting factor. More specifically, for all $k \geq 500$, $A_k = A_{500}$ and $b_k = b_{500}$. For all $k \geq 0$, $R_k = 0.1I_{7\times 7}$ and λ=0.95. For this example, Figure 4.6 shows that ε_k approaches zero, however, Figure 4.7 shows that $\|P_k\|$ grows without bound after the data lose persistency.

4.8 Numerical Simulations of VR-RLS with Noisy Data

We now investigate the effect of R_k, α_k, and λ on VR-RLS when the data have noise. More specifically, for all $k \geq 0$, M_k and $N_{k,i}$ are generated from zero mean Gaussian distributions with variances depending on $\text{SNR}_{\psi,\text{i}}$ and SNR_{β}, respectively. Figure 4.8 shows the effect of noise on standard RLS for different SNR values. In this example,

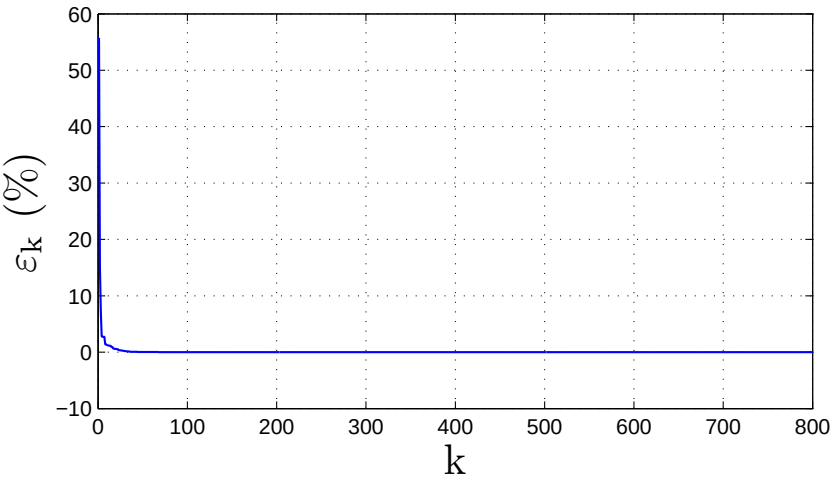


Figure 4.6: Effect of loss of persistency on convergence of x_k to $x_{k,\text{opt}}$. The data lose persistency at the 500th step. In this example, ε_k approaches zero.

a smaller value of SNR yields a larger asymptotic value of ε_k.

In the next example, we examine the convergence of ε_k for standard RLS when ψ_k and β_k have constant bias. We consider three cases of constant bias, specifically, for all $k \geq 0$, $N_k = (0.2)1_{7\times 1}$ and $M_k = 0.2$, $N_k = (0.2)1_{7\times 1}$ and $M_k = 0$ or $N_k = 0_{7\times 1}$ and $M_k = 0.2$. For this example, Figure 4.9 shows that bias increases the asymptotic value of ε_k. Furthermore, bias in β_k yields a higher asymptotic value of ε_k than an equal percent of bias in ψ_k.

4.8.1 Effect of R_k

In this section, we examine the effect of R_k where R_k is constant. In the following example, we test VR-RLS for three different values of R_k. Specifically, for all $k \geq 0$, $R_k = I_{7\times 7}$, $R_k = 0.1I_{7\times 7}$ or $R_k = 0.01I_{7\times 7}$. Furthermore, $\text{SNR}_{\psi,\text{i}} = \text{SNR}_{\beta} = 5$ and, for all $k \geq 0$, A_k and b_k are the same. For this example, Figure 4.10 shows that smaller values of R_k can result in larger peak values of ε_k.

Recall that, Figure 4.1 showed that smaller values of R_k can yield faster convergence of ε_k to zero. However, if the data have noise, then Figure 4.10 shows that the transient response of ε_k can be worse for smaller values of R_k. As the SNR increases,

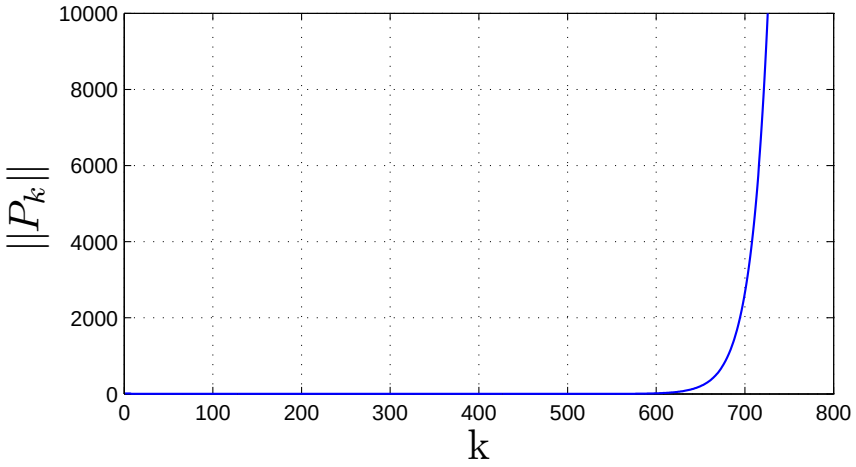


Figure 4.7: Effect of loss of persistency on $\|P_k\|$ for standard RLS with $\lambda = 0.95$. The data lose persistency at the 500th step. In this example, $\|P_k\|$ grows without bound.

Figure 4.10 converges to Figure 4.1.

4.8.2 Effect of α_k

Figure 4.11 compares VR-RLS with $\alpha_k = x_{k-1}$ and $\alpha_k = x_0$, where, $\text{SNR}_{\psi,\text{i}} = \text{SNR}_\beta = 5$ and for all $k \geq 0$ $R_k = I_{7\times 7}$. For this example, Figure 4.11 shows that the transient response of ε_k can be worse for $\alpha_k = x_{k-1}$ than it is for $\alpha_k = x_0$.

Recall that, Figure 4.4 showed that setting $\alpha_k = x_{k-1}$ can yield faster convergence of ε_k to zero than setting $\alpha_k = x_0$. However, if the data have noise, then Figure 4.11 shows that the transient response of ε_k can be worse with $\alpha_k = x_{k-1}$ than it is with $\alpha_k = x_0$. As the SNR increases, Figure 4.11 converges to Figure 4.4.

Next, we compare VR-RLS for different choices of α_k. More specifically, we let $\alpha_k = L_\nu(k)$ where

$$L_\nu(k) \triangleq \begin{cases} x_{k-1}, & 0 < k \leq \nu, \\ x_{k-\nu}, & k > \nu, \end{cases}$$

where ν is a positive integer. In the following example, we test VR-RLS for three different ν. Specifically, $\nu = 1$, $\nu = 5$, $\nu = 10$. In all cases, for all $k \geq 0$, A_k and

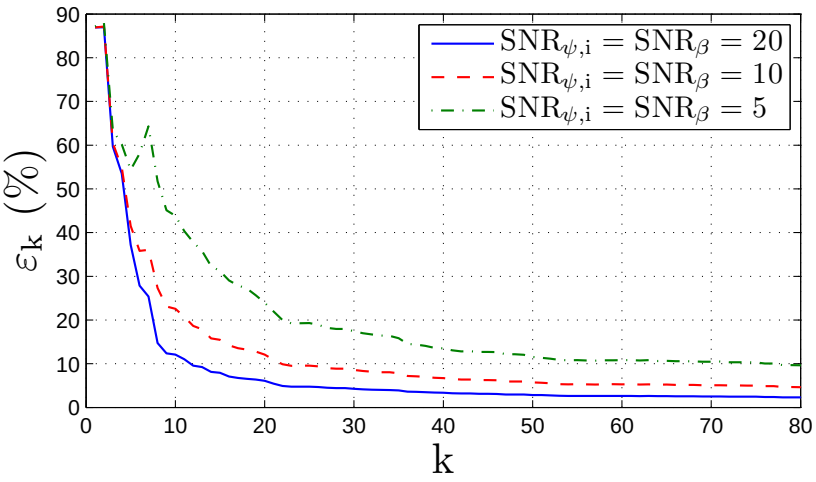


Figure 4.8: Effect of noise on standard RLS. In this example, smaller values of SNR yield larger asymptotic values of ε_k.

b_k are the same, $R_k = I_{7\times 7}$ and $\mathrm{SNR}_{\beta} = \mathrm{SNR}_{\psi,\mathrm{i}} = 5$. For this example, Figure 4.12 shows that larger values of ν can yield better transient performance of ε_k.

Next, we let $\alpha_k = W_{\rho}(k)$ where

$$W_{\rho}(k) \triangleq \begin{cases} x_0, & k = 1, \\ \frac{1}{k-1}\sum_{i=1}^{k-1} x_{k-i}, & 1 < k \leq \rho, \\ \frac{1}{\rho}\sum_{i=1}^{\rho} x_{k-i}, & k > \rho, \end{cases}$$

where ρ is a positive integer. In the following example, we test VR-RLS for three different values of ρ. Specifically, $\rho = 1$, $\rho = 5$, $\rho = 10$. In all cases, for all $k \geq 0$, A_k and b_k are the same, $R_k = I_{7\times 7}$ and $\mathrm{SNR}_{\beta} = \mathrm{SNR}_{\psi,\mathrm{i}} = 5$. For this example, Figure 4.13 shows that larger values of ρ can yield better transient performance of ε_k than smaller values of ρ.

4.8.3 Effect of Forgetting Factor

In this section, we examine standard RLS with forgetting factor. In the following example, we test RLS for three values of λ, specifically λ=1, λ=0.95 or λ=0.9. Let $\mathrm{SNR}_{\psi,\mathrm{i}} = \mathrm{SNR}_{\beta} = 5$, and, for all $k \geq 0$, $R_k = 0.1I_{7\times 7}$. For this example, Figure 4.14

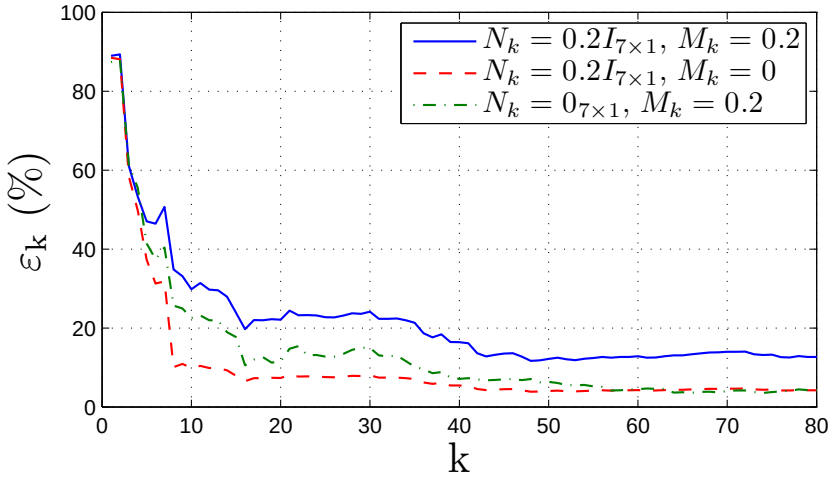


Figure 4.9: Effect of bias on standard RLS. For this example, bias increases the asymptotic value of ε_k. Furthermore, bias in β_k yields a higher asymptotic value of ε_k than an equal percent of bias in ψ_k.

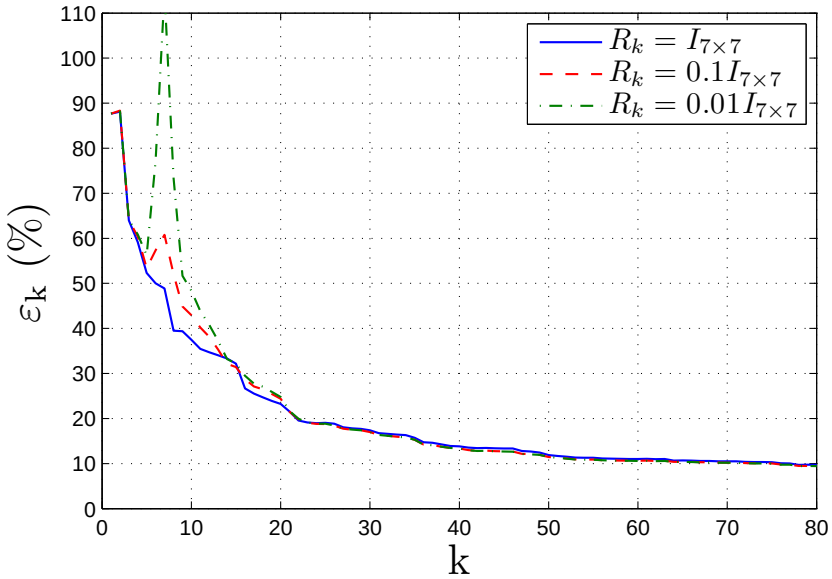


Figure 4.10: Effect of R_k on convergence of x_k to $x_{k,\text{opt}}$. For this example, smaller values of R_k can result in larger peak values of ε_k.

shows that smaller values of λ yield larger asymptotic value of ε_k.

Next, we let

$$x_{k,\text{opt}} = \begin{cases} z_1, & 0 \le k \le 500 \\ z_2, & k > 500 \end{cases}$$

For this example, Figure 4.15 shows that, for $k \le 500$, smaller values of λ yield larger

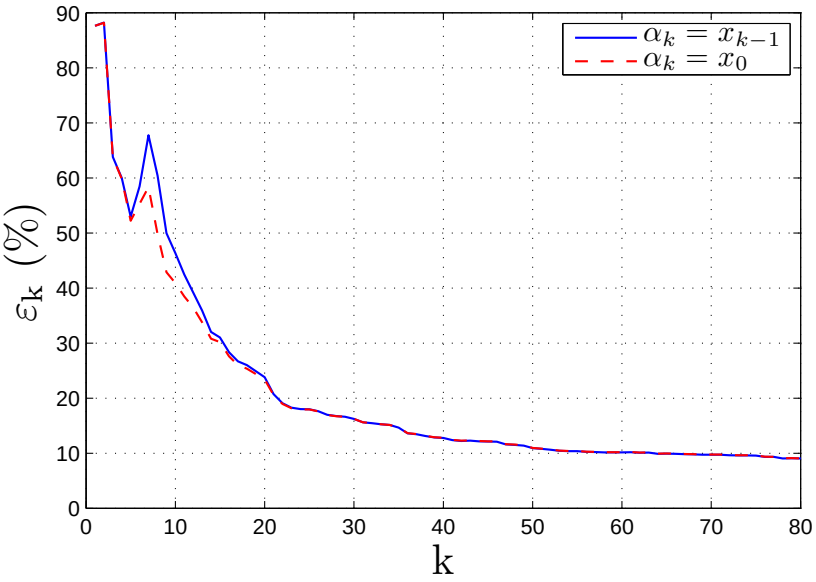


Figure 4.11: Effect of α_k on convergence x_k to $x_{k,\text{opt}}$. For this example, this figure shows that the transient response of ε_k can be worse for $\alpha_k = x_{k-1}$ than it is for $\alpha_k = x_0$.

asymptotic values of ε_k. For $k > 500$, $x_{k,\text{opt}} \neq x_{500,\text{opt}}$, and a smaller value of λ yields faster convergence of ε_k to its asymptotic value.

4.8.4 Loss of Persistency

In this section, we study the effect of loss of persistency on standard RLS with forgetting factor. More specifically, for all $k \geq 500$, $A_k = A_{500}$ and $b_k = b_{500}$. For all $k \geq 0$, $R_k = 0.1I_{7\times 7}$ and λ=0.95 and $\text{SNR}_{\psi,\text{i}} = \text{SNR}_\beta = 5$.

In the first example, the data have noise in both ψ_k and β_k. For this example, Figure 4.16 shows that ε_k increases after the data lose persistency, but ε_k remains bounded.

Next, there is no noise in ψ_k and β_k but only bias in ψ_k. More specifically, for all $k \geq 0$, $N_k = 0.5I_{7\times 1}$ and $M_k = 0$. For this example, Figure 4.17 shows that ε_k grows without bound.

Next, there is no noise in ψ_k and β_k but only bias in β_k. More specifically, for all

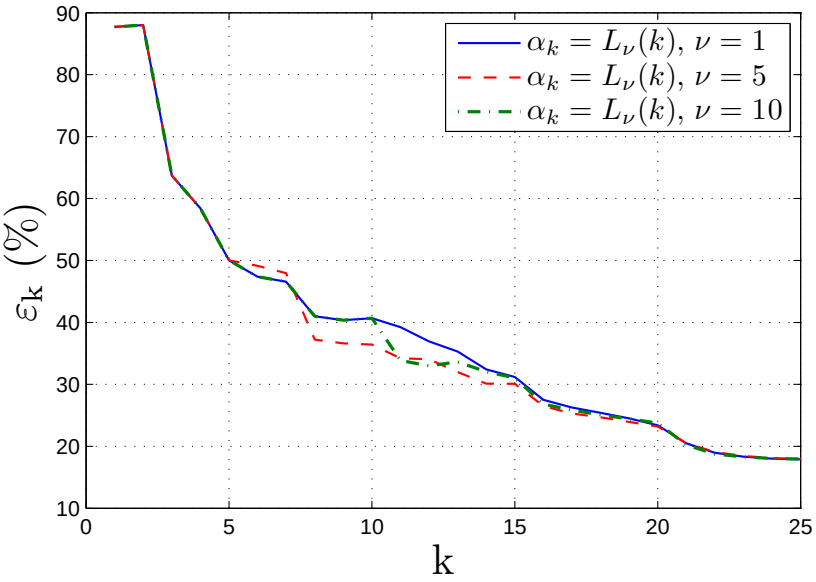


Figure 4.12: Convergence of x_k to $x_{k,\text{opt}}$. For this example, larger values of ν yield better transient performance of ε_k.

$k \geq 0$, $N_k = 0_{7\times 1}$ and $M_k = 0.5$. For this example, Figure 4.18 shows that ε_k grows without bound.

Next, there is noise in β_k but no noise in ψ_k. More specifically, for all $k \geq 0$, $N_k = 0_{7\times 1}$ and $\text{SNR}_\beta = 5$. For this example, Figure 4.19 shows that ε_k grows without bound.

If there is noise in ψ_k but no noise in β_k then $\lambda^{-k}(\sum_{i=0}^{k} A_i + R_k)$ is always full rank. Therefore, ε_k and $\|P_k\|$ remain bounded after the data lose persistency.

4.9 Conclusions

In this chapter, we presented a growing-window variable-regularization recursive least squares (GW-VR-RLS) algorithm. This algorithm allows for a time-varying regularization term in the RLS cost function. More specifically, GW-VR-RLS allows us to vary both the weighting in the regularization as well as what is being weighted, that is, the regularization term can weight the difference between the next state estimate

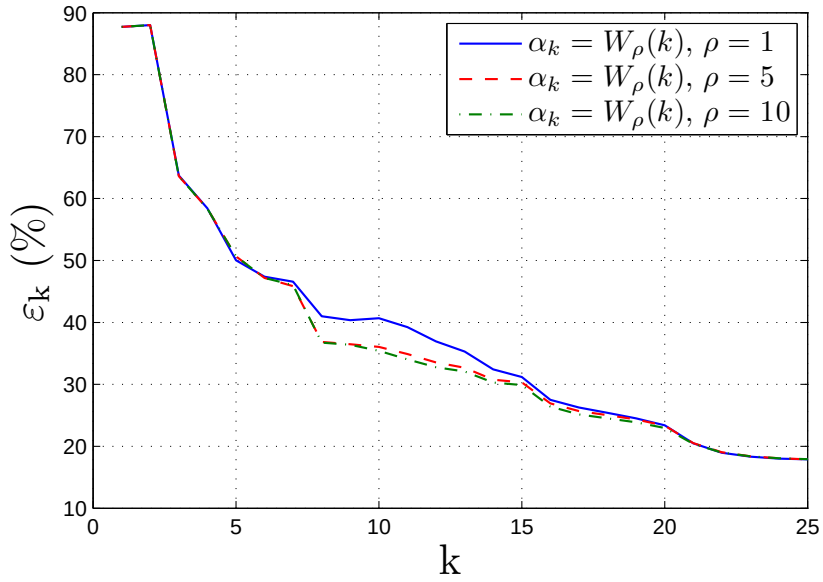


Figure 4.13: Convergence of x_k to $x_{k,\text{opt}}$. In this example, larger values of ρ yield better transient performance of ε_k than smaller values of ρ.

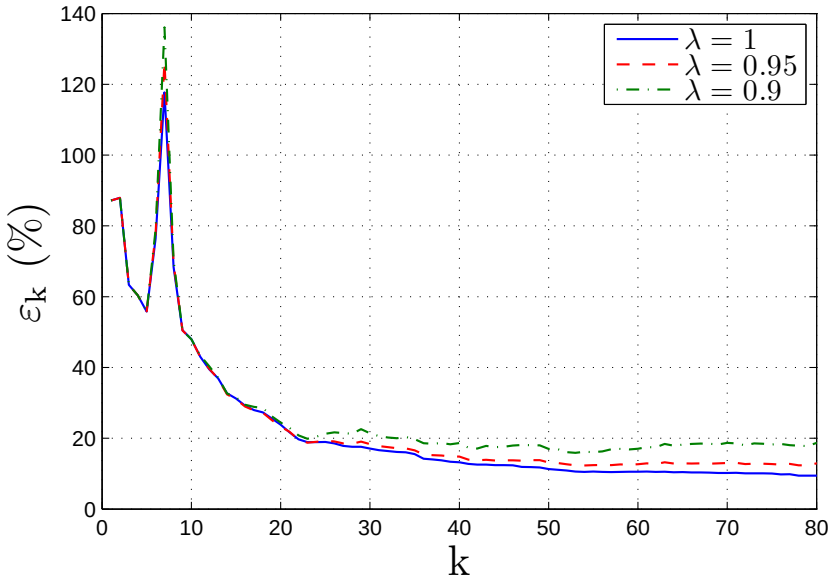


Figure 4.14: Effect of forgetting factor on convergence of x_k to $x_{k,\text{opt}}$. For this example, this figure shows that smaller values of λ yield larger asymptotic value of ε_k.

and a time-varying vector of parameters rather than the initial state estimate.

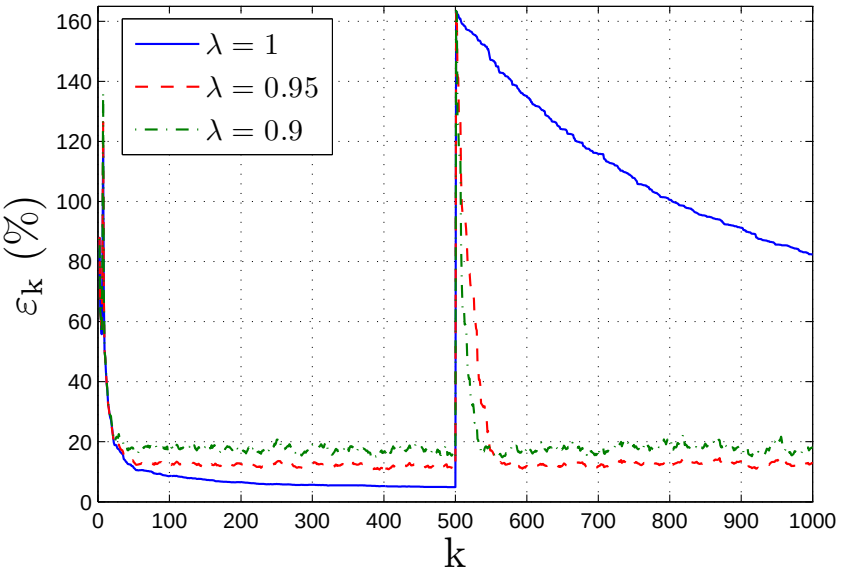


Figure 4.15: Effect of forgetting factor on convergence of x_k to $x_{k,\text{opt}}$. For $k \leq 500$, smaller values of λ yield larger asymptotic values of ε_k. For $k > 500$, $x_{k,\text{opt}} \neq x_{500,\text{opt}}$, and a smaller value of λ yields faster convergence of ε_k to its asymptotic value.

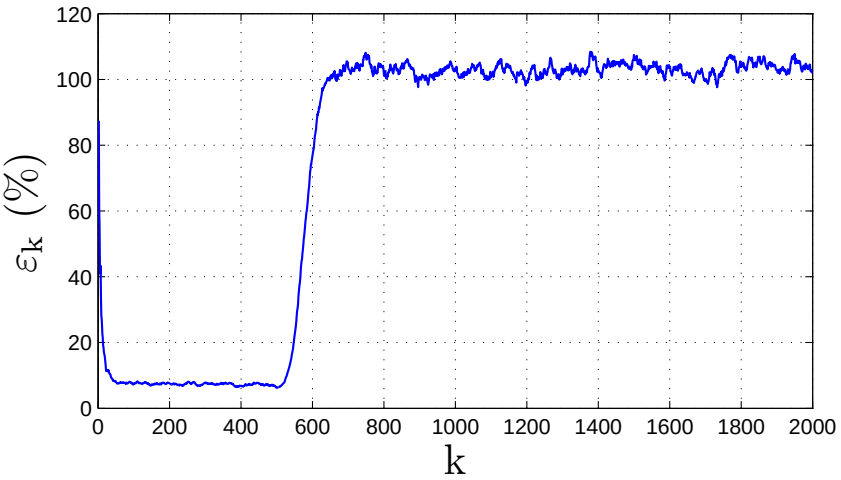


Figure 4.16: Effect of loss of persistency on convergence of x_k to $x_{k,\text{opt}}$. The data lose persistency at the 500th step. In this example, ε_k increases after the data lose persistency, but ε_k remains bounded.

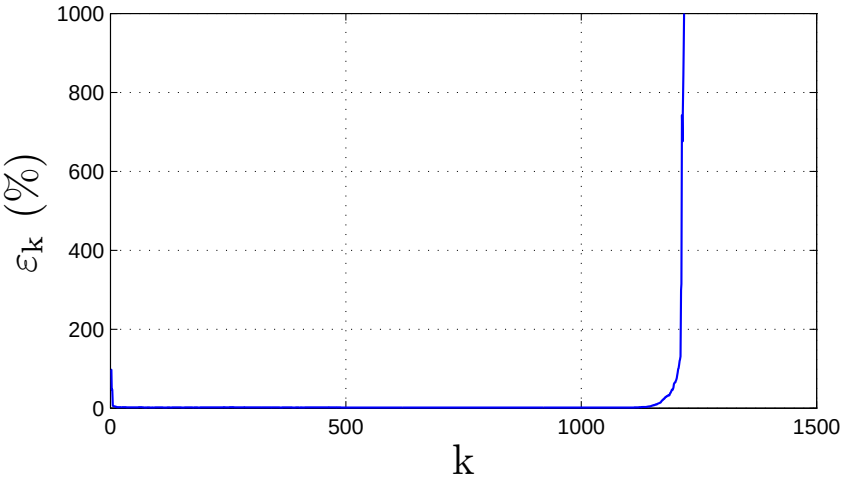


Figure 4.17: Effect of loss of persistency on convergence of x_k to $x_{k,\text{opt}}$. The data lose persistency at the 500th step and there is bias ψ_k. For this example, ε_k grows without bound after the data lose persistency.

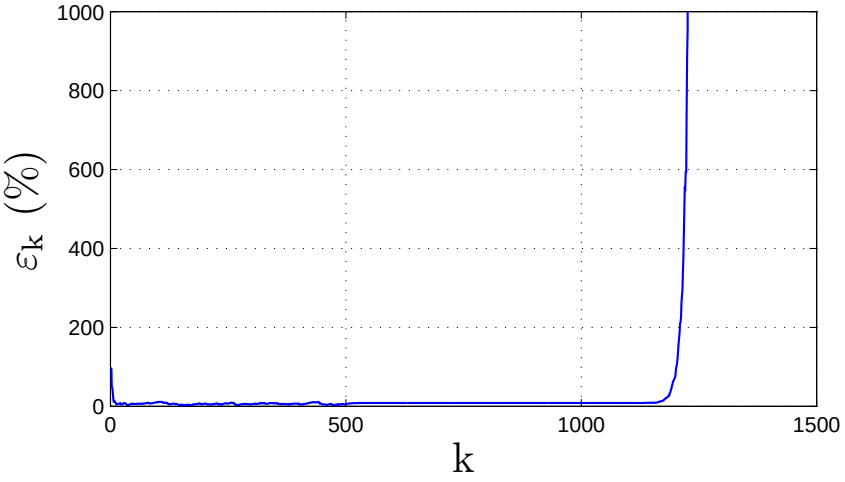


Figure 4.18: Effect of loss of persistency on convergence of x_k to $x_{k,\text{opt}}$. The data lose persistency at the 500th step and there is bias β_k. For this example, ε_k grows without bound after the data lose persistency.

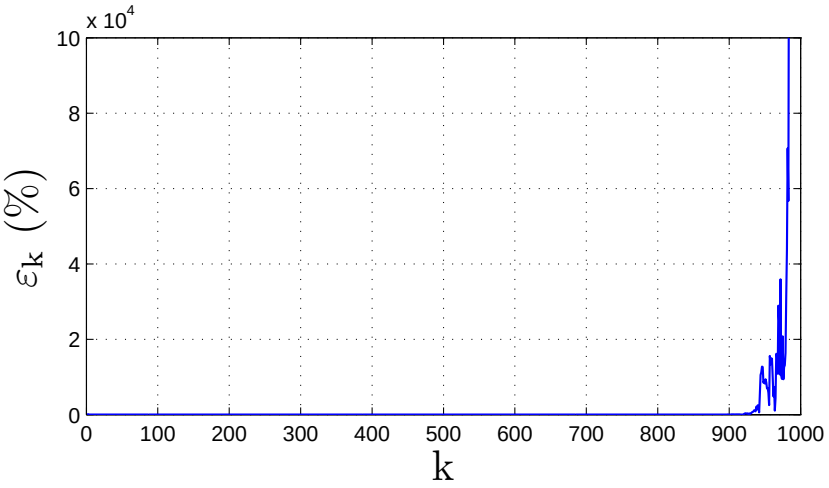


Figure 4.19: Effect of loss of persistency on convergence of x_k to $x_{k,\text{opt}}$. The data lose persistency at the 500th step and there is noise in β_k. For this example, ε_k grows without bound after the data lose persistency.

CHAPTER 5

Input and State Estimation in the Ionosphere-Thermosphere

In this chapter, we formulate RCUIO for state and input estimation in the ionosphere-thermosphere using the Global Ionosphere-Thermosphere Model and real satellite data. Section 5.2 describes the Global Ionosphere-Thermosphere Model and the GRACE and CHAMP satellites that are used in the numerical experiments. In Section 5.3, we formulate RCUIO for GITM and describe the implementation of RCUIO on a multiprocessor GITM simulation. Section 5.4 presents state and input estimation results. Finally, in Section 5.5, we give conclusions and future directions.

5.1 Introduction

This chapter is concerned with modeling and prediction of space weather effects. In the near-Earth environment, the effects of space weather are primarily manifested by the properties of the ionosphere and thermosphere, which influence radio propagation and satellite drag. The sun is one of the primary drivers of the ionosphere and thermosphere. In particular, extreme ultraviolet (EUV) and X-ray radiation produce photo-ionization, which, in turn, through chemistry and heating, drives the formation of the ionosphere and shapes the thermosphere. In addition, the effect of the EUV and X-ray radiation is sufficient to render the ionosphere-thermosphere a strongly

driven system.

Since a significant portion of EUV and X-ray radiation is absorbed by the atmosphere, it is not possible to measure these quantities from the ground. Instead, a proxy is used. The most common proxy for EUV and X-ray radiation is the flux solar irradiance at a wavelength of 10.7 cm ($F_{10.7}$), which is measured (in units of 10^{-22} W Hz^{-1} m^{-2} = 1 solar flux unit (SFU)) by the Dominion radio observatory in Penticton, Canada [66]. A shortcoming of this technique is that $F_{10.7}$ does not have a one-to-one correlation with each of the wavelengths in the EUV and X-ray bands, and thus the measured $F_{10.7}$ is often a misrepresentation of the true solar spectrum.

Although our ultimate goal is to estimate the true flux in multiple EUV and X-ray wavelength bins, a more attainable intermediate goal is to estimate the value of $F_{10.7}$ that best characterizes the ionosphere and thermosphere. The ability to estimate $F_{10.7}$ from alternative measurements can provide a cross check on the available measurements, while also providing an illustrative proof-of-concept demonstration of the adaptive state estimation algorithm described below as a first step toward estimating X-ray and EUV in multiple bands. Furthermore, current models do not fully capture the dynamics of the ionosphere-thermosphere, in which case $F_{10.7}$ can be used as an input to the model for the purpose of eliminating the errors between real measurements and simulated measurements. This study thus attempts to specify $F_{10.7}$ based on simulated measurements of the atmosphere as well as with real satellite data. The specified $F_{10.7}$ can then be used to obtain improved estimates of the state of the ionosphere and thermosphere globally and possibly predict its future evolution. This is a problem of state and input estimation.

To estimate $F_{10.7}$, we use the Global Ionosphere Thermosphere Model (GITM) [32]. GITM simulates the density, temperature, and winds in the thermosphere and ionosphere across the globe from 100 km to 600 km altitude, depending on the solar conditions at the time. The main inputs to GITM are the high-latitude electrodynamics

(i.e., the aurora and the associated electric fields), tides from the lower atmosphere, and the brightness of the sun at various wavelengths, which can be proxied through the use of $F_{10.7}$. GITM solves for the chemistry, dynamics, and thermodynamics of the upper atmosphere self-consistently by accounting for interactions among various species of ions and neutrals.

In this work, we use the retrospective cost unknown input observer (RCUIO) technique given in Chapter 2 to estimate the unknown solar driver $F_{10.7}$ using the Global Ionosphere-Thermosphere Model and satellite measurements. RCUIO assumes that the input to the system is unknown, and uses retrospective optimization to construct an input to the adaptive estimator that minimizes the retrospective cost function given in Chapter 2. The retrospectively optimized input is then used to asymptotically drive the error between the measured output and the estimator output to zero. In this way, RCUIO asymptotically estimates the unknown input to the system and the unknown states of the system. A useful feature of RCUIO is that an explicit nonlinear or linearized model is not required. In addition, unlike ensemble-based data-assimilation algorithms [23, 24, 67], RCUIO uses only one copy of the system model and thus is ensemble-free.

The derivation of the RCUIO algorithm given in Chapter 2 is based on a linear dynamics model, and the modeling information needed to implement the algorithm consists of components of the impulse response. However, since RCUIO does not require an explicit model of the dynamics (which may, for example, be in the form of a computer code as in the case of GITM), RCUIO can be applied to nonlinear systems. In this case, ersatz modeling information can be chosen based on the qualitative behavior of the system, or can be determined by trial and error. Numerical examples show that RCUIO is effective on nonlinear plants, which is presumably due to the fact that extremely limited modeling information is required by the algorithm.

In [68], RCUIO was used to estimate a constant $F_{10.7}$ in 3D GITM using simulated

measurement data, where the measurement was assumed to be at a fixed position in the terrestrial atmosphere. The goal of this work is to extend this application to the case where the measurements are obtained from real or simulated satellites. Furthermore, the ability of the method to estimate a time-varying $F_{10.7}$ using simulated data is demonstrated.

5.2 GITM and Satellites

GITM is a three-dimensional, spherical model that solves the Navier-Stokes equations for the thermosphere in a spherical coordinate system. For the ionosphere, the continuity equation is solved, while the momentum and energy equations are simplified to be steady state. This approach is effective in the ionosphere, where the time scales for changes in the ion and electron velocities and temperatures are extremely small. In the cases presented here, the grid resolution in GITM is set to 5° latitude by 5° longitude. The altitude spacing is roughly 1/3 of the scale height, and is fixed in time. The vertical direction is treated differently from the horizontal direction because of gravity. In GITM, the full vertical momentum equation is computed along with the major neutral species. Frictional terms couple the species and capture Eddy diffusion in the lower thermosphere. In the upper thermosphere all of the species separate out and reach a roughly hydrostatic balance. In the horizontal directions, all species move with the bulk wind speed. Viscosity couples the neutral winds at different altitudes. The magnetic field is given as the International Geophysical Reference Field model. The ion velocities are separated into parallel and perpendicular components, which is a common technique [e.g., Kelley]. GITM is fully described in [32] and has been used to study various geophysical phenomena such as non-hydrostatic wind structures [73]], circulation in the ionosphere, and the influence of ion variability and grid resolution on heating in the thermosphere [74].

The inputs to GITM include the solar luminosity (as described later in this study), tides at the lower boundary, which are specified by the MSIS empirical atmosphere model [75–77], the Horizontal Wind Model [78], the high-latitude electric field, which is specified by the [79] model, and the aurora, which is specified by the [80] model.

The grid structure within GITM is fully parallel and covers the entire planetary surface by using a block-based two-dimensional domain decomposition in the horizontal coordinates [73]. The number of latitude and longitude blocks can be specified at run time in order to modify the horizontal resolution. GITM has been run on up to 256 processors with a resolution as fine as 0.31° latitude by 2.5° longitude over the entire globe with 50 vertical levels, covering a vertical domain from 100 km to roughly 600 km [32]. This flexibility can be used to validate consistency by running the estimator at various levels of resolution. In summary, GITM can be implemented on multiple processors with a nonuniform grid, using a variety of models of electric fields, magnetic fields, auroral particle precipitation, solar EUV and X-ray drivers, and particle energy deposition.

In this work, we use neutral density data from CHAMP (Challenging Mini Satellite Payload [81]) and GRACE (Gravity Recovery and Climate Experiment [82]) satellites. The CHAMP data are used as measurements to obtain state and input estimates, whereas the GRACE data are used as a metric for assessing the accuracy of state estimates. Figure 5.1 shows the trajectories of CHAMP and GRACE from 02:12:00 to 03:40:00 UTC on 2002-11-21. Note that, since only one satellite is used to obtain measurements, this satellite may be located where the solar zenith angle is greater than 90°. When this occurs, the effect of $F_{10.7}$ on the estimator output is significantly delayed [83]. This delay can be mitigated by using multiple satellites, such that the solar zenith angle at the location of at least one satellite is less that 90°. However, in this work we use only one satellite.

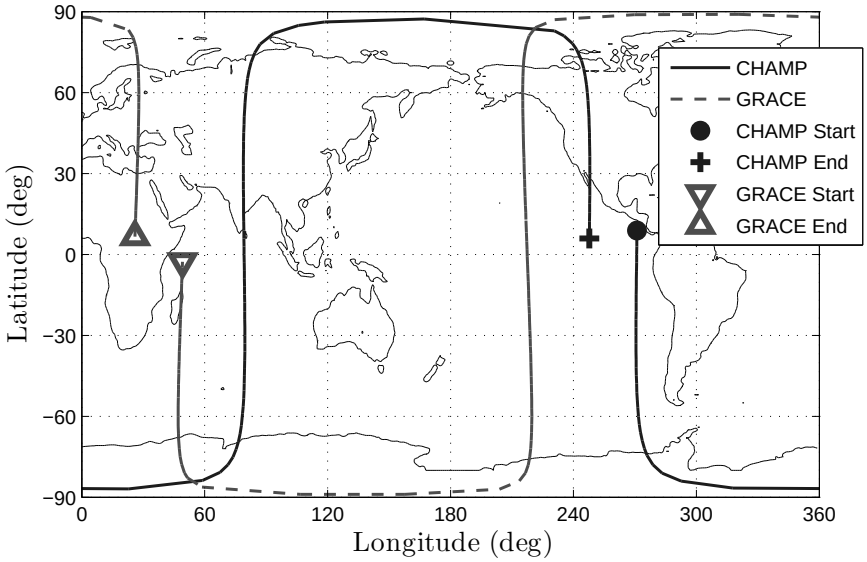


Figure 5.1: This plot shows CHAMP and GRACE trajectories on 2002-11-21.

5.3 RCUIO formulation for GITM

We now consider state and driver estimation in the ionosphere and thermosphere. Consider the GITM model of the ionosphere thermosphere system represented by

$$\hat{x}(k+1) = \hat{f}(\hat{x}(k), v(k), \hat{u}(k)),$$
$$\hat{y}(k) = \hat{h}(\hat{x}(k)),$$

where $\hat{x}(k) \in \mathbb{R}^{31n_g}$ is the estimated state, where n_g is the number of grid points, $\hat{y}(k) \in \mathbb{R}^p$ are the estimator outputs at satellite locations, where p is the number of satellites, and $\hat{u}(k) \in \mathbb{R}^{37}$ is the estimated input. Within this formulation, GITM uses EUVAC, a model of the solar EUV flux [84], to provide an estimate of $u(k)$, given by

$$\hat{u}(k) = \hat{\bar{\mathrm{F}}}_{10.7}(k)E + Q,$$

where $E \in \mathbb{R}^{37}$, $Q \in \mathbb{R}^{37}$, and

$$\hat{\bar{\mathrm{F}}}_{10.7}(k) \triangleq \left(\hat{\mathrm{F}}_{10.7}(k) + \hat{\mathrm{F}}_{10.7\mathrm{a}}(k)\right)/2,$$

where $\hat{\mathrm{F}}_{10.7}(k) \in \mathbb{R}$ is the estimated solar flux at a wavelength of 10.7 cm and $\hat{\mathrm{F}}_{10.7\mathrm{a}} \in \mathbb{R}$ is an 81-day average of $\hat{\mathrm{F}}_{10.7}(k)$. In other words, GITM uses the EUVAC model to estimate $u(k)$ using $\hat{\bar{\mathrm{F}}}_{10.7}(k)$. The matrices E and Q are given in [84, Table 1]. Finally, we define the error $z(k) = y(k) - \hat{y}(k)$.

Our goal is to estimate $x(k)$ by finding an input $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ that minimizes $\bar{J}$. We implement RCUIO on GITM with $\tilde{\mathcal{H}} = H_1$. In order to select a suitable value of H_1, we use the following modifications of RCUIO. First, we use statistical knowledge of the desired rate of change of $\hat{\bar{\mathrm{F}}}_{10.7}$ during a 24-hour period, and design a low-pass filter that limits the rate of change of the output $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ from the adaptive subsystem (2.28). This prevents (2.28) from converging to a system that yields a highly oscillatory $\hat{\bar{\mathrm{F}}}_{10.7}(k)$. Second, we saturate the output from (2.28) in order to prevent GITM from yielding nonphysical states. We set these saturation limits to $70 \leq \hat{\bar{\mathrm{F}}}_{10.7}(k) \leq 400$. Finally, we account for unknown dynamics and measurement noise. More specifically, if $z(k)$ is nonzero, RCUIO will attempt to minimize it by changing $\hat{\bar{\mathrm{F}}}_{10.7}(k)$. However, since the response to a change in $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ on the output $\hat{y}(k)$ is slow and $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ alone cannot counteract the effect of all other drivers in GITM, RCUIO will yield an oscillatory $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ in an attempt to minimize $z(k)$. For some values of H_1, this can cause $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ to alternate between the saturation limits. To prevent this, we apply another low-pass filter to the signal $z(k)$. A schematic for estimating states and input in the ionosphere-thermosphere using GITM and RCUIO is shown in Figure 5.2. Note that in Figure 5.2, the input to (2.28) is $z_{\mathrm{f}}(k)$, and the output from (2.28) is $\hat{\bar{\mathrm{F}}}_{10.7,\mathrm{f}}(k)$.

In this work, we consider the case where $m = 1$, $p = 1$, $g = 1$ and $s = 1$, for which

the computational complexity of RCUIO is $\mathcal{O}(n_c^2)$. An implementation of RCUIO on a four-processor GITM simulation with low resolution (16,200 grid points) from 2002-11-24 to 2002-12-28 with $n_c = 100$ completes in four hours, 3 minutes of which are devoted to RCUIO equations (2.28), (2.44), (2.46), and (2.47).

$u(k)$: Solar EUV and X-ray radiation
$v(k)$: Electro-dynamic Drivers (known)
$y(k)$: Satellite measurements
$\hat{u}(k)$: Estimated solar EUV and X-ray radiation
$\hat{y}(k)$: GITM output at satellite location
P_1, P_2 : Low-pass filters

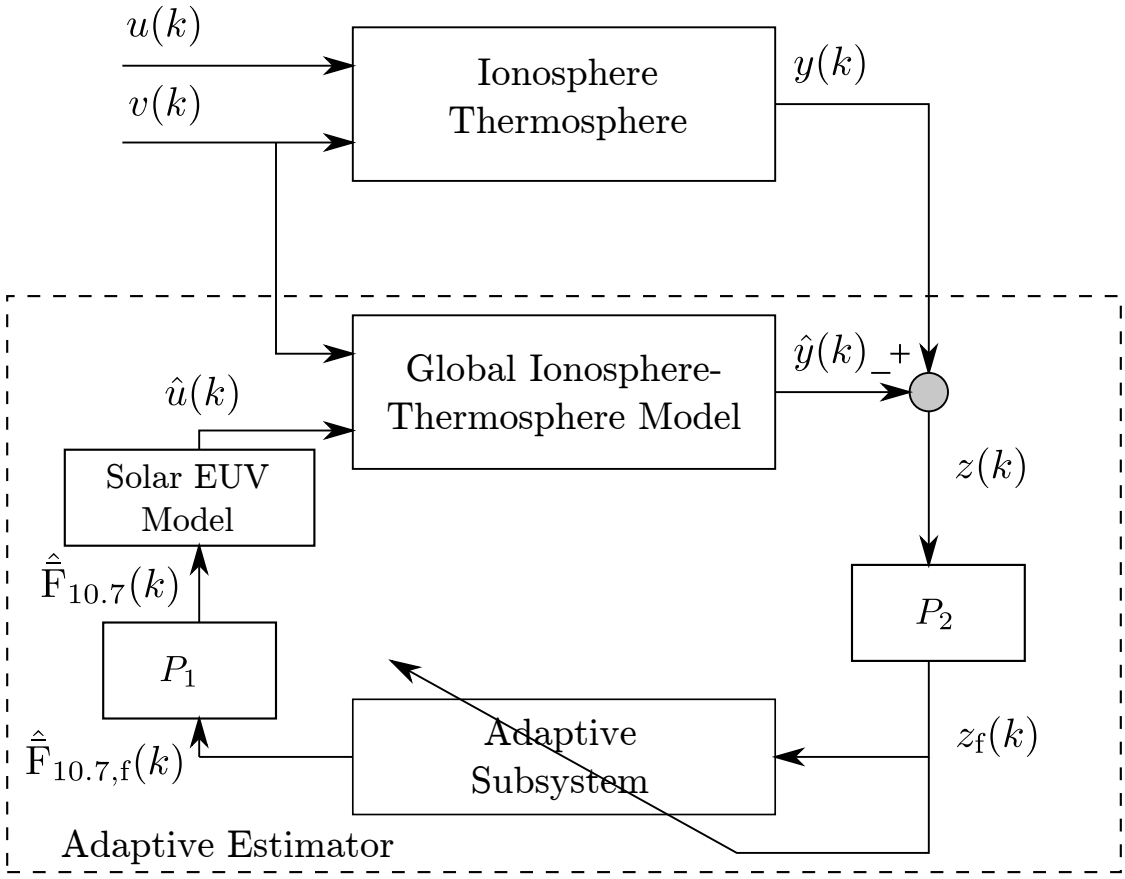


Figure 5.2: RCUIO setup for GITM. The known input $v(k)$ is used in GITM, whereas the unknown input $u(k)$ must be estimated. The objective is to construct an estimate of $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ that, when used in the EUVAC model, yields an estimate $\hat{u}(k)$ that minimizes the retrospective cost function.

The frequency of computing RCUIO update equations and updating $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ de-

pends on the frequency of available measurements. For the purposes of this work, GITM has a fixed time step of $T_s = 2$ sec, but we use a measurement to update $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ only every 60 sec.

Due to the low computational burden of RCUIO compared to GITM, we implement RCUIO on one processor, and send all information needed by RCUIO from other processors to this processor. A flowchart summarizing the implementation of RCUIO with GITM on four processors is shown in Figure 5.3.

5.4 State and Input Estimation Results For GITM

We divide the numerical examples into two main setups. The first setup uses simulated satellite data. In this setup, we run GITM with a specified "true" $\bar{\mathrm{F}}_{10.7}(k)$, and record neutral mass density at CHAMP satellite locations, which we call "truth data" and label it $y(k)$. Furthermore, we record the neutral particle temperature at an altitude of 400 km above Ann Arbor, MI, which is at 42.3° N latitude, 83.7° W longitude, and label it $x_{\mathrm{AA}}(k)$. We use $x_{\mathrm{AA}}(k)$ as a metric for assessing the accuracy of state estimates, and do not use it in RCUIO. Next, we combine RCUIO with GITM, and use $y(k)$ to estimate $\bar{\mathrm{F}}_{10.7}(k)$ and states.

The second setup is the real satellite data case. In this setup, the neutral mass density data measured by CHAMP (the "truth data") is labeled $y(k)$, while the neutral mass density data measured by GRACE is labeled $y_G(k)$. First, we run GITM with the measured $\bar{\mathrm{F}}_{10.7}(k)$ and record the neutral mass density at CHAMP locations and GRACE locations, which are labeled $\hat{y}_m(k)$ and $\hat{y}_{G,m}(k)$, respectively. Next, we combine RCUIO and GITM, and use $y(k)$ to estimate $\bar{\mathrm{F}}_{10.7}(k)$ and states. The neutral mass density output from GITM with RCUIO at CHAMP and GRACE locations are labeled $\hat{y}(k)$ and $\hat{y}_G(k)$, respectively. Note that data from GRACE are used only as a metric for assessing the accuracy of state estimates, and are not used

by RCUIO. We further divide this setup into two cases. First, in RCUIO, we use GITM with photoelectron heating. When photoelectron heating is used in GITM, then the neutral density output from GITM at CHAMP locations using measured $\bar{\mathrm{F}}_{10.7}(k)$ closely matches CHAMP neutral density measurements. However, it should be noted that the photoelectron heating efficiency coefficient that yields low error between GITM and CHAMP is obtained by trial and error, and cannot be calculated or measured. In the second case, in RCUIO, we use GITM without photoelectron heating. In this case, GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$ yields a large error between the outputs from GITM at CHAMP locations and CHAMP measurements. In this case, RCUIO will use $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ as an input to GITM in order to correct the errors between CHAMP measurements and the output from GITM at CHAMP locations, and thus account for the inaccuracies incurred by removing photoelectron heating from GITM.

Let $p(k) \in \mathbb{R}$ be an arbitrary signal, and let T be a positive integer. Then, for all $k \geq T$, define the windowed average of the signal $p(k)$ as

$$\mu_{T,p}(k) \triangleq \frac{1}{T} \sum_{i=k-T+1}^{k} p(i),$$

where T is the interval over which the signal is averaged. Similarly, for all $k \geq T$, define the windowed standard deviation of the signal $p(k)$ as

$$\sigma_{T,p}(k) \triangleq \sqrt{\frac{1}{T} \sum_{i=k-T+1}^{k} \left(p(i) - \mu_{T,p}(i)\right)^2}.$$

Next, Let ℓ be the number of data points, and define the root mean square value of $p(k)$ as

$$\mathrm{RMS}(p) \triangleq \sqrt{\frac{1}{\ell} \sum_{i=1}^{\ell} p(k)^2}.$$

When GITM is used with RCUIO, we keep $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ at a constant value of 100 SFU for the first 24 h, after which RCUIO is turned on. This allows the response due to initial conditions to decay significantly.

We implement GITM on four processors with a resolution of 5° latitude and 5° longitude. The time step for GITM is set at 2 sec, and RCUIO is used to update $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ every 60 sec. In all figures in this section, the vertical black line indicates when RCUIO is switched on. Finally, the numerical experiments consider the period from 2002-11-24 to 2002-12-06.

5.4.1 Simulated Satellite Data Case 1: Constant $\mathbf{F}_{10.7}$

We first consider the case where the truth data are generated by setting, for all $k \geq 0$, $\bar{\mathrm{F}}_{10.7}(k) = 150$ SFU. Since truth data are generated from GITM itself and there is no modeling error, we do not use the filter P_2 so that $z_{\mathrm{f}}(k) = z(k)$. Next, we set the filter P_1 as

$$\hat{\bar{\mathrm{F}}}_{10.7}(k) = 0.922\hat{\bar{\mathrm{F}}}_{10.7}(k-1) + 0.0039\hat{\bar{\mathrm{F}}}_{10.7,\,\mathrm{f}}(k) + 0.0039\hat{\bar{\mathrm{F}}}_{10.7,\,\mathrm{f}}(k-1),$$

which is approximately equal to a 20-min average of $\hat{\bar{\mathrm{F}}}_{10.7,\,\mathrm{f}}(k)$. Recall that $\hat{\bar{\mathrm{F}}}_{10.7,\,\mathrm{f}}(k)$ is the output from the adaptive subsystem as shown in Figure 5.2. Finally, RCUIO parameters are chosen as $\tilde{\mathcal{H}} = H_1 = 0.1$, $\gamma = 100$, and $\eta = 0$. Figure 5.4 shows that $\hat{y}(k)$ converges to within 0.11×10^{-12} kg m^{-3} of $y(k)$ in less than 72 h. The orbital period of CHAMP is 90 min, during which it traverses from the day side to the night side of Earth, causing an oscillatory signal in $y(k)$. Therefore, for all subsequent examples, we show the 90-min windowed mean $\mu_{90,y}(k)$ and standard deviation $\sigma_{90,y}(k)$ of $y(k)$. The standard deviation indicates the amplitude of the variation, while the mean indicates the presence of a bias in the result, both of which RCUIO is attempting to reduce. Figure 5.5 shows $\mu_{90,y}(k)$, $\mu_{90,\hat{y}}(k)$, $\sigma_{90,y}(k)$, and

$\sigma_{90,\hat{y}}(k)$. This plot shows that $\mu_{90,y}(k)$ and $\sigma_{90,y}(k)$ converge to within 0.12×10^{-12} kg m^{-3} of $\mu_{90,\hat{y}}(k)$ and $\sigma_{90,\hat{y}}(k)$ in less than 72 h. In other words, RCUIO corrects both the mean and variance of the output from GITM to match the truth output.

Figure 5.6 shows that $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ converges to within 10 SFU of $\bar{\mathrm{F}}_{10.7}$ in less than 96 h. Figure 5.7 shows the true temperature $x_{\mathrm{AA}}(k)$ and estimated temperature $\hat{x}_{\mathrm{AA}}(k)$ above Ann Arbor, and shows that $\hat{x}_{\mathrm{AA}}(k)$ converges to within 10 K of $x_{\mathrm{AA}}(k)$ in less than 96 h.

As noted in Section 4, if one satellite measurement is available, then the solar zenith angle at the satellite location may become greater than 90°. When this occurs, the response of GITM to a change in $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ is delayed. Figure 5.8 shows the time periods during which the solar zenith angle at the satellite location is greater than 90°, as indicated by the shaded regions. Despite this, RCUIO is able to force the output from GITM to match the true output. In addition, it may be possible to increase the rate of convergence of the estimates by using multiple satellites to reduce the time periods during which the minimum solar zenith angle at all satellites is greater than 90°.

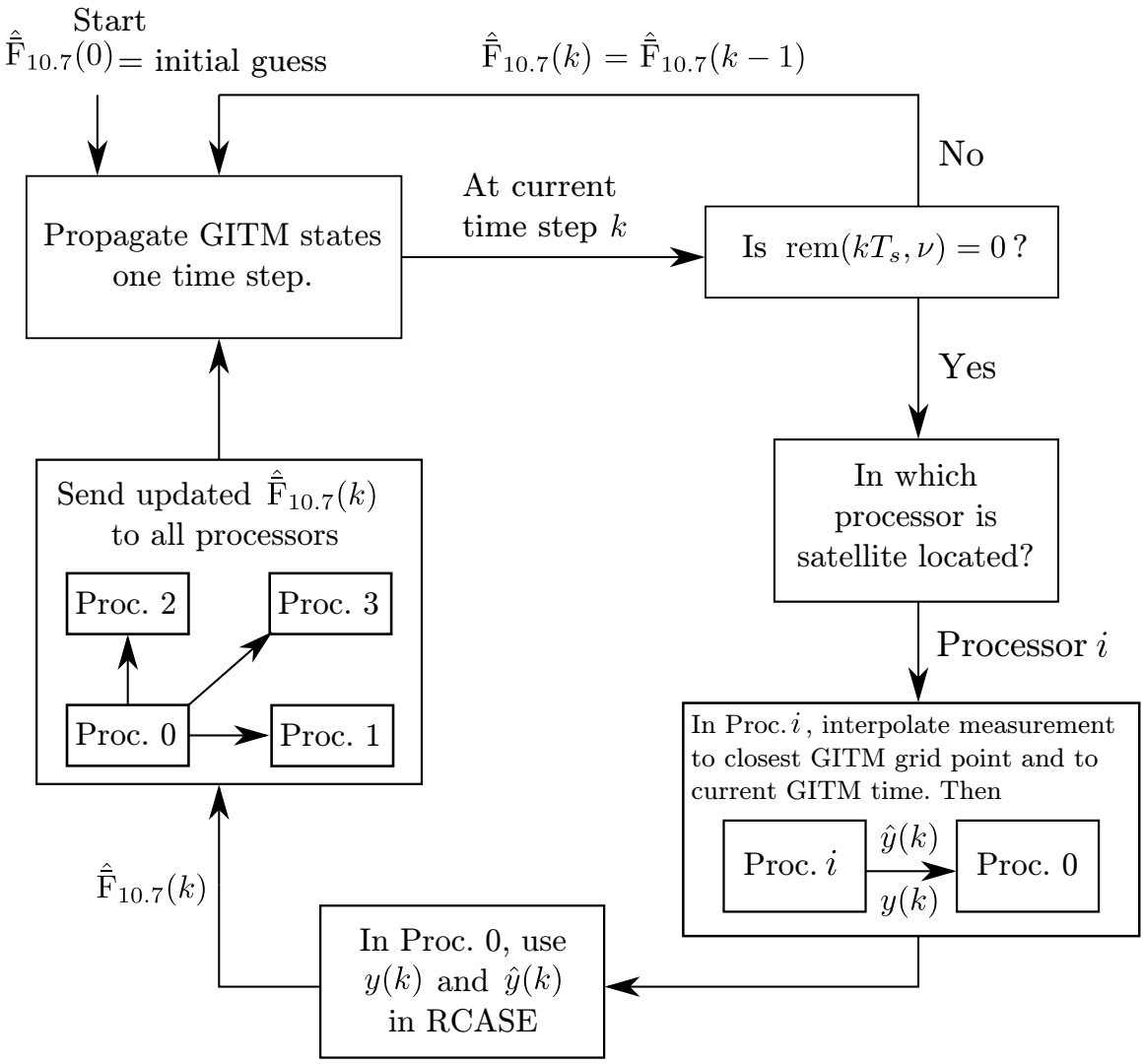


Figure 5.3: Implementation of RCUIO on a four-processor GITM simulation. Each processor simulates the ionosphere-thermosphere dynamics in one quadrant of the Earth's atmosphere. If, at the current time step k, $\text{rem}(kT_s, \nu) = 0$, then the processor that simulates the quadrant of the atmosphere in which the satellite is located uses satellite data to obtain $y(k)$, and uses $\hat{x}(k)$ to obtain $\hat{y}(k)$. These estimates are then sent to Processor 0 where RCUIO uses them to obtain $\hat{\bar{F}}_{10.7}(k)$, which is then sent to all processors to allow GITM to propagate states to the next time step. If $\text{rem}(kT_s, \nu) \neq 0$, then RCUIO is not used and $\hat{\bar{F}}_{10.7}(k)$ remains unchanged from its previous value.

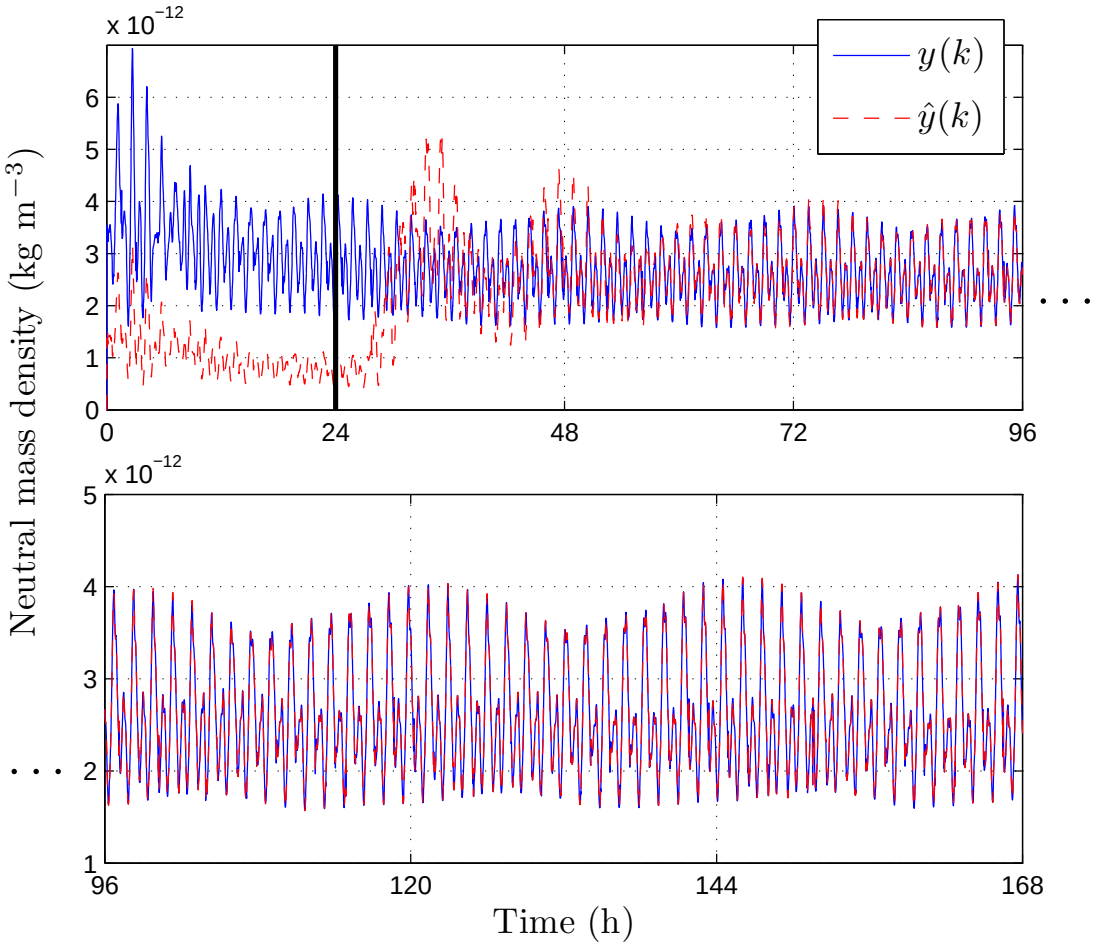


Figure 5.4: Measured output $y(k)$ and the estimated output $\hat{y}(k)$ for the case of simulated CHAMP satellite data with a constant true $\bar{\mathrm{F}}_{10.7}(k)$. In this example, $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ is kept at a constant value of 100 for the first 24 h, after which RCUIO is turned on. This plot shows that $\hat{y}(k)$ converges to within 0.11×10^{-12} kg m^{-3} of $y(k)$ in less than 72 h.

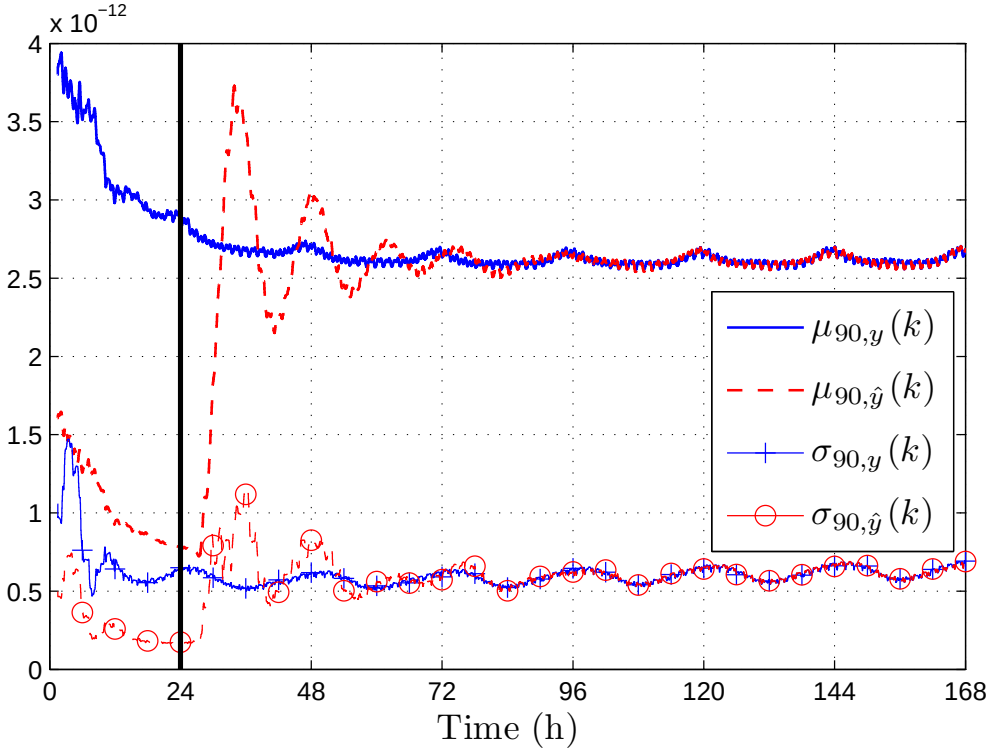


Figure 5.5: $\mu_{90,y}(k)$, $\mu_{90,\hat{y}}(k)$, $\sigma_{90,y}(k)$, and $\sigma_{90,\hat{y}}(k)$ for the case of simulated CHAMP satellite data with a constant true $\bar{\mathrm{F}}_{10.7}(k)$. This plot shows that, after RCUIO is turned on, $\mu_{90,y}(k)$ and $\sigma_{90,y}(k)$ converge to within 0.12×10^{-12} kg m^{-3} of $\mu_{90,\hat{y}}(k)$ and $\sigma_{90,\hat{y}}(k)$, respectively, in less than 72 h.

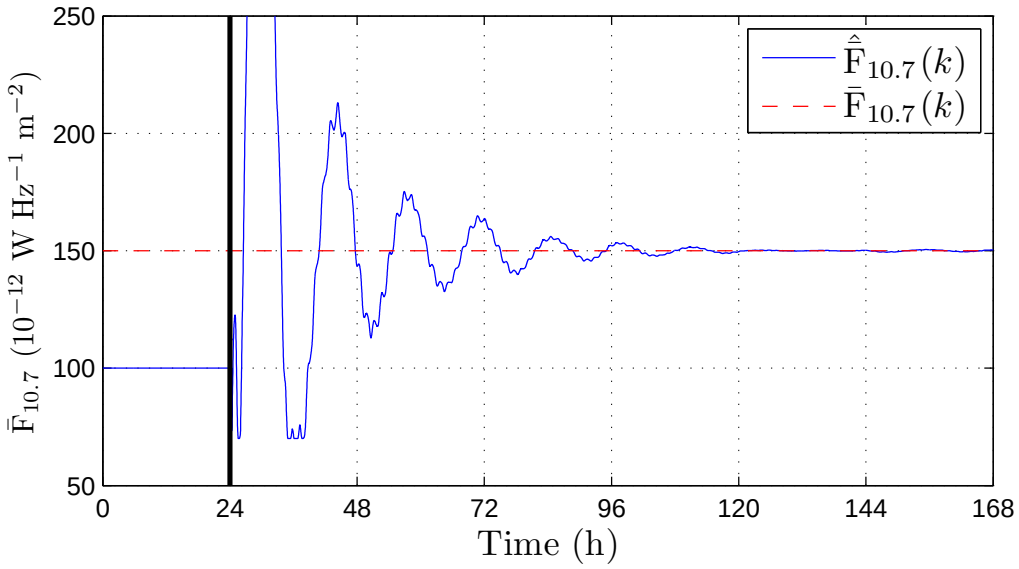


Figure 5.6: True $\bar{F}_{10.7}(k)$ and estimated $\bar{\hat{F}}_{10.7}(k)$ for the case of simulated CHAMP satellite data with a constant true $\bar{F}_{10.7}(k)$. This plot shows that, after RCUIO is turned on, $\bar{\hat{F}}_{10.7}(k)$ converges to within 10 SFU of $\bar{F}_{10.7}(k)$ in less than 96 h.

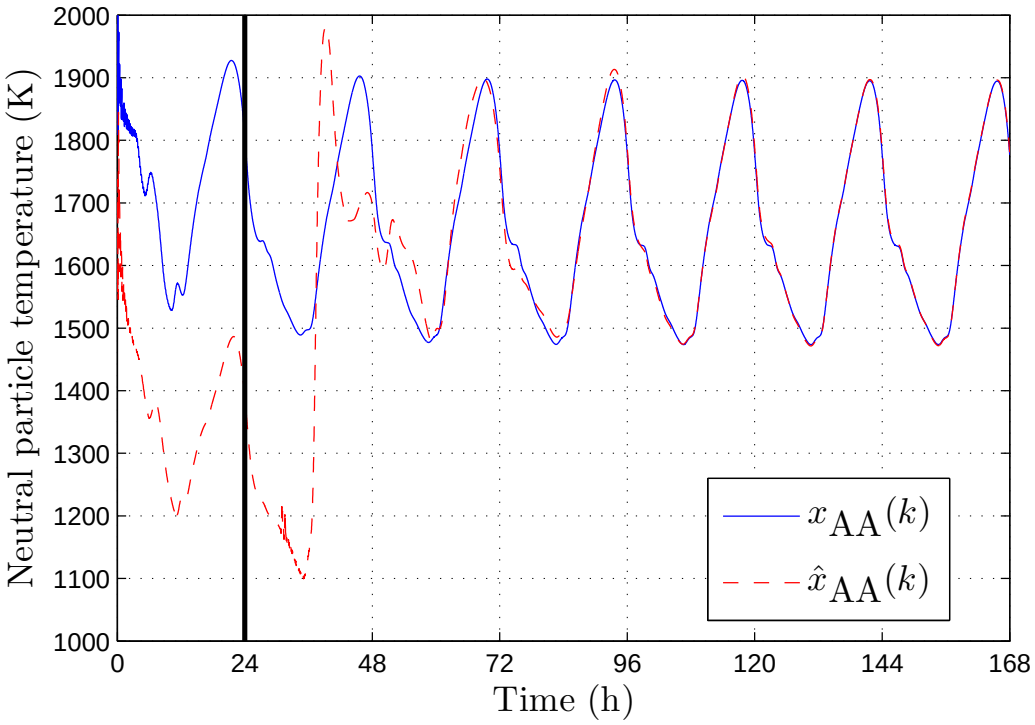


Figure 5.7: True $x_{\mathrm{AA}}(k)$ and estimated $\hat{x}_{\mathrm{AA}}(k)$ for the case of simulated CHAMP satellite data with a constant true $\bar{\mathrm{F}}_{10.7}(k)$. This plot shows that, after RCUIO is turned on, $\hat{x}_{\mathrm{AA}}(k)$ converges to within 10 K of $x_{\mathrm{AA}}(k)$ in less than 96 h.

5.4.2 Simulated Satellite Data Case 2 : Varying $\mathbf{F}_{10.7}$

In this example, we consider the case where the truth data are generated by using a time-varying $\bar{\mathrm{F}}_{10.7}$ and recording the neutral mass density at CHAMP locations from 2002-11-24 to 2002-12-06. For this example, we set the filters P_1 and P_2 as in the previous example. The RCUIO parameters are chosen to be $\tilde{\mathcal{H}} = H_1 = 0.1$, $\gamma = 100$, and $\eta = 0$. Figure 5.9 shows $\mu_{90,y}(k)$, $\mu_{90,\hat{y}}(k)$, $\sigma_{90,y}(k)$, and $\sigma_{90,\hat{y}}(k)$. This plot shows that $\mu_{90,y}(k)$ and $\sigma_{90,y}(k)$ converge to within 0.22×10^{-12} kg m^{-3} of $\mu_{90,\hat{y}}(k)$ and $\sigma_{90,\hat{y}}(k)$, respectively, in less than 96 h. RCUIO is able to track the averaged CHAMP measurements after 96 h.

Figure 5.10 shows the true and estimated $\bar{\mathrm{F}}_{10.7}$, and shows that $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ converges

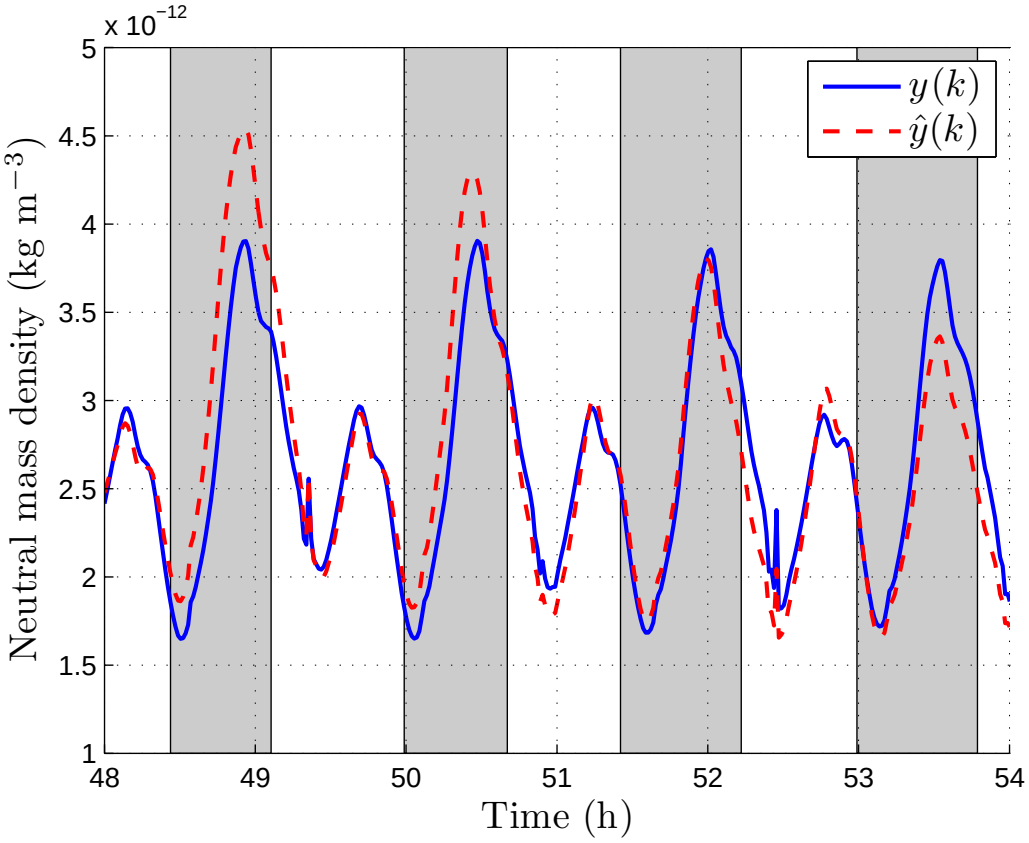


Figure 5.8: Time periods during which the solar zenith angle at satellite location is greater than 90° for the case of simulated CHAMP satellite data with a constant $\bar{F}_{10.7}(k)$.

to within 10 SFU of $\bar{F}_{10.7}$ in less than 96 h. Figure 5.11 shows the true temperature $x_{AA}(k)$ and estimated temperature $\hat{x}_{AA}(k)$, and shows that $\hat{x}_{AA}(k)$ converges to within 10 K of $x_{AA}(k)$ in less than 96 h.

5.4.3 Real Satellite Data Case 1 : GITM with photoelectron heating

We now consider the case where the truth data are recorded by CHAMP from 2002-11-24 to 2002-12-06 and we use GITM with photoelectron heating. For this example,

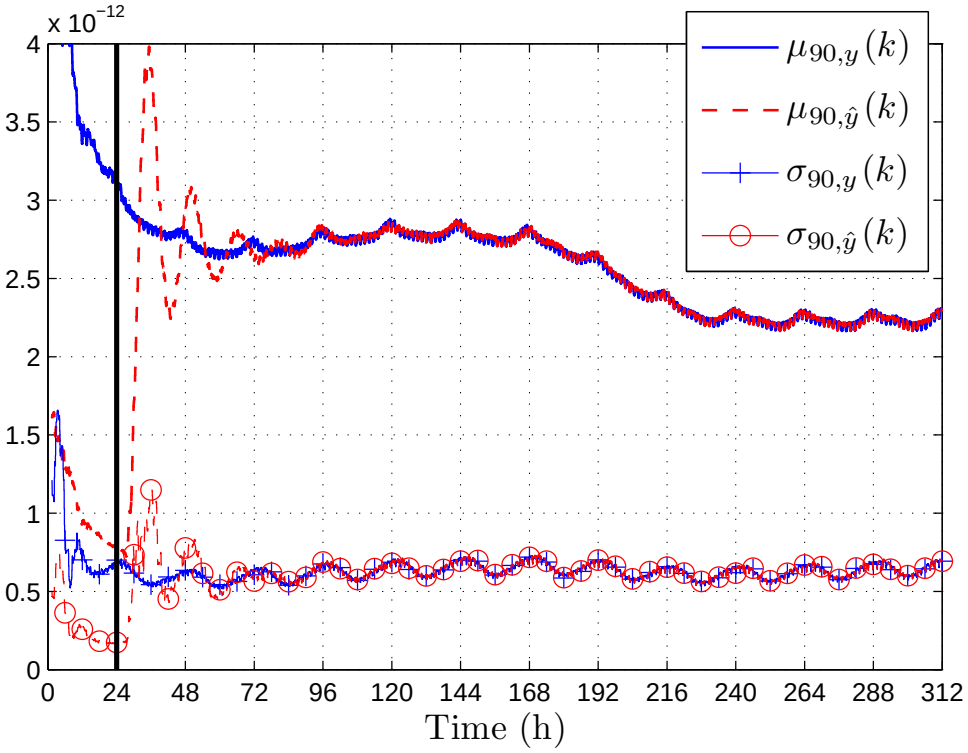


Figure 5.9: $\mu_{90,y}(k)$, $\mu_{90,\hat{y}}(k)$, $\sigma_{90,y}(k)$, and $\sigma_{90,\hat{y}}(k)$ for the case of simulated CHAMP satellite data with a time-varying truth $\bar{\mathrm{F}}_{10.7}(k)$. This plot shows that, after RCUIO is turned on, $\mu_{90,y}(k)$ and $\sigma_{90,y}(k)$ converge to within 0.22×10^{-12} kg m^{-3} of $\mu_{90,\hat{y}}(k)$ and $\sigma_{90,\hat{y}}(k)$, respectively, in less than 96 h.

P_2 is set as

$$z_{\mathrm{f}}(k) = 0.922 z_{\mathrm{f}}(k-1) + 0.0039 z(k) + 0.0039 z(k-1),$$

and P_1 is set as

$$\hat{\bar{\mathrm{F}}}_{10.7}(k) = 0.984 \hat{\bar{\mathrm{F}}}_{10.7}(k-1) + 0.0078 \hat{\bar{\mathrm{F}}}_{10.7,\,\mathrm{f}}(k) + 0.0078 \hat{\bar{\mathrm{F}}}_{10.7,\,\mathrm{f}}(k-1).$$

The filters P_1 and P_2 approximately average $z(k)$ and $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ over 90-min windows, respectively. Finally, the RCUIO parameters are chosen as $\tilde{\mathcal{H}} = H_1 = 0.175$, $\gamma = 100$, and $\eta = 0$. We calculate the RMS of $z(k)$ after 144 h in order to minimize the effect

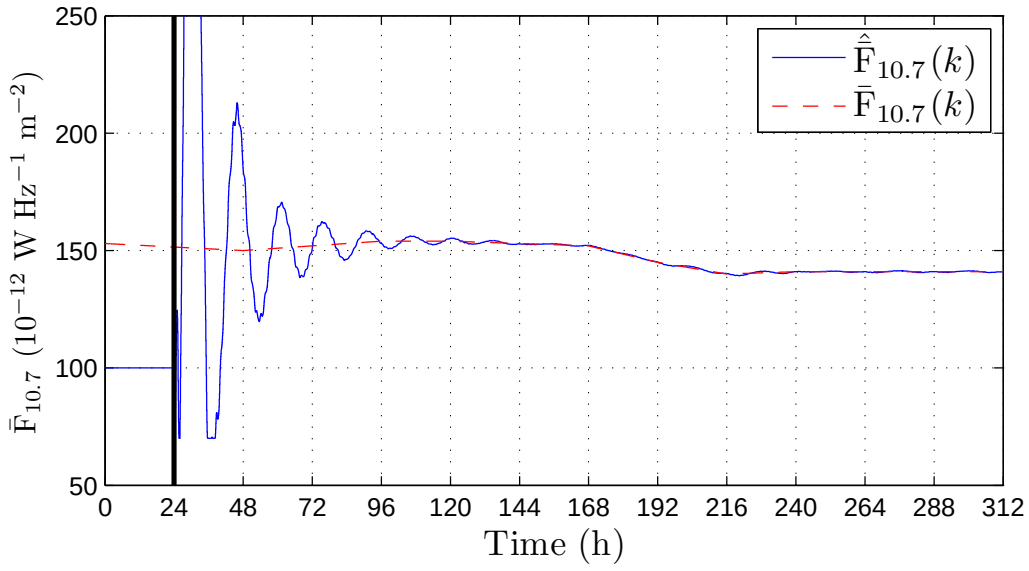


Figure 5.10: True $\bar{\mathrm{F}}_{10.7}(k)$ and estimated $\bar{\hat{\mathrm{F}}}_{10.7}(k)$ for the case of simulated CHAMP satellite data with a time-varying truth $\bar{\mathrm{F}}_{10.7}(k)$. This plot shows that, after RCUIO is turned on, $\bar{\hat{\mathrm{F}}}_{10.7}(k)$ converges to within 10 SFU of $\bar{\mathrm{F}}_{10.7}(k)$ in less than 96 h.

of transients in $\hat{y}(k)$ generated during the convergence of the adaptive subsystem (2.28). First, we consider the measurements from CHAMP. Figure 5.12 shows the windowed mean and variance of $y(k)$, $\hat{y}(k)$, and $\hat{y}_m(k)$. For this example, GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$ yields RMS$(z) = 6.5 \times 10^{-13}$, and GITM with RCUIO yields RMS$(z) = 6.1 \times 10^{-13}$. In other words, GITM with RCUIO yields 6% reduction in RMS(z) compared to GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$.

Figure 5.13 shows the measured and estimated $\bar{\mathrm{F}}_{10.7}(k)$. This plot shows that $\mu_{1440,\bar{\hat{\mathrm{F}}}_{10.7}}(k)$ (the average of $\bar{\hat{\mathrm{F}}}_{10.7}(k)$ over 1 day) converges to within 6 SFU of the measured values of $\bar{\mathrm{F}}_{10.7}(k)$ in 72 h.

Next, we consider data from GRACE to assess the quality of the state estimates. Define $z_G(k) \triangleq y_G(k) - \hat{y}_G(k)$. Figure 5.14 shows the windowed mean and variance of $y_G(k)$, $\hat{y}_G(k)$, and $\hat{y}_{G,m}(k)$. For this example, GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$ yields RMS$(z_G) = 4 \times 10^{-13}$, whereas GITM with RCUIO yields RMS$(z_G) = 3.6 \times 10^{-13}$. Therefore, GITM with RCUIO yields 11% reduction in RMS(z_G) than GITM with

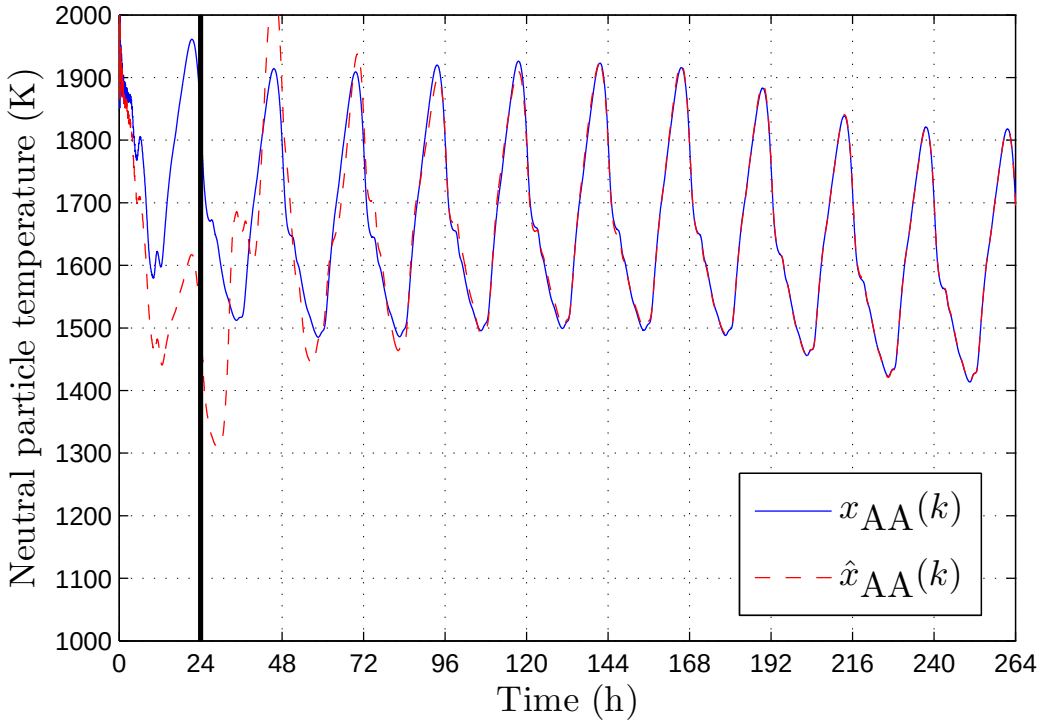


Figure 5.11: True $x_{\mathrm{AA}}(k)$ and estimated $\hat{x}_{\mathrm{AA}}(k)$ for the case of simulated CHAMP satellite data with a time-varying true $\bar{\mathrm{F}}_{10.7}(k)$. This plot shows that, after RCUIO is turned on, $\hat{x}_{\mathrm{AA}}(k)$ converges to within 10 K of $x_{\mathrm{AA}}(k)$ in less than 96 h.

measured $\bar{\mathrm{F}}_{10.7}(k)$.

5.4.4 Real Satellite Data Case 2 : GITM without photoelectron heating

We now consider the case where the truth data are recorded by CHAMP from 2002-11-24 to 2002-12-06 and we use GITM without photoelectron heating. For this example, P_2 is set as

$$z_{\mathrm{f}}(k) = 0.922 z_{\mathrm{f}}(k-1) + 0.0039 z(k) + 0.0039 z(k-1),$$

and P_1 is set as

$$\hat{\bar{\mathrm{F}}}_{10.7}(k) = 0.984\hat{\bar{\mathrm{F}}}_{10.7}(k-1) + 0.0078\hat{\bar{\mathrm{F}}}_{10.7,\,\mathrm{f}}(k) + 0.0078\hat{\bar{\mathrm{F}}}_{10.7,\,\mathrm{f}}(k-1).$$

The filters P_1 and P_2 approximately average $z(k)$ and $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ over 90-min windows, respectively. Finally, the RCUIO parameters are chosen as $\tilde{\mathcal{H}} = H_1 = 0.175$, $\gamma = 100$, and $\eta = 0$. We calculate the RMS of $z(k)$ after 144 h in order to minimize the effect of transients in $\hat{y}(k)$ generated during the convergence of the adaptive subsystem (2.28). First, we consider the measurements from CHAMP. Figure 5.15 shows the windowed mean and variance of $y(k)$, $\hat{y}(k)$, and $\hat{y}_m(k)$. For this example, GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$ yields $\mathrm{RMS}(z) = 2.23 \times 10^{-12}$, and GITM with RCUIO yields $\mathrm{RMS}(z) = 5.99 \times 10^{-13}$. In other words, GITM with RCUIO yields 74.6% reduction in $\mathrm{RMS}(z)$ compared to GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$. In this example, RCUIO uses $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ as an input to GITM in order to correct the errors between the measured output and the output from GITM, and thus account for the intentional modeling inaccuracies incurred by neglecting photoelectron heating in GITM. In the case where the model of the ionosphere-thermosphere is the truth model and $\bar{\mathrm{F}}_{10.7}(k)$ accurately specifies the solar EUV and X-ray distribution, we expect $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ to converge to $\bar{\mathrm{F}}_{10.7}(k)$, as shown in Example 6.5.4.1 and Example 6.5.4.2.

Figure 5.16 shows the measured and estimated $\bar{\mathrm{F}}_{10.7}(k)$. This plot and Figure 5.15 show that RCUIO yields $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ with lower $\mathrm{RMS}(z)$ compared to GITM with measured values of $\bar{\mathrm{F}}_{10.7}(k)$. In this example, RCUIO uses $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ as an input to GITM in order to correct the errors between CHAMP satellite data and the computed output from GITM.

Next, we consider data from GRACE to assess the quality of the state estimates. Define $z_G(k) \triangleq y_G(k) - \hat{y}_G(k)$. Figure 5.17 shows the windowed mean and variance of $y_G(k)$, $\hat{y}_G(k)$, and $\hat{y}_{G,m}(k)$. For this example, GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$ yields

RMS(z_G) $= 5.4950 \times 10^{-12}$, whereas GITM with RCUIO yields RMS(z_G) $= 4.307 \times 10^{-13}$. Therefore, GITM with RCUIO yields 21.61% reduction in RMS(z_G) than GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$, which shows that GITM with RCUIO yields better state estimates than GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$ by reducing the errors between the measured output and the output from GITM.

5.5 Conclusions

This chapter considered data assimilation based on the Global Ionosphere-Thermosphere Model (GITM). For the case of simulated satellite measurements, GITM was used as both the truth model and the data assimilation model, whereas, for the case of real satellite measurements, GITM was used as the data assimilation model. The data assimilation methodology used in this work is Retrospective Cost Unknown Input Observer (RCUIO), which has the ability to estimate both states and unknown inputs. Since RCUIO is based on an adaptive subsystem technique, it does not require an ensemble, and thus it is less computationally expensive than ensemble-based data assimilation methods.

For this work, we assumed that the only available measurements are the neutral density data obtained from the CHAMP satellite. For the case of simulated CHAMP data based on the GITM truth model, RCUIO was able to estimate $\bar{\mathrm{F}}_{10.7}$ and states in the constant and time-varying true $\bar{\mathrm{F}}_{10.7}$ scenarios. For the case of real CHAMP data and using GITM with photoelectron heating, RCUIO was able to estimate $\bar{\mathrm{F}}_{10.7}$ accurately, and, for the case of real CHAMP data and using GITM without photoelectron heating, the estimates of $\bar{\mathrm{F}}_{10.7}$ provided by RCUIO reduced the RMS error between the measured output and the output from GITM by 74.6% compared to GITM using the measured $\bar{\mathrm{F}}_{10.7}$. For all of the GITM state and input estimation examples, the computational cost of the RCUIO subsystem adaptation was equivalent

to approximately 1% of the computational cost of GITM.

The specific application of RCUIO in the present chapter, namely, the use of neutral density measurements to estimate $F_{10.7}$, is intended as a proof of concept illustration of RCUIO on a highly nonlinear, physically realistic system. Although measurements of $F_{10.7}$ are available from observations, the estimates provided by RCUIO at a modest computational cost may, for practical purposes, be useful for validation. In addition, this work provides a first step toward estimating drivers that cannot easily be measured or estimated by existing techniques. RCUIO may also be useful for applications that are distinct from the application considered in this work, which provides the details of the approach to facilitate future investigations.

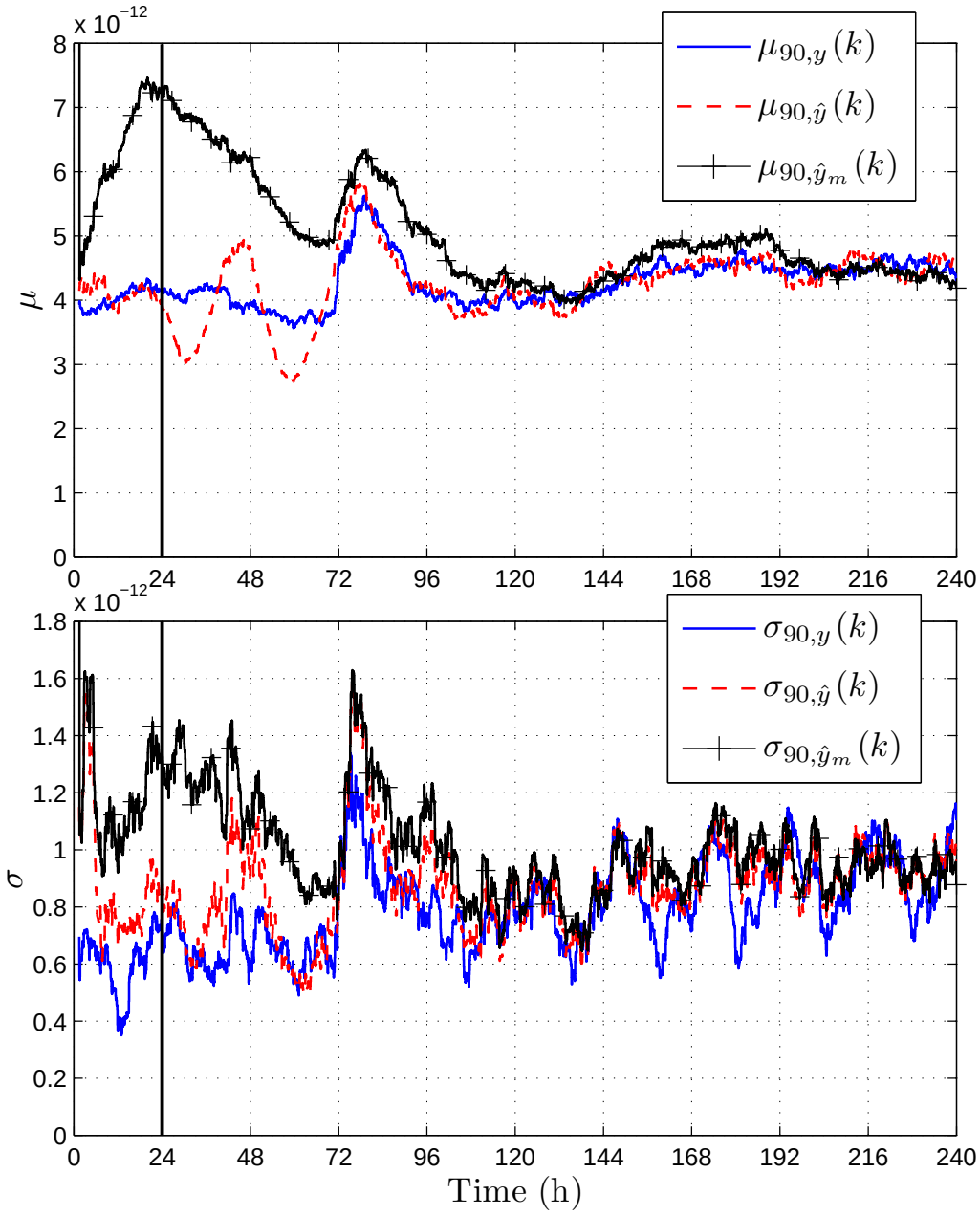


Figure 5.12: $\mu_{90,y}(k)$, $\mu_{90,\hat{y}}(k)$, $\mu_{90,\hat{y}_m}(k)$, $\sigma_{90,y}(k)$, $\sigma_{90,\hat{y}}(k)$, and $\sigma_{90,\hat{y}_m}(k)$ for the case of real CHAMP satellite data and GITM with photoelectron heating. For this example, GITM with RCUIO yields 6% lower RMS(z) compared to GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$.

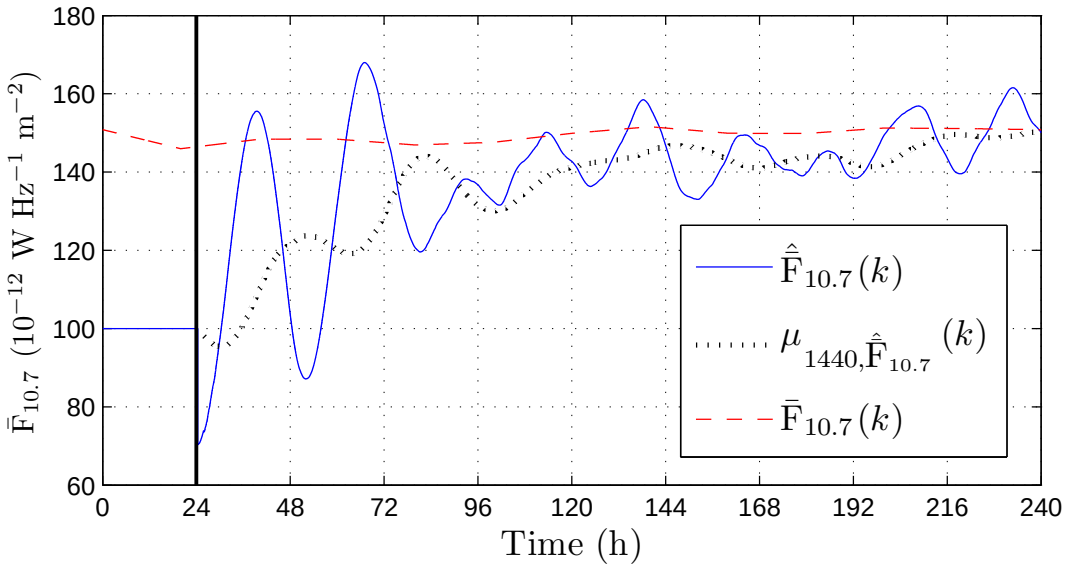


Figure 5.13: Measured and estimated $\bar{\mathrm{F}}_{10.7}(k)$ for the case of real CHAMP satellite data and GITM with photoelectron heating. This plot shows that $\mu_{1440,\hat{\bar{\mathrm{F}}}_{10.7}}(k)$ (the average of $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ over 1 day) converges to within 6 SFU of the measured values of $\bar{\mathrm{F}}_{10.7}(k)$ in 72 h.

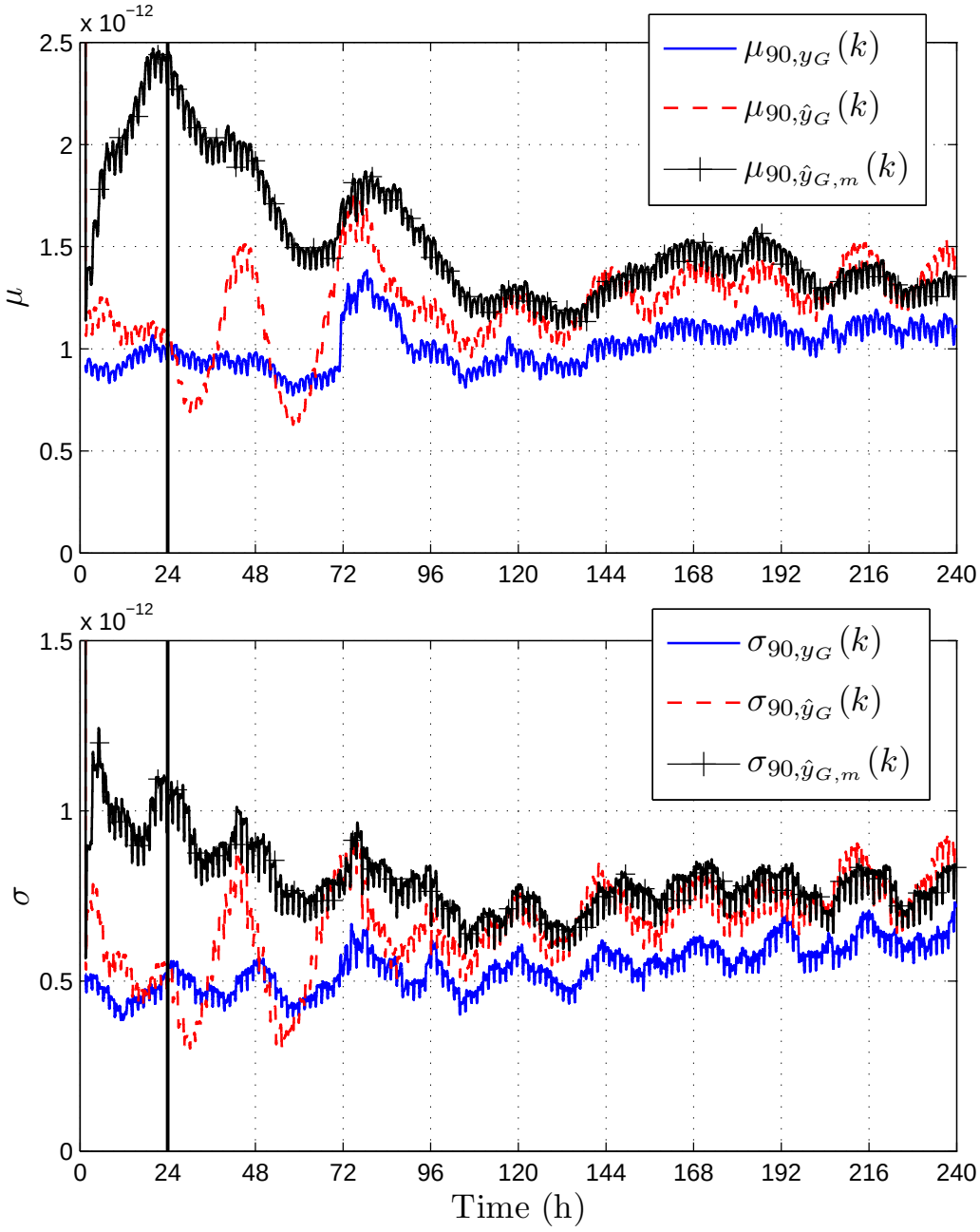


Figure 5.14: This plot shows $\mu_{90,y_G}(k)$, $\mu_{90,\hat{y}_G}(k)$, $\mu_{90,\hat{y}_{G,m}}(k)$, $\sigma_{90,y_G}(k)$, $\sigma_{90,\hat{y}_G}(k)$, and $\sigma_{90,\hat{y}_{G,m}}(k)$ for real GRACE satellite data and the case of real CHAMP satellite data and GITM with photoelectron heating. For this example, GITM with RCUIO yields 11% reduction in RMS(z_G) compared to GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$.

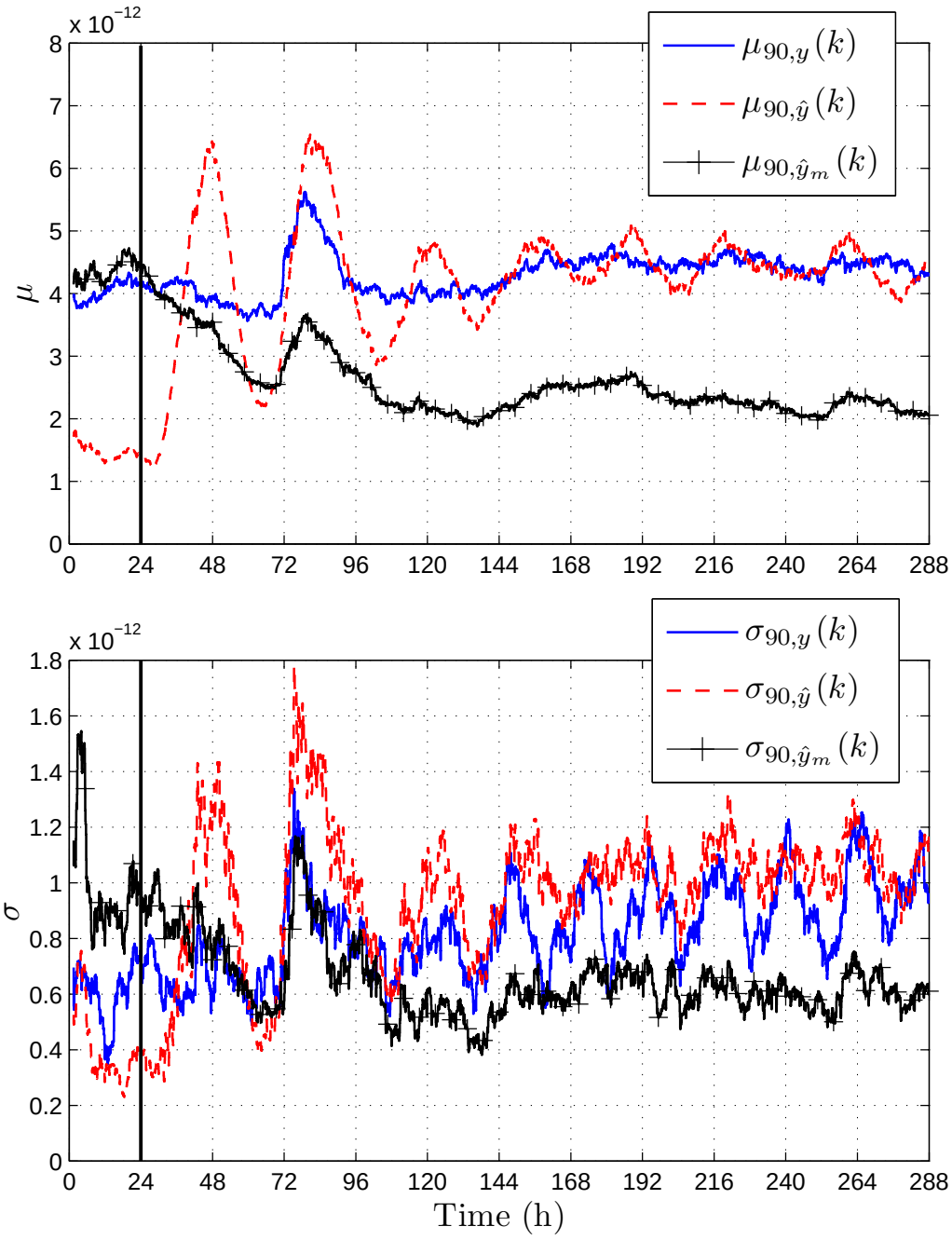


Figure 5.15: $\mu_{90,y}(k)$, $\mu_{90,\hat{y}}(k)$, $\mu_{90,\hat{y}_m}(k)$, $\sigma_{90,y}(k)$, $\sigma_{90,\hat{y}}(k)$, and $\sigma_{90,\hat{y}_m}(k)$ for the case of real CHAMP satellite data and GITM without photoelectron heating. For this example, GITM with RCUIO yields 74.6% lower RMS(z) compared to GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$, which shows that, despite the intentional model error, RCUIO reduces the errors between CHAMP satellite data and the computed output from GITM.

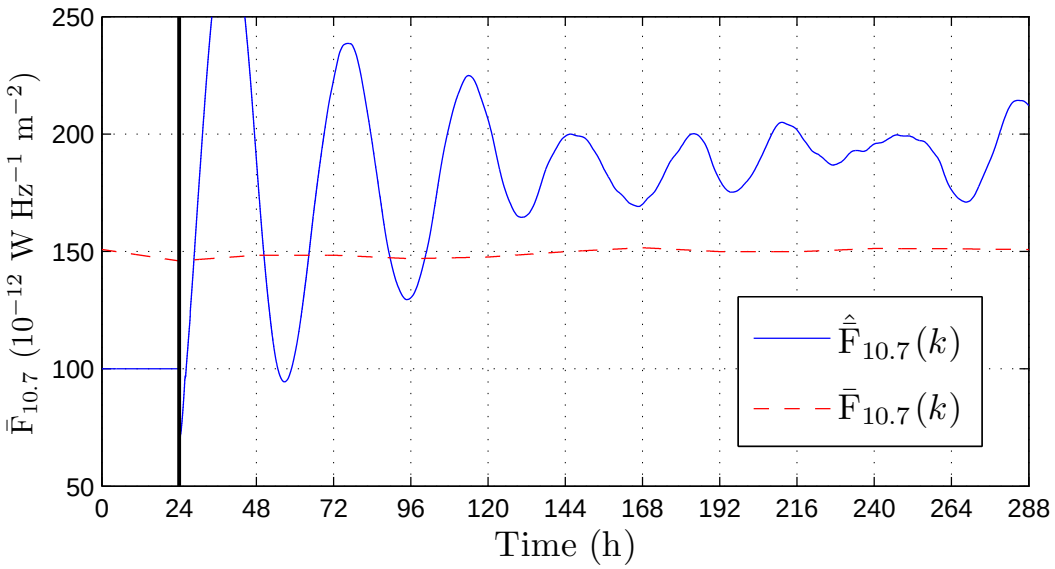


Figure 5.16: Measured and estimated $\bar{\mathrm{F}}_{10.7}(k)$ for the case of real CHAMP satellite data and GITM without photoelectron heating. This plot and Figure 5.15 show that RCUIO yields $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ with lower RMS(z) compared to GITM with measured values of $\bar{\mathrm{F}}_{10.7}(k)$. In this example, RCUIO uses $\hat{\bar{\mathrm{F}}}_{10.7}(k)$ as an input to GITM in order to correct the errors between CHAMP satellite data and the computed output from GITM.

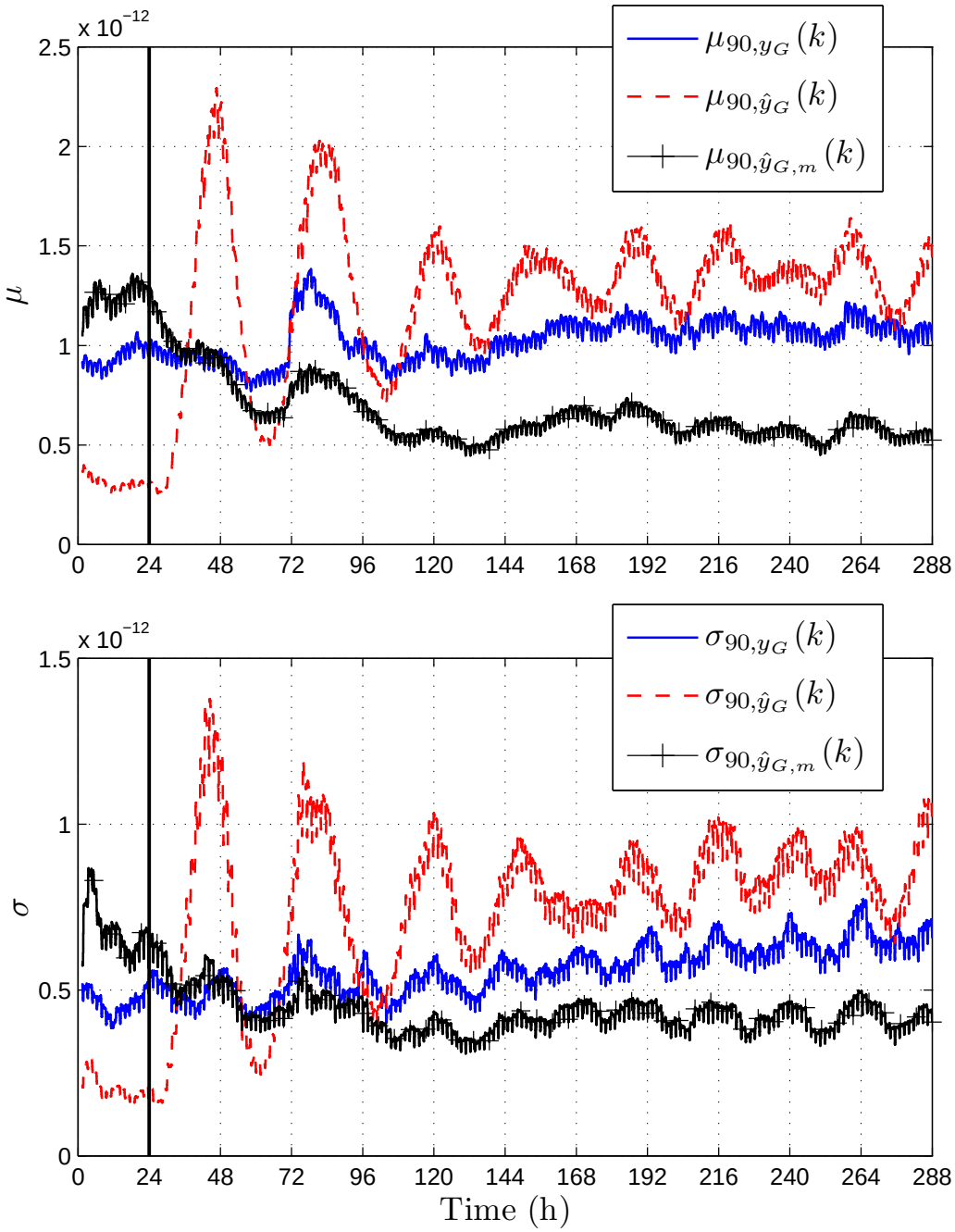


Figure 5.17: This plot shows $\mu_{90,y_G}(k)$, $\mu_{90,\hat{y}_G}(k)$, $\mu_{90,\hat{y}_{G,m}}(k)$, $\sigma_{90,y_G}(k)$, $\sigma_{90,\hat{y}_G}(k)$, and $\sigma_{90,\hat{y}_{G,m}}(k)$ for real GRACE satellite data and the case of real CHAMP satellite data and GITM without photoelectron heating. For this example, GITM with RCUIO yields 21.61% reduction in RMS(z_G) compared to GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$, which shows that RCUIO yields better state estimates than GITM with measured $\bar{\mathrm{F}}_{10.7}(k)$ by reducing the errors between the CHAMP satellite data and the computed output from GITM.

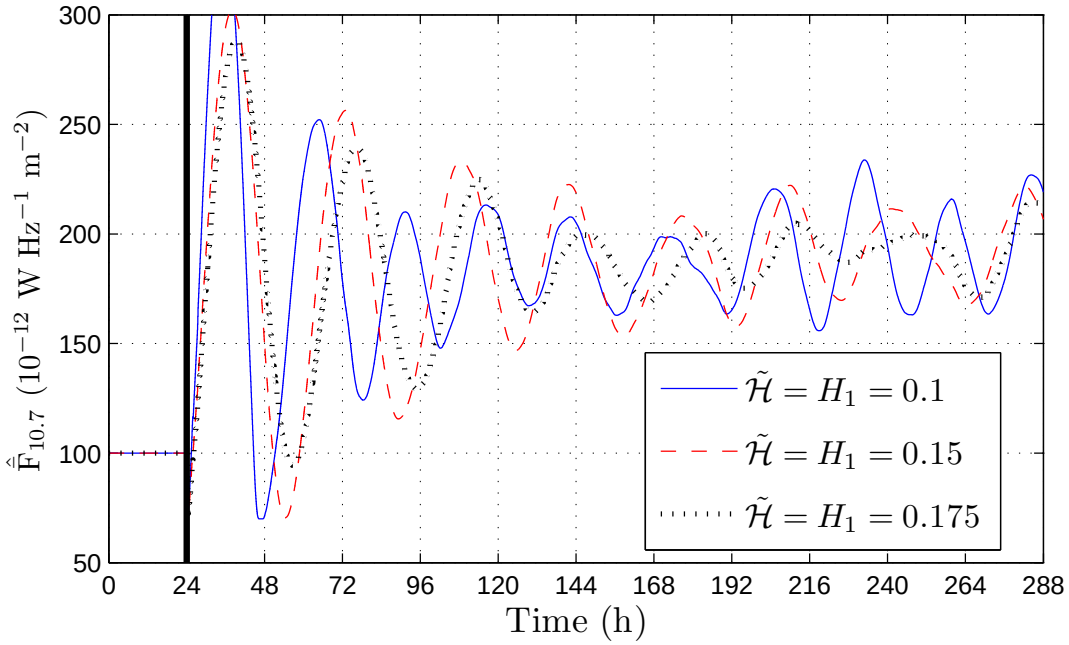


Figure 5.18: $\hat{\tilde{\mathrm{F}}}_{10.7}(k)$ for the case of real CHAMP satellite data and GITM without photoelectron heating using four different values of $\tilde{\mathcal{H}}$. The parameter $\tilde{\mathcal{H}}^{-1}$ acts as an effective gain of the adaptive subsystem.

CHAPTER 6

Conclusions and Future Work

6.1 Conclusions

In this work, we considered the problem of estimating unknown states and inputs using a retrospective cost unknown input observer, which is inspired by retrospective cost adaptive control. RCAC has several desirable features, such as applicability to nonminimum-phase systems and the requirement of limited plant modeling information. In Chapter 2, we developed the architecture of RCUIO starting from the case where the first Markov parameter is nonzero and only one past measurement is used. In this case, RCUIO was linked to the direct input reconstruction approach. Next, RCUIO was derived for the general case where the first Markov parameter can be non-zero and multiple past measurements can be used to update the adaptive feedback system. We then concluded Chapter 2 by numerically demonstrating RCUIO on linear and nonlinear systems.

A major contribution of this work is the development of a new sliding window, variable regularization recursive least squares algorithm. RCUIO depends on recursive least squares to update the adaptive feedback system. However, recursive least squares has one major disadvantage that, if the data are persistently exciting, then its ability to adapt diminishes every ℓ steps, where ℓ is a positive integer. In the case of RCUIO, this means that, after a certain number of steps, the adaptive feed-

back system will no longer adapt to changes in the input signal or the plant itself. Although this disadvantage can be mitigated by the use of a forgetting factor, the forgetting factor itself can cause instability of the recursive least squares algorithm in the case of nonpersistent data. The SW-VR-RLS algorithm developed in this work does not suffer from these disadvantages. In Chapter 3, we develop the SW-VR-RLS algorithm, analyze its computational complexity, convergence properties, and its numerical stability. We also provide numerical comparisons to other adaptive filtering algorithms, and illustrate the effect of the various parameters of SW-VR-RLS. Finally, we provide an example where we compare RCUIO with standard RLS, and RCUIO with SW-VR-RLS.

In Chapter 4, we developed a growing window, variable regularization recursive least squares algorithm. In this chapter, we analyzed the convergence properties of GW-VR-RLS, and performed an extensive numerical analysis of the algorithm in and without the presence of noise.

In Chapter 5, we considered the problem of estimating states and input in the ionosphere-thermosphere using RCUIO and the Global Ionosphere Thermosphere Model. More specifically, we used real neutral density measurements from CHAMP satellite and simulated neutral density measurements from GITM at CHAMP locations to estimate the unknown input $F_{10.7}$ and states in the ionosphere and thermosphere. For the case of simulated CHAMP data based on the GITM truth model, RCUIO was able to estimate $F_{10.7}$ and states in the constant and time-varying true $F_{10.7}$ scenarios. For the case of real CHAMP data and using GITM with photoelectron heating, RCUIO was able to estimate $F_{10.7}$ accurately, and, for the case of real CHAMP data and using GITM without photoelectron heating, the estimates of $F_{10.7}$ provided by RCUIO reduced the RMS error between the measured output and the output from GITM by 74.6% compared to GITM using the ground measured $F_{10.7}$.

6.2 Future Work

This dissertation showed that RCUIO is a viable approach for unknown state and input estimation for nonminimum-phase systems or nonlinear systems. However, one of the biggest hurdles for RCUIO is the lack of sufficient conditions under which it is guaranteed to work. This has been explored to some extent in [31, 85], and remains an active area of research.

This work also explored SW-VR-RLS in detail, and analyzed its convergence properties rigorously. However, this analysis was done in a system identification framework, where the identification method does not interact with the system. However, when SW-VR-RLS is used with RCUIO, then SW-VR-RLS interacts with the system itself, as it is used to update the adaptive feedback system. Furthermore, in the analysis of SW-VR-RLS, we assumed that the data are generated by a truth system, where the true unknown parameters are labeled x_*. Then, we showed that under certain conditions, the estimated parameters converge to the true unknown parameters. For RCUIO, these parameters are the coefficients of the adaptive feedback system, and it is not obvious what these parameters should converge to. In [86] and [87], this is addressed by assuming that there exists an ideal feedback system, and the adaptive feedback system converges to the behavior of the ideal feedback system.

Finally, this work applied RCUIO to estimate an unknown input and states in the ionosphere-thermosphere. As mentioned in Chapter 5, the ultimate goal is to estimate the solar EUV and X-ray inputs in multiple wavelengths instead of identifying $F_{10.7}$, which is only a proxy for the EUV and X-ray inputs. However, to estimate the EUV and X-ray inputs there are two main hurdles. First, multiple measurements are required, and it is unclear how the delay in measurements due to the delayed effect of the inputs on the far side of the sun will affect RCUIO. One option is to use total electron content measurements, which are more widely available than neutral density measurements, and also at more locations in the Earth's atmosphere. The second

hurdle is the selection of $\tilde{\mathcal{H}}$, which is not a scalar for MIMO systems, and must be chosen by trial and error. An approach to address this issue is to first take a SISO case and assume that the measurement is at the subsolar point, and tune $\tilde{\mathcal{H}}$ to obtain the desired response from RCUIO. Then, when other measurements are added, their value in $\tilde{\mathcal{H}}$ is defined as a function of the distance of that measurement location from the subsolar point.

BIBLIOGRAPHY

[1] S. Gillijns and B. De Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems," *Automatica*, vol. 43, no. 1, pp. 111–116, 2007.

[2] R. E. Kalman, "Contributions to the theory of optimal control," *Bol. Soc. Mat. Mexicana*, vol. 5, no. 2, pp. 102–119, 1960.

[3] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[4] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of basic Engineering*, vol. 83, no. 3, pp. 95–108, 1961.

[5] J. Baras, A. Bensoussan, and M. James, "Dynamic observers as asymptotic limits of recursive filters: Special cases," *SIAM Journal on Applied Mathematics*, vol. 48, no. 5, pp. 1147–1158, 1988.

[6] Y. Song and J. W. Grizzle, "The extended kalman filter as a local asymptotic observer for nonlinear discrete-time systems," in *American Control Conference, 1992.* IEEE, 1992, pp. 3365–3369.

[7] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97.* International Society for Optics and Photonics, 1997, pp. 182–193.

[8] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *Automatic Control, IEEE Transactions on*, vol. 45, no. 3, pp. 477–482, 2000.

[9] D. Alspach and H. Sorenson, "Nonlinear bayesian estimation using gaussian sum approximations," *Automatic Control, IEEE Transactions on*, vol. 17, no. 4, pp. 439–448, 1972.

[10] M. Leibman, T. Edgar, and L. Lasdon, "Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques," *Computers & chemical engineering*, vol. 16, no. 10, pp. 963–986, 1992.

[11] B. Ristic, S. Arulampalm, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications.* Artech House Publishers, 2004.

[12] P. K. Kitanidis, "Unbiased minimum-variance linear state estimation," *Automatica*, vol. 23, pp. 775–778, 1987.

[13] S. Bhattacharyya, "Observer design for linear systems with unknown inputs," *IEEE Trans. Autom. Control*, vol. 23, pp. 483–484, 1978.

[14] M. Darouach and M. Zasadzinski, "Unbiased minimum variance estimation for systems with unknown exogenous inputs," *Automatica*, vol. 33, no. 4, pp. 717–719, 1997.

[15] M. Hou and P. Muller, "Design of observers for linear systems with unknown inputs," *IEEE Trans. Autom. Control*, vol. 37, pp. 871–875, 1992.

[16] M. E. Valcher, "State observers for discrete-time linear systems with unknown inputs," *IEEE Trans. Autom. Control*, vol. 44, pp. 397–401, 1999.

[17] Y. Xiong and M. Saif, "Unknown disturbance inputs estimation based on a state functional observer design," *Automatica*, vol. 39, pp. 1389–1398, 2003.

[18] T. Floquet and J. P. Barbot, "State and unknown input estimation for linear discete-time systems," *Automatica*, vol. 44, pp. 1883–1889, 2006.

[19] S. Sundaram and C. N. Hadjicostis, "Delayed observers for linear systems with unknown inputs," *IEEE Trans. Autom. Control*, vol. 52, pp. 334–339, 2007.

[20] H. J. Palanthandalam-Madapusi and D. S. Bernstein, "Unbiased minimum-variance filtering for input reconstruction," in *Proc. Amer. Cont. Conf.*, New York City, NY, 2007, pp. 5712–5717.

[21] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*. Chapman & Hall, 2011, vol. 24.

[22] G. Evensen, "The ensemble kalman filter: Theoretical formulation and practical implementation," *Ocean dynamics*, vol. 53, no. 4, pp. 343–367, 2003.

[23] P. L. Houtekamer and H. L. Mitchell, "Data assimilation using an ensemble Kalman filter technique," *Monthly Weather Review*, vol. 126, no. 3, pp. 796–811, Mar 1998.

[24] J. L. Anderson, "An ensemble adjustment Kalman filter for data assimilation," *Monthly Weather Review*, vol. 129, no. 12, pp. 2884–2903, Dec 2001.

[25] R. L. Williams, D. A. Lawrence *et al.*, *Linear state-space control systems*. Wiley, 2007.

[26] W. Kerwin and J. Prince, "On the optimality of recursive unbiased state estimation with unknown inputs," *Automatica*, vol. 36, no. 9, pp. 1381–1383, 2000.

[27] M. Darouach, M. Zasadzinski, and S. J. Xu, "Full-order observers for linear systems with unknown inputs," *Automatic Control, IEEE Transactions on*, vol. 39, no. 3, pp. 606–609, 1994.

[28] S. Gillijns and B. De Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough," *Automatica*, vol. 43, no. 5, pp. 934–937, 2007.

[29] M.-S. Chen and C.-C. Chen, "Unknown input observer for linear non-minimum phase systems," *Journal of the Franklin Institute*, vol. 347, no. 2, pp. 577–588, 2010.

[30] A. M. DAmato, E. D. Sumer, and D. S. Bernstein, "Retrospective cost adaptive control for systems with unknown nonminimum-phase zeros," in *AIAA Guid. Nav. Contr. Conf., Portland, OR*, 2011.

[31] A. M. D'Amato and D. S. Bernstein, "Adaptive forward-propagating input reconstruction for nonminimum-phase systems," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 598–603.

[32] A. J. Ridley, Y. Deng, and G. T'oth, "The global ionosphere/thermosphere model," *J. Atmos. Sol-Terr. Phys.*, vol. 68, no. 8, pp. 839–864, 2006.

[33] D. S. Bernstein and W. M. Haddad, "Steady-state kalman filtering with an h error bound," *Systems & Control Letters*, vol. 12, no. 1, pp. 9–16, 1989.

[34] L. Xie and Y. C. Soh, "Robust kalman filtering for uncertain systems," *Systems & Control Letters*, vol. 22, no. 2, pp. 123–129, 1994.

[35] I. R. Petersen and A. V. Savkin, *Robust Kalman filtering for signals and systems with large uncertainties*. Springer, 1999.

[36] H. Kwakernaak and R. Sivan, "The maximally achievable accuracy of linear optimal regulators and linear optimal filters," *Automatic Control, IEEE Transactions on*, vol. 17, no. 1, pp. 79–86, 1972.

[37] Z. Ding, "Adaptive estimation and rejection of unknown sinusoidal disturbances in a class of non-minimum-phase nonlinear systems," in *Control Theory and Applications, IEE Proceedings-*, vol. 153, no. 4. IET, 2006, pp. 379–386.

[38] W. Lan, B. M. Chen, and Z. Ding, "Adaptive estimation and rejection of unknown disturbances through measurement feedback for a class of nonminimum phase nonlinear mimo systems," in *Control and Automation, 2005. ICCA'05. International Conference on*, vol. 2. IEEE, 2005, pp. 685–690.

[39] R. Mehra, "On the identification of variances and adaptive kalman filtering," *Automatic Control, IEEE Transactions on*, vol. 15, no. 2, pp. 175–184, 1970.

[40] S.-H. Wang, E. Wang, and P. Dorato, "Observing the states of systems with unmeasurable disturbances," *Automatic Control, IEEE Transactions on*, vol. 20, no. 5, pp. 716–717, 1975.

[41] Y. Guan and M. Saif, "A novel approach to the design of unknown input observers," *Automatic Control, IEEE Transactions on*, vol. 36, no. 5, pp. 632–635, 1991.

[42] C.-S. Hsieh, "Robust two-stage kalman filters for systems with unknown inputs," *IEEE Trans. Autom. Control*, vol. 45, no. 12, pp. 2374–2378, 2000.

[43] L. Ljung and T. Söderström, *Theory and practice of Recursive Identification*. The MIT Press, 1983.

[44] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction, and Control*. Prentice Hall, 1984.

[45] T. Söderström and P. Stoica, *System Identification*. Prentice–Hall International, 1989.

[46] J. N. Juang, *Applied System Identification*. Upper Saddle River, NJ: Prentice-Hall, 1993.

[47] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. Addison-Wesley, 1995.

[48] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.

[49] L. Ljung, *System Identification: Theory for the User, 2nd ed.* Prentice-Hall Information and Systems Sciences, 1999.

[50] G. Tao, *Adaptive Control Design and Analysis*. Wiley, 2003.

[51] A. H. Sayed, *Adaptive Filters*. Hoboken, New Jersey: John Wiley and Sons, Inc., 2008.

[52] F. Gustafsson, *Adaptive Filtering and Change Detection*. Wiley, 2000.

[53] J. Jiang and Y. Zhang, "A novel variable-length sliding window blockwise least-squares algorithm for on-line estimation of time-varying parameters," *Int. J. Adaptive Contr. Sig. Proc.*, vol. 18, pp. 505–521, 2004.

[54] M. Belge and E. L. Miller, "A sliding window RLS-like adaptive algorithm for filtering alpha-stable noise," *IEEE Sig. Proc. Let.*, vol. 7, no. 4, pp. 86–89, 2000.

[55] B. Y. Choi and Z. Bien, "Sliding-windowed weighted recursive least-squares method for parameter estimation," *Electronics Let.*, vol. 25, no. 20, pp. 1381–1382, 1989.

[56] H. Liu and Z. He, "A sliding-exponential window RLS adaptive algorithm: properties and applications," *Sig. Proc.*, vol. 45, no. 3, pp. 357–368, 1995.

[57] T. van Waterschoot, G. Rombouts, and M. Moonen, "Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement," *Signal Processing*, vol. 88, pp. 594–611, 2008.

[58] A. Houacine and G. Demoment, "Chandrasekhar adaptive regularizer for adaptive filtering," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, April 8-11 1986, pp. 2967–2970.

[59] S. L. Gay, "Dynamically regularized fast RLS with application to echo cancellation," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 7-10 1996, pp. 957–960.

[60] Y. S. Choi, H. C. Shin, and W. J. Song, "Affine projection algorithms with adaptive regularization matrix," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 14-19 2006, pp. 201–204.

[61] J. W. Stokes and J. C. Platt, "Robust RLS with round robin regularization including application to stereo acoustic echo cancellation," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Sig. Proc.*, Toulouse, France, May 14–19 2006, pp. 736–739.

[62] K. Chen, J. Lu, and B. Xu, "A method to adjust regularization parameter of fast affine projection algorithm," in *Proc. 8th IEEE Int. Conf. on Signal Processing*, Beijing, China, November 16-20 2006.

[63] D. Challa, S. L. Grant, and A. Mohammad, "Variable regularized fast affine projections," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Honolulu, HI, April 15-20 2007, pp. 89–92.

[64] J. B. Hoagg, A. Ali, M. Mossberg, and D. S. Bernstein, "Sliding window recursive quadratic optimization with variable regularization," in *Proc. Amer. Contr. Conf.*, San Francisco, CA, June 29–July 1 2011, pp. 3275–3280.

[65] A. A. Ali, J. B. Hoagg, M. Mossberg, and D. S. Bernstein, "Growing window recursive quadratic optimization with variable regularization," in *Proc. Conf. Dec. Contr.*, Atlanta, GA, December 15–17 2010, pp. 496–501.

[66] National Research Council Canada, "Dominion radio astrophysical observatory," 2012, [accessed 18-December-2012]. [Online]. Available: http://archive.nrc-cnrc.gc.ca/eng/facilities/hia/radio-astrophysical.html/

[67] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *J. Geophys. Res.*, vol. 99, pp. 10,143–10,162, 1994.

[68] K. Agarwal, A. A. Ali, A. M. D'Amato, A. J. Ridley, and D. S. Bernstein, "Retrospective-cost-based adaptive state estimation and input reconstruction for the global ionosphere-thermosphere model," in *Guid. Nav. Contr. Conf.* Minneapolis, MN: AIAA-2012-4601, 2012.

[69] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach.* Princeton, New Jersey: Princeton University Press, 2008.

[70] T. Gansler, J. Benesty, S. L. Gay, and M. M. Sondhi, "A robust proportionate affine projection algorithm for network echo cancellation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Istanbul, Turkey, 2000, pp. 793–796.

[71] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 508–518, 2000.

[72] D. S. Bernstein, *Matrix Mathematics*, 2nd ed. Princeton University Press, 2009.

[73] Y. Deng, A. D. Richmond, A. J. Ridley, and H.-L. Liu, "Assessment of the non-hydrostatic effect on the upper atmosphere using a general circulation model (gcm)," *Geophys. Res. Lett.*, vol. 35, p. L01104, 2008.

[74] E. Yiğit and A. Ridley, "Role of variability in determining the vertical wind speeds and structure," *Journal of Geophysical Research*, vol. 116, no. A12, p. A12305, 2011.

[75] A. Hedin, "A revised thermospheric model based on mass spectrometer and incoherent scatter data: Msis-83," *Journal of Geophysical Research*, vol. 88, no. A12, pp. 10 170–10, 1983.

[76] ——, "Msis-86 thermospheric model," *Journal of Geophysical Research*, vol. 92, no. A5, pp. 4649–4662, 1987.

[77] ——, "Extension of the msis thermosphere model into the middle and lower atmosphere," *Journal of Geophysical Research*, vol. 96, no. A2, pp. 1159–1172, 1991.

[78] D. Drob, J. Emmert, G. Crowley, J. Picone, G. Shepherd, W. Skinner, P. Hays, R. Niciejewski, M. Larsen, C. She *et al.*, "An empirical model of the earth's horizontal wind fields: Hwm07," *Journal of Geophysical Research*, vol. 113, no. A12, p. A12304, 2008.

[79] D. Weimer, "Improved ionospheric electrodynamic models and application to calculating joule heating rates," *Journal of geophysical research*, vol. 110, no. A5, p. A05306, 2005.

[80] T. Fuller-Rowell and D. Evans, "Height-integrated pedersen and hall conductivity patterns inferred from the tiros-noaa satellite data," *Journal of Geophysical Research*, vol. 92, no. A7, pp. 7606–7618, 1987.

[81] GFZ German Research Centre for Geosciences, "The CHAMP mission," 2012, [accessed 18-December-2012]. [Online]. Available: http://op.gfz-potsdam.de/champ/

[82] University of Texas Center for Space Research, "Gravity recovery and climate experiment," 2012, [accessed 18-December-2012]. [Online]. Available: http://www.csr.utexas.edu/grace/

[83] D. J. Pawlowski and A. J. Ridley, "Modeling the thermospheric response to solar flares," *J. Geophys. Res.*, vol. 113, p. A10309, 2008.

[84] P. G. Richards, J. A. Fennely, and D. G. Torr, "EUVAC: A solar EUV flux model for aeronomic calculations," *J. Geophys. Res.*, vol. 99, pp. 8981–8992, 1994.

[85] E. D. Sumer, A. M. D'Amato, A. V. Morozov, J. B. Hoagg, and D. S. Bernstein, "Robustness of retrospective cost adaptive control to markov-parameter uncertainty," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 6085–6090.

[86] J. B. Hoagg and D. S. Bernstein, "Retrospective cost adaptive control for nonminimum-phase discrete-time systems, part 1: The ideal controller and error system," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 893–898.

[87] ——, "Retrospective cost model reference adaptive control for nonminimum-phase systems," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 6, pp. 1767–1786, 2012.