Non-linear Discrete-time Observer Design By Sliding Mode

A thesis submitted for the degree of
Doctor of Philosophy

By

Mohammed Algarawi

School of Engineering and Design,
Brunel University
September 2006

Abstract

Research into observer design for non-linear discrete-time systems has produced many design methods. There is no general design method however and that provides the motivation to search for a new simple and realizable design method.

In this thesis, an observer for non-linear discrete-time systems is designed using the sliding mode technique. The equation of the observer error is split into two parts; the first part being a linearized model of the system and the second part an uncertain vector.

The sliding mode technique is introduced to eliminate the uncertainty caused by the uncertain vector in the observer error equation. By choosing the sliding surface and the boundary layer, the observer error is attracted to the sliding surface and stays within the sliding manifold. Therefore, the observer error converges to zero.

The proposed technique is applied to two cases of observers for non-linear discrete-time systems. The second case is chosen to be a particular practical system, namely the non-linear discrete-time ball and beam system. The simulations show that the sliding mode technique guarantees the convergence of the observer error for both systems.

Dedication

To

My late father,

My mother
for her faithful prayers,

My wife
for her endless support and encouragement,

And to my children
Faris, Abdullah and Faiy

Acknowledgements

My continuous thanks are for Allah, the creator of the universe. Then I would like to express my appreciation to my supervisor, Dr. Peter Turner, for his encouragement, discussions, and guidance. I also appreciate his careful reading and corrections of the thesis were invaluable in its preparation.

I would like also to express my appreciation to my previous supervisor, Dr. John Marsh, who aroused my interest in sliding mode control, for his discussions and guidance.

My gratitude goes also to my colleagues at the Department of Electronic and Computer Engineering of Brunel University and all my friends who have contributed to this work one way or another.

Contents

List of Figures

List of Symbols

$<, >$	Less than, greater than
$\leq, \geq$	Less than or equal, greater than or equal
$\in$	Belongs to, in the set
$\subset$	Subset of
e	Exponential
$\forall$	For any
$\exists$	There exists
$\Rightarrow$	Implies that
$\lVert \cdot \rVert$	norm of matrix
$\Re^n$	Real n-dimensional space
$\Re^{n \times m}$	Set of real (n x m) matrices
$\square$	End of proof

Chapter 1
Introduction

1.1 Motivation

The study of control systems is an old subject, even before the eighteen century when James Watt invented the centrifugal governor for the speed control of a steam engine [1], [2]. In the late 1950's, various techniques in modern control theory were developed using state variables [1], [3], [4]. When the state variable technique is used to design a control system, state variable feedback of the state vector, which either is known or can be measured, is fedback to control the process.

In the operation of real control systems, knowledge of the values of the state variables of the system is usually a prerequisite for feedback control. In practice, direct measurements of all states required for state feedback are not available. Therefore, it is necessary to estimate the system states from the available measurements of the system output.

Estimation of the state of a dynamical system can be done using an estimator or an observer. The design of state observers is a fundamental problem in modern systems theory. State observers can be designed for both continuous-time and discrete-time systems. In addition, they can be designed for linear and non-linear systems.

Due to the extensive use of digital computers in control systems, research into controllers and observers for non-linear discrete-time systems is increasing.

A discrete-time observer utilizes sequences of both the input and output of a system to produce an estimate of the state vector of the system in a finite number of sampling periods.

The development of theory for state observer design for non-linear discrete-time systems has been a subject of intense research [9], [15], [32], [33], [35], [38], [39], [40], [44]. Even so, there is no general design method, and this provides the motivation to search for a new simple and realisable design method for an observer for non-linear discrete-time plants.

In control system theory, a method of designing a practical observer for a non-linear discrete-time system must achieve:

1. **Consistency.** Convergence of the observer error must be guaranteed.

2. **Robustness**. The observer must be stable in the presence of bounded uncertainties.

3. **Computational complexity**. The computational procedures should be simple and the number of computations should not be excessively large.

1.2 Review of Non-linear Systems

The subject of non-linear control deals with the analysis and design of control systems for non-linear systems. A non-linear system is a system containing at least one non-linear component [5]. In state space, a non-linear continuous-time system can be described by a first-order vector differential equation of the form,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{1.1}$$

termed the state equation, where $\boldsymbol{f}(\cdot)$ is a continuously differentiable function, $\boldsymbol{x} \in \Re^n$ is the state vector, and $\boldsymbol{u} \in \Re^m$ is the input vector, and a non-linear algebraic equation of the form,

$$\boldsymbol{y}(t) = \boldsymbol{h}(\boldsymbol{x}(t),\ \boldsymbol{u}(t)) \tag{1.2}$$

termed the output equation where $\boldsymbol{y} \in \Re^p$ is the output vector of the system.

A non-linear continuous-time observer can be described by an equation of the form,

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{g}(\boldsymbol{z}(t),\ \boldsymbol{u}(t),\ \boldsymbol{y}(t)), \tag{1.3}$$

A non-linear system in discrete-time can be described by a first order vector difference equation of the form,

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k)) \tag{1.4}$$

and a non-linear algebraic equation of the form,

$$\boldsymbol{y}(k) = \boldsymbol{h}(\boldsymbol{x}(k),\ \boldsymbol{u}(k)) \tag{1.5}$$

A very large part of the systems literature of theory is concerned with the study of linear systems. For a linear system, the equations (1.1) and (1.2) take the special form,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{1.6}$$

$$y(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \tag{1.7}$$

where $\boldsymbol{A}$ is an (n x n) real matrix, $\boldsymbol{B}$ is an (n x m) real matrix, $\boldsymbol{C}$ is a (p x n) real matrix, and $\boldsymbol{D}$ is a (p x m) real matrix.

Robustness is one of the essential concepts in control theory. Robust control is a design method that focuses on the reliability (robustness) of the control algorithm [5]. Robustness is usually defined as the minimum requirement a control system has to satisfy to be useful in a practical environment. Once the controller is designed, its parameters do not change and control performance is guaranteed. The design of a robust control system is typically based on the worst case scenario, so that the system usually does not work at optimal status in sense of control performance under normal circumstances [69]. Robust control methods are well suited in applications where the control system stability and reliability are the top priorities, process dynamics are known, and variation ranges for uncertainties can be estimated [70]. Aircraft and spacecraft controls are some examples of these systems.

In the control engineering literature, many methods such as feedback linearization and sliding mode control etc. have been developed to deal with the non-linearity in non-linear systems. Some of these methods will be discussed briefly in sections 1.3 and 1.4. The design of controllers and observers for non-linear continuous-time and discrete-time systems are briefly reviewed in this chapter. Robots [6], [7], [8], Underwater vehicle [9], hydraulic servosystems, induction motors

[10], [11], [12], and satellite attitude control [13] are examples of non-linear systems.

1.3 Control of Non-linear Systems

The control of continuous-time non-linear systems has been well studied [5], [14]. There are many techniques which have been developed to design the controllers for non-linear systems such as linearization, adaptive control and sliding mode control etc. Most of these techniques involve manipulating the non-linear system to be similar to a linear system because of the powerful analysis tools which have been developed for linear systems [5], [14].

Some of these techniques are briefly discussed as follows:

1.3.1 The linearization method

1.3.1.1 The linearization method is a formalization of the argument that a non-linear system should behave similar to a linear system (i.e. the linearized approximation) for a region about the operating point [5]. The linearized approximation can then be used to analyse and learn about the behaviour of the non-linear system [5], [14].

Consider the non-linear system given in equation (1.1) which can be written as,

$$\Delta \dot{x}(t) = \left[\frac{\partial f(x(t),\ u(t))}{\partial x} \right]_{(x=x(0),\ u=u(0))} (x(t) - x(0))$$
$$+ \left[\frac{\partial f(x(t),\ u(t))}{\partial u} \right]_{(x=x(0),\ u=u(0))} (u(t) - u(0))$$
$$+ f_{h.o.t.}$$
$$(1.8)$$

where $f_{h.o.t.}$ denote the higher order terms in x and u, and $(x(0),\ u(0))$ is the equilibrium point.

Let,

$$A = \left[\frac{\partial f(x(t),\ u(t))}{\partial x} \right]_{(x=x(0),\ u=u(0))} \tag{1.9}$$

$$B = \left[\frac{\partial f(x(t),\ u(t))}{\partial u} \right]_{(x=x(0),\ u=u(0))} \tag{1.10}$$

Then, the linearized model of the original non-linear system (2.1) is,

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1.11}$$

so, the stable design of the linear system guarantees the local stability of the original non-linear system [5].

The linearization method has some limitations, which are,

1. Linearization is an approximation in the neighbourhood of the operating point. Therefore, it only predicts the local

behaviour of the non-linear system in the vicinity of the operating point [14].

2. The dynamics of the non-linear system are much richer than the dynamics of the linear system. In other words, there are some phenomena that can take place only in the presence of non-linearity and that can not be described or predicted by the linear system. Such phenomena include finite escape time, limit cycles, etc. [14], [15].

   These limitations of the linearization method may mean it is not adequate for the study of certain non-linear systems. Hence, other methods have been developed for the analysis of non-linear systems.

1.3.1.2 **The extended linearization method.** It is used when the control system must be controlled over a wide range of operating points. Even though the extended linearization method can produce satisfactory system behaviour, it does not guarantee any global stability [15].

1.3.1.3 **Pseudolinearization.** It was introduced by Reboulet *et al* [16] which is an approximation of a non-linear system using Taylor series expansion. State feedback and state coordinate are derived which enable the resulting closed-loop state equation in the new coordinate to have a family of Jacobian linearizations that is independent of operating point. The application of the extended linearization and Pseudolinearization methods are limited to simple non-linear systems because they require exact knowledge of the system dynamics heavy computation for implementation.

1.3.1.4 The feedback linearization method. It is an approach to non-linear system design based on the idea of using state feedback to transform a non-linear system into a linear one. This transformation is achieved by exact state transformation and feedback which is different than the conventional linearization that is achieved by linear approximations of the dynamics [5].

Full (complete) linearization is achieved by input-state feedback linearization; on the other hand, partial linearization is achieved by input-output feedback linearization [5].

The feedback linearization method also has some limitations [5]:

1. The full state has to be measured.
2. It can not be used for all non-linear systems.
3. The robustness is not guaranteed in the presence of uncertainty or unmodeled dynamics.

1.3.2 Adaptive control

In early 1950s there was extensive research on adaptive control in connection with the design of the autopilot for high performance airplane but after correct proofs of stability of adaptive systems in early 1980s, investigation of the necessity of these assumptions has sparked interesting research into the robustness of adaptive control [17] and also when microprocessor enabled cost-effective implementations.

An adaptive control system can be defined as a feedback control system intelligent enough to adjust its characteristics in a changing environment so as to operate in an optimal manner according to some specified criteria.

An adaptive controller differs from an ordinary controller in that the controller parameters are variable, and there is a mechanism for adjusting these parameters online based on signals in the system [5], [17]. On other words, it is assumed in adaptive system that the parameters are adjusted all the times [17].

There are many approaches to design adaptive controllers such as Model-reference adaptive control (MRAC), self-tuning controllers (STC), and gain scheduling [17].

The Model-reference adaptive control, which is shown in figure (1.1), is one of the major approaches in adaptive control. The desired performance is expressed as a reference model, which gives the wished response to an input signal. The adjustment mechanism changes the parameters of the regulator by minimizing the error between the system output and the reference model. If the error is equal to zero, then the perfect model following is achieved [18].

The self-tuning controllers proceeds in two stages: The parameters of the system are estimated first in real time from the input to the output of the system. Then a control signal is generated based on the parameter estimation and control algorithm [19].

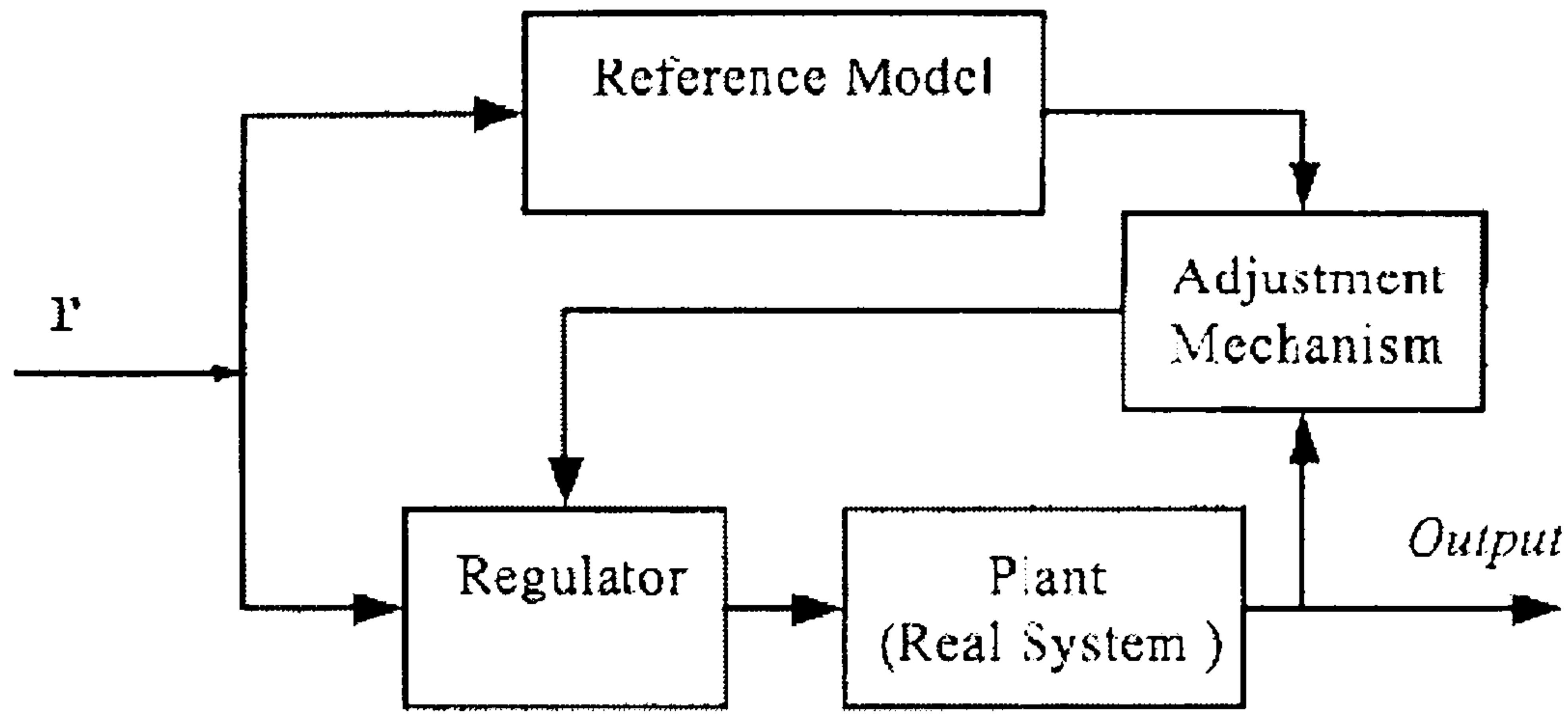

Figure 1.1: Block diagram of the Model-reference adaptive control (MRAC) [18]

Gain scheduling is an open-loop compensation. On other word, it is a system with feedback control in which the feedback gain is adjusted by feedforward compensation [17]. Gain scheduling is a very useful technique. It has the drawback that it may require a considerable effort to obtain a schedule by performing control design for many different operating conditions. But auto-tuning can conveniently be used to generate gain schedules semi-automatically by repeating automatic tuning at several operating conditions that cover the full operating range. Industrial experience has shown that substantial improvements for simple control loops can be obtained with schedules that only have a few entries [20], [21].

A great number of researches in control systems applications are in adaptive control such as aerospace [22], process control, ship steering [23], [24], robotics [25] and other industrial control systems.

Some adaptive controllers suffer from several problems such as: [18]

1. They do not track time-varying parameters very well.
2. Typically only asymptotic results are proved may be poor.

3. The control signal can get quite large in comparison to what the control signal would be if the plant parameters and state were known and the ideal LTI compensator was applied.

The adaptive control can be very useful and give good closed loop performance. However, that does not mean that adaptive control is the universal tool that should always be used [17]. also, it should be pointed out that the use of adaptive controller will not replace good knowledge, which is still needed to the specification, the structure of the controller, and the design method [17].

1.4 Review of Observers

The observers are indispensable tools for engineering. Their main function is extracting otherwise unmeasurable variables for a vast range of applications such as feedback control [87] and system health monitoring [88]. In engineering practice, an observer is used for a number of purposes, such as removing phase lag in feedback, reducing the use of costly sensors [88] and estimating disturbances [89]. There are many techniques which have been developed to design an observer for linear and non-linear continuous-time and discrete-time systems. These techniques can be classified into three categories, linear observers, non-linear observers and disturbance observer. The first two classes are concerned with state estimation based on a mathematical plant model; the other is concerned with disturbance estimation based on input output data. For the first two classes, sophistication of observer design gradually grew. Initially, it was found that a better estimate could be obtained if more accurate information was incorporated into the observer. This includes knowledge of noise and disturbances characterized by deterministic, differential [90], polynomial [91], bounded [73], and stochastic [30] descriptions. Consequently, many of these enhancements were proposed at the cost of detailed model information. In practice, it has been recognized that one can not rely entirely on mathematical models [28]. This leads to the third class of observers developed for practical disturbances [91], [92], [93]. This class of observers compliments the first two classes in practical control problems with significant nonlinearity and uncertainty. They are primarily motivated by the need for effective disturbance rejection in control of mechanical systems.

1.4.1 Linear Observer

1.4.1.1 Input Based Observer. If the output measurements for the system are not available, the input based observer is used to estimate the system states. If the input $\boldsymbol{u}$, the initial conditions are available and the system model in the observer is accurate, then the system states can be determined using only inputs.

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{u} \tag{1.12}$$

1.4.1.2 Output Based Observer. The system states can be estimated using the output measurements

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{L}(\boldsymbol{y} - \boldsymbol{C}\boldsymbol{z}) \tag{1.13}$$

where $\boldsymbol{L}$ is the observer gain and it is chosen such that the observer error is converge to zero by making the eigenvalues of $(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})$ to have negative real part. The low pass noise filter and approximate differentiator are types of the output based observer.

1.4.1.3 Luenberger Observer. Also known as closed loop observer. It is a combination of the input based observer and the output based observer method [26]. Since the initial conditions are not always available, the Luenberger observer uses the feedback of the estimated states and the measured data to eliminate the need of the initial conditions.

$$\dot{\boldsymbol{z}} = \boldsymbol{Az} + \boldsymbol{Bu} + \boldsymbol{L}(y - \boldsymbol{Cz}) \tag{1.14}$$

where $\boldsymbol{L}$ is the observer gain. The main advantage of this method is the ability of using the inputs, the output measurements and the system modal to reduce the noise and phase lag without knowledge of the initial conditions. Today, most of observers are based on the structure established by Luenberger observer with difference in methods of choosing the observer gain $\boldsymbol{L}$.

1.4.1.4 Proportional Integral Observer. It was developed by Beale *et al* [27] for linear system which is a modification of Luenberger observer by adding an integral gain $\boldsymbol{L}_i$ to Luenberger observer equation (2.14) to become,

$$\dot{\boldsymbol{z}} = \boldsymbol{Az} + \boldsymbol{Bu} + \boldsymbol{L}(y - \boldsymbol{Cz}) + \boldsymbol{L}_i \int (y - \boldsymbol{Cz}) \tag{1.15}$$

The main idea of using the extra integral gain is to enhance the correction term by accumulating observer error by time.

1.4.2 Non-linear Observer

1.4.2.1 Non-linear Luenberger observer. it is a modification of the linear Luenberger observer. It will estimate the state $\boldsymbol{x}$(t) using the input and measured output data and feedback the estimated state along with the measured data. The Luenberger observer established the structure the structure of most observers are based with a difference of choosing the observer feedback gain L. The Luenberger observer is limited by the requirement that non-linear system knowledge is known and it is not explicitly designed to handle disturbances [28].

1.4.2.2 Kalman Filter. It was one of the first observers to include the formulation of disturbances [29], [30]. Some assumptions have to be made about the unknown disturbance, so the Kalman filter will minimizes the 2-norm of the observer error. The Kalman filter has not been widely applied to industrial applications, probably, due to the complexity of the implementation [28], [31].

1.4.2.3 Extended Kalman Filter. It was the first major effort to adapt the Kalman filter for non-linear systems. The linearization technique is used at each time step to get $\boldsymbol{A}(t)$ and $\boldsymbol{C}(t)$ to then be used in the standard Kalman filter. Also it involves introducing arbitrary diagonal matrices to take the approximation error into account [28], [32],[33].

1.4.2.4 Extended linearization method. Just as for control of non-linear system design, the extended linearization method [15] can be used to obtain linear error dynamics for the observer. The non-linear observer is constructed such that the eigenvalues of the linearized error equation are placed at specified values which are locally invariant with respect to any fixed operating point.

1.4.2.5 An extended Luenberger-type observer. This type of observer for single input single output (SISO) non-linear systems was developed by Gauthier *et al.* [35]. Under a certain assumption that the non-linear function is globally Lipschitz with respect to any norm, i.e. if

$$\|f(x)-f(z)\| \le \gamma \|x-z\| \tag{1.16}$$

is satisfied for any x and z where $\gamma>0$, then an exponential observer can be designed. Since this observer is a high gain observer, it is very sensitive to measurement noise [36]. De Leon-Morales *et. al* [37] modified this high gain observer by reducing the high gain characteristic to become more robust with respect to measurements noise and disturbances.

In Gauthier *et. al.*[71], a constant gain observer has been proposed for general single output systems that are uniformly infinitesimally observable under some regularity assumptions on the vector fields. However, the main shortcoming of the proposed observer is that its practical construction is difficult to realize because the computation of the observer gain is not direct and constructive [72].

1.4.2..6 Metric observer. It was developed by Lohmiller *et al.* in [9], [38], and [39]. These observers are described by Euler coordinates. This method enables a non-linear observer to be considered at a given point in the state space rather than individual trajectories. The dynamics of the observer can be shaped by changing the coordinate representation of the system.

1.4.2.7 H_∞ Observer: it is another type of observer which optimizes a cost function based on an assumption about the disturbance. This formulation is significant because it uses a unique characterization of the disturbance. Kalman minimizes the minimum squared error because it is a mathematically manageable optimization problem. Using infinity norms, the $H\infty$ observer is able to minimize the maximum or worst case disturbance [73], [74], [75]. The observer is guaranteed to be optimal under a user defined upper bound γ.

1.4.2.8 Sliding mode observer. It was developed by Slotine *et al.* , [35], [40], [41], [42]. Sliding mode observers differ from Luenberger and other observers in that there is a non-linear discontinuous term injected into the observer depending on the observer error. The sliding mode observers are more robust as the discontinuous term enables the observer to reject disturbances, and also a class of mismatch between the system and observer. The discontinuous term is designed to drive the trajectories of the observer so that the state estimation error vector is forced onto and subsequently remains on sliding surface. In most cases, the sliding surface is set to be the difference between the observer and system output which is therefore forced to zero. When a sliding mode is achieved the system will experience a reduced order motion which is insensitive to a class of system mismatch. On other words, It is based on the attractive manifold that will attract the observer error to slide to zero once the error trajectories reach the sliding surface regardless of any uncertainties or disturbances [1], [2], [41], [42]. This method will be discussed in detail in chapters 3 and 4.

1.4.3 Disturbance Observers

Since the introduction of the disturbance observer as an unknown input state space observer by Johnson, it has been widely used to deal with the disturbance on the systems [28], [55]. The mean idea of disturbance observer is to augment the system with a fictitious dynamical system which generates the disturbances, and then an observer is designed to estimate the states of both the system and the disturbance generator. By using the reconstructed disturbance state for feedback, disturbance rejection is accomplished.

As the feedback signal is only an estimate of the actual disturbance acting on the system, the disturbance observer does not control the plant which means that a normal feedback controller is still needed to achieve performance. The benefit of the disturbance observer is that it adds disturbance rejection to the nominal feedback controller without affecting the system performance.

1.4.3.1 Disturbance observer. It was studied by [28], [55] to estimate the disturbance. It is different from state observer because it estimates external disturbances and observer modal discrepancies that effectively appear at the system input. By feedback the estimated disturbance, the system disturbance rejection is accomplished. The Disturbance Observer is usually written in transfer function instead of state space form [28]. Figure (1.2) shows block diagram of a control system with a disturbance observer.

1.4.3.2 Unknown input observer. It was developed by Hostetter *et. al.* [43] to use the disturbance observer's concept in state space representation. By using state space equations, the Unknown input observer defines assumptions about the rate of disturbance changes. The disturbance input is made to satisfy a differential equation. The ability to estimate states and disturbances simultaneously is a practical advantage of the unknown input observer over the disturbance observers.

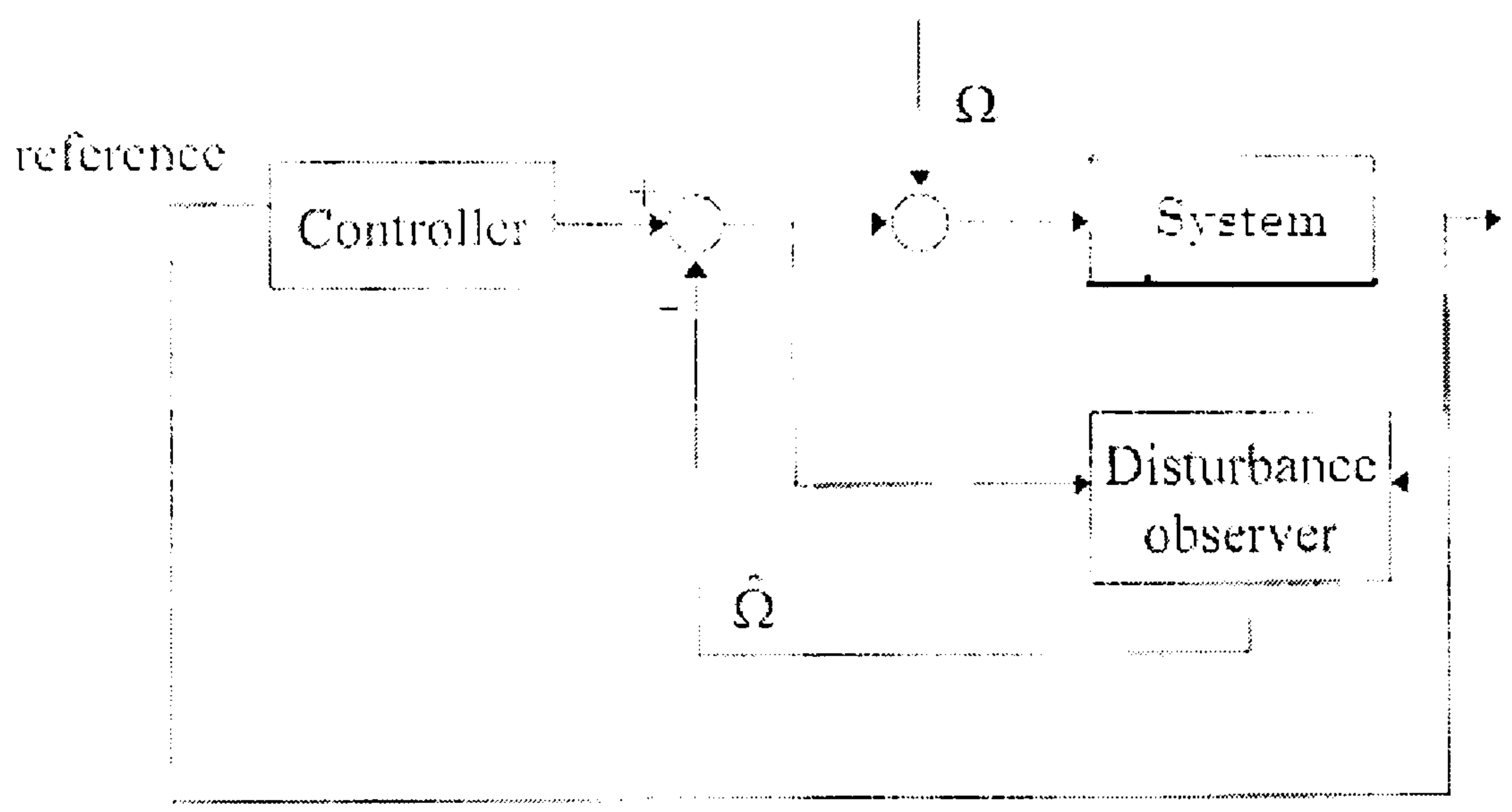

Figure 1.2: Control system with a disturbance observer

1.4.3.3 Perturbation observers. It estimates the unmodeled system variation in addition to external disturbances. The Lipschitz non-linear observer is type of the perturbation observers which was studied by Rajamani [44] for a class of non-linear systems represented by a linear time-invariant (LTI) system with a perturbation which is a Lipschitz non-linearity i.e.

$$\left\|\varphi(x,u)-\varphi(z,u)\right\|_2 \leq \gamma\left\|x-z\right\| \tag{1.16}$$

where φ is a Lipschitz non-linearity with a Lipschitz constant $\gamma > 0$. A sufficient condition for stability in terms of the eigenvalues and the eigenvectors is obtained from the linear stability matrix. An algorithm is presented for obtaining the observer gain matrix in order to achieve asymptotic stability.

1.4.3.4 **Extended State Observer**: Most observers are made to handle slight perturbations for a modelled system; however the extended state observer was designed to remove the requirement of a modelled system by rejecting un-modelled dynamics [28]. The extended state observer uses a simple canonical form so the un-modelled dynamics appear at the disturbance estimation portion. This decisively captures the subtle but important design methodology shift between modern estimators and disturbance estimators. It encompasses realistic disturbances and un-modelled plant variations while remaining simple.

After a half century of continuous research and development, observers have become an integral part of control theory and practice. Starting from linear observer to non-linear and disturbance observers proceeded with two distinct schools of thought: One, linear and non-linear estimation relies on a detailed mathematical model of the system and seeks optimal solutions. The other, disturbance estimation, acknowledges the limit of available partial system dynamic information, and seeks to estimate the disturbance, i.e. the discrepancy between the model and the real system. In some cases, the disturbance observers provide both the state and disturbance estimation. The model-based methods provide rigorous and, in many cases, optimal solutions. The disturbance estimation strategy is less known but addresses the uncertain nature of physical processes; and it seems to offer a more practical design framework to deal with real world control problems.

1.5 Research Aim and Objectives

The aim of the thesis is to present the design of an observer for non-linear discrete-time systems using sliding mode.

The main objectives are as follows:

1.5.1 To present a theoretical framework for the sliding mode technique for non-linear discrete-time systems.

1.5.2 To use the framework to design a sliding mode observer.

1.5..3 To illustrate the design technique on two non-linear discrete-time systems.

1.6 Organization of the Thesis

In chapter 2, the problem is formulated for a non-linear discrete-time system for which an observer is designed. The observer design method is applied to two case studies of controllers and observers for discrete-time non-linear systems and as expected the design method fails to guarantee the convergence of the observer error. is shown to diverge.

In chapter 3, the sliding mode technique is introduced and discussed as a way of solving the particular problem found in chapter 3. A literature review of the sliding mode technique for linear and non-linear systems is presented. As an introduction to the sliding mode technique, it is applied to a case of linear control system. The advantages and disadvantages of using the idea of sliding mode control are presented.

Chapter 4 is the main focus of the thesis where the sliding mode technique is applied to the design of an observer for non-linear discrete-time systems. The design is applied to the two case studies investigated in chapter 2.

Finally, in chapter 5, conclusions are given with a summary of the findings of the thesis and suggestions for future research.

Chapter 2

Problem Formulation

2.1 Introduction

The previous chapter discussed the control of non-linear systems and the associated problem of observers for non-linear systems. The main objective of any design method for an observer is to guarantee the convergence of the observer error to zero.

In order to analyze any observer design method, a system has to be set up to which the method is applied. In this chapter, observers for two cases of non-linear systems are studied to investigate the convergence of the observer error.

2.2 Non-linear Discrete-time Observer Design

Consider a non-linear discrete-time state space system as,

$$\boldsymbol{x}(k+1) = \boldsymbol{f}\left(\boldsymbol{x}(k), \boldsymbol{u}(k)\right) \tag{2.1}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) \tag{2.2}$$

where $\boldsymbol{f}(\cdot)$ is continuously differential, $\boldsymbol{x} \in \Re^n$ is the state vector, $\boldsymbol{u} \in \Re^m$ is the input, $\boldsymbol{y} \in \Re^p$ is the output of the system.

Assume that the system described by equations (2.1) and (2.2) is a single input and single output (SISO) system with a linear input u,

$$\boldsymbol{x}(k+1)=\boldsymbol{f}(\boldsymbol{x}(k))+\boldsymbol{B}u(k) \tag{2.3}$$

$$y(k)=\boldsymbol{C}\boldsymbol{x}(k) \tag{2.4}$$

where $\boldsymbol{B}$ is the input matrix and $\boldsymbol{C}$ is the output matrix. Assume that the pair $(\boldsymbol{A}_x,\boldsymbol{B})$ are controllable and the pair $(\boldsymbol{A}_x,\boldsymbol{C})$ are observable where $\boldsymbol{A}_x$ is the Jacobian matrix of $\boldsymbol{f}(\boldsymbol{x})$ around the equilibrium point $(\boldsymbol{x}_0,\boldsymbol{u}_0)$.

The system described by equations (2.3) and (2.4) has a linear combination of the input added to the states and the output is a linear combination of the states. This means that the states are used to model the non-linearities inherent in the dynamics of the system.

An observer can be designed to produce an estimate of the state vector, $\boldsymbol{z}(k)$, in the same form as the original system (2.3) with an additional input depending on the difference between the measured values and the estimated values of the output vector.

$$\begin{aligned}\boldsymbol{z}(k+1)&=\boldsymbol{f}(\boldsymbol{z}(k))+\boldsymbol{B}u(k)\\&\quad+\boldsymbol{l}(k)\big(y(k)-\boldsymbol{C}\boldsymbol{z}(k)\big)\end{aligned} \tag{2.5}$$

where $\boldsymbol{l}(k)$ is the vector of observer feedback gains. The dependence on k is due to the non-linearity of the system i.e. the observer feedback gain $\boldsymbol{l}$ changes over time and needs to be calculated at each time step, k.

A block diagram of the whole system is shown in fig. 2.1. Note that so far this follows the design of an observer for a linear system apart from the dependence of $\boldsymbol{l}$ on the time step, k.

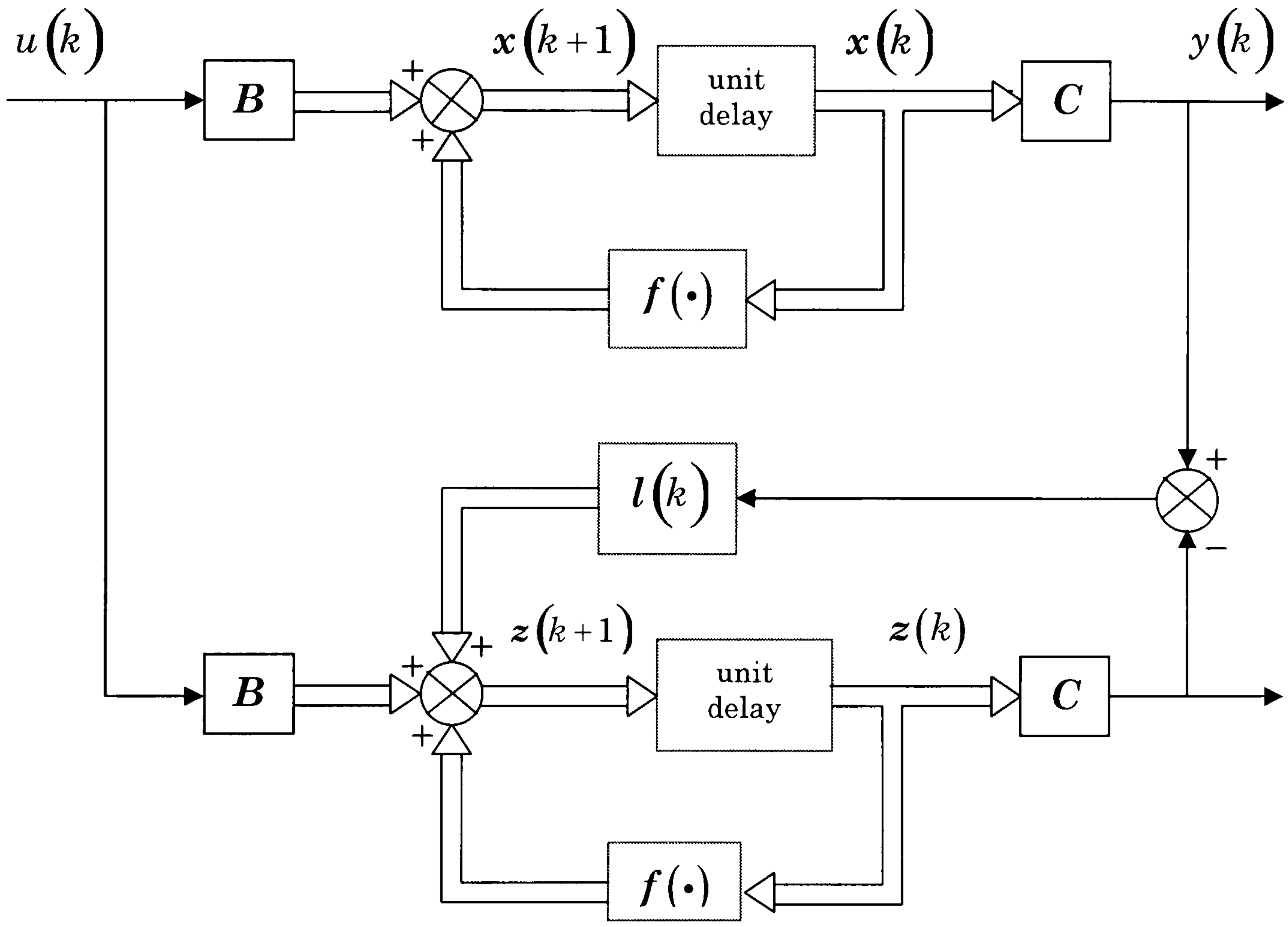

Figure 2.1: Block diagram for Non-linear Discrete-time Observer.

Let us define an observer error vector, $\boldsymbol{e}(k)$, as the difference between the true state, $\boldsymbol{x}(k)$, and the state estimate, $\boldsymbol{z}(k)$.

$$\boldsymbol{e}(k) = \boldsymbol{x}(k) - \boldsymbol{z}(k) \tag{2.6}$$

then,

$$e(k+1) = x(k+1) - z(k+1) \tag{2.7}$$

From equations (2.3) and (2.5), the observer error can be written,

$$e(k+1) = f(x(k)) - f(z(k)) - l(k)Ce(k) \tag{2.8}$$

Definition 2.1:

Let us define a compact set K such that $x(k), z(k) \in \mathrm{K}$ and assume that $f(\cdot): \mathrm{K} \subset \Re^n \to \Re^n$ has continuous partial derivatives of second order over K, where $f(0)=0$ and K contains the origin.

By using Taylor series expansion [18], let $f(x(k))$ be expanded about $z(k)$,

$$\begin{aligned} f(x(k)) = f(z(k)) + A_z(k)e(k) \\ + R_k(e(k), z(k)) \end{aligned} \tag{2.9}$$

where $R_k(e(k), z(k))$ is the Taylor residual and

$$A_z(k) = \left.\frac{df(k)}{dx}\right|_{x=x(k)} = \left.\frac{df(k)}{dz}\right|_{z=z(k)} \tag{2.10}$$

Since the continuous function is bounded when it is evaluated over a compact set and from definition (2.1), the function $f(\cdot)$ is bounded whenever $\boldsymbol{x}(k), \boldsymbol{z}(k) \in \mathbf{K}$. Therefore the Taylor residual $\boldsymbol{R}_k$ is bounded which means that there exists a positive constant β such that

$$\left\|\boldsymbol{R}_k\right\| \leq \beta\left\|\boldsymbol{e}(k)\right\|^2 \tag{2.11}$$

Substitute equation (2.9) into equation (2.8), to yield,

$$\boldsymbol{e}(k+1)=\left[\boldsymbol{A}_{\boldsymbol{z}}(k)-\boldsymbol{l}(k)\boldsymbol{C}\right]\boldsymbol{e}(k)+\boldsymbol{R}_k\left(\boldsymbol{e}(k),\boldsymbol{z}(k)\right) \tag{2.12}$$

Therefore, from equation (2.12) the observer error consists of two parts, the first part is $\left[\boldsymbol{A}_{\boldsymbol{z}}(k)-\boldsymbol{l}(k)\boldsymbol{C}\right]\boldsymbol{e}(k)$ and the second part is the Taylor's residual $\boldsymbol{R}_k\left(\boldsymbol{e}(k),\boldsymbol{z}(k)\right)$, which is an uncertain vector. In the case of a linear time invariant system, the observer error equation is,

$$\boldsymbol{e}(k+1)=\left[\boldsymbol{A}-\boldsymbol{l}\boldsymbol{C}\right]\boldsymbol{e}(k) \tag{2.13}$$

where the observer feedback gain $\boldsymbol{l}$ can be calculated using the pole placement method provided the pair $(\boldsymbol{A},\boldsymbol{C})$ is observable. Therefore, the non-linear observer equation (2.13) consists of $\left[\boldsymbol{A}_{\boldsymbol{z}}(k)-\boldsymbol{l}(k)\boldsymbol{C}\right]\boldsymbol{e}(k)$, a linearized part, which is similar to the linear case and the observer feedback gain, $\boldsymbol{l}(k)$, can be calculated using pole placement method at assigned observer closed-loop poles at each time step (k). The Taylor's residual $\boldsymbol{R}_k\left(\boldsymbol{e}(k),\boldsymbol{z}(k)\right)$, which is the second part of the observer error equation (2.12), is an uncertain vector that may cause the divergence of the observer error.

2.3 Case Studies

In this section, the designs of two non-linear discrete-time observers are presented. They are studied to investigate the convergence of the observer error of these systems.

2.3.1 Case 1

Consider a non-linear discrete-time system as,

$$\boldsymbol{x}(k+1)=\boldsymbol{f}(\boldsymbol{x}(k))+\boldsymbol{B}u(k) \tag{2.14}$$

$$y(k)=\boldsymbol{C}\boldsymbol{x}(k) \tag{2.15}$$

where,

$$\boldsymbol{f}(\boldsymbol{x}(k))=\begin{bmatrix} 0.85x_1(k)+0.5x_2(k) \\ 0.1x_1^2(k)+0.3x_2(k) \end{bmatrix} \tag{2.16}$$

Given the input vector,

$$\boldsymbol{B}=\begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.17}$$

and measurement vector,

$$\boldsymbol{C}=[0.5 \;\; -2.2], \tag{2.18}$$

Let the observer for the non-linear system described by equations (2.14) and (2.15) be described as

$$\begin{aligned} \boldsymbol{z}(k+1) = \boldsymbol{f}\left(\boldsymbol{z}(k)\right) + \boldsymbol{B}\boldsymbol{u}(k) \\ + \boldsymbol{l}(k)\left(y(k) - \boldsymbol{C}\boldsymbol{z}(k)\right) \end{aligned} \tag{2.19}$$

where $\boldsymbol{x}(k) \in \mathcal{K}$, $\boldsymbol{z}(k) \in \boldsymbol{S}$ and $\kappa \subset \boldsymbol{S} \subset \Re^n$ is a compact set and $\boldsymbol{l}(k)$ is the vector of observer feedback gains.

Then the observer error can be described as

$$\boldsymbol{e}(k) = \boldsymbol{x}(k) - \boldsymbol{z}(k) \tag{2.20}$$

and from equation (2.13),

$$\begin{aligned} \boldsymbol{e}(k+1) = \left[\boldsymbol{A}_z(k) - \boldsymbol{l}(k)\boldsymbol{C}\right]\boldsymbol{e}(k) \\ + \boldsymbol{R}_k\left(\boldsymbol{e}(k), \boldsymbol{z}(k)\right) \end{aligned} \tag{2.21}$$

The Jacobian matrix $\boldsymbol{A}_z(k)$ is obtained as follows:

$$\boldsymbol{A}_{\boldsymbol{z}}(k) = \left.\frac{d\boldsymbol{f}(\boldsymbol{z})}{d\boldsymbol{z}}\right|_{\boldsymbol{z}=\boldsymbol{z}(k)} \tag{2.22}$$

$$\boldsymbol{A}_z(k) = \begin{bmatrix} 0.85 & 0.5 \\ 0.2z_1(k) & 0.3 \end{bmatrix} \tag{2.23}$$

The state trajectory of this system is shown in figure (2.2) with a system input,

$$u(k) = 0.05\ ,\ k \geq 0, \tag{2.24}$$

i.e. a step input of size 0.05

and initial conditions selected as:

$$\boldsymbol{x}(0)=\begin{bmatrix}0.9\\0.7\end{bmatrix} \tag{2.25}$$

An observer can be implemented using the pole placement method. The observer initial conditions are selected as,

$$\boldsymbol{z}(0)=\begin{bmatrix}5\\4\end{bmatrix} \tag{2.26}$$

and the closed-loop poles of the observer are chosen to be,

$$\lambda= [-0.3751 \quad -0.6592] \tag{2.27}$$

Simulation code has been written using Matlab to simulate the observer of the non-linear discrete-time system (2.19). The observer feedback gain $l(k)$ is calculated using Ackermann's formula for pole placement with the *"Acker"* function in Matlab. The assigned observer closed-loop poles in equation (2.27) are kept constant and the observer gains are calculated at each time step (k).

Figure (2.2) shows that the non-linear discrete-time system (2.14) is stable for the system used. It should be noted that if a larger step input is used, the system is unstable. The observer trajectory is shown in figure (2.3). The observer error of the non-linear discrete-time observer (2.19) is shown in figure (2.4) where divergence of the observer error can be seen.

2.3.2 Case 2: Ball and Beam System

In order to study further the observer design method for non-linear discrete-time systems, a practical non-linear system is needed. A ball and beam non-linear system is chosen for this further investigation. A formulation of the non-linear discrete-time ball and beam system is described in Appendix B.

The state equation for the non-linear discrete-time ball and beam system can be shown to be of the form of equation (2.3) (Appendix B) i.e.

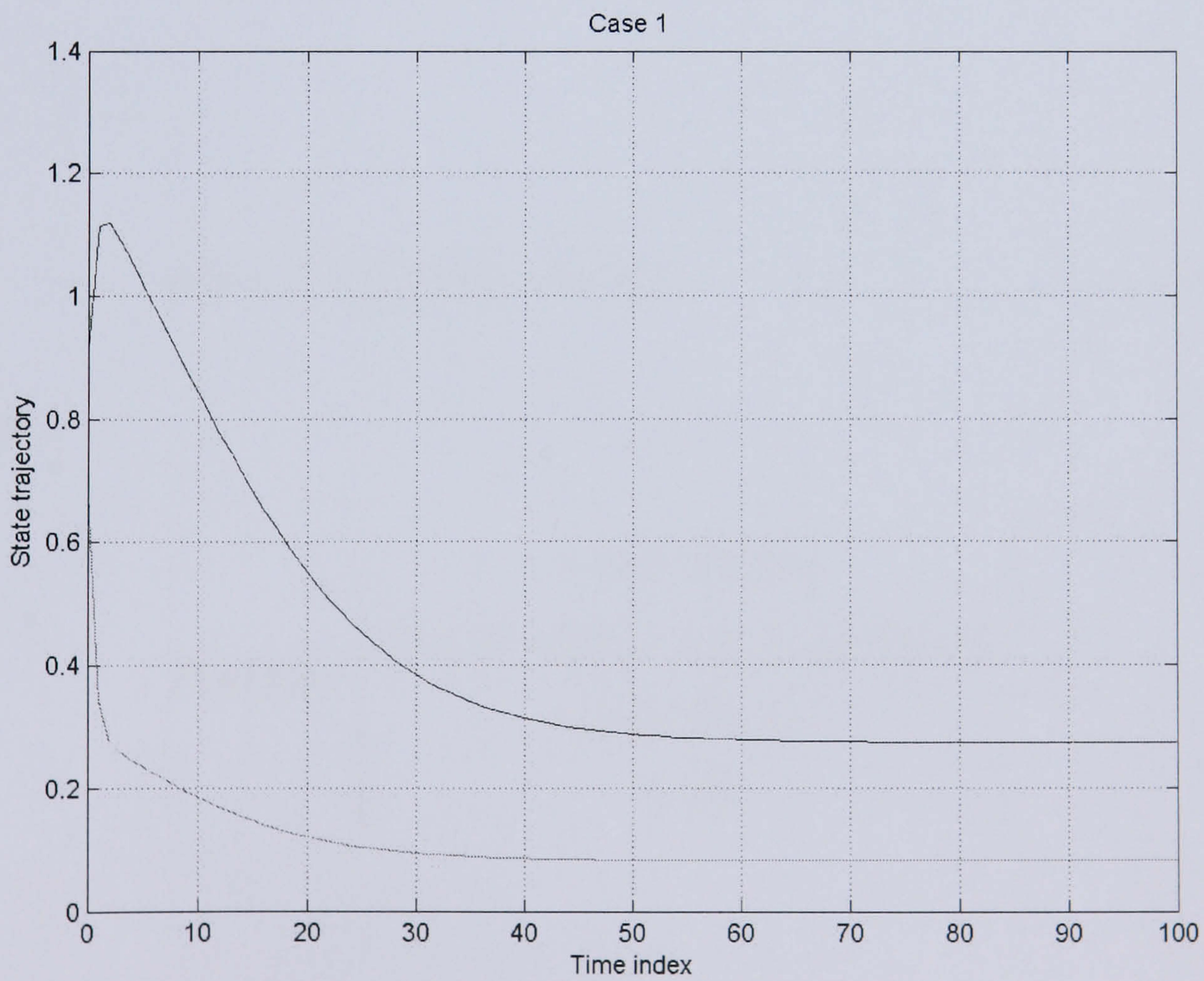

Figure 2.2: State trajectory for non-linear discrete-time system: case 1

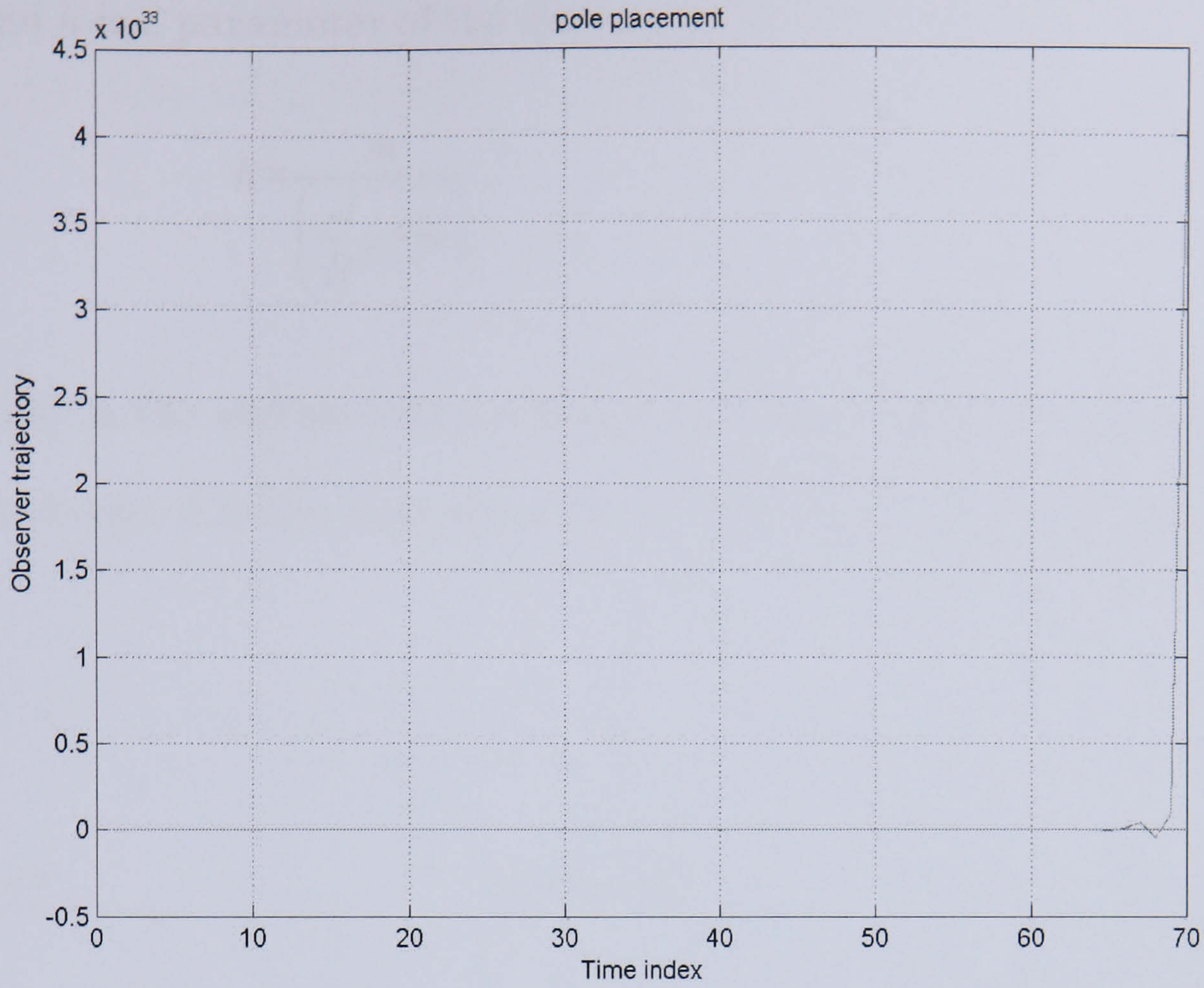

Figure 2.3: Observer trajectory for case 1

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k)) + \boldsymbol{B}u(k) \tag{2.29}$$

where

$$\boldsymbol{f}(\boldsymbol{x}(k)) = \begin{bmatrix} x_1(k) + Tx_2(k) \\ hTx_1(k)x_4^2(k) + x_2(k) - ghTx_3(k) \\ x_3(k) + Tx_4(k) \\ x_4(k) \end{bmatrix}$$

(2.30)

and h is a parameter of the system given by:

$$h = \frac{m}{\left(\frac{J}{R^2} + m\right)}, \tag{2.31}$$

$u(k)$ is the system input which is the torque applied to the centre of the beam, T is the sampling time, m is the ball's mass, R is the ball's

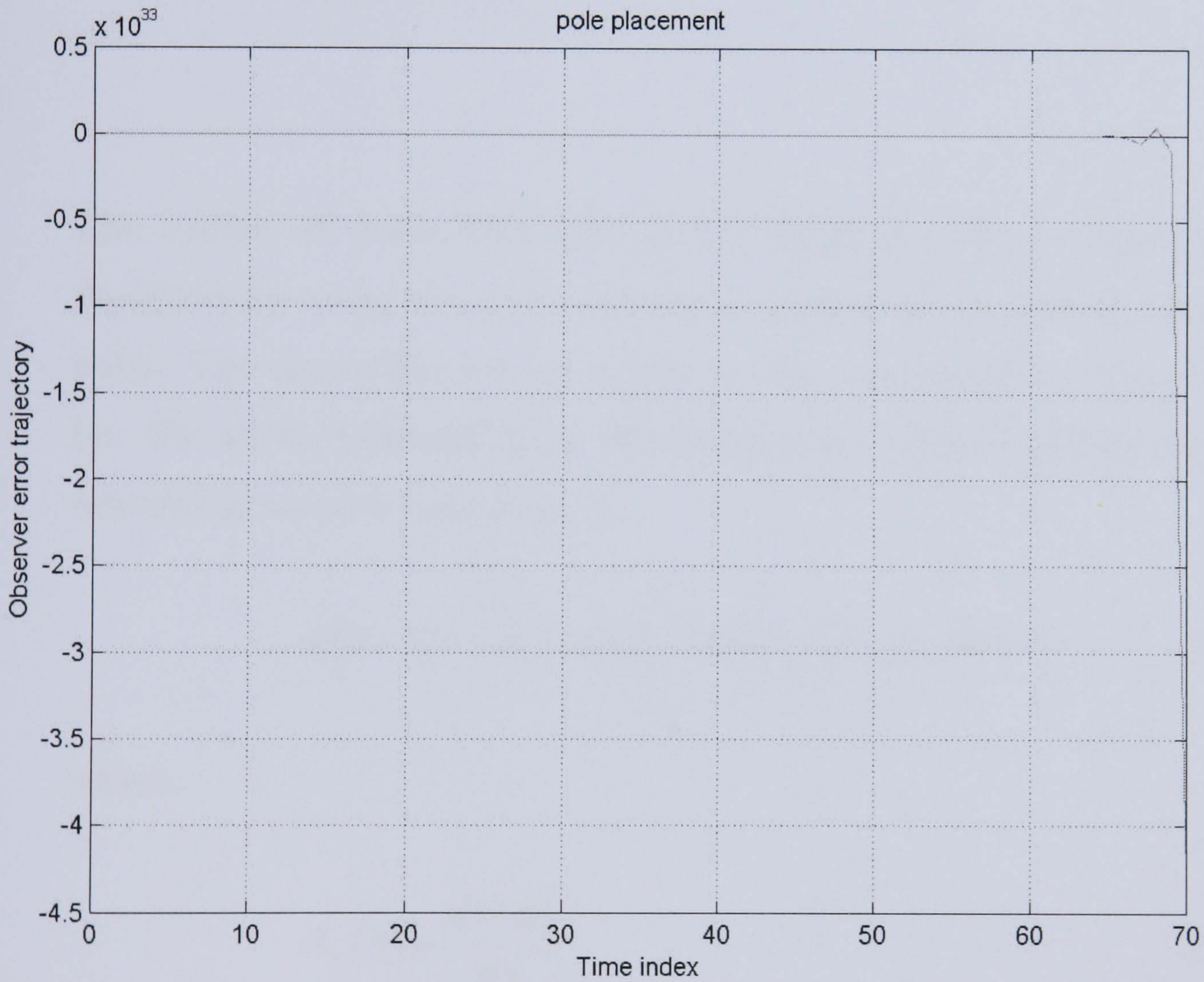

Figure 2.4: Observer error trajectory for case 1.

radius, J is the ball's moment of inertia and g is the acceleration due gravity, see table (B.1).

The input vector is,

$$\boldsymbol{B}=\begin{bmatrix}0\\0\\0\\\boldsymbol{T}\end{bmatrix} \tag{2.32}$$

and the measurement vector is,

$$\boldsymbol{C}=[\ 1\ 0\ 0\ 0], \tag{2.33}$$

The vector of state feedback gains, $\boldsymbol{M}(k)$, for this system can be designed by using the pole placement method for assigned closed-loop poles. The dependence on k is due to the non-linearity of the system i.e. the state feedback gain $\boldsymbol{M}$ changes over time and needs to be calculated at each time step, k.

$$\boldsymbol{x}(k+1)=\left[\boldsymbol{A}_x\left(\boldsymbol{x}(k)\right)-\boldsymbol{BM}(k)\right]\boldsymbol{x}(k)+\boldsymbol{B}r(k) \tag{2.34}$$

where,

$$\boldsymbol{A}_x(k)=\left.\frac{d\boldsymbol{f}(\boldsymbol{x})}{d\boldsymbol{x}}\right|_{x=x(k)} \tag{2.35}$$

and $r(k)$ is the reference input.

So, the non-linear discrete-time control system (2.34) can be rewritten as,

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k)) + \boldsymbol{B}r(k) \tag{2.36}$$

$$y(k) = \boldsymbol{C}\boldsymbol{x}(k) \tag{2.37}$$

The block diagram of the ball and beam control system (equations 2.36 and 2.37) is shown in figure (2.5).

The discrete-time ball and beam control system in equation (2.36) can be simulated using Matlab as before. The initial state vector is selected as,

$$\boldsymbol{x}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.38}$$

and the closed-loop poles are chosen as,

$$\boldsymbol{\lambda}_s = [0.9\ 0.95\ 0.95\text{-}j0.1\ \ 0.95\text{+}j0.1] \tag{2.39}$$

Again by using the pole placement method, the state feedback gain, $\boldsymbol{M}(k)$, is calculated using the "*Acker*" function in Matlab. Figure (2.6) shows the state trajectories of the non-linear discrete-time ball and beam system (2.33) when a step input $r(k)$ is applied. The trajectories show that the state feedback has stabilized the system with states $x_2(k)$, $x_3(k)$ and $x_4(k)$ settling at zero. This is to be expected as $x_2(k)$

and $x_4(k)$ are the velocities of the ball and beam respectively which should go to zero, and $x_3(k)$ is the beam angle which if non-zero would keep the ball moving i.e. $x_2(k) \neq 0$ and $x_1(k)$ would be changing. The ball position, $x_1(k)$ settles at a non-zero value determined by the reference input, $r(k)$.

Let the observer for the closed-loop non-linear system (2.36) be described as

$$\begin{aligned} \boldsymbol{z}(k+1) &= \boldsymbol{f}(\boldsymbol{z}(k)) + \boldsymbol{B}r(k) \\ &\quad + \boldsymbol{l}(k)\big(y(k) - \boldsymbol{C}\boldsymbol{z}(k)\big) \end{aligned} \tag{2.40}$$

where $\boldsymbol{l}(k)$ is the vector of observer feedback gains.

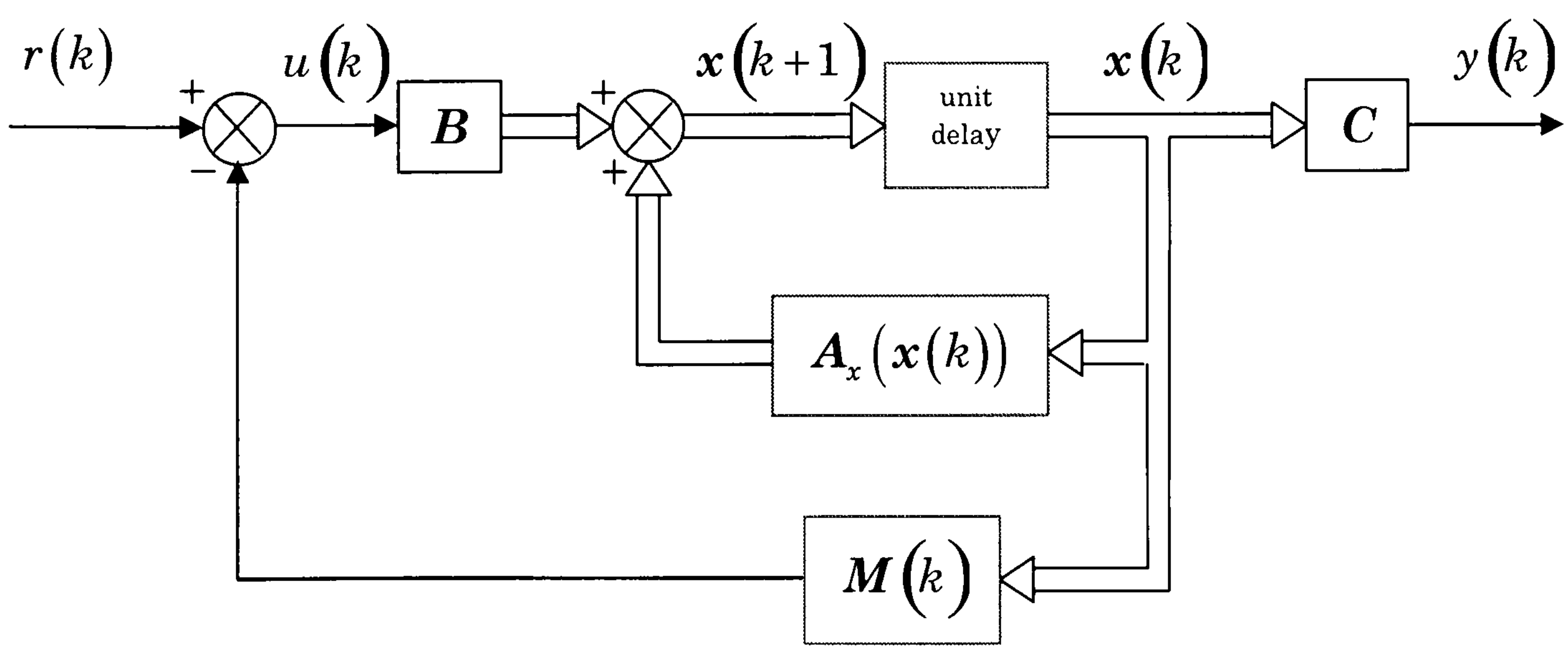

Figure 2.5: Block diagram for the non-linear discrete-time ball and beam control system.

The observer error is defined as before,

$$e(k) = x(k) - z(k) \tag{2.41}$$

and

$$e(k+1) = \left[A_z(k) - l(k)C\right]e(k) + R_k\left(e(k), z(k)\right) \tag{2.42}$$

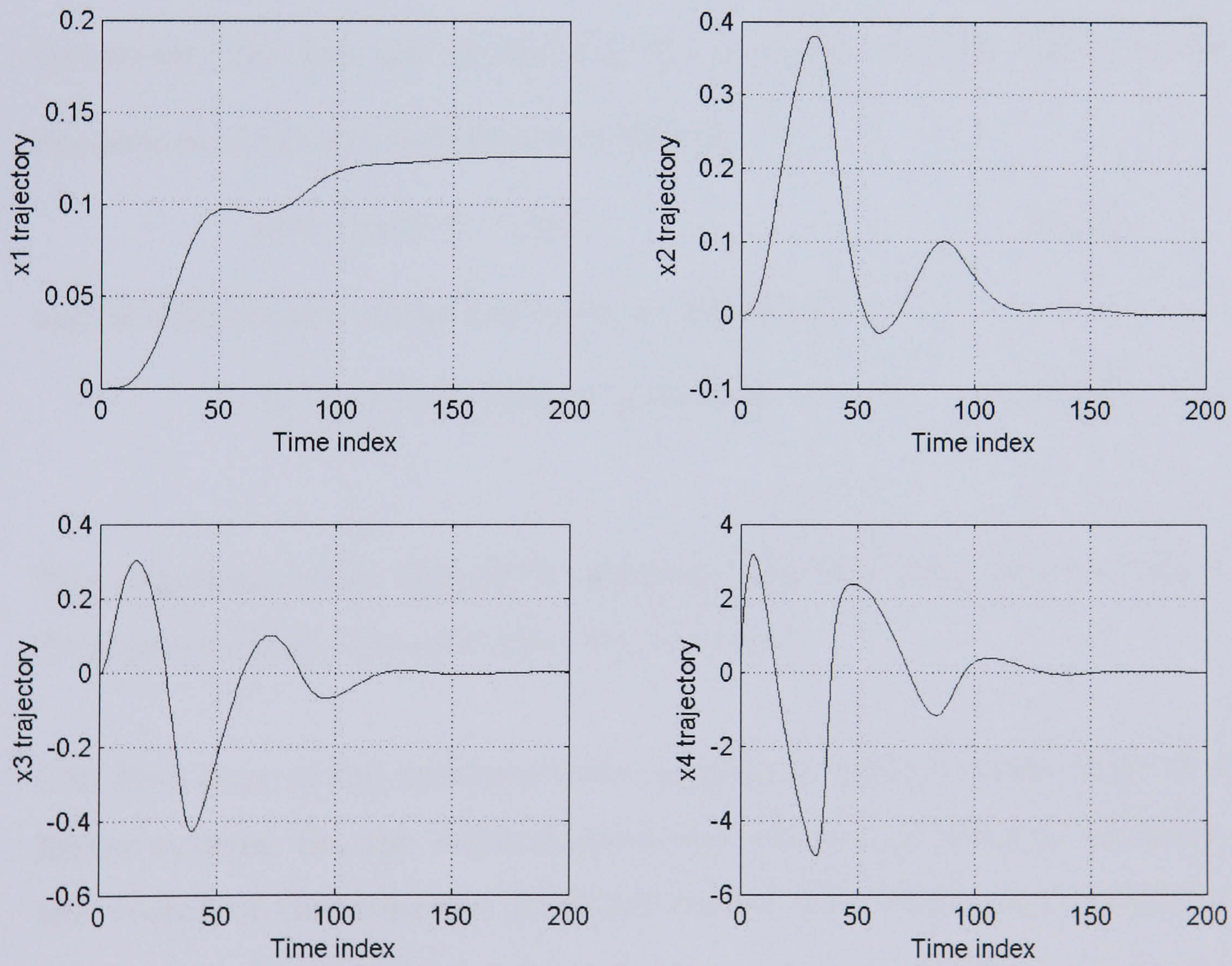

Figure 2.6: State trajectory for non-linear discrete-time Ball & beam control system.

The Jacobian matrix $A_z(k)$ is obtained as,

$$A_z(k) = \left. \frac{df(z)}{dz} \right|_{z=z(k)} \tag{2.43}$$

$$A_z(k) = \begin{bmatrix} 1 & T & 0 & 0 \\ hTz_4^2(k) & 1 & -ghT & 2hTz_1(k)z_4(k) \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & T \end{bmatrix} \tag{2.44}$$

By using the pole placement method, an observer for the non-linear discrete-time ball and beam (equation 2.34) is simulated using Matlab with the observer initial conditions selected as,

$$z(0)= [0.03 \;\; 0 \;\; 0 \; 0]]^T , \tag{2.45}$$

and the fixed poles of the observer are chosen to be:

$$\lambda_o= [0.8 \;\; 0.9 \;\; 0.9\text{-}j0.2 \;\; 0.9\text{+}j0.2] \tag{2.46}$$

The observer errors should converge to zero but from figure (2.8), a divergence of the observer error can be seen.

The first part of the observer error equation (2.41) is of the form of a linear system. So, the Ackermann's formula can be used to calculate the vector of the observer feedback gains, $l(k)$. Since the system is non-linear, this calculation must be done at each time step (k) as the Jacobian matrix, $A_z(k)$, is changing. The second part of equation (2.41) is the Taylor's residual, $R_k(e(k),z(k))$, which is an uncertain vector and which causes the divergence of the observer error.

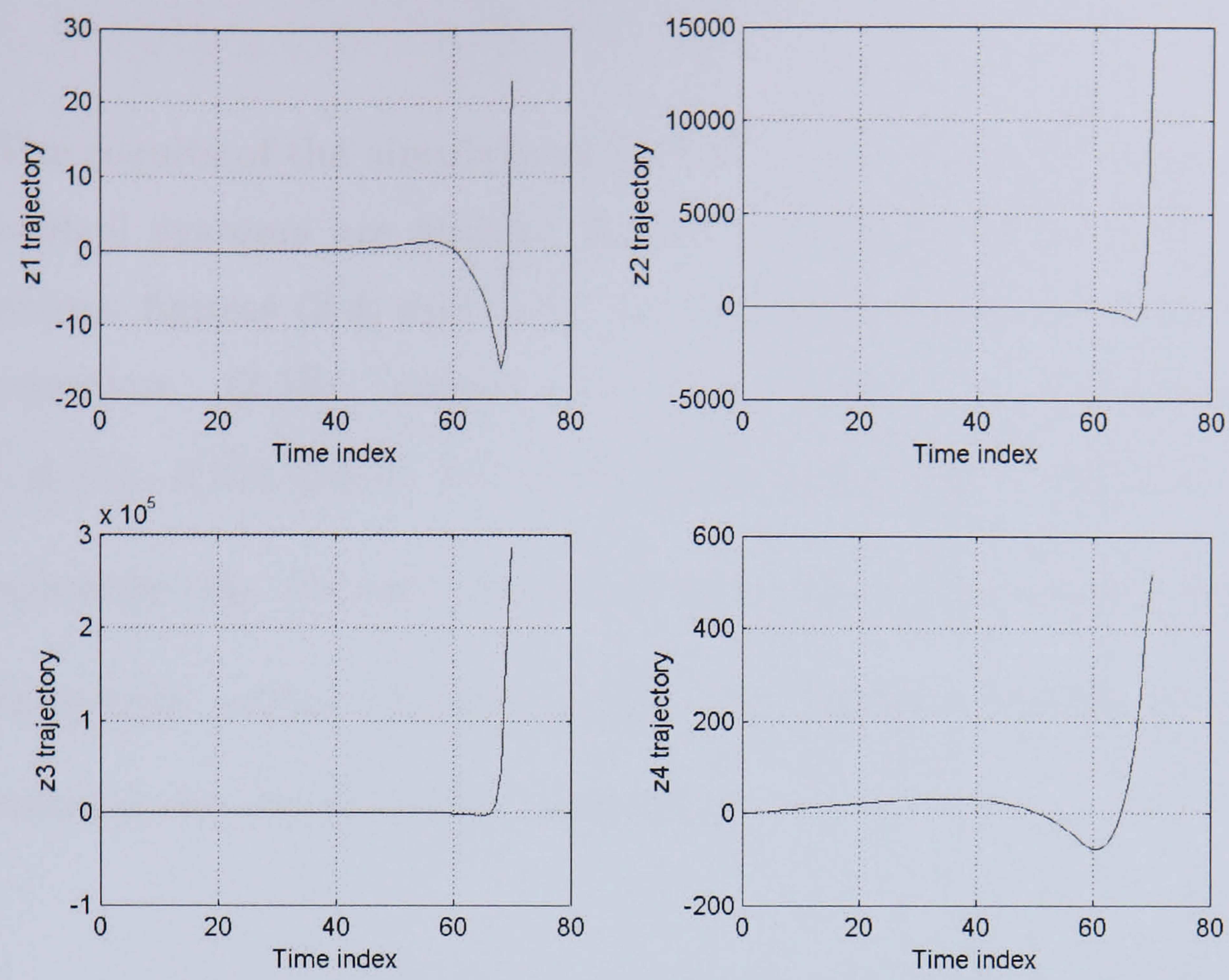

Figure 2.7: Observer trajectory of the ball and beam observer.

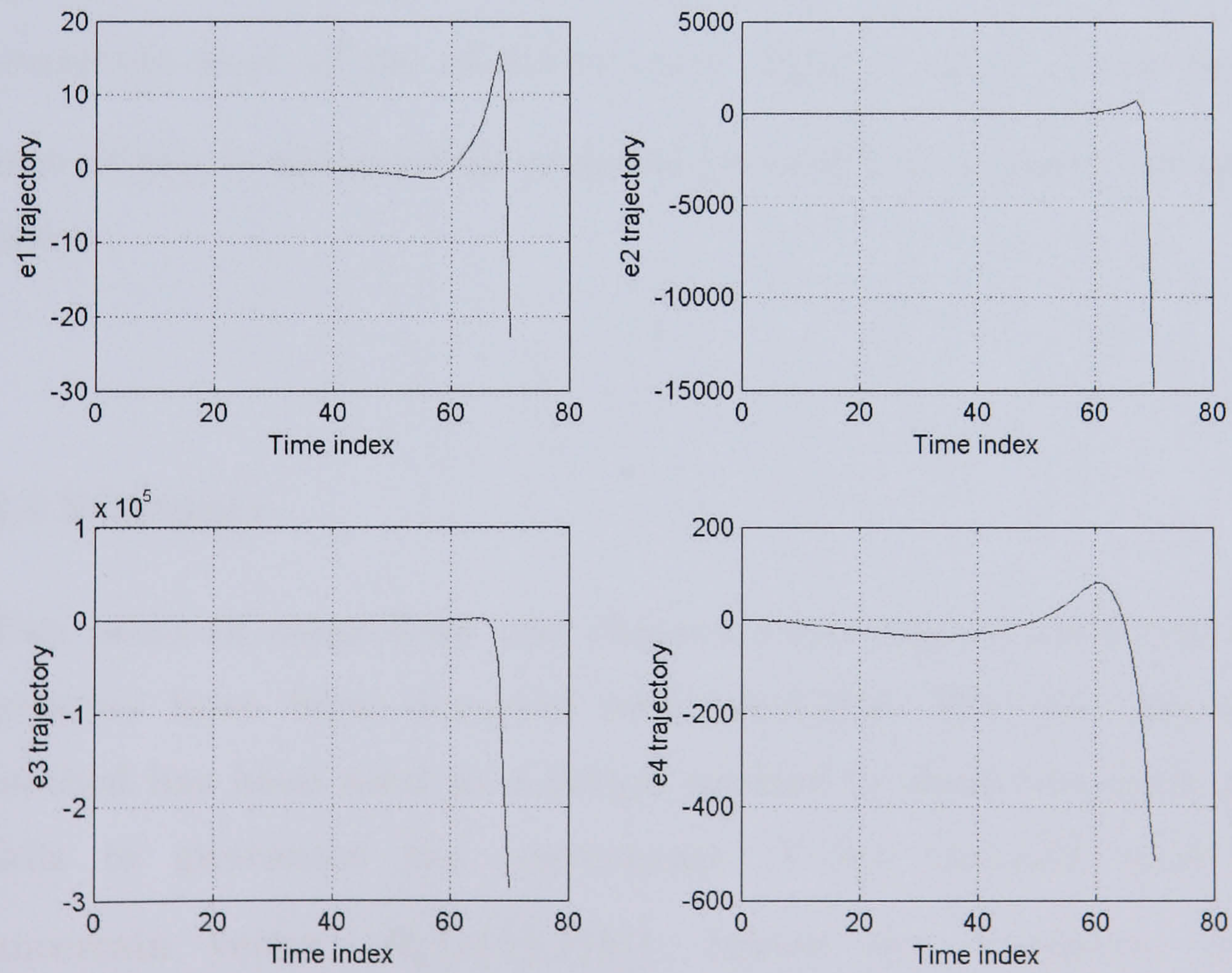

Fig 2.8: Observer error for the non-linear discrete-time ball & beam observer.

The results of the simulations for both cases (1) and (2) show that the control systems are stables, figures (2.2) and (2.6) and the observer errors, figures (2.4) and (2.8) , are diverging. Since the observer error equation (2.13) consist of two parts, a linearized part, $\left[\boldsymbol{A}_{\boldsymbol{z}}(k)-\boldsymbol{l}(k)\boldsymbol{C}\right]\boldsymbol{e}(k)$, where the pole placement method is used to calculate the observer feedback gain $\boldsymbol{l}(k)$. The second part is an uncertain vector $\boldsymbol{R}_k\left(\boldsymbol{e}(k),\boldsymbol{z}(k)\right)$, the Taylor's residual, which is causing the observer error to diverge.

So, the pole placement method has failed in the design of a non-linear discrete-time observer. Therefore, another method has to be found to guarantee the convergence of the observer error i.e. it should force the uncertain part of the observer error $\boldsymbol{R}_k\left(\boldsymbol{e}(k),\boldsymbol{z}(k)\right)$ to converge to zero. If this is achieved the observer error will necessarily converge to zero.

2.4 Summary

Two cases of controllers and observers for discrete-time non-linear systems have been designed and simulated. The pole placement method has been used as a design method in these two cases and it fails to guarantee the convergence of the observer error. The uncertain vector, $\boldsymbol{R}_k\left(\boldsymbol{e}(k),\boldsymbol{z}(k)\right)$, causes the divergence of the observer error. Therefore, another method or technique needs to be found to force the uncertain vector $\boldsymbol{R}_k\left(\boldsymbol{e}(k),\boldsymbol{z}(k)\right)$ to go to zero so that the observer error will converge.

Chapter 3
Sliding Mode Control

3.1 Introduction

In the early 1960's, the Russian scientists Enl'yanove and Barbashin came up with the idea of variable structure control systems (VSCS). These ideas did not appear outside Russia however until U. Utkis published a book in 1976 called Control Systems of Variable Structure [45], and V. Utkin published a survey paper in 1977 [46].

Variable structure control systems are a class of systems where the control law is changing during the control process according to some defined rules that depend on the state of the system. In other words, a variable structure control systems is characterized by a suite of feedback control laws and a switching function. The switching function has as its input some measure of the current system behaviour and produces as output a particular feedback controller which should be used at that instant in time. The result is a variable structure system, which may be considered as a combination of subsystems where each subsystem has a fixed control structure and is valid for regions of system behaviour [47].

Sliding mode control (SMC) is a type of variable structure control (VSC). In sliding mode control, variable structure control systems are designed to drive and then keep the system state to lie within a neighbourhood of the switching function.

Most of the variable structure control and sliding mode control literature consider systems that are both in continuous-time and linear, for example [5], [11], [48], [50], [54], [56], and [57]. Some of the literature does also consider non-linear systems in continuous-time [11], [12], [24], [25], [40], [54], [57], as well as discrete-time linear systems [51], [54].

Most of the design techniques for sliding-mode control assume that all the system states are accessible to the control law. In practice, all of these states are not physically available for feedback. In this case, a full state feedback sliding mode controller cannot be implemented unless an observer is used to estimate the unmeasured states, or the design methods must be modified such that only a subset of the states are required to implement the control law. The output feedback Sliding mode control has been paid many attentions in recent years, Zak *et al.* [58] developed a geometric condition to guarantee the existence of the sliding surface and the stability of the sliding mode. Edwards *et al.* [56] provided a canonical form on which the design problem of sliding mode output feedback control is converted to a static output feedback control problem.

Aitken et. al. [76] presented a discrete-time sliding mode observer for discrete-time linear time-invariant systems. They showed that the discontinuous compensation signal in the discrete-time observer causes a limit cycle around the sliding surface. To prevent such a phenomenon, they proposed the use of a nonswitching compensation signal using the concept of the discrete-time equivalent control.

Jiang *et. al.* [77] design a non-linear adaptive controller for single-input single-output feedback linearisable nonlinear systems. A fictitious state is introduced to represent the system perturbation which includes the combined effect of system nonlinearities, uncertainties and external disturbances. A sliding-mode state and perturbation observer is designed to estimate the system states and the fictitious state. The non-linear adaptive controller has a simple structure as only one output is required to implement it and its design does not require the detailed information of system model but it is able to adapt itself to the system variation and external disturbances. Sliding mode state and perturbation observer can be regarded as an extended-order conventional sliding-mode state observer proposed in [40] However, the accuracy of state estimation in sliding mode state and perturbation observer depends on the estimation error of the perturbation, rather than the upper bounds of the perturbation required in sliding mode state and perturbation observer.

Veluvolu *et.al.* [66] proposed a design method for a discrete-time sliding mode (DSM) nonlinear observer for a class of nonlinear uncertain systems. A strategy is employed to avoid switching across the sliding manifold and the sliding trajectory is confined to a boundary layer once it converges to the sliding manifold. the selection of the sliding mode gain and the boundary layer is based on disturbance bounds.

Choi [59] developed a method for designing sliding mode controllers by presenting a sufficient and necessary condition in terms of linear matrix inequality (LMI) with a matrix equation constraint. Ji *et al.* [60] proposed sufficient and necessary condition for the existence problem is developed by two matrix inequalities, one of which is bilinear matrix inequality. Then an iterative linear matrix inequality (ILMI) approach is presented to solve such kind of matrix inequalities.

Davila *et. al.* [67] proposed a sufficient condition for robust asymptotic stability of the sliding mode dynamics and a sufficient condition is given for the existence of such a sliding mode observer in terms of LMI. The proposed control scheme guarantees the asymptotic stability of the closed-loop system containing observer dynamic and observer error dynamic.

Veluvolu *et. al.* [68] proposed a design method for Sliding mode observer for a class of nonlinear uncertain systems. The original system is divided into three interconnected subsystems, and then multiple sliding modes are then introduced to compensate for multiple disturbance terms in the subsystems by appending them to constant gain observer.

Although sliding mode control can be thought of as a particular approach to robust controller design, its methodology is quite different from other conventional robust control methodologies such as H_2 and H_∞ approaches, where the control law is constructed by minimizing various norms associated with the transfer functions of interest [78]. Traditionally, in sliding mode control the design of a switching surface relies on various methods such as pole placement, eigenstructure assignment, quadratic minimization and so on[78].

As an introduction to the sliding mode control, a linear control system will be discussed in this chapter whilst in chapter (4), the sliding mode technique for non-linear control systems will be discussed.

3.2 Applications of the Sliding Mode Control

Applications that use the theory of variable structure control and the theory of sliding mode control have increased since these theories were published outside Russia.

3.2.1. Motor Control

Control of electrical motors has been a popular application of sliding mode control. The technique has been applied to the control of dc motors, synchronous motors, and induction motors. The following are just a few of many references in the literature [12], [48], [49].

3.2.2. Aircraft Control

Variable structure control has been applied to the a variety of flight problems such as a realization of asymptotically decoupled control of roll angle, angle of attack, and sideslip in the presence of rapid maneuvering [79], and flight control [80], [81]

3.2.3. Spacecraft Control

The sliding mode control has been applied to spacecraft rotation damping [82]] and orientation control [83].

3.2.4. Flexible space Structure Control

Studies have been made on the use of sliding mode control for stabilization, regulation, and maneuvering of large flexible space structures [52], [84].

3.2.5. Other Applications

hydraulic servosystems [50], furnace temperature control [42], [51], helicopter flight regulation [42], robots [7], [8], power system stabilizer [42], [53], chemical process regulation [42] and satellite attitude control [13] are examples of the applications that use variable structure control and sliding mode control theories. Undoubtedly, quite other published works of great interest have been missed.

3.3 Theory of Sliding Mode Control

Sliding mode control can be defined as a control law that switches rapidly between two values of gains with the objective of bringing the system's state trajectory onto a specified surface. In other words, the sliding mode control consists of a control law that switches with infinite speed to drive the system onto a specified state trajectory, which is called the sliding surface, and is then capable of keeping the state on this surface [61]. Figure (3.1) shows the sliding mode in a continuous-time linear system and figure (3.2) shows the oscillation in vicinity of sliding surface in continuous-time.

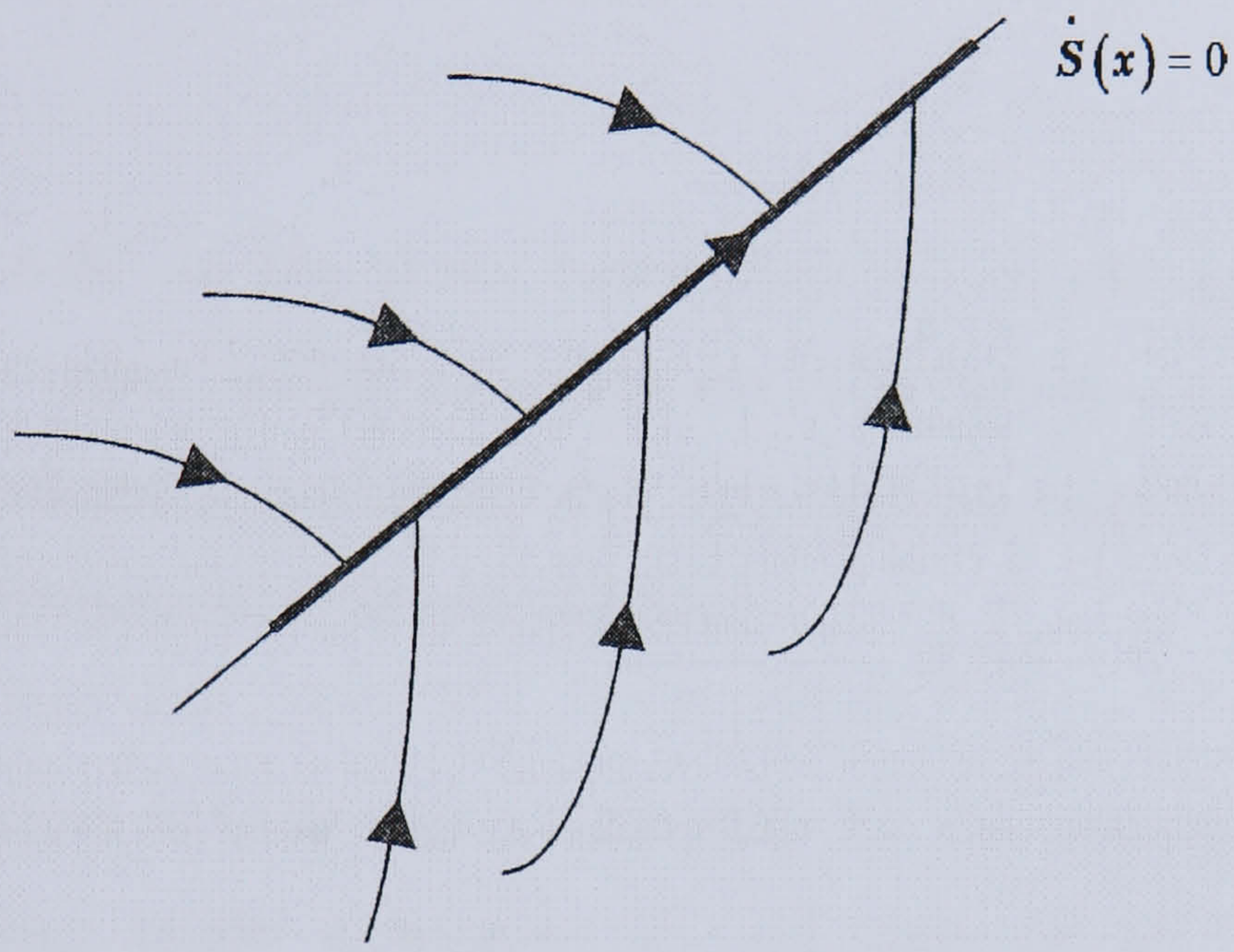

Figure 3.1: Sliding mode in a continuous-time linear system [37].

To explain the sliding mode control method, consider the linear discrete-time control system [54],

$$\boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}\boldsymbol{u}(k)+\boldsymbol{D}\boldsymbol{r}(k) \tag{3.1}$$

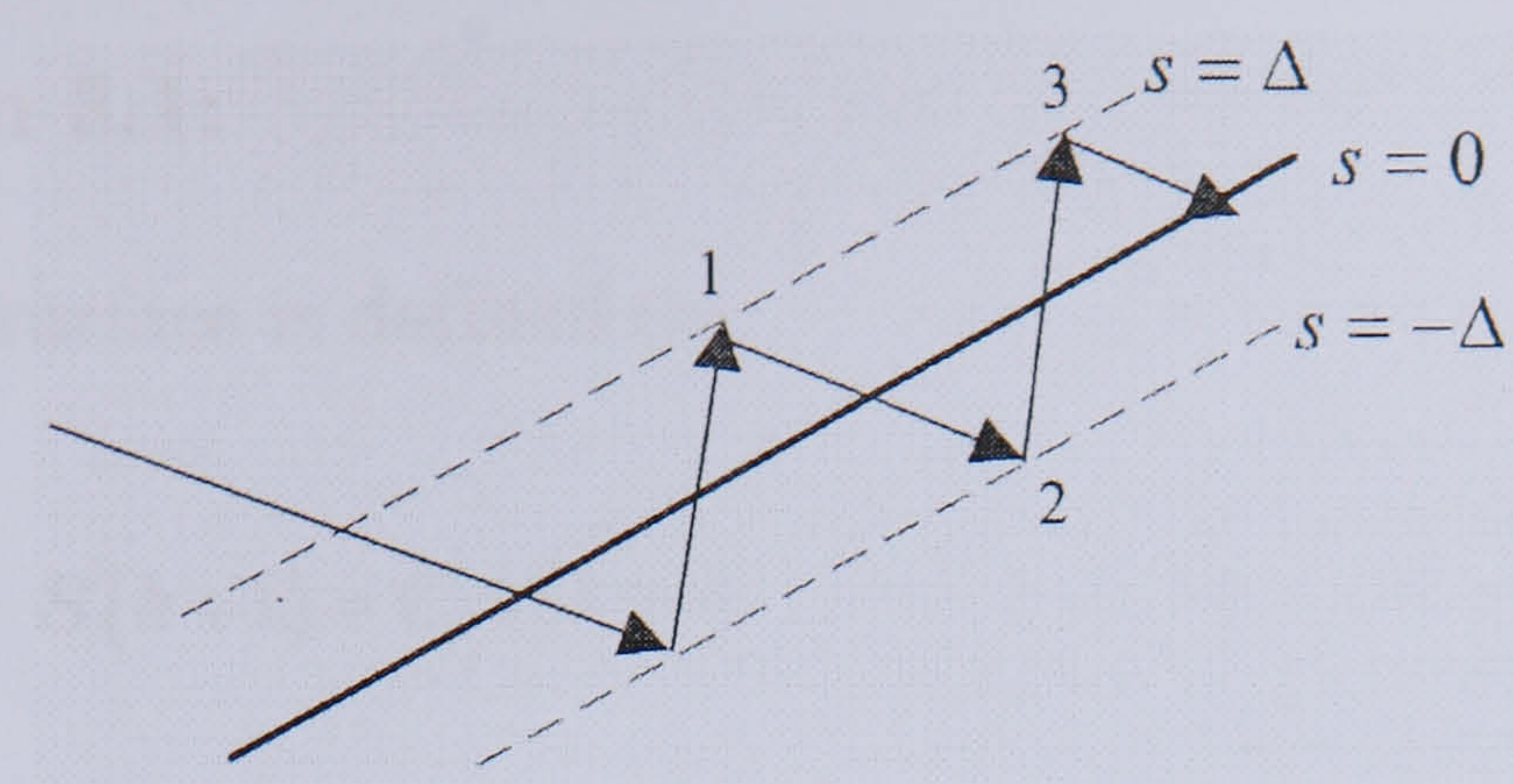

Figure 3.2: Oscillation in vicinity of sliding surface in continuous-time [54].

where $\boldsymbol{x} \in \Re^n$ is the state vector, $\boldsymbol{u} \in \Re^m$ is the input, and $\boldsymbol{r}$ is the reference input which is unknown. The matrices $\boldsymbol{A}$ and $\boldsymbol{D}$ are real valued unknown matrices with appropriate dimensions and the pair ($\boldsymbol{A}$, $\boldsymbol{B}$) is assumed to be controllable.

The objective is to design a controller for the system in equation (3.1). The sliding mode dynamics do not depend on the control input $\boldsymbol{u}(k)$ but depend on the switching (sliding) surface equations. Therefore, the design procedure should consist of two stages. First, the equation of the sliding surface is selected to design the desired dynamics of this motion in accordance with some performance criterion. Then, the discontinuous control should be found such that the state would reach the sliding manifold and such that the sliding mode exists in this manifold. So, the sliding mode design is decoupled into two sub problems of lower dimension and after a finite interval preceding the sliding motion, the system will possess the desired dynamic behaviour [54].

Definition 3.1:

The sliding surface is defined as

$$S(k+1) = C_s x(k+1) \tag{3.2}$$

where $C_s \in \Re^{mxn}$.

The reason for using a sliding surface, sometimes called a switching function, is clear since the function given in equation (3.2) is used to decide which control law is in use at any point.

Definition 3.2:

The sliding manifold where the sliding mode exists is defined as

$$\Omega = \{x(k) | S(k+1) = 0\} \tag{3.3}$$

Definition 3.3:

The dynamics of a discrete-time system are given by the state equation [61],

$$x(k+1) = f(x(k), u(k)) \tag{3.4}$$

$$y(k) = g(x(k), u(k))$$

where $x \in \Re^n, u \in \Re^m$, $y \in \Re^p$, and $m \leq n$. The reference [61] considers a linear system so in fact $f(\cdot)$ and $g(\cdot)$ are linear functions of $x(k)$ and $u(k)$. Discrete-time sliding mode takes place on a subset Σ of the manifold $\Omega = \{x(k) | S(x(k)) = 0\}$, $S \in \Re^m$, if there exists an open neighbourhood $\aleph$ of this subset such that for each $\boldsymbol{x}(k) \in \aleph$ it follows that $\boldsymbol{S}(\boldsymbol{f}(\boldsymbol{x}(k+1))) \in \Sigma$.

According to definition (3.3), the sliding mode existence condition is of the form

$$\boldsymbol{S}(k+1) = \boldsymbol{Cx}(k+1) \tag{3.5}$$

for any $\boldsymbol{x}(k) \in \aleph$.

From equations (3.1) and (3.2)

$$\boldsymbol{S}(k+1) = \boldsymbol{CAx}(k) + \boldsymbol{CDr}(k) + \boldsymbol{CBu}(k) \tag{3.6}$$

the control $\boldsymbol{u}(k)$ is designed by setting equation (3.6) equal to zero.

$$\boldsymbol{S}(k+1) = \boldsymbol{CAx}(k) + \boldsymbol{CDr}(k) + \boldsymbol{CBu}(k) = 0 \tag{3.7}$$

$$\boldsymbol{u}(k) = -(\boldsymbol{CB})^{-1}(\boldsymbol{CAx}(k) + \boldsymbol{CDr}(k)) \tag{3.8}$$

In according to definitions (3.2) and definitions (3.3) with equation (3.5), the sliding mode exists if the matrix $(\boldsymbol{CB})$ has an inverse, otherwise the control $\boldsymbol{u}(k)$ will not be able hold the sliding mode inside the silding manifold.

The control law, which will yield motion in the manifold $\boldsymbol{S}(k+1) = 0$, will be called equivalent control $\boldsymbol{u}_{eq}(k)$ and it can be represented as a sum of two linear functions,

$$\begin{aligned}\boldsymbol{u}_{eq}(k) = -(\boldsymbol{CB})^{-1}\boldsymbol{S}(k) - (\boldsymbol{CB})^{-1}((\boldsymbol{CA} - \boldsymbol{C})\boldsymbol{x}(k) \\ + \boldsymbol{CDr}(k))\end{aligned} \tag{3.9}$$

and

$$S(k+1) = \left(S(k) - Cx(k)\right) + CAx(k) + CDr(k) + CBu(k)$$

(3.10)

$$S(k+1) = S(k) + (CA - C)x(k) + CDr(k) + CBu(k)$$

(3.11)

Definition 3.4:

The control can vary within

$$\|u(k)\| \leq \varphi \tag{3.12}$$

where φ is the boundary layer thickness.

Assumption 3.1:

The available control resources are such that

$$\left\|(CB)^{-1}\right\| \cdot \left\|(CA - C)x(k) + CDr(k)\right\| < \varphi \tag{3.13}$$

otherwise, the control resources are insufficient to stabilize the system.

Then the control $\boldsymbol{u}(k)$,

$$\boldsymbol{u}(k)=\begin{cases}\boldsymbol{u}_{eq}(k) & \text{for } \left\|\boldsymbol{u}_{eq}(k)\right\|\le\varphi \\ \\ \varphi\dfrac{\boldsymbol{u}_{eq}(k)}{\left\|\boldsymbol{u}_{eq}(k)\right\|} & \text{for } \left\|\boldsymbol{u}_{eq}(k)\right\|>\varphi\end{cases} \tag{3.14}$$

Equation (3.14) represents the control law of the sliding mode.

Equation (3.14) shows that $\boldsymbol{u}(k)=\boldsymbol{u}_{eq}(k)$ for $\left\|\boldsymbol{u}_{eq}(k)\right\|\le\varphi$ yields motion in the sliding manifold Ω. But for the case when $\left\|\boldsymbol{u}_{eq}(k)\right\|>\varphi$ i.e. outside the sliding manifold, substitute equation (3.14) into equation (3.11),

$$\begin{aligned}\boldsymbol{S}(k+1)=\boldsymbol{S}(k)&+(\boldsymbol{CA}-\boldsymbol{C})\boldsymbol{x}(k)+\boldsymbol{CDr}(k)\\ &+\boldsymbol{CB}\left(\varphi\frac{\boldsymbol{u}_{eq}(k)}{\left\|\boldsymbol{u}_{eq}(k)\right\|}\right) \qquad \text{for} \qquad \left\|\boldsymbol{u}_{eq}(k)\right\|>\varphi\end{aligned}$$

(3.15)

Substitute equation (3.9) in equation (3.15),

$$\begin{aligned}\boldsymbol{S}(k+1)=\boldsymbol{S}(k)&+(\boldsymbol{CA}-\boldsymbol{C})\boldsymbol{x}(k)+\boldsymbol{CDr}(k)\\ &-\left[\varphi\frac{\left(\boldsymbol{S}(k)+\left((\boldsymbol{CA}-\boldsymbol{C})\boldsymbol{x}(k)+\boldsymbol{CDr}(k)\right)\right)}{\left\|\boldsymbol{u}_{eq}(k)\right\|}\right]\\ &\qquad\qquad \text{for} \qquad \left\|\boldsymbol{u}_{eq}(k)\right\|>\varphi\end{aligned}$$

(3.16)

$$S(k+1) = \left(S(k) + (CA - C)x(k) + CDr(k)\right)\left(1 - \frac{\varphi}{\|u_{eq}(k)\|}\right)$$

$$\text{for} \qquad \|u_{eq}(k)\| > \varphi$$

(3.17)

If the norm of both sides of equation (3.17) is taken,

$$\|S(k+1)\| = \|S(k) + (CA - C)x(k) + CDr(k)\|\left(1 - \frac{\varphi}{\|u_{eq}(k)\|}\right)$$

(3.18)

$$\|S(k+1)\| = \|S(k) + (CA - C)x(k) + CDr(k)\|$$
$$-\left(\frac{\varphi\left(\|S(k) + (CA - C)x(k) + CDr(k)\|\right)}{\|u_{eq}(k)\|}\right)$$

(3.19)

By substituting once again equation (3.9) into equation (3.19) yields,

$$\|S(k+1)\| = \|S(k) + (CA - C)x(k) + CDr(k)\| - \left(\frac{\varphi}{\|(CB)^{-1}\|}\right)$$

(3.20)

$$\|S(k+1)\| \leq \|S(k)\| + \|(CA - C)x(k) + CDr(k)\| - \left(\frac{\varphi}{\|(CB)^{-1}\|}\right)$$

(3.21)

from equation (3.13),

$$\left\|(CA-C)x(k)+CDr(k)\right\|<\frac{\varphi}{\left\|(CB)^{-1}\right\|} \tag{3.22}$$

Then,

$$\left\|S(k+1)\right\|<\left\|S(k)\right\| \tag{3.23}$$

Therefore, $\left\|S(k)\right\|$ will be decreasing and after a finite number of steps $\left\|u_{eq}(k)\right\|<\varphi$ is achieved and the sliding mode will take place after $k>k_{sm}$ [61]. The state trajectory for this system with respect to time is shown in figure (3.3).

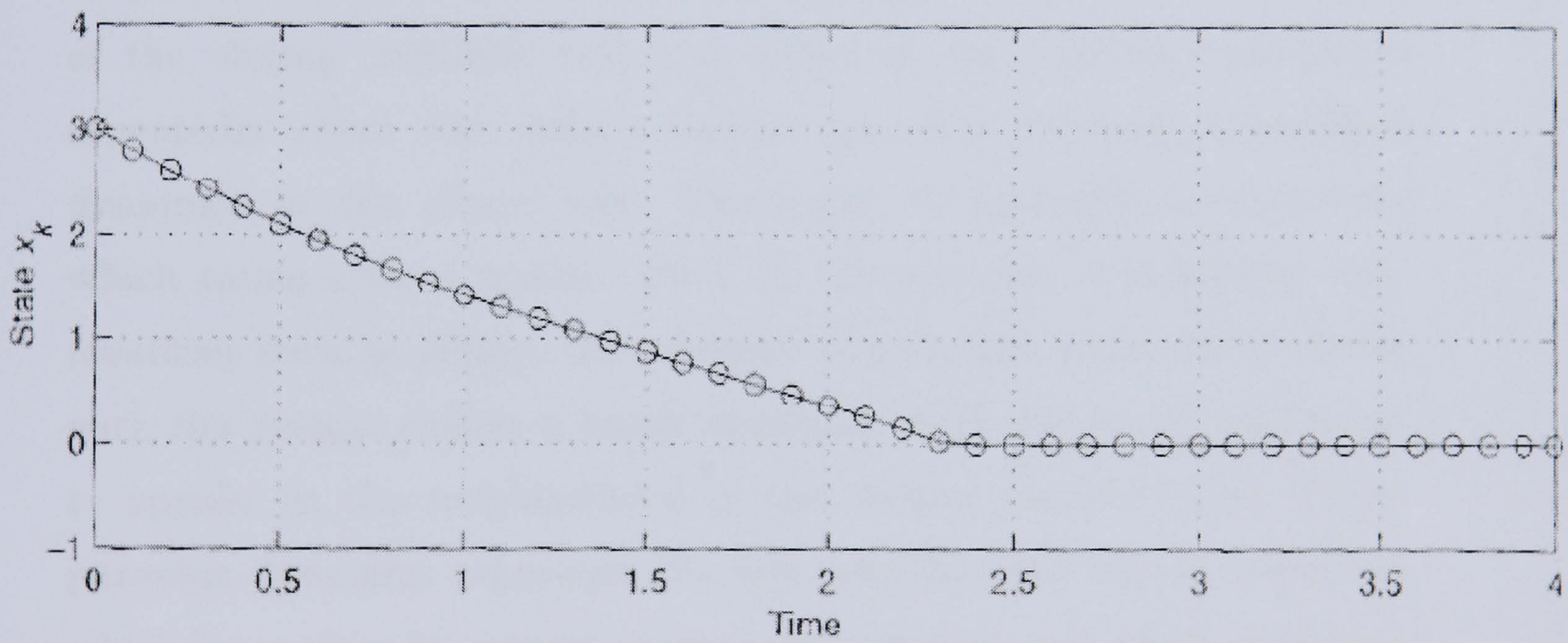

Figure 3.3: Sliding mode control for linear discrete-time system [20].

Equations (3.15) ~ (3.23) show that under assumption (3.1) and by using the control $u(k)$, equation (3.14), the sliding surface will be attractive and once the system reaches the sliding manifold it will keep the motion inside it regardless of the unknown parameters or uncertainties such as the reference input $r(k)$.

To summarize the operation of the sliding mode control,

1. After finite time, the system will reach the sliding manifold Ω.
2. Once the system reaches the sliding mode for all $k > k_{sm}$, its trajectory motion is confined to the sliding manifold Ω and the order of the closed-loop system dynamics is less than the order of the uncontrolled system.

3.4 Chattering

The chattering phenomenon is generally perceived as the high frequency switching between the two different controls at the vicinity of the sliding manifold will take place as the system trajectories repeatedly cross the sliding surface possibly exciting unmodeled dynamics in the closed loop. There are two possible mechanisms which cause such a motion. First, in the absence of switching non-idealities such as delays, the presence of parasitic dynamics in series with the system causes a small amplitude high-frequency oscillation to appear in the neighborhood of the sliding manifold [54]. These parasitic dynamics represent the fast actuator and sensor dynamics which, according to control engineering practice, are often neglected in the open-loop model used for control design if the associated poles are well damped, and outside the desired bandwidth of the feedback control system [54]. Generally, the motion of the real system is close to that of an ideal system in which the parasitic dynamics are neglected, and the difference between the ideal and the real motion, which is on the order of the neglected time constants, decays rapidly. The interactions between the parasitic dynamics and VSC generate a non-decaying oscillatory component of finite amplitude and frequency, and this is generically referred to as chattering.

Second, the switching non-idealities alone can cause such high-frequency oscillations. Since the cause of the resulting chattering phenomenon is due to time delays, discrete-time control design techniques, such as the design of an extrapolator can be applied to mitigate the switching delays [85]. Since it is necessary to compensate for the switching delays by using a discrete-time control design approach, a practical sliding mode control design may have to be unavoidably approached in discrete time.

The piecewise linear approximation of the switching element in a boundary layer of the sliding manifold is an approach to reduce the effects of chattering [86]. Inside the boundary layer, the switching function is approximated by a linear feedback gain. In order for the system behavior to be close to that of the ideal sliding mode, particularly when an unknown disturbance is to be rejected, sufficiently high gain is needed. This proposed method has wide acceptance by many sliding mode researchers, but unfortunately it does not resolve the core problem of the robustness of sliding mode as exhibited in chattering [86]. Most of them used a straightforward approach to avoid chattering: the sign function of the discontinuous control is approximated by the saturation function.

3.5 Advantages and disadvantages of sliding mode control.

3.5.1 Advantages

1. The main feature of sliding mode control is to keep the system insensitive to internal parameter variation such as uncertainties and external disturbance.

2. In sliding mode control systems, the controller structure is changed to obtain the desired behaviour by using a high speed switching feedback control. The sliding mode control system drives the trajectory of the closed-loop system onto a specified surface, which is called the sliding surface, and ensures the trajectory of the closed-loop system stays on this sliding surface there after.

3. One of the most intriguing aspects of sliding mode control is the discontinuous nature of the control action whose primary function is to switch between two distinctively different systems.

3.5.2 Disadvantages

1. After the sliding mode has been achieved at k_{sm}, the system trajectory cannot be backtracked beyond the manifold $\Omega = \{\boldsymbol{x}(k) | \boldsymbol{S}(k+1) = 0\}$ unlike in a system without sliding mode control. In other words, at any point $k \geq k_{sm}$ it is not possible to determine the time k_{sm} or to reverse calculate the trajectory for $k < k_{sm}$ based on information of the systems state at $k = k_0$ [27].

2. Chattering is a major problem in continuous-time sliding mode control design [21], [27] and without proper treatment; the chattering may be a major obstacle to the implementation of sliding in a wide range of applications.

3.6 Summary

Sliding mode control for linear discrete-time systems has been discussed. It shows that by designing the sliding surface and the control u_k, the state trajectory is forced to the sliding manifold, i.e. the desired trajectory. In addition, after the state trajectory reaches this manifold, it will stay there. The non-linear discrete-time sliding mode will be discussed in chapter (4).

Chapter 4
Non-linear Discrete-time Observer Design by Sliding Mode

4.1 Introduction

In chapter 2, observers for non-linear discrete-time systems are designed using a pole placement method. The observer errors for these systems do not converge to zero. In addition, a linearization method has been used to try to force the observer error to converge but it does not work. Therefore, there is a need to find another method to guarantee the observer errors converge to zero. The previous chapter described the sliding mode technique for linear discrete-time controller. It appeared to be a method that could be used successfully to force the observer error to converge. So, this method with some modification will be applied in this chapter to the non-linear discrete-time observers in order to force the observer error to converge.

4.2 Using Sliding Mode in Non-linear Discrete-time Observer Design

Similar to section 2.2, consider the single input single output (SISO) non-linear discrete-time state space system as before,

$$\boldsymbol{x}(k+1)=\boldsymbol{f}(\boldsymbol{x}(k))+\boldsymbol{B}u(k) \tag{4.1}$$

$$y(k)=\boldsymbol{C}\boldsymbol{x}(k) \tag{4.2}$$

where the pair $\left(\boldsymbol{A}_x, \boldsymbol{B}\right)$ are controllable, the pair $\left(\boldsymbol{A}_x, \boldsymbol{C}\right)$ are observable and $\boldsymbol{A}_x$ is the Jacobian matrix of $f(x)$ around the equilibrium point $\left(\boldsymbol{x}_0, \boldsymbol{u}_0\right)$.

The non-linear discrete-time observer of the system (4.1-2) will be designed to estimate the states of the system. This non-linear observer will be in the form,

$$\boldsymbol{z}(k+1) = \boldsymbol{f}\left(\boldsymbol{z}(k)\right) + \boldsymbol{B}u(k) + \boldsymbol{l}(k)\left(y(k) - \boldsymbol{C}\boldsymbol{z}(k)\right) \tag{4.3}$$

where $\boldsymbol{z}(k)$ is the estimated state vector and $\boldsymbol{l}(k)$ is the observer feedback gain.

The observer error is defined as,

$$\boldsymbol{e}(k) = \boldsymbol{x}(k) - \boldsymbol{z}(k) \tag{4.4}$$

and

$$\boldsymbol{e}(k+1) = \left[\boldsymbol{A}_z(k) - \boldsymbol{l}(k)\boldsymbol{C}\right]\boldsymbol{e}(k) + \boldsymbol{R}_k\left(\boldsymbol{e}(k), \boldsymbol{z}(k)\right) \tag{4.5}$$

where $\boldsymbol{R}_k\left(\boldsymbol{e}(k), \boldsymbol{z}(k)\right)$ is an uncertain vector.

From chapter 3, the design of non-linear discrete-time observers using the pole placement method or linearizing method does not guarantee the convergence of the observer errors. From section 3.3, this divergence is seen to be caused by the uncertain vector $\boldsymbol{R}_k\left(\boldsymbol{e}(k), \boldsymbol{z}(k)\right)$. To solve this problem, another technique has to be used to guarantee the convergence of the observer errors. This chapter investigates the use of the sliding mode technique to solve the problem. Figure (4.1) shows the block diagram for the suggested non-linear discrete-time sliding mode observer.

Let the observer equation (4.3) be modified by adding an auxiliary observer control $v(k)$ to the observer,

$$\boldsymbol{z}(k+1) = \boldsymbol{f}(\boldsymbol{z}(k)) + \boldsymbol{B}u(k) + \boldsymbol{l}(k)(y(k) - \boldsymbol{C}\boldsymbol{z}(k)) + \boldsymbol{D}v(k) \tag{4.6}$$

where $v(k)$ is a scalar and $\boldsymbol{D} \in \Re^{n\times 1}$ is the auxiliary vector.

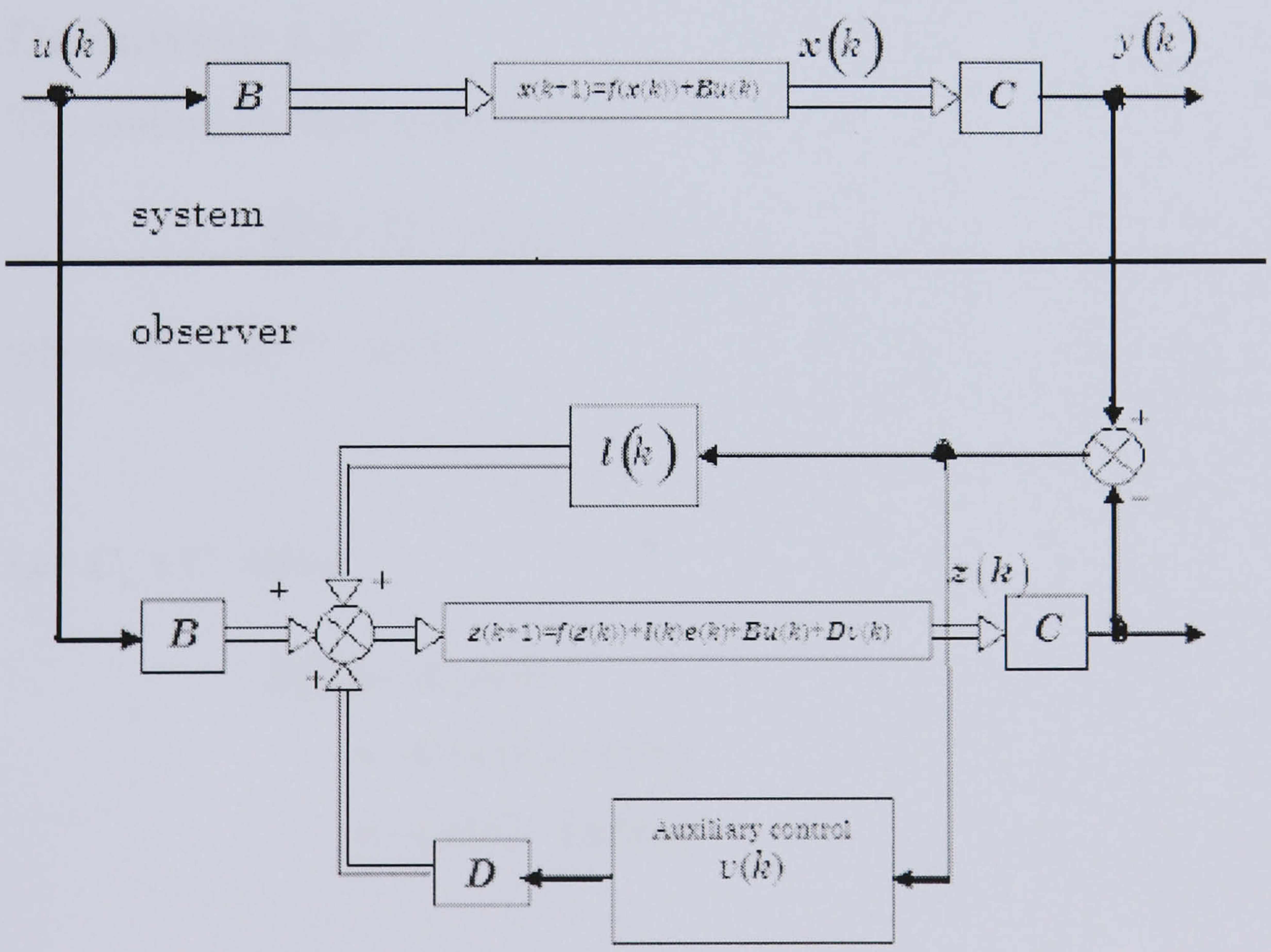

Figure 4.1: Block diagram for the non-linear discrete-time sliding mode observer.

Figure (4.1) shows the block diagram of for the non-linear discrete-time sliding mode observer presented in this research. The error difference between the output of the control system and observer is fed back to the observer and also it been used to calculate the auxiliary control *υ(k)* which is fed back to the observer.

Then the observer error,

$$\boldsymbol{e}(k+1)=\left[\boldsymbol{A}_{\boldsymbol{z}}(k)-\boldsymbol{l}(k)\boldsymbol{C}\right]\boldsymbol{e}(k)+\boldsymbol{R}_k\left(\boldsymbol{e}(k),\boldsymbol{z}(k)\right)-\boldsymbol{D}\upsilon(k) \tag{4.7}$$

To perform the sliding mode design, a sliding surface and a sliding manifold where the sliding mode will occur have to be defined.

Definition 4.1:

The sliding surface is defined as,

$$\boldsymbol{S}(k+1)=-\boldsymbol{C}_s\boldsymbol{e}(k+1) \tag{4.8}$$

where $\boldsymbol{C}_s \in \Re^{mxn}$, m=1.

Let $\boldsymbol{C}_s=\boldsymbol{C}$, then,

$$\begin{aligned}\boldsymbol{S}(k)&=-\boldsymbol{C}_s\boldsymbol{e}(k)\\&=-\boldsymbol{C}\left(\boldsymbol{x}(k)-\boldsymbol{z}(k)\right)\\&=-\left(y(k)-\boldsymbol{C}\boldsymbol{z}(k)\right)\end{aligned} \tag{4.9}$$

Definition 4.2:

The sliding manifold is defined as,

$$\Omega=\left\{\boldsymbol{e}(k)\middle|\boldsymbol{S}(k+1)=0\right\} \tag{4.10}$$

Then,

$$\boldsymbol{S}(k+1) = -\boldsymbol{Ce}(k+1) = 0 \tag{4.11}$$

Substituting equation (4.7) into equation (4.11) yields,

$$\boldsymbol{S}(k+1) = -\boldsymbol{C}\left\{\left[\boldsymbol{A}_z(k) - \boldsymbol{l}(k)\boldsymbol{C}\right]\boldsymbol{e}(k) + \boldsymbol{R}_k - \boldsymbol{D}v(k)\right\} = 0$$

$$= -\boldsymbol{C}\left[\boldsymbol{A}_z(k) - \boldsymbol{l}(k)\boldsymbol{C}\right]\boldsymbol{e}(k) - \boldsymbol{CR}_k + \boldsymbol{CD}v(k) = 0 \tag{4.12}$$

where $\boldsymbol{R}_k = \boldsymbol{R}_k\left(\boldsymbol{e}(k), \boldsymbol{z}(k)\right)$

In according to definitions (4.1) and definitions (4.2) with equation (4.7), the sliding mode exists if the matrix $(\boldsymbol{CD})$ has an inverse, otherwise the control $\boldsymbol{u}(k)$ will not be able hold the sliding mode inside the sliding manifold.

Let

$$\boldsymbol{A}_s(k) = [\boldsymbol{A}_z(k) - \boldsymbol{l}(k)\boldsymbol{C}] \tag{4.13}$$

The control law, which will yield motion in the manifold $\boldsymbol{S}(k+1) = 0$, will be called equivalent control $\boldsymbol{v}_{eq}(k)$ and it can be represented as a sum of two functions.

Then,

$$v_{eq}(k) = (\boldsymbol{CD})^{-1}\boldsymbol{CA}_s(k)\boldsymbol{e}(k) + (\boldsymbol{CD})^{-1}\boldsymbol{CR}_k \tag{4.14}$$

where $v_{eq}(k)$ is the equivalent auxiliary observer control and $\boldsymbol{CD} \neq 0$.

At this stage, $\boldsymbol{e}(k)$ is unknown because the system states, $\boldsymbol{x}(k)$, are unknown. $\boldsymbol{e}(k)$ will not be used in the simulations nor in practice to calculate $v(k)$ but it is only used here to develop the theoretical part of the design.

Rewriting equations (4.12) and (4.14) to become a sum of two functions, i.e. by adding

$$(\boldsymbol{CD})^{-1}\boldsymbol{S}(k)-(\boldsymbol{CD})^{-1}\boldsymbol{S}(k)=0 \tag{4.15}$$

to the right hand sides of both equations.

$$\begin{aligned} \boldsymbol{v}_{eq}(k) &= (\boldsymbol{CD})^{-1}\boldsymbol{S}(k)-(\boldsymbol{CD})^{-1}\boldsymbol{S}(k)+(\boldsymbol{CD})^{-1}\boldsymbol{A}_s(k)\boldsymbol{e}(k) \\ &\quad +(\boldsymbol{CD})^{-1}\boldsymbol{CR}_k \end{aligned} \tag{4.16}$$

from equation (4.9), $\boldsymbol{S}(k)=-\boldsymbol{C}_s\boldsymbol{e}(k)$. Then,

$$\begin{aligned} \boldsymbol{v}_{eq}(k) &= (\boldsymbol{CD})^{-1}\boldsymbol{S}(k)+(\boldsymbol{CD})^{-1}\boldsymbol{C}\left[\boldsymbol{A}_s(k)+\boldsymbol{I}\right]\boldsymbol{e}(k) \\ &\quad +(\boldsymbol{CD})^{-1}\boldsymbol{CR}_k \end{aligned} \tag{4.17}$$

The reason of deriving $\boldsymbol{v}_{eq}(k)$ is to find out the limitation of $\boldsymbol{v}(k)$.

and similar to equation (4.12),

$$\boldsymbol{S}(k+1)=-\boldsymbol{S}(k)-\boldsymbol{C}\left[\boldsymbol{A}_s(k)+\boldsymbol{I}\right]\boldsymbol{e}(k)-\boldsymbol{CR}_k+\boldsymbol{CDv}(k) \tag{4.18}$$

Definition 4.3: (Boundary layer)

The auxiliary observer control can vary within,

$$v(k)\leq\varphi \tag{4.19}$$

Since the non-linear observer error equation (4.5) consists of $\left[A_z(k)-l(k)C\right]e(k)$, a linearized part, which is similar to the linear case and the observer feedback gain, $l(k)$, can be calculated using pole placement method at assigned observer closed-loop poles at each time step (k). The Taylor's residual R_k, which is the second part of the observer error equation (4.5), is an uncertain vector that may cause the divergence of the observer error.

From definition (2.1) and since the continuous function is bounded when it is evaluated over a compact set, then the function $f(\cdot)$ is bounded whenever $x(k), z(k) \in \mathbf{K}$. Therefore the Taylor residual R_k, which cause divergence of the observer error, is bounded

Assumption 4.1:

For bounded uncertain vector R_k, the available observer resources are such that

$$\left\|(CD)^{-1}\right\| \cdot \left\|C\left[A_s(k)+I\right]e(k)+CR_k\right\| < \varphi \tag{4.20}$$

otherwise, the observer resources are insufficient to stabilize the system.

Definition 4.4:

The Lyapunov function is

$$V(k)=\left\|S(k)\right\| \tag{4.21}$$

Theorem 4.1:

For the non-linear discrete-time observer (4.6), the auxiliary control $v(k)$ under assumption (4.1) guarantees the achievement of a stable discrete sliding motion on the sliding manifold $S(k+1)=0$, and hence of the convergence of the observer error if $v(k)$ is chosen as:

$$v(k)=\begin{cases}(\boldsymbol{CD})^{-1}S(k) & \left\|(\boldsymbol{CD})^{-1}S(k)\right\|\le\varphi \\ \varphi\dfrac{(\boldsymbol{CD})^{-1}S(k)}{\left\|(\boldsymbol{CD})^{-1}S(k)\right\|} & \left\|(\boldsymbol{CD})^{-1}S(k)\right\|>\varphi\end{cases} \tag{4.22}$$

Proof:

From definition (4.4), the sliding manifold Ω is attractive if

$$\boldsymbol{V}(k+1)<\boldsymbol{V}(k) \tag{4.23}$$

$$\Rightarrow \quad \left\|S(k+1)\right\|<\left\|S(k)\right\| \qquad \forall\ k\ge 0 \tag{4.24}$$

Substitute equation (4.21) into equation (4.18),

$$S(k+1)=-S(k)-\boldsymbol{C}\left[\boldsymbol{A}_s(k)+\boldsymbol{I}\right]\boldsymbol{e}(k)-\boldsymbol{CR}_k+\varphi\frac{S(k)}{\left\|(\boldsymbol{CD})^{-1}S(k)\right\|}$$

$$\text{for}\quad \left\|(\boldsymbol{CD})^{-1}S(\boldsymbol{k})\right\|>\varphi$$

(4.25)

$$S(k+1) = S(k)\left(\frac{\varphi}{\left\|(\boldsymbol{CD})^{-1} S(k)\right\|} - 1\right) - \boldsymbol{C}\left[\boldsymbol{A}_s(k) + \boldsymbol{I}\right]\boldsymbol{e}(k) - \boldsymbol{CR}_k$$

$$\text{for} \quad \left\|(\boldsymbol{CD})^{-1} S(k)\right\| > \varphi \tag{4.26}$$

If the norm of both sides of equation (4.26) is taken

$$\left\|S(k+1)\right\| \leq \left\|S(k)\right\| \cdot \left\|\left(1 - \frac{\varphi}{\left\|(\boldsymbol{CD})^{-1} S(k)\right\|}\right)\right\| + \left\|\boldsymbol{C}\left[\boldsymbol{A}_s(k) + \boldsymbol{I}\right]\boldsymbol{e}(k) + \boldsymbol{CR}_k\right\| \tag{4.27}$$

$$\left\|S(k+1)\right\| \leq \left\|S(k)\right\| - \frac{\varphi\left\|S(k)\right\|}{\left\|(\boldsymbol{CD})^{-1} S(k)\right\|} + \left\|\boldsymbol{C}\left[\boldsymbol{A}_s(k) + \boldsymbol{I}\right]\boldsymbol{e}(k) + \boldsymbol{CR}_k\right\| \tag{4.28}$$

$$\left\|S(k+1)\right\| \leq \left\|S(k)\right\| - \frac{\varphi}{\left\|(\boldsymbol{CD})^{-1}\right\|} + \left\|\boldsymbol{C}\left[\boldsymbol{A}_s(k) + \boldsymbol{I}\right]\boldsymbol{e}(k) + \boldsymbol{CR}_k\right\| \tag{4.29}$$

Under assumption (4.1)

$$\left\|\boldsymbol{C}\left[\boldsymbol{A}_s(k) + \boldsymbol{I}\right]\boldsymbol{e}(k) + \boldsymbol{CR}_k\right\| < \frac{\varphi}{\left\|(\boldsymbol{CD})^{-1}\right\|} \tag{4.30}$$

then,

$$\left\|S(k+1)\right\| < \left\|S(k)\right\| \tag{4.31}$$

Equation (4.31) satisfies the Lyapunov function in definition (4.4). Therefore $\|S(k)\|$ decreases monotonically and after a number of steps, the observer error will converge to zero. □

Theorem (4.1) shows that under the assumption (4.1) and by choosing the right boundary layer φ, the sliding manifold will attract the observer error to the sliding surface and keep it inside the sliding manifold which will force the observer error to converge to zero. Therefore, regardless of uncertainties such as $\boldsymbol{R}_k(\boldsymbol{e}(k),\boldsymbol{z}(k))$ or disturbances to the system, Theorem (4.1) guarantees the convergence of the observer error.

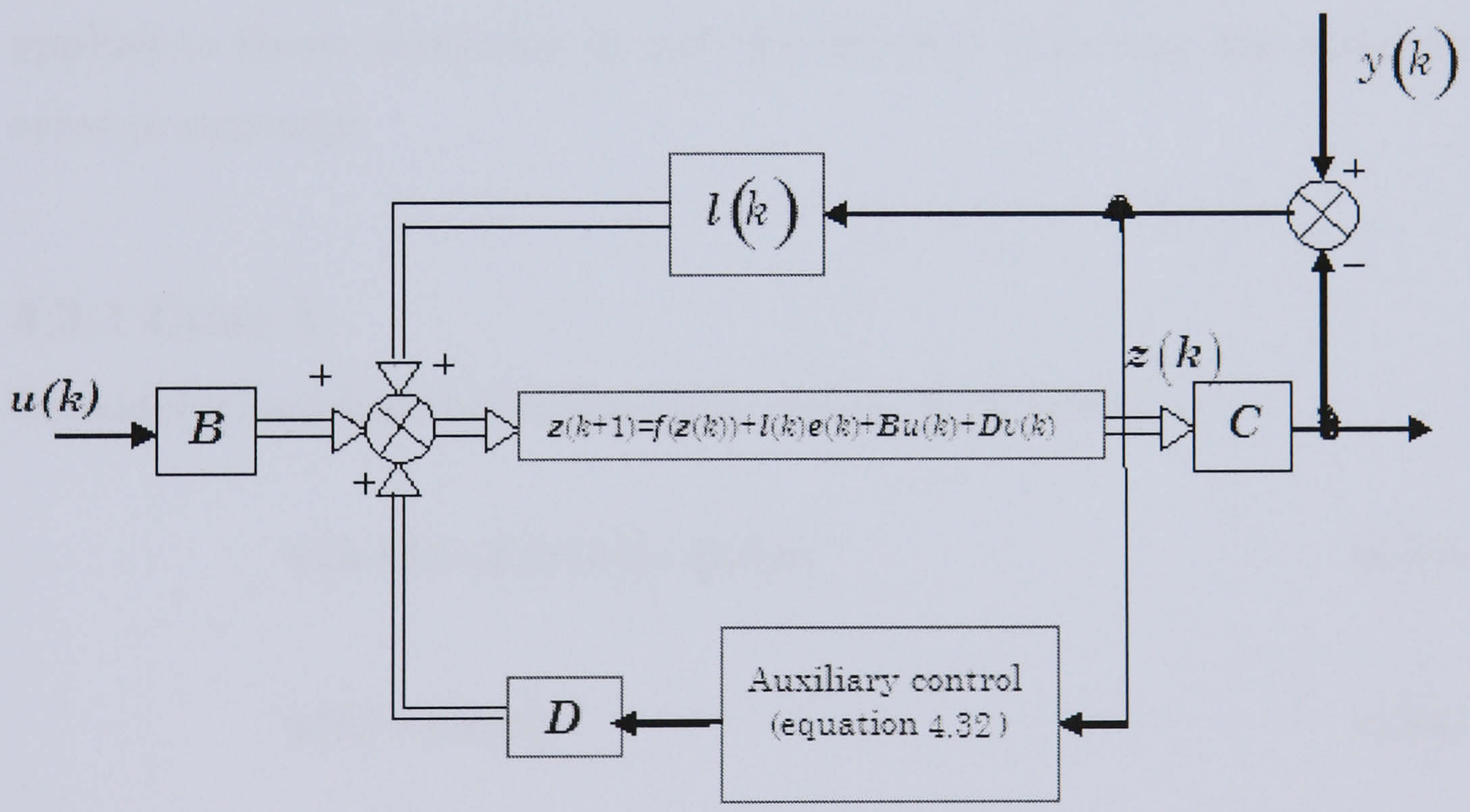

Figure 4.2: Block diagram of the non-linear discrete-time sliding mode observer.

Figure (4.2) shows the block diagram of the designed non-linear discrete-time sliding mode observer. The auxiliary control $\upsilon(k)$ is chosen according to theorem (4.1) and equation (4.9) to be,

$$\upsilon(k)=\begin{cases}-(\boldsymbol{CD})^{-1}\left(y(k)-\boldsymbol{Cz}(k)\right) & \left\|(\boldsymbol{CD})^{-1}\left(y(k)-\boldsymbol{Cz}(k)\right)\right\|\le\varphi \\ \operatorname{sgn}\left(-(\boldsymbol{CD})^{-1}\left(y(k)-\boldsymbol{Cz}(k)\right)\right)\varphi & \left\|(\boldsymbol{CD})^{-1}\left(y(k)-\boldsymbol{Cz}(k)\right)\right\|>\varphi\end{cases} \tag{4.32}$$

4.3 Cases studies

In chapter 3, two cases of non-linear discrete-time observers are studied. Both of these observers have divergent observer errors. In this section, the sliding mode method presented in section 4.2 will be applied to these observers to test this method of forcing the observer error to converge.

4.3.1 Case 1

Consider a non-linear discrete-time system (2.3) and (2.4),

$$\boldsymbol{x}(k+1)=\boldsymbol{f}\left(\boldsymbol{x}(k)\right)+\boldsymbol{B}u(k) \tag{4.33}$$

$$y(k)=\boldsymbol{Cx}(k) \tag{4.34}$$

where (as before),

$$\boldsymbol{f}\left(\boldsymbol{x}(k)\right)=\begin{bmatrix}0.85x_1(k)+0.5x_2(k)\\ 0.1x_1^2(k)+0.3x_2(k)\end{bmatrix}, \tag{4.35}$$

$$\boldsymbol{B}=\begin{bmatrix}0\\1\end{bmatrix} \tag{4.36}$$

and

$$C = [\ 0.5 \ \ -2.2\ ] \tag{4.37}$$

Let the sliding mode observer of the system in equation (4.33) be described as

$$\boldsymbol{z}(k+1) = \boldsymbol{f}(\boldsymbol{z}(k)) + \boldsymbol{B}u(k) + \boldsymbol{l}(k)(y(k) - \boldsymbol{Cz}(k)) + \boldsymbol{D}v(k) \tag{4.38}$$

where $\boldsymbol{D} \in \Re^{nxm}$ is the auxiliary matrix and $v(k)$ is the auxiliary control.

$$\boldsymbol{A}_z(k) = \begin{bmatrix} 0.85 & 0.5 \\ 0.2z_{1}(k) & 0.3 \end{bmatrix} \tag{4.39}$$

The non-linear discrete-time sliding mode observer (equation 4.35) is simulated using Matlab with an input control selected as,

$$u(k) = 0.05 \qquad \text{for } k \geq 0, \tag{4.40}$$

The auxiliary vector and the boundary layer are chosen using trial and error method to be,

$$\boldsymbol{D} = \begin{bmatrix} -0.7 \\ 0.275 \end{bmatrix} \tag{4.41}$$

and,

$$\varphi = 10 \tag{4.42}$$

The initial conditions are selected as,

$$\boldsymbol{x}(0)=\begin{bmatrix}0.9\\0.7\end{bmatrix}, \tag{4.43}$$

and

$$\boldsymbol{z}(0)=\begin{bmatrix}5\\4\end{bmatrix} \tag{4.44}$$

Figure (4.3) shows the sliding mode observer trajectory while the sliding mode observer error is shown in figure (4.4). This figure shows that the observer error is converging even though it is not converging to zero. The auxiliary control for the sliding mode observer is shown in figure (4.5).

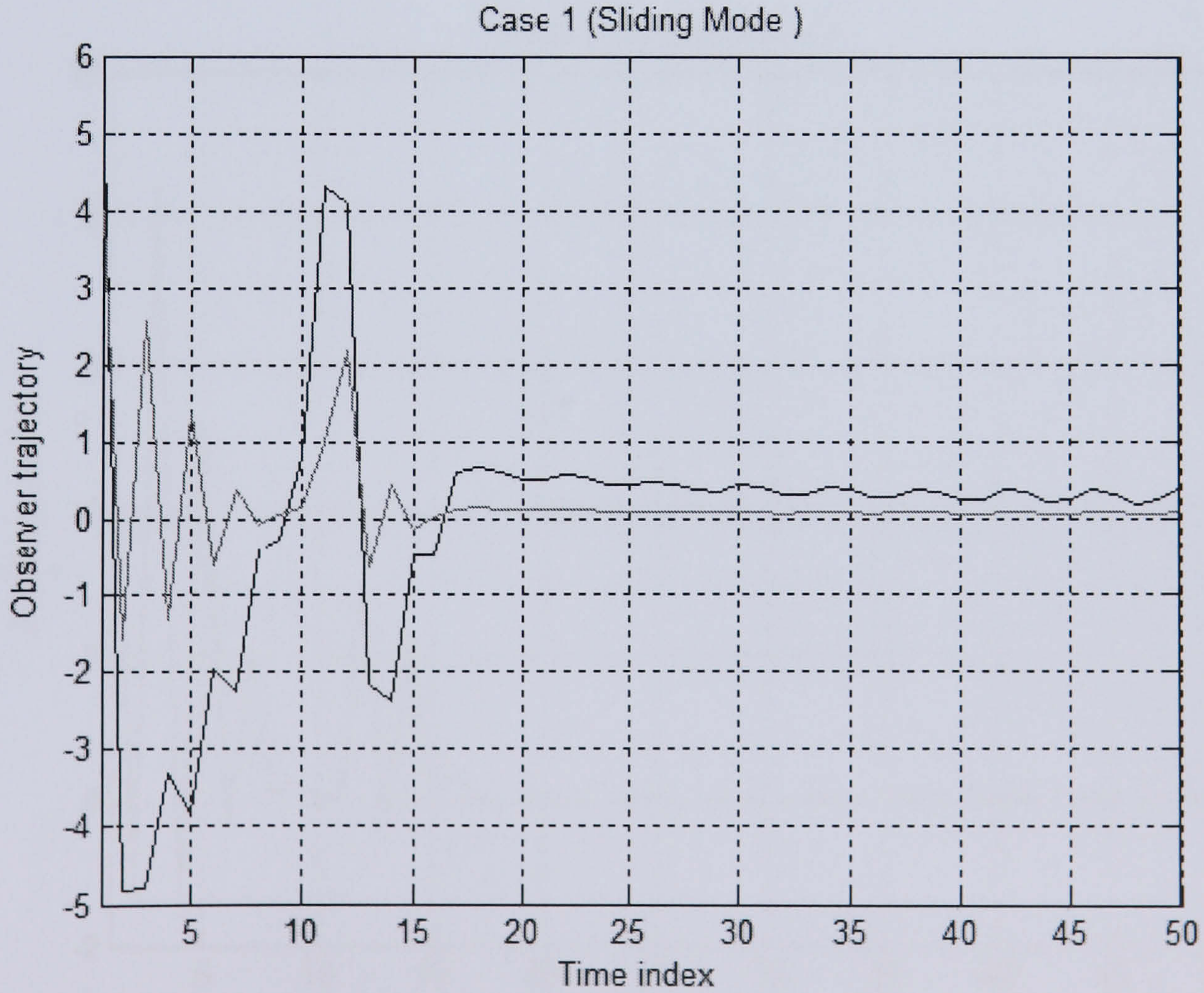

Figure 4.3: Case 1 observer trajectories.

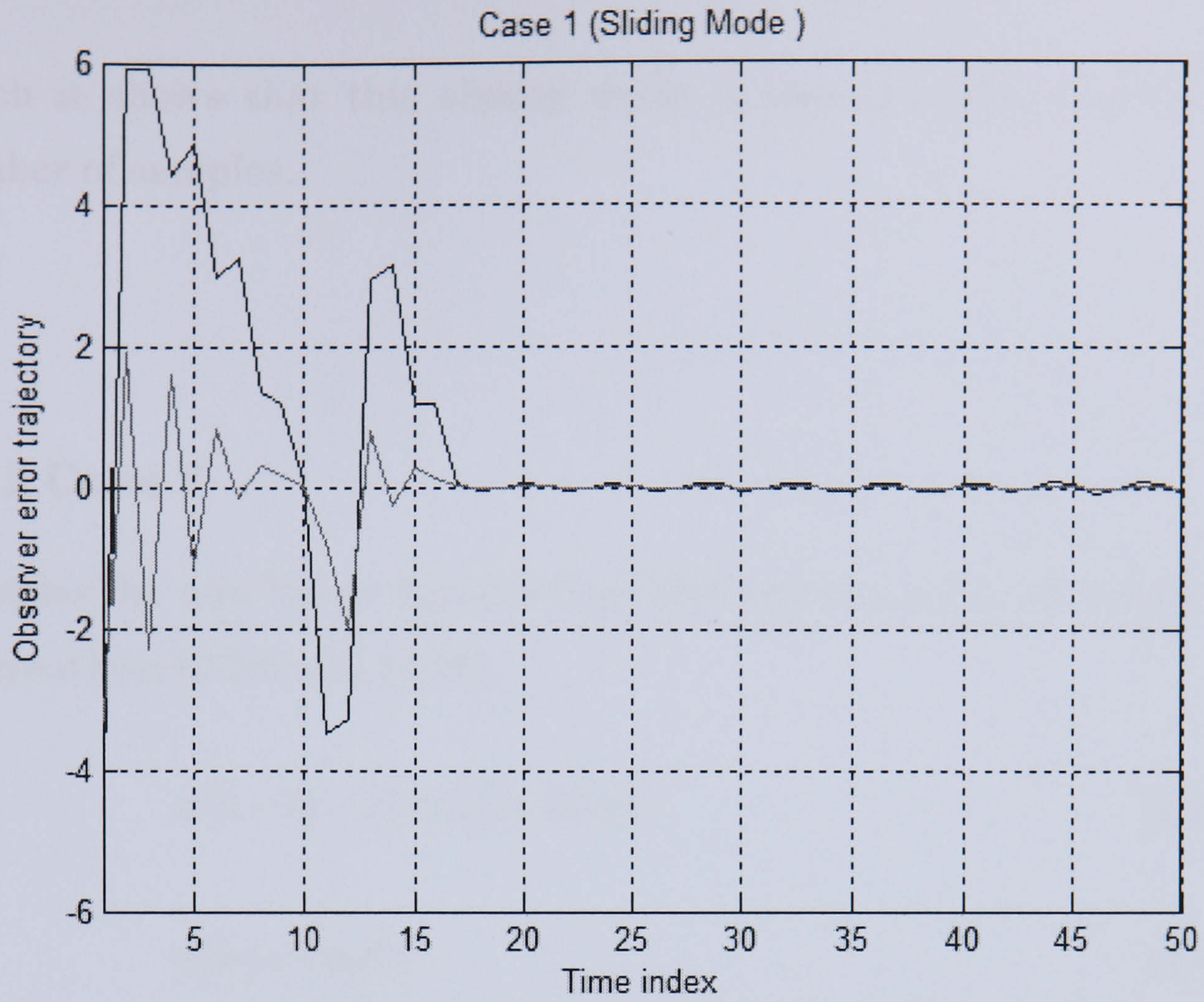

Figure 4.4: Case 1 sliding mode observer error.

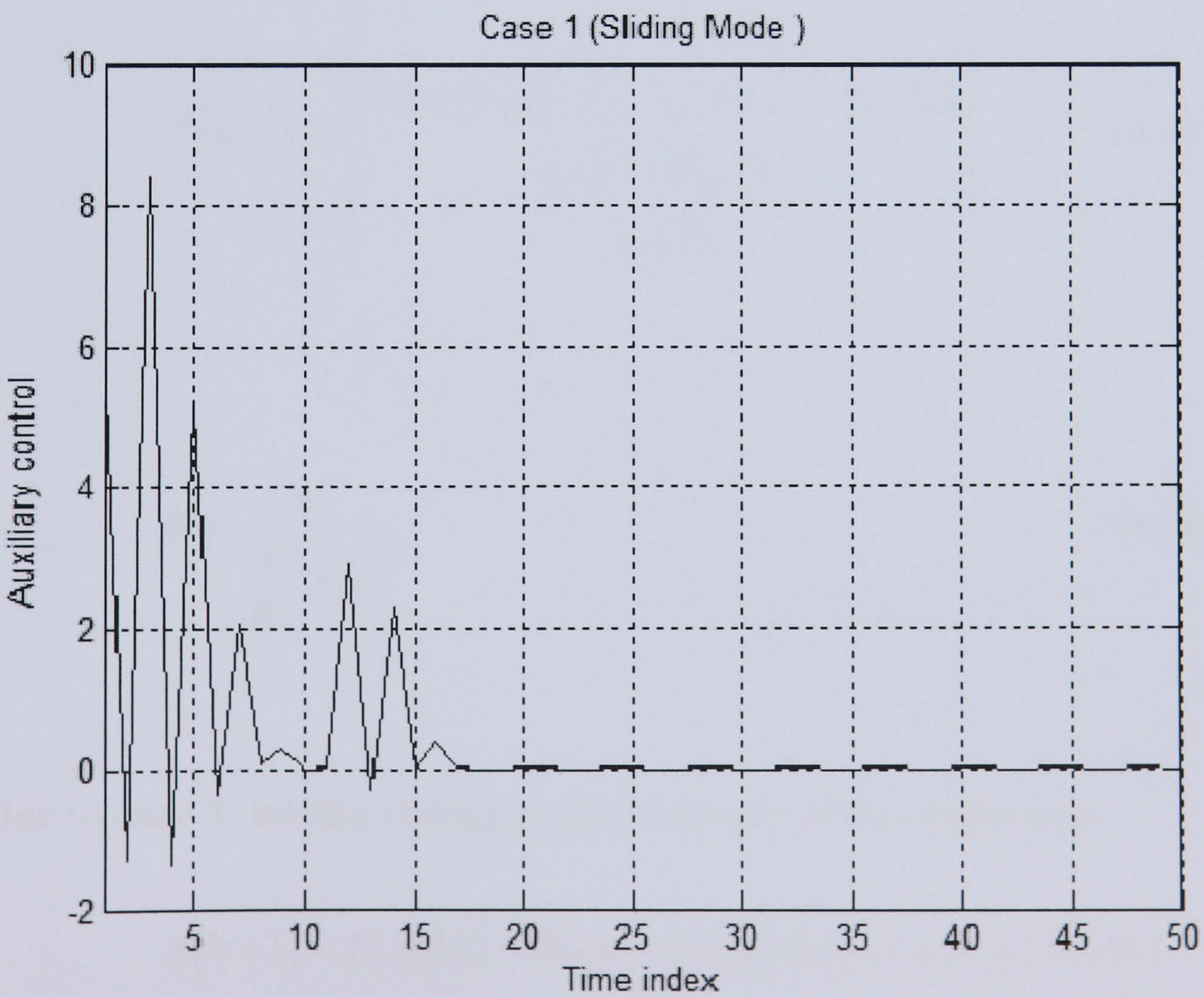

Figure 4.5: Case 1 auxiliary observer control.

which it shows that this sliding mode is free of chattering after a number of samples.

4.3.2 Case 2

Consider the non-linear discrete-time ball and beam system described by equations (2.29) and (2.30),

$$\boldsymbol{x}(k+1)=\boldsymbol{f}(\boldsymbol{x}(k))+\boldsymbol{B}u(k) \tag{4.45}$$

$$y(k)=\boldsymbol{C}\boldsymbol{x}(k) \tag{4.46}$$

where

$$\boldsymbol{f}(\boldsymbol{x}(k))=\begin{bmatrix} x_1(k)+Tx_2(k) \\ hTx_1(k)x_4^2(k)+x_2(k)-ghTx_3(k) \\ x_3(k)+Tx_4(k) \\ x_4(k) \end{bmatrix} \tag{4.47}$$

and,

$$h=\frac{m}{\left(\frac{J}{R^2}+m\right)}, \tag{4.48}$$

Similar to case 1, let the sliding mode observer of the system be,

$$\boldsymbol{z}(k+1)=\boldsymbol{f}(\boldsymbol{z}(k))+\boldsymbol{B}u(k)+\boldsymbol{l}(k)(y(k)-\boldsymbol{C}\boldsymbol{z}(k))+\boldsymbol{D}v(k) \tag{4.49}$$

where $\boldsymbol{D} \in \Re^{n \mathrm{x} \mathrm{m}}$ is the auxiliary matrix, m = 1, and $v(k)$ is the auxiliary control.

$$\boldsymbol{A}_z(k) = \begin{bmatrix} 1 & T & 0 & 0 \\ hTz_4^2(k) & 1 & -ghT & 2hTz_1(k)z_4(k) \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & T \end{bmatrix} \tag{4.50}$$

The non-linear discrete-time sliding mode observer (equation 4.49) is simulated using Matlab with,

$$\boldsymbol{C} = [1\ 0\ 0\ 0], \tag{4.51}$$

and the sampling time,

$$\mathrm{T} = 0.01 \tag{4.52}$$

Similar to case 1, the auxiliary vector and the boundary layer are chosen using trial and error method to be,

$$\boldsymbol{D} = \begin{bmatrix} -0.0005 \\ 0.72 \\ 1.23 \\ 36 \end{bmatrix} \qquad 4.53)$$

$$\boldsymbol{\varphi} = 0.0245, \tag{4.54}$$

Figure (4.6) shows the sliding mode observer trajectories while figure (4.7) shows the sliding mode observer error. It can be seen that the observer error is converging but it does not converge to zero. The observer error is oscillating and after number of samples it reaches the sliding manifold and then stays within it which shows the convergence of the observer error. The non-zero convergence may be due to the choice of the boundary layer φ or the auxiliary matrix $\boldsymbol{D}$.

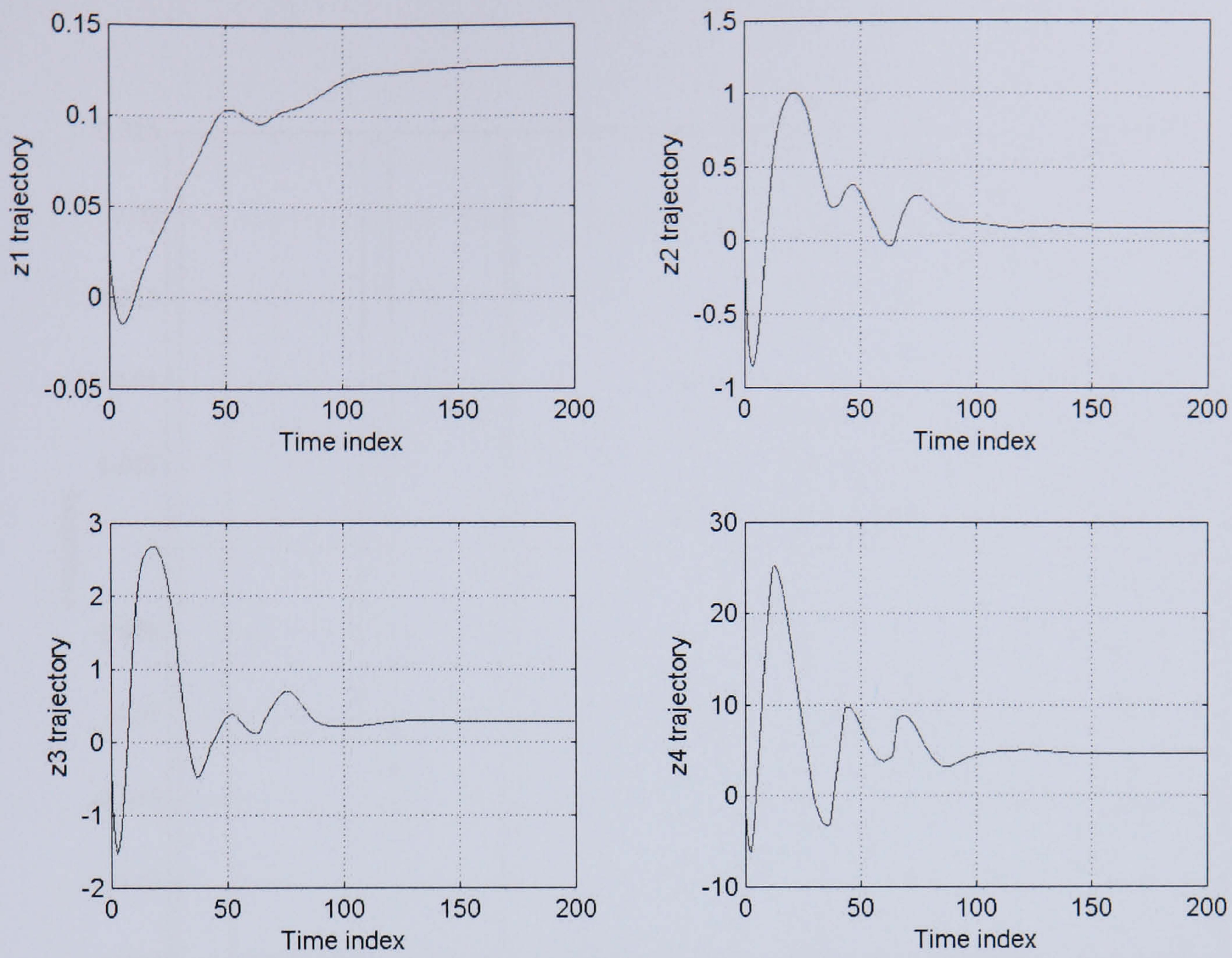

Figure 4.6: The sliding mode observer trajectory for the ball and beam system.

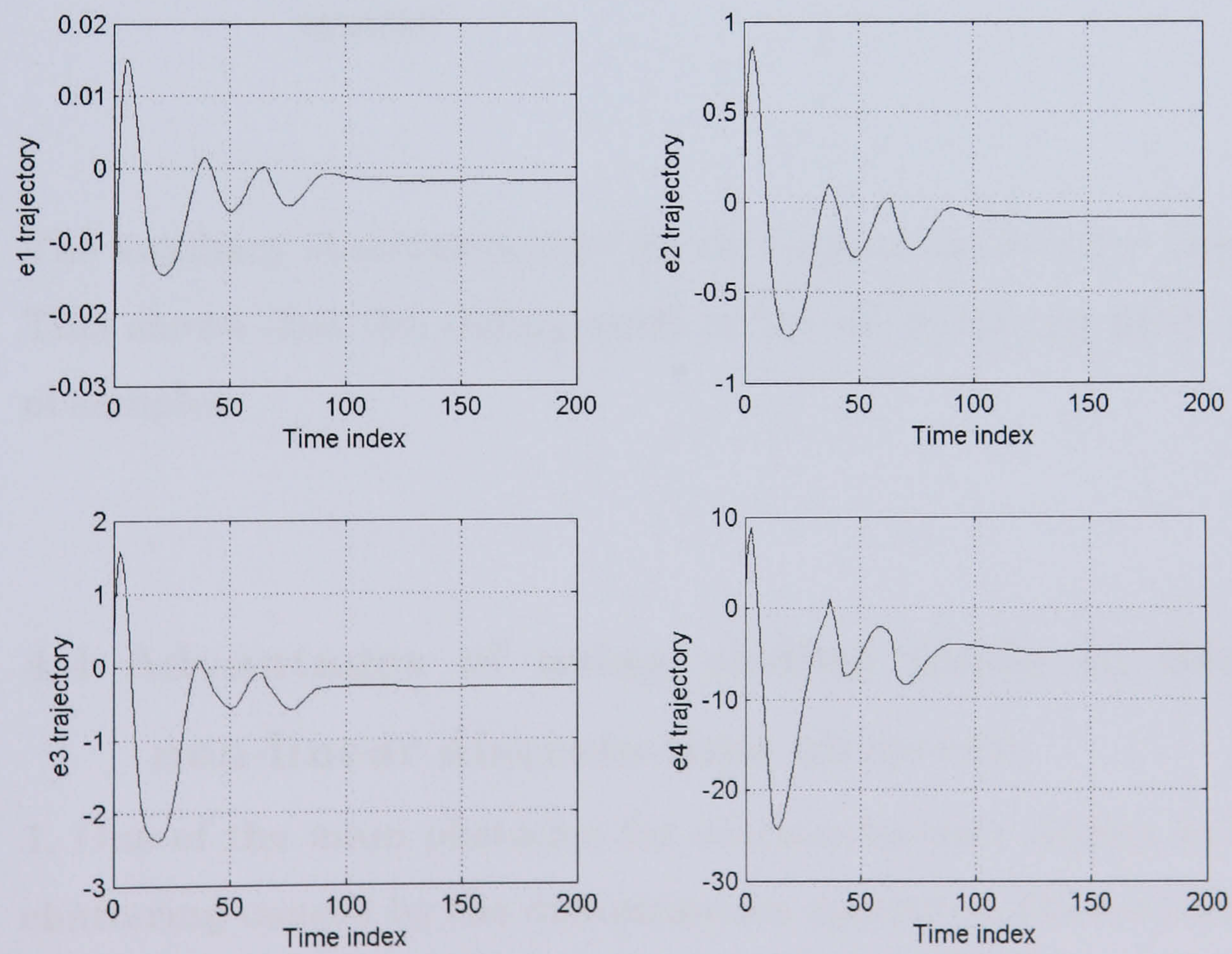

Figure 4.7: The sliding mode observer error for the ball and beam system.

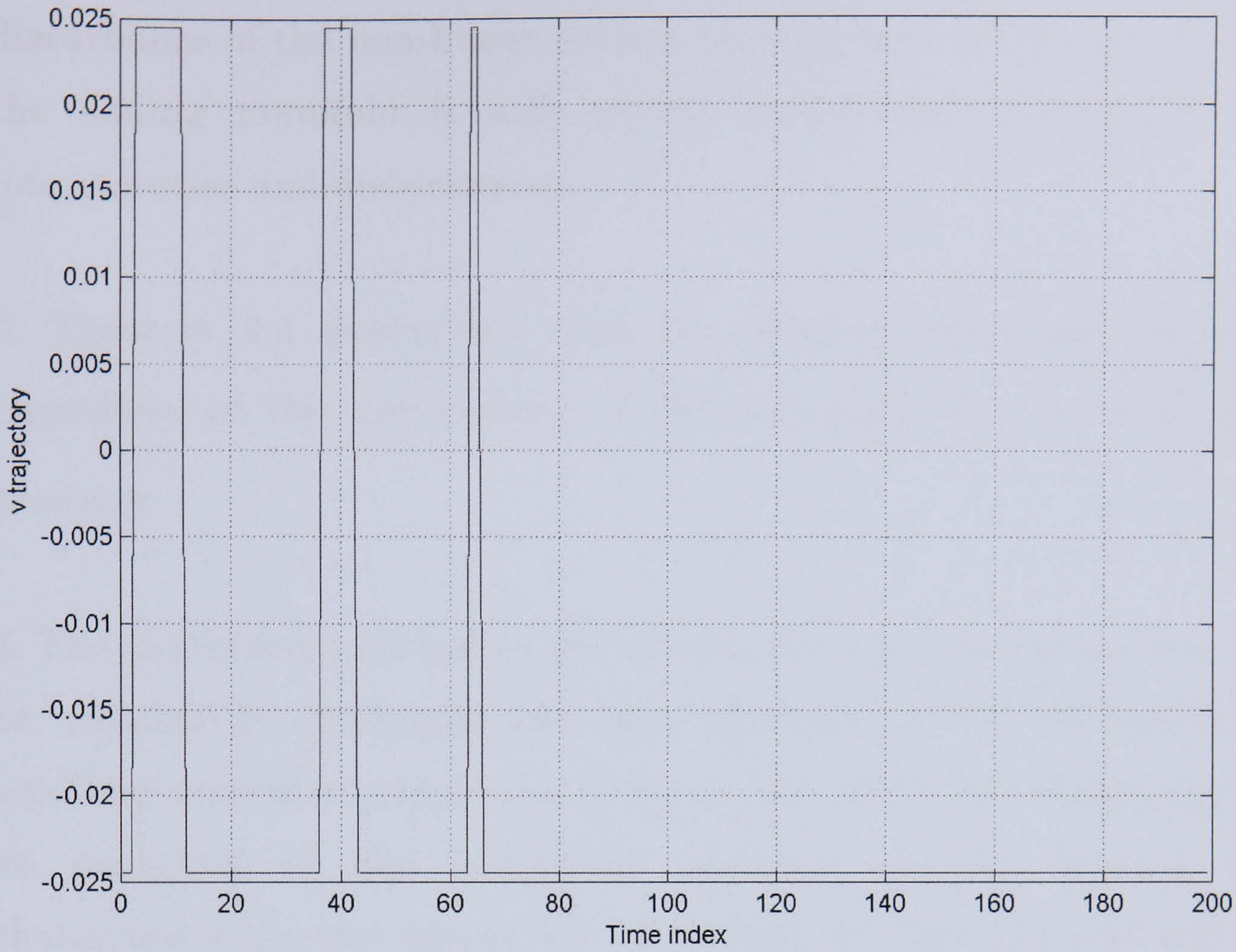

Figure 4.8: The auxiliary observer control for the ball and beam system.

The auxiliary observer control for this system is shown in Figure (4.8). This shows that the sliding mode is free of chattering after a number of samples.

4.4 Advantages of using sliding mode in designing non-linear discrete-time observer.

1. One of the main obstacles for continuous-time sliding mode is the chattering caused by the discontinuous control on the continuous-time system. On the other hand, the discrete-time sliding mode control is free of chattering for the discrete-time system.

2. The sliding mode is less sensitive to the uncertainty or to the disturbance of the non-linear system because once the trajectory is in the sliding manifold it will remain inside and will neglect the uncertainties and disturbances.

3. Theorem 4.1 guarantees that the observer error will converge regardless of the uncertainty of $\boldsymbol{R}_k(\boldsymbol{e}(k),\boldsymbol{z}(k))$ for the non-linear observer.

4. The chattering problem in the continuous-time control systems can be avoided by replacing the sign function of the discontinuous auxiliary control by saturation function [62]. While the chattering can be neglected in the non-linear observer designs because this chattering will occur during the calculation in computer and will not affect the control system.

4.5 Summary

The problem of divergence of the non-linear observer error, which is raised in chapter 2 due to the uncertainty of the Taylor remainder, $\boldsymbol{R}_k(\boldsymbol{e}(k),\boldsymbol{z}(k))$ is solved by using the discrete-time sliding mode technique. By choosing the boundary layer, the sliding surface is attractive and once the observer trajectory enters this boundary layer, it will remain within it. Therefore, the non-linear discrete-time observer error will converge. Theorem 4.1 guarantees that the observer error will converge to zero regardless of the uncertainty of $\boldsymbol{R}_k(\boldsymbol{e}(k),\boldsymbol{z}(k))$ for the non-linear observer.

In summary, using the sliding mode method guarantees the convergence of the observer error regardless of any uncertainties or disturbances to the observer of non-linear discrete-time systems.

Chapter 5

Conclusions

5.1 Summary of Findings

In this thesis, the design of an observer for a non-linear discrete-time system is presented. The design method splits the observer error equation into two parts. The first part is the observer error for the linearized system and the second part is an uncertain vector, which is the Taylor's residual.

The pole placement method is used in this general design which is then applied to two non-linear discrete-time systems; one of them is a practical system, which is the non-linear discrete-time ball and beam system. The observer errors for both systems are diverging. The second part of the observer error equation, which is the Taylor's residual, is causing the divergence of the observer error, since the first part of the observer error can be calculated using a pole placement method at assigned closed-loop poles.

The sliding mode technique is introduced in the design of the observer for a general non-linear discrete-time system to force the uncertain vector to converge. Sliding mode is a control law that switches at infinite speed to drive the system on the specified trajectory. The sliding mode dynamics do not depend on the system or the observer feedback gains but rather depends on the switching surface. An auxiliary control is added to the observer state-space equation to force the observer error onto the sliding surface and then to keep the observer error on this surface.

By defining the Lyapunov function as in equation (4.21) and under assumption (4.1), theorem (4.1) shows that $\|S(k)\|$ is decreasing monotonically, therefore $\|e(k)\|$ is decreasing. So, theorem (4.1) guarantees the convergence of the observer error if the auxiliary observer control, equation (4.22), has been chosen. So, by choosing the right boundary layer φ, the sliding manifold will attract the observer error to the sliding surface and keep it inside the sliding manifold. Therefore, the observer error will converge.

Simulations are presented for the observers for two non-linear discrete-time systems used as case studies mentioned above, after introducing the sliding mode technique. These simulations show the convergence of the observer error.

5.2 Contribution of the Thesis

Since there is no general design method for observers for non-linear discrete-time systems which will guarantee all the characteristics stated in chapter 1, the sliding mode design method, which is developed in this thesis, addresses some of them,

1. **Consistency**. Theorem (4.1) proves that the sliding mode technique guarantees the convergence the observer error.

2. **Robustness**. The observer error is attractive to the sliding surface and once it reaches the sliding manifold it will stay in the manifold regardless of the uncertainties. Therefore, by using the

sliding mode technique, the stability of the observer in the presence of bounded uncertainties is guaranteed.

3. **Computational complexity**. The sliding mode technique does not require complex computational procedures and the number of computations in this technique is not excessively large.

5.3 Future Work

Some of the issues presented in this research thesis may become the subject of further research.

5.3.1 The boundary layer φ, where the auxiliary observer control $v(k)$ can vary, is one of the key issues in the sliding mode design method. Research for a methodology procedure is needed to design the value of the boundary layer φ for the non-linear sliding mode observer for non-linear discrete-time systems.

5.3.2 The auxiliary vector $\boldsymbol{D}$ is also one of the key issues in the sliding mode design method. Research for a methodology procedure is needed to design the value of the auxiliary vector $\boldsymbol{D}$.

5.3.3 The sliding mode technique used in this thesis has been applied to single-input, single-output (SISO) non-linear discrete-time systems. Therefore, an investigation to extend this technique to multi-input, multi-output (MIMO) non-linear discrete-time systems could be undertaken.

Reference

1. Ogata, K., "Modern control engineering," 4th ed. 2002, Englewood Cliffs, N.J.: Prentice Hall.

2. Dorf, R.C., "Modern control systems," 5th ed. Addison-Wesley series in electrical engineering. 1989, Reading, Mass.: Addison-Wesley.

3. Palm, W.J., "Modeling, analysis, and control of dynamic systems," 2nd ed. 1999, New York ; Chichester: Wiley.

4. Franklin, G.F., J.D. Powell, and A. Emami-Naeini, "Feedback control of dynamic systems," 2nd ed. Addison-Wesley series in electrical and computer engineering. Control engineering. 1991, Reading, Mass.: Addison-Wesley.

5. Slotine, J.J.E. and W. Li, "Applied nonlinear control," 1991, Englewood Cliffs, N.J.: Prentice Hall.

6. Meng, J., Z. Zhang, G. Biswas, and N Sarkar, "Hybrid fault adaptive control of a wheeled mobile robot," Mechatronics, IEEE/ASME Transactions on, 2003. 8(2): pp. 226-233.

7. Nigrowsky, P., "Variable structure control of robot manipulators : (the example of SPRINTA). ver." PhD's Thesis, Brunel University, 1999.

8. Guldner, J. and V.I. Utkin, "Sliding mode control for gradient tracking and robot navigation using artificial potential fields," Robotics and Automation, IEEE Transactions on, 1995. 11(2): pp. 247-254.

9. Lohmiller, W. and J.J.E. Slotine, "Applications of metric observers for nonlinear systems," Control Applications, Proceedings of the 1996 IEEE International Conference on, 1996. pp. 367-372.

10. Gao, Z., T.G. Habetler, R.G. Harley, and R.S. Colby, "An adaptive Kalman filtering approach to induction machine stator winding temperature estimation based on a hybrid thermal model," Industry Applications Conference, 2005. 1: pp. 2-9.

11. Benchaib, A., A. Rachid, and E. Audrezet, "Sliding mode input-output linearization and field orientation for real-time control of induction motors," Power Electronics, IEEE Transactions on, 1999. 14(1): pp. 3-13.

12. Lin, F.J., R.J. Wai, R.H. Kuo, and D.C. Liu, "A Comparative study of sliding mode and modal reference adaptive speed observers for induction motor drive," Electric Power System Research, 1998. 44: pp. 163-174.

13. Elmali, H. and N. Olgac, "Satellite attitude control via sliding mode with perturbation estimation," Control Theory and Applications, IEE Proceedings of, 1996. 143(3): pp. 276-282.

14. Qu, Z., "Robust control of nonlinear uncertain systems," Wiley series in nonlinear science. 1998, New York: Wiley.

15. Baumann, W. and W. Rugh, "Feedback control of nonlinear systems by extended linearization," Automatic Control, IEEE Transactions on, 1986. 31(1): pp. 40-46.

16. Roboulet, C. and C. Champetier, "A new method for linearizing nonlinear systems: Pseudolinearization," International Journal of Control, 1984. 40: pp. 631-638.

17. Aström, K.J. and B. Wittenmark, "Adaptive control," 1989: Addison-Wesley.

18. Huang, K.Y., H.C. Chin, and Y.C. Huang, "A model reference adaptive control strategy for interruptible load management," Power Systems, IEEE Transactions on, 2004. 19(1): pp. 683-689.

19. Kim, Y.H., K., Y.S. Yoo, and K.H. Oh, " Self-tuning adaptive controllers using state observers," Power Conversion Conference, Conference Record of the. 1993. pp. 278-283.

20. Wittenmark, B., "A survey of adaptive control applications," Dynamic Modeling Control Applications for Industry Workshop, IEEE Industry Applications Society, 1997. pp. 32-36.

21. Astrom, K.J. and B. Wittenmark, "A survey of adaptive control applications," Decision and Control, Proceedings of the 34th IEEE Conference on, 1995. 1: pp. 649-654.

22. Luo, W., Y.C. Chu, and K.V Ling, "Inverse optimal adaptive control for attitude tracking of spacecraft," Automatic Control, IEEE Transactions on, 2005. 50(11): pp. 1639-1654.

23. Shen, Z., C. Guo, and H Li, "General Fuzzified CMAC Based Model Reference Adaptive Control for Ship Steering," Intelligent Control, Proceedings of the IEEE International Symposium on, 2005. pp. 1263-1268.

24. Antonelli, G., F. Caccavale, S. Chiaverini, and G. Fusco, "A novel adaptive control law for underwater vehicles," Control Systems Technology, IEEE Transactions on, 2003. 11(2): pp. 221-232.

25. Subbarao, K., A. Verma, and J.L. Junkins, "Model reference adaptive control of constrained cooperative manipulators," Control Applications, Proceedings of the 2001 IEEE International Conference on, 2001. pp. 553-558.

26. Luenberger, D., "Observers for multivariable systems," Automatic Control, IEEE Transactions on, 1966. 11(2): pp. 190-197.

27. Beale, S. R. and B. Shafai. "Robust control system design with proportional integral observer," Decision and Control, Proceedings of the 27th IEEE Conference on. 1988. 1: pp. 554–557.

28. Radke, A. and Z. Gao, "A survey of state and disturbance observers for practitioners," American Control Conference, Proceedings of the, 2006. pp. 5183-5188.

29. Kalman, R. and R. Bucy, "New results in linear filtering and prediction theory," Basic Engineering, Transactions of the ASME - Journal of, 1961. 83: pp. 95-107.

30. Kalman, R.E., "A new approach to linear filtering and prediction problems," Basic Engineering, Transaction of the ASME-Journal of, 1960. 82(series D): pp. 35-45.

31. Leigh, J.R., "Control theory," 2nd ed. 2004, London: Institution of Electrical Engineers.

32. Boutayeb, M., H. Rafaralahy, and M. Darouach, "Convergence analysis of the extended Kalman filter used as an observer for nonlinear deterministic discrete-time systems," Automatic Control, IEEE Transactions on, 1997. 42(4): pp. 581-586.

33. Boutayeb, M. and D. Aubry, "A strong tracking extended Kalman observer for nonlinear discrete-time systems," Automatic Control, IEEE Transactions on, 1999. 44(8): pp. 1550-1556.

34. Michigan, University of, "Example: Modeling the Ball and Beam Experiment. Control Tutorials for Matlab," Available from:http://www.engin.umich.edu/group/ctm/examples/ball/ball.html.

35. Gauthier, J.P., H. Hammouri, and S. Othman, "A simple observer for nonlinear systems applications to bioreactors," Automatic Control, IEEE Transactions on, 1992. 37(6): pp. 875-880.

36. Park, C.W., S. Lee, J. Lee, and H.G. Sung, "Nonlinear Observer Based Flux Estimation of Induction Motors," in Electromagnetic Field Computation, 12th Biennial IEEE Conference on. 2006. pp. 329-329.

37. De Leon-Morales, J., K. Busawon, and G. Acosta-Villarreal, “Digital implementation of an observer-based controller for a rigid robot,” Decision and Control, Proceedings of the 37th IEEE Conference on, 1998. 1: pp. 957-962.

38. Lohmiller, W. and J.J.E. Slotine, “On metric observers for nonlinear systems,” Control Applications, Proceedings of the 1996 IEEE International Conference on, 1996. pp. 320-326.

39. Lohmiller, W. and J.J.E. Slotine, “On metric controllers and observers for nonlinear systems,” Decision and Control, Proceedings of the 35th IEEE, 1996. 2: pp 1477-1482.

40. Slotine, J.J.E., Hedrick, J.K., and Misawa, E, “On Sliding Observers for Nonlinear Systems,” A.S.M.E. J. Dynamic Systems, Measurement, and Control, 1987. 109: pp. 245-252.

41. Sira-Ramirez, H., “Dynamical sliding mode control strategies in the regulation of nonlinear chemical processes,” International Journal of Control, 1992. 56(1): pp. 1-21.

42. Sira-Ramirez, H., M. Zribi, and S. Ahmad, “Dynamical sliding mode control approach for vertical flight regulation in helicopters,” Control Theory and Applications, IEE Proceedings of, 1994. 141(1): pp. 19-24.

43. Hostetter, G.H. and J. Meditch, “On the generalization of observers to systems with unmeasurable, unknown inputs,” Automatica, 1973. 9: pp. 721-724.

44. Rajamani, R., “Observers for Lipschitz nonlinear systems,” Automatic Control, IEEE Transactions on, 1998. 43(3): pp. 397-401.

45. Itkis, U., “Control systems of variable structure,” 1976, New York

46. Utkin, V., “Variable structure systems with sliding modes,” Automatic Control, IEEE Transactions on, 1977. 22(2): pp. 212-222.

47. Edwards, C. and S. K. Spurgeon, “Sliding mode control: theory and applications,” 1998, London: Taylor & Francis.

48. Baik, I.C., K.H. Kim, and M.J. Youn, “Robust nonlinear speed control of PM synchronous motor using adaptive and sliding mode control techniques,” Electric Power Applications, IEE Proceedings , 1998. 145(4): pp. 369-376.

49. McCann, R. A. and I. Husain, “Application of a sliding mode observer for switched reluctance motor drives,” in Industry Applications Conference, 1997. 1: pp. 525-532.

50. Cho, S.H. and K.A. Edge, “Adaptive Sliding Mode tracking control of hydraulic servosystems with unknown non-linear friction and modelling error,” Proceedings of the Institution of Mechanical Engineers, Part I, Journal of Systems & Control Engineering, 2000. 214(4): pp. 247-257.

51. Spurgeon, S. and C. Edwards, “Non-linear control of a 175 kW gas burner,” Computing & Control Engineering Journal, 1995. 6(1): pp. 29-34.

52. Allen, M., F. Bernelli.-Zazzera, R. Scattolini, "Sliding mode control of a large flexible space structure," Control Engineering Practice, 2000. 8(8): pp. 861-871.

53. Lee, S.S. and J.K. Park, "Design of power system stabilizer using observer/sliding mode, observer/sliding mode-model following and H[infinity]/sliding mode controllers for small-signal stability study," Electrical Power & Energy Systems, International Journal of, 1998. 20(8): pp. 543-553.

54. Young, K.D., V.I. Utkin, and U. Ozguner, "A control engineer's guide to sliding mode control," Control Systems Technology, IEEE Transactions on, 1999. 7(3): pp. 328-342.

55. Schrijver, E., J. van Dijk, "Disturbance Observers for Rigid Mechanical Systems: Equivalence, Stability, and Design," Dynamic Systems, Measurement, and control, Journal of. 2002. 124: pp. 539-548.

56. Edwards, C., A. Akoachere, and S.K. Spurgeon, "Sliding-mode output feedback controller design using linear matrix inequalities," Automatic Control, IEEE Transactions on, 2001. 46(1): pp. 115-119.

57. Hace, A., K. Jezernik, and M. Terbuc, "VSS motion control for a laser-cutting machine," Control Engineering Practice, 2001. 9(1): pp. 67-77.

58. Zak, S. H. and S. Hui, "On variable structure output feedback controllers for uncertain dynamic systems," Automatic Control, IEEE Transactions on, 1993. 38(10): pp. 1509-1512.

59. Choi, H.H., “A new method for variable structure control system design: a linear matrix inequality approach,” Automatica, 1997. 33: pp. 2089–2092.

60. Xiang, J., H. Su, and J. Chu, “Robust sliding mode output feedback control design using ILMI approach,” American Control Conference, Proceedings of the, 2005. 6: pp. 4078-4083.

61. Utkin, V.I. “Sliding mode control in electromechanical systems,” 1999, London: Taylor and Francis.

62. Lewis, A.S. and A. Sinha, “Sliding mode output feedback control of mechanical systems with bounded disturbances,” American Control Conference, Proceeding Of the. 1995. pp. 3735-3739

63. Vande Vegte, J., “Feedback control systems,” 3rd ed. 1994, Englewood Cliffs, N.J.: Prentice Hall.

64. Stefani, R.T., B. Shahian, C.J. Savant, and G.H. Hostetter, “Design of feedback control systems,” 4th ed. 2002, Oxford: Oxford University Press.

65. Khalil, H.K., “Nonlinear systems,” 3rd ed. 2002, Upper Saddle River, NJ; London: Prentice Hall.

66. Veluvolu, K.C., Y.C. Soh, W. Cao, and Z.Y. Liu, "Discrete-time sliding mode observer design for a class of uncertain nonlinear systems," American Control Conference, proceeding of the, 2006. pp. 2605-2610.

67. Davila, J., L. Fridman, and A. Levant, "Second-order sliding-mode observer for mechanical systems," Automatic Control, IEEE Transactions on, 2005. 50(11), pp. 1785-1789.

68. Veluvolu, K.C., Y.C. Soh, W. Cao, and Z.Y. Liu, "Observer with multiple sliding modes for a class of nonlinear uncertain systems," American Control Conference, Proceedings of the, 2005. pp. 2445-2450.

69. Qu, Z., and Y. Jin, "Robust control of nonlinear systems in the presence of unknown exogenous dynamics," Decision and Control, Proceedings of the IEEE Conference on, 2001. pp. 2784–2790.

70. Qu, Z. and Y. Jin, "Robust Control of Nonlinear Systems in the Presence of Unknown Exogenous Dynamics," Automatic Control, IEEE Transactions on, 2003. 48(2): pp. 336-343.

71. Gauthier, J.P., and I.A.K. Kupka, "Observability and observes for nonlinear systems," SIAM Journal of Control Optimization, 1994, 32: pp. 975-994.

72. Busawon, K., M. Farza, and H. Hammouri, "Observer design for a special class of nonlinear systems," Control, International Journal of, 1998, 71: pp. 405-418.

73. Doyle, J.C. and G. Stein, "Multivariable feedback design: Concepts for a classical/modern synthesis," Automatic Control, IEEE Transactions on, 1981, 26(1): pp. 4-16.

74. Burl, J.B., “Linear Optimal Control: H_2 and H_∞ methods,” 1999, Addison Wesley.

75. Doyle, J.C., K. Glover, P.P. Khargonekar, and B.A. Francis, “State space solutions to standard H_2 and H_∞ control problems,” Automatic Control, IEEE Transactions on, 1989, 34(8): pp. 831–847.

76. Aitken, V.C. and H.M. Schwartz. “Towards robust discrete-time sliding mode observers,” Proceedings of 1995 American control conference, 1995. pp. 3730-3734.

77. Jiang, L. and Q.H. Wu, “Nonlinear adaptive control via sliding-mode state and perturbation observer,” Control Theory and Applications, IEE Proceedings of, 2002. 149(4): pp. 269-277.

78. Juang, J.C. and C.M. Lee, “Design of sliding mode controllers with bounded L2 gain performance: an LMI approach,” Control, International Journal of, 2005. 78(9): pp. 647–661.

79. Singh, S.N., “Asymptotically decoupled discontinuous control of systems and nonlinear aircraft maneuver,” Aerospace and Electronic Systems, IEEE Transactions on, 1989. 25(3), pp. 380-391.

80. Mudge, S.K. and R.J. Patton, “Variable structure control laws for aircraft maneuvers,” Control ’88., International Conference on, 1988. pp. 564-568.

81. Zinober, A.S.I., M.K. Yew, and R.J. Patton, “Variable structure model-following control of flight dynamics,” Control ’88, International Conference on, 1988. pp. 707-712.

82. Sira-Ramirez, H. and T.A.W. Dwyer, "Variable structure controller design for spacecraft rotation damping," Automatic Control, IEEE Transactions on, 1987. 32(5): pp. 435-438.

83. Singh, S.N., and A. Iyer, "Nonlinear Decoupling sliding mode control and attitude control of spacecraft," Aerospace and Electronic Systems, IEEE Transactions on, 1989. 25(5): pp. 621-633.

84. Singh, S.K., "Decentralized variable structure control for tracking in nonlinear systems," International Journal of Control, 1990. 52(4): pp. 811-831.

85. Utkin, V.I., "Sliding mode control in discrete-time and difference systems," Variable Structure and Lyapunov Control, A. S. Zinober, Springer-Verlag, London, 1993, pp. 83–103.

86. Young, K.K.D., P.V. Kokotovic, and V.I. Utkin, "A singular perturbation analysis of high-gain feedback systems," Automatic Control, IEEE Transactions on, 1977. 22(6): pp. 931–938.

87. Friedland, B., "The Control Handbook," 1996, CRC Press, IEEE Press.

88. Chen, C.T., "Linear System Theory and Design," Oxford Series in Electrical and Computer Engineering, 3rd ed. 1998, Oxford University Press.

89. Hori, Y. and K. Shimura, "Position/force control of multi-axis robot manipulator based on the tdof robust servo controller for each joint," in American Control Conference, Proceedings of the, 1992. 1: pp. 753–757.

90. Franklin, G.F., J.D. Powell, and A. Emami-Naeini, "Feedback Control of Dynamic Systems," 4th ed. Addison-Wesley Longman Publishing Co., Inc., 2002.

91. Johnson, C., "Accommodation of external disturbances in linear regulator and servomechanism problems," Automatic Control, IEEE Transactions on, 1971. 16(6): pp. 635–644.

92. Johnson, C., "Theory of disturbance-accommodating controllers," Control and Dynamic Systems, 1976. 12: pp. 387-489.

93. Profeta, J.A., W.G. Vogt, and M.H. Mickle, "Disturbance estimation and compensation in linear systems," Aerospace and Electronic systems, IEEE Transactions on, 1990. 26(2): pp. 225–231.

Appendices

Appendix A
Discretization of Continuous-Time Systems

A.1 Introduction

The operation that transforms a continuous-time signal into a discrete-time signal is called discretization. The sample and hold circuit and analog to digital converter convert the continuous-time signal into a sequence of numerically binary words. Such an analog to digital (A/D) conversion process is called coding or encoding [87].

In this thesis, a solution method to design an observer for a non-linear discrete-time system is implemented in chapters (2) and (4). So, a practical system is needed to which the solution method is applied. A ball and beam non-linear continuous-time system has been chosen. Therefore, the discretization methods for transforming a continuous-time state-space system to a discrete-time state-space system will be discussed in this Appendix.

A.2 Discretization of Continuous-Time Linear Systems

In the design of a digital controller for a continuous-time system, the conversion of the continuous-time state-space equations into discrete-time state-space equations is needed. This conversion can be done by introducing imaginary samplers and imaginary hold devices into the continuous-time system. The error introduced by discretization may be made negligible by using a very small sampling period compared with a significant time constant of the system [87]. Some of the basics of the discretization method, which is well published, will be reviewed in this section [63], [87].

Given a linear continuous-time state space system:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{A.1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \tag{A.2}$$

where $\boldsymbol{x} \in \Re^n$ is the state vector, $\boldsymbol{y} \in \Re^p$ is the output vector and $\boldsymbol{u} \in \Re^m$ is the input vector. $\boldsymbol{A}$ is an $n \times n$ state matrix, $\boldsymbol{B}$ is an $n \times m$ input matrix and $\boldsymbol{C}$ is an $p \times n$ output matrix. The pair $(\boldsymbol{A}, \boldsymbol{B})$ is assumed controllable.

The solution of the state space system (A.1) is

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}(0) + \int_0^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)\,d\tau \tag{A.3}$$

if the initial time is taken as t_0, then

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)\,d\tau \tag{A.4}$$

Let $\boldsymbol{u}(\tau) = \boldsymbol{u}(t_0)$ fixed for $t_0 \le \tau < t$ and if $\boldsymbol{A}$ is non-singular, then

$$\begin{aligned} \int_{t_0}^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\,d\tau &= \left[e^{\boldsymbol{A}(t-\tau)}\left(-\boldsymbol{A}^{-1}\boldsymbol{B}\right)\right]_{t_0}^t \\ &= \left[e^{\boldsymbol{A}(t-t_0)} - \boldsymbol{I}\right]\left(\boldsymbol{A}^{-1}\boldsymbol{B}\right) \end{aligned} \tag{A.5}$$

and this is dependent only on the time difference $(t - t_0)$.

Let $(t-t_0)=T$ where T is the sampling time, i.e. by solving the continuous-time state-space equation over 1 sample.

Let

$$\boldsymbol{F} = e^{AT} \tag{A.6}$$

$$\boldsymbol{G} = \int_{t_0}^{t} e^{A(t-\tau)}\boldsymbol{B}d\tau \tag{A.7}$$

From equations (A.4 ~ A.7):

$$\boldsymbol{x}(T) = \boldsymbol{F}\boldsymbol{x}(0) + \boldsymbol{G}\boldsymbol{u}(0) \tag{A.8}$$

$$\boldsymbol{x}(2T) = \boldsymbol{F}\boldsymbol{x}(T) + \boldsymbol{G}\boldsymbol{u}(T) \tag{A.9}$$

$$\boldsymbol{x}(3T) = \boldsymbol{F}\boldsymbol{x}(2T) + \boldsymbol{G}\boldsymbol{u}(2T) \tag{A.10}$$

Therefore, the general description of state space system in discrete-time is,

$$\boldsymbol{x}\left((k+1)T\right) = \boldsymbol{F}\boldsymbol{x}\left(kT\right) + \boldsymbol{G}\boldsymbol{u}\left(kT\right) \tag{A.11}$$

To simplify equation (A.11), assume that $kT = k$ and $(k+1)T = k+1$, then, equation (A.11) can be written as,

$$\boldsymbol{x}\left(k+1\right) = \boldsymbol{F}\boldsymbol{x}\left(k\right) + \boldsymbol{G}\boldsymbol{u}\left(k\right) \tag{A.12}$$

Also,

$$y\left(k\right) = \boldsymbol{C}\boldsymbol{x}\left(k\right) \tag{A.13}$$

where $\boldsymbol{u}\left(k\right)=$ constant for $t_k \le t < t_{k+1}$ with initial condition $\boldsymbol{x}\left(0\right)$.

Equations (A.12) and (A.13) comprise the discrete-time state-space model of the continuous-time state space system (A.1) and (A.2).

A.3 Discretization Method by Approximation

In the preceding sections, the discretization method has been discussed which solves the continuous-time state space system at a sampling time $T = (t - t_0)$. In this section, an approximation of the continuous-time state space system can be used.

Given a linear continuous-time state space system (A.1):

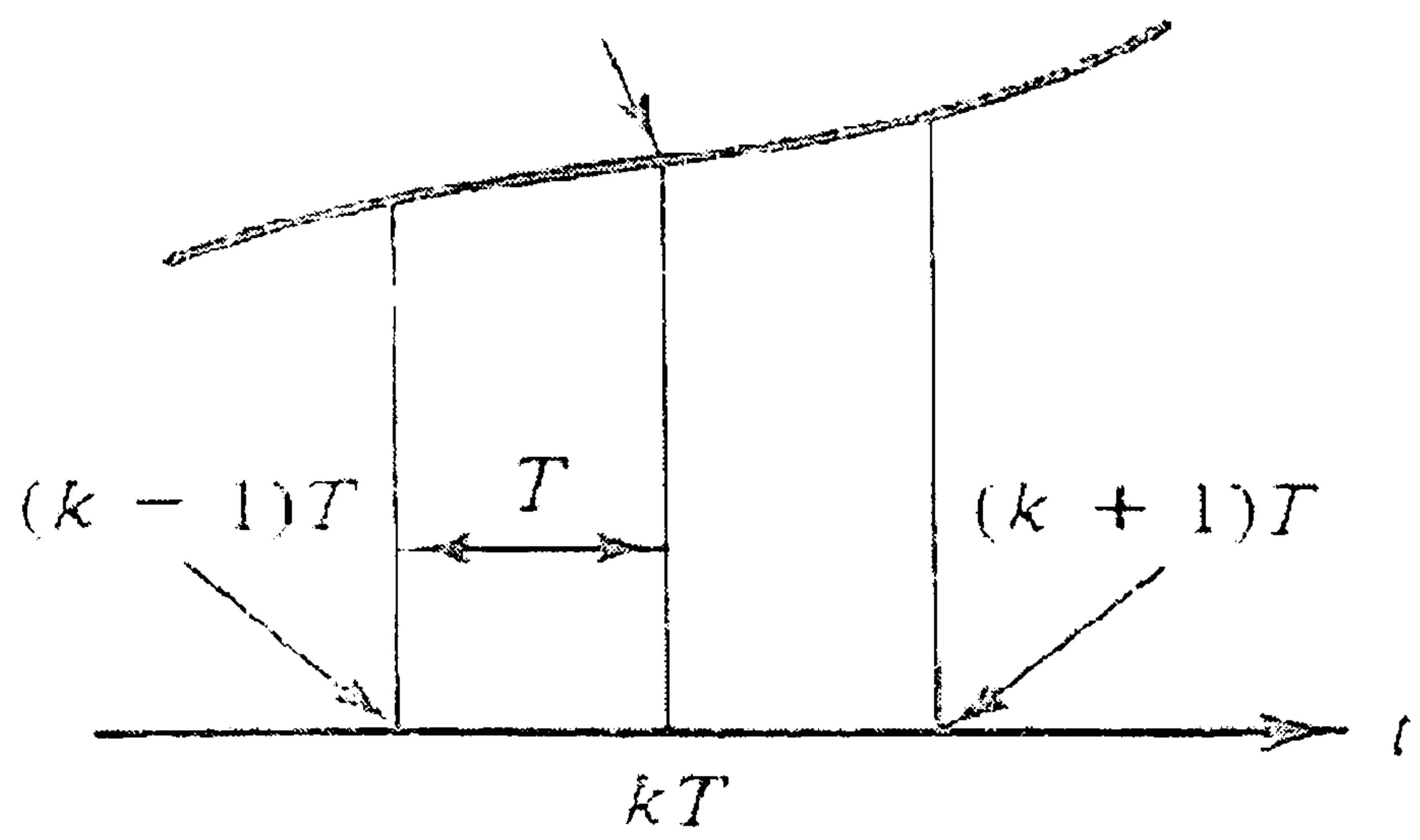

Figure A.1: Finite difference.

From calculus, the derivative, $\dot{x}(t)$ is defined by,

$$\dot{x}(t) = \lim_{\Delta t \to 0} \frac{x(t + \Delta t) - x(t)}{\Delta t} \tag{A.15}$$

Let $\Delta t = T$ and $t = kT$ and since $\dot{x}(t)$ is the slope of the $x(t)$ curve at $t = kT$, figure (A.1).

$$\dot{x}(t) = \lim_{T \to 0} \frac{x((k+1)T) - x(kT)}{T} \tag{A.16}$$

If T is small,

$$\dot{x}(t) \approx \frac{x((k+1)T) - x(kT)}{T} \tag{A.17}$$

Therefore, as the sampling time T becomes very small the approximation (A.17) will be exact. This approximation is called the first difference approximation.

Equation (A.17) can be rewritten as, for T is small,

$$\dot{x}(t) = \frac{x((k+1)T) - x(kT)}{T} \tag{A.18}$$

$$\dot{x}(t) = \frac{x(kT) - x((k-1)T)}{T} \tag{A.19}$$

$$\dot{x}(t) = \frac{x((k+1)T) - x((k-1)T)}{2T} \tag{A.20}$$

Equation (A.18) is called first forward difference approximation, equation (A.19) is called first backward difference approximation and equation (A.20) is called first central difference approximation. If $x(k+1)$ is available, the first central difference approximation will be the best approximation [63].

The second difference approximation of $\ddot{x}(t)$ can be derived as the first difference approximation at $t = kT$,

$$\ddot{x}(t) = \frac{x((k+2)T) - 2x((k+1)T) + x(kT)}{T^2} \tag{A.21}$$

$$\ddot{x}(t) = \frac{x(kT) - 2x((k-1)T) + x((k-2)T)}{T^2} \tag{A.22}$$

$$\ddot{x}(t) = \frac{x((k+1)T) - 2x(kT) + x((k-1)T)}{T^2} \tag{A.23}$$

Equation (A.21) is called second forward difference approximation, equation (A.22) is called second backward difference approximation and equation (A.23) is called second central difference approximation [63].

A.4 Discretization of Continuous-Time Non-linear Systems

Control theory and practice have been very successful in dealing with continuous-time linear control systems and have been implemented in linear discrete-time system. On the other hand, the situation is radically different for non-linear continuous-time systems. Although several methods for discretizing non-linear continuous-time systems have emerged, none of them can be sufficient to encompass all non-linear continuous-time systems.

Consider the non-linear continuous-time state space system:

$$\dot{x}(t) = f(x(t)) + Bu(t) \tag{A.24}$$

$$y(t) = Cx(t) \tag{A.25}$$

One of the methods of discretizing a non-linear continuous-time system described by equation (A.24) is by linearizing and then discretizing it by using any of the methods described in sections (A.2) and (A.3).

Another method is to use the approximation method explained in section (A.3) directly to approximate the non-linear continuous-time system using,

$$\dot{x}(t) = \frac{x((k+1)T) - x(kT)}{T} \tag{A.26}$$

given the assumption that the sampling time T is very small.

This method is implemented in appendix (B) to discretize the non-linear continuous-time system.

Appendix B

The Ball and Beam System

B.1 Problem Setup

A practical application is useful to illustrate the discretizing method by approximation for non-linear continuous-time systems. A ball and beam system [34] is chosen as the problem.

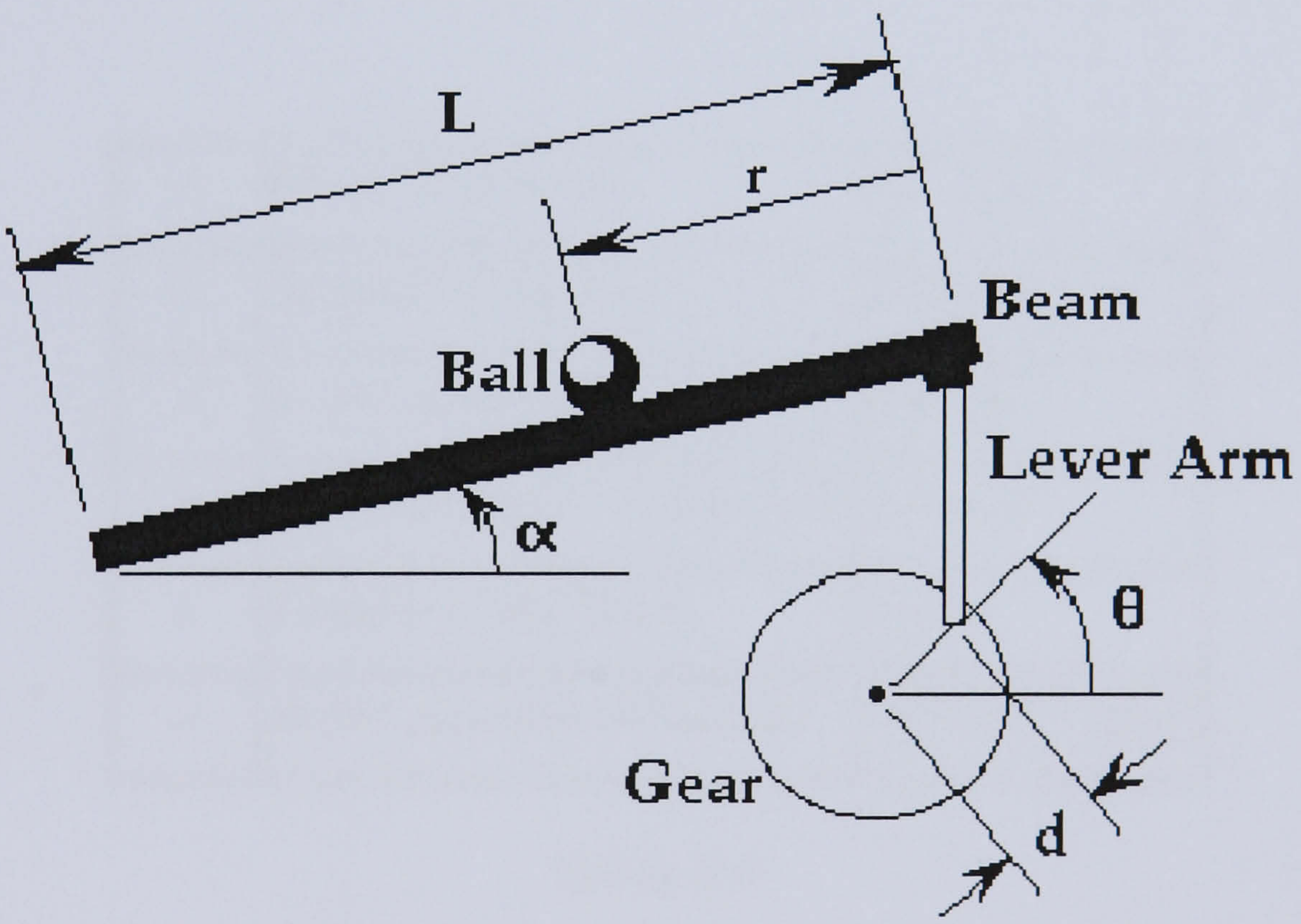

Figure B.1: Ball and Beam system [34].

A ball is placed on a beam, see figure (B.1), where it is allowed to roll with one degree of freedom along the length of the beam. A lever arm is attached to the beam at one end and the other end of the lever arm is attached to the servo gear. As the servo gear turns by an angle theta (θ), the lever changes the angle of the beam by alpha (α). When the beam angle alpha is changed from the horizontal position, gravity causes the ball to roll along the beam. A controller will be designed for this system so that the ball's position (r) can be manipulated.

For this problem, it is assumed that the ball rolls without slipping and the friction between the ball and beam is negligible. The constants and variables for this problem are defined as table (B.1).

m	Mass of the ball	0.11 kg
R	Radius of the ball	0.015 m
d	Lever arm offset	0.03 m
g	Gravitational acceleration	9.8 m/s^2
L	Length of the beam	1.0 m
J	Ball's moment of inertia	9.99x10^{-6} kgm^2

Table B.1

The Lagrangian equations of motion for the ball are [34],

$$\left(\frac{J}{R^2}+m\right)\ddot{r}(t)-mr(t)\left(\dot{\alpha}(t)\right)^2+mg\sin\left(\alpha(t)\right)=0 \qquad \text{(B.1)}$$

$$\ddot{r}(t)=\frac{mr(t)(\dot{\alpha}(t))^2}{\left(\frac{J}{R^2}+m\right)}-\frac{mg\sin(\alpha(t))}{\left(\frac{J}{R^2}+m\right)} \tag{B.2}$$

The equation which relates the beam angle to the angle of the gear can be approximated by the equation,

$$\alpha=\frac{d}{L}\theta$$

(B.3)

If it is assumed that the gear and lever arm are not to be used, but instead a motor at the centre of the beam applies torque to the beam, in order to control the position, and assuming the beam angle (α) is small, then,

$$\sin(\alpha(t))\approx\alpha(t) \tag{B.4}$$

Then,

$$\ddot{r}(t)=\frac{mr(t)(\dot{\alpha}(t))^2}{\left(\frac{J}{R^2}+m\right)}-\frac{mg\alpha(t)}{\left(\frac{J}{R^2}+m\right)} \tag{B.5}$$

$$\ddot{\alpha}(t)=\tau \tag{B.6}$$

The system input is torque (τ) and the output to be controlled is ball position (r).

B.2 Discretization Method by Approximation

The approximation technique discussed in Appendix A can now be used to discretize the non-linear continuous-time system (B.1-2).

Using the state variable approach, let,

$$x_1(t) = r(t) \tag{B.7}$$

$$x_2(t) = \dot{r}(t) \tag{B.8}$$

$$x_3(t) = \alpha(t) \tag{B.9}$$

$$x_4(t) = \dot{\alpha}(t) \quad \text{B.10)}$$

Then,

$$\dot{x}_1(t) = x_2(t) \quad \text{B.11)}$$

$$\dot{x}_2(t) = \frac{m x_1(t)\left(x_4(t)\right)^2}{\left(\frac{J}{R^2} + m\right)} - \frac{m g x_3(t)}{\left(\frac{J}{R^2} + m\right)} \tag{B.12}$$

$$\dot{x}_3(t) = x_4(t) \tag{B.13}$$

$$\dot{x}_4(t) = u(t) \tag{B.14}$$

where, $u(t)$ is the system input.

Let

$$h = \frac{m}{\left(\frac{J}{R^2} + m\right)}$$

Then the state space model becomes,

$$\dot{x}_1(t) = x_2(t) \tag{B.15}$$

$$\dot{x}_2(t) = hx_1(t)(x_4(t))^2 - ghx_3(t) \tag{B.16}$$

$$\dot{x}_3(t) = x_4(t) \quad \text{B.17)}$$

$$\dot{x}_4(t) = u(t) \tag{B.18}$$

The differential equations (B.15~18) can be approximated by using the first forward difference approximation equation [63],

$$\dot{x}_i(t) = \frac{x_i((k+1)T) - x_i(kT)}{T} \tag{B.19}$$

for $t = kT$, i = 1,2,...4. T is the sampling time and $k = 1,2,3,\ldots..(t/T)$.

By substituting equation (B.19) into equations (B.15~18),

$$\frac{x_1((k+1)T) - x_1(kT)}{T} = x_2(kT) \tag{B.20}$$

$$x_1((k+1)T) = x_1(kT) + Tx_2(kT) \tag{B.21}$$

$$\frac{x_2\left((k+1)T\right)-x_2\left(kT\right)}{T}=hx_1\left(kT\right)\left(x_4\left(kT\right)\right)^2 \\ -ghx_3\left(kT\right)$$

(B.22)

$$x_2\left((k+1)T\right)=x_2\left(kT\right)-ghTx_3\left(kT\right) \\ +hTx_1\left(kT\right)\left(x_4\left(kT\right)\right)^2$$

(B.23)

$$\frac{x_3\left((k+1)T\right)-x_3\left(kT\right)}{T}=x_4\left(kT\right) \tag{B.24}$$

$$x_3\left((k+1)T\right)=x_3\left(kT\right)+Tx_4\left(kT\right) \tag{B.25}$$

$$\frac{x_4\left((k+1)T\right)-x_4\left(kT\right)}{T}=u\left(kT\right) \tag{B.26}$$

$$x_4\left((k+1)T\right)=x_4\left(kT\right)+Tu\left(kT\right) \tag{B.27}$$

So, the continuous-time system (B.15~18) can be approximated by using equations (B.21), (B.23), (B.25) and (B.27).

Therefore, the discrete-time ball and beam can be written as:

$$x_1\left((k+1)T\right)=x_1\left(kT\right)+Tx_2\left(kT\right) \qquad \text{B.28)}$$

$$x_2\left((k+1)T\right)=x_2\left(kT\right)-ghTx_3\left(kT\right) \\ +hTx_1\left(kT\right)\left(x_4\left(kT\right)\right)^2 \tag{B.29}$$

$$x_3\left((k+1)T\right)=x_3\left(kT\right)+Tx_4\left(kT\right) \tag{B.30}$$

$$x_4\left((k+1)T\right)=x_4\left(kT\right)+Tu\left(kT\right) \tag{B.31}$$

Since the sampling time T is constant, the discrete-time can be represented as:

$$kT=k \tag{B.32}$$

$$(k+1)T=(k+1) \tag{B.33}$$

Let us rewrite $x_i\left((k+1)T\right)$ (equations (B.28~31)) in the form,

$$\boldsymbol{x}(k+1)=\boldsymbol{f}\left(\boldsymbol{x}(k)\right)+\boldsymbol{B}u(k) \tag{B.34}$$

where

$$\boldsymbol{f}\left(\boldsymbol{x}(k)\right)=\begin{bmatrix} x_1(k)+Tx_2(k) \\ hTx_1(k)x_4^2(k)+x_2(k)-ghTx_3(k) \\ x_3(k)+Tx_4(k) \\ x_4(k) \end{bmatrix} \tag{B.35}$$

$$\boldsymbol{B}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ T \end{bmatrix}$$

and $u(k)$ is the system input which is the torque applied to the centre of the beam.

The feedback controller $\boldsymbol{M}(k)$ for this system (B.34) can be designed by using Ackermann's formula for assigned closed-loop poles figure (B.2).

$$\boldsymbol{x}(k+1) = \left[\boldsymbol{A}(\boldsymbol{x}(k)) - \boldsymbol{B}\boldsymbol{M}(k)\right]\boldsymbol{x}(k) + \boldsymbol{B}y_{ref}(k) \tag{B.36}$$

where $\boldsymbol{A}(\boldsymbol{x}(k))$ is the linear approximation to $\boldsymbol{f}(\boldsymbol{x}(k))$

$$\boldsymbol{A}(\boldsymbol{x}(k)) = \left[\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]_{\boldsymbol{x}=\boldsymbol{x}(k)}$$

and $y_{ref}(k)$ is the reference (desired) value of the output, $y(k)$.

So the closed-loop non-linear discrete-time control system (B.35) can be written as,

$$\boldsymbol{x}(k+1) = \boldsymbol{f}'(\boldsymbol{x}(k)) + \boldsymbol{B}y_{ref}(k) \tag{B.37}$$

$$y(k) = \boldsymbol{C}\boldsymbol{x}(k) \tag{B.38}$$

where $\boldsymbol{f}'(\boldsymbol{x}(k))$ is a new non-linear function incorporating the state feedback gain matrix $\boldsymbol{M}(k)$ i.e.

$$\boldsymbol{f}'(\boldsymbol{x}(k)) = \boldsymbol{A}(\boldsymbol{x}(k)) - \boldsymbol{B}\boldsymbol{M}(k) \tag{B.39}$$

The closed-loop non-linear discrete-time control system (B.37) and (B.38) will be used in chapters 2 and 4.

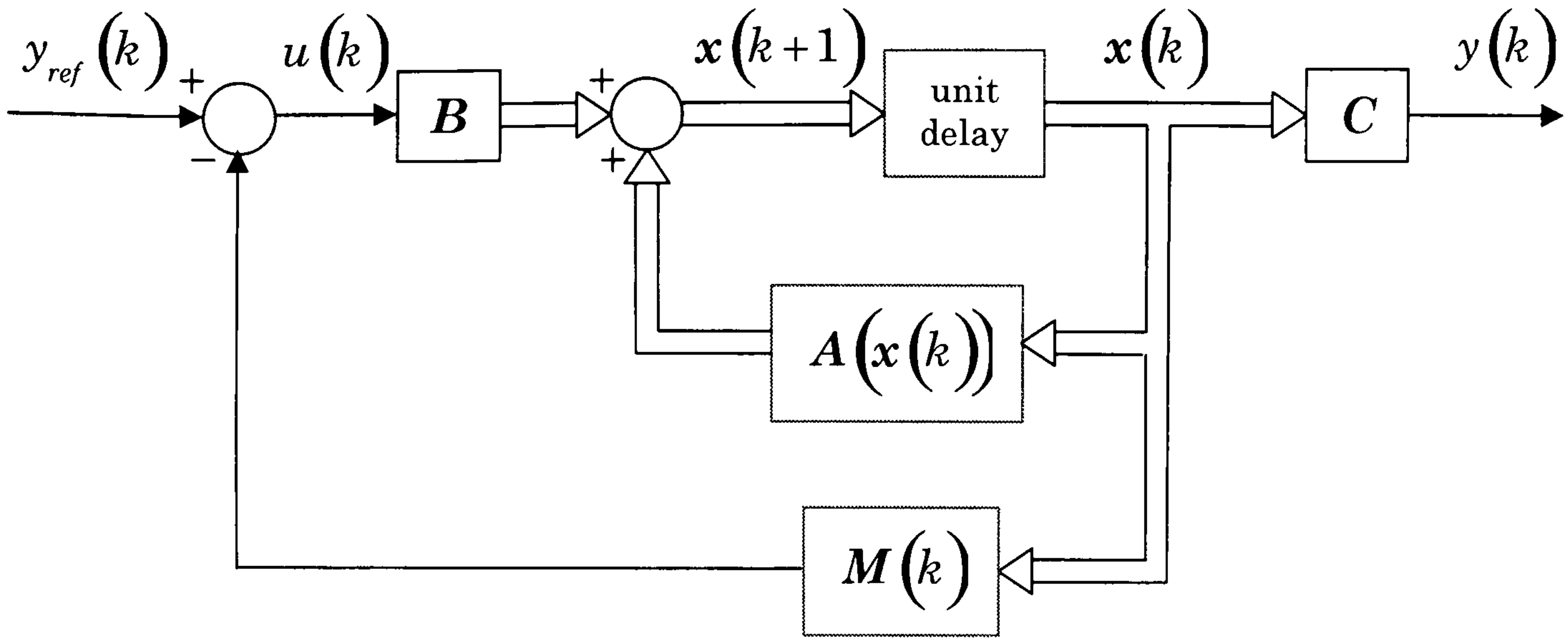

Figure B.2: Block diagram for the non-linear discrete-time ball and beam control system

Let,

$$r(k) = y_{ref}(k) \tag{B.40}$$

and

$$\boldsymbol{f}(\boldsymbol{x}(k)) = \boldsymbol{f}'(\boldsymbol{x}(k)) = \boldsymbol{A}(\boldsymbol{x}(k)) - \boldsymbol{BM}(k) \tag{B.41}$$

So the closed-loop non-linear discrete-time control system (B.37) and (B.38) can be written as,

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k)) + \boldsymbol{B}r(k) \tag{B.42}$$

$$y(k) = \boldsymbol{Cx}(k) \tag{B.43}$$

Appendix C

Matlab Codes

%C1

```
% program obstestcase1.m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                   %
% This program test the observer of case 1 by using pole            %
% placement method for a particular set of x0, z0, C and            %
% observer poles p. Ackermann's formula is used (function           %
% <acker> in matlab).                                               %
%                                                                   %
% Plots of state-, observer and observer error trajectories are     %
% generated at the end.                                             %
%                                                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
clf
k=input('number of iterations =  ')
p=input('desired observer poles')
x0=input('Initial state vector is  ')
x(1,:)=x0;

C=input('measurement vector is  ')
z0=input('Initial observer state vector is  ')
z(1,:)=z0;

Az0=[0.85 0.5;0.2*z0(1) 0.3];
L(1,:)=acker(Az0',C',p)

e=[];
e(1,:)=x0-z0;

for i=2:k
    y=C*x(i-1,:)';
    z(i,1)=0.85*z(i-1,1)+0.5*z(i-1,2)+L(i-1,1)*(y-C*z(i-1,:)');
    z(i,2)=0.1*z(i-1,1)^2+0.3*z(i-1,2)+L(i-1,2)*(y-C*z(i-1,:)')+0.05;

    x(i,1)=0.85*x(i-1,1)+0.5*x(i-1,2);
    x(i,2)=0.1*x(i-1,1)^2+0.3*x(i-1,2)+0.05;
    e(i,:)=x(i,:)-z(i,:);

    Az=[0.85 0.5;0.2*z(i,1) 0.3];
    L(i,:)=acker(Az',C',p);
    Acl=Az-L(i,:)'*C
end
```

```
figure(1)
clf
plot(x); grid
title('pole placement method')
xlabel('Time index'),
ylabel('State trajectory')


figure(2)
clf
plot(z);grid;
title('pole placement method')
xlabel('Time index'),
ylabel('Observer trajectory')

figure(3)
clf
plot(e);grid;
title('pole placement method')
xlabel('Time index'),
ylabel('Observer error trajectory')
```

%C2

```
% program statetestcase2.m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                 %
% This program test the controller of case 2, Non-linear         %
% Discrete-time Ball and beam system, by using pole              %
% placement method for a particular set of x0, poles p.          %
% Ackermann's formula is used (function <acker> in Matlab).      %
%                                                                 %
% Plots of state trajectories is generated at the end.           %
%                                                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
clf
k = input ('number of iterations =  ')
x0 = input ('Initial state vector is  ')
x(1,:) = x0;
p = input('State's closed-loop poles   ')

r = input('reference input  ')
T= input('sampling time T  =  ')

m=0.111;                    % Mass of the ball kg
Js=9.99e-6 ;                % ball's moment of inertia kgm^2
g=-9.8;                     % gravitational acceleration  m/s^2
R=0.015;                    % radius of the ball m
Ks=450;                     % Scaling factor

F=m*T/((Js/(R^2))+m);
H=m*g*T/((Js/(R^2))+m);
b  =  [0;0;0;T];

Ax0=[1 T 0 0  ; F*x(1,4)^2  1  -H  2*F*x(1,1)*x(1,4) ; 0 0 1 T ; 0 0 0 1 ];
M(1,:)=acker(Ax0,b,p);
Aclx=(Ax0-b*M(1,:));

for i=2:k
    x(i,1)=x(i-1,1)+T*x(i-1,2)-b(1)*M(i-1,:)*x(i-1,:)';
    x(i,2)=F*x(i-1,1)*x(i-1,4)^2+x(i-1,2)-H*x(i-1,3)-b(2)*M(i-1,:)*x(i-1,:)'
;
    x(i,3)=x(i-1,3)+T*x(i-1,4)-b(3)*M(i-1,:)*x(i-1,:)'  ;
    x(i,4)=x(i-1,4)-b(4)*M(i-1,:)*x(i-1,:)'+b(4)*Ks*r;
```

```
    Ax=[1 T 0 0  ; F*x(i,4)^2  1  -H  2*F*x(i,1)*x(i,4) ; 0 0 1 T ; 0 0 0 1 ];
    M(i,:)=acker(Ax,b,p);
    Aclx=Ax-b*M(i,:);
 end

subplot(2,2,1),plot(x(:,1)); grid
xlabel('Time index')
ylabel('x1 trajectory')

subplot(2,2,2),plot(x(:,2));grid
xlabel('Time index')
ylabel('x2 trajectory')

subplot(2,2,3), plot(x(:,3)); grid
xlabel('Time index'),
ylabel('x3 trajectory')

subplot(2,2,4),plot(x(:,4));grid
xlabel('Time index')
ylabel('x4 trajectory')
```

%C3

```
% program obstestcase2.m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                              %
% This program test the observer  of case 2, Non-linear        %
% Discrete-time Ball and beam system, by using pole            %
% placement method for a particular set of x0, z0, C and       %
% observer poles p. Ackermann's formula is used (function      %
% <acker> in matlab).                                          %
%                                                              %
% Plots of state-, observer, and observer error trajectories are %
% generated at the end.                                        %
%                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
clf
k = input ('number of iterations =  ')
x0 = input ('Initial state vector is  ')
x(1,:) = x0;
px = input('State’s closed-loop poles   ')

C = input('measurement vector is  ')
z0 = input('Initial observer state vector is  ')
z(1,:) = z0;
po = input('desired observer poles')

r = input('reference input  ')
T= input('sampling time T  =  ')

m=0.111;                    % Mass of the ball kg
Js=9.99e-6 ;                % ball's moment of inertia kgm^2
g=-9.8;                     % gravitational acceleration  m/s^2
R=0.015;                    % radius of the ball m
Ks=450;                     % Scaling factor

F=m*T/((Js/(R^2))+m);
H=m*g*T/((Js/(R^2))+m);
b  =  [0;0;0;T];

Az0= [1 T 0 0  ; F*z(1,4)^2  1  -H  2*F*z(1,1)*z(1,4) ; 0 0 1 T ; 0 0 0 1 ];
L(1,:)=acker(Az0',C',po);
Aclz=Az0-L(1,:)'*C;
```

```
Ax0=[1 T 0 0  ; F*x(1,4)^2  1  -H  2*F*x(1,1)*x(1,4) ; 0 0 1 T ; 0 0 0 1 ];
M(1,:)=acker(Ax0,b,px);
Aclx=(Ax0-b*M(1,:));


e=[];
e(1,:)=x0-z0;

for i=2:k
    y=C*x(i-1,:)';
    x(i,1)=x(i-1,1)+T*x(i-1,2)-b(1)*M(i-1,:)*x(i-1,:)';
    x(i,2)=F*x(i-1,1)*x(i-1,4)^2+x(i-1,2)-H*x(i-1,3)-b(2)*M(i-1,:)*x(i-1,:)'
;
    x(i,3)=x(i-1,3)+T*x(i-1,4)-b(3)*M(i-1,:)*x(i-1,:)'  ;
    x(i,4)=x(i-1,4)-b(4)*M(i-1,:)*x(i-1,:)'+b(4)*Ks*r;

    z(i,1)=z(i-1,1)+T*z(i-1,2)+L(i-1,1)*(y- C*z(i-1,:)');
    z(i,2)=F*z(i-1,1)*z(i-1,4)^2+z(i-1,2)-H*z(i-1,3)+L(i-1,2)*(y- C*z(i-
1,:)');
    z(i,3)=z(i-1,3)+T*z(i-1,4)+L(i-1,3)*(y - C*z(i-1,:)');
    z(i,4)=z(i-1,4)+L(i-1,4)*(y - C*z(i-1,:)')+b(4)*Ks*r;

    e(i,:)=x(i,:)-z(i,:);
    Az=[1 T 0 0  ; F*z(i,4)^2  1  -H  2*F*z(i,1)*z(i,4) ; 0 0 1 T ; 0 0 0 1 ];
    L(i,:)=acker(Az',C',po);
   Aclz=Az-L(i,:)'*C;

   Ax=[1 T 0 0  ; F*x(i,4)^2  1  -H  2*F*x(i,1)*x(i,4) ; 0 0 1 T ; 0 0 0 1 ];
   M(i,:)=acker(Ax,b,px);
   Aclx=Ax-b*M(i,:);

end

figure(1)
clf
subplot(2,2,1),plot(x(:,1)); grid
xlabel('Time index')
ylabel('x1 trajectory')

subplot(2,2,2),plot(x(:,2));grid
xlabel('Time index')
ylabel('x2 trajectory')

subplot(2,2,3), plot(x(:,3)); grid
xlabel('Time index'),
ylabel('x3 trajectory')

subplot(2,2,4),plot(x(:,4));grid
```

```
xlabel('Time index')
ylabel('x4 trajectory')

figure(2)
clf
subplot(2,2,1),plot(z(:,1)); grid
xlabel('Time index')
ylabel('z1 trajectory')

subplot(2,2,2),plot(z(:,2));grid
xlabel('Time index')
ylabel('z2 trajectory')

subplot(2,2,3), plot(z(:,3)); grid
xlabel('Time index'),
ylabel('z3 trajectory')

subplot(2,2,4),plot(z(:,4));grid
xlabel('Time index')
ylabel('z4 trajectory')

figure(3)
clf
subplot(2,2,1),plot(e(:,1)); grid
xlabel('Time index')
ylabel('e1 trajectory')

subplot(2,2,2),plot(e(:,2));grid
xlabel('Time index')
ylabel('e2 trajectory')

subplot(2,2,3), plot(e(:,3)); grid
xlabel('Time index'),
ylabel('e3 trajectory')

subplot(2,2,4),plot(e(:,4));grid
xlabel('Time index')
ylabel('e4 trajectory')
```

%C4

% program obslid1case1.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% This program test the observer of case 1 by using Sliding      %
% Mode for a particular set of x0, z0, C, observer poles p, Cs,  %
% auxiliary vector D, and boundary layer (phi).                      %
%                                                                              %
% Plots of state-, observer, observer error trajectories, and      %
% auxiliary control are generated at the end.                         %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
clf
k=input ('number of iterations = ')
x0=input('Initial state vector is  ')
x (1,:)=x0;

C= input ('measurement vector is  ')
z0= input('Initial observer state vector is  ')
z(1,:)=z0;
p= input('desired observer poles')

D=input ('auxiliary vector is ')

phi=input ('boundary layer is  ')
u=0.05;

Az0= [0.85 0.5;0.2*z0(1) 0.3];
L (1,:)=acker(Az0',C',p);
e= [];
v= [];
e (1,:)=x(1,:)-z(1,:);

for i=2:k
    y(i-1)=C*x (i-1,:)';
    s (i-1) =-(y(i-1)-C*z(i-1,:)');
    if norm (inv(C*D')*s(i-1))<=phi
        v(i-1)=inv(C*D')*s(i-1);
    else
        v(i-1)=phi*inv(C*D')*s(i-1)/norm(inv(C*D')*s(i-1));
    end
```

```
    x(i,1)=0.85*x(i-1,1)+0.5*x(i-1,2);
    x(i,2)=0.1*x(i-1,1)^2+0.3*x(i-1,2)+u;

    z(i,1)=0.85*z(i-1,1)+0.5*z(i-1,2)+L(i-1,1)*(y-C*z(i-1,:)')+D(1)*v(i-1);
    z(i,2)=0.1*z(i-1,1)^2+0.3*z(i-1,2)+L(i-1,2)*(y-C*z(i-1,:)')+D(2)*v(i-
    1)+u;

    e(i,:)=x(i,:)-z(i,:);

    Az=[0.85 0.5;0.2*z(i,1) 0.3];
    L(i,:)=acker(Az',C',p);
    Acl=Az-L(i,:)'*C;
end

figure (1)
clf
plot (x); grid
title ('Case 1 (Sliding Mode )')
xlabel ('Time index'),
ylabel ('State trajectory')

figure (2)
clf
plot (z);grid;
title ('Case 1 (Sliding Mode )')
xlabel ('Time index'),
ylabel ('Observer trajectory')

figure (3)
clf
plot (e); grid
title ('Case 1 (Sliding Mode )')
xlabel ('Time index'),
ylabel ('Observer error trajectory')

figure (4)
plot (v); grid
title ('Case 1 (Sliding Mode )')
xlabel ('Time index'),
ylabel ('Auxiliary control')
```

%C5

```
% program case2slid.m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                        %
% This program test the observer  of case 2, Non-linear   %
% Ball and beam system, by using Sliding Mode for a       %
% particular set of x0, z0,  C, observer poles p, auxiliary %
% vector D, and boundary layer (phi).                     %
%                                                        %
% Plots of state-, observer, observer error trajectories, and %
% auxiliary control are generated at the end.             %
%                                                        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
clf
k = input ('number of iterations =  ')
x0 = input ('Initial state vector is  ')
x(1,:) = x0;
px = input('State's closed-loop poles   ')

C = input('measurement vector is  ')
z0 = input('Initial observer state vector is  ')
z(1,:) = z0;
po = input('desired observer poles')

r = input('reference input  ')
T= input('sampling time T  =  ')

D = input(' auxiliary vector is  ')
phi = input(' boundary layer is  ')

m=0.111;              % Mass of the ball kg
Js=9.99e-6 ;          % ball's moment of inertia kgm^2
g=-9.8;               % gravitational acceleration  m/s^2
R=0.015;              % radius of the ball m
Ks=450;               % Scaling factor

F=m*T/((Js/(R^2))+m);
H=m*g*T/((Js/(R^2))+m);
b  =  [0;0;0;T];


Az0= [1 T 0 0  ; F*z(1,4)^2  1  -H  2*F*z(1.1)*z(1.4) ; 0 0 1 T ; 0 0 0 1 ];
```

```
L(1,:)=acker(Az0',C',po);
Aclz=Az0-L(1,:)'*C;

Ax0=[1 T 0 0  ; F*x(1,4)^2  1  -H  2*F*x(1,1)*x(1,4) ; 0 0 1 T ; 0 0 0 1 ];
M(1,:)=acker(Ax0,b,px);
Aclx=(Ax0-b*M(1,:));


e=[];
e(1,:)=x0-z0;
v=[];

for i=2:k
     y=C*x(i-1,:)';
     s(i-1)=-Cs*(x(i-1,:)-z(i-1,:))';
     if  norm(inv(Cs*D')*s(i-1))<=phi
        v(i-1)=inv(Cs*D')*s(i-1);
    else
       v(i-1)=phi*inv(Cs*D')*s(i-1)/norm(inv(Cs*D')*s(i-1));
    end

   x(i,1)=x(i-1,1)+T*x(i-1,2)-b(1)*M(i-1,:)*x(i-1,:)';
   x(i,2)=F*x(i-1,1)*x(i-1,4)^2+x(i-1,2)-H*x(i-1,3)-b(2)*M(i-1,:)*x(i-1,:)' ;
   x(i,3)=x(i-1,3)+T*x(i-1,4)-b(3)*M(i-1,:)*x(i-1,:)'  ;
   x(i,4)=x(i-1,4)-b(4)*M(i-1,:)*x(i-1,:)'+b(4)*Ks*r;

   z(i,1)=z(i-1,1)+T*z(i-1,2)+L(i-1,1)*(y- C*z(i-1,:) ') +D(1)*v(i-1);
   z(i,2)=F*z(i-1,1)*z(i-1,4)^2+z(i-1,2)-H*z(i-1,3)+L(i-1,2)*(y- C*z(i-1,:)
')+D(2)*v(i-1);
   z(i,3)=z(i-1,3)+T*z(i-1,4)+L(i-1,3)*(y- C*z(i-1,:) ')+D(3)*v(i-1);
   z(i,4)=z(i-1,4)+L(i-1,4)*(y- C*z(i-1,:) ')+b(4)*Ks*r+D(4)*v(i-1);

   e(i,:)=x(i,:)-z(i,:);

   Az=[1 T 0 0  ; F*z(i,4)^2  1  -H  2*F*z(i,1)*z(i,4) ; 0 0 1 T ; 0 0 0 1 ];
   L(i,:)=acker(Az',C',po);
   Aclz=Az-L(i,:)'*C;

   Ax=[1 T 0 0  ; F*x(i,4)^2  1  -H  2*F*x(i,1)*x(i,4) ; 0 0 1 T ; 0 0 0 1 ];
   M(i,:)=acker(Ax,b,px);
   Aclx=Ax-b*M(i,:);

end

figure(1)
clf
subplot(2,2,1),plot(x(:,1)); grid
xlabel('Time index')
```

```
ylabel('x1 trajectory')

subplot(2,2,2),plot(x(:,2));grid
xlabel('Time index')
ylabel('x2 trajectory')

subplot(2,2,3), plot(x(:,3)); grid
xlabel('Time index'),
ylabel('x3 trajectory')


subplot(2,2,4),plot(x(:,4));grid
xlabel('Time index')
ylabel('x4 trajectory')

figure(2)
clf
subplot(2,2,1),plot(z(:,1)); grid
xlabel('Time index')
ylabel('z1 trajectory')

subplot(2,2,2),plot(z(:,2));grid
xlabel('Time index')
ylabel('z2 trajectory')

subplot(2,2,3), plot(z(:,3)); grid
xlabel('Time index'),
ylabel('z3 trajectory')

subplot(2,2,4),plot(z(:,4));grid
xlabel('Time index')
ylabel('z4 trajectory')

figure(3)
clf
subplot(2,2,1),plot(e(:,1)); grid
xlabel('Time index')
ylabel('e1 trajectory')

subplot(2,2,2),plot(e(:,2));grid
xlabel('Time index')
ylabel('e2 trajectory')


subplot(2,2,3), plot(e(:,3)); grid
xlabel('Time index'),
ylabel('e3 trajectory')
```

```
subplot(2,2,4),plot(e(:,4));grid
xlabel('Time index')
ylabel('e4 trajectory')

figure(4)
clf
plot(v); grid
xlabel('Time index')
ylabel('v trajectory')
```