





This is to certify that the  
thesis entitled

ANALYSIS OF MULTIRATE DIGITAL CONTROL BASED ON  
HIGH GAIN OBSERVER AND SINGULAR PERTURBATION THEORY

presented by

Muhammad Yasin

has been accepted towards fulfillment  
of the requirements for

M.S. degree in Electrical Eng

Hassan Khalil  
Major professor

Date 1/23/1996

**PLACE IN RETURN BOX** to remove this checkout from your record.  
**TO AVOID FINES** return on or before date due.

DATE DUE	DATE DUE	DATE DUE
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

**MSU is An Affirmative Action/Equal Opportunity Institution**

c:\circ\datedue.pn3-p.1

**ANALYSIS OF MULTIRATE DIGITAL  
CONTROL BASED ON HIGH GAIN  
OBSERVER AND SINGULAR  
PERTURBATION THEORY**

**By**

**Muhammad Yasin**

**A THESIS**

**Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of**

**MASTER OF SCIENCE**

**Department of Electrical Engineering**

**1996**

## **ABSTRACT**

### **ANALYSIS OF MULTIRATE DIGITAL CONTROL BASED ON HIGH GAIN OBSERVER AND SINGULAR PERTURBATION THEORY**

By

Muhammad Yasin

A multirate control structure allows the designer to accommodate multiple information rates and implement required control computations within the finite computational capabilities of an onboard computer. The purpose of this study was to analyze the multirate scheme of a stable, linear, and time invariant system. The output feedback control is so designed that it uses a very high gain observer in its feedback path. The dynamics of the control are slow as compared to the dynamics of the observer. The behavior of the system is also analyzed using singular perturbation theory to see how the fast and slow modes behave in first cycle. Also, singular perturbation theory is used to show that the performance of the system under state feedback control can be recovered by using the output feedback control which uses a very high gain observer in its feedback path. The performance of the system is also studied by changing different parameters like, initial conditions, value of  $\epsilon$ , and sampling times. In computer simulations, it is seen that saturating the control helps in reducing the peaking phenomenon, that occurs due to different initial conditions.

**I dedicate this work to my biggest supporters in this world; my family; my  
parents, brothers, and sisters.**

## **ACKNOWLEDGMENTS**

First of all, I am thankful to Allah Almighty for providing me with the opportunity to undertake and complete this task. I am thankful to the Pakistan Army and the National University of Science and Technology (NUST), Pakistan, for sponsoring this study at Michigan State University.

I would like to express my deep thanks and sincere appreciation to all persons who have contributed by inspiration, counseling, and assistance in the completion of this study. The advice, professional guidance, and supervision of Dr. Hassan Khalil, have been invaluable in the completion of this study. He has been very kind in sparing his valuable time and making me understand through even minutest details of the subject. Also, I would like to extend my deepest thanks to Dr Shanblatt, Who helped me in the completion of my studies somehow or the other. I also want to convey my special thanks to Lt Col Rahat Khan, Muhammad Jamal Khattak, Abdul Mujeer Qureshi, graduate students at MSU, for helping me and providing their sincere advice during the course of my studies. I will take this opportunity to thank the Pakistan Student Association and Pakistani community for their care and concern.

My deepest love and thanks to my parents, who brought me up in a best manner, for their unconditional support from the day I was born, and for my brothers and sisters for their prayers that do not stop.

Above all, I deeply believe that my accomplishment and success are bestowed upon me by the Almighty to whom I am very thankful.

Muhammad Yasin



# TABLE OF CONTENTS

List of figures.....	viii
List of Symbols.....	ix

## CHAPTER 1 INTRODUCTION AND HISTORICAL BACKGROUND

1.1	Introduction.....	1
1.2	Historical Background.....	2
	1.2.1 Multirate Digital Control.....	2
	1.2.2 Singular Perturbation Theory.....	3
1.3	Objectives of The Thesis.....	4
1.4	Scheme of The Thesis.....	5

## CHAPTER 2 HIGH GAIN OBSERVER

2.1	Observers in General.....	6
	2.1.1 Introduction.....	6
	2.1.2 Formal Definition.....	7
	2.1.3 Mathematical Description of Observability.....	7
2.2	Observers in Linear Systems.....	10
	2.2.1 Task of Observer.....	10
	2.2.2 Observer Design.....	10
	2.2.3 Determination of System Matrices of The Observer.....	13
2.3	High Gain Observer.....	14
	2.3.1 Output Feedback Control.....	14
	2.3.2 Design of Observer Gain Matrix, $L$ .....	18
2.4	The Peaking Phenomenon.....	20
	2.4.1 Reasons of Peaking.....	20
	2.4.2 Methods to Reduce Peaking.....	22

## CHAPTER 3 ANALYSIS BASED ON SINGULAR PERTURBATION

3.1	Introduction.....	24
3.2	Derivation of Singularly Perturbed System.....	25
	3.2.1 Continuous Time Singularly Perturbed System.....	25

3.2.2	Discretization of Singularly Perturbed System.....	29
3.3	Multirate Implementation of Singularly Perturbed Difference Equation.....	32

## CHAPTER 4 SIMULATION RESULTS

4.1	Construction of The System.....	43
4.1.1	The Controlled System: A dc Servomotor.....	43
4.1.2	Poles of The System.....	45
4.1.3	Design of State Feedback Control.....	47
4.1.4	Construction of an Observer.....	49
4.1.5	Combined System - Single Rate Implementation.....	51
4.2	Simulations.....	53
4.2.1	State Feedback Control.....	53
4.2.2	Combined System - Single Rate Implementation.....	54
4.2.3	Multirate Implementation.....	56
4.2.4	Performance of Multirate Scheme Under State Feedback Control.....	58
4.2.5	Behavior of Multirate scheme in First Cycle.....	58
4.3	Effects of Changing Different Parameters.....	59
4.3.1	Effects of Changing The Initial Conditions.....	59
4.3.1a	Performance of Multirate Under State Feedback Control.....	59
4.3.1b	Performance of Multirate Under Output Feedback Control.....	60
4.3.2	Effects of Saturating The Control.....	61
4.3.3	Effects of Changing $\epsilon$ .....	62
4.3.4	Effects of Changing Sampling Times.....	65
4.4	Observations and Results.....	67
4.5	Recommendations.....	68
	CONCLUSION.....	69
	APPENDICES.....	70
	BIBLIOGRAPHY	

# LIST OF FIGURES

2.1 : System with an observable subsystem $S_{u1}$ and an unobservable subsystem $S_{u2}$ .....	8
2.2 : Two connected dynamic systems: $S_1$ the original system and $S_2$ the observer.....	11
2.3 : Block diagram showing implementation of equation (17).....	15

# LIST OF SYMBOLS

$A, B, C, D$	Continuous time state space matrices
$A_o, B_o, C_o, D_o$	Continuous time observer matrices
$AA, BB, CC, DD$	System matrices of combined system
$A_c$	Canonical form of $A$
$\bar{A}, \bar{B}$	Continuous time matrices of singularly perturbed system
$a, b, c, d$	Discrete time state space matrices
$a_o, b_o, c_o, d_o$	Discrete time observer matrices
$a_1, b_1, c_1, d_1$	Discrete time system matrices
$\Phi$	State transition matrix
$L$	Observer gain matrix
$K$	Gain matrix
$I$	Identity matrix
$D(\epsilon)$	Matrix dependent on $\epsilon$
$D^{-1}$	Inverse of $D$
$P$	Pole vector
$y$	Output vector of the state space system
$u$	Input vector of state space system

$x$	Plant state vector
$O$	Observer state vector
$M$	Observability grammian
$z$	Fast variable of singularly perturbed system, z-transform
$t$	Time
$t_s$	Settling time
$t_0$	Initial time
$t_f$	Final time
$T_c$	Sampling time of the control
$T_o$	Sampling time of observer
$\Theta_m$	Angular position of motor
$\omega$	Angular velocity
$\hat{x}$	Estimate of $x$
$\hat{\omega}$	estimate of $\omega$
$\varepsilon$	Parameter of high gain observer
$e$	estimation error
$v$	error term
$\delta, \delta_1$	Perturbation terms
$n$	Ratio of $T_c$ and $T_o$ , fast time scale
$k$	Slow time scale
$f_s$	Sampling frequency
$f_{obs}$	Observer frequency

$\lambda$	Eigenvalue
$J$	Moment of inertia
$R$	Resistance
$K_b$	Back emf constant
$K_T$	Torque constant
$\tau_m$	Mechanical time constant
$\xi$	Damping ratio

# CHAPTER 1

## INTRODUCTION AND HISTORICAL BACKGROUND

### 1.1 INTRODUCTION

Multiple sample-rate digital control systems are of prominent interest in current control research, development and applications. A multirate control structure allows the designer to accommodate multiple information rates and implement required control computations within the finite computational capabilities of an onboard computer. Multirate digital control is a significant area of research and applications that is motivated by practical implementation techniques. The motivation for multirate control has traditionally been in aerospace applications where guidance and control laws must be designed to accommodate multiple rates of sensor measurements and finite throughput capabilities of onboard computers. Modern aerospace vehicles and systems are described by high order dynamic models which typically include phenomena covering a wide range of characteristic frequencies and instrumentation measurements available at multiple rates. Multirate design techniques should soon find further utility in control application for highly distributed systems, such as, communication networks, and power plant / power distribution networks where the characteristic frequencies and time constants of a local station's dynamics may differ significantly from those of the network as a whole. The historical development of multirate digital control and singular perturbation theory is outlined in next section.

## 1.2 HISTORICAL BACKGROUND

### 1.2.1 Multirate Digital Control

The use of digital control, or more precisely, sample data control, originated in radar applications in world war II. Because the rotating antenna of a radar system illuminates a target intermittently, early radar aided tracking and fire control systems had to be designed to utilize data in sampled form. Methods of effective design of control systems using sampled data were under initial development during the later 1940's, and multirate system theory followed these efforts in the early 1950's. Initially, researchers developed multirate techniques as a method of evaluating more conventional types of controllers such as continuous systems and single rate sampled data systems. For example, one could study the inter sample behavior of a signal or output of a single rate control system by introducing a phantom sampler, i.e., a fictitious sampler that operates at a rate some integer ratio higher than that of the controller. A significant early contribution to this general method of analysis, known as frequency decomposition, was made by Shlansy and Ragazzini [1], who described the use of this technique in error sampled control system development. Shortly following the origin of the frequency decomposition technique, a similar frequency domain technique, known as switch decomposition, was developed.

The switch decomposition technique, attributed to Kranc [2], provided a means of representing a multirate control structure as an equivalent single rate controller, this representation accomplished, the controller could be designed and analyzed using single



rate technique. In the late 1960's Jury [3] showed an equivalence of the switch decomposition technique and the frequency decomposition technique.

Time domain methods of multirate stability analysis and design were initiated by Kalman and Bertram [4] with the publication of their state space stability analysis technique in 1959. Barry [5] published a paper in 1975 in which he described the design of multirate regulator and showed that its performance was superior to a single rate regulator having the slow sampling rate. During 1979-81, research at " The Analytic Sciences Corporation " ( TASC ) developed a new multirate control design technique based on an optimal estimation and control formulation. In parallel with the work at TASC, Amit and Powel [6], independently investigated a similar optimal control formulation, among other things, some practical considerations for implementing multirate control laws and a highly efficient method for solving periodic Riccati equation related to the design technique. Different design techniques for multirate digital control are discussed in [7].

### 1.2.2 Singular Perturbation Theory

Singularly perturbed systems, and more generally, multi time scale systems, often occur naturally due to the presence of small parasitic parameters, typically small time constants, or masses etc., multiplying time derivatives, or in more disguised form due to the presence of large feedback gains and weak coupling. While singular perturbation theory, a traditional tool of fluid dynamics and nonlinear mechanics, embraces a wide variety of dynamic phenomenon possessing slow and fast modes, its assimilation in control is recent and rapidly developing. The methods of singular perturbation for initial

and boundary value problem approximations and stability were already largely established in 1960's, when they first became a means for simplified computations of optimal trajectories. It was soon recognized that singular perturbations are present in most classical and modern control schemes based on reduced order models which disregard high frequency parasitics.

This led to the development of time scale methods for a variety of applications including state feedback, output feedback, filter and observer design. Singular perturbation methods also proved useful for the analysis of high gain feedback systems and interpretation of other model order reduction techniques. More recently they have been applied to modeling and control of dynamic networks and certain classes of large scale systems. This versatility of singular perturbation methods is due to their use of time scale properties which are common to both linear and nonlinear dynamic systems.

The first survey [8] of control theory applications of singular perturbations in 1976, included 130 references. The period of 1976-83 have witnessed an even faster growth of this research area both in theoretical depth and breadth of applications, as evidenced by surveys and books given in section A of the references of [9]. A brief overview of the 80 years of traditional singular perturbations by O'Malley (1982A), [10], lists 64 major references during that period. An analysis of singularly perturbed adaptive system appears in Ioannou and Kokotovic (1983A), [11].

### **1.3 OBJECTIVES OF THE THESIS**

I was assigned to accomplish the following objectives.

- a. Design a state feedback control using slow sampling period.

- b. Design of observer based output feedback control using the multirate scheme.
- c. Carry out analysis of multirate scheme using singular perturbation theory.
- d. Study the behavior of the system under output feedback control.
- e. Study the effect of changing different parameters in multirate implementation.
- f. Study the effect of saturation of the output feedback control.
- g. Study the behavior of slow and fast variables using singular perturbation theory.
- h. Study the behavior of slow and fast variables in one cycle while keeping the control constant.

#### 1.4 SCHEME OF THE THESIS

Chapter 1 deals with the general introduction and background of multirate digital control and singular perturbation theory. In chapter 2, design of high observer is discussed. Also the peaking phenomenon and its remedies are discussed in the same chapter. In chapter 3, analysis of multirate scheme, using singular perturbation theory, is done. First the system is brought into standard singularly perturbed form, and then the approach of [12] is followed to study the behavior of slow variables for one cycle. It is also shown that the output feedback control recovers the performance under state feedback control using singular perturbation theory. Chapter 5 deals with the simulations and their results, and finally the recommendations and conclusion is given.

## **CHAPTER 2**

# **HIGH GAIN OBSERVER**

### **2.1. OBSERVERS IN GENERAL**

#### **2.1.1 Introduction**

One of the objectives in control systems design is the achievement of suitable eigenvalue locations in order to ensure satisfactory dynamic performance. In modern control theory, linear state feedback provides an appropriate compensation technique to meet this objective under the assumption that all state variables can be used in forming feedback signals. Unfortunately, this assumption is not always valid in practice. A well known approach to overcome this difficulty is to generate the feedback control law via an estimate of the state vector. The estimation is performed using an asymptotic state estimator, called an observer, which employs only the available directly measurable input and output signals. Hence, the problem of designing controllers for systems with incomplete state measurements is equivalent to constructing observer based controllers.

The idea of observability and controllability was introduced by R.E Kalman in the 1950's. Proceeding from a figurative description, it can be said that a system is observable if it is possible to observe the states by using the output of the system, that is, to look inside the system from the output. That means, all the states must have a connection to the output. A system is said to be completely observable if all of its states are observable. It is some times advantageous to have an observable system, especially

when the system to be controlled has unstable states. Of course, an observable system is innocuous if the system is asymptotically stable.

The idea of observability is explained with the help of figure 2.1. The figure describes a system with input  $u$ , output  $y$  and two subsystems with the states  $x_1$  and  $x_2$ . The subsystem  $S_{u_1}$  is observable because the state  $x_1$  has a connection to the output of the system, whereas the subsystem  $S_{u_2}$  is not observable because the state  $x_2$  is not connected to the output of the system.

### 2.1.2 Formal Definition

The formal definition of observability is as given below:-

“ A linear system is said to be observable at time  $t_0$  if  $x(t_0)$  can be determined from the output function  $y(t_0, \tau)$  within a finite time interval  $t_0 < \tau < t_f$ . If this is true for all  $t$  and  $x(t)$ , then the system is said to be completely observable.” Referring to the formal definition of observability, it is clear that the observability depends on the output  $y$  and the state  $x$ .

An equivalent definition of observability states that:-

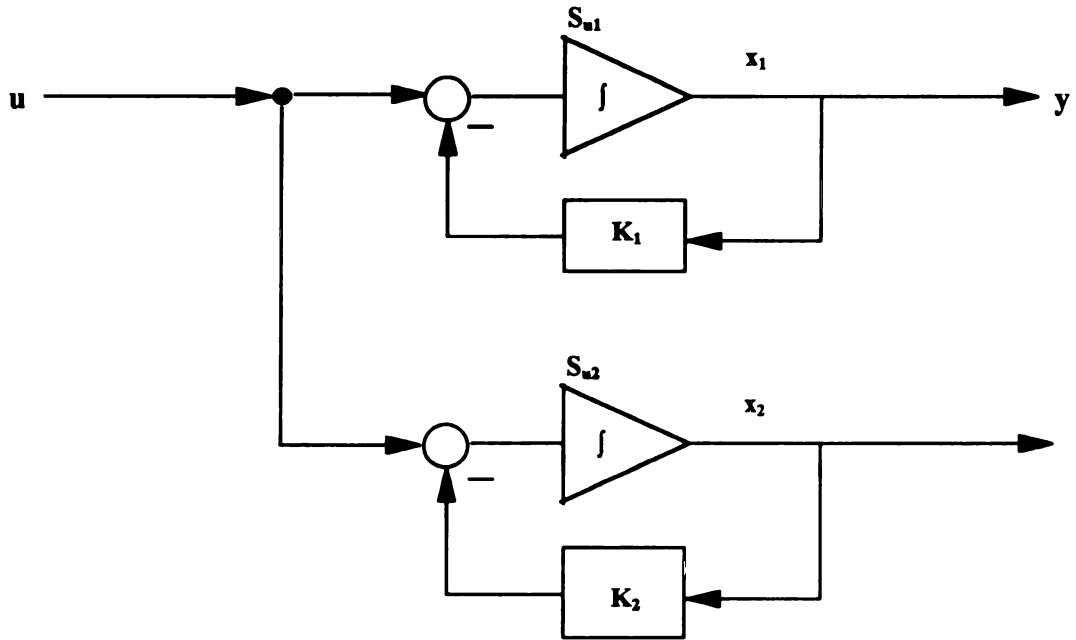
“A system is observable on interval  $[t_0, t_f]$  if any  $x(t_0)$  is uniquely determined by the corresponding response  $y(t)$  for  $t \in [t_0, t_f]$ , where  $t_f > t_0$ .”

### 2.1.3 Mathematical Description of Observability

The observability phenomenon can be described mathematically by using the theorem given below. The theorem states that, given the system:-

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1)$$

$$y(t) = Cx(t) \quad (2)$$



**Figure 2.1:** System with an observable subsystem  $S_{u1}$  and an unobservable subsystem  $S_{u2}$ .

The system (1) - (2) is observable on interval  $[t_0, t_f]$ , if and only if, the observability grammian

$$M(t_0, t_f) = \int_{t_0}^{t_f} \Phi^T(t, t_0) C^T(t) C(t) \Phi(t, t_0) dt$$

is invertible ( i.e. nonsingular ). Also  $M(t_0, t_f)$  is symmetric and positive semi definite.

Now consider the output given by :-

$$y(t) = C \Phi(t, t_0) x(t_0)$$

and assume that the matrix  $M(t_0, t_f)$  is noninvertible ( i.e. singular ). If this is true, there will exist an initial state  $x(t_0)$  different from zero which can produce a zero output for interval of any length. Therefore, the matrix  $C \Phi(t, t_0)$  is not of full rank  $n$ , the order of the system. Therefore, all the states are not transferred to the output and the system is not observable.

In most cases, the calculations of the observability grammian is very complicated. Therefore, as an alternate method, we can use a simpler algebraic theorem to test the observability of the system. The theorem states that a linear time invariant system

$$\dot{x}(t) = A x(t)$$

with the output given by:-

$$y(t) = C x(t)$$

is observable, if and only if, the rank  $r(N)$  of the observability test matrix, given by:-

$$N = [A^T \quad A^T C^T \quad \dots \quad (A^{n-1})^T C^T]$$

is equal to  $n$ , the order of the system.

## 2.2. OBSERVERS IN LINEAR SYSTEMS

### 2.2.1 Task of Observer

The basic task of an observer is to reconstruct one or more missing states of a given system by using the system output, provided that the system is completely observable. The observer has to estimate the state such that the error between the actual state  $x(t)$  and the estimated state  $\hat{x}(t)$ , that is

$$e(t) = x(t) - \hat{x}(t) \quad (3)$$

tends to zero. To build up an observer, a second dynamic system  $S_2$  is connected to the output of the original system  $S_1$ . Such a system is shown in figure 2.2. The system  $S_2$  has almost the same structure as the original system  $S_1$ . Both  $S_1$  and  $S_2$  are assumed to be linear and time invariant. The output of the system  $S_1$  is  $y$ , which is the input of the second system  $S_2$ ,  $u$  is the input to the system  $S_1$ . The output of the system  $S_2$  is called the estimated state  $\hat{x}$ .

### 2.2.2 Observer Design

Consider following linear, time invariant system  $S_1$ , derived from (1)-(2)

$$\dot{x}(t) = Ax(t) \quad (4)$$

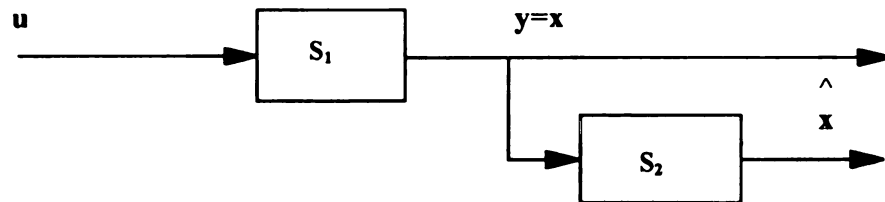
when system  $S_2$  is added to  $S_1$ , it becomes

$$\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + LCx(t) \quad (5)$$

with  $y(t) = Cx(t)$  and  $L$  is an  $n \times 1$  gain matrix. Now suppose the transformation  $T$ , which satisfies

$$LC = TA - \hat{A}T \quad (6)$$





**Figure 2.2:** Two connected dynamic systems:  $S_1$  the original system and  $S_2$  the observer.

Also if  $\hat{x}(0) = Tx(0)$  then  $\hat{x}(t) = Tx(t)$ . The general solution of (5) is given by:-

$$\hat{x}(t) = Tx(t) + e^{\hat{A}t} (\hat{x}(0) - Tx(0)) \quad (7)$$

To prove this solution subtract (4), after premultiplying by T, from (5), we get

$$\hat{\dot{x}}(t) - T\dot{x}(t) = \hat{A}\hat{x}(t) + LCx(t) - TAx(t) \quad (8)$$

now using (6) in (8), we get

$$\hat{\dot{x}}(t) - T\dot{x}(t) = \hat{A}(\hat{x}(t) - Tx(t)) \quad (9)$$

The solution of differential equation (9) is given with (7).

Now for a forced system, apply the transformation T on the input u. So we get

$$\hat{\dot{x}}(t) = \hat{A}\hat{x}(t) + LCx(t) + T\hat{B}u(t) \quad (10)$$

The system  $S_2$  given by equation (5) or (10) is called as an observer and the matrix L is called the observer gain. The poles of the observer should be chosen on the following basis:-

- a. They should be chosen arbitrarily.
- b. They are independent of the poles of the original system.
- c. They should be chosen so as to be in the left half of the s-plane, or equivalently, inside the unit circle of z-plane, to meet the stability conditions.
- d. They should be placed at such locations so that the observer system is faster than the control.

To get an observer with the same order as that of the original system, we let the transformation T be the identity matrix I. Therefore, (10) becomes:-

$$\hat{\dot{x}}(t) = \hat{A}\hat{x}(t) + LCx(t) + \hat{B}u(t) \quad (11)$$

### 2.2.3 Determination of System Matrices of The Observer

To determine the system matrices  $A$ ,  $B$  and the gain matrix  $L$ , we consider the error  $e(t)$  defined by (3). Taking derivative of (3), we get

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) \quad (12)$$

Substituting (1) and (11) in (12), we get

$$\dot{e}(t) = Ax(t) + Bu(t) - [\hat{A}\hat{x}(t) + LCx(t) + \hat{B}u(t)]$$

or

$$\dot{e}(t) = Ax(t) + Bu(t) - \hat{A}\hat{x}(t) - LCx(t) - \hat{B}u(t)$$

Now substituting  $\hat{x}(t)$  from (3) in above, we get

$$\dot{e}(t) = Ax(t) + Bu(t) - \hat{A}(x(t) - e(t)) - LCx(t) - \hat{B}u(t)$$

or

$$\dot{e}(t) = \hat{A}e(t) + (A - \hat{A} - LC)x(t) + (B - \hat{B})u(t) \quad (13)$$

When the error  $e(t)$  tends to zero, the coefficient matrices of (13) must also be zero, therefore,

$$A - \hat{A} - LC = 0$$

which gives

$$\hat{A} = A - LC \quad (14)$$

and  $B - \hat{B} = 0$

which gives

$$\hat{B} = B \quad (15)$$

It is seen that  $\hat{B}$  is the matrix B of the original system. The observer gain L can be chosen arbitrarily to get the desired eigenvalues of the system. The matrix L can be determined by control design methods, e.g., pole placement.

Now substituting (14) and (15) in (11), we get the observer equation depending on the system matrices of the original system.

$$\dot{\hat{x}}(t) = (A - LC)\hat{x}(t) + LCx(t) + Bu(t) \quad (16)$$

or

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \quad (17)$$

The block diagram showing implementation of (17) is shown in figure 2.3.

## 2.3. HIGH GAIN OBSERVER

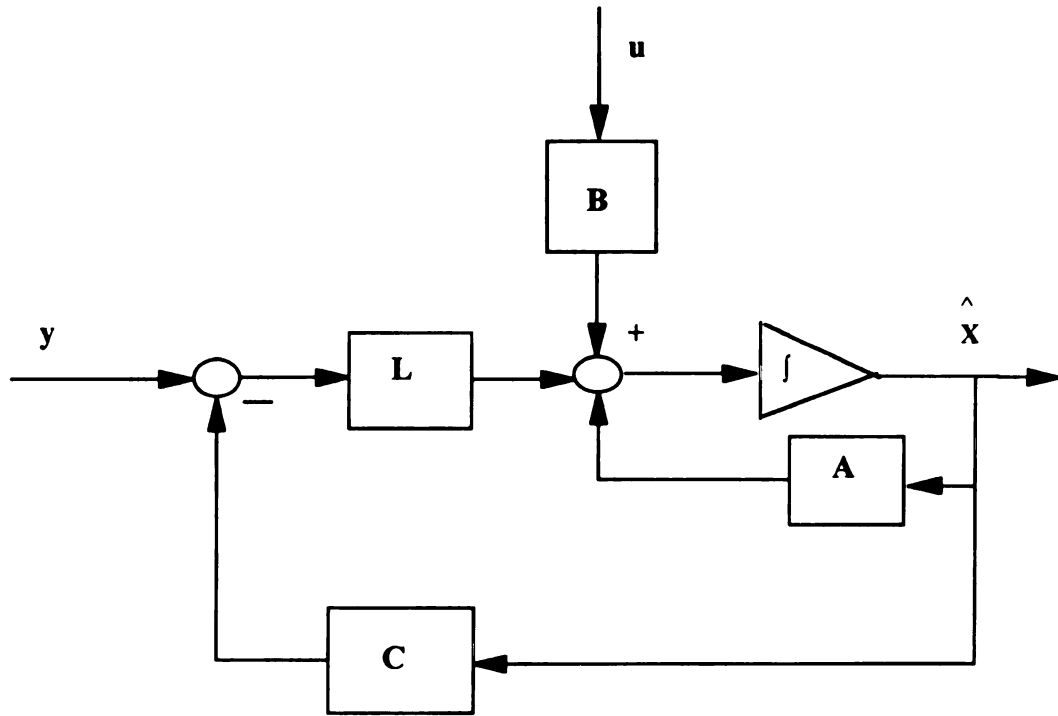
### 2.3.1 Output Feedback Control

Consider a fully linearizable system, refer to [13], represented by:-

$$\dot{x}(t) = Ax(t) + B[f_1(x) + G_1(x)u(t)] \quad (18)$$

$$y(t) = Cx(t) \quad (19)$$

In equation (18), the input term is a function of the state x and the input u(t) with a coefficient matrix  $G_1(x)$ . The basic structural property of a linearizable system is the fact that it can be transformed into coordinates where the nonlinearity of the system ( or a part of it ), satisfies the matching conditions. This permits the cancellation of the nonlinearity by the state feedback. However, exact cancellation is almost impossible due to parameter uncertainty, model simplification etc. The fact that the nonlinearity satisfies the matching condition implies that error in modeling the nonlinearity will also satisfy the matching



**Figure 2.3:** Block diagram showing implementation of equation (17).

condition. Therefore, robust control techniques like variable structural control, min-max control, and high gain feedback can be used to robustify the linearization.

To implement controllers using output feedback, we need to estimate the unmeasured state variables. The state estimators will have to be designed to preserve the robustness achieved with state feedback. The output feedback controller is designed in two steps. In first step, a state feedback control is designed to achieve the control task, and in the second step, an observer is designed to recover the performance achieved with state feedback. To get a description of the system which can be treated well, it is assumed that the system can be linearized perfectly at a chosen state depending upon system requirements. The linearizing state feedback control:-

$$u = G_1^{-1}(x)[-f_1(x) + v] \quad (20)$$

cancels the nonlinearity of the system. This cancellation, however requires perfect knowledge of the nonlinearities  $f_1(\cdot)$  and  $G_1(\cdot)$ . Since perfect knowledge is almost impossible, it is more realistic to assume that we only know the models  $f_{1n}(\cdot)$  and  $G_{1n}(\cdot)$  of the actual nonlinearities  $f_1(\cdot)$  and  $G_1(\cdot)$ . To consider the nonlinearity, a perturbation term  $\delta(x, u)$ , depending on the state  $x$  and the input  $u$  is added. Therefore, the system (18) can be represented as:-

$$\dot{x}(t) = Ax(t) + B[-f_{1n}(x) + G_{1n}(x)u(t)] + B\delta(x, u) \quad (21)$$

The key feature of this perturbation is the fact that it satisfies the matching conditions, that is, it enters the state equation exactly at the same point where the control enters.

Suppose now that a state feedback control has been designed as:-

$$u(t) = G_{1n}^{-1}(x)[-f_{1n}(x) + \Phi(t, x)] \quad (22)$$

where  $\Phi(t, x)$  is linear or nonlinear control that ensures robust stabilization or tracking in the presence of the uncertainty  $\delta$ . Now substituting (22) in (21), we get

$$\dot{x}(t) = Ax(t) + B\Phi(t, x) + B\delta(x, u) \quad (23)$$

Next task is to estimate the state  $x$ . Consider the full order observer given by:-

$$\dot{\hat{x}}(t) = A\hat{x}(t) + B[f_{1n}(\hat{x}) + G_{1n}(\hat{x})u(t)] + L(y(t) - C\hat{x}(t)) \quad (24)$$

which reflects the nominal nonlinear model of the state equation together with an error driving term. Now using the estimate  $\hat{x}$ , (22) can be written as:-

$$u(t) = G_{1n}^{-1}(\hat{x})[-f_{1n}(\hat{x}) + \Phi(t, \hat{x})] \quad (25)$$

Substituting (25) in (24), we get

$$\dot{\hat{x}}(t) = A\hat{x}(t) + B\Phi(t, \hat{x}) + L(y(t) - C\hat{x}(t)) \quad (26)$$

Defining the estimation error as:-

$$e(t) = x(t) - \hat{x}(t) \quad (27)$$

gives

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) \quad (28)$$

Substituting (23) and (26) in (28), we get

$$\dot{e}(t) = Ax(t) - A\hat{x}(t) + B\Phi(t, x) - B\Phi(t, \hat{x}) + B\delta(x, u) - L(y - C\hat{x}(t)) \quad (29)$$

We see that the term  $B\Phi(t, \hat{x})$  and  $B\Phi(t, x)$  do not cancel each other perfectly like they do in linear control. Therefore, we define another term  $\delta_1$  dependent on time, the state of the original system, and the state of the observer as:-

$$\delta_1(t, x, \hat{x}) = \Phi(t, x) - \Phi(t, \hat{x}) \quad (30)$$

Thus, using (30) and (27), (29) can be written as:-

$$\dot{e}(t) = (A - LC)e(t) + B[\delta(x, u) + \delta_1(t, x, \hat{x})] \quad (31)$$

where  $\delta(x, u)$  is the perturbation term which represents uncertainty in state equation, and  $\delta_1(t, x, \hat{x})$  is an error induced by the presence of the nonlinearities.

### 2.3.2 Design of Observer Gain Matrix, L

We need to design the observer gain matrix L to stabilize the matrix (A-LC) while rejecting the effect of the perturbation term. This is ideally achieved if we could design L such that the transfer function of the observer given by:-

$$H(s) = (sI - A + LC)^{-1} B \quad (32)$$

is identically zero. Since this ideal situation can be achieved under very restrictive conditions, we try to design L as a function of some parameter  $\varepsilon$ , ( $\varepsilon > 0$ ), such that H(s) approaches zero asymptotically as  $\varepsilon$  tends to zero, point wise in s.

The challenging task in nonlinear problems is performing closed loop stability analysis with such observer design to ensure that the closed loop system with the observer recovers robustness properties achieved with state feedback. In [14] and [15] Esfandiari and Khalil use singular perturbation to design the observer gain L. The singular perturbation analysis will be discussed in chapter 3. By representing the closed loop system as a singularly perturbed one, they show the recovery of robustness properties in the presence of nonvanishing perturbations. In particular, they show that the output feedback controller recovers ultimate boundedness achieved under state feedback control. Saberi and Sannuti [16] use singular perturbations and show recovery of robust



global stabilization of the origin. Petersen and Holot [17] use an  $H_\infty$  approach to design L. Gibbens and Fu [18] also use the idea of Petersen and Holot to show the recovery of robust tracking in the presence of nonvanishing perturbations. Teel and Praly [19] use singular perturbation approach to achieve semi global stabilization.

A common feature of all these observer designs, when the relative degree of the system is higher than one, is that they are asymptotic approximators of the output derivatives. Now consider a second order system having system matrices, A as  $2 \times 2$ , and C as  $1 \times 2$ , and let the observer gain for this system be given by:-

$$L = [a \quad b]^T$$

For (A-LC) to be Hurwitz, we must choose a and b to be positive. The transfer function  $H(s)$ , given by (32), approaches zero asymptotically, if and only if, its  $H_\infty$  norm does so. By calculating the  $H_\infty$  norm of  $H(s)$ , it can be seen that a necessary condition for making the norm arbitrarily small is that  $b \gg a \gg 1$ . The various observer designs reviewed earlier arrive at this condition using different procedures. The singular perturbation approach of [14] and [16] starts from the outset by choosing the observer gain L as :-

$$L = \frac{1}{\varepsilon^2} \begin{bmatrix} \alpha\varepsilon \\ 1 \end{bmatrix}$$

for some  $\alpha > 0$ .

All the procedures used achieved the condition  $b \gg a \gg 1$  by choosing  $a = O(1/\varepsilon)$  and  $b = O(1/\varepsilon^2)$  for a small positive parameter  $\varepsilon$ . To see that these observers are approximate differentiators, refer back to the observer equation (26), and notice that for

sufficiently large  $a$  and  $b/a$ , the transfer function from  $\Phi$  to  $\hat{x}$  is almost zero while the transfer function from  $y$  to  $\hat{x}$  is given by:-

$$(sI - A + LC)^{-1} L = \frac{1}{\frac{1}{b}s^2 + \frac{a}{b}s + 1} \begin{bmatrix} \frac{a}{b}s + 1 \\ s \end{bmatrix} \approx \begin{bmatrix} 1 \\ s \end{bmatrix}$$

where the approximation is valid on any compact frequency interval for sufficiently large  $a$  and  $b/a$ . Keeping in view the above discussion, for our thesis we design the observer gain for second order system as:-

$$L = \begin{bmatrix} 1 & 1 \\ \epsilon & \epsilon^2 \end{bmatrix}^T$$

We use this form of observer gain matrix, which can produce very large numbers. That is why, it is called as high gain observer.

## 2.4. THE PEAKING PHENOMENON

### 2.4.1 Reasons of peaking

There are some intrinsic difficulties with controllers which use high gain observers. First, such controllers amplify measurement noise. Second, they produce large overshoots in the transient response of the closed loop system, which we refer to as peaking phenomenon. These large overshoots, in the transient response, can even destabilize the system completely.

The observer gain  $L$  is designed to bring the transfer function  $H(s)$ , given by (32), as close to zero as possible. This requires the observer gain to be very large. The high gain  $L$  assigns the eigenvalues of  $(A-LC)$  far to the left in the left half of the  $s$ -plane. Typically,  $L$  depends on a small parameter  $\epsilon$ , such that as  $\epsilon$  tends to zero, the eigenvalues

approach infinity. The corresponding modes in the state transition matrix  $\exp[(A-LC)t]$  take the form  $\exp(-\alpha t/\epsilon)$  for some  $\alpha > 0$ . Hence they decay to zero with in an  $O(\epsilon)$  time interval. For a relative degree one system, the coefficients of these exponential modes will be bounded as  $L$  approaches infinity because the initial states are totally bounded.

For the systems of relative degree higher than one, the coefficients of the exponential modes are necessarily driven to infinity as  $L$  increases towards infinity. Let us use the second order system, considered previously, to illustrate this point. We already know that reducing the transfer function  $H(s)$  to zero requires designing the observer gain  $L = [a \ b]^T$  such that  $b \gg a \gg 1$ . By calculating the state transition matrix, it can be seen that the 2-1 element is given by:

$$\frac{-b}{\sqrt{b - \frac{a^2}{4}}} e^{-a/2 t} \sin\left(\sqrt{b - \frac{a^2}{4}} t\right)$$

when  $4b > a^2$ , and

$$\frac{-b}{\sqrt{a^2 - 4b}} \left[ \exp\left\{-\left(\frac{a + \sqrt{a^2 - 4b}}{2}\right)t\right\} - \exp\left\{-\left(\frac{a - \sqrt{a^2 - 4b}}{2}\right)t\right\} \right]$$

when  $4b < a^2$ .

The magnitude of the coefficient of the exponential mode is greater than  $\sqrt{b}$  in the first case and  $b/a$  in the second case. Thus, as we increase  $a$  and  $b/a$ , we drive this coefficient towards infinity. This causes an impulsive behavior in the transient response of the estimation error. It is to be mentioned here that, the peaking phenomenon is

associated with the process of designing  $L$  to reduce  $H(s)$  to zero, irrespective of the particular design procedure used to design  $L$ .

#### 2.4.2 Methods to Reduce Peaking

Following methods help in reducing the peaking phenomenon.

- a. In the presence of peaking, we have a bigger incentive to use singular perturbation theory. This process involves scaling some of the variables by  $\varepsilon$  dependent factor. Once the system is brought into the standard singularly perturbed form, then it is known from singular perturbation theory that, there is no peaking when the initial states are of the order of  $O(1)$ . Peaking is induced only by initial conditions which are of the order of  $O(\varepsilon^{-\beta})$  with  $\beta > 0$ . Thus the mechanism, through which the peaking is induced, has been represented as the value of the initial conditions of some of the state variables. Knowing the scaling factor used to arrive at the singularly perturbed form, we can clearly see how the initial conditions are scaled. Through this process we can see how the region of attraction of the origin will shrink as  $\varepsilon$  decreases. We also see that any destabilization effect, as  $\varepsilon$  tends to zero, will be induced by initial conditions of the fast variables which take the form  $O(\varepsilon^{-\beta})$ . Although the results of singular perturbation theory are limited to cases when the initial states are of order  $O(1)$ , realizing the role of the initial conditions of the fast variable points in the direction which lead to the breakthrough result on globally bounded control.
- b. We can drive the controller and/or the observer into saturation so that it does not exceed a certain value. But the limits of saturation will have to be defined very

carefully. The bounded output feedback control recovers the performance of the bounded state feedback controller as  $\epsilon$  tends to zero.

- c. As a second possibility, we can use multirate sampling where we use, in digital control, different sampling times for the control and the observer.

## CHAPTER 3

# ANALYSIS BASED ON SINGULAR PERTURBATION

### 3.1. INTRODUCTION

The singular perturbation approach to the analysis design of continuous time systems possessing multiple time scales has matured over the past 25 years. Recently, there has been interest in extending singular perturbation ideas to multiple time scale discrete time systems. Singularly perturbed systems, and more generally, multi time scale systems often occur naturally due to the presence of small parasitic parameters, typically small time constants, or masses etc., multiplying time derivatives, or in more disguised form, due to the presence of large feedback gains and weak coupling.

Various mathematical forms have been used to represent a two time scale discrete time systems. Discretization and sampled data control of systems described by singularly perturbed differential equations result in singularly perturbed difference equations. Singularly perturbed difference equations might also arise naturally in inherently discrete time systems. In this chapter, the two time scale nature of the system is exploited to represent the system in the singularly perturbed form. At the end of the chapter, it confirms the intuitive idea that the slow variables can be measured at a rate slower than that of fast variables without degrading the performance of the system.

## 3.2. DERIVATION OF SINGULARLY PERTURBED SYSTEM

### 3.2.1 Continuous Time Singularly Perturbed System

To motivate the form of the singularly perturbed difference equation, we start by considering its continuous time analog first. Consider the continuous, linear, time invariant system defined as:-

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1)$$

$$y(t) = Cx(t) \quad (2)$$

and the state feedback control is given by:-

$$u(t) = -Kx(t) + Nr(t) \quad (3)$$

where A, B, C are system matrices in canonical form. A is  $n \times n$ , B is  $n \times m$ , and C is  $p \times n$ . K is the gain matrix. Now consider the observer equation

$$\dot{\hat{x}}(t) = A\hat{x}(t) + L(y(t) - C\hat{x}(t)) \quad (4)$$

and the output feedback control, given by:-

$$u(t) = -K\hat{x}(t) + Nr(t) \quad (5)$$

where  $\hat{x}(t)$  is the estimate of  $x(t)$ . Now consider the error difference between the actual and the estimated states, given by:-

$$e(t) = x(t) - \hat{x}(t) \quad (6)$$

which gives

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) \quad (7)$$

Using (1) and (4) in (7), we get

$$\dot{e}(t) = Ax(t) + Bu(t) - A\hat{x}(t) - L(y(t) - C\hat{x}(t))$$

Now using (2) in above equation, we get

$$\dot{e}(t) = Ax(t) + Bu(t) - A\hat{x}(t) - LCx(t) + LC\hat{x}(t)$$

or

$$\dot{e}(t) = (A - LC)(x(t) - \hat{x}(t)) + Bu(t)$$

Now using (6) in above equation, we get

$$\dot{e}(t) = (A - LC)e(t) + Bu(t) \quad (8)$$

The system matrix A can be written as:-

$$A = A_c + B\Delta \quad (9)$$

where  $A_c$  is in canonical form of  $n \times n$ , and  $\Delta$  is some arbitrary matrix.

Therefore

$$A - LC = (A_c + B\Delta) - LC$$

using above equation in (8), we get

$$\dot{e}(t) = (A_c + B\Delta - LC)e(t) + Bu(t) \quad (10)$$

Now choose  $D(\epsilon)$  such that  $D^{-1}(\epsilon)$  exists and  $\epsilon > 0$  is a small parameter. In general

$$D(\epsilon) = \begin{bmatrix} \epsilon^{n-1} & 0 & \dots & \dots & \dots & 0 \\ 0 & \ddots & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \epsilon^2 & & \vdots \\ \vdots & & & & \epsilon & 0 \\ 0 & \dots & \dots & \dots & 0 & 1 \end{bmatrix}$$



$$D^{-1}(\varepsilon) = \begin{bmatrix} \frac{1}{\varepsilon^{n-1}} & 0 & \dots & \dots & \dots & 0 \\ 0 & \ddots & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \frac{1}{\varepsilon^2} & & \vdots \\ \vdots & & & & \frac{1}{\varepsilon} & 0 \\ 0 & \dots & \dots & \dots & 0 & 1 \end{bmatrix}$$

where  $n$  is the order of the system. We define another variable as

$$z(t) = D^{-1}(\varepsilon)e(t) \quad (11)$$

Taking derivative of (11), we get

$$\dot{z}(t) = D^{-1}(\varepsilon)\dot{e}(t) \quad (12)$$

Substituting (10) in (12), we get

$$\dot{z}(t) = D^{-1}(\varepsilon)[(A_c + B\Delta - LC)e(t) + Bu(t)]$$

Using (11) in above equation, we get

$$\dot{z}(t) = D^{-1}(\varepsilon)[(A_c + B\Delta - LC)D(\varepsilon)z(t) + Bu(t)]$$

or

$$\dot{z}(t) = D^{-1}(\varepsilon)(A_c - LC)D(\varepsilon)z(t) + D^{-1}(\varepsilon)B\Delta D(\varepsilon)z(t) + D^{-1}(\varepsilon)Bu(t) \quad (13)$$

Also choose the observer gain matrix  $L$ , such that  $(A-LC)$  is Hurwitz. ( The design procedure for  $L$  has already been discussed in chapter 2 ). In general,

$$L = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \dots & \alpha_n \\ \varepsilon & \varepsilon^2 & & & \varepsilon^n \end{bmatrix}^T$$

The general canonical form of the system matrices is:-

$$A_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ \vdots & \vdots & \vdots & & & 1 \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 1 \end{bmatrix}$$

$$C = [1 \ 0 \ \cdots \ \cdots \ \cdots \ 0]$$

Therefore

$$A_c - LC = \begin{bmatrix} \frac{-\alpha_1}{\varepsilon} & 1 & 0 & \cdots & \cdots & 0 \\ \frac{-\alpha_2}{\varepsilon^2} & 0 & 1 & & & \vdots \\ \vdots & \vdots & & \ddots & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ \vdots & \vdots & & & & 1 \\ \frac{-\alpha_n}{\varepsilon^n} & 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

$$D^{-1}(\varepsilon)(A_c - LC)D(\varepsilon) = \frac{1}{\varepsilon} \begin{bmatrix} -\alpha_1 & 1 & 0 & \cdots & \cdots & 0 \\ -\alpha_2 & 0 & 1 & & & \vdots \\ \vdots & \vdots & & \ddots & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ \vdots & \vdots & & & & 1 \\ -\alpha_n & 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix} = \frac{1}{\varepsilon} A_0 \quad (14)$$

Similarly we can prove that:-

$$D^{-1}(\varepsilon)B = \begin{bmatrix} \frac{1}{\varepsilon^{n-1}} & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & & & & \vdots \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \frac{1}{\varepsilon^2} & & \vdots \\ \vdots & & & & \frac{1}{\varepsilon} & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 1 \end{bmatrix} = B \quad (15)$$

Using the results (14) and (15) in (13), we get

$$\dot{z}(t) = \frac{1}{\varepsilon} A_0 z(t) + B \Delta D(\varepsilon) z(t) + B u(t)$$

or

$$\varepsilon \dot{z}(t) = (A_0 + \varepsilon B \Delta D(\varepsilon)) z(t) + \varepsilon B u(t) \quad (16)$$

Now writing (1) and (16) in matrix form, we get

$$\begin{bmatrix} \dot{x}(t) \\ \varepsilon \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A_0 + \varepsilon B \Delta D(\varepsilon) \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ \varepsilon B \end{bmatrix} u(t) \quad (17)$$

or

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & \frac{A_0 + \varepsilon B \Delta D(\varepsilon)}{\varepsilon} \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u(t) \quad (18)$$

Equations (17) and (18) are the standard form of continuous time singularly perturbed system.  $x(t)$  is the slow variable and  $z(t)$  is the fast variable. Now we define output feedback control that consists of both slow and fast variables. Using (6) in (5), we get

$$u(t) = -K(x(t) - e(t)) + N r(t) \quad (19)$$

substituting (11) in (19), we get

$$u(t) = -K(x(t) - D(\varepsilon)z(t)) + N r(t) \quad (20)$$

### 3.2.2 Discretization of Singularly Perturbed System

We can write equation (18) in the form:-

$$\dot{\bar{x}}(t) = \bar{A} \bar{x}(t) + \bar{B} u(t) \quad (21)$$

where

$$\bar{A} = \begin{bmatrix} A & 0 \\ 0 & \frac{A_0 + \varepsilon B \Delta D(\varepsilon)}{\varepsilon} \end{bmatrix} \quad \bar{B} = \begin{bmatrix} B \\ B \end{bmatrix}$$

and

$$\dot{\bar{x}}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} \quad \bar{x}(t) = \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}$$

Now we discretize (21) at a fast sampling period  $\varepsilon T$ , to get

$$\bar{x}(n+1) = A_d \bar{x}(n) + B_d u(n) \quad (22)$$

where  $n$  is the fast time scale and  $A_d$  and  $B_d$  are discrete time matrices . In general

$$A_d = e^{AT}$$

$$B_d = \int_0^T e^{A^i t} dt B$$

Therefore, using above equations, we get

$$A_d = \exp(\bar{A}\varepsilon T) = \begin{bmatrix} \exp(A\varepsilon T) & 0 \\ 0 & \exp((A_0 + \varepsilon B\Delta D(\varepsilon))T) \end{bmatrix}$$

Now taking these matrix elements one by one:-

$$\exp(A\varepsilon T) = I + \varepsilon \left[ AT + \varepsilon \frac{(AT)^2}{2!} + \dots \right]$$

or we can write this as:-

$$\exp(A\varepsilon T) = I + \varepsilon A_1(\varepsilon) \quad (23)$$

From (23), we have

$$A_1(0) = AT$$

Similarly, we get

$$\exp((A_0 + \varepsilon B\Delta D(\varepsilon))T) = e^{A_0 T} + O(\varepsilon) \quad (24)$$

Therefore,

$$A_d = \begin{bmatrix} I + \varepsilon A_1(\varepsilon) & 0 \\ 0 & e^{A_0 T} + O(\varepsilon) \end{bmatrix}$$

Now consider  $B_d$ , given by:-

$$B_d = \begin{bmatrix} \int_0^{\varepsilon T} e^{A' t} dt B \\ \int_0^{\varepsilon T} \exp\left(\left(\frac{A_0 + \varepsilon B\Delta D(\varepsilon)}{\varepsilon}\right)t\right) dt B \end{bmatrix}$$

Consider the first matrix element of  $B_d$

$$\int_0^{\varepsilon T} e^{A' t} dt B = \varepsilon \int_0^T e^{A \varepsilon s} ds B = \varepsilon B_1 \quad (25)$$

where

$$B_1 = \int_0^T e^{A \varepsilon s} ds B \quad (25A)$$

Similarly

$$\int_0^{\varepsilon T} \exp\left(\left(\frac{A_0 + \varepsilon B\Delta D(\varepsilon)}{\varepsilon}\right)t\right) dt B = \varepsilon \int_0^T \exp((A_0 + \varepsilon B\Delta D(\varepsilon))s) ds B = \varepsilon B_2 \quad (26)$$

where

$$B_2 = \int_0^T \exp((A_0 + \varepsilon B\Delta D(\varepsilon))s) ds B \quad (26A)$$

Therefore  $B_d$  is written as:-

$$B_d = \begin{bmatrix} \varepsilon B_1 \\ \varepsilon B_2 \end{bmatrix}$$

Substituting  $A_d$  and  $B_d$  in (22), we get

$$\begin{bmatrix} x(n+1) \\ z(n+1) \end{bmatrix} = \begin{bmatrix} I + \varepsilon A_1(\varepsilon) & 0 \\ 0 & e^{A_0 T} + O(\varepsilon) \end{bmatrix} \begin{bmatrix} x(n) \\ z(n) \end{bmatrix} + \begin{bmatrix} \varepsilon B_1 \\ \varepsilon B_2 \end{bmatrix} u(n) \quad (27)$$

or we can write it as

$$x(n+1) = (I + \varepsilon A_1(\varepsilon))x(n) + \varepsilon B_1 u(n) \quad (28)$$

$$z(n+1) = (e^{A_0 T} + O(\varepsilon))z(n) + \varepsilon B_2 u(n) \quad (29)$$

where

$$u(n) = -K(x(n) - D(\varepsilon)z(n)) + Nr(n) \quad (30)$$

Equation (27) is the discrete time version of (18) sampled at a fast sampling period  $\varepsilon T$ , with the output feedback control given by (30).  $x(n)$  is called slow variable and  $z(n)$  is fast variable. Equation (27) is expressed in fast time scale  $n$  and is called as singularly perturbed difference equation. Equation (27) shows that the eigenvalues of the system take the form:-

$$\begin{aligned} \lambda_i(\varepsilon) &= 1 + \varepsilon p_i(\varepsilon) & i &= 1, 2, \dots, m_1 \\ \lambda_{j+m_1}(\varepsilon) &= q_j(\varepsilon) & j &= 1, 2, \dots, m_2 \end{aligned}$$

where  $p_i(\varepsilon)$  and  $q_j(\varepsilon)$  are continuous at  $\varepsilon = 0$  and  $q_j(0)$  are the eigenvalues of  $e^{A_0 T}$ .

Since  $\text{Re} \lambda(A_0) < 0$ , then  $|\lambda(e^{A_0 T})| < 1$ , the fast modes in (27) are asymptotically stable and reach steady state.

### 3.3. Multirate Implementation of Singularly Perturbed Difference Equation

It is possible to obtain a description of the system in a slow time scale if the fast modes are asymptotically stable, see [12]. A slow time scale  $k$  is defined by the

relationship  $n = k/\varepsilon$ , where  $k = 0, 1, 2, \dots$ . Now consider the interval  $\frac{k}{\varepsilon} \leq n < \frac{k+1}{\varepsilon}$  such that the control (30) is kept constant over this interval. Then for this interval, the control law is written as:-

$$u\left(\frac{k}{\varepsilon}\right) = -K \left[ x\left(\frac{k}{\varepsilon}\right) - D(\varepsilon)z\left(\frac{k}{\varepsilon}\right) \right] + Nr\left(\frac{k}{\varepsilon}\right) \quad (31)$$

and

$$u(n) = u\left(\frac{k}{\varepsilon}\right) \quad \frac{k}{\varepsilon} \leq n < \frac{k+1}{\varepsilon} \quad (32)$$

Now substituting (31) in (28) and (29), respectively, results in a closed loop system, described for the given interval, given by

$$x(n+1) = (I + \varepsilon A_1(\varepsilon))x(n) + \varepsilon B_1 \left[ -K \left\{ x\left(\frac{k}{\varepsilon}\right) - D(\varepsilon)z\left(\frac{k}{\varepsilon}\right) \right\} + Nr\left(\frac{k}{\varepsilon}\right) \right] \quad (33)$$

Now starting from  $\frac{k}{\varepsilon}$ , we calculate the state at  $\frac{k+1}{\varepsilon}$ , therefore (33) gives

$$\begin{aligned} x\left(\frac{k+1}{\varepsilon}\right) &= (I + \varepsilon A_1(\varepsilon))^{\varepsilon} x\left(\frac{k}{\varepsilon}\right) + \sum_{j=\frac{k}{\varepsilon}}^{\frac{k+1}{\varepsilon}-1} (I + \varepsilon A_1(\varepsilon))^{\varepsilon^{k+1-1-j}} \varepsilon B_1 \\ &\quad \left[ -K \left\{ x\left(\frac{k}{\varepsilon}\right) - D(\varepsilon)z\left(\frac{k}{\varepsilon}\right) \right\} + Nr\left(\frac{k}{\varepsilon}\right) \right] \\ &= \left[ (I + \varepsilon A_1(\varepsilon))^{\varepsilon} - \sum_{j=\frac{k}{\varepsilon}}^{\frac{k+1}{\varepsilon}-1} (I + \varepsilon A_1(\varepsilon))^{\varepsilon^{k+1-1-j}} \varepsilon B_1 K \right] x\left(\frac{k}{\varepsilon}\right) + \\ &\quad \left[ \sum_{j=\frac{k}{\varepsilon}}^{\frac{k+1}{\varepsilon}-1} (I + \varepsilon A_1(\varepsilon))^{\varepsilon^{k+1-1-j}} \varepsilon B_1 K D(\varepsilon) \right] z\left(\frac{k}{\varepsilon}\right) + \end{aligned}$$

$$\left[ \sum_{j=\frac{k}{\varepsilon}}^{\frac{k+1}{\varepsilon}-1} (I + \varepsilon A_1(\varepsilon))^{\frac{k+1}{\varepsilon}-1-j} \varepsilon B_1 N \right] r\left(\frac{k}{\varepsilon}\right)$$

$$x\left(\frac{k+1}{\varepsilon}\right) = I + II + III \quad (34)$$

Now solving for I, II, III separately. Rewriting I, we get

$$I = \left[ (I + \varepsilon A_1(\varepsilon))^{\frac{k+1}{\varepsilon}} + \varepsilon \sum_{j=\frac{k}{\varepsilon}}^{\frac{k+1}{\varepsilon}-1} \left( (I + \varepsilon A_1(\varepsilon))^{\frac{k+1}{\varepsilon}-1-j} (-B_1 K) \right) \right] x\left(\frac{k}{\varepsilon}\right) \quad (35)$$

The expressions given below, which have been used in [12],

$$(I + \varepsilon A(\varepsilon))^{\frac{1}{\varepsilon}} = e^{A(0)} + O(\varepsilon) \quad (36)$$

$$\varepsilon \sum_{j=0}^{\frac{1}{\varepsilon}-1} (I + \varepsilon A(\varepsilon))^{\frac{1}{\varepsilon}-1-j} = \int_0^1 e^{A(0)t} dt + O(\varepsilon) \quad (37)$$

hold for any matrix A that is analytic at  $\varepsilon = 0$ . Now using (36) and (37), equation (35) gives:-

$$I = \left[ \left( e^{A_1(0)} + O(\varepsilon) \right) + \left\{ \int_0^1 e^{A_1(0)t} dt + O(\varepsilon) \right\} B_0 \right] x\left(\frac{k}{\varepsilon}\right)$$

where  $B_0 = -BK$

Now using the fact that  $A_1(0) = AT$ , we have  $e^{A_1(0)} = e^{AT}$  and therefore, we have

$$\int_0^1 e^{A_1(0)t} dt = \int_0^1 e^{AT} dt$$

and after making change of variables, we get

$$\int_0^1 e^{A_1(0)t} dt = \frac{1}{T} \int_0^T e^{At} dt$$



Therefor

or

Now t

or

where

Now c

or

where

Now

Therefore, I can now be written as:-

$$I = \left[ \left( e^{AT} + O(\varepsilon) \right) + \left\{ \frac{1}{T} \int_0^T e^{At} dt + O(\varepsilon) \right\} B_0 \right] x\left(\frac{k}{\varepsilon}\right)$$

or

$$I = \left[ e^{AT} + \frac{1}{T} \int_0^T e^{At} dt B_0 + O(\varepsilon) \right] x\left(\frac{k}{\varepsilon}\right) \quad (38)$$

Now take the second part of (34), i.e., II, and using the same procedure as before, we get

$$II = \left[ \left\{ \frac{1}{T} \int_0^T e^{At} dt + O(\varepsilon) \right\} B_1 KD(\varepsilon) \right] z\left(\frac{k}{\varepsilon}\right)$$

or

$$II = \left[ \frac{1}{T} \int_0^T e^{At} dt B_f + O(\varepsilon) \right] z\left(\frac{k}{\varepsilon}\right) \quad (39)$$

where  $B_f = B_1 KD(\varepsilon)$

Now consider III of (34) and use (37) as before; we get

$$III = \left[ \left\{ \frac{1}{T} \int_0^T e^{At} dt + O(\varepsilon) \right\} B_1 N \right] r\left(\frac{k}{\varepsilon}\right)$$

or

$$III = \left[ \frac{1}{T} \int_0^T e^{At} dt B_i + O(\varepsilon) \right] r\left(\frac{k}{\varepsilon}\right) \quad (40)$$

where  $B_i = B_1 N$

Now substituting (38), (39), and (40) in (34), we get

$$x\left(\frac{k+1}{\varepsilon}\right) = \left[ e^{AT} + \frac{1}{T} \int_0^T e^{At} dt B_0 + O(\varepsilon) \right] x\left(\frac{k}{\varepsilon}\right) + \left[ \frac{1}{T} \int_0^T e^{At} dt B_f + O(\varepsilon) \right] z\left(\frac{k}{\varepsilon}\right) + \left[ \frac{1}{T} \int_0^T e^{At} dt B_i + O(\varepsilon) \right] r\left(\frac{k}{\varepsilon}\right) \quad (41)$$

Taking into account the fast variable  $z(n)$ , substituting (31) in (29) results in closed loop system:-

$$z(n+1) = \left( e^{A_0 T} + O(\varepsilon) \right) z(n) + \varepsilon B_2 \left[ -K \left\{ x\left(\frac{k}{\varepsilon}\right) - D(\varepsilon) z\left(\frac{k}{\varepsilon}\right) \right\} + N r\left(\frac{k}{\varepsilon}\right) \right] \quad (42)$$

As before, starting from  $\frac{k}{\varepsilon}$ , we calculate the state at  $\frac{k+1}{\varepsilon}$ .

$$\begin{aligned} z\left(\frac{k+1}{\varepsilon}\right) &= \left( e^{A_0 T} + O(\varepsilon) \right)^{1\varepsilon} z\left(\frac{k}{\varepsilon}\right) + \sum_{j=\frac{k}{\varepsilon}}^{k+1-\frac{1}{\varepsilon}} \left( e^{A_0 T} + O(\varepsilon) \right)^{\varepsilon^{k+1-1-j}} \varepsilon B_2 \\ &\quad \left[ -K \left\{ x\left(\frac{k}{\varepsilon}\right) - D(\varepsilon) z\left(\frac{k}{\varepsilon}\right) \right\} + N r\left(\frac{k}{\varepsilon}\right) \right] \\ &= \left[ \left( e^{A_0 T} + O(\varepsilon) \right)^{1\varepsilon} + \sum_{j=\frac{k}{\varepsilon}}^{k+1-\frac{1}{\varepsilon}} \left( e^{A_0 T} + O(\varepsilon) \right)^{\varepsilon^{k+1-1-j}} \varepsilon B_2 K D(\varepsilon) \right] z\left(\frac{k}{\varepsilon}\right) + \\ &\quad \left[ \sum_{j=\frac{k}{\varepsilon}}^{k+1-\frac{1}{\varepsilon}} \left( e^{A_0 T} + O(\varepsilon) \right)^{\varepsilon^{k+1-1-j}} \varepsilon (-B_2 K) \right] x\left(\frac{k}{\varepsilon}\right) + \\ &\quad \left[ \sum_{j=\frac{k}{\varepsilon}}^{k+1-\frac{1}{\varepsilon}} \left( e^{A_0 T} + O(\varepsilon) \right)^{\varepsilon^{k+1-1-j}} \varepsilon (B_2 N) \right] r\left(\frac{k}{\varepsilon}\right) \\ z\left(\frac{k+1}{\varepsilon}\right) &= I + II + III \quad (43) \end{aligned}$$

Now consider I, II, III separately.

$$I = \left[ \left( e^{A_0 T} + O(\varepsilon) \right)^{1\varepsilon} + \sum_{j=\frac{k}{\varepsilon}}^{k+1-\frac{1}{\varepsilon}} \left( e^{A_0 T} + O(\varepsilon) \right)^{\varepsilon^{k+1-1-j}} \varepsilon B_2 K D(\varepsilon) \right] z\left(\frac{k}{\varepsilon}\right)$$

Since  $\text{Re } \lambda(A_0) < 0$ ,  $\exp(A_0 T)$  is bounded, and the fact that:-

$$\frac{1}{\varepsilon} \left\| \left( e^{A_0 T} + O(\varepsilon) \right)^{1/\varepsilon} \right\| \leq \frac{1}{\varepsilon} \gamma^{1/\varepsilon} \leq k_1 \quad 0 < \varepsilon < \varepsilon_1 \quad 0 < \gamma < 1$$

is established, therefore,

$$\left( e^{A_0 T} + O(\varepsilon) \right)^{1/\varepsilon} = O(\varepsilon)$$

Now for the second part of I, we let  $e^{A_0 T} + O(\varepsilon) = A_2 \quad i = j - \frac{k}{\varepsilon}$ , so we get

$$\sum_{j=\frac{k}{\varepsilon}}^{\frac{k+1}{\varepsilon}-1} \left( e^{A_0 T} + O(\varepsilon) \right)^{k+1-j} = \sum_{i=0}^{\frac{k}{\varepsilon}-1} (A_2)^{1-i} = O(1) \quad \varepsilon \rightarrow 0$$

Therefore,

$$\sum_{j=\frac{k}{\varepsilon}}^{\frac{k+1}{\varepsilon}-1} \left( e^{A_0 T} + O(\varepsilon) \right)^{k+1-j} \varepsilon B_2 K D(\varepsilon) = O(\varepsilon)$$

So for I, we get

$$I = \left( O(\varepsilon) + O(\varepsilon) \right) z\left(\frac{k}{\varepsilon}\right) = O(\varepsilon) z\left(\frac{k}{\varepsilon}\right) \quad (44)$$

Now consider part II of equation (43), i.e.,

$$II = \left[ \sum_{j=\frac{k}{\varepsilon}}^{\frac{k+1}{\varepsilon}-1} \left( e^{A_0 T} + O(\varepsilon) \right)^{k+1-j} \varepsilon (-B_2 K) \right] x\left(\frac{k}{\varepsilon}\right)$$

and using the same argument as above, we get

$$II = O(\varepsilon) x\left(\frac{k}{\varepsilon}\right) \quad (45)$$

and proceeding in similar way for III, we get

$$III = O(\varepsilon)r\left(\frac{k}{\varepsilon}\right) \quad (46)$$

Now substituting (44), (45), and (46) in (43), we get

$$z\left(\frac{k+1}{\varepsilon}\right) = O(\varepsilon)z\left(\frac{k}{\varepsilon}\right) + O(\varepsilon)x\left(\frac{k}{\varepsilon}\right) + O(\varepsilon)r\left(\frac{k}{\varepsilon}\right) \quad (47)$$

Now writing (41) and (47) in matrix form, we get

$$\begin{bmatrix} x\left(\frac{k+1}{\varepsilon}\right) \\ z\left(\frac{k+1}{\varepsilon}\right) \end{bmatrix} = \begin{bmatrix} e^{AT} + \frac{1}{T} \int_0^T e^{At} dt B_0 + O(\varepsilon) & \frac{1}{T} \int_0^T e^{At} dt B_f + O(\varepsilon) \\ O(\varepsilon) & O(\varepsilon) \end{bmatrix} \begin{bmatrix} x\left(\frac{k}{\varepsilon}\right) \\ z\left(\frac{k}{\varepsilon}\right) \end{bmatrix} + \begin{bmatrix} \frac{1}{T} \int_0^T e^{At} dt B_i + O(\varepsilon) \\ O(\varepsilon) \end{bmatrix} r\left(\frac{k}{\varepsilon}\right)$$

or

$$= \begin{bmatrix} A_{11} + O(\varepsilon) & A_{12} + O(\varepsilon) \\ O(\varepsilon) & O(\varepsilon) \end{bmatrix} \begin{bmatrix} x\left(\frac{k}{\varepsilon}\right) \\ z\left(\frac{k}{\varepsilon}\right) \end{bmatrix} + \begin{bmatrix} B_{11} + O(\varepsilon) \\ O(\varepsilon) \end{bmatrix} r\left(\frac{k}{\varepsilon}\right) \quad (48)$$

where

$$A_{11} = e^{AT} + \frac{1}{T} \int_0^T e^{At} dt B_0 \quad (49)$$

$$A_{12} = \frac{1}{T} \int_0^T e^{At} dt B_f \quad (50)$$

$$B_{11} = \frac{1}{T} \int_0^T e^{At} dt B_i \quad (51)$$

For sufficiently small  $\varepsilon$ , equation (48) can be written as slow sub system, given by:-

$$\begin{bmatrix} \bar{x}(\frac{k+1}{\varepsilon}) \\ \bar{z}(\frac{k+1}{\varepsilon}) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(\frac{k}{\varepsilon}) \\ z(\frac{k}{\varepsilon}) \end{bmatrix} + \begin{bmatrix} B_{11} \\ 0 \end{bmatrix} r(\frac{k}{\varepsilon}) \quad (52)$$

with same initial conditions.

Sampling (18) at fast sampling period  $\varepsilon T$  resulted in (48) where the control is kept constant over the interval  $\frac{k}{\varepsilon} \leq n < \frac{k+1}{\varepsilon}$ . However dropping the  $O(\varepsilon)$  term from (48) results in (52) which is almost same system, with some error, that can be obtained by sampling the original system (1)-(3) at slower sampling period  $T$ . That is, the singularly perturbed system (18) sampled at fast sampling period  $\varepsilon T$  is  $O(\varepsilon)+v$  close to the slow system (1)-(3) sampled at slow sampling period  $T$ , i.e.,

$$x(\frac{k}{\varepsilon}) - \bar{x}(\frac{k}{\varepsilon}) = O(\varepsilon) + v \quad k > 0 \quad (53)$$

where  $v$  is the error term given by  $A_{12}$ .

For zero initial conditions, there is no error and the difference between the two models, the model under state feedback and the model under output feedback, is exactly  $O(\varepsilon)$ . However, for different initial conditions, the error  $v$  is induced which is given by  $A_{12}$ . To see as to why this error is induced, consider the first cycle, i.e., the interval from 0 to  $1/\varepsilon$ . Due to different initial conditions, the two models start from different states having more difference, and by the time they reach  $1/\varepsilon$  from 0, the difference between the two states will have decreased, thus decreasing the error. Now, the starting conditions for the two models will almost be same but not exactly the same. In fact, after one cycle the difference between the two models will be  $O(T_c)$ , i.e.,

$$x\left(\frac{1}{\varepsilon}\right)_{state\ feedback} - x\left(\frac{1}{\varepsilon}\right)_{output\ feedback} = O(T_c) \quad (54)$$

This is an error which will be carried along after one cycle. However, this error can be reduced significantly by having small values of  $T_c$  and  $\varepsilon$ . Therefore, for smaller values of  $T_c$  and  $\varepsilon$ , the error induced will be very small and we shall have closeness between the two models. To see the closeness between the two models we proceed as following:-

From (48), we have

$$x\left(\frac{k+1}{\varepsilon}\right) = (A_{11} + O(\varepsilon))x\left(\frac{k}{\varepsilon}\right) + (A_{12} + O(\varepsilon))z\left(\frac{k}{\varepsilon}\right) + (B_{11} + O(\varepsilon))r\left(\frac{k}{\varepsilon}\right) \quad (55)$$

and  $z\left(\frac{k+1}{\varepsilon}\right)$  is given by (47). Now tracing back the different matrices we get

$$A_{11} = e^{AT} + \frac{1}{T} \int_0^T e^{At} dt \left( - \int_0^T e^{A\varepsilon s} ds \right) BK$$

$$A_{12} = \frac{1}{T} \int_0^T e^{At} dt B_1 KD(\varepsilon)$$

and

$$B_{11} = \frac{1}{T} \int_0^T e^{At} dt \left( \int_0^T e^{A\varepsilon s} ds \right) BN$$

Substituting the expressions of  $A_{11}$ ,  $A_{12}$  and  $B_{11}$  in (55) and rearranging, we get

$$x\left(\frac{k+1}{\varepsilon}\right) = e^{AT} x\left(\frac{k}{\varepsilon}\right) + \frac{1}{T} \int_0^T e^{At} dt \left[ - \int_0^T e^{A\varepsilon s} ds BK x\left(\frac{k}{\varepsilon}\right) + \int_0^T e^{A\varepsilon s} ds BN r\left(\frac{k}{\varepsilon}\right) \right] + \left[ \frac{1}{T} \int_0^T e^{At} dt B_1 KD(\varepsilon) \right] z\left(\frac{k}{\varepsilon}\right) + O(\varepsilon) \left[ x\left(\frac{k}{\varepsilon}\right) + z\left(\frac{k}{\varepsilon}\right) + r\left(\frac{k}{\varepsilon}\right) \right] \quad (56)$$

Now for small values of  $\varepsilon$ , i.e., for  $\varepsilon$  tending to zero, we drop out the  $O(\varepsilon)$  term, and

$$\int_0^T e^{A\varepsilon s} ds = T \quad D(\varepsilon) = D(0)$$

Therefore, (56) now becomes:-

$$x\left(\frac{k+1}{\varepsilon}\right) = e^{AT} x\left(\frac{k}{\varepsilon}\right) + \int_0^T e^{At} dt B \left[ -Kx\left(\frac{k}{\varepsilon}\right) + Nr\left(\frac{k}{\varepsilon}\right) \right] + \left[ \frac{1}{T} \int_0^T e^{At} dt B_1 KD(0) \right] z\left(\frac{k}{\varepsilon}\right) \quad (57)$$

Now using (3) for  $t = k / \varepsilon$  in (57), we get

$$x\left(\frac{k+1}{\varepsilon}\right) = e^{AT} x\left(\frac{k}{\varepsilon}\right) + \int_0^T e^{At} dt Bu\left(\frac{k}{\varepsilon}\right) + \left[ \frac{1}{T} \int_0^T e^{At} dt B_1 KD(0) \right] z\left(\frac{k}{\varepsilon}\right) \quad (58)$$

Looking at (58), the first two terms represent the same model that is obtained by sampling (1) at slower sampling period  $T$ . The third term in (58) represent the error  $v$  present due to fast variable having different initial conditions. Now since  $|\lambda(e^{AT})| < 1$ , and  $|\lambda(\exp(A_0T))| < 1$ , the system (48) is asymptotically stable for sufficiently small  $\varepsilon$ .

Hence, the difference in deviation from steady state will become zero, i.e.,

$$\tilde{x}\left(\frac{k}{\varepsilon}\right) \rightarrow 0 \quad \tilde{z}\left(\frac{k}{\varepsilon}\right) \rightarrow 0 \quad \text{as } k \rightarrow \infty$$

That means:-

$$\tilde{x}\left(\frac{k+1}{\varepsilon}\right) \rightarrow 0 \quad \tilde{z}\left(\frac{k+1}{\varepsilon}\right) \rightarrow 0 \quad \text{as } k \rightarrow \infty$$

Since we have considered an interval  $\frac{k}{\varepsilon} \leq n < \frac{k+1}{\varepsilon}$ , during which  $x(n)$  and  $z(n)$  are

bounded functions of  $x\left(\frac{k}{\varepsilon}\right)$  and  $z\left(\frac{k}{\varepsilon}\right)$ , therefore

$$\tilde{x}(n) \rightarrow 0 \quad \tilde{z}(n) \rightarrow 0 \quad \text{as } n \rightarrow \infty$$

Hence,  $x(n)$  and  $z(n)$  are also bounded for all  $n \geq 0$ .



As a result of all this discussion, we conclude that we can recover the performance of the system under state feedback using output feedback control. We see that the system is also asymptotically stable and bounded and reach steady state value after the first cycle. However, the closeness of the two systems is followed after first cycle when the initial conditions are different.

# CHAPTER 4

## SIMULATION RESULTS

In this chapter, the formulation of the state feedback control, output feedback control using high gain observer, and results of different simulations are discussed. The concept of multirate control is used to design output feedback control. Then the effects of changing different parameters like initial conditions,  $\epsilon$ , and sampling time are discussed. Also the peaking phenomenon and saturation of control is discussed. Finally the results, and future recommendations are given at the end.

### 4.1 CONSTRUCTION OF THE SYSTEM

#### 4.1.1 The Controlled System: A dc Servomotor

For my thesis, I took the dc servomotor as the system to be controlled [20]. The motor can be described as a linear, time invariant system. We have the following second order model for the dc servomotor:

$$\dot{\Theta}_m(t) = \omega(t) \tag{1}$$

$$\dot{\omega}(t) = -\frac{\omega(t)}{\tau_m} + \frac{u_a(t)}{k_b \tau_m} \tag{2}$$

where

$\Theta_m(t)$  = position of the motor

$u_a(t)$  = input voltage

$\omega(t)$  = angular velocity

$J =$ moment of inertia of motor and load	: $2.6 \times 10^{-5} \text{ Kgm}^2$
$R =$ Resistance	: 1.6 ohms
$K_b =$ Back emf constant	: 0.0782 Vs
$K_T =$ Torque constant of motor	: 0.076 Nm/A
$\tau_m =$ Mechanical time constant of motor	: $\frac{JR}{K_b K_T}$

Now writing (1) and (2) in matrix form, we get

$$\begin{bmatrix} \dot{\Theta}_m(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\tau_m} \end{bmatrix} \begin{bmatrix} \Theta_m(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{K_b \tau_m} \end{bmatrix} u_a(t) \quad (3)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \Theta_m(t) \\ \omega(t) \end{bmatrix} \quad (4)$$

It is to be noted that there is only one output of the motor,  $\Theta_m$ . Now replacing the variables by their specific values of the motor, (3) is written as:

$$\begin{bmatrix} \dot{\Theta}_m(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -142.86 \end{bmatrix} \begin{bmatrix} \Theta_m(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1826.85 \end{bmatrix} u_a(t) \quad (5)$$

or, we may write (4) and (5) in the standard form of state space model, given by:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (6)$$

$$y(t) = Cx(t) \quad (7)$$

where

$$\dot{x}(t) = \begin{bmatrix} \dot{\Theta}_m(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix}$$

$$x(t) = \begin{bmatrix} \Theta_m(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

and the system matrices are given by:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -142.86 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1826.85 \end{bmatrix}$$

$$C = [1 \quad 0] \quad D = [0]$$

Equations (6) and (7) represent continuous time state space model. To get a discrete time model, we have to discretize the given system. Now selecting ‘ zero order hold ‘ and the sampling time as  $T_c = 5$  ms, and using the Matlab command ‘ c2dm ’, we get the discretized system as given below:

$$x(k+1) = ax(k) + bu(k) \quad (8)$$

$$y(k) = cx(k) \quad (9)$$

where a, b, and c are discrete time matrices, and are given by:

$$a = \begin{bmatrix} 1 & 0.0036 \\ 0 & 0.4895 \end{bmatrix} \quad B = \begin{bmatrix} 0.0182 \\ 6.5277 \end{bmatrix}$$

$$c = [1 \quad 0]$$

#### 4.1.2 Poles of The System

In control system design, we can use the pole placement method to place our poles, if all the states are available. However, even if some of the states are not available, we can use an observer in the feedback path. Using pole placement allows us to choose the poles anywhere in the s-plane or z-plane, depending on the desired behavior of the system. To find the pole location, we assume a second order system with the following transfer function:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (10)$$

where  $\xi$  is the damping ratio and  $\omega_n$  is the undamped natural frequency. Now if we specify the relative overshoot to be maximum 5%, then using the relation:

$$rel. overshoot = e^{\frac{-\xi\pi}{\sqrt{1-\xi^2}}}$$

the damping ratio is found to be  $\xi = \frac{1}{\sqrt{2}}$ . Also, for the system, the settling time is assumed to be  $t_s = 1s$ . For a second order system, we have

$$t_s = \frac{4.6}{\xi\omega_n}$$

and after substituting the values, we get  $\omega_n = 6s^{-1}$ . Now the poles can be calculated by setting the characteristic equation of our assumed transfer function (10) to zero. With the above results, we get

$$s^2 + 2 \cdot \frac{1}{\sqrt{2}} \cdot 6s + (6)^2 = 0$$

which gives

$$s_{1,2} = -\left(\frac{6}{\sqrt{2}}\right) \pm j\left(\frac{6}{\sqrt{2}}\right)$$

The poles  $s_{1,2}$  lie on 45 degree axis, in the left half of s-plane, so they meet the stability criterion. Now the poles can be discretized using the z-transform relation:

$$z = e^{sT}$$

By specifying the values, Matlab directly calculates the poles in z-transform. The discrete time poles are given by:

$$z_{1,2} = 0.9788 \pm j0.0208$$

For using in Matlab program, we define the pole vector as:

$$P = [0.9788+j0.0208 \quad 0.9788-j0.0208]$$

### 4.1.3 Design of State Feedback Control

The assumed control law is given by:

$$u(t) = -Kx(t) + Nr(t) \quad (11)$$

This is for continuous time system. Since we are dealing with discrete time system, so the control law for such system is defined as:

$$u(k) = -Kx(k) + Nr(k) \quad (12)$$

where  $K$  is a  $1 \times 2$  gain matrix,  $x(k)$  is  $2 \times 1$  state matrix. The state matrix contains two states  $x_1$  and  $x_2$ . State  $x_1$  is the output which is the difference between the reference position and the actual position of the motor, and state  $x_2$  is the angular velocity  $\omega$ . The  $Nr(k)$  is the weighted step forward function, which is needed to force the tracking error to zero when the motor reaches the steady state value.

Now the gain  $K$  has to be designed such that the matrix  $(a-bK)$  meets the Hurwitz stability criterion. Matlab command 'place' is used to calculate the gain  $K$ . We have found out the gain  $K$  to be:

$$K = [0.0270 \quad -0.0718]$$

The eigenvalues of  $(a-bK)$  are by:

$$eig(a - bK) = 0.9788 \pm j0.0208$$

which lie inside the unit circle of the z-plane.

To find the value of  $N$ , consider a discrete system with a transfer function  $H(z)$  and step input  $r(k)$  with its z-transform given by:

$$r(z) = \frac{z}{z-1} \cdot r$$

To force the output  $Y(z)$  to approach the value of the step input for discrete time  $k \rightarrow \infty$ , that is,  $\lim_{k \rightarrow \infty} Y(k)$ , we get the following equation:

$$\begin{aligned} \lim_{k \rightarrow \infty} Y(k) &= \lim_{k \rightarrow \infty} (z-1)Y(z) = \lim_{z \rightarrow 1} (z-1) \cdot H(z) \cdot \frac{z}{z-1} r \\ &= H(1)r \end{aligned} \quad (13)$$

To reach now the value of the step input, we have to multiply the step output  $r$  by the inverse of the limit of the transfer function, which is  $H(1)^{-1}$ . Now substituting (12) in (8), we get

$$x(k+1) = (a-bK)x(k) + bN r(k) \quad (14)$$

Now applying z-transform, we get

$$zIx(z) = (a-bK)x(z) + bN r(z)$$

Solving this equation for  $x(z)$  and then substituting in (9), we get

$$Y(z) = c(zI - a + bK)^{-1} bN r(z)$$

Now taking limits, we get

$$\lim_{z \rightarrow 1} Y(z) = c(I - a + bK)^{-1} bNr(1) \quad (15)$$

Now comparing (15) with (13), we get

$$N = H(1)^{-1} = \left( c(I - a + bK)^{-1} b \right)^{-1} \quad (16)$$

And after substituting the values, we get  $N = 0.0270$ .

#### 4.1.4 Construction of an Observer

To check, whether our system is observable or not, we use the method described in chapter 2. We find the rank of observability test matrix given by:

$$N = \begin{bmatrix} A^T & A^T C^T \end{bmatrix}$$

Now using the given matrices of the plant, we get

$$N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

It is obvious that the rank of N is 2, equal to the order of the system, therefore, the given system is completely observable.

The observer structure has the same order as the original system. Therefore, the observer equations are given by:

$$\dot{\hat{\Theta}}_m = \hat{\omega} + \frac{a_1}{\epsilon} (\Theta_m - \hat{\Theta}_m) \quad (17)$$

$$\dot{\hat{\omega}} = \frac{a_2}{\epsilon^2} (\Theta_m - \hat{\Theta}_m) \quad (18)$$

The parameters  $a_1$  and  $a_2$  are chosen to be one. The observer gain matrix L is chosen to be:

$$L = \begin{bmatrix} 1 & 1 \\ \epsilon & \epsilon^2 \end{bmatrix}^T$$

The equations (17) and (18) can be written in matrix form as:

$$\begin{bmatrix} \dot{\hat{\Theta}} \\ \dot{\hat{\omega}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\epsilon} & 1 \\ -\frac{1}{\epsilon^2} & 0 \end{bmatrix} \begin{bmatrix} \hat{\Theta} \\ \hat{\omega} \end{bmatrix} + \begin{bmatrix} \frac{1}{\epsilon} \\ \frac{1}{\epsilon^2} \end{bmatrix} \Theta_m \quad (19)$$



where  $\Theta_m$  is the output of the motor. Now if we choose the value of  $\varepsilon = 0.01667$ , which corresponds to  $\frac{1}{\varepsilon} = 60$ , the gain matrix  $L$  is given by  $L = [60 \quad 3600]^T$  which corresponds to very high gain. With this value of  $\varepsilon$ , (19) can be written as:

$$\begin{bmatrix} \dot{\hat{\Theta}} \\ \dot{\hat{\omega}} \end{bmatrix} = \begin{bmatrix} -60 & 1 \\ -3600 & 0 \end{bmatrix} \begin{bmatrix} \hat{\Theta} \\ \hat{\omega} \end{bmatrix} + \begin{bmatrix} 60 \\ 3600 \end{bmatrix} \Theta_m \quad (20)$$

The reason for choosing this value of  $\varepsilon$  are:

a. The poles of the high gain observer are located relatively farther to the left in the left half of the  $s$ -plane than the poles of the control. This means that the convergence of the estimation error is faster than the actual control. The characteristic equation of the unforced observer system is given by:

$$\det[sI - A] = s^2 + 60s + 3600 = 0$$

Therefore, the poles of the observer are located at:

$$s_{1,2} = -30 \pm j30\sqrt{3}$$

b. Increasing the gain, that is making  $\varepsilon \rightarrow 0$ , means also increasing the frequency band of the observer. Practical experience shows that the frequency of the observer should be at least 10 times smaller than the sampling frequency. We assume that  $f_{\text{obs}} = 60$  Hz. The sampling frequency is assumed to be  $f_s = 2000$  Hz, which corresponds to a sampling time of 0.5 ms. Therefore,  $f_{\text{obs}} \ll f_s$ . We denote the sampling time of the observer by  $T_o$ , then  $T_o = 0.5$  ms. Comparing the sampling time  $T_c$  of the control, and  $T_o$

of the high gain observer, we see that the ratio is given by  $n = \frac{T_c}{T_o} = 10$ . Thus, each tenth observer update is pushed to the control.

Now consider (20). This is a continuous time observer equation. We can write it as:

$$\dot{O}(t) = A_o O(t) + B_o \Theta_m(t) \quad (21)$$

$$y_o(t) = C_o O(t) \quad (22)$$

where 'O' stands for observer state, and

$$\dot{O}(t) = \begin{bmatrix} \dot{\hat{\Theta}}(t) \\ \dot{\hat{\omega}}(t) \end{bmatrix} = \begin{bmatrix} \dot{O}_1(t) \\ \dot{O}_2(t) \end{bmatrix}$$

and 
$$O(t) = \begin{bmatrix} \hat{\Theta}(t) \\ \hat{\omega}(t) \end{bmatrix} = \begin{bmatrix} O_1(t) \\ O_2(t) \end{bmatrix}$$

Now we discretize the observer system (21) - (22) using 'zero order hold' and sampling time of  $T_o = 0.5$  ms, using the Matlab command 'c2dm'. We get

$$O(k+1) = a_o O(k) + b_o \Theta_m(k) \quad (23)$$

$$y(k) = c_o O(k) \quad (24)$$

where 
$$a_o = \begin{bmatrix} 0.7042 & 0.0043 \\ -15.3134 & 0.9595 \end{bmatrix} \quad b_o = \begin{bmatrix} 0.2955 \\ 15.3134 \end{bmatrix} \quad c_o = [1 \quad 0]$$

#### 4.1.5 Combined System - Single Rate Implementation

The term 'combined' is used to indicate that the observer is brought into the feedback loop, but both the control and the observer use the same sampling rate.

Equations (8), (9) and (12) are rewritten as:

$$x(k+1) = ax(k) + bu(k) \quad (8)$$

$$y(k) = cx(k) \quad (9)$$

$$u(k) = -Kx(k) + Nr(k) \quad (12)$$

Now the output of the observer  $O(k)$  is used in the control law (12) instead of the state  $x(k)$ . Therefore, (12) takes the form of output feedback control given by:

$$u(k) = -KO(k) + Nr(k) \quad (25)$$

In (23),  $\Theta_m(k)$  is the input to the observer. Since the input of the observer is, in fact, output of the actual system, therefore, substituting (9) in (23), we get the observer equation based on the output of the system as:

$$O(k+1) = b_o cx(k) + a_o O(k) \quad (26)$$

Now substituting (25) in (8), we get the closed loop system as:

$$x(k+1) = ax(k) - bKO(k) + bNr(k) \quad (27)$$

Now writing (26) and (27) in matrix form, we get

$$\begin{bmatrix} x(k+1) \\ O(k+1) \end{bmatrix} = \begin{bmatrix} a & -bK \\ b_o c & a_o \end{bmatrix} \begin{bmatrix} x(k) \\ O(k) \end{bmatrix} + \begin{bmatrix} bN \\ 0 \end{bmatrix} r(k) \quad (28)$$

and

$$y(k) = \begin{bmatrix} c & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ O(k) \end{bmatrix} \quad (29)$$

or it can be written as a new closed loop system as:

$$H(k+1) = AA H(k) + BB r(k) \quad (30)$$

$$y(k) = CC H(k) \quad (31)$$

where

$$H(k+1) = \begin{bmatrix} x(k+1) \\ O(k+1) \end{bmatrix} \quad H(k) = \begin{bmatrix} x(k) \\ O(k) \end{bmatrix}$$

and after substituting the values, we get

$$AA = \begin{bmatrix} 1 & 0.0036 & -0.0005 & 0.0013 \\ 0 & 0.4895 & -0.1762 & 0.4685 \\ 0.2958 & 0 & 0.7042 & 0.0043 \\ 15.3134 & 0 & -15.3134 & 0.9595 \end{bmatrix}$$

$$BB = \begin{bmatrix} 0.0005 \\ 0.1762 \\ 0 \\ 0 \end{bmatrix} \quad CC = [1 \ 0 \ 0 \ 0]$$

It is to be seen that zero padding is done in BB and CC to make the dimensions compatible with the system, since the order of the combined system is 4.

## 4.2 SIMULATIONS

After constructing the system, now we go to the computer simulations. I used UNIX system to implement all the simulations.

### 4.2.1 State Feedback Control

The Matlab program, that implements the state feedback control discussed in section 4.1.1 - 4.1.3, is given in Appendix A. In this Matlab program, the continuous time system matrices are defined first, then the system is discretized at sampling time  $T_c = 5$  ms, and using the Matlab command 'c2dm'. Then the poles are calculated using z-transform formula  $z = \exp(sT_c)$ . The gain matrix K is then calculated based on the poles calculated earlier, by using the Matlab command 'place'. The weighting factor N is calculated. Then the matrices for the closed loop system are defined as A1 and B1. Then

defining  $r$  as a step input for  $k = 1000$ , (or it can be varied), the closed loop system is implemented using 'dstep' command from Matlab. The output  $y$  and the states  $x_1$  and  $x_2$  are stored in a vector  $[y,x]$ . The state feedback control is then calculated using the state vector  $x$ . To fix the x-axis to time scale, a time vector is defined, and finally the 'plot' command is used to get the plots for desired states / output.

The resulting plots for the control  $U$ , and the states  $x_1$  and  $x_2$  and the output  $y$  are given in Appendices A1, A2, A3 and A4, respectively. Looking at the plot for  $U$ , we see that it has a spurious peak in the transient period, it reaches steady state value after about 1s, which is very close to our assumption of  $t_s = 1$  s as given in section 4.1.2. The control finally settles down to zero. Note that the maximum amplitude reached by the control in its transient period is about 0.22. Now looking at the plot for the state  $x_1$ , which is the same as the output  $y$ , we see that this also reaches its steady state value after about 1s, which means that the output is exactly tracking the input which is a step input in this case. The state  $x_2$  is following the control  $U$ , except for the difference in the transient period, where the amplitude of  $x_2$  is much more than the amplitude of control  $U$ .

#### 4.2.2 Combined System - Single Rate Implementation

The Matlab program that implements the combined system, discussed in section 4.1.5, is given in Appendix B. In addition to the explanation given in section 4.2.1, the continuous time observer matrices dependent on  $\epsilon$ , given by  $A_o$ ,  $B_o$ ,  $C_o$  and  $D_o$  are also defined. For single rate implementation, both the control and the observer are discretized at the same sampling time  $T_c$ , by using the Matlab command 'c2dm'. Then the closed loop system, with the observer placed in the feedback path, is simulated using 'dstep'

command from Matlab. The output  $y$  and the states  $x_1$ ,  $x_2$ ,  $O_1$ ,  $O_2$  are stored in a vector  $[y,x]$ . The states  $x_1$  and  $x_2$  are the states of the controller, and  $O_1$ ,  $O_2$  are the states of the observer which represent the estimates of the states  $x_1$  and  $x_2$ , respectively. Time vector is also defined for x-axis, and finally the 'plot' command is used to get the desired graphs of the states.

The resulting plots for the output feedback control  $U$ , states  $x_1$ ,  $x_2$ ,  $O_1$ ,  $O_2$  and the output  $y$  are given in Appendices B1, B2, B3, B4, B5 and B6, respectively. Looking at the graph for  $U$ , we see that it attains its steady state value after about two seconds, and finally settles down to zero. This delay in settling time is caused by the observer in the feedback path. However, it is to be noted that this control graph is almost similar to the one described in section 4.2.1. Now looking at the graph for state  $x_1$ , which is also the output of the system, it is seen that it also reaches its steady state value of one after about two seconds. Comparing this one, with the graph for  $x_1$  of section 4.2.1, we note that both are similar except for the small difference in the overshoots in transient period and difference in settling time. The state  $x_2$  is following the control  $U$ , with large overshoot in the transient period. The plots for  $O_1$  and  $O_2$  are same as that for  $x_1$  and  $x_2$ , since  $O_1$  is the estimate of  $x_1$  and  $O_2$  is the estimate of  $x_2$ . This is because the initial conditions for both are the same and also the sampling time is the same. The point to note is that, the performance under output feedback control using single rate implementation is similar to the performance under state feedback control. Since zero initial conditions are assumed for both the systems, no peaking is observed in this implementation. The peaking will be observed when the initial conditions are different, as already discussed in chapter 2.

### 4.2.3 Multirate Implementation

Now we develop a Matlab program that will simulate the system (30) - (31) using two different sampling rates. The dynamics of the control are slow and the dynamics of the observer are fast. The Matlab program that simulates this scheme is given in Appendix C. Note that the two sampling times are now different, the dynamics of the observer are ten times faster than that of the control. The value of  $\epsilon$  is assumed to be 0.01667 as before. Now since the dynamics of the control and the observer are different, so they cause an unstable behavior of the system. Therefore, before they enter the plant, both should have the same (fast) dynamics. To cater for this point, we sample the actual system given by continuous time matrices A, B, C, and D, at fast sampling period  $T_o$ . Therefore, we get discrete time matrices  $a_1$ ,  $b_1$ ,  $c_1$ , and  $d_1$ , and

$$x(k+1) = a_1 x(k) + b_1 u(k) \quad (32)$$

and (26) will now become:

$$O(k+1) = b_o c_1 x(k) + a_o O(k) \quad (33)$$

with  $u(k)$  defined by (25), the output feedback control. Now we want to keep the control (25) constant for ten points and after the tenth point, we want to update the control, while the states of the observer are calculated at every point. So we define 'n' as fast time scale, and select 'n' as a ratio of the sampling time of the control to the sampling time of the observer, i.e.,  $n = \frac{T_c}{T_o}$ . Since  $T_c = 5$  ms and  $T_o = 0.5$  ms, therefore the ratio comes out to be  $n = 10$ . Also we define  $k$  as the slow time scale. The choice for the number of points  $k$  depends on how long it takes for the output feedback control to reach its steady state

value. In our case, we have chosen the number of points of  $k$  for multirate to be equal to the number of points used for the state feedback control. Then we need to make a loop in the Matlab program, which, will carry out  $n \times k$  calculations using (25), (32) and (33). The Matlab command 'dlsim' is used to implement this scheme. The initial conditions are also required to run the loop, and these initial conditions are defined as  $x_0$  and  $O_0$  for the control and the observer, respectively. The results are stored in a vector  $[h,g]$ , where  $h$  is the output and  $g$  represents the states.

Now one cycle consists of ten such calculations before the control is updated for the next cycle. Therefore, there is a need to store the results for each cycle. The program is so designed such that after the very first cycle, the last value of each cycle becomes the initial conditions for the next cycle. Therefore, to avoid double value storage in the vectors  $gg$  and  $uu$ , we leave the last value of the results of each cycle and store them up to  $n-1$  values. By this way, we shall be stacking the results of all the cycles in one vector without having any of the double values. However, there is a problem observed in this procedure, that is, instead of storing ten values for each cycle, we are, in fact, storing nine values. That way, we loose information which corresponds to one complete cycle of fast sampling period, i.e.,  $T_o$ . To take care of this problem, we choose  $n = \frac{T_c}{T_o} + 1$ , so that, now after storing  $n-1$  points, we shall have stored, in fact, ten points for each cycle, instead of storing nine points for each cycle as previously. This procedure is shown in Appendix C6.



This Matlab program for multirate scheme is run for same initial conditions for both the control and the observer. The initial conditions  $x_0$  and  $O_0$  are assumed to be zero in this case. We can see, from Appendices C1, C2, C3, C4, and C5, that the behavior of the output feedback control  $u$ , and the states  $x_1$ ,  $x_2$ ,  $O_1$ , and  $O_2$  is exactly the same as it was for single rate implementation of the combined system discussed in section 4.2.2, which means, that, for same (zero) initial conditions, the performance under output feedback control using multirate scheme, is same as the performance under state feedback control using single rate scheme.

#### 4.2.4 Performance of Multirate Scheme Under State Feedback Control

This Matlab program can be modified to obtain the results if we use state feedback control instead of the output feedback control inside the loop of the program. To implement this scheme, we replace the equation for  $u_0$  inside the loop, by the equation:

$$u_0 = -Kx_0 + Nr \quad (34)$$

This scheme is shown in Appendix D. This program is run for the same (zero) initial conditions for both the systems. Now looking at the Appendices D1, D2, D3, D4, and D5, we observe that these graphs are very much similar, in fact, the same, to the one obtained by the state feedback control designed in section 4.2.1. This was the case when the initial conditions are the same and assumed to be zero for both systems. The effects of different initial conditions will be discussed later in this chapter.

#### 4.2.5 Behavior of Multirate Scheme in First Cycle

Now to see the behavior of multirate scheme in first cycle, we modify the Matlab program given in Appendix C. We keep  $n = 10$  and change  $k$  from  $k = 100$  to  $k = 1$ , so

that the output feedback control is constant over this one cycle. We also assume that the initial conditions are the same (zero) for both systems. This scheme is shown in Appendix E. We see from the Appendix E1 that the control is constant for one cycle. The Appendices E2, E3 show that the states  $x_1$  and  $x_2$  are rising in the first cycle. From Appendix E4, E5 it can be seen that the state  $O_1$  is trying to follow the state  $x_1$ , but state  $O_2$  has more steeper trajectory; however it is also rising to attain its maximum value. The important point to be noted here is that all the states attain very low amplitude value in the first cycle.

### **4.3 EFFECTS OF CHANGING DIFFERENT PARAMETERS**

All the cases discussed in section 4.2 correspond to the same (zero) initial conditions, the value of  $\varepsilon = 0.01667$  and  $T_c = 5$  ms,  $T_o = 0.5$  ms. Now let us see as to what happens if we change these parameters.

#### **4.3.1 Effects of Changing the Initial Conditions**

##### **4.3.1a Performance of Multirate Under State Feedback Control**

Now the initial conditions are made different for the plant and the observer, i.e.,

$$x_o = [0 ; 0] \quad \text{and} \quad O_o = [1 ; 1]$$

This scheme is shown in Appendix F. The resulting graphs for  $uu$ ,  $x_1$ ,  $x_2$ , shown in Appendices F1, F2, and F3, respectively, are similar to the one for the case when the initial conditions were the same (zero). However, we can notice the change in the behavior of  $O_1$  and  $O_2$  given in Appendices F4, F5, respectively. We can see the change in the transient period of both the observer states. This is, because the initial conditions

for the observer are different from the initial conditions of the plant,  $O_o = [1 ; 1]$  in this case. The two systems starting at different initial states will naturally exhibit different behavior. The overshoots in the transient response of  $O_1$  and  $O_2$  do not cause any change in the control  $uu$ , and the states  $x_1$  and  $x_2$ , because the control law (34) is not using any of the states  $O_1$  and  $O_2$ , i.e., the observer is bypassed. However, it is seen that all the states reach their steady state values after about 1s. The overshoots in the transient period of  $O_2$  is of the order of  $O(\frac{1}{\varepsilon})$ .

#### 4.3.1b Performance Under Output Feedback Control

The performance of the system under output feedback control, with the same (zero) initial conditions, was discussed in section 4.2.3. In this section, we consider the performance of the same system, but with different initial conditions for the two systems. The initial conditions for the observer are assumed to be  $O_o = [1 ; 1]$ , whereas the initial conditions for the plant are  $x_o = [0 ; 0]$ . This scheme is represented in Appendix G, and the resulting graphs for control  $uu$ , and the states  $x_1$ ,  $x_2$ ,  $O_1$ , and  $O_2$  are given in Appendices G1, G2, G3, G4, and G5, respectively. Now the results are quite different from all the previous cases. From the graphs, we can observe the large overshoots in the transient response, which we call as peaking phenomenon, which was discussed in chapter 2. Now since the initial conditions for the observer are  $O_o = [1 ; 1]$ , there is an impulsive behavior, i.e., large overshoot, in the transient period. These overshoots are fed to the control given by (25). Therefore, the control (25) also exhibits the large overshoot in its transient period, which is obvious from the graph for  $uu$ , given in Appendix G1.

These overshoots are then fed to the plant via the control (25), which causes the states  $x_1$  and  $x_2$  to exhibit peaking in their transient period. Looking at the graphs for different states, it can be seen that the transient response of  $O_1$  is transmitted to state  $x_1$  and the transient response of  $O_2$  is transmitted to state  $x_2$ . Since the output of plant is the input to the observer, these states disrupted with peaking are fed to the observer, which like before, enter the plant via the control (25). This process continues and we observe very large overshoots in the transient period. The overshoots in the states  $x_2$  and  $O_2$  are of the order of  $O(\frac{1}{\epsilon})$ .

However, all the states reach the steady state value after the first cycle. This is because, the fast modes, i.e.,  $O_1$  and  $O_2$  decay rapidly in the first cycle as discussed in chapter 3. After the first cycle, the response of the system is due to the slow modes, i.e.,  $x_1$  and  $x_2$ . The behavior of the system in the first cycle, for different initial conditions, is given in Appendix H. We can see from Appendices H4 and H5, that the fast modes are decaying rapidly from their starting position. This proves the idea discussed in chapter 3, where we used singularly perturbed approach to prove that, the response of the system, after first cycle, is due to the slow modes as the fast modes decay to zero in first cycle.

#### 4.3.2 Effects of Saturating The Control

As discussed earlier in chapter 2, we can reduce peaking by saturating the control. In this scheme, we set the limits of output feedback control so that it can swing between specified values. This scheme of saturating the control is shown in Appendix I. A subloop is inserted in the outer loop of the Matlab program, and the maximum swing limits are set

to  $\pm 0.5$ . This value of swing limits is chosen keeping in view the maximum swing of the state feedback control. After setting the saturation limits of the output feedback control, the program is run.

The resulting graphs for saturated control, and the states are given in Appendices I1, I2, I3, I4 and I5, respectively. Looking at the graph for control, we see that, the maximum swing has been limited to  $\pm 0.5$ . It is important to note the behavior of the output  $x_1$ . Due to the output feedback control saturated, the negative part of overshoot, which had an amplitude of about -5 in case of nonsaturated control, has been significantly reduced to an amplitude of -0.4. Also the positive part of overshoot has been reduced to about 1.3 from about 2.5. As a result of saturation, the graph for  $x_1$  has come closer to the one obtained under state feedback control. Similarly, due to saturation, the overshoots in the transient period of  $x_2$  has been reduced significantly. Also, the behavior of  $x_2$  under saturated output feedback control is close to the one under state feedback control. The effect of saturation is also obvious on the observer state  $O_1$ , which exhibits similar behavior as state  $x_1$ . However, the observer state  $O_2$ , refer to Appendix I5, does not saturate much as the negative part of overshoot remains unchanged, however, the positive part of overshoot reduces from about 7 to about 3. This is because, the saturation does not eliminate peaking form  $O_2$ . Therefore, the idea of reducing peaking phenomenon by saturating the control, discussed in chapter 2, is confirmed here.

### 4.3.3 Effects of Changing $\epsilon$

Now to see the effects of changing the value of  $\epsilon$ , we keep the other parameters like  $T_c$ ,  $T_o$ , initial conditions, same as before and change only  $\epsilon$ . Now we assume the

value of  $\varepsilon$  ten times smaller than the previous value, i.e., now we have  $\varepsilon = 0.001667$ , which corresponds to  $\frac{1}{\varepsilon} = 600$ . The observer gain matrix  $L$ , now has large numbers, i.e.,

$$L = [600 \quad 360000]^T$$

The poles of the observer are now located, even farther to the left in the left half of the  $s$ -plane than the poles of the control. The poles of the observer with this value of  $\varepsilon$  are located at:

$$s_{1,2} = -300 \pm j300\sqrt{3}$$

This means that the convergence of the estimation error will be even more faster than the control as compared to the previous case. The Matlab program that simulates this scheme for multirate case is given in Appendix J. The resulting graphs for the control and the states are given in Appendices J1, J2, J3, J4, and J5, respectively. In this simulation, the initial conditions are assumed to be the same (zero). We can see that, the control and all the states attain their steady state value after about 1s. Whereas for the same program with  $\varepsilon = 0.01667$ , as given in Appendix C, it took about 2s for the control and the states to reach their steady state values. The graphs for  $\varepsilon = 0.001667$  are very much close to the one obtained under state feedback control given in Appendix A.

Now let us see the effect of having different initial conditions. In the Matlab program given in Appendix K, the initial conditions for the observer are set to  $O_o = [1 \ ; \ 1]$  and for the plant  $x_o = [0 \ ; \ 0]$ . The resulting graphs are given in Appendices K1, K2, K3, K4, and K5, respectively. We can observe the peaking phenomenon, but the point to note is that, the peaking is observed for a very short period in the transient period. It seems to

be a sort of pulse with very short time period and large amplitude in the first cycle, after which the system reaches its steady state value. The effect of decreasing the value of  $\varepsilon$ , has caused relatively larger overshoots in the control and the states  $x_2$  and  $O_2$ , as can be seen from the graphs given in Appendices H1, H3, and H5, respectively. Since, due to different initial conditions, state of the observer  $O_2$  exhibits more peaking in the transient period, which is being fed to the control, therefore the same effect appears in the transient response of the control. Now at the output, we have only one state  $x_1$ , which is fed back to the observer, therefore the state  $O_2$  remains unaffected thereafter. However, the control and ultimately the state  $x_2$ , will be effected by the state  $O_2$ . That is why, we see the large overshoots in the transient periods of control and  $x_2$ . We also observe that, as  $\varepsilon$  is made smaller, the overshoots in the transient periods of  $x_1$  and  $O_1$  have been reduced as compared to the case of previous value of  $\varepsilon$ . Also notice that, the amplitude of the overshoot of  $O_2$  is about ten times larger as compared to the case when  $\varepsilon$  was ten times larger.

Now let us saturate the control to reduce the peaking. The saturation level for the control is fixed to  $\pm 0.5$  as before. This scheme is given in Appendix L. The resulting graphs for control and the states are given in Appendices L1, L2, L3, L4, and L5, respectively. Looking at the graph for the output  $x_1$ , we note that the response has been improved a lot as compared to the case when  $\varepsilon = 0.01667$ . The negative peak in the transient period has been reduced significantly, and this graph is closer to the one given in Appendix A2. Therefore, by reducing the value of  $\varepsilon$ , we get the response under output

feedback control closer to the one under state feedback control. The effect of saturation is also obvious from the graphs given in Appendices L3 and L4.

Now let us see the behavior in the first cycle for  $\varepsilon = 0.001667$ . For the same (zero) initial conditions, the response in first cycle is almost same as discussed earlier for the case when  $\varepsilon = 0.01667$ . However, for initial conditions different from the previous case, i.e., for  $O_o = [1 ; 1]$  and  $x_o = [0 ; 0]$ , the response is different. The scheme when the initial conditions are the same (zero) is given in Appendix M and the corresponding graphs for control and the states are given in Appendices M1, M2, M3, M4, and M5, respectively. And for different initial conditions, the scheme is given in Appendix N, with corresponding graphs given in Appendices N1, N2, N3, N4, and N5, respectively. In the first cycle, the states  $x_1$  and  $x_2$  are still rising, but as is seen from Appendix N4, the state  $O_1$  is reaching its steady state value, after decaying rapidly. Also, observing the state  $O_2$  from Appendix N5, we see that, after decaying very rapidly and attaining the maximum overshoot, the state is trying to reach the steady state value. This shows that the decay of fast modes in the first cycle is even more rapid as compared to the case when  $\varepsilon = 0.01667$ . This also confirms the idea discussed in chapter 3.

#### 4.3.4 Effects of Changing Sampling Times

Now let us assume a sampling time of 10 ms for the control, i.e.,  $T_c = 10$  ms. To keep up the ratio of 100 between the sampling times of the control and the observer, we assume sampling time for observer to be 0.1 ms, i.e.,  $T_o = 0.1$  ms. The Matlab program for the simulation of state feedback control, using new sampling time, is shown in Appendix P. Since the poles of the system are dependent on the sampling time due to the



relation given by  $z = \exp(sT_c)$ , therefore they will change depending on the value of the sampling time. The program for new sampling time is run for 500 sampling points, and the corresponding graphs for the control  $U$ , and the states  $x_1, x_2$  are given in Appendices P1, P2, and P3, respectively. Looking at these graphs, we see that they are exactly similar to the one given in Appendix A. It takes about 1s to reach the steady state value as for the case when the sampling time was  $T_c = 5$  ms.

For multirate implementation, we keep the control constant for  $n = 100$ . Now if we sample the control at  $T_c = 10$  ms, then to get  $n = 100$ , we sample the observer at  $T_o = 0.1$  ms as  $n = \frac{T_c}{T_o}$ . That means, the dynamics of the observer are about 100 times faster than the dynamics of the control. Using these sampling times means that every 100th update of observer is being fed to the control. In this case when  $n = 100$ , we shall repeat the computations for  $k = 500$  to get the interval of 5s on x-axis. This scheme of multirate implementation is given in Appendix Q. The resulting graphs for the control and the states are given in Appendices Q1, Q2, Q3, Q4, and Q5, respectively. These simulations are for the case when the initial conditions for the two systems are different, i.e.,  $O_o = [1 ; 1]$ , and  $x_o = [0 ; 0]$ . Looking at these graphs, we see that, the behavior of the control and the states is almost similar to the one given in Appendix G. The only difference is in the amplitude of the overshoots in the transient period, i.e., the peaking of little less amplitude has been introduced. This peaking can be reduced by saturating the control as before. It is also seen that, the steady state value is reached after about 1.5s which is exactly the same as was found in case of previous values of the sampling times.

Therefore, changing the sampling times have not produced any significant change in the behavior of the system.

#### 4.4 OBSERVATIONS AND RESULTS

After going through all the simulations, as given in Appendices A-Q, we conclude the following results.

- a. The state feedback control designed was stable and bounded, and reaches its steady state value.
- b. The system worked good under state feedback control. All the states were stable and bounded and reached their steady state value.
- c. The output feedback control designed also achieved stability and boundedness and reached the steady state value. As a result, all the states were stable and bounded and reached their steady state values.
- d. The performance of the system under output feedback control was as good as under state feedback control. In other words, we could recover the performance of the system under state feedback control by using the output feedback control.
- e. In the first cycle, the fast modes decay rapidly and after the first cycle, the response of the system is due to the slow modes with small error induced by the fast mode.
- f. As  $\varepsilon \rightarrow 0$ , the convergence of the estimation error is much faster, and the system's performance under output feedback control is very much closer to the performance under state feedback control.

- g. Changing the initial conditions, i.e., having different initial conditions for the plant and the observer induces peaking.
- h. Saturating the control helps in reducing the peaking phenomenon significantly.
- i. The effect of changing the sampling times is not very significant, and ultimately it does not effect much the performance of the system, as long as the sampling time is chosen small enough.
- j. The ideas discussed in chapter 2 and 3 were confirmed.

#### 4.5 RECOMMENDATIONS

I have the following recommendations for future work.

- a. In my simulations, I have used only one input, that is, the step input. These simulations should be run with different inputs, such as, impulse input, ramp input, etc. to see the behavior of the system.
- b. In the analysis given in chapter 3, the observer is assumed to be continuous time, whereas [12] suggests that the observer should be in discrete time. Therefore, the same analysis should be done with the observer converted in discrete time.
- c. In my simulations, the saturation of the control could not eliminate peaking significantly from the observer state  $O_2$  . Therefore, efforts may be done to reduce it.
- d. The simulations are performed for a stable system. However, the same program should be tested for an unstable system and compare the results.

## **CONCLUSION**

The performance of a linear, time invariant system was tested using the output feedback control that uses a very high gain observer in its feedback path. The behavior of the system was checked, theoretically, in the first cycle using singular perturbation theory. The simulation results later, proved that the fast modes decay in the first cycle rapidly and the response of the system after the first cycle, is due to the slow modes. Also, simulation results proved that, the performance of the system under output feedback control was as good as under state feedback control. In other words, we could recover the performance of the system under state feedback control by using the output feedback control. It was also proved that, the saturation of the control helps in reducing the peaking that is caused by having different initial conditions for the two systems. It is also seen that by making the value of  $\varepsilon$  very small, the system's performance under output feedback control can be brought very close to the performance under state feedback control.

# **APPENDICES**

## APPENDIX A

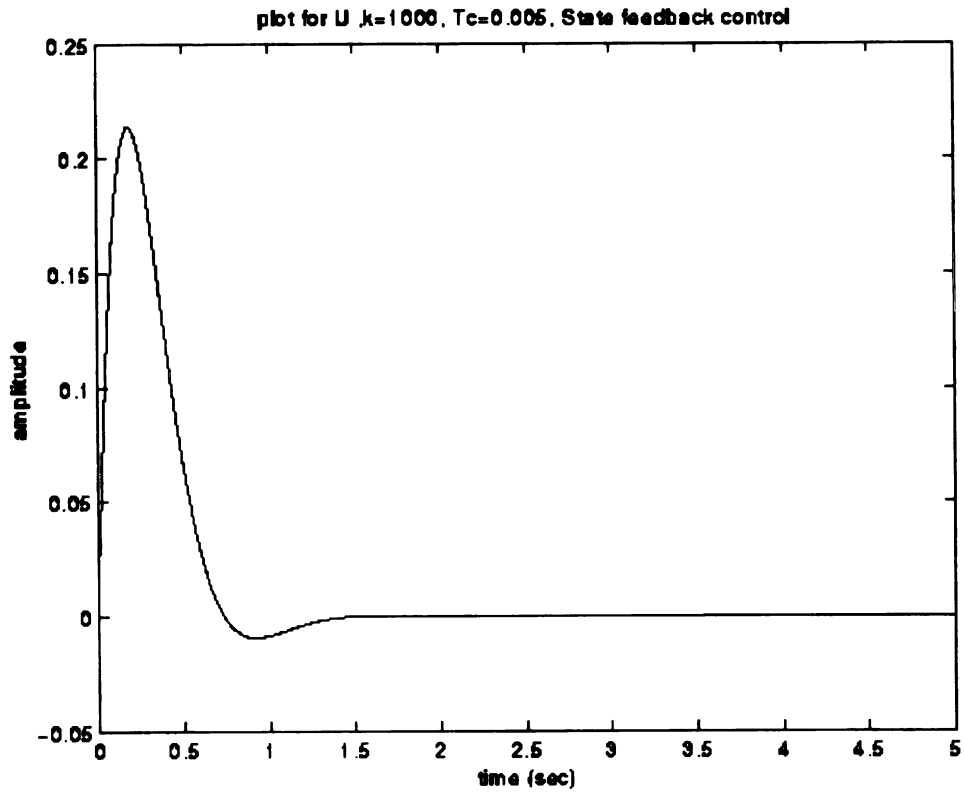
### Matlab program for State Feedback Control

```

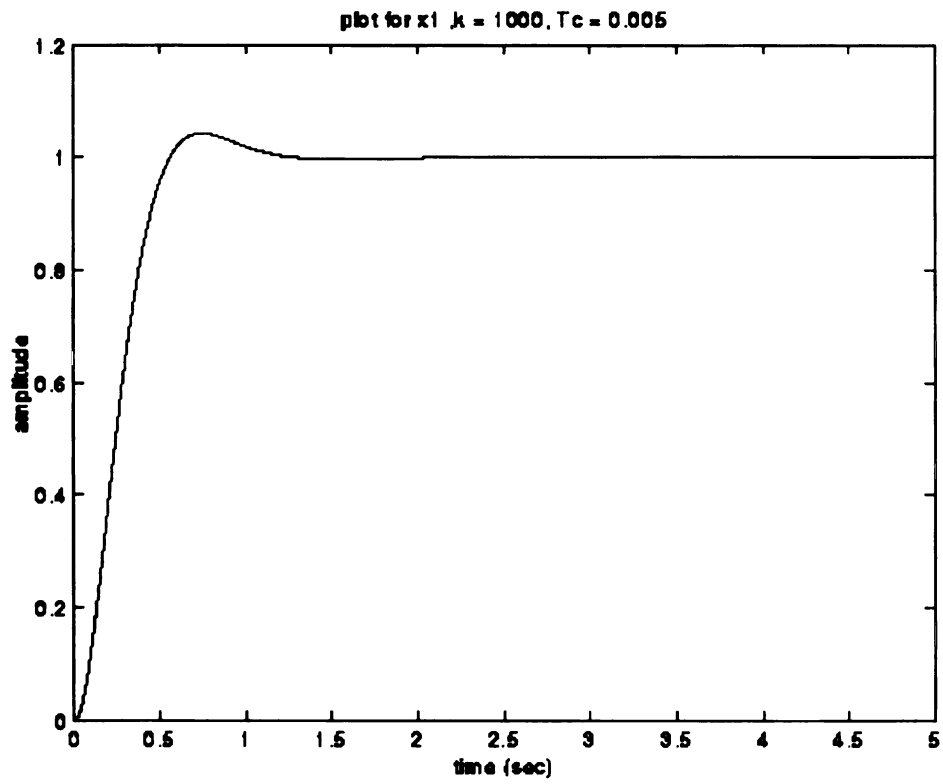
A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh')
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2]
K = place(a,b,Pc)
I = [1 0;0 1];
N = inv(c*inv(I-a+b*K)*b)
A1 = [a-b*K];
B1 = [b*N];
IU = [1];
k = 1000;
step = [k];
NN = N*ones(k,1);
[y,x] = dstep(A1, B1, c, d, IU, step);
U = (x*(-1)*K')+NN;
x1 = x(:,1);
x2 = x(:,2);
for k = 1:length(U)
    t(k,1)=k*Tc;
end
whos;
plot(t,U); title(' plot for U , k = 1000, Tc = 0.005, state feedback control ')
xlabel (' time (sec) ');
ylabel (' amplitude ');
end

```

## APPENDIX A1

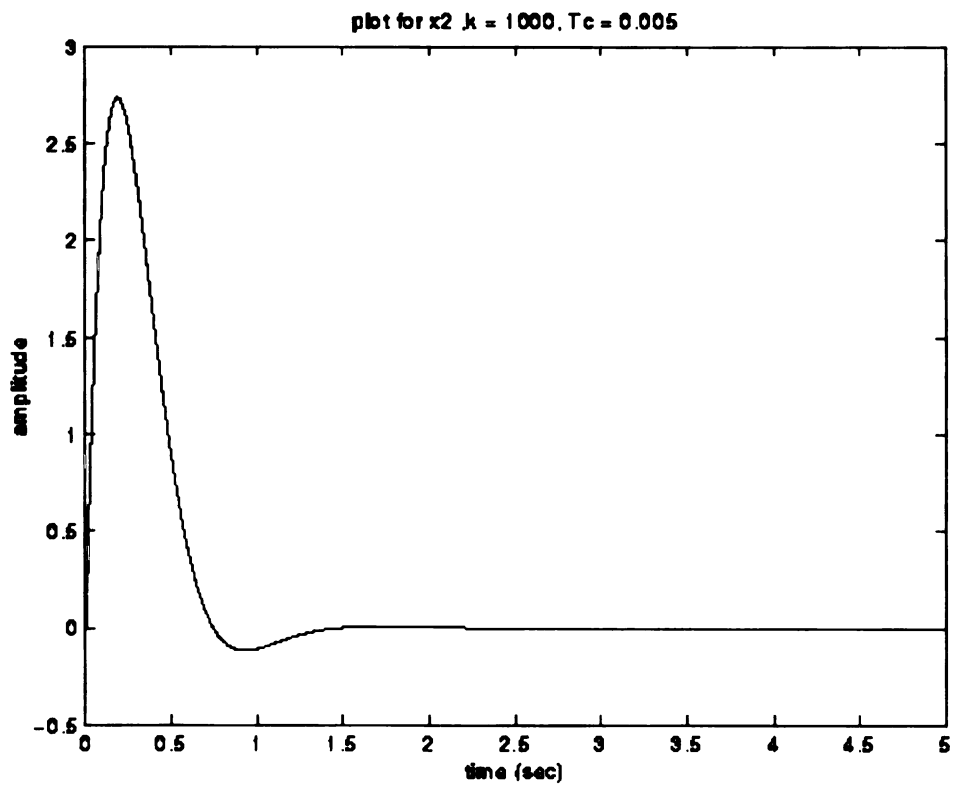


## APPENDIX A2

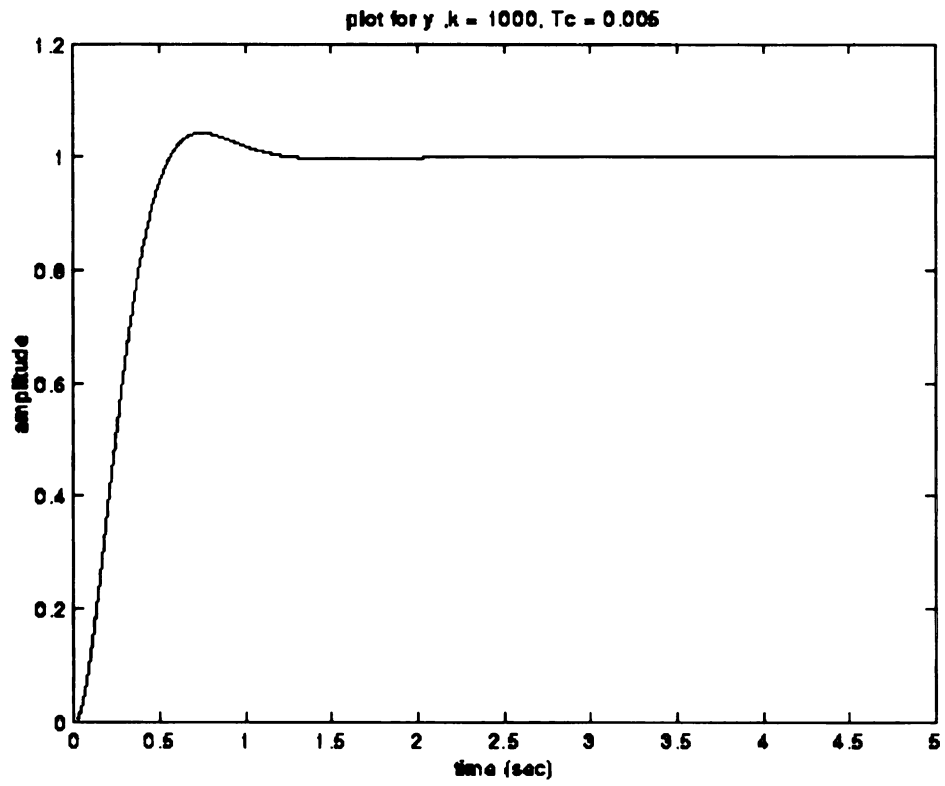




## APPENDIX A3



## APPENDIX A4



## APPENDIX B

### Combined System - Single Rate Implementation

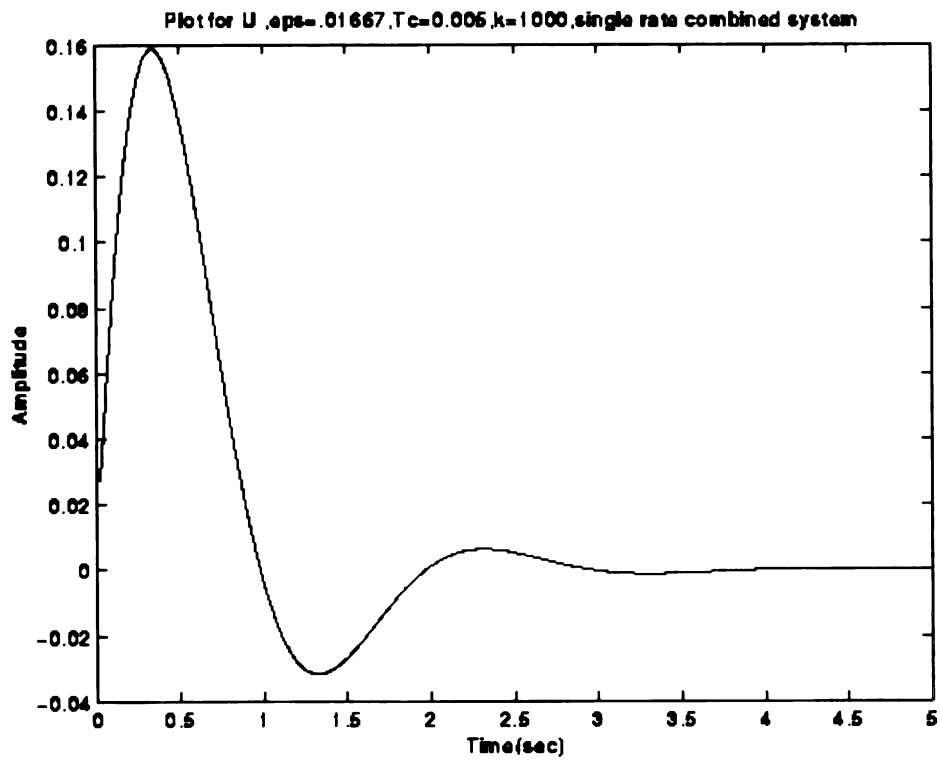
```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(s1*Tc);
Z2 = exp(s2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
I = [1 0;0 1];
N = inv(c*inv(I-a+b*K)*b);
ε = 0.01667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
[ao,bo,co,do] = c2dm(Ao,Bo,Co,Do,Tc,'zoh');
AA = [a -b*K;bo*c ao];
BB = [b*N;0;0];
CC = [c 0 0];
DD = [0];
k = 1000;
step = [k];
IU = [1];
NN = N*ones(k,1);
[y,x] = dstep(AA, BB, CC, DD, IU, step);
K = [0 0 K];
U = ( x*(-1)*K' )+NN;
x1 = x(:,1);
x2 = x(:,2);
O1 = x(:,3);
O2 = x(:,4);
for k = 1:length(U)
    t(k,1) = k*Tc;
end

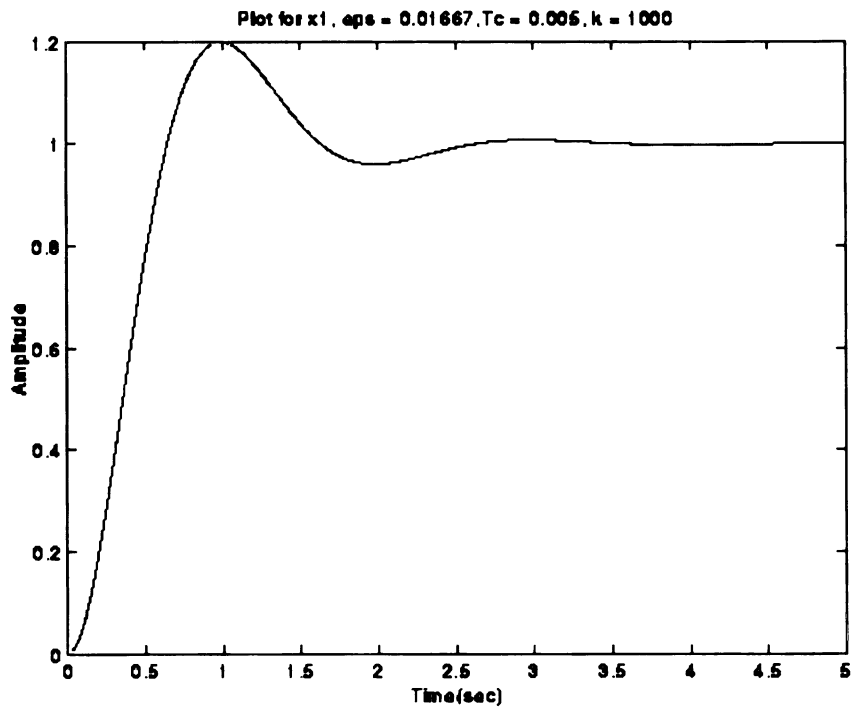
```

```
plot(t,U); title ( ' graph for U,  $\epsilon = 0.01667$ ,  $T_c = 0.005$ ,  $k = 1000$ , single rate combined  
system ' )  
xlabel ( ' Time(sec) ' );  
ylabel ( ' Amplitude ' );  
end
```

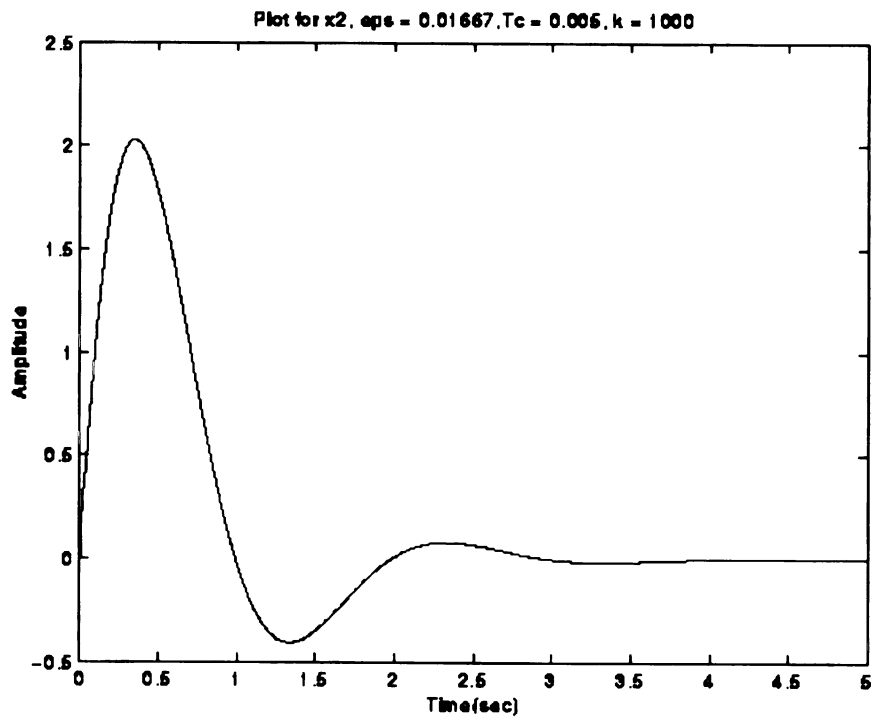
## APPENDIX B1



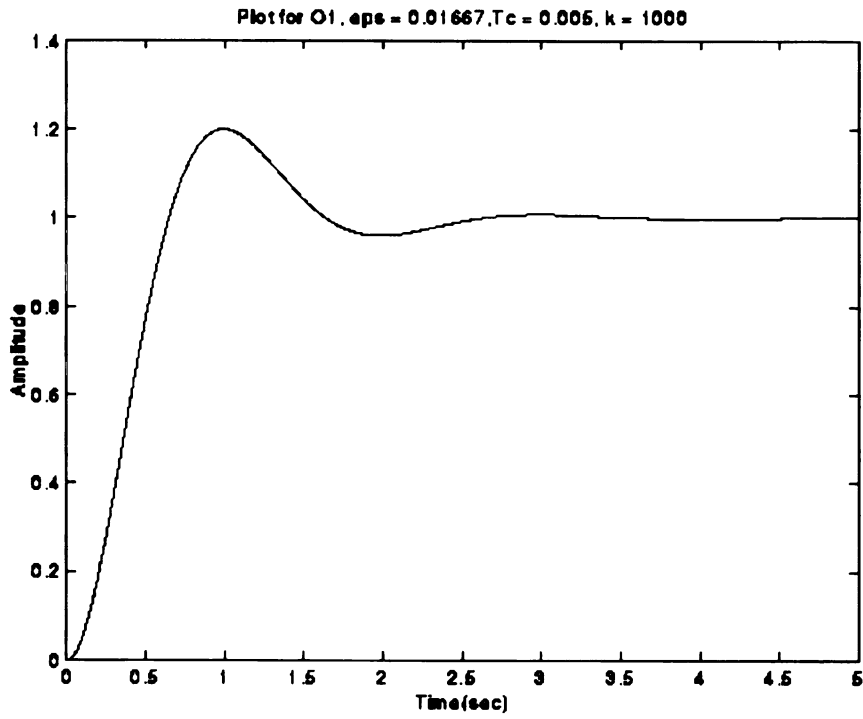
## APPENDIX B2



## APPENDIX B3

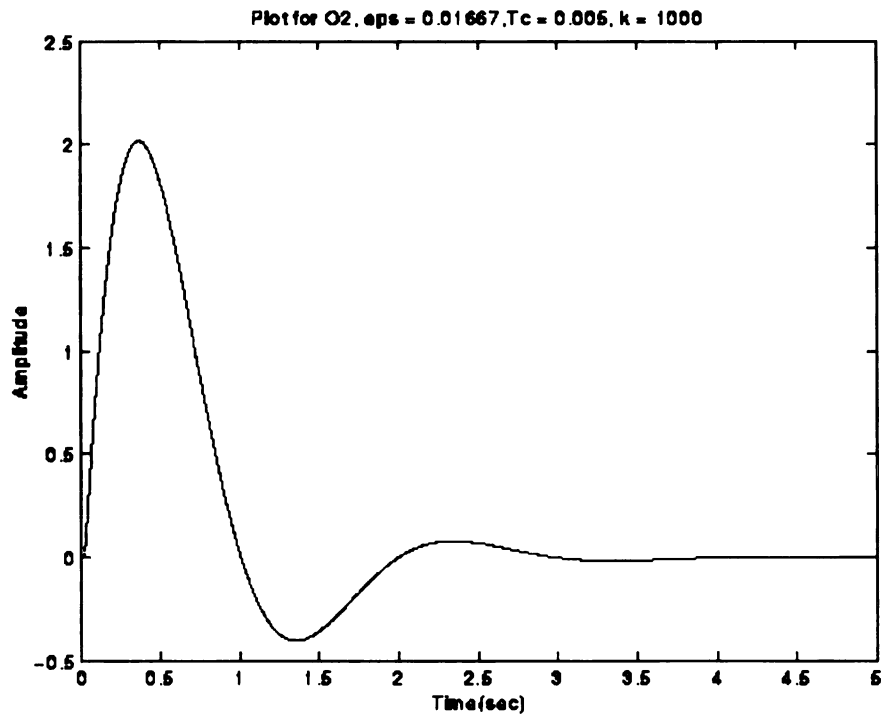


## APPENDIX B4

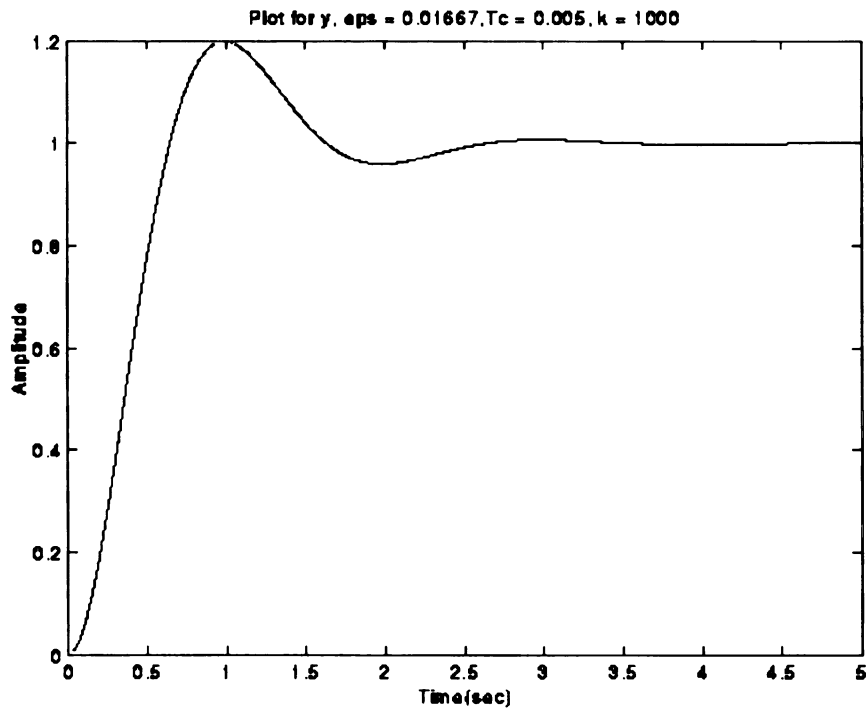




## APPENDIX B5



## APPENDIX B6



## APPENDIX C

### Multirate Implementation - Zero Initial Conditions

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.01667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [0;0];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1000;
for k = 1:kmax
    if k = 1
        g1 = go;
    elseif k > 1

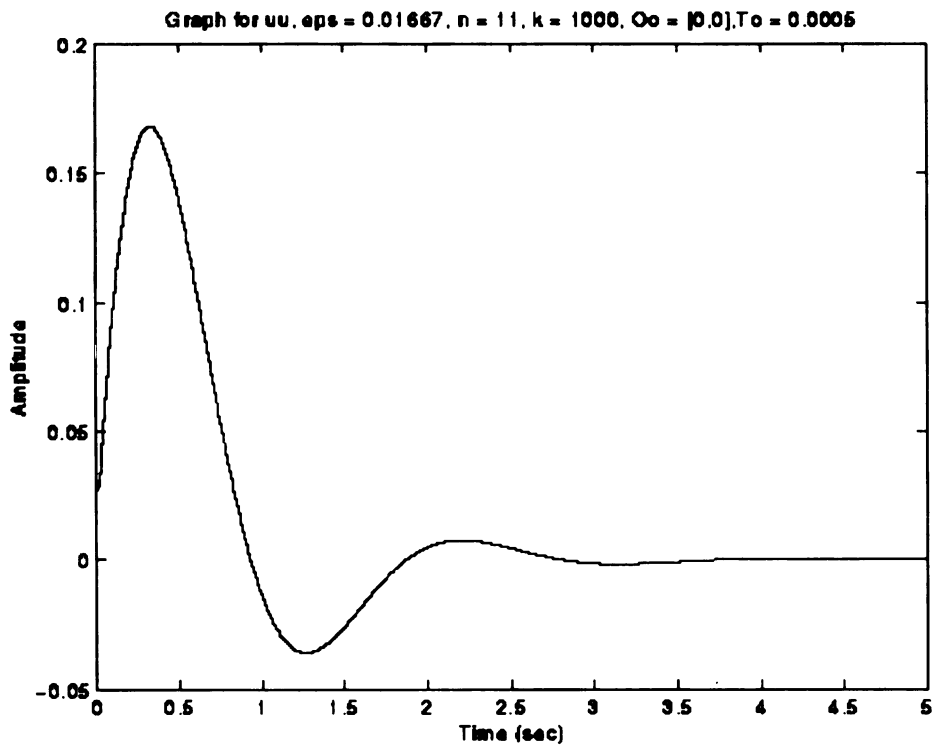
```

```

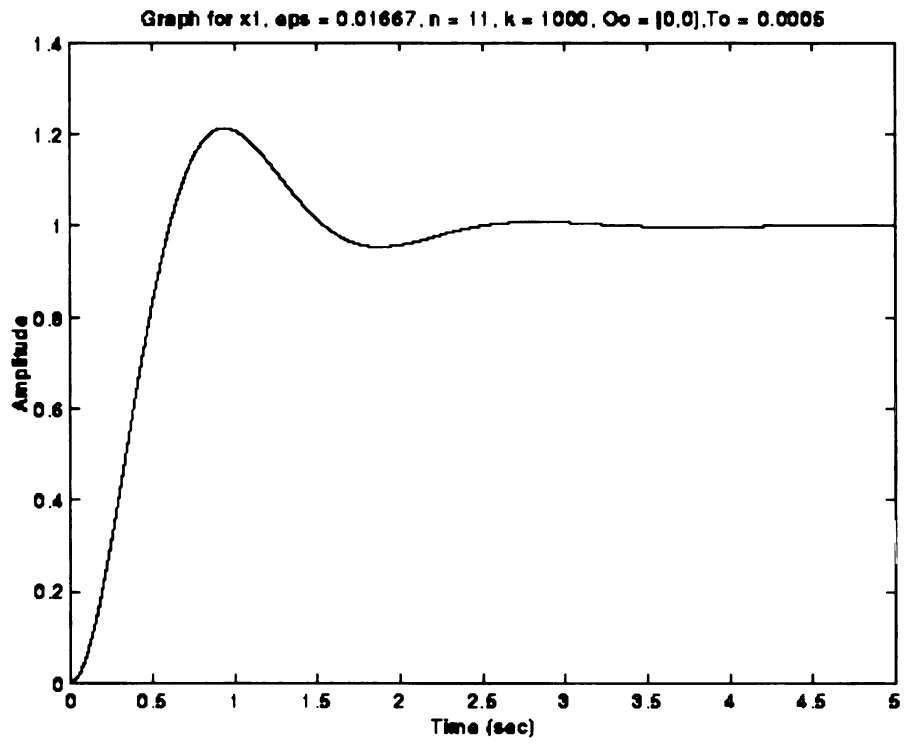
g1 = (g(n,:))';
end
Oo = g1(3:4);
uo = -K*Oo+N*r;
u = uo*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title(' Graph for O1, ε = 0.01667, n = 11, k = 1000, Oo = [0;0], To = 0.0005 ')
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

```

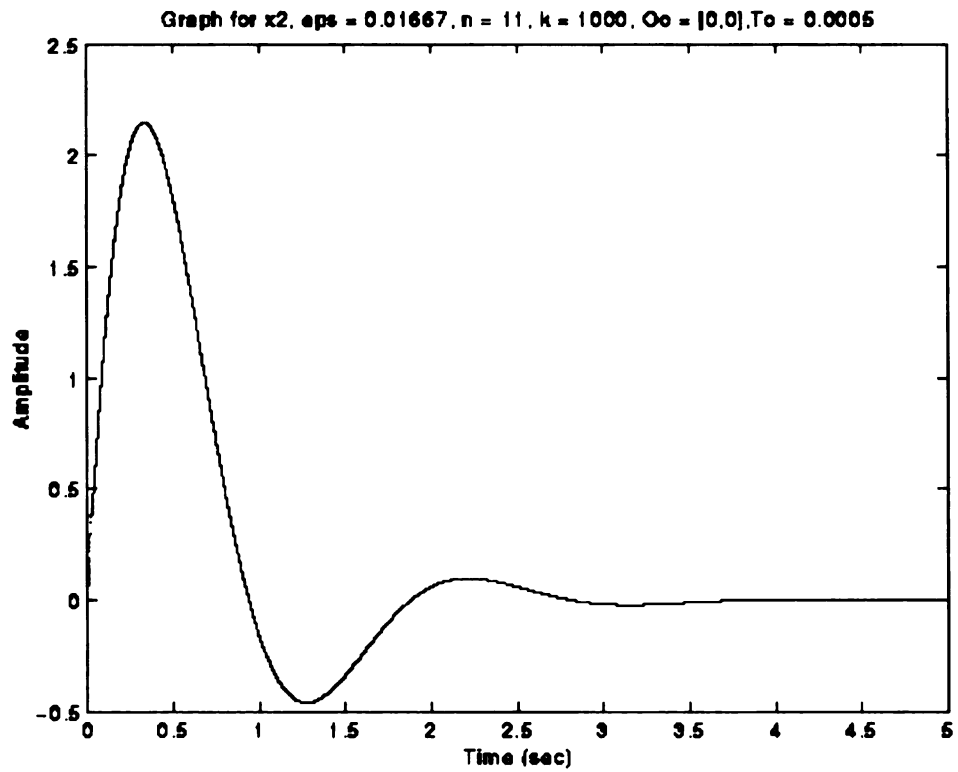
## APPENDIX C1



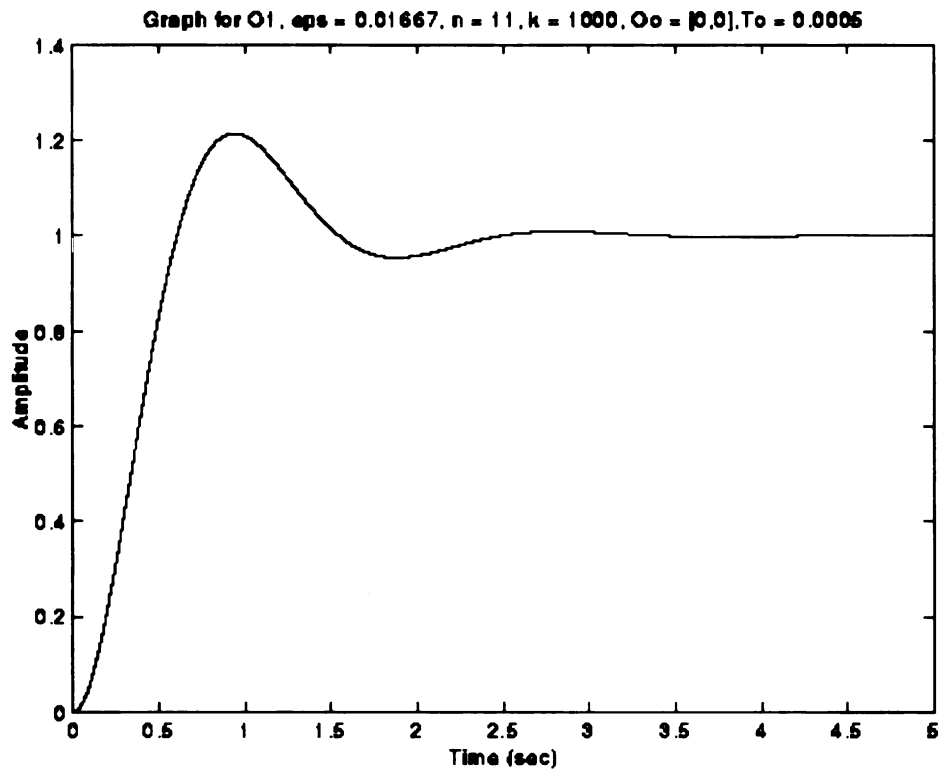
## APPENDIX C2



## APPENDIX C3

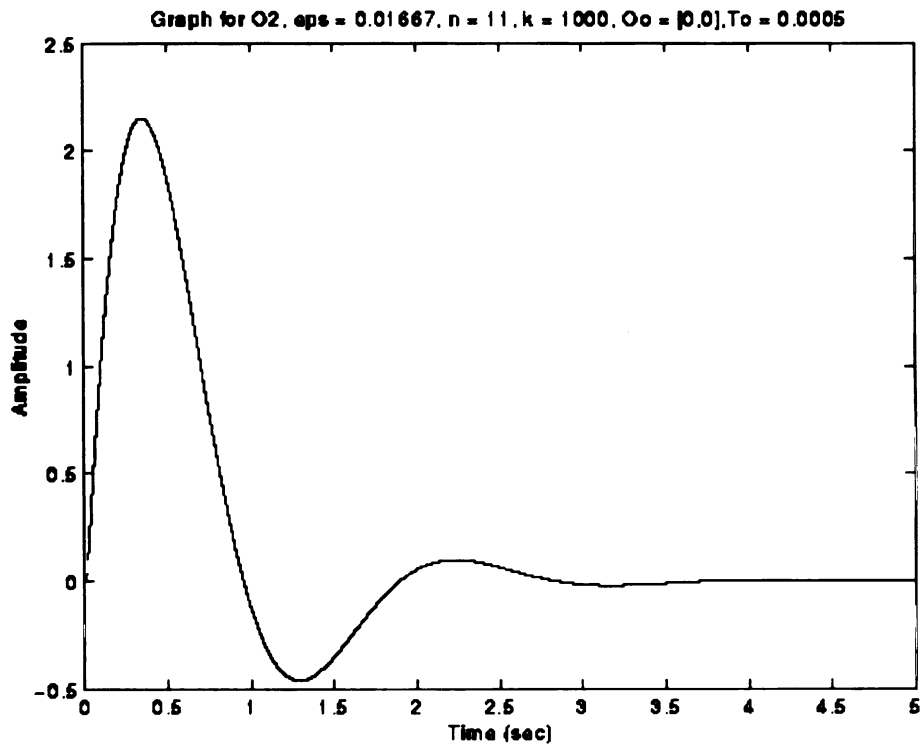


## APPENDIX C4





## APPENDIX C5



## APPENDIX C6

go =

0  
0  
0  
0

gg =

[]

uu =

[]

u =

0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287

h =

1.0e-03 \*

0  
0.0064  
0.0250  
0.0550  
0.0956  
0.1461  
0.2058  
0.2740  
0.3501  
0.4337  
0.5242

g =

0	0	0	0
0.0000	0.0253	0	0
0.0000	0.0489	0.0000	0.0000
0.0001	0.0709	0.0000	0.0001
0.0001	0.0913	0.0000	0.0002
0.0001	0.1103	0.0000	0.0003
0.0002	0.1281	0.0000	0.0006
0.0003	0.1446	0.0000	0.0009
0.0004	0.1599	0.0000	0.0014
0.0004	0.1742	0.0000	0.0019
0.0005	0.1875	0.0000	0.0027

gg =

0	0	0	0
0.0000	0.0253	0	0
0.0000	0.0489	0.0000	0.0000
0.0001	0.0709	0.0000	0.0001
0.0001	0.0913	0.0000	0.0002
0.0001	0.1103	0.0000	0.0003
0.0002	0.1281	0.0000	0.0006
0.0003	0.1446	0.0000	0.0009
0.0004	0.1599	0.0000	0.0014
0.0004	0.1742	0.0000	0.0019

uu =

0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287  
0.0287

u =

0.0289  
0.0289  
0.0289  
0.0289  
0.0289  
0.0289  
0.0289  
0.0289  
0.0289  
0.0289

h =

0.0005  
0.0006  
0.0007  
0.0008  
0.0009  
0.0011  
0.0012  
0.0013  
0.0014  
0.0016  
0.0017



**APPENDIX D****Multirate Implementation - Under State Feedback**

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.01667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [0;0];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1000;
for k = 1:kmax
    if k = 1
        g1 = go;
    elseif k > 1

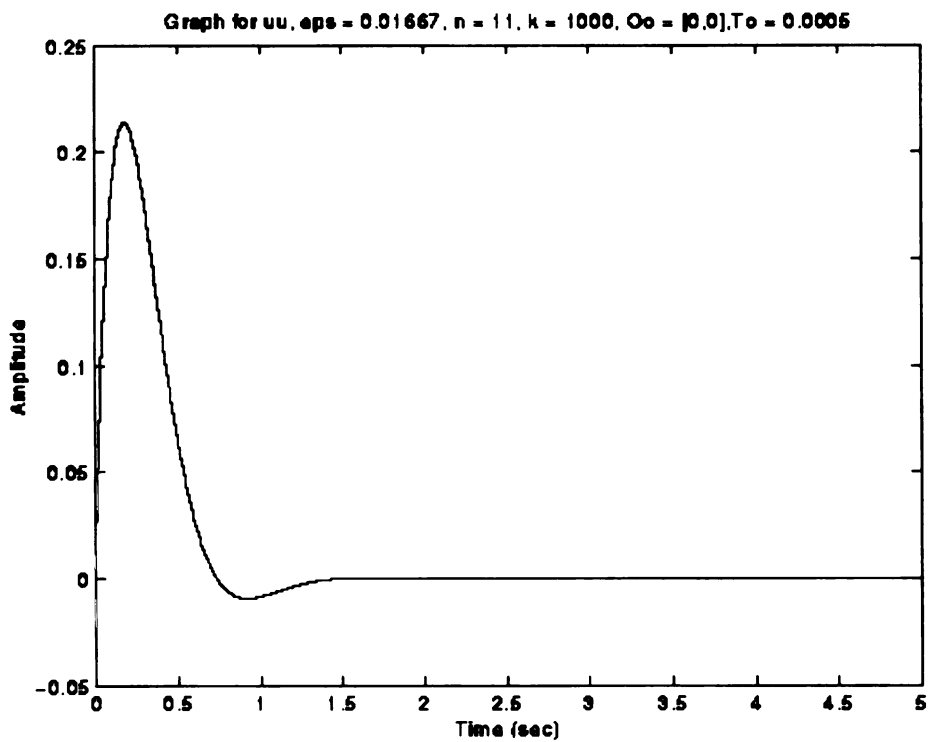
```

```

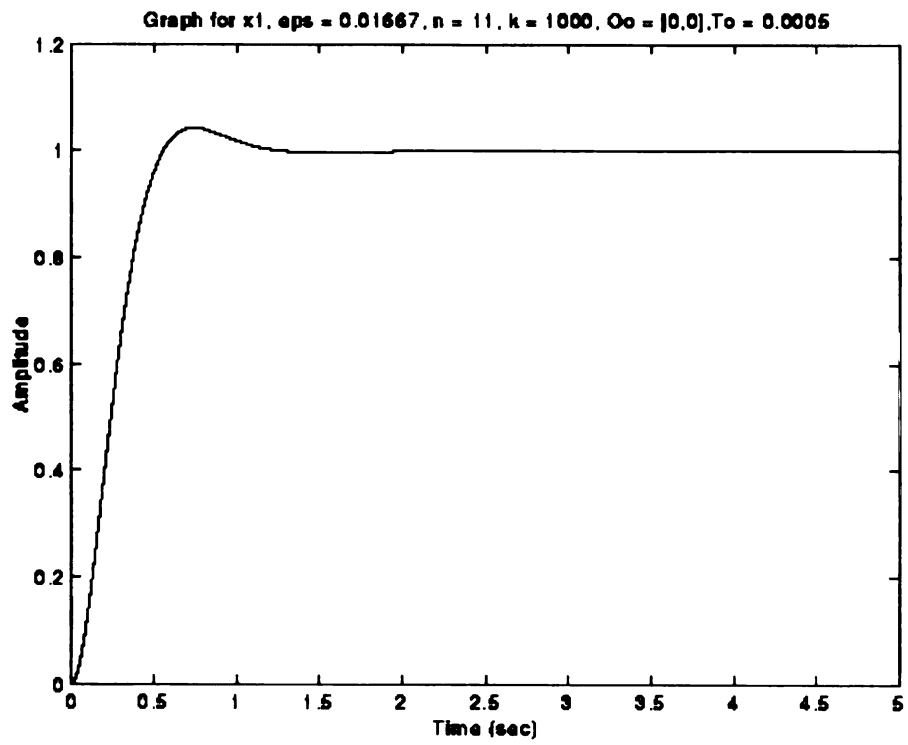
g1 = (g(n,:))';
end
x0 = g1(1:2);
u0 = -K*x0+N*r;
u = u0*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title( ' Graph for O1, ε = 0.01667, n = 11, k = 1000, Oo = [0;0], To = 0.0005 ' )
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

```

## APPENDIX D1

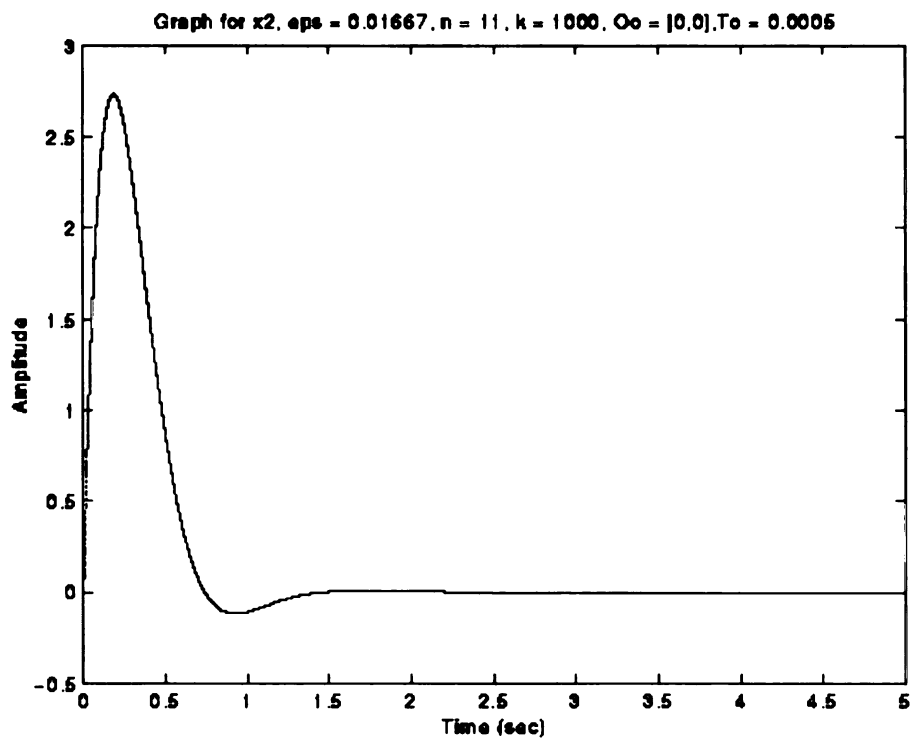


## APPENDIX D2

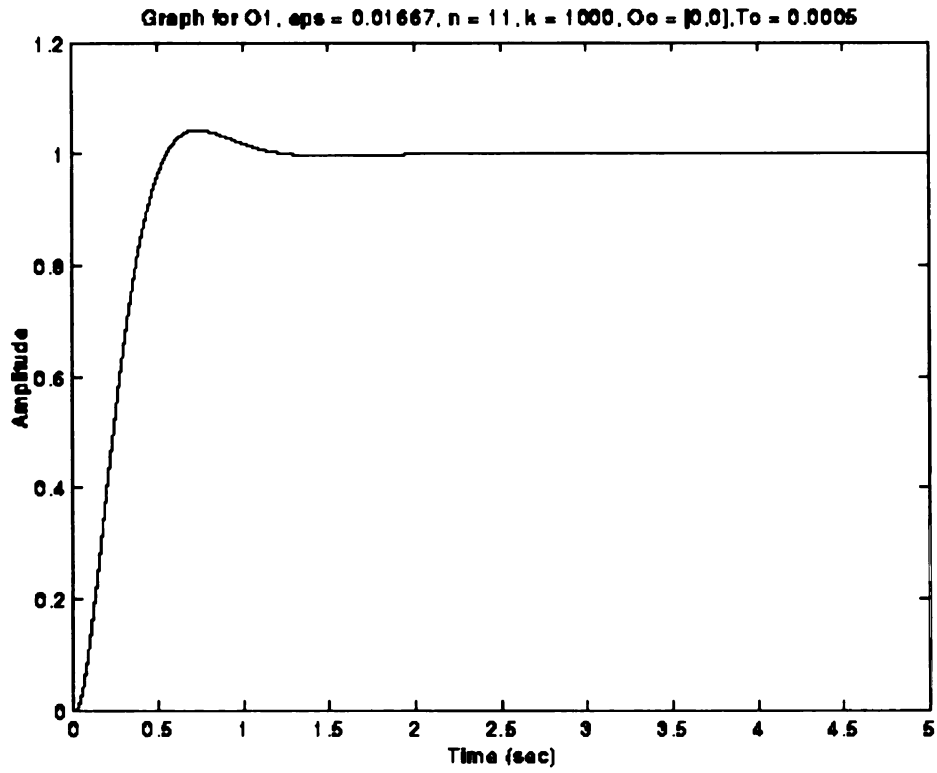




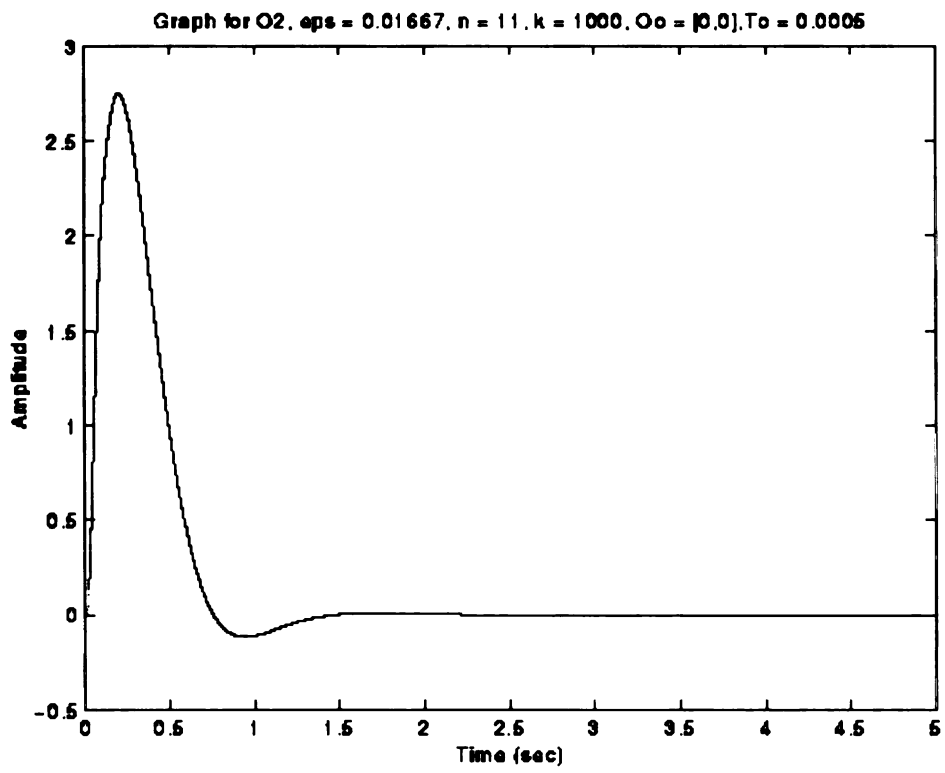
## APPENDIX D3



## APPENDIX D4



## APPENDIX D5



**APPENDIX E****Multirate Implementation - Behavior in First Cycle**

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.01667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [0;0];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1;
for k = 1:kmax
    if k = 1
        g1 = go;
    elseif k > 1

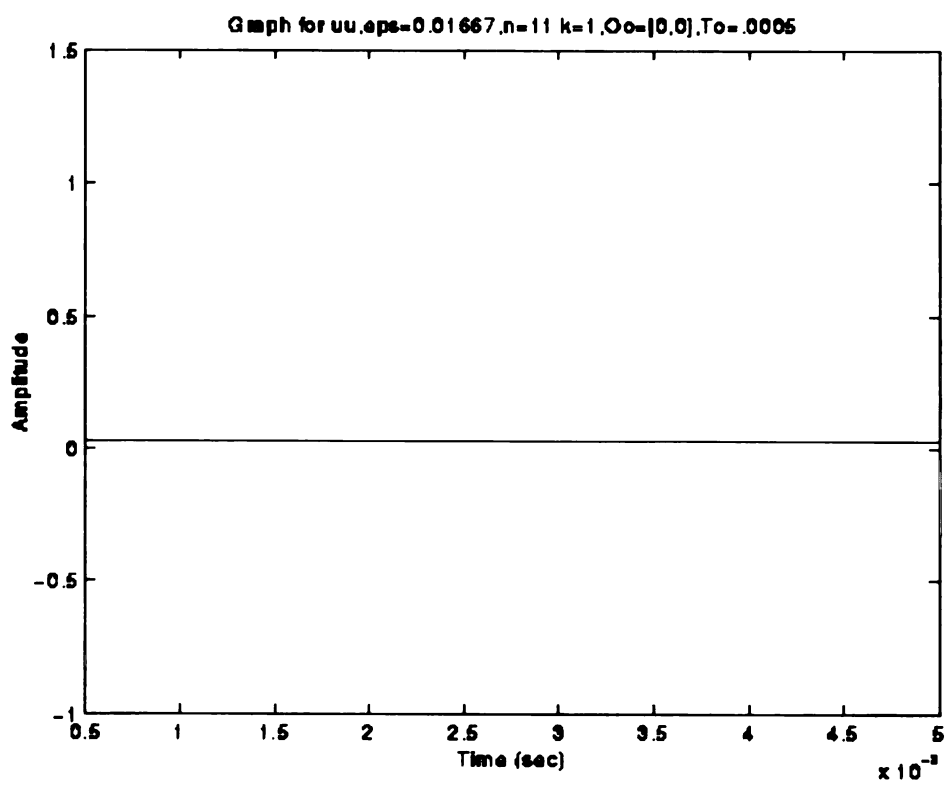
```

```

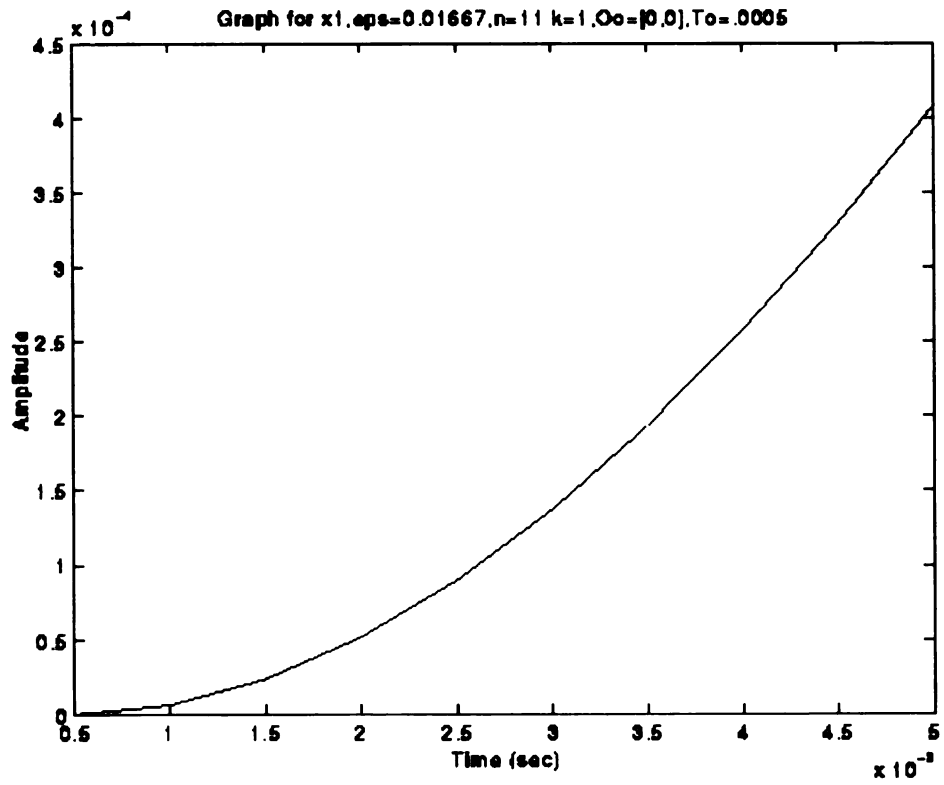
g1 = (g(n,:))';
end
Oo = g1(3:4);
uo = -K*Oo+N*r;
u = uo*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title( ' Graph for O1, ε = 0.01667, n = 11, k = 1, Oo = [0;0], To = 0.0005 ' )
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

```

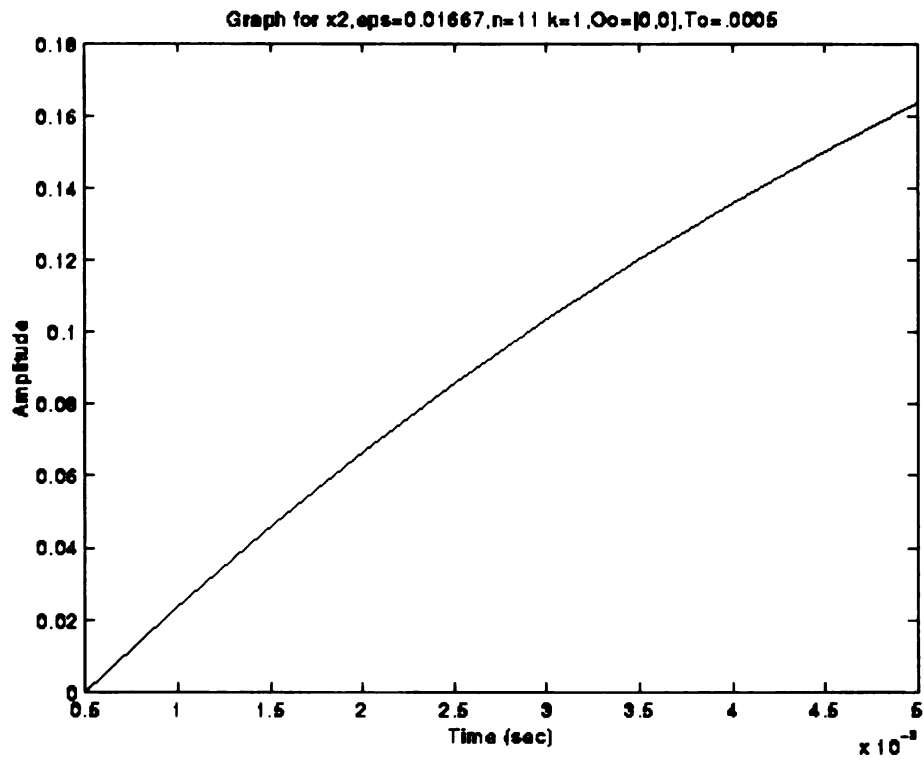
## APPENDIX E1



## APPENDIX E2

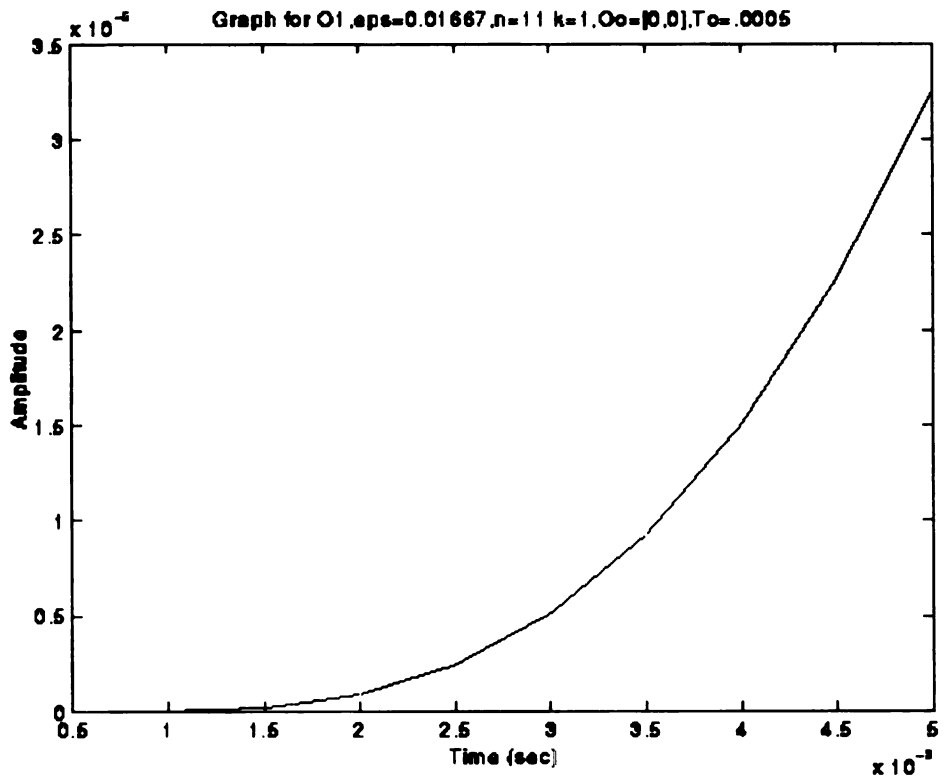


## APPENDIX E3

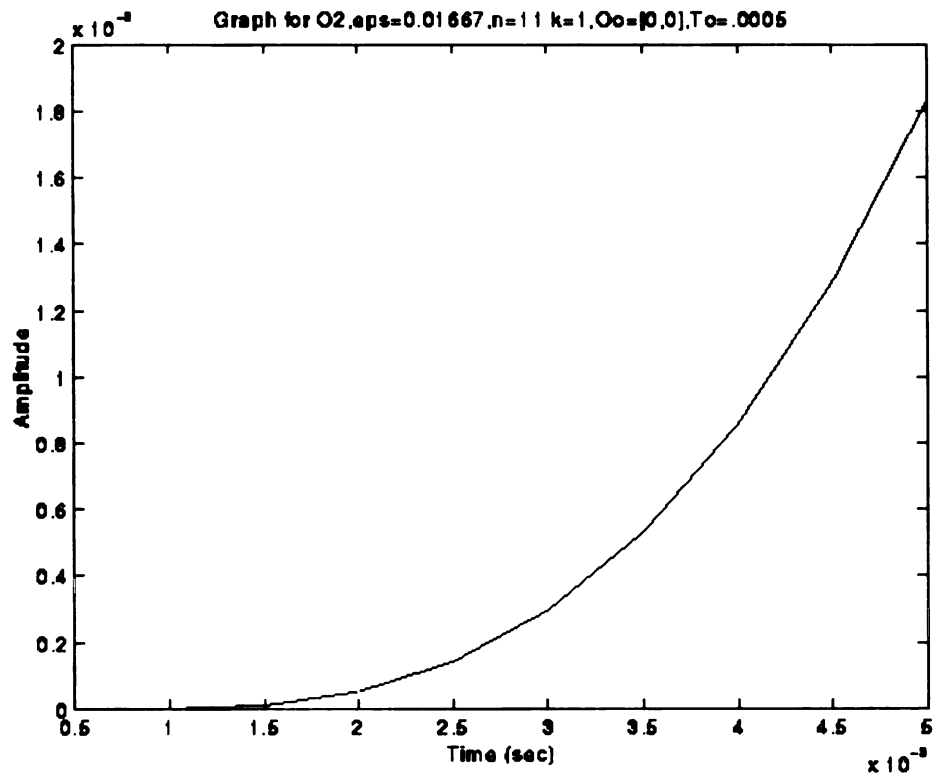




### APPENDIX E4



## APPENDIX E5



**APPENDIX F****Multirate Implementation Under State Feedback Control  
For Different Initial Conditions**

```

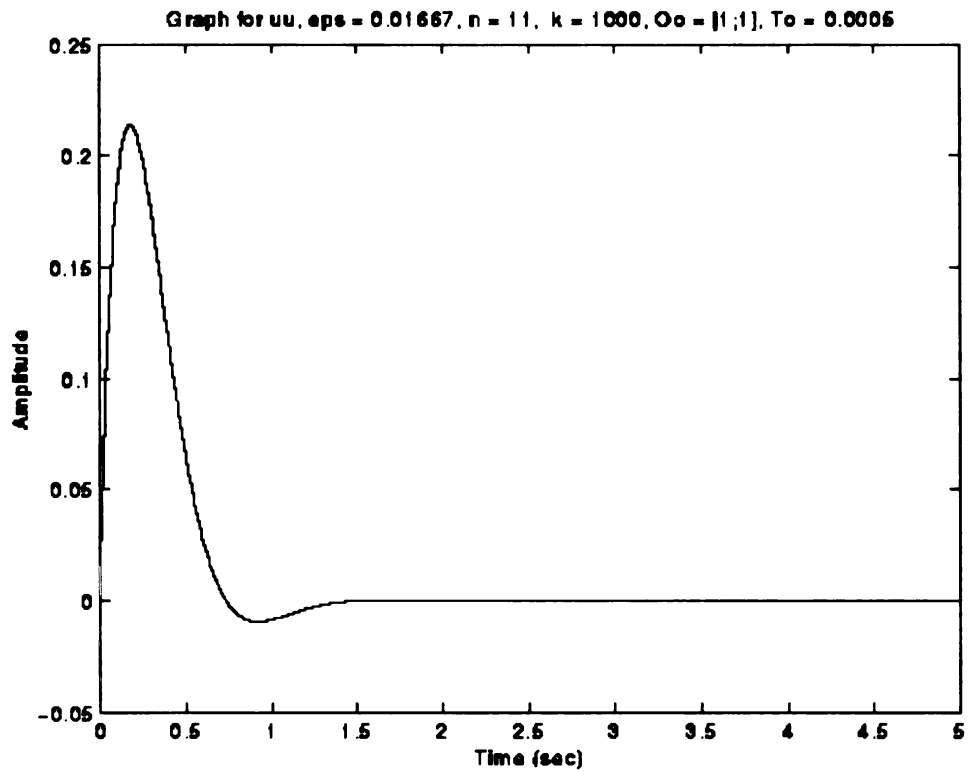
A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.01667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [1;1];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1000;
for k = 1:kmax
    if k = 1
        g1 = go;

```

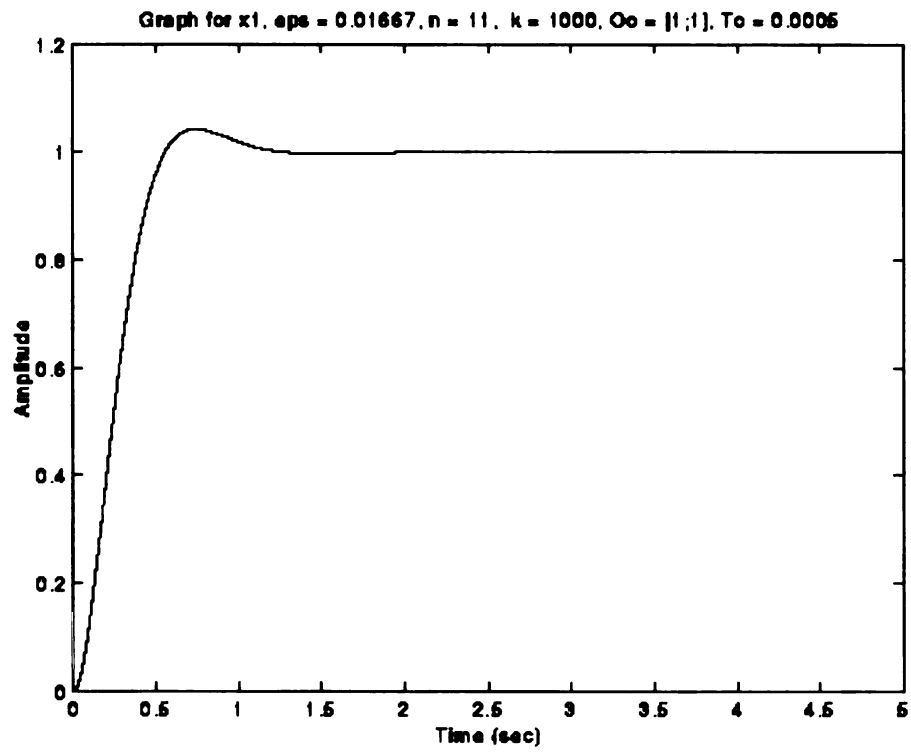
```

elseif k > 1
    g1 = (g(n,:))';
    end
    x0 = g1(1:2);
    u0 = -K*x0+N*r;
    u = u0*ones(n,1);
    [h,g] = dlsim(aa, bb, cc, dd, u, g1);
    gg = [gg;g(1:n-1,:)];
    uu = [uu;u(1:n-1)];
    x = gg(:,1:2);
    O = gg(:,3:4);
    x1 = gg(:,1);
    x2 = gg(:,2);
    O1 = gg(:,3);
    O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title( ' Graph for O1,  $\varepsilon = 0.01667$ ,  $n = 11$ ,  $k = 1000$ ,  $O_0 = [1;1]$ ,  $To = 0.0005$  ' )
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

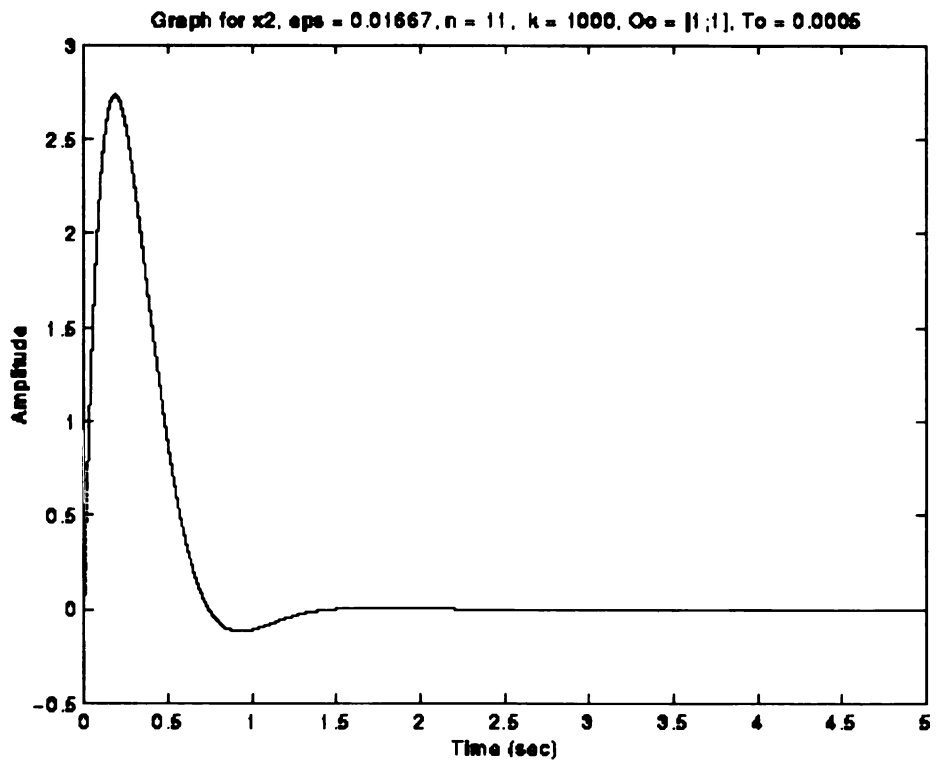
```

**APPENDIX F1**

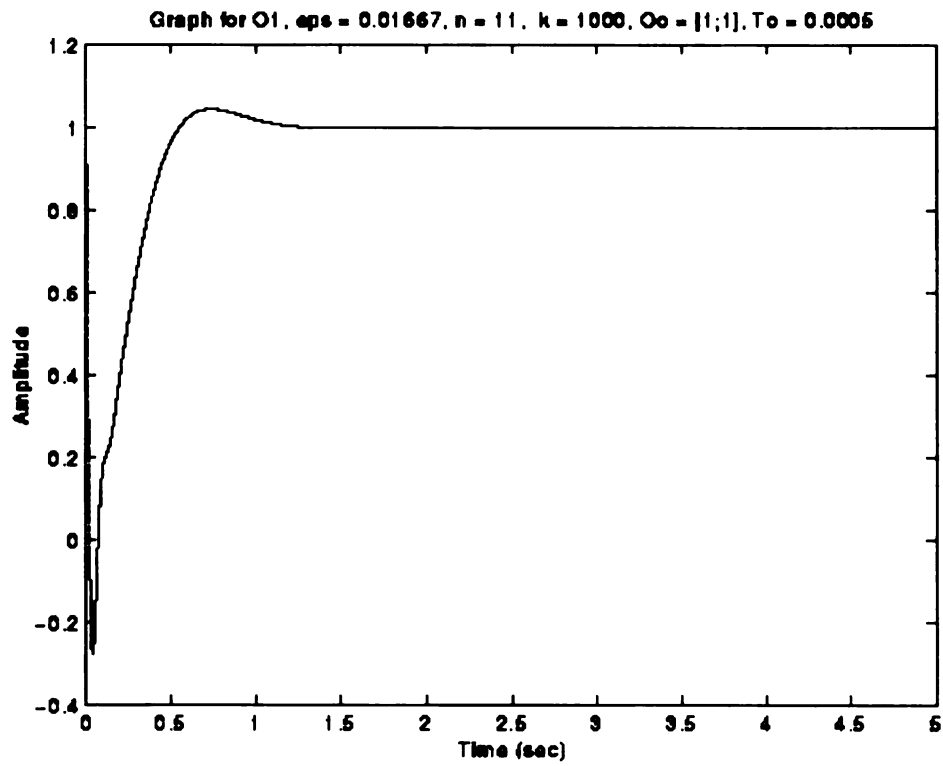
## APPENDIX F2



## APPENDIX F3

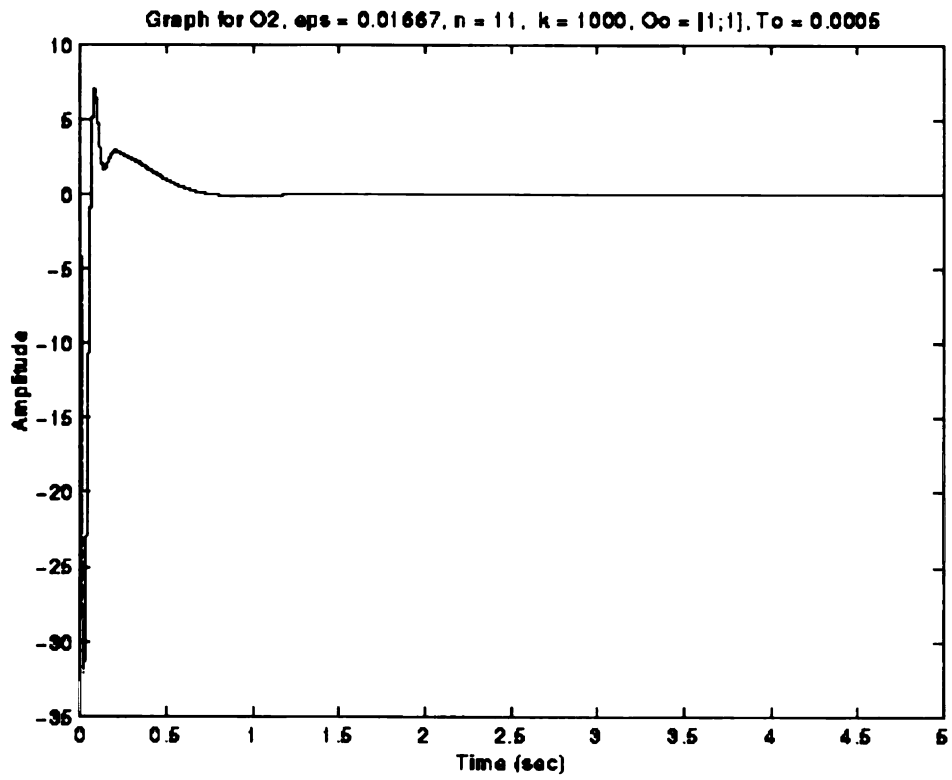


## APPENDIX F4





## APPENDIX F5



**APPENDIX G****Multirate Implementation Under Output Feedback Control  
For Different Initial Conditions**

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.01667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [1;1];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1000;
for k = 1:kmax
    if k = 1
        g1 = go;

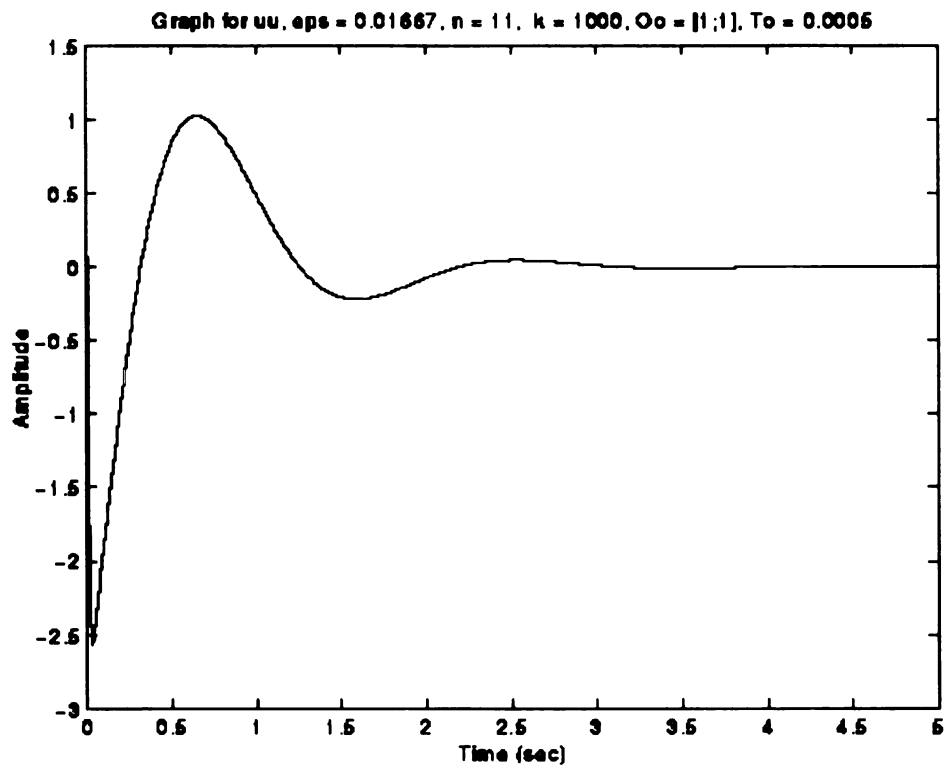
```

```

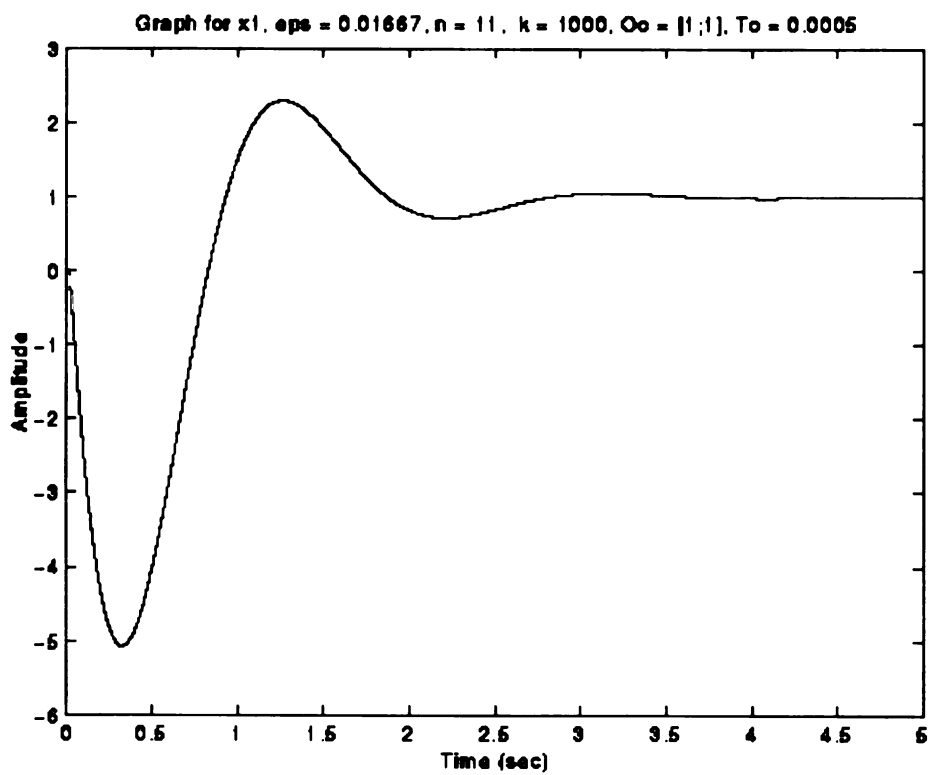
elseif k > 1
g1 = (g(n,:))';
end
Oo = g1(3:4);
uo = -K*Oo+N*r;
u = uo*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title( ' Graph for O1, ε = 0.01667, n = 11, k = 1000, Oo = [1;1], To = 0.0005 ' )
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

```

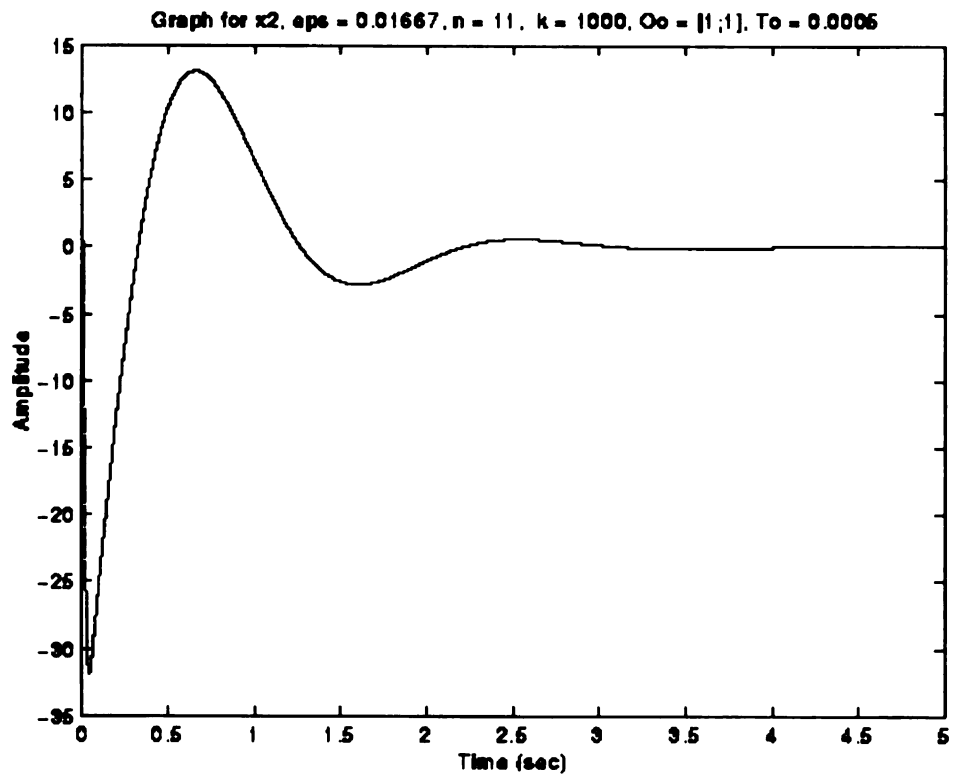
## APPENDIX G1



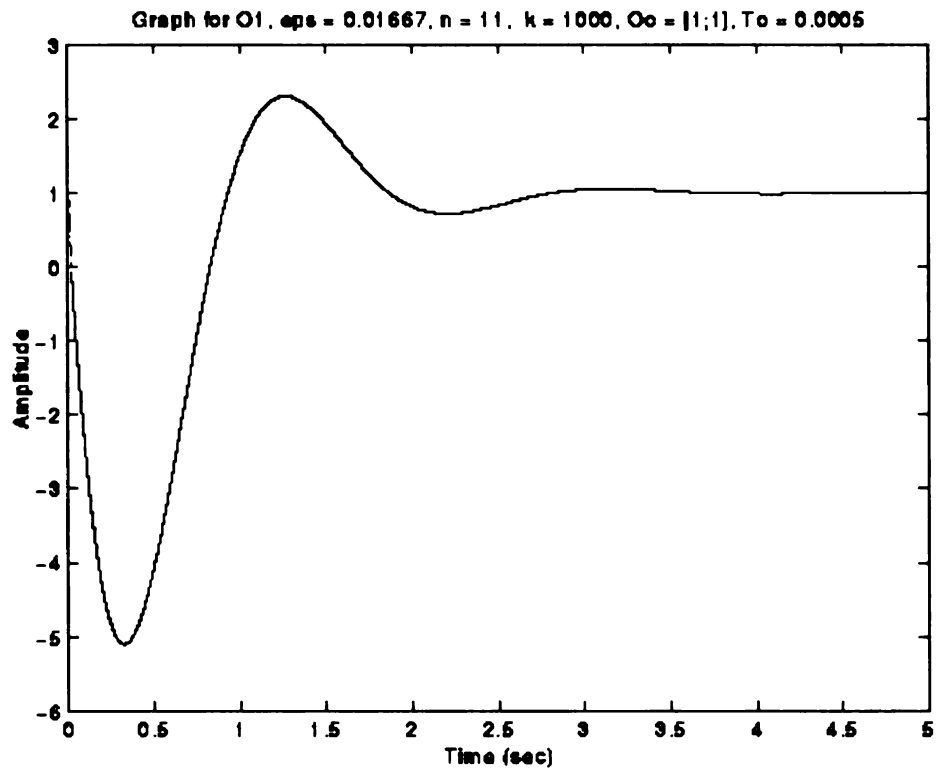
## APPENDIX G2



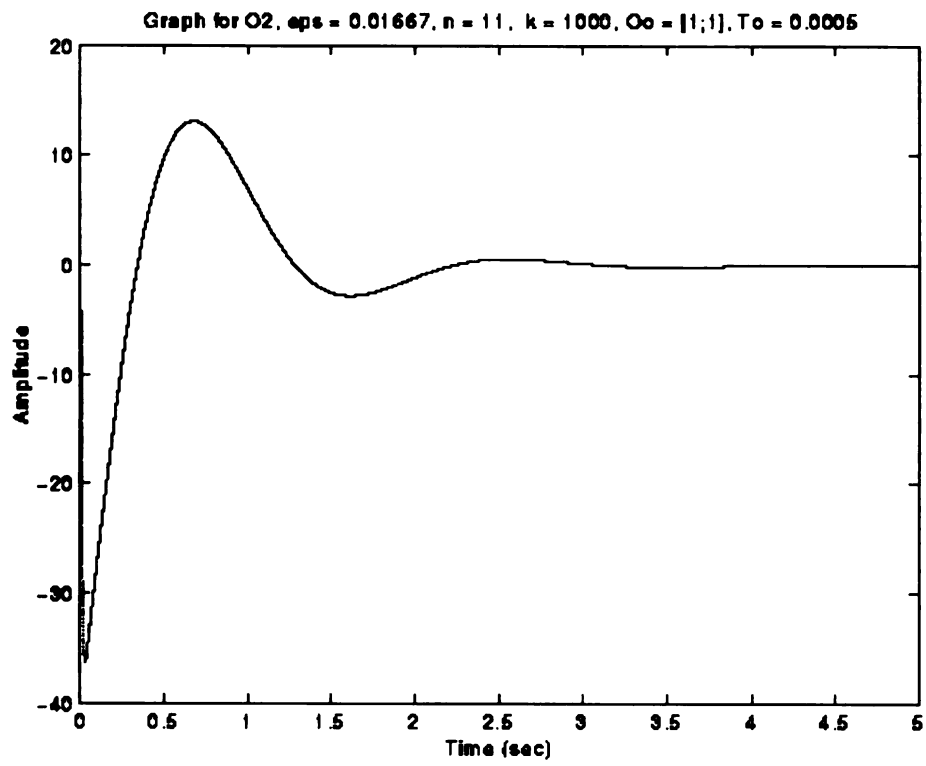
## APPENDIX G3



## APPENDIX G4



## APPENDIX G5





**APPENDIX H****Multirate Implementation - Behavior in First Cycle  
For Different Initial Conditions**

```

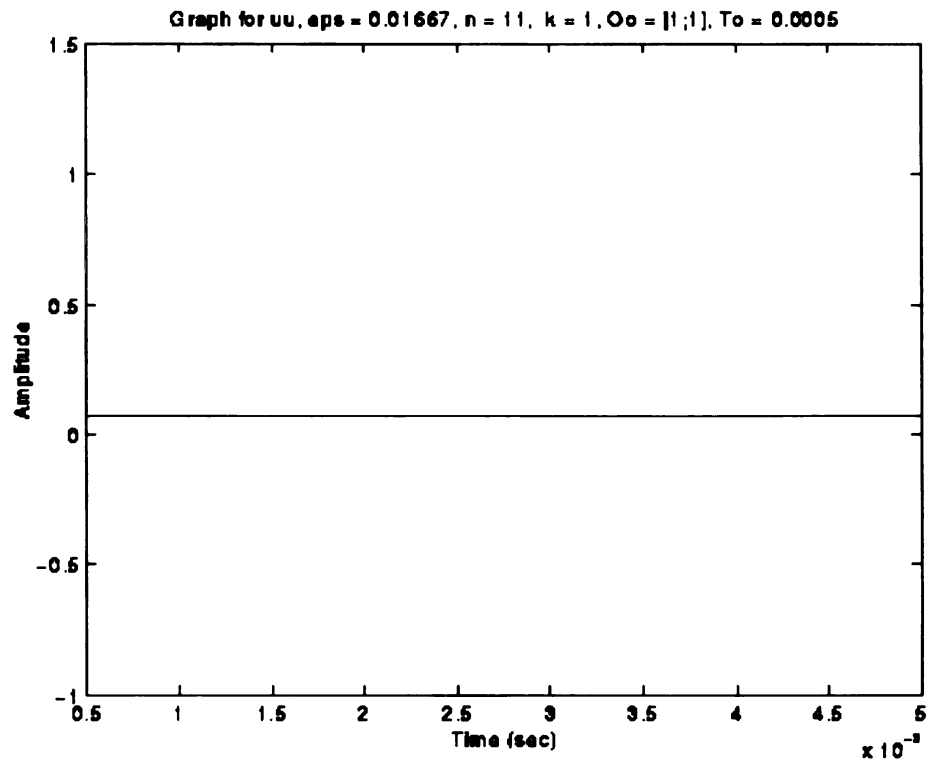
A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.01667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [1;1];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1;
for k = 1:kmax
    if k = 1
        g1 = go;

```

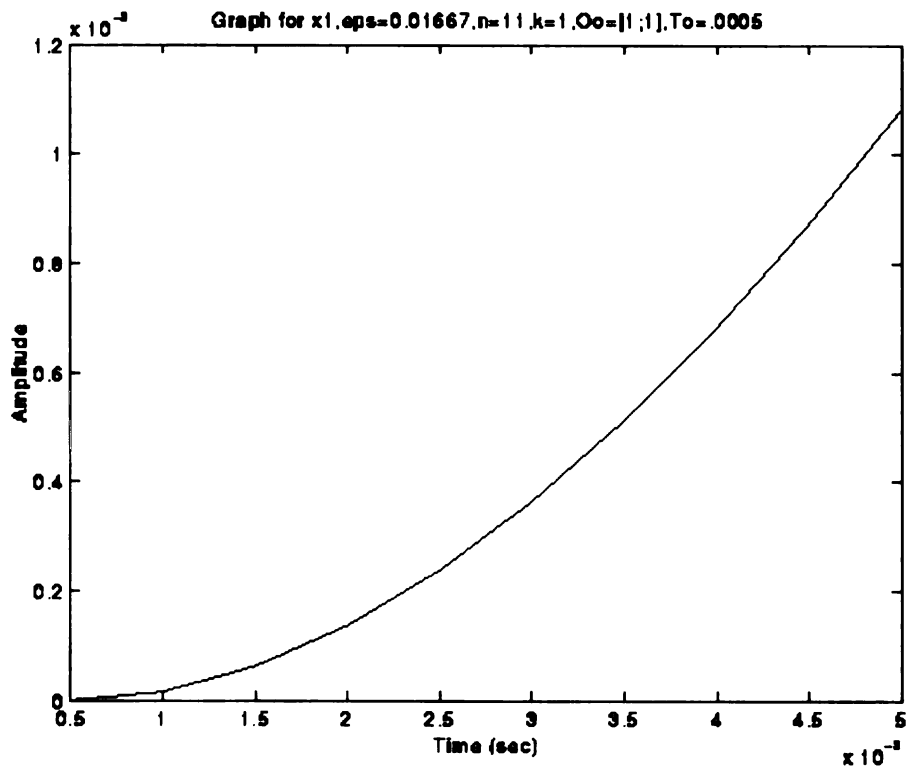
```

elseif k > 1
    g1 = (g(n,:))';
    end
    Oo = g1(3:4);
    uo = -K*Oo+N*r;
    u = uo*ones(n,1);
    [h,g] = dlsim(aa, bb, cc, dd, u, g1);
    gg = [gg;g(1:n-1,:)];
    uu = [uu;u(1:n-1)];
    x = gg(:,1:2);
    O = gg(:,3:4);
    x1 = gg(:,1);
    x2 = gg(:,2);
    O1 = gg(:,3);
    O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title(' Graph for O1,  $\varepsilon = 0.01667$ ,  $n = 11$ ,  $k = 1$ ,  $O_o = [1;1]$ ,  $To = 0.0005$  ')
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

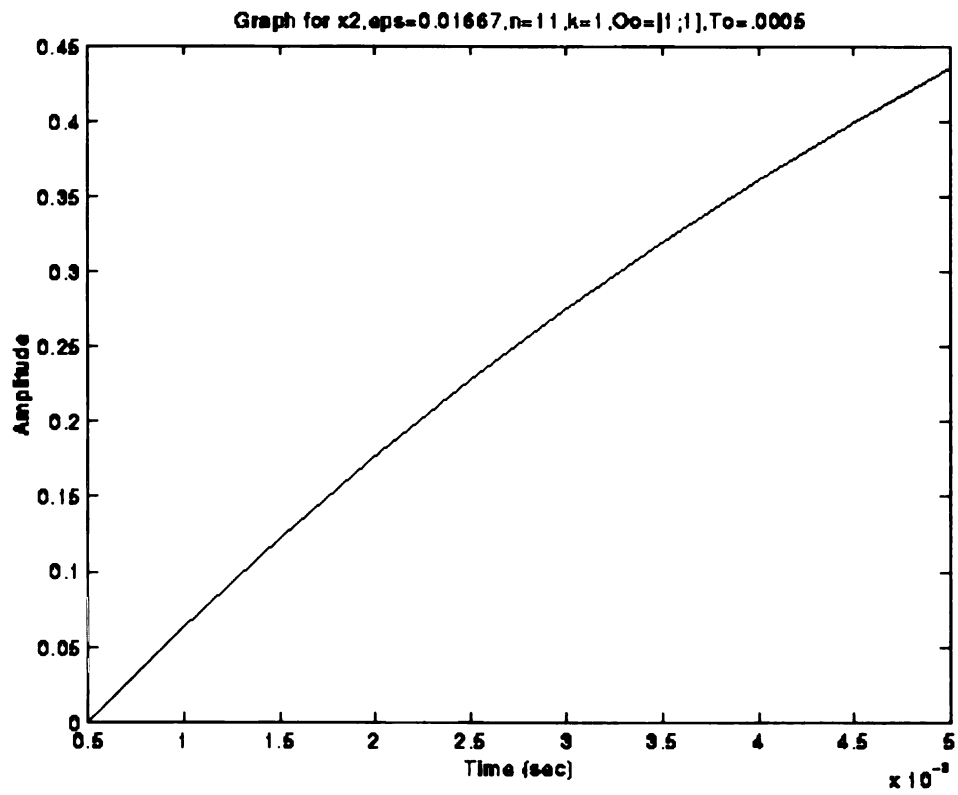
```

**APPENDIX H1**

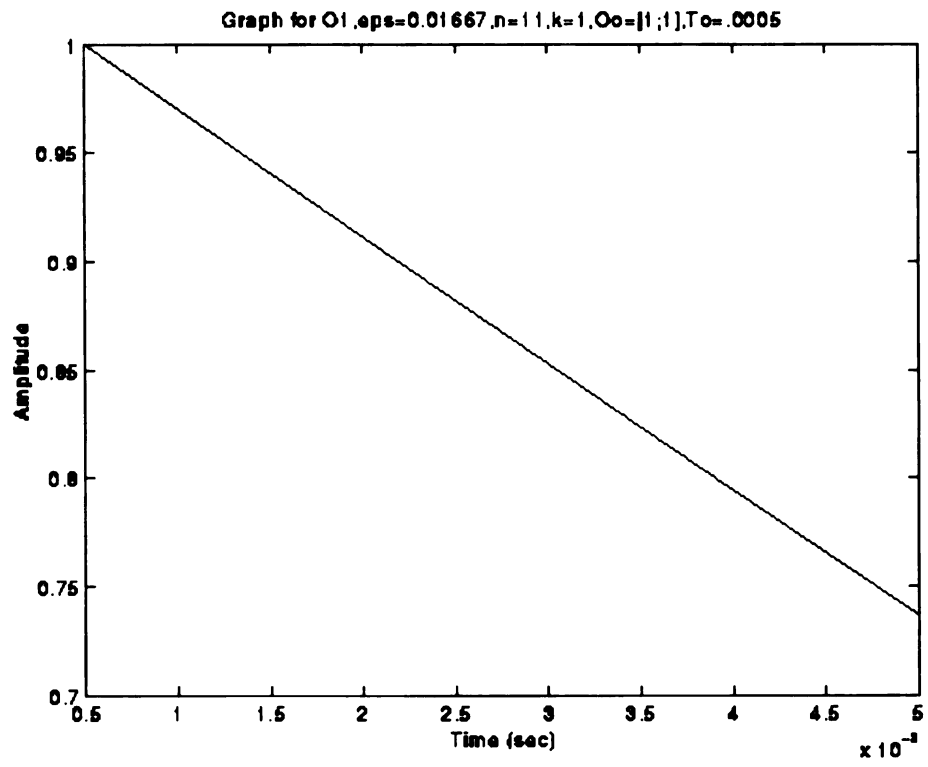
## APPENDIX H2



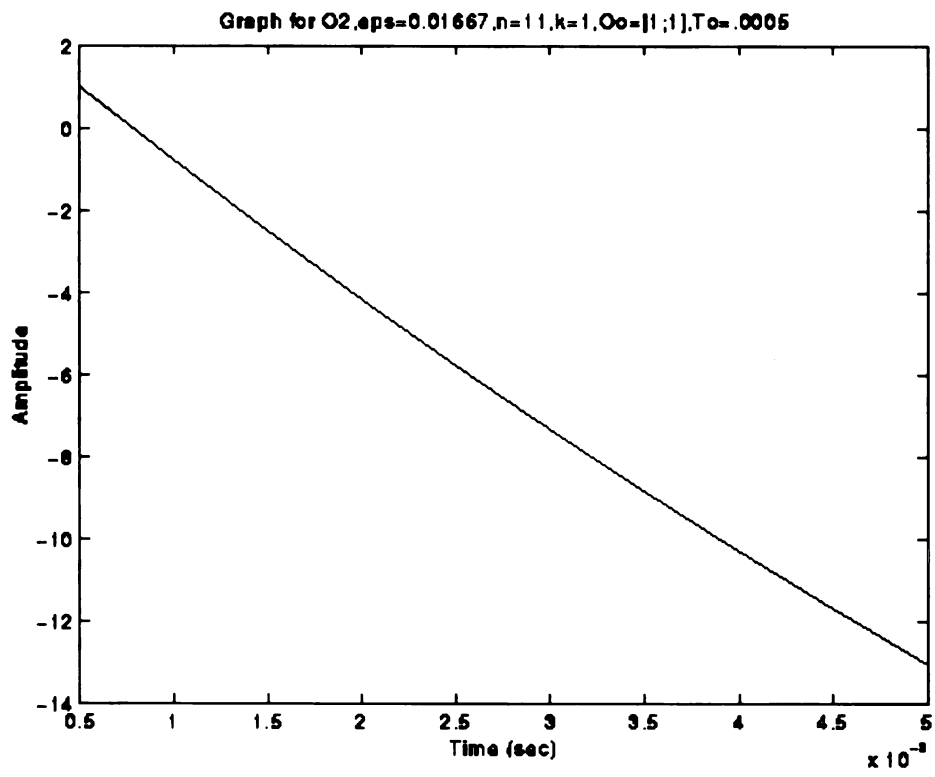
## APPENDIX H3



## APPENDIX H4



## APPENDIX H5



**APPENDIX I****Multirate Implementation - Control Saturated**

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.01667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [1;1];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
m = 0.5;
n = 11;
kmax = 1000;
for k = 1:kmax
    if k = 1
        g1 = go;

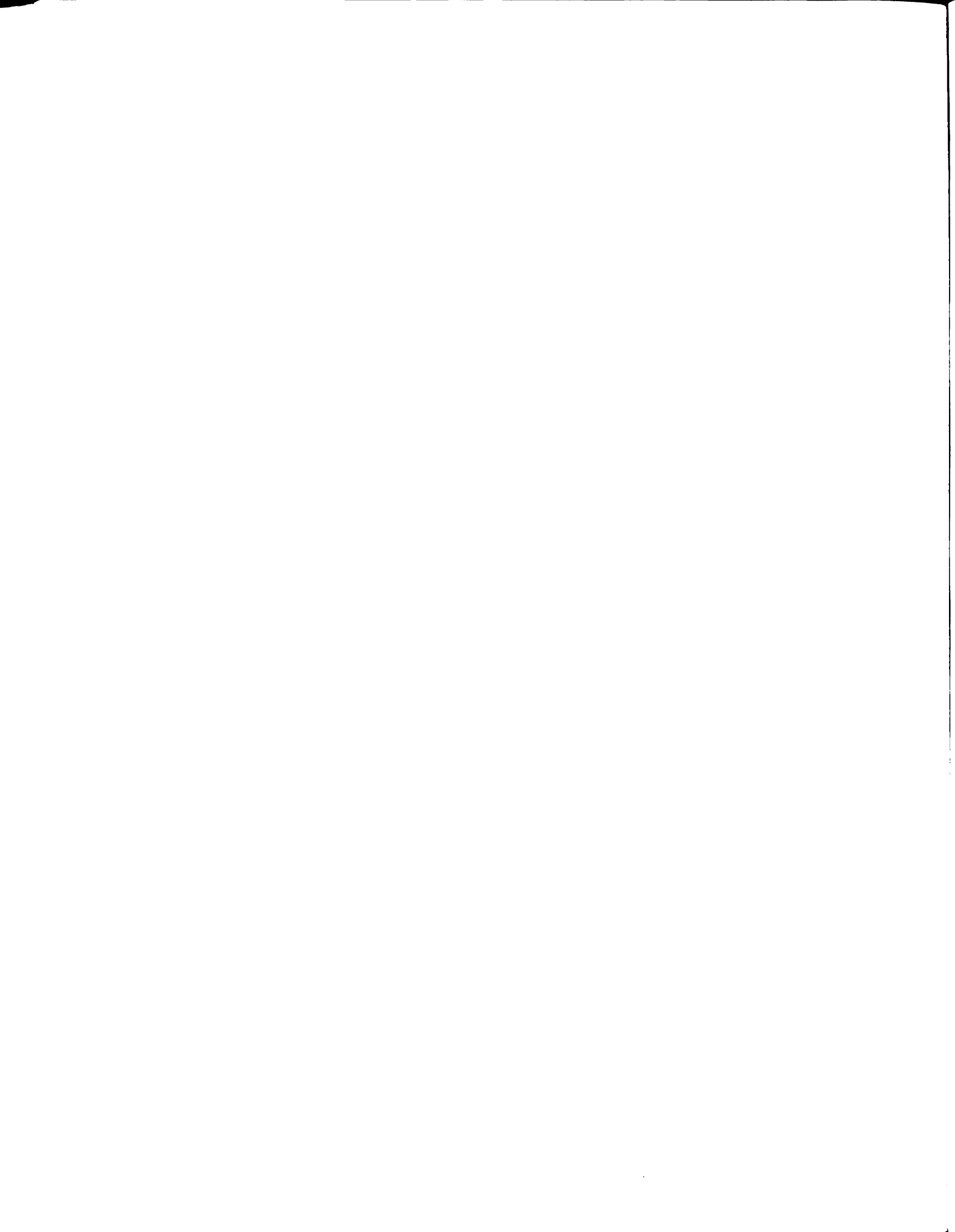
```

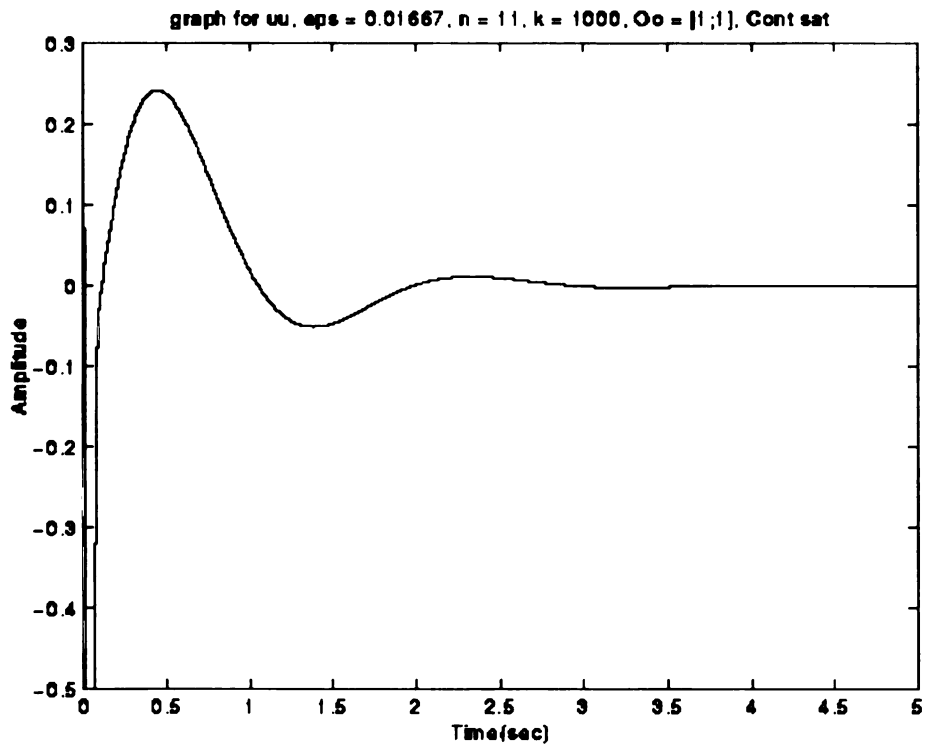


```

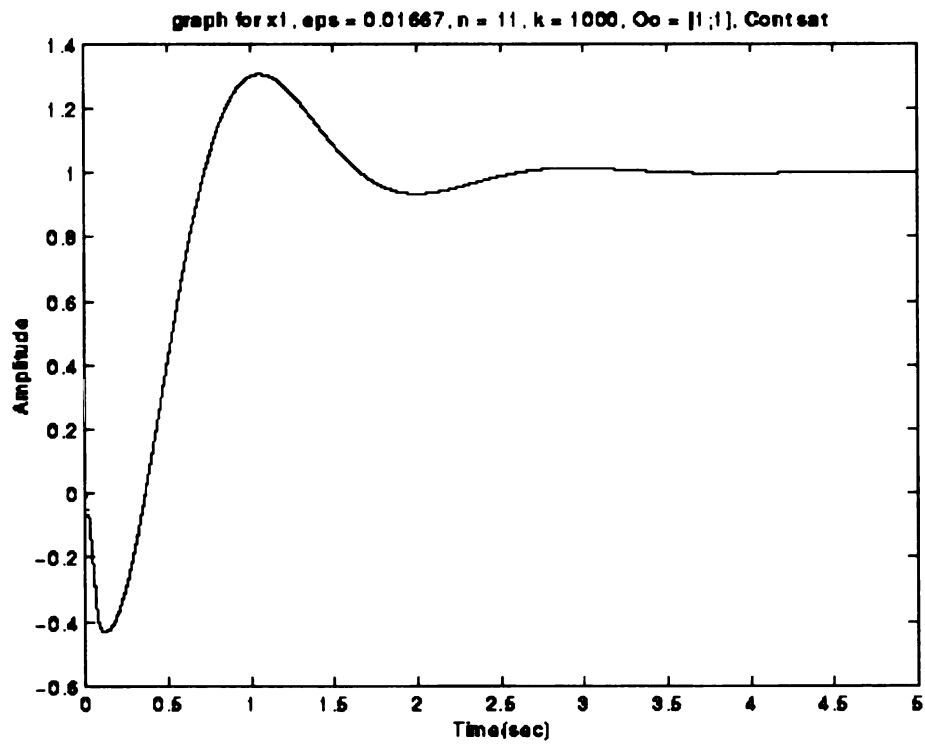
elseif k > 1
g1 = (g(n,:))';
end
Oo = g1(3:4);
uo = -K*Oo+N*r;
    if abs(uo) <= m
        u1 = uo;
    else if abs(uo) > m
        u1 = m*sign(uo);
    end
u = u1*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title(' Graph for O1, ε = 0.01667, n = 11, k = 1000, Oo = [1;1], To = 0.0005 ')
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

```

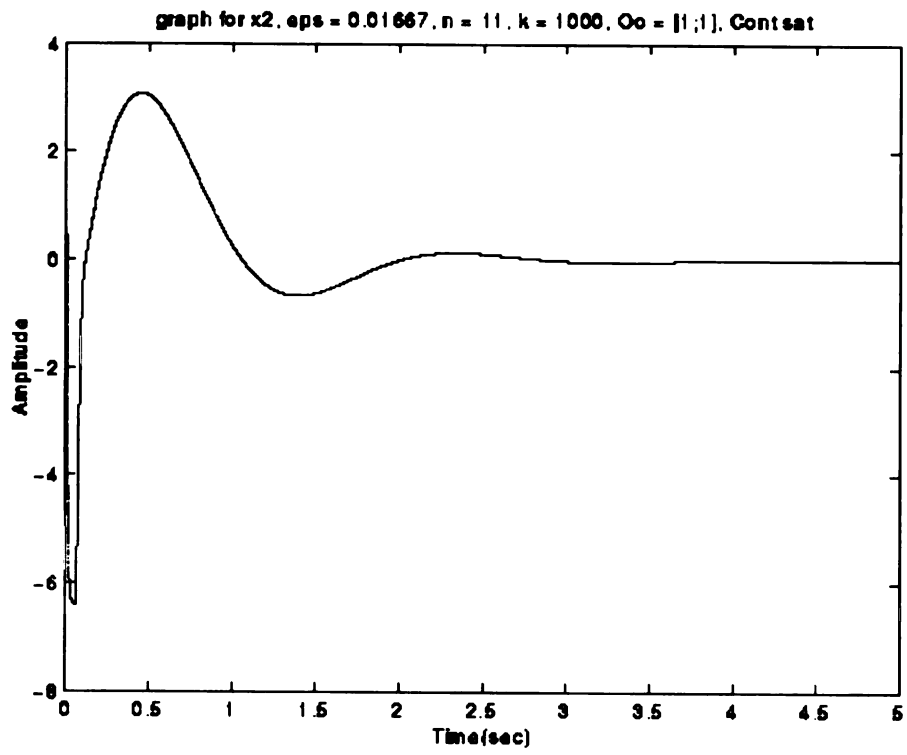


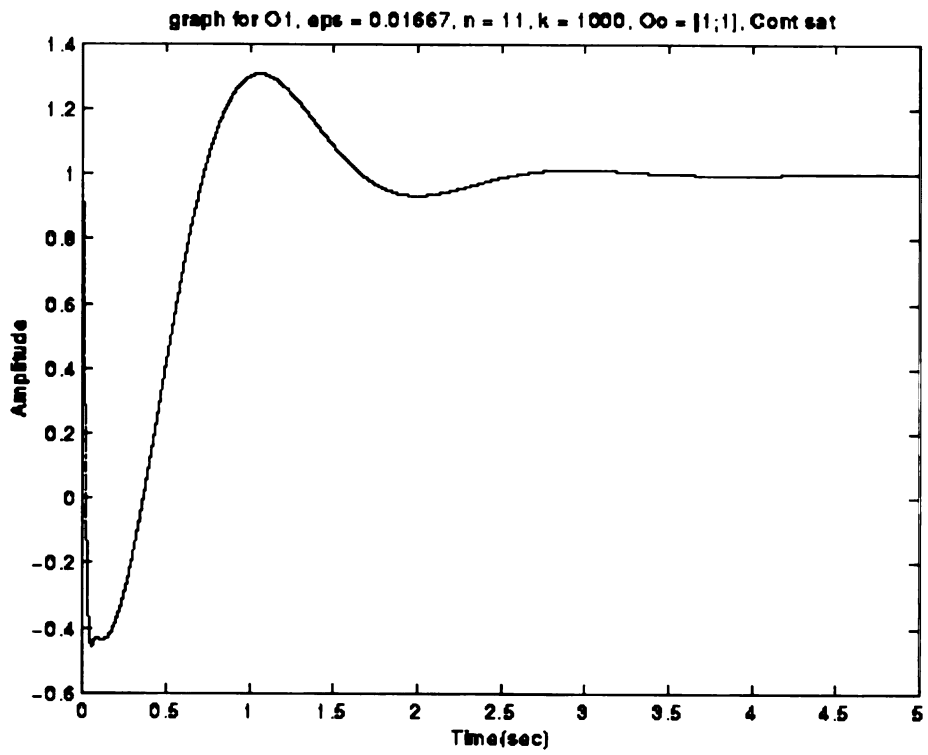
**APPENDIX II**

## APPENDIX I2

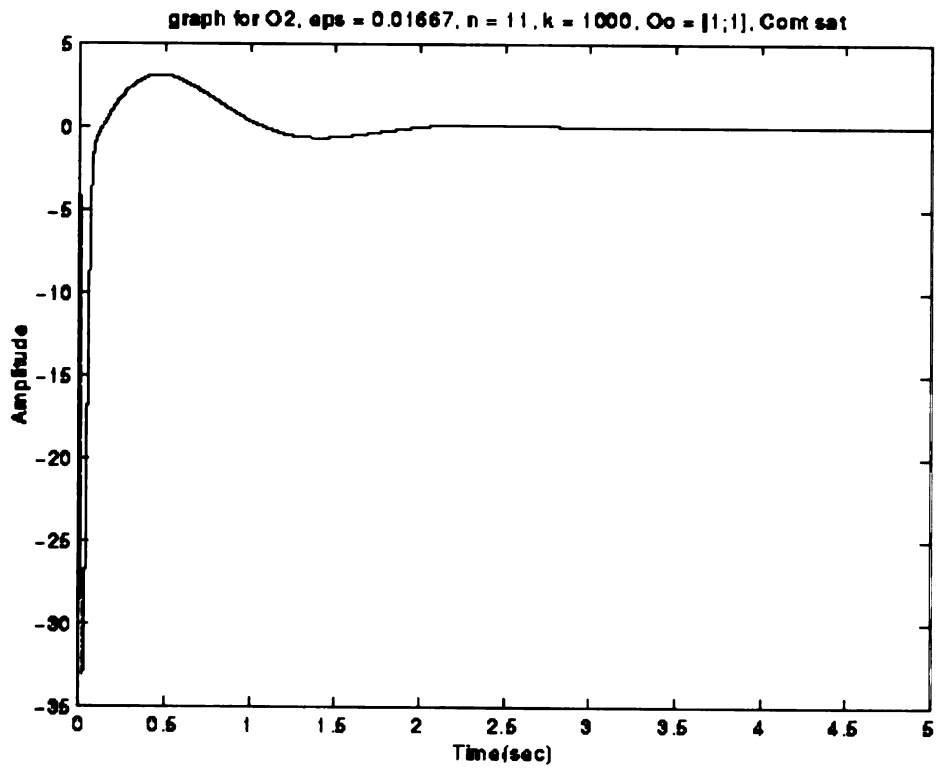


## APPENDIX I3



**APPENDIX I4**

## APPENDIX I5



**APPENDIX J**

**Multirate Implementation Under Output Feedback Control  
With Different Value of  $\varepsilon$**

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.001667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [0;0];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1000;
for k = 1:kmax
    if k = 1
        g1 = go;

```

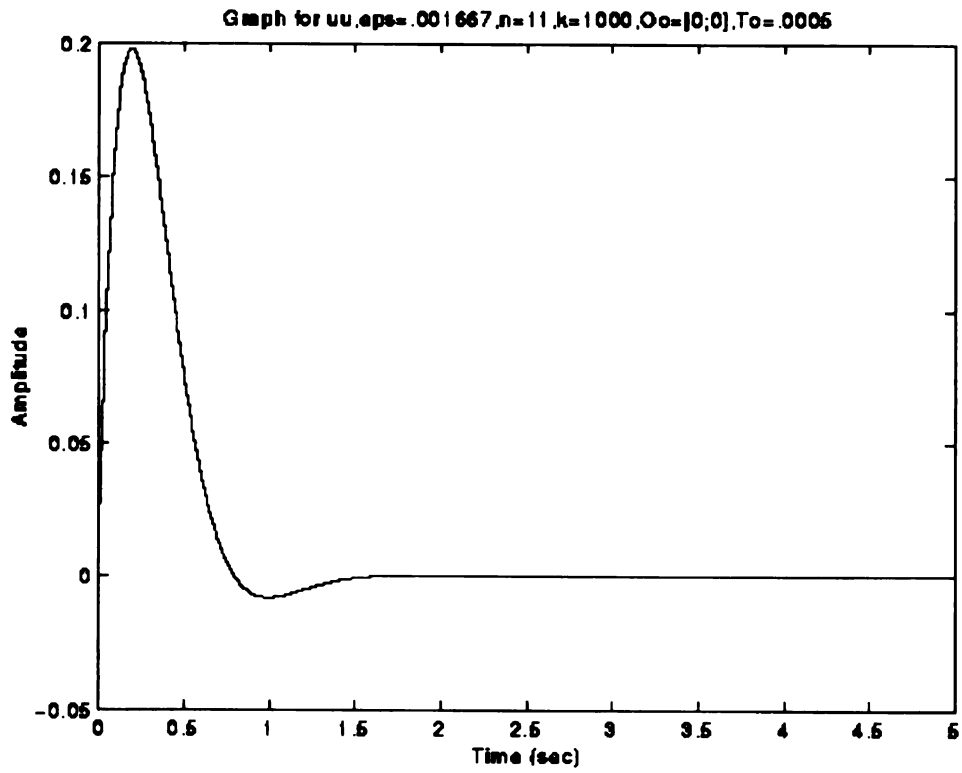


```

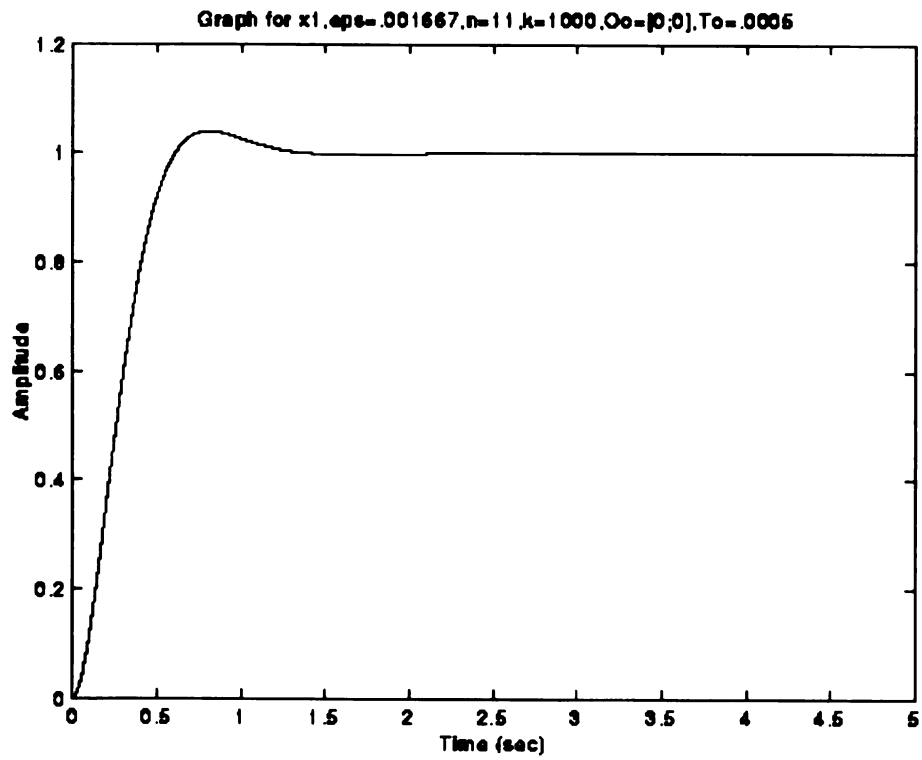
elseif k > 1
g1 = (g(n,:))';
end
Oo = g1(3:4);
uo = -K*Oo+N*r;
u = uo*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title( ' Graph for O1, ε = 0.001667, n = 11, k = 1000, Oo = [0;0], To = 0.0005 ' )
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

```

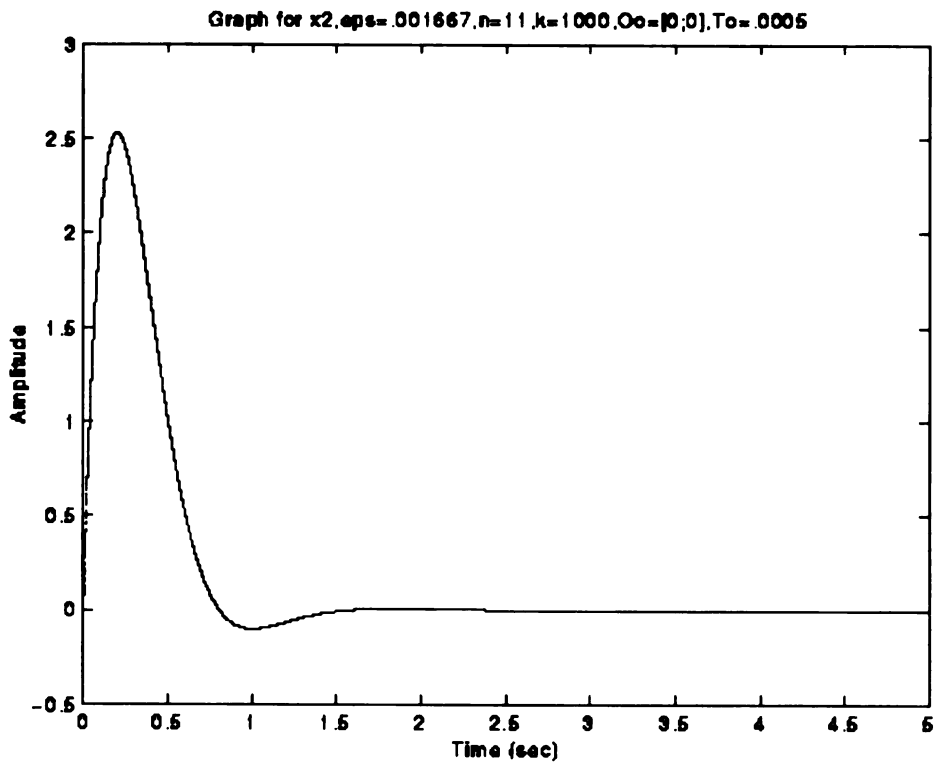
## APPENDIX J1



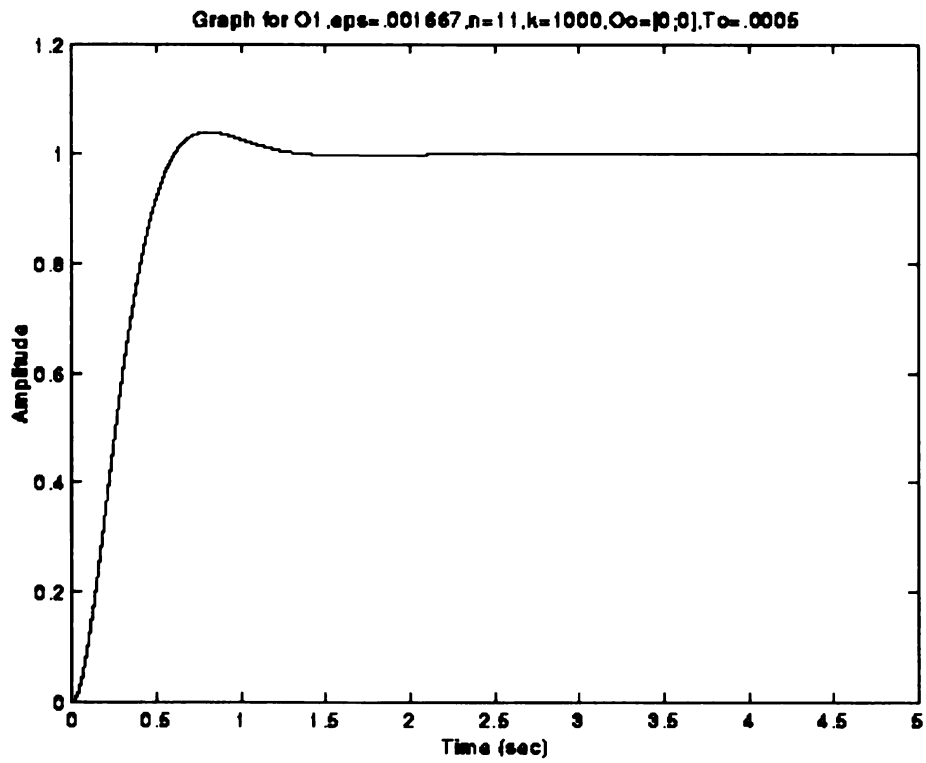
## APPENDIX J2



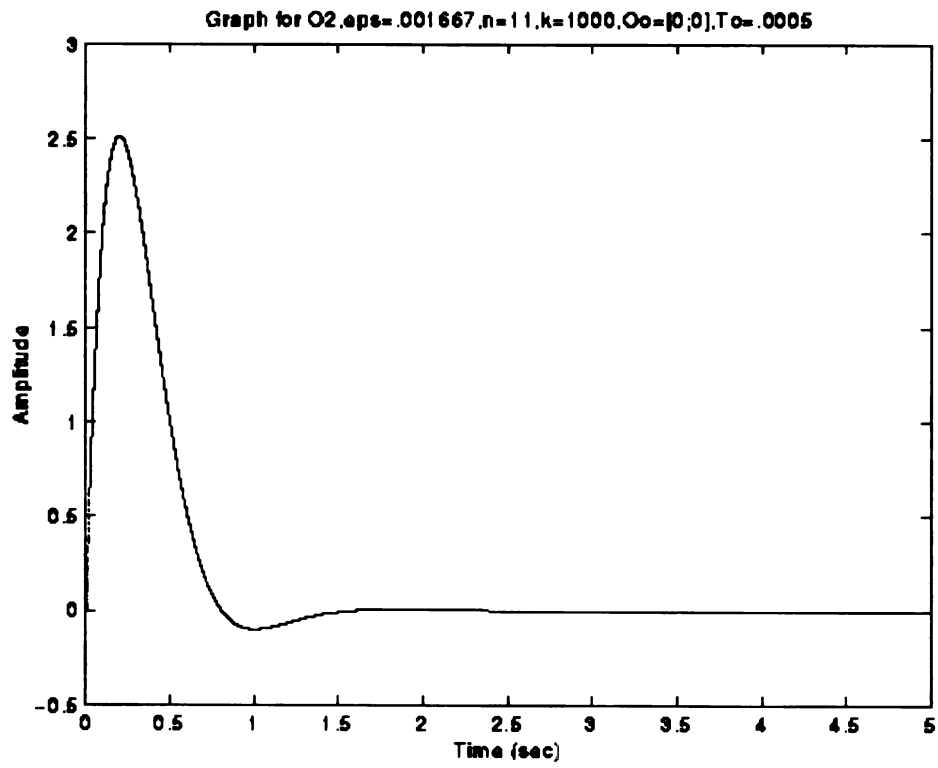
### APPENDIX J3



## APPENDIX J4



## APPENDIX J5



## APPENDIX K

### Multirate Implementation Under Output Feedback Control With Different Value of $\varepsilon$ & Different Initial Conditions

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.001667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [1;1];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1000;
for k = 1:kmax
    if k = 1
        g1 = go;

```

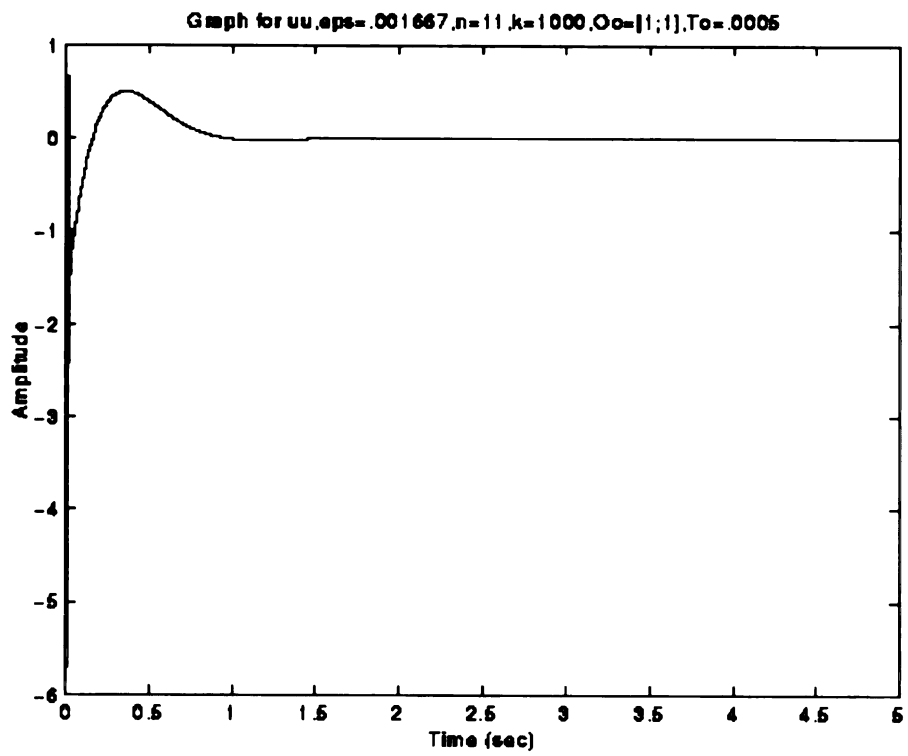
```

elseif k > 1
g1 = (g(n,:))';
end
Oo = g1(3:4);
uo = -K*Oo+N*r;
u = uo*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title(' Graph for O1, ε = 0.001667, n = 11, k = 1000, Oo = [1;1], To = 0.0005 ')
    xlabel('Time (sec)');
    ylabel('Amplitude');
end

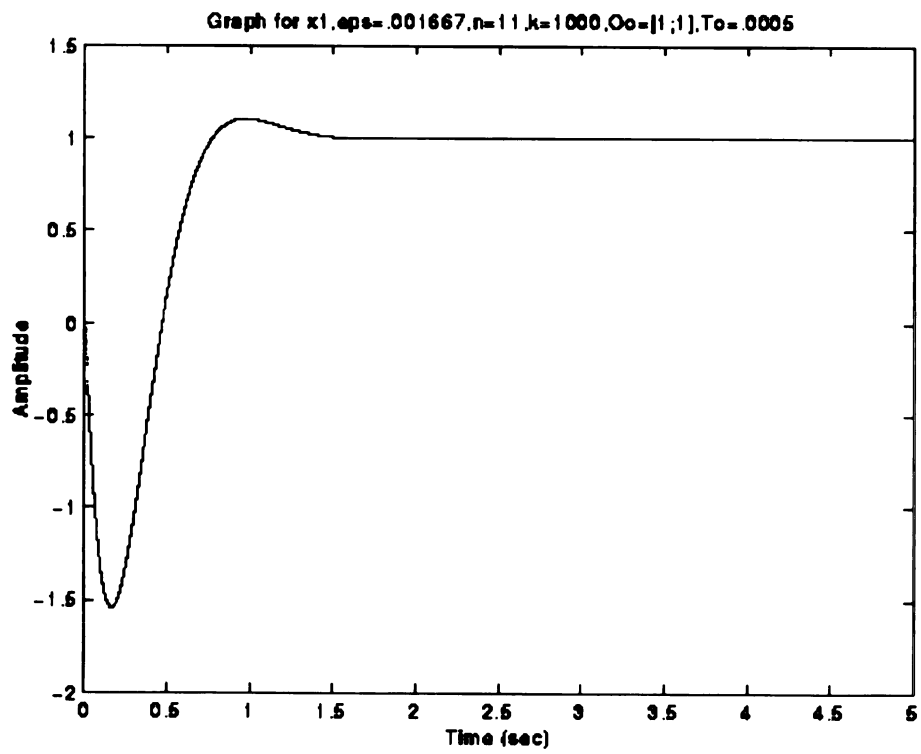
```



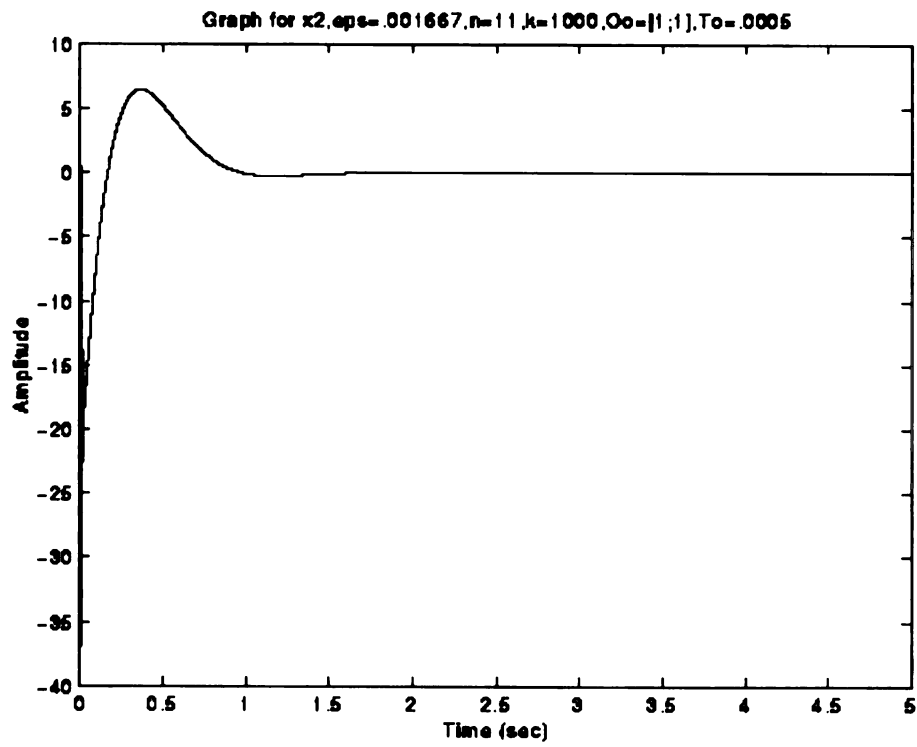
## APPENDIX K1

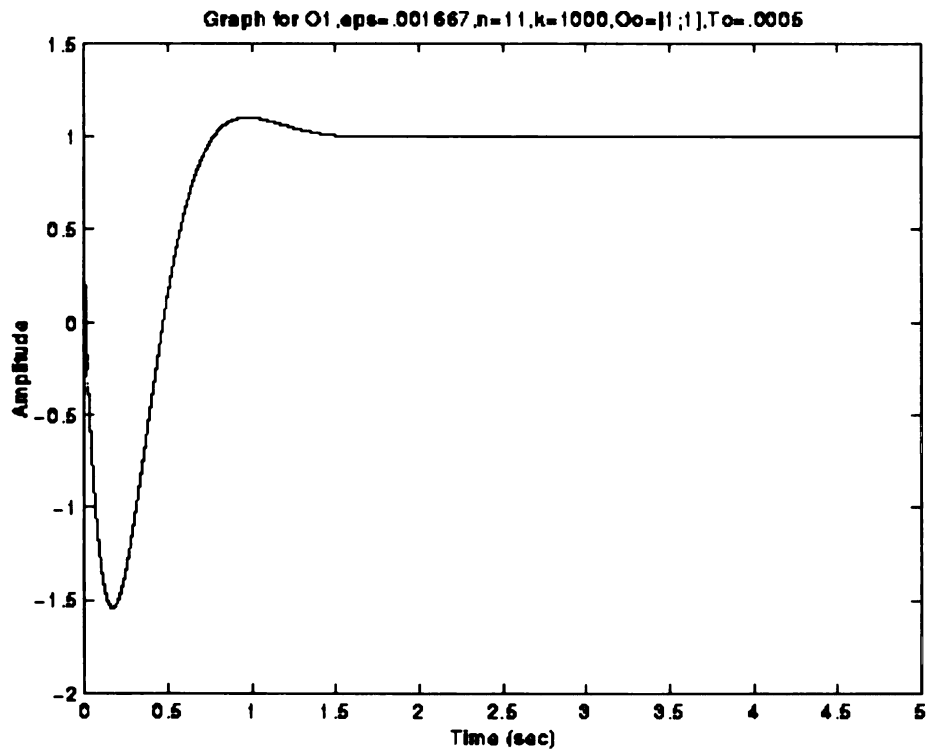


## APPENDIX K2

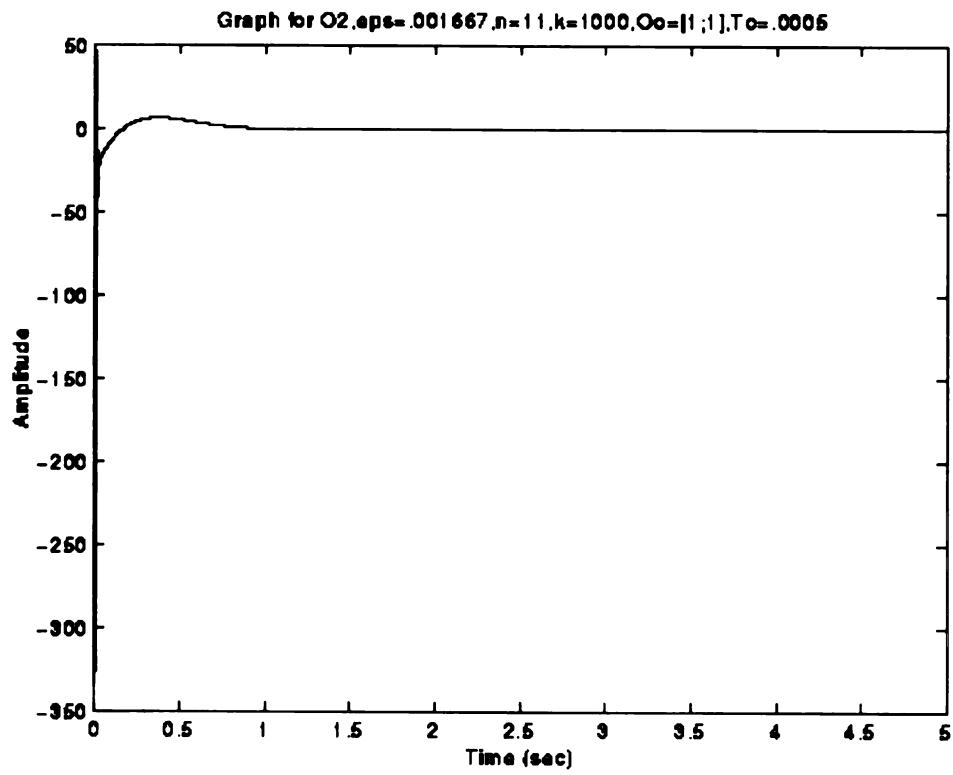


## APPENDIX K3



**APPENDIX K4**

## APPENDIX K5



**APPENDIX L****Multirate Implementation - Control Saturated  
With Different Value of  $\varepsilon$** 

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.001667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [1;1];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
m = 0.5;
n = 11;
kmax = 1000;
for k = 1:kmax
    if k = 1

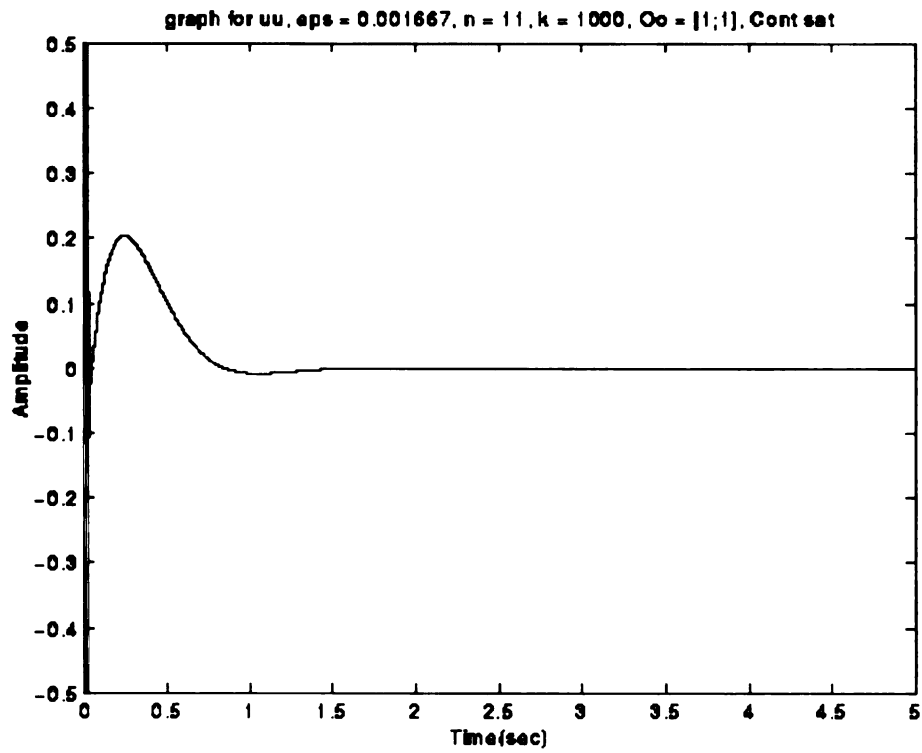
```

```

g1 = g0;
elseif k > 1
g1 = (g(n,:))';
end
Oo = g1(3:4);
uo = -K*Oo+N*r;
    if abs(uo) <= m
        u1 = uo;
    else if abs(uo) > m
        u1 = m*sign(uo);
    end
u = u1*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title(' Graph for O1, ε = 0.001667, n = 11, k = 1000, Oo = [1;1], To = 0.0005 ')
    xlabel('Time (sec)');
    ylabel('Amplitude');
end

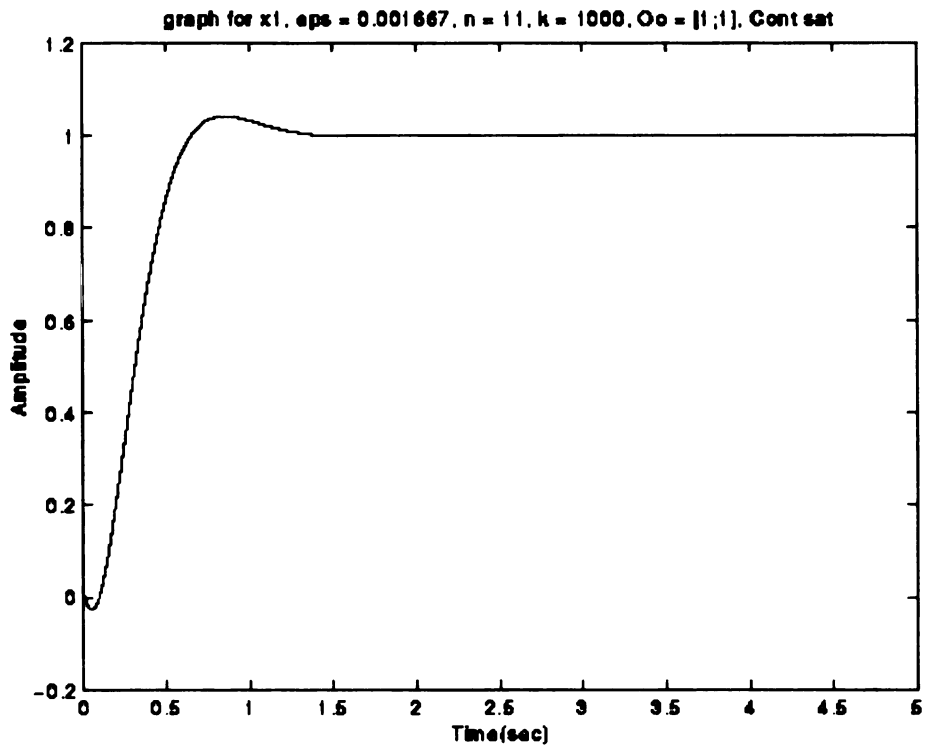
```

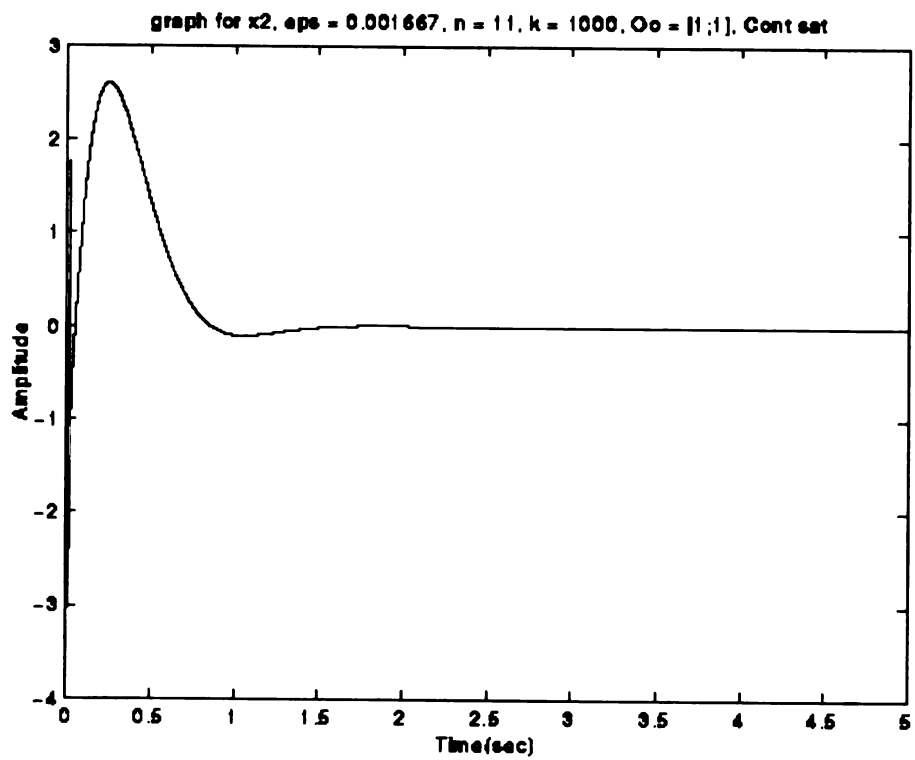
## APPENDIX L1



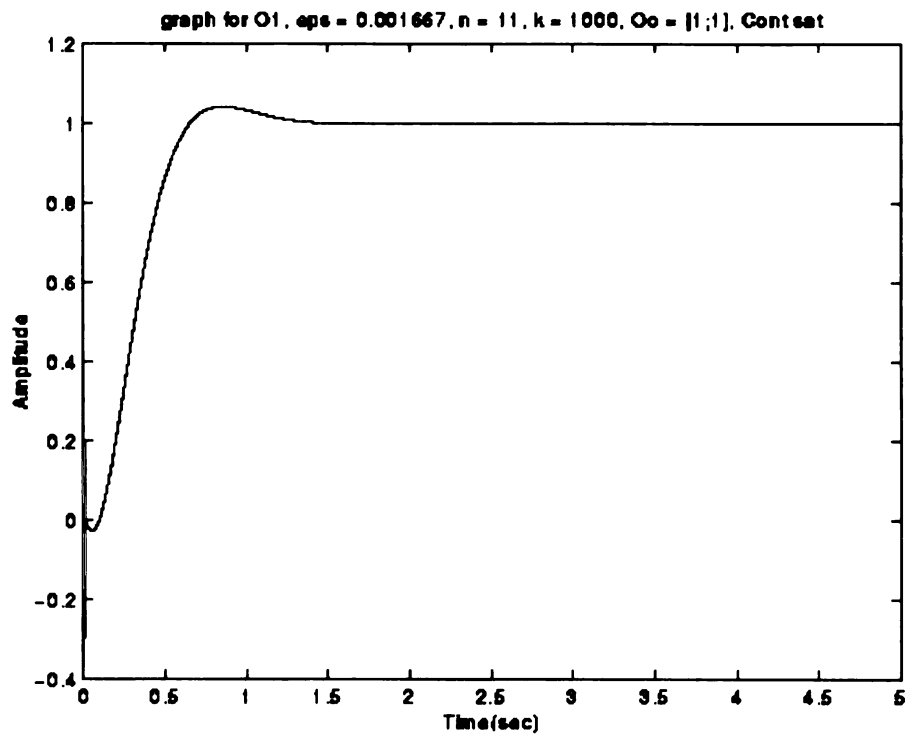


## APPENDIX L2

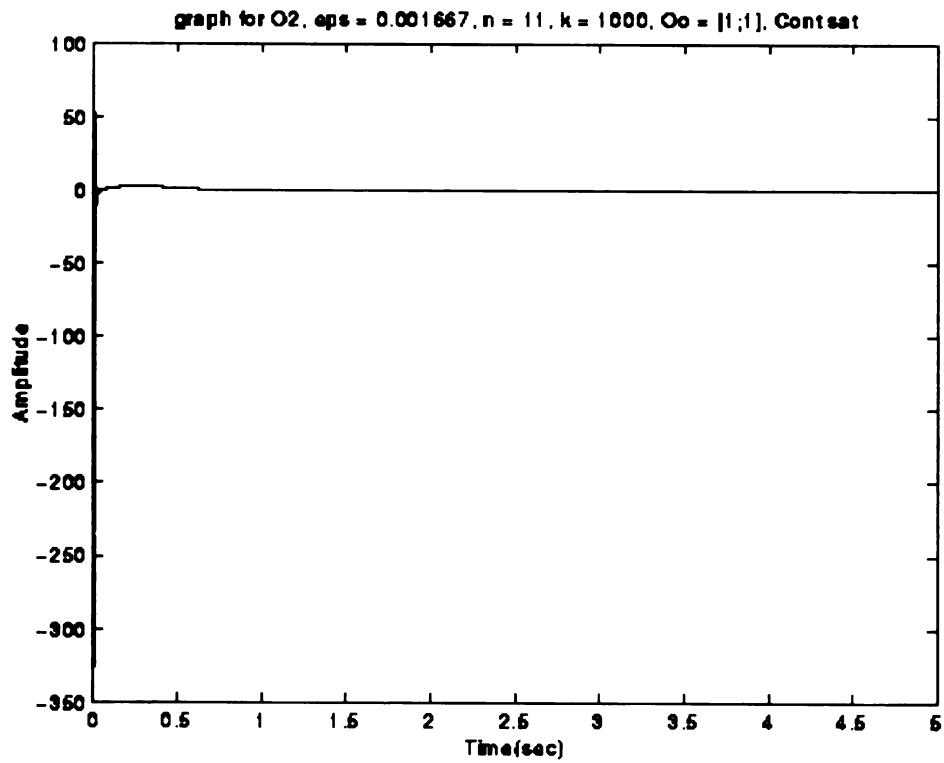


**APPENDIX L3**

## APPENDIX L4



## APPENDIX L5



## APPENDIX M

### Multirate Implementation - Behavior in First Cycle With Different Value of $\varepsilon$

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.001667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [0;0];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1;
for k = 1:kmax
    if k = 1
        g1 = go;

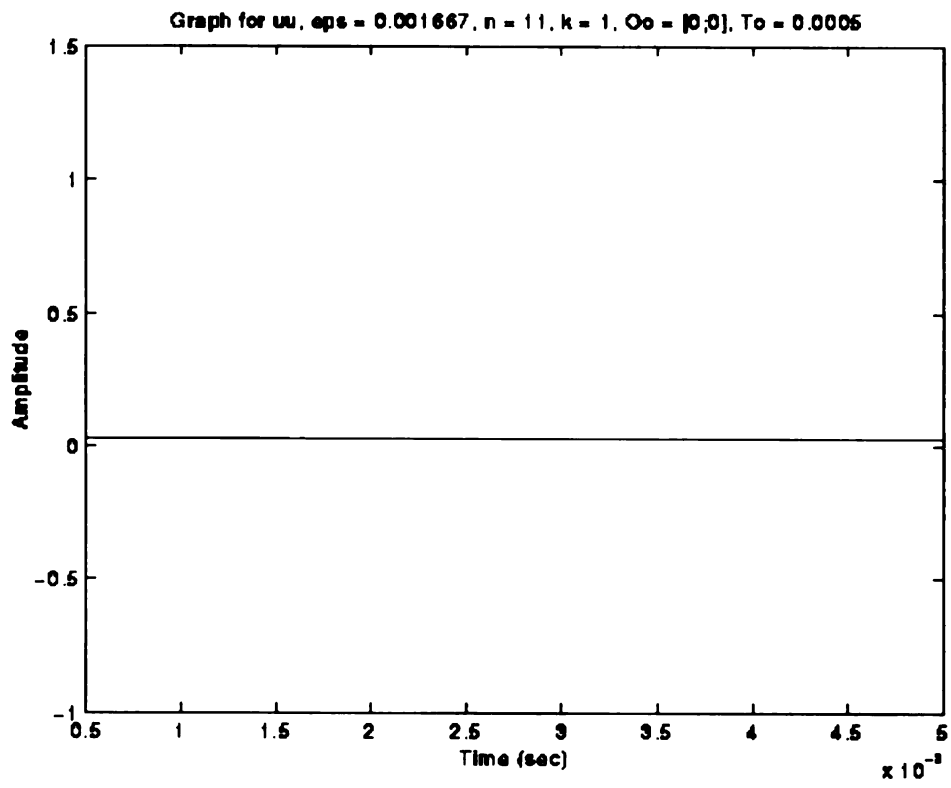
```

```

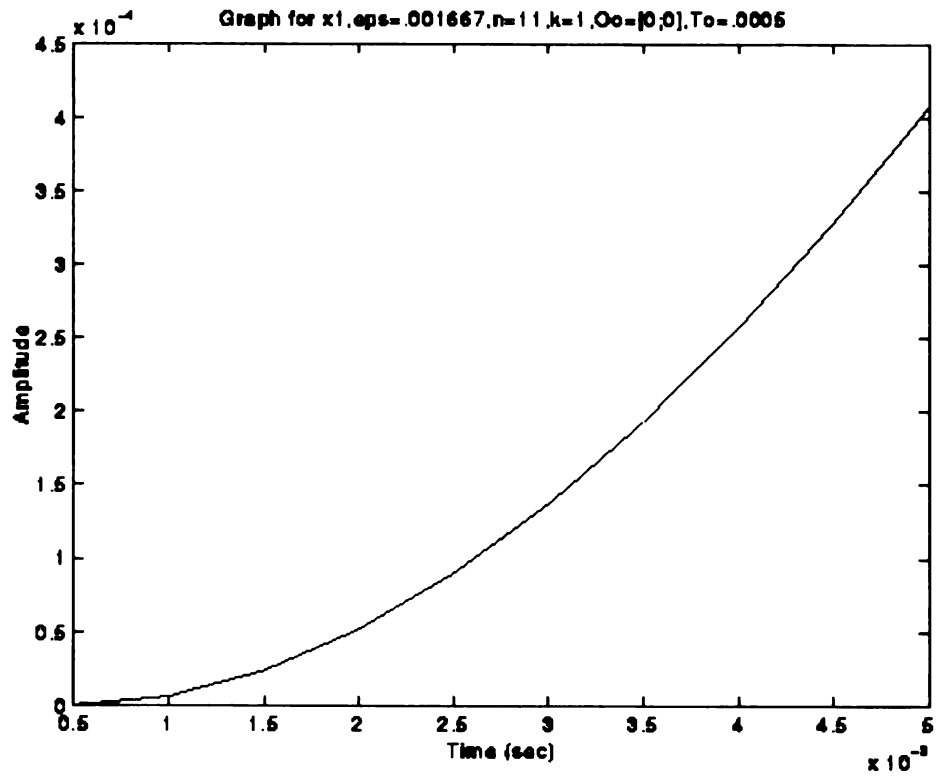
elseif k > 1
g1 = (g(n,:))';
end
O0 = g1(3:4);
u0 = -K*O0+N*r;
u = u0*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*T0;
end
whos;
plot(t,O1); title( ' Graph for O1, ε = 0.001667, n = 11, k = 1, O0 = [0;0], T0 = 0.0005 ' )
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

```

## APPENDIX M1

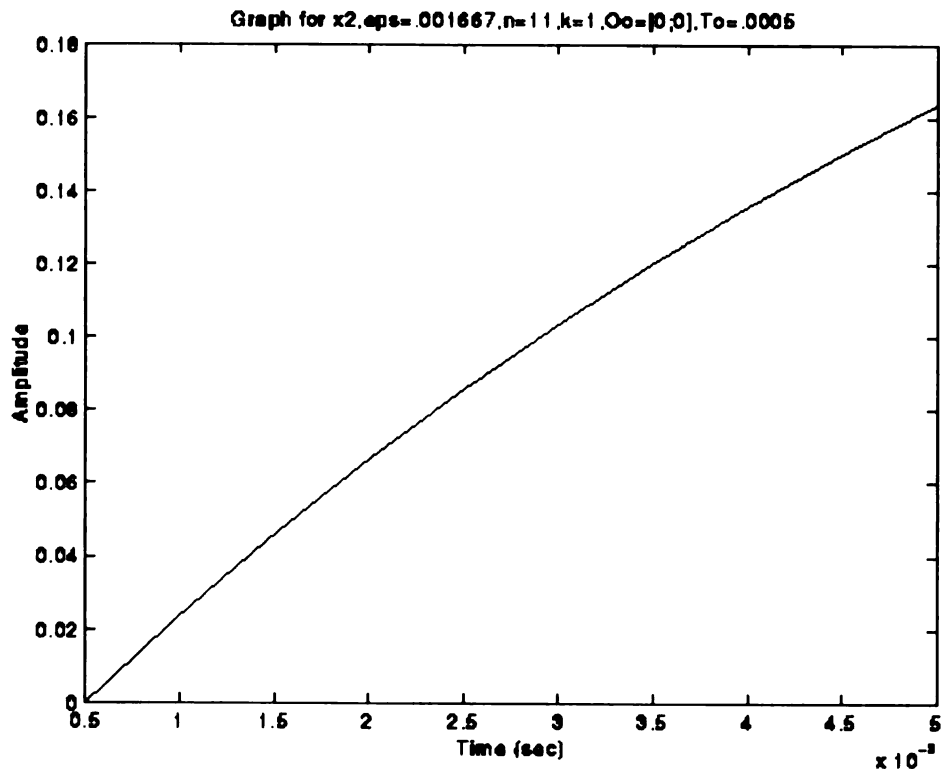


## APPENDIX M2

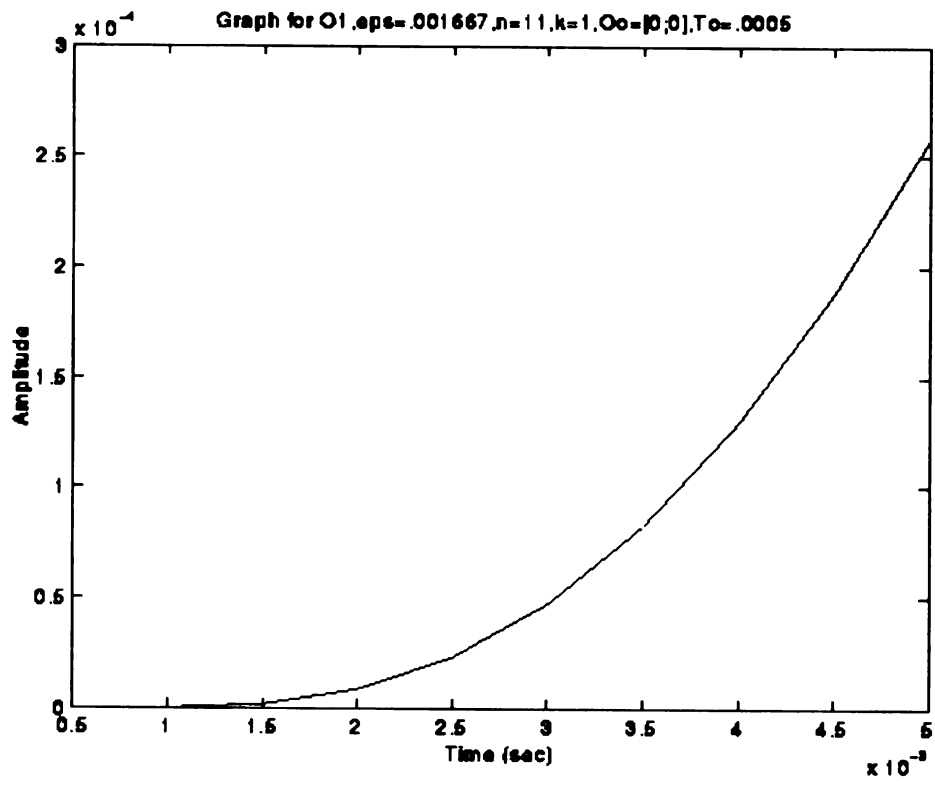




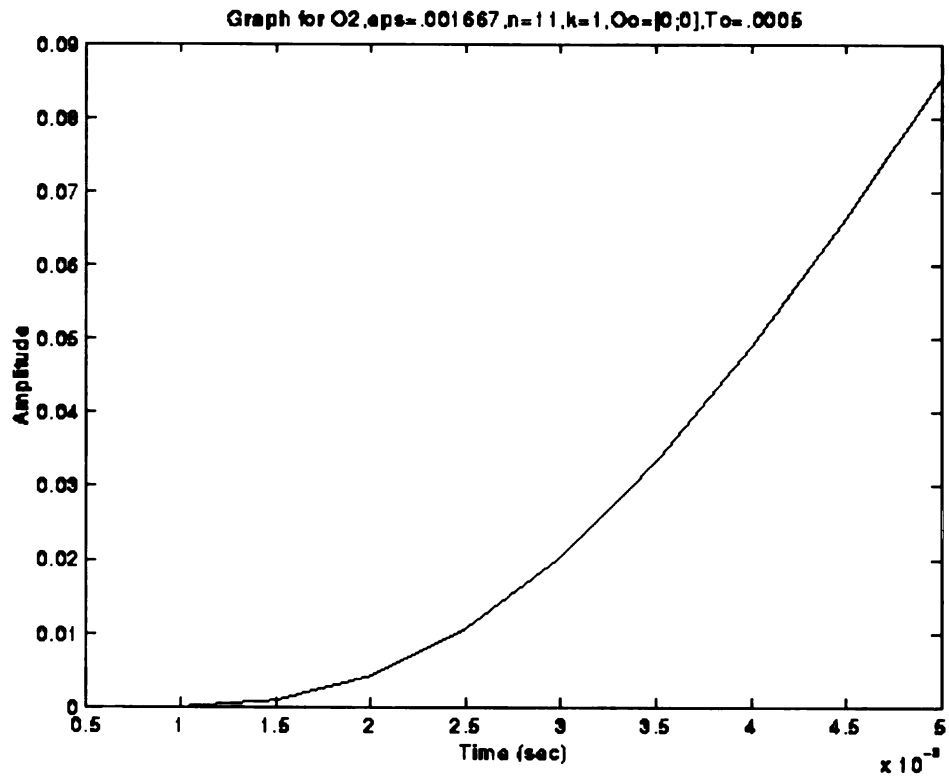
## APPENDIX M3



## APPENDIX M4



## APPENDIX M5



## APPENDIX N

### Multirate Implementation - Behavior in First Cycle With Different Value of $\varepsilon$ & Different Initial Conditions

```

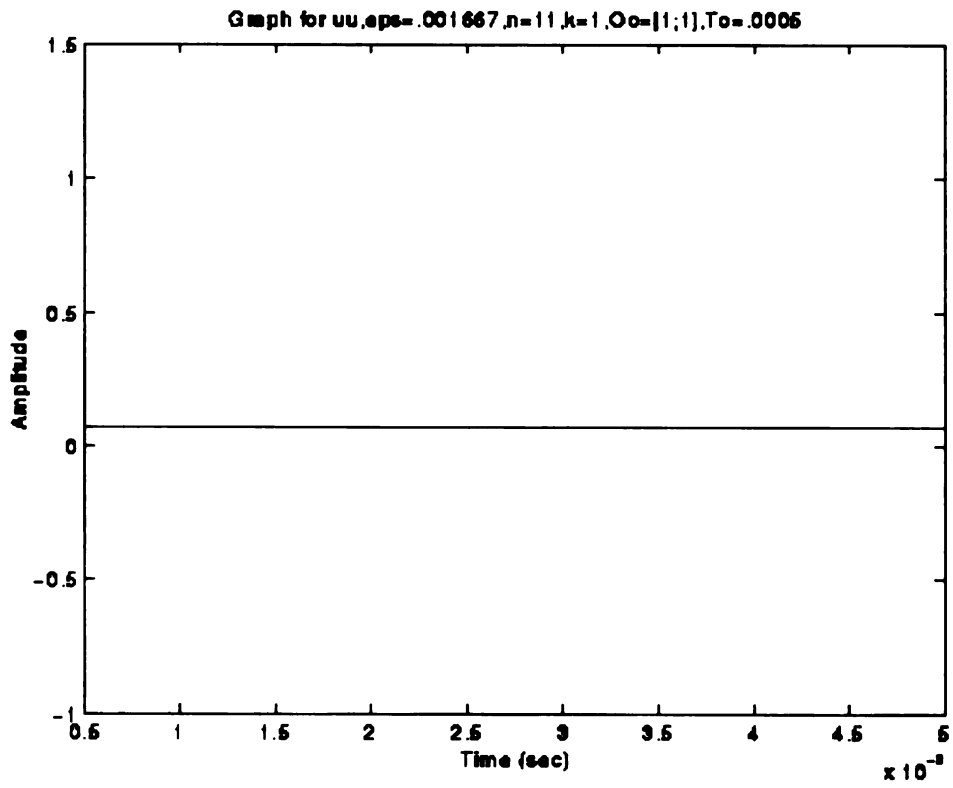
A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.005;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.001667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0005;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [1;1];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 11;
kmax = 1;
for k = 1:kmax
    if k = 1
        g1 = go;

```

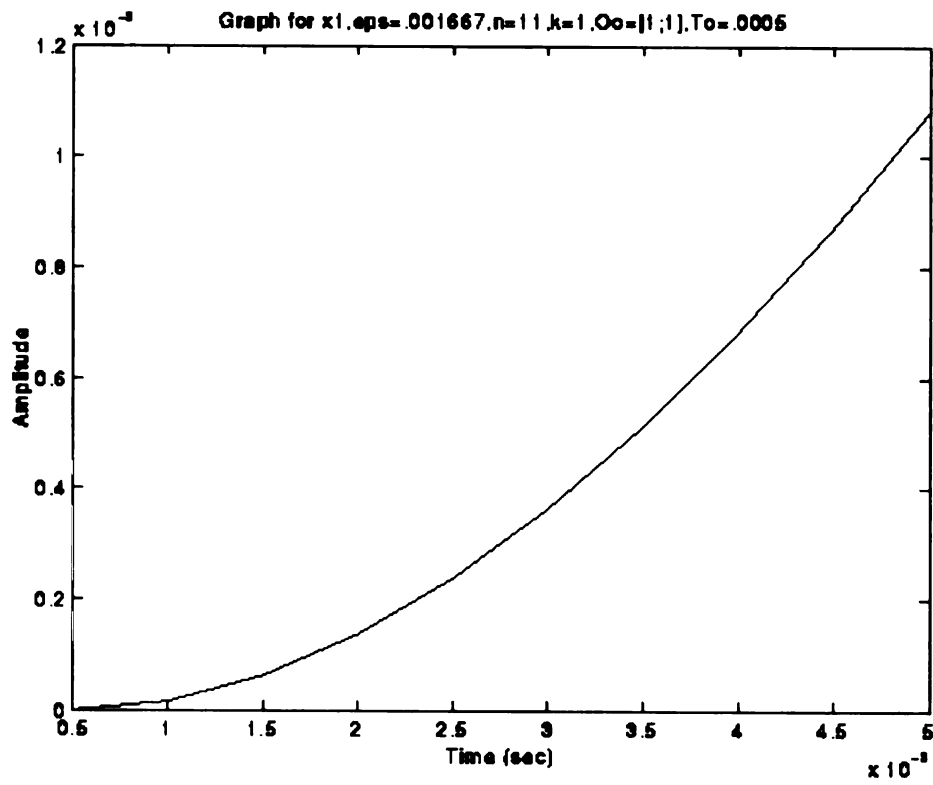
```

elseif k > 1
g1 = (g(n,:))';
end
Oo = g1(3:4);
uo = -K*Oo+N*r;
u = uo*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title( ' Graph for O1, ε = 0.001667, n = 11, k = 1, Oo = [1;1], To = 0.0005 ' )
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

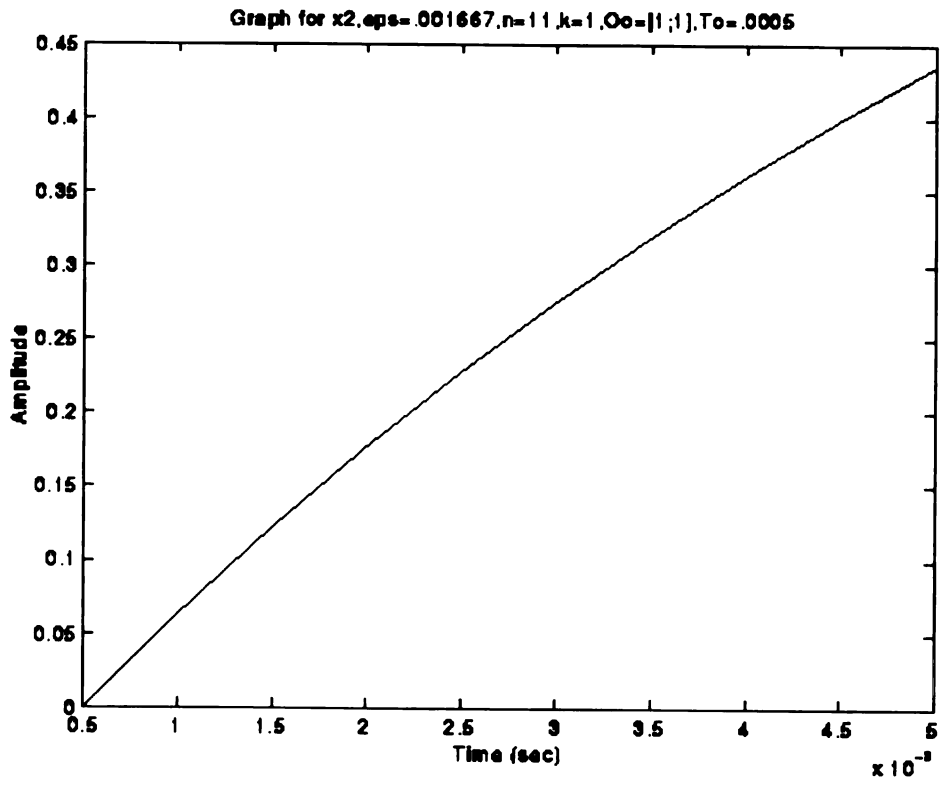
```

**APPENDIX N1**

## APPENDIX N2

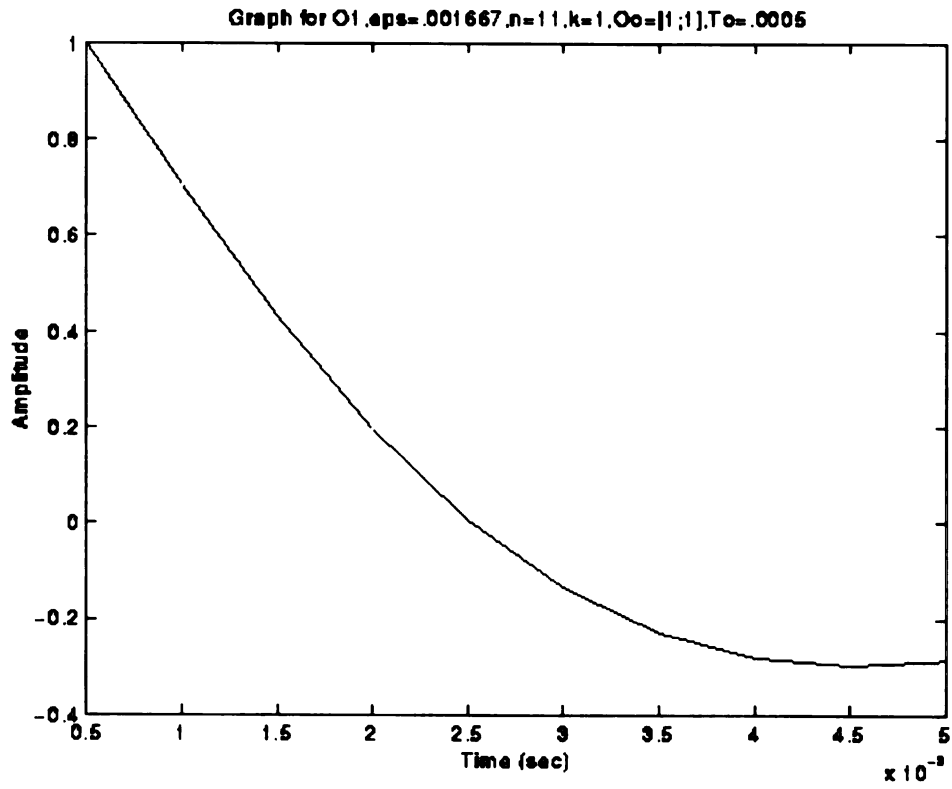


### APPENDIX N3

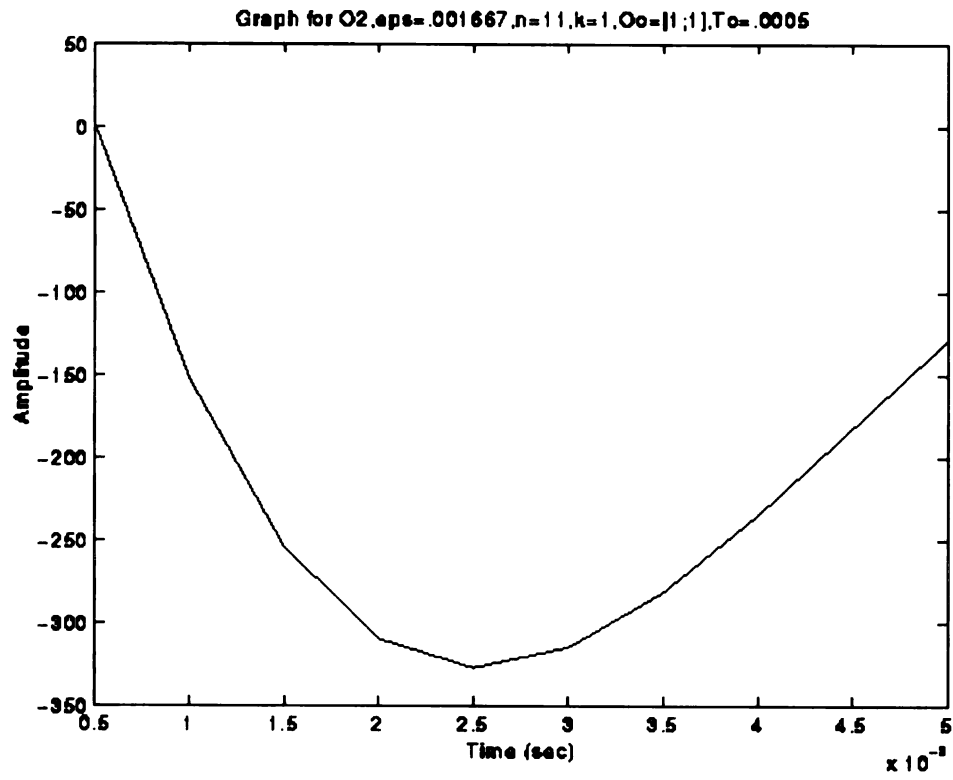




## APPENDIX N4



## APPENDIX N5



## APPENDIX P

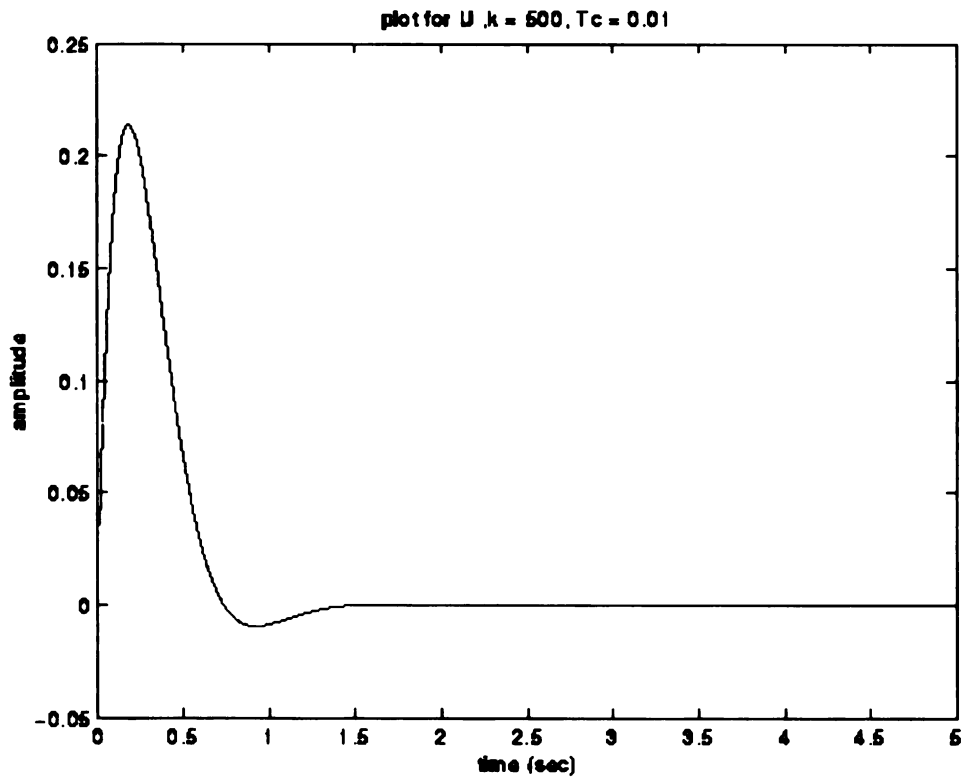
### Matlab program for State Feedback Control

#### With New Value of $T_c$

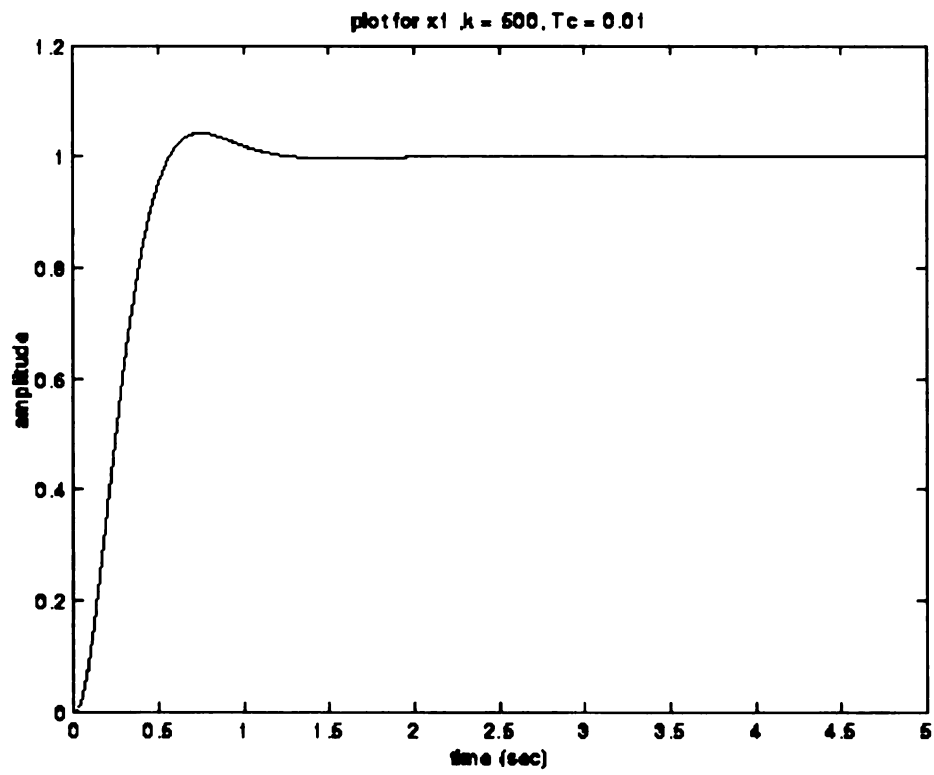
```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
Tc = 0.01;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh')
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2]
K = place(a,b,Pc)
I = [1 0;0 1];
N = inv(c*inv(I-a+b*K)*b)
A1 = [a-b*K];
B1 = [b*N];
IU = [1];
k = 500;
step = [k];
NN = N*ones(k,1);
[y,x] = dstep(A1, B1, c, d, IU, step);
U = (x*(-1)*K')+NN;
x1 = x(:,1);
x2 = x(:,2);
for k = 1:length(U)
    t(k,1)=k*Tc;
end
whos;
plot(t,U); title( ' plot for U , k = 500, Tc = 0.01, state feedback control ' )
xlabel ( ' time (sec) ' );
ylabel ( ' amplitude ' );
end

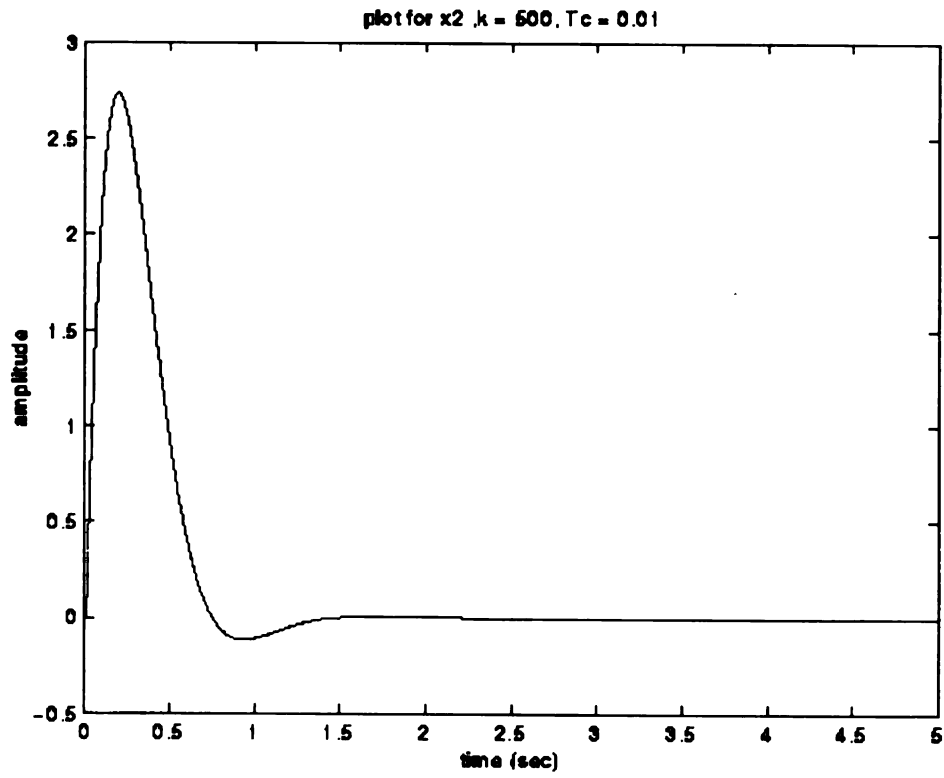
```

**APPENDIX P1**

## APPENDIX P2



## APPENDIX P3



## APPENDIX Q

### Multirate Implementation Under Output Feedback Control With New Values of $T_c$ and $T_o$

```

A = [0 1;0 -142.86];
B = [0;1826.85];
C = [1 0];
D = [0];
I = [1 0;0 1];
e = [0 0;0 0];
Tc = 0.01;
[a,b,c,d] = c2dm(A,B,C,D,Tc,'zoh');
S1 = (-6/sqrt(2))+i*(6/sqrt(2));
S2 = (-6/sqrt(2))-i*(6/sqrt(2));
Z1 = exp(S1*Tc);
Z2 = exp(S2*Tc);
Pc = [Z1 Z2];
K = place(a,b,Pc);
N = inv(c*inv(I-a+b*K)*b);
ε = 0.01667;
Ao = [-1/ε 1;-1/(ε^2) 0];
Bo = [1/ε;1/(ε^2)];
Co = [1 0];
Do = [0];
To = 0.0001;
[ao,bo,co,do] = c2dm( Ao, Bo, Co, Do, To, 'zoh' );
[a1,b1,c1,d1] = c2dm( A, B, C, D, To, 'zoh' );
aa = [a1 e;bo*c1 ao];
bb = [b1;0;0];
cc = [c1 0 0];
dd = [0];
xo = [0;0];
Oo = [1;1];
r = 1;
go = [xo;Oo];
gg = [];
uu = [];
n = 101;
kmax = 500;
for k = 1:kmax
    if k = 1
        g1 = go;

```

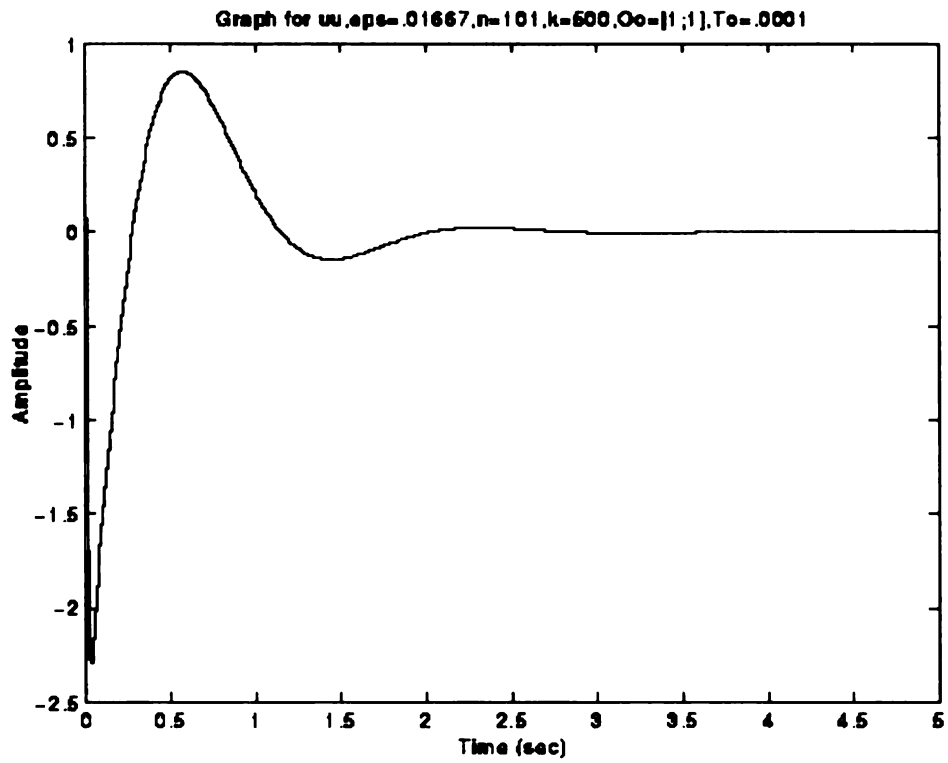
```

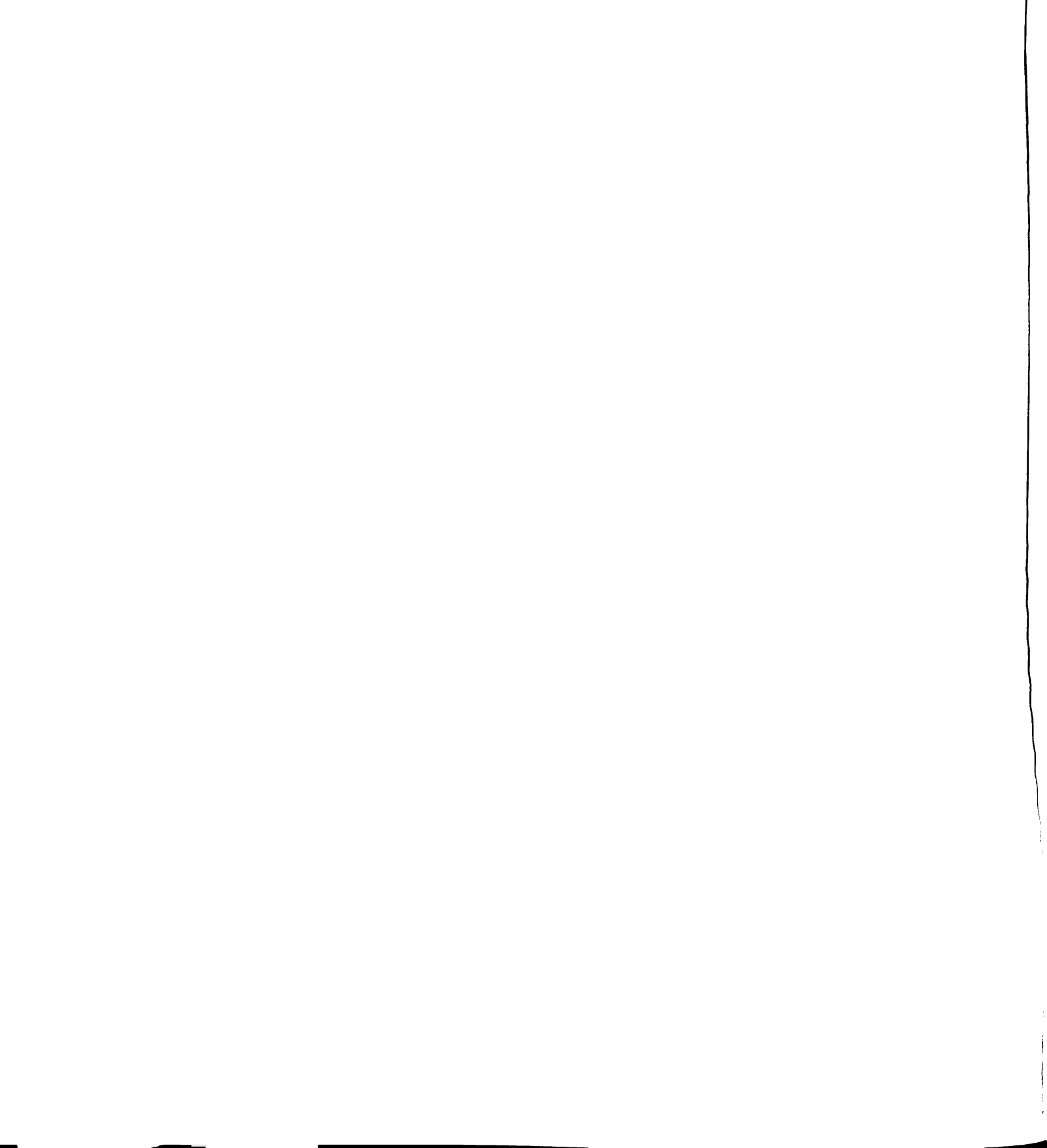
elseif k > 1
g1 = (g(n,:))';
end
Oo = g1(3:4);
uo = -K*Oo+N*r;
u = uo*ones(n,1);
[h,g] = dlsim(aa, bb, cc, dd, u, g1);
gg = [gg;g(1:n-1,:)];
uu = [uu;u(1:n-1)];
x = gg(:,1:2);
O = gg(:,3:4);
x1 = gg(:,1);
x2 = gg(:,2);
O1 = gg(:,3);
O2 = gg(:,4);
end
for k = 1:length(O1)
    t(k,1) = k*To;
end
whos;
plot(t,O1); title(' Graph for O1, ε = 0.01667, n = 101, k = 500, Oo = [1;1], To = 0.0001 ')
    xlabel ('Time (sec)');
    ylabel ('Amplitude');
end

```

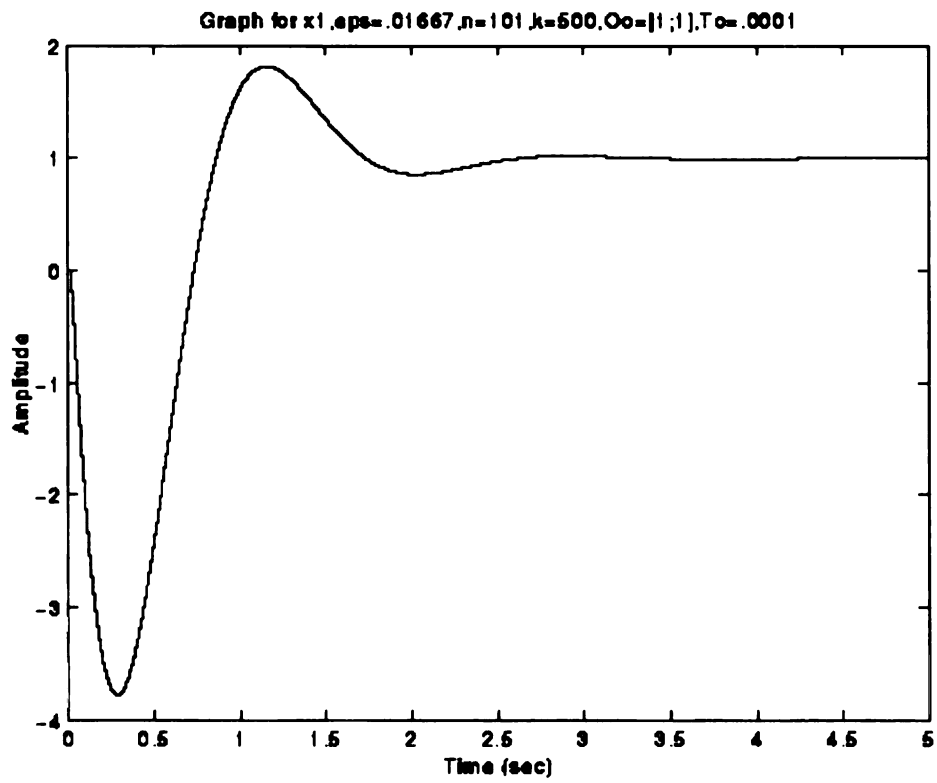


## APPENDIX Q1

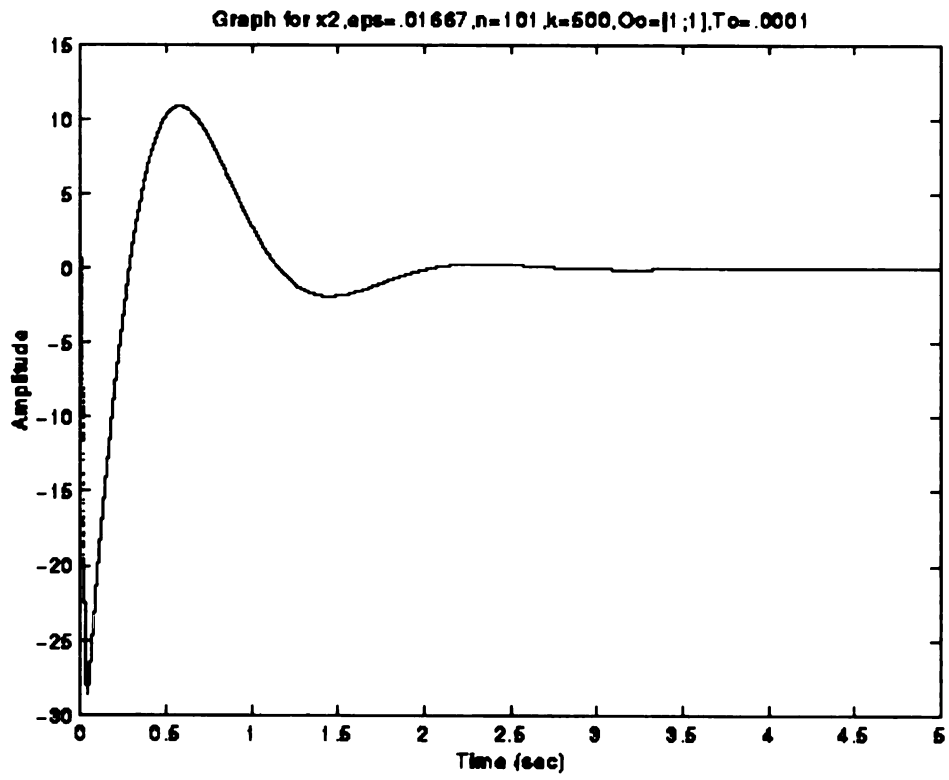




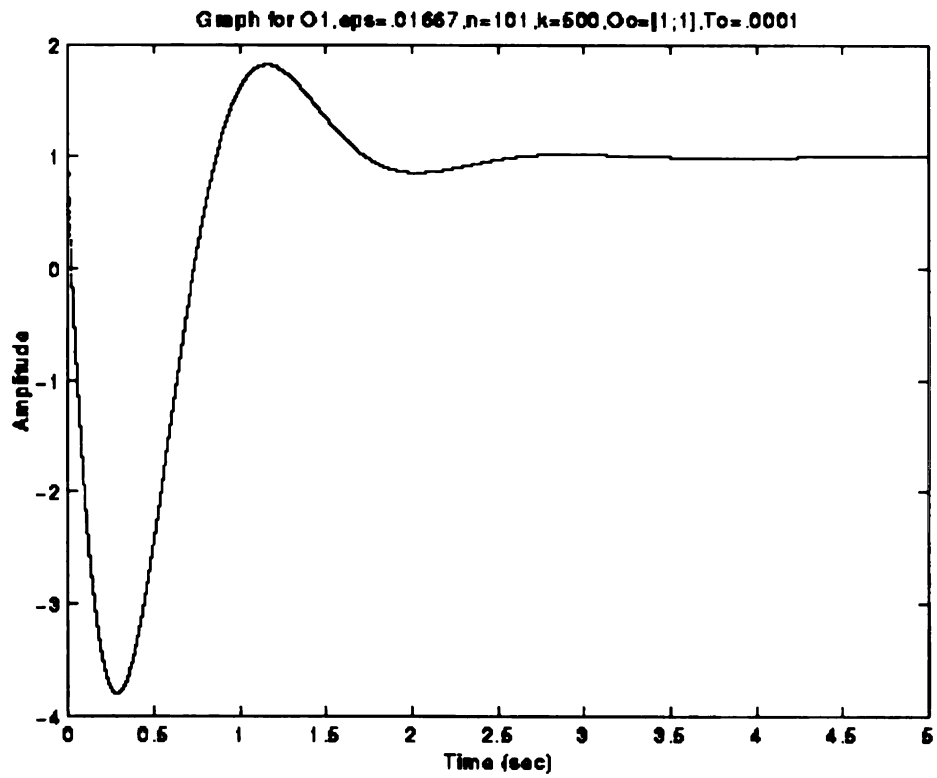
## APPENDIX Q2



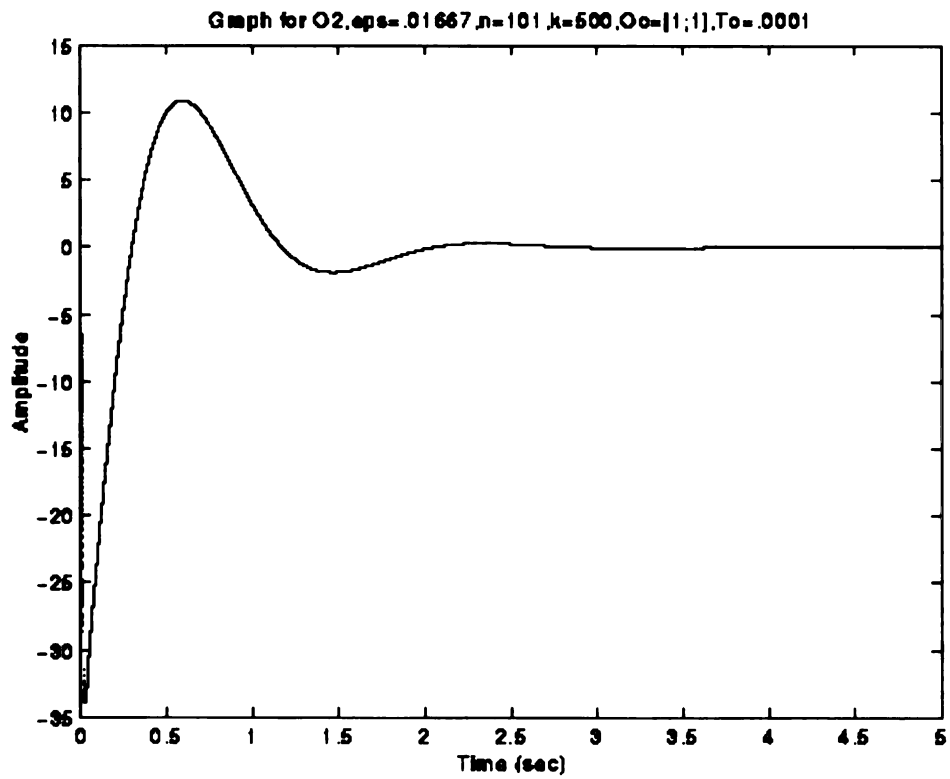
## APPENDIX Q3



## APPENDIX Q4



## APPENDIX Q5



# **BIBLIOGRAPHY**

## BIBLIOGRAPHY

- [1] J. Sklansky and J.R. Ragazzini, " Analysis of errors in sampled data feedback systems," AIEE Trans., vol.74, part II, pp. 65-71, may 1955.
- [2] G.M. Kranc, : " The analysis of multiple rate sampled systems," Report No. T-11/B, Electronic research laboratory, Dept. of Electrical engineering, Columbia university, New York, NY, May 1956.
- [3] E.I Jury, Sampled date control systems, John Wiley and sons, Inc., New York, 1958.
- [4] R.E. Kalman and J. Bertram, " A unified approach to the theory of sampling systems," J. Franklin Institute, vol.267, pp.405-436, May 1959.
- [5] P.E. Barry, " Optimal control of multirate systems," Report No. RN-361, Research Department, Grumman Aerospace Corporation, July 1975.
- [6] N.Amit, " Optimal control of multirate digital control systems, " Report No. SUDDAR # 523, Stanford university, Stanford, CA, July 1980.
- [7] D.P. Glason, " Development and applications of multirate digital control," IEEE Contr. Syst. Magazine, vol.3, pp. 2-8, 1983.
- [8] P.V. Kokotovic, R.E. O'Malley, Jr., and P. Sannuti, " Singular perturbation and order reduction in control theory - An overview," Automatica, vol.12, pp. 123-132, 1976.
- [9] V.R. Saksena, J. O'Reilly, and P.V. Kokotovic, " Singular perturbation and time scale methods in control theory: Survey 1976-1983," Automatica, vol.20, pp.273-293, 1984.
- [10] O'Malley, R.E. Jr. (1982). Book review. Bulletin AMS, 7, 414.
- [11] Ioannou, P.A. and P.V. Kokotovic (1983). Adaptive systems with reduced models. Lecture notes in control information science, 47. Springer, Berlin.
- [12] Bakhtiar Litkouhi, and Hassan Khalil, " Multirate and composite control of two time scale discrete time systems," IEEE Transaction on automatic control, vol.AC-30. No.7, July 1985.



- [13] Hassan Khalil, "Robustness issues in output feedback control of linearizable systems," Unpublished, 1993.
- [14] F. Esfandiari and H.K. Khalil. Observer based design of uncertain systems: recovering state feedback robustness under matching conditions. In proc. Allerton conf., pp.97-106, Monticello, IL, September 1987.
- [15] F. Esfandiari and H.K. Khalil. Output feedback stabilization of fully linearizable systems. *Int.J.contr.*, 56: 1007-1037, 1992.
- [16] A. Saberi and P. Sannuti. Observer design for loop transfer recovery and for uncertain dynamical systems. *IEEE Trans. Automat. contr.*, 35(8): 878-897, August 1990.
- [17] I.R. Petersen and C.V. Holot. High gain observers applied to problems in disturbance attenuation, H-infinity optimization and the stability of uncertain linear systems. In proc. American control conf., pp 2490-2496, Atlanta, GA, June 1988.
- [18] P.W. Gibbens and M. Fu. Output feedback control for output tracking of nonlinear uncertain systems. Technical report EE9121, University of Newcastle, Australia, 1991.
- [19] A. Teel and L. Praly. Tools for semiglobal stabilization by partial state and output feedback. 1992.
- [20] Oliver Fuchs. Digital implementation of a high gain observer. A thesis, Michigan state University, U.S.A, 1994.

MICHIGAN STATE UNIV. LIBRARIES



31293015610979