

Observer Incorporated Neoclassical Controller Design: A Discrete Perspective

Winston Alexander Baker
Marquette University

Recommended Citation

Baker, Winston Alexander, "Observer Incorporated Neoclassical Controller Design: A Discrete Perspective" (2010). *Master's Theses (2009 -)*. 59.
https://epublications.marquette.edu/theses_open/59

OBSERVER INCORPORATED NEOCLASSICAL CONTROLLER DESIGN:
A DISCRETE PERSPECTIVE

By

W. Alexander Baker Jr.

A Thesis submitted to the Faculty of the Graduate School, Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

August 2010

ABSTRACT
OBSERVER INCORPORATED NEOCLASSICAL CONTROLLER DESIGN:
A DISCRETE PERSPECTIVE

W. Alexander Baker Jr.

Marquette University, 2010

Control theory has generally been divided into two categories, modern control and classical control. Modern control uses state feedback to alter the pole locations of a given system. Classical control uses pre-compensation to alter the zeroes of the system and uses output feedback to adjust the poles to bring stability to the system. The drawback is that the application of classical control techniques can be a lengthy, complicated and iterative design process and in the end, classical control techniques still do not give information about the state of the system. Neoclassical control combines classical control techniques with the state feedback approach of modern control to stabilize the system, eliminate the steady state error, provide relevant internal state information, and reduce the time it takes to design the controller.

This thesis explores the application of neoclassical control to discrete-time systems. The mass-spring-damper, magnetic levitation, and ball and beam systems are discretized using the zero-order-hold or the Euler approximation. State-feedback control is used to modify the pole locations for these systems. A discrete-time integrator is put in series to eliminate the steady-state error for a step input. The pre-compensator is also put in series to replace the numerator of the open-loop system with a desired numerator. The unit output-feedback is then used to close the loop. The closed-loop system will have a step response which matches the discrete-time optimal ITAE, Bessel, or Butterworth transfer functions.

An observer is added to estimate the state of the plant in this work. The observer is applied to the discrete-time mass-spring-damper, the magnetic levitation, and the ball and beam systems in such a way that the error in the state estimate will be driven to zero within the desired period of time. This will allow the application of this controller to systems when the state is not known or measurable.

ACKNOWLEDGEMENTS

W. Alexander Baker Jr.

I want to express my gratitude to my parents, Karen P. Baker and Winston A. Baker Sr., for their love and support all of these years. I want to thank my sister, Monika J. Baker, for keeping me level-headed throughout the writing process. I want to show my appreciation to the rest of my family for being the crowd that keeps cheering me on.

I want to thank my friends who helped me keep my perspective on life and on my research.

Lastly, I want to thank my esteemed professors. I am grateful to Dr. Fabien Josse for giving me the chance to prove myself. I thank Dr. Edwin Yaz for providing me direction and believing in me. I want to show my gratitude Dr. Chung Seop Jeong for giving me the tools I needed to succeed. And I express my appreciation to Dr. Susan Schneider for her guidance throughout the entire research process.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
LIST OF TABLES	iv
LIST OF FIGURES	v
Chapter 1: Introduction.....	1
1.1: History of Control Theory	1
1.2: Overview of Continuous-Time Neoclassical control.....	2
Chapter 2: Continuous-time Neoclassical Control Review	5
2.1: Neoclassical components	5
2.1.1: The Plant	6
2.1.2: State-feedback	7
2.1.3: The pre-compensator, integrator and output feedback	8
2.2: Standard Transfer Functions.....	8
2.2.1: ITAE Transfer Functions	9
2.2.3: Bessel Transfer Functions	14
2.3: Implementation of the continuous-time neoclassical controller.....	16
2.4: Limitations of neoclassical control	19
2.5: Conclusion.....	19
Chapter 3: Theory of Discrete-time Neoclassical Control.....	20
3.1 Overview of discrete-time neoclassical control.....	20
3.2: Discrete-time Plant.....	21
3.2.1: Zero Order Hold (ZOH).....	22
3.2.2: Euler Approximation	23
3.3: Discretization of the continuous-time integrator	24
3.4: Observer Based Discrete-time Neoclassical Control.....	24
3.4.1: Observer implementation.....	25
3.5: Conclusion.....	28
Chapter 4: Case Studies without Observers.....	29
4.1: The Mass-Spring Damper system.....	29
4.2: Magnetic Levitation system	39
4.3: The Ball and Beam System	50

4.4: Conclusion.....	60
Chapter 5: Case Studies with Observers	61
5.1: The Mass-Spring-Damper System.....	61
5.2: Magnetic Levitation system	66
5.3: The Ball and Beam System	73
5.4: Conclusion.....	79
Chapter 6: Conclusion and Future Work.....	80
6.1: Future Work.....	81
Bibliography	82
Appendix A: MATLAB Code	83
Mass-Spring Damper code.....	83
Magnetic Levitation code.....	87
Ball and Beam code	92

LIST OF TABLES

Table 4- 1: Table of parameters to be used for the Mass-Spring-Damper system	31
Table 4- 2: Table of parameters to be used for the Magnetic Levitation system	42
Table 4- 3: Table of parameters to be used for the Ball and Beam simulations	52
Table 5- 1: Controller gains for $T_s=1$ sec and Observer gains for $T_s=0.1$ sec. for the Mass-Spring-Damper system	63
Table 5- 2: Controller gains for $T_{sc}=1$ sec. and Observer gains for $T_{so}=0.1$ sec. for the Magnetic Levitation system.....	69
Table 5- 3: Controller gains for $T_{sc}=1$ sec. and Observer gains for $T_{so}=0.1$ sec. for the Ball and Beam system	75

LIST OF FIGURES

Figure 1- 1: Block diagram of the neoclassically controlled system (1)	2
Figure 2- 1: Block diagram of the neoclassically controlled system	6
Figure 2- 2: 2nd through 5th order of ITAE optimized transfer function step responses .	11
Figure 2- 3: 2nd through 5th order of Butterworth optimized transfer function step responses.....	13
Figure 2- 4: 2nd through 5th order of Bessel optimized transfer function step responses	15
Figure 2- 5: Neoclassical controller design flow chart.....	18
Figure 3- 1: Discrete-time neoclassical controller block diagram	21
Figure 3- 2: Discrete-time neoclassical controller block diagram with an observer.....	25
Figure 4- 1: A diagram of the Mass-Spring-Damper system with relevant force vectors (1)	29
Figure 4- 2 : The open-loop Mass-Spring-Damper system for a unit step input	32
Figure 4- 3: The (a) step response and (b) control input of the neoclassical ITAE controller for the Mass-Spring-Damper system.....	35
Figure 4- 4: The (a) step response and (b) control input of the neoclassical Bessel controller for the Mass-Spring-Damper system.....	36
Figure 4- 5: The (a) step response and (b) control input of the neoclassical Butterworth controller for the Mass-Spring-Damper system.....	37
Figure 4- 6: The (a) step responses and (b) control inputs of the neoclassical controllers for the Mass-Spring-Damper system.....	38
Figure 4- 7: A diagram of the Magnetic Levitation system.....	40
Figure 4- 8: The unit step response for the open-loop Magnetic Levitation system	44
Figure 4- 9: The (a) step response and (b) control input of the neoclassical ITAE controller for the Magnetic Levitation system.....	45

Figure 4- 10: The (a) step response and (b) control input of the neoclassical Butterworth controller for the Magnetic Levitation system.....	47
Figure 4- 11: The (a) step response and (b) control input of the neoclassical Bessel controller for the Magnetic Levitation system.....	48
Figure 4- 12: A comparison of the ITAE, Bessel, and Butterworth (a) step responses and (b) control inputs.....	49
Figure 4- 13: A diagram of the Ball and Beam system dynamics	50
Figure 4- 14: The unit step response of the open-loop Ball and Beam system.	54
Figure 4- 15: The (a) step response and (b) control input of the neoclassical ITAE controller for the Ball and Beam system	55
Figure 4- 16: The (a) step response and (b) control input of the neoclassical Butterworth controller for the Ball and Beam system	56
Figure 4- 17: The (a) step response and (b) control input of the neoclassical Bessel controller for the Ball and Beam system	57
Figure 4- 18: A comparison of the ITAE, Bessel, and Butterworth (a) step responses and (b) control inputs for the Ball and Beam system	58
Figure 4-18c: An expanded view of the control inputs.....	59
Figure 5- 1: A diagram of the Mass-Spring-Damper system (1).....	61
Figure 5- 2: Discrete-time ITAE Neoclassical controller (a) step response and(b) state estimate error for the Mass-Spring-Damper system	64
Figure 5- 3: Discrete-time Bessel Neoclassical controller (a) step response and (b) state estimate error for the Mass-Spring-Damper system	65
Figure 5- 4: Discrete-time Butterworth Neoclassical controller (a) step response and (b) state estimate error for the Mass-Spring-Damper system	66
Figure 5- 5: Diagram of the Magnetic Levitation system	67
Figure 5- 6: Discrete-time ITAE Neoclassical controller (a) step response and (b) state estimate error for the Magnetic Levitation system	70

Figure 5- 7: Discrete-time Bessel Neoclassical controller (a) step response and (b) state estimate error for the Magnetic Levitation system	71
Figure 5- 8: Discrete-time Butterworth Neoclassical controller (a) step response and (b) state estimate error for the Magnetic Levitation system	72
Figure 5- 9: Diagram of the Ball and Beam system.....	73
Figure 5- 10: Discrete-time ITAE Neoclassical controller (a) step response and (b) state estimate error for the Ball and Beam system.....	76
Figure 5- 11: Discrete-time Bessel Neoclassical controller (a) step response and (b) state estimate error for the Ball and Beam system.....	77
Figure 5- 12: Discrete-time Butterworth Neoclassical controller (a) step response and (b) state estimate error for the Ball and Beam system.....	78

Chapter 1: Introduction

1.1: History of Control Theory

Control theory has a long history, from James Watt's fly-ball governor to J.C. Maxwell's use of differential equations to mathematically describe the performance of a system (1). Control theory also has a diverse history, particularly in the World War II and Cold War era. During this time, the United States and western civilization focused on frequency domain control techniques while the Soviet Union focused on time domain techniques (1). These control methods helped control systems from automatic pilots of planes to the telephone system. In the years following World War II, the techniques, now referred to as classical control, were emphasized academically. Combining root locus techniques with proportional, integral, and derivative control was, and still is, an effective method of control. Various combinations of proportional, integral, and derivative control are used in signal filtering and the performance characteristic control of systems. For linear, time-invariant (LTI), single input single out (SISO) systems, classical control works well, however, it is not optimal for controlling complex systems with multiple inputs and multiple outputs (MIMO). As the space race was heating up between the USSR and the USA, modern control theory became the more efficient alternative for controlling complex systems. Modern control allows complex systems to be broken down into state variables. These state variables are controlled via state-feedback control. State-feedback control allows the poles of a system transfer function to be relocated without altering the zeros in the process. If a particular state is unknown, modern control uses observers in conjunction with the controller in order to estimate the unknown states. The observer is designed in such a way that the difference between the state and the state

estimate is reduced to zero within a predetermined period of time, allowing the state estimate to replace the actual state, which is not measurable.

1.2: Overview of Continuous-Time Neoclassical control

Individually, classical and modern control work reasonably well. However, if classical control techniques are combined with modern control techniques, the overall control design process can be more efficient than classical control or modern control alone. This concept is the idea behind the development of neoclassical control. By combining elements of classical control with elements of modern control, it is possible to achieve a closed-loop transfer function for a linear, time-invariant, SISO system that is equivalent to a chosen standard transfer function. The process is done in such a way that the only design parameter is the settling time. The standard transfer functions chosen for this thesis are ones which are optimized for systems with a unit step input. The block diagram of the continuous-time neoclassical controller is shown in Figure 1-1.

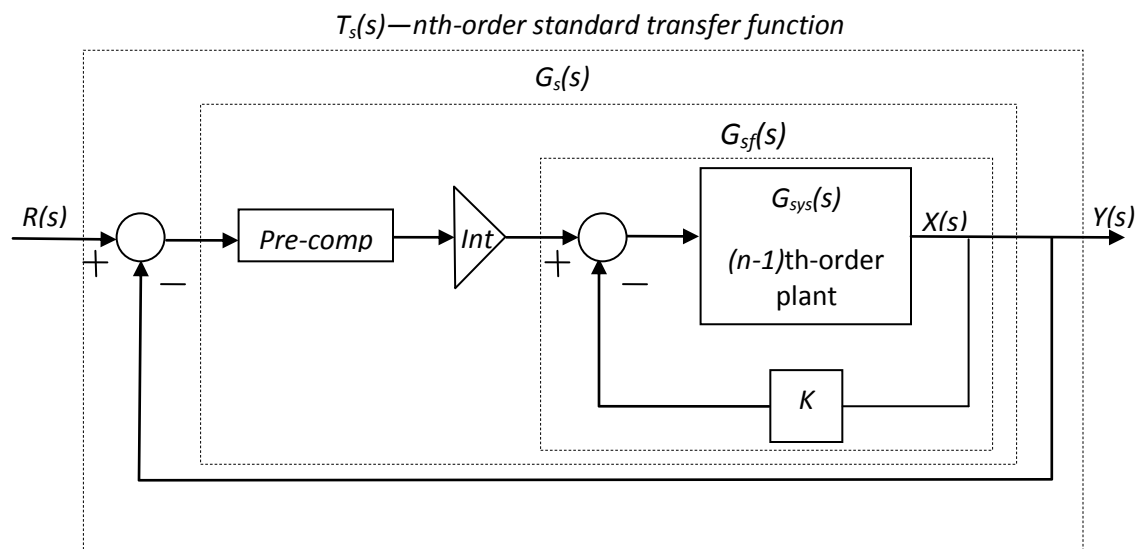


Figure 1- 1: Block diagram of the neoclassically controlled system (1)

In order to match a desired transfer function, with known poles and zeros, the controller must be able to modify the plant's poles to the desired values and cancel the plant's zeroes and replace them with the desired zeroes from the closed-loop transfer function.

Neoclassical Control combines modern control's state-feedback pole placement technique, as shown in the $G_{sf}(s)$ block, with the classical control's zero-pole cancellation in the Pre-compensator(Pre-comp). To eliminate the possibility of a steady-state error for the step response, an integrator (int) from classical control is also used. The integrator's inclusion in the Neoclassical control design raises the order of the system, which forces the controller to match a standard transfer function which is one order higher than the order of the plant. By enclosing these three elements in a unit output feedback, the effect of variations of the plant's parameters on the controller performance is further reduced. This makes the controller more robust. (2)

When the continuous-time neoclassical control design was tested on a linearized mass-spring damper (MSD) system and a linearized magnetic levitation (ML) system, the results were the successful application of the control method (1). The step response matched the response of the ITAE standard transfer function, the steady state error was zero and the settling time was 1 second (1). These parameters were achieved without the need for the trial and error, time consuming root locus-based techniques of classical control. The neoclassical controller makes the design process more efficient and less iterative.

The goal of this thesis is to expand on the current theory of neoclassical control design. The number of standard transfer functions used in the neoclassical controller will be increased with the inclusion of the Butterworth transfer function and the Bessel transfer function. The neoclassical controller will be applied to the 4th order ball and beam system. The ball and beam system is a type 4 system. This will mark the first time that neoclassical control design has been used on a system which is not type zero. The neoclassical controller will be designed in discrete-time for the first time. An observer will also be incorporated into the neoclassical control design to allow for state estimation.

This thesis will take a closer look at continuous-time neoclassical control in chapter 2. This analysis will include a look at three types of standard optimal transfer functions, the previously used Integral of Time multiplied by the Absolute Error (ITAE) transfer function, the Bessel transfer function, and the Butterworth transfer function. This discussion will be followed by an examination of discrete-time neoclassical control in Chapter 3. This chapter will include a discussion about the discretization of continuous-time systems and standard transfer functions. The 4th chapter will display the results of the case studies for the discrete-time neoclassical controller. The 5th chapter will take a second look at the benchmark systems from the previous case studies with addition of an observer for the case where the state is not available for feedback. The last chapter will be a summary of the thesis, ideas for future expansion of the research, and the conclusion.

Chapter 2: Continuous-time Neoclassical Control Review

In order to adequately investigate discrete-time neoclassical control, the current understanding of continuous-time neoclassical control must be reinforced and expanded. Three standard transfer functions will be used in the design of the neoclassical controllers: the Bessel, Butterworth, and ITAE transfer functions. The systems for which the neoclassical controllers will be designed are the Mass-Spring-Damper, the Magnetic Levitation, and the Ball and Beam systems. Full derivations of the models of all systems will be presented in chapter 4.

2.1: Neoclassical components

As stated in chapter 1, neoclassical control combines state-feedback control with zero-pole cancelation in the pre-compensator. The neoclassical controller design is intended for use on linear, time-invariant (LTI), single input single output (SISO) systems. An integrator is added in series to eliminate steady-state error for the step response. The loop is then closed using unity gain output feedback to improve the robustness of the controller and to set the constant term of the denominator. The block diagram for the neoclassical control block diagram can be seen in Figure 2-1. The purpose of this chapter is to highlight the important components of neoclassical control theory and to show how each component is brought together into the neoclassical control design.

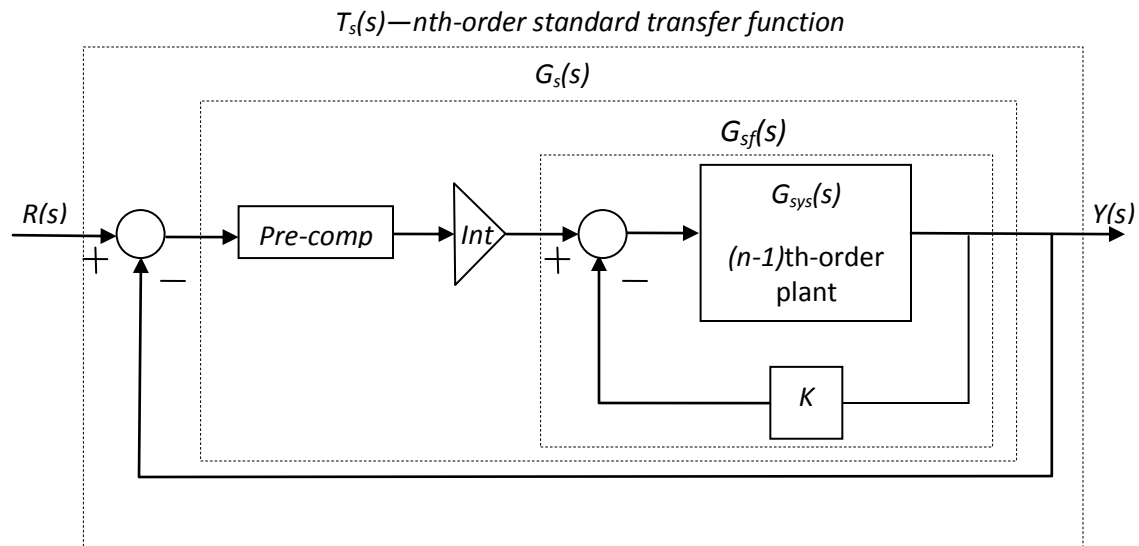


Figure 2- 1: Block diagram of the neoclassically controlled system

2.1.1: The Plant

The plant is the system for which the controller will be designed. The dynamics of the $(n-1)$ th order system can be represented by both a state-space model and a transfer function as,

$$\text{State-space model} \quad \begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (2.1)$$

$$\text{Transfer function} \quad H(s) = \frac{b_{n-1}s^{n-1} + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}, \quad (2.2)$$

where it is assumed that $b_n=0$, which will be the case in all examples in this thesis.

The transfer function is obtained from the state-space model using the formula,

$$H(s) = C(sI - A)^{-1}B + D. \quad (2.3)$$

Since one of the elements of neoclassical control involves pole-placement, it is essential that the plant be controllable. A system is considered controllable if the system can start at any arbitrary initial state and be driven to an arbitrary final state in finite time

with a finite magnitude control action. Controllability of the system can be determined by determining the rank of the controllability matrix,

$$W_c = [B \quad AB \quad \dots \quad A^{n-1}B], \quad (2.4)$$

where A and B are the matrices from the state-space model of the system. If the controllability matrix is full rank, the system is controllable.

2.1.2: State-feedback

State-feedback control makes it possible to easily adjust the pole locations of a controllable system. The concept starts with a state-space model of the system.

$$\dot{x} = Ax + Bu \quad (2.5a)$$

$$y = Cx + Du. \quad (2.5b)$$

The input, u, is set to be the linear combination of the reference signal, r, and the control input, u_c . The control input is defined as,

$$u_c = -Kx. \quad (2.6)$$

Substituting u into the state equation yields:

$$\dot{x} = (A - BK)x + Br. \quad (2.7)$$

The state-feedback allows the poles of the plant to be altered in order to match the desired overall transfer function by choosing the gain, K. In the case of neoclassical control, pole placement is used so that when the integrator and pre-compensator are put in series with the state-feedback loop and the unity gain output feedback is used to

enclose the system, the resulting closed-loop transfer function will be equivalent to the standard transfer function of choice.

The control gain, K , can be calculated by hand using Ackermann's formula (3),

$$K = [(p_1 - \alpha_1)(p_2 - \alpha_2) \cdots (p_n - \alpha_n)] \overline{W}_c W_c^{-1} \quad (2.8)$$

where \overline{W}_c is the controllability matrix of the controllable canonical form of the system.

The parameters, p_n , are the desired pole locations of the system while the parameters, α_n , are the pole locations of the system to be controlled.

2.1.3: The pre-compensator, integrator and output feedback

The pre-compensator is the ratio of the desired numerator and the plant's numerator. The purpose of the pre-compensator is to replace the numerator of the plant with the desired standard transfer function's numerator. The integrator is used to eliminate steady state error of type-zero systems. The continuous-time integrator is represented as $1/s$. However, the integrator will also raise the order of the system by one, which is why neoclassical control matches the closed-loop transfer function to a standard transfer function that is one order higher than the plant. The unit output feedback is used to adjust the constant term of the denominator and to improve the controller's robustness.

2.2: Standard Transfer Functions

In choosing the standard transfer functions that are used for neoclassical control, there are many factors to consider. These factors include the rise time, the overshoot, and the settling time resulting from the step responses of these transfer functions. Rise time is the interval of time required for the step response of a system to go from 10% to 90% of

its final value. Overshoot is the percentage difference between the steady state value and the maximum value of the step response. Settling time is the minimum time required for the system response to remain within a band of $\pm 2\%$ of the steady state value (4) (2);

In this thesis, three standard optimal transfer functions will be used: Integral of Time multiplied by Absolute Error (ITAE), Butterworth, and Bessel. These transfer functions have an optimality property in either the frequency- or the time-domain response. All three of these standard transfer functions take an all pole form,

$$H(s) = \frac{a_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} \quad (2.9)$$

for an nth order system.

2.2.1: ITAE Transfer Functions

The ITAE transfer function is optimized to minimize the effect of the initial error over time for a unit step input (5). The mathematical representation of the error function is,

$$ITAE = \int_{t_0}^{t_f} t |e(t)| dt . \quad (2.10)$$

The ITAE was used in previous work on continuous-time neoclassical control (1). For a step input, the denominators of the transfer functions that adhere to the ITAE optimization for a step input are,

$$2^{\text{nd}} \text{ order} \quad s^2 + 1.4 \omega_n s + \omega_n^2, \quad (2.11a)$$

$$3^{\text{rd}} \text{ order} \quad s^3 + 1.75\omega_n s^2 + 2.15\omega_n^2 s + \omega_n^3, \quad (2.11b)$$

$$4^{\text{th}} \text{ order} \quad s^4 + 2.1\omega_n s^3 + 3.4\omega_n^2 s^2 + 2.7\omega_n^3 s + \omega_n^4, \quad (2.11c)$$

$$5^{\text{th}} \text{ order} \quad s^5 + 2.8\omega_n s^4 + 5\omega_n^2 s^3 + 5.5\omega_n^3 s^2 + 3.4\omega_n^4 s + \omega_n^5, \quad (2.11d)$$

where ω_n is the natural frequency(2,6). The ITAE transfer functions can be scaled by setting the natural frequency to a desired 2% settling time, $T_{s,\text{desired}}$, and then find the normalized ITAE transfer function's settling time, $T_{s,\text{norm}}$. The normalized transfer functions can be found by setting ω_n equal to 1. Once the settling time of the normalized ITAE transfer function is found, then the equation,

$$\omega_n = \frac{T_{s,\text{norm}}}{T_{s,\text{desired}}}, \quad (2.12)$$

is used to find the ω_n which scales ITAE transfer function to the desired settling time..

The step responses for the ITAE transfer function with a settling time of 1 second are simulated and the results are displayed below.

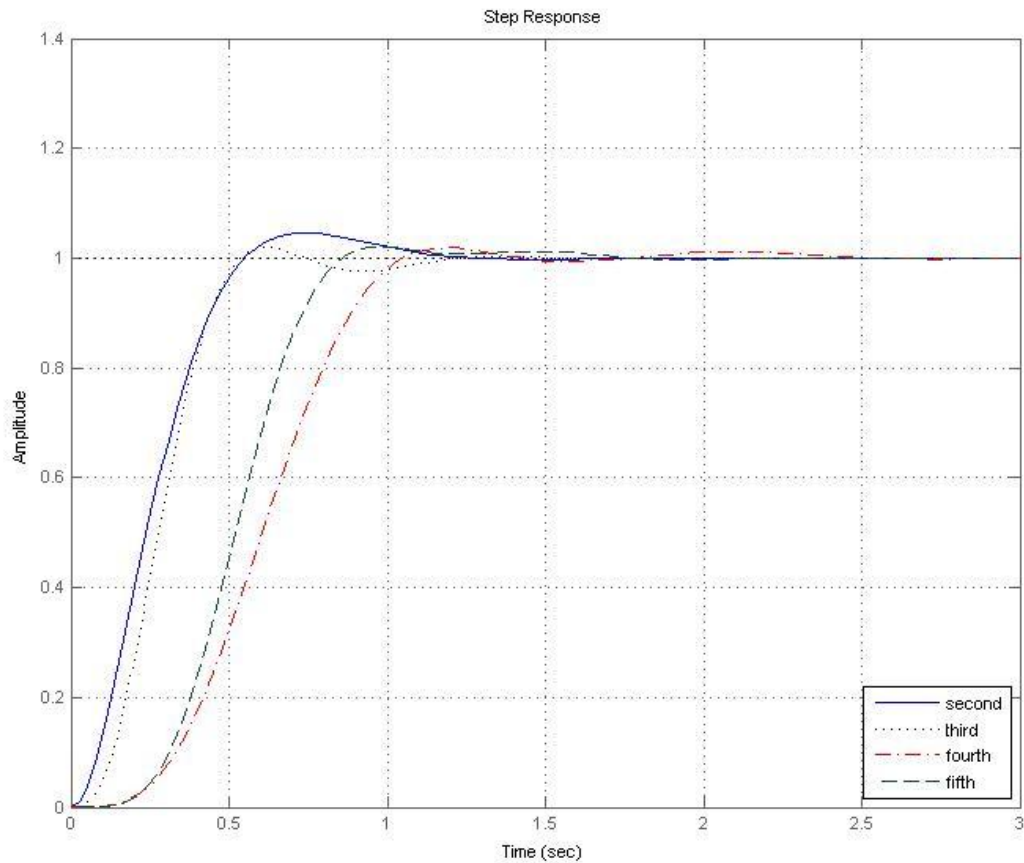


Figure 2- 2: 2nd through 5th order of ITAE optimized transfer function step responses

The step responses settle at the desired settling time. The even numbered orders of ITAE reach the steady state value from above while the odd numbered orders reach the steady state value from below. These transfer functions exhibit a small overshoot.

2.2.2: Butterworth Transfer Functions

The Butterworth transfer function has the property of being maximally flat in the pass band of the frequency response (7). For a step input, the denominators of the normalized Butterworth transfer functions are:

$$2^{\text{nd}} \text{ order} \quad s^2 + \sqrt{2}s + 1 \quad (2.13a)$$

$$3^{\text{rd}} \text{ order} \quad s^3 + 2s^2 + 2s + 1 \quad (2.13b)$$

$$4^{\text{th}} \text{ order} \quad s^4 + 2.6131s^3 + 3.4142s^2 + 2.6131s + 1 \quad (2.13c)$$

$$5^{\text{th}} \text{ order} \quad s^5 + 3.236 s^4 + 5.236 s^3 + 5.236 s^2 + 3.236s + 1 \quad (2.13d)$$

The Butterworth transfer function can be scaled by the 2% settling time of the normalized Butterworth transfer function. The roots of the normalized Butterworth transfer function, p_n , are used to find the pole locations for the desired settling time, p_d , through the following equation:

$$p_d = \frac{T_{s,norm}}{T_{s,desired}} p_n. \quad (2.14)$$

The step responses for the Butterworth standard transfer functions with a settling time of 1 second are shown below.

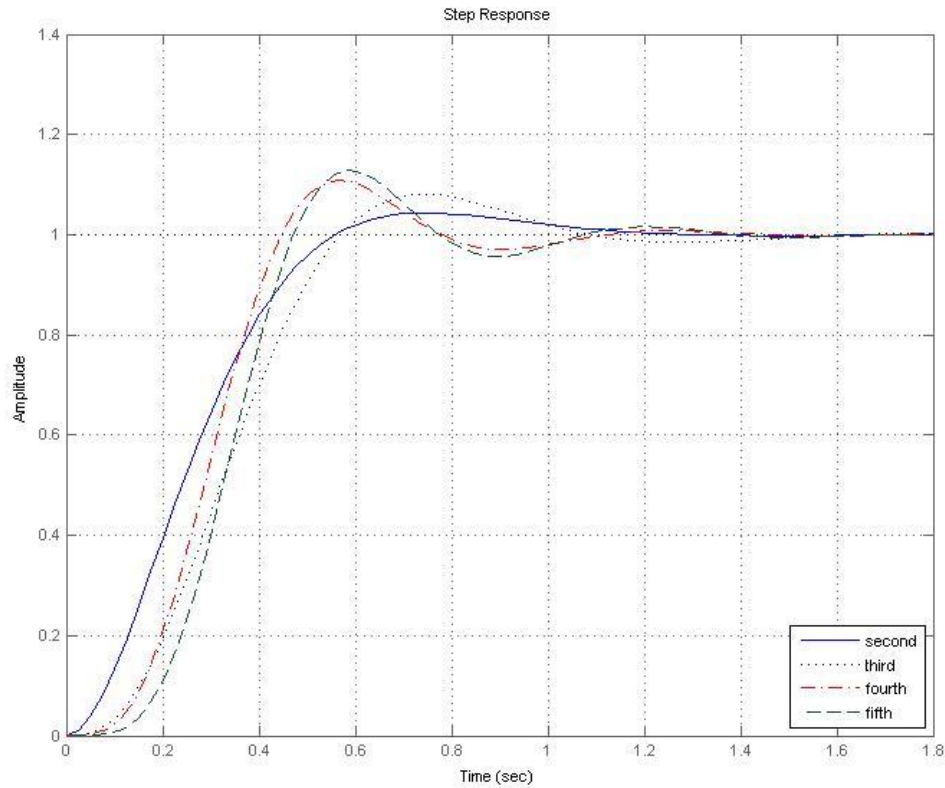


Figure 2- 3: 2nd through 5th order of Butterworth optimized transfer function step responses

The Butterworth step response's percent overshoot increases as the order of the transfer function increases. However all orders do show a settling time of 1 second. The overshoot does appear to be higher than the overshoot seen with the ITAE step response.

2.2.3: Bessel Transfer Functions

The step response of the Bessel transfer functions exhibits minimal overshoot (3). For step inputs, the denominators of the Bessel transfer functions with a 2% settling time of 1 second are,

$$2^{\text{nd}} \text{ order} \quad s^2 + 7.53s + 18.9, \quad (2.15a)$$

$$3^{\text{rd}} \text{ order} \quad s^3 + 12.43s^2 + 64.33s + 133.2, \quad (2.15b)$$

$$4^{\text{th}} \text{ order} \quad s^4 + 19,09s^3 + 163.9s^2 + 730.2s + 1394, \quad (2.15c)$$

$$5^{\text{th}} \text{ order} \quad s^5 + 25.61 s^4 + 306 s^3 + 2090 s^2 + 8026 s + 13700, \quad (2.15d)$$

To scale the Bessel polynomial, the equation,

$$p_d = \frac{1}{T_{s.\text{desired}}} p_n, \quad (2.16)$$

is used, where p_n is the pole for the Bessel polynomial and p_d is the scaled Bessel polynomial. The roots of the Bessel polynomials with a 1% settling time of 1 second can be found in (3).

The simulated step responses of the 2nd through 5th order Bessel transfer functions are shown in Figure 2-4.

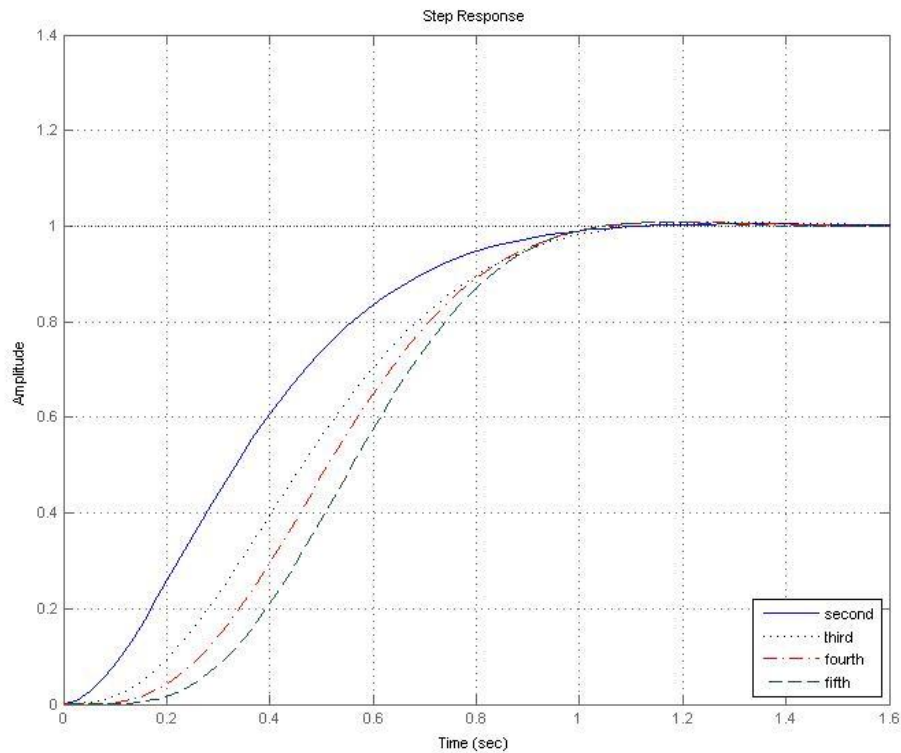


Figure 2- 4: 2nd through 5th order of Bessel optimized transfer function step responses

All of the Bessel responses have the minimal overshoot that is expected. The response slows as the order increases and they all settle at 1 second.

This thesis will use all three standard optimal transfer functions to demonstrate the flexibility neoclassical control has for the designer. The transient responses are already pre-determined by each standard transfer function and the only design parameter the designer needs to pick is the settling time.

2.3: Implementation of the continuous-time neoclassical controller.

The implementation of the neoclassical controller can be shown using a transfer function representation. In order to design a neoclassical controller for a LTI, SISO system, the state-feedback controller gains need to be found. Starting with the desired transfer function,

$$T(s) = \frac{a_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}, \quad (2.17)$$

the unity gain output feedback is removed, resulting in the function,

$$G(s) = \frac{T(s)}{1-T(s)} \quad (2.18)$$

or

$$G(s) = \frac{a_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s}. \quad (2.19)$$

The next step is to remove integrator:

$$G(s) = \left(\frac{1}{s}\right) \frac{a_0}{s^{n-1} + a_{n-1}s^{n-2} + \dots + a_1} \quad (2.21)$$

$$G_{sf}(s) = G(s) \Big/ \frac{1}{s} \quad (2.22)$$

$$G_{sf}(s) = \frac{a_0}{s^{n-1} + a_{n-1}s^{n-2} + \dots + a_1} \quad (2.23)$$

Now that the transfer function for the inner loop in Figure 2-1 has been derived, the poles of $G_{sf}(s)$ can be used to find the desired poles for the block.

With the poles for $G_{sf}(s)$ and plant model known, the feedback gain, K , can be calculated and applied to the plant. The integrator is then put in series with the plant. The

pre-compensator is also put in series to cancel and replace the numerator of the open-loop system. The unity gain output feedback is used to close the outer loop and the resulting system will be equivalent to the desired optimal transfer function. Figure 2-5 summarizes the design process.

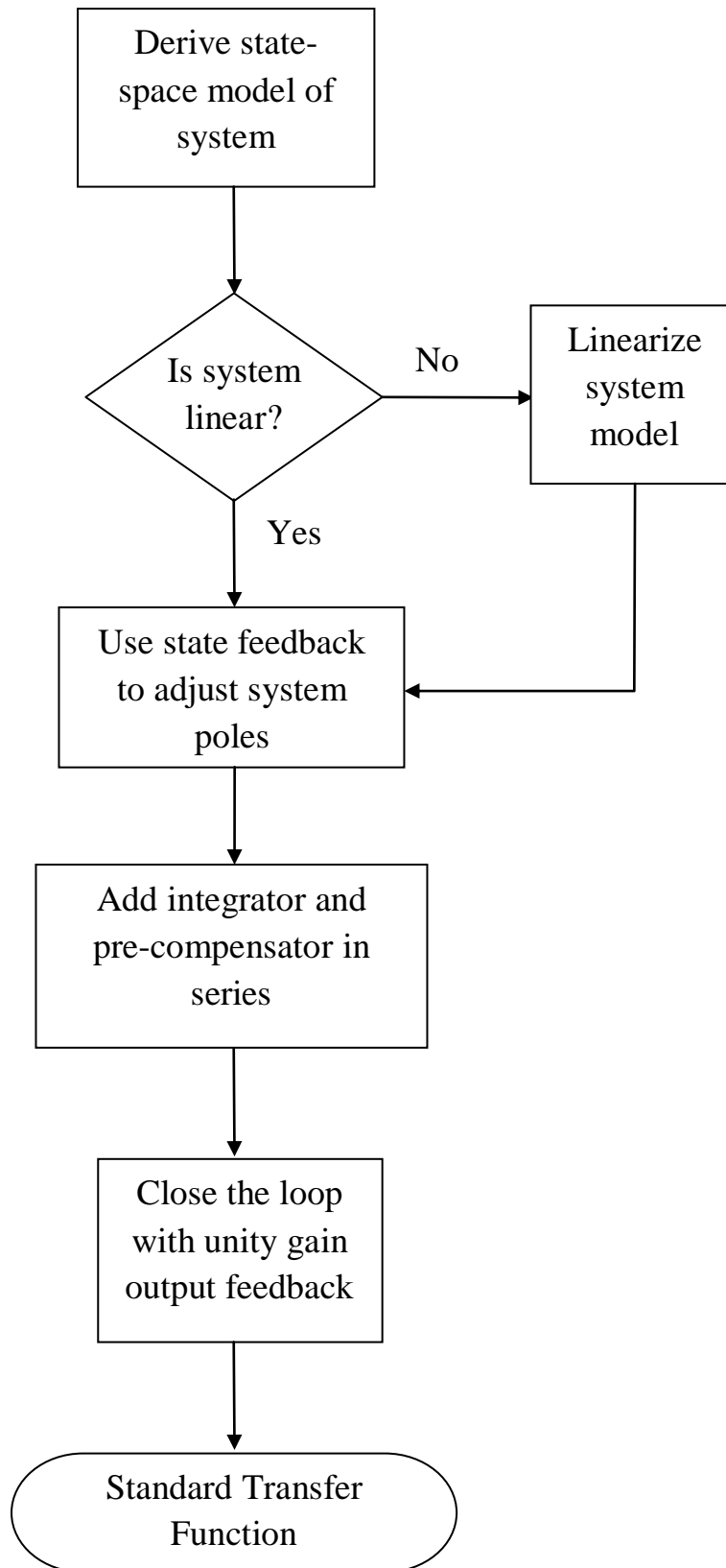


Figure 2- 5: Neoclassical controller design flow chart.

2.4: Limitations of neoclassical control

Neoclassical control works for LTI and SISO systems. While there are well defined cases where neoclassical control works well, the cases where neoclassical control might not work have not been closely examined to this point. For continuous-time systems, if the zeroes of the open-loop system are in the right-half complex plane, the neoclassical controller can become unstable. This instability is due to the pre-compensator, which adds the poles necessary to cancel out the finite zeroes of the plant. Theoretically, this is not a problem since the zeros and the poles will still cancel. However, in practice, the zeros and the poles would need to be exactly the same to cancel out; even the slightest difference in the pole location and the zero location will make the system unstable due to the imperfection of the attempted zero-pole cancellation. In the real world, exact zero-pole cancellation is nearly impossible.

2.5: Conclusion

This chapter has reviewed the continuous-time neoclassical control by investigating the various components and how they are put together. This chapter has introduced the standard optimal transfer functions which will be used in the design of the neoclassical controllers. In the next chapter, the neoclassical control theory will be applied to the discrete-time domain. An observer will also be introduced to the neoclassical control theory to estimate unknown states.

Chapter 3: Theory of Discrete-time Neoclassical Control

This chapter will focus on the neoclassical control theory from a discrete-time control perspective. This chapter will discuss the discretization process, revisit all of the pieces of the neoclassical control design with an emphasis on discrete-time operation, and explore the limitations that can influence the way neoclassical control is applied. At the end of the chapter, an observer will be incorporated into the neoclassical control design to allow for estimation of unknown states.

3.1 Overview of discrete-time neoclassical control

In recent years, technology has stepped away from continuous-time systems in favor of discrete-time systems. Discrete-time technology uses samples of a continuous-time system response, which amounts to a low-pass filtering operation. Therefore the effects of wideband noises are attenuated. Since control of discrete-time systems is desired from this viewpoint, an understanding of discrete-time control is necessary.

Discrete-time system models can take the form of a transfer function or a state-space model, just like continuous-time systems. Transfer functions represent the input-output relationship of LTI, SISO systems. In continuous-time, transfer functions are system representations in the s -domain. For discrete-time systems, transfer functions are system representations in the z -domain.

The discrete-time neoclassical model is shown in Figure 3.1. The block diagram is essentially the same as the continuous-time neoclassical controller block diagram. However, there are differences between the application of neoclassical control in continuous time and discrete time. This section will explore these differences,

particularly the development of the discrete-time model of the plant, the closed-loop standard transfer function, and the discrete-time form of the integrator.

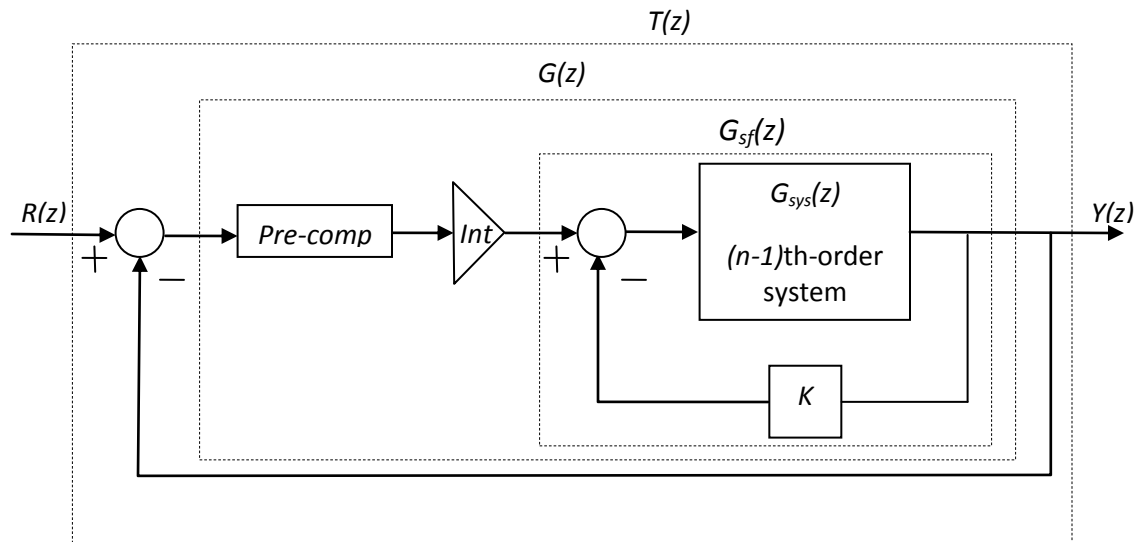


Figure 3- 1: Discrete-time neoclassical controller block diagram

3.2: Discrete-time Plant

To apply discrete-time control to continuous-time systems, the system must be discretized or transformed into the discrete-time domain. Before discretization can happen, it is important to choose a proper sampling interval, T . A sampling interval that is too large can miss variations in the system response, leading to misrepresentation of the system dynamics. However, making the sampling interval too small can lead to computational difficulty and could end up storing redundant information that does not improve the quality of the control action. The sampling interval that will be used throughout this thesis is 0.01 s. A more detailed method that can be used to choose a sampling interval can be found in (3).

3.2.1: Zero Order Hold (ZOH)

There are many forms of discretization. Two of the most commonly used are the zero order hold (ZOH) and the Euler approximation. A continuous-time state-space model:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du,\end{aligned}\tag{3.1}$$

is transformed into a ZOH equivalent discrete-time state-space model:

$$\begin{aligned}x_{k+1} &= \Phi x_k + \Gamma u_k \\ y_k &= Cx_k + Du_k,\end{aligned}\tag{3.2}$$

which, by construction, exactly matches the output of the continuous-time model at the sampling instants.

The discrete-time system matrices, Φ and Γ , are obtained from the continuous-time matrices by via the following equations (3):

$$\Phi = e^{AT}\tag{3.3}$$

$$\Gamma = \int_0^T e^{A\tau} B d\tau\tag{3.4}$$

whereas the matrices in the measurement equation of both the continuous-time and discrete-time systems are the same

The poles, z_{pi} , of the ZOH model are related to the poles, s_{pi} , of the continuous time model by the ZOH pole mapping function,

$$z_{pi} = e^{s_{pi}T}, \quad (3.5)$$

where T represents the sampling period (3).

The ZOH maps poles in the left-hand-plane of the s -domain map to poles within the unit circle in the z -domain. Poles in the right-hand-plane in the s -domain are mapped outside of the unit circle in the z -domain. Poles on the $j\omega$ -axis in the s -domain are mapped onto the unit circle in the z -domain.

3.2.2: Euler Approximation

The first order Euler Approximation uses the Taylor series expansion of the ZOH and eliminates all terms in T higher than the first order in the state matrices;

$$\Phi \approx I + AT, \quad (3.6)$$

$$\Gamma \approx TB. \quad (3.7)$$

In the Euler Approximation, the discrete-time system will have the same number of zeroes in the transfer function as the continuous-time system. If the continuous-time system had zeroes in the right half of the complex plane, then the Euler approximation of that system will have the same number of zeroes outside of the unit circle.

3.3: Discretization of the continuous-time integrator

The s-domain transfer function representation of the integrator is $1/s$. The z-transform can be used to transform this expression from the s-domain to the z-domain.

The continuous-time integrator can be z-transformed into the following form:

$$\frac{1}{s} \rightarrow \frac{Tz}{z-1} = \frac{Y(z)}{U(z)} \quad (3.8)$$

However, the use of the z-transform can introduce a half interval advance into the system. The inherent causality problem becomes an issue in the implementation of the system. To compensate for the causality problem, the modified z-transform is used with one step delay to obtain (8):

$$\frac{1}{s} \rightarrow \frac{T}{z-1} = \frac{Y(z)}{U(z)} \quad (3.9)$$

3.4: Observer Based Discrete-time Neoclassical Control

Until now, neoclassical control design has been based on the assumption that all of the state information is known. However, in real world problems, not all the state information may be known or measurable. In order to be able to control such systems, modern control theory provides the observer to estimate the unknown states. Before an observer can be designed for the system, it must be established that the system is observable. The test for observability involves constructing the observability matrix, W_o , using the system matrices,

$$W_o = \begin{bmatrix} C \\ C\Phi \\ C\Phi^2 \\ \vdots \\ C\Phi^{n-1} \end{bmatrix} \quad (3.10)$$

If W_o is full rank, then the system is observable.

3.4.1: Observer implementation

Observers can be easily incorporated in the neoclassical control design. There are two types of observers, full-order observers and reduced-order observers. For this thesis, full-order observers will be used. Figure 3.2 shows the incorporation of the observer into the neoclassical control block diagram.

\hat{x}

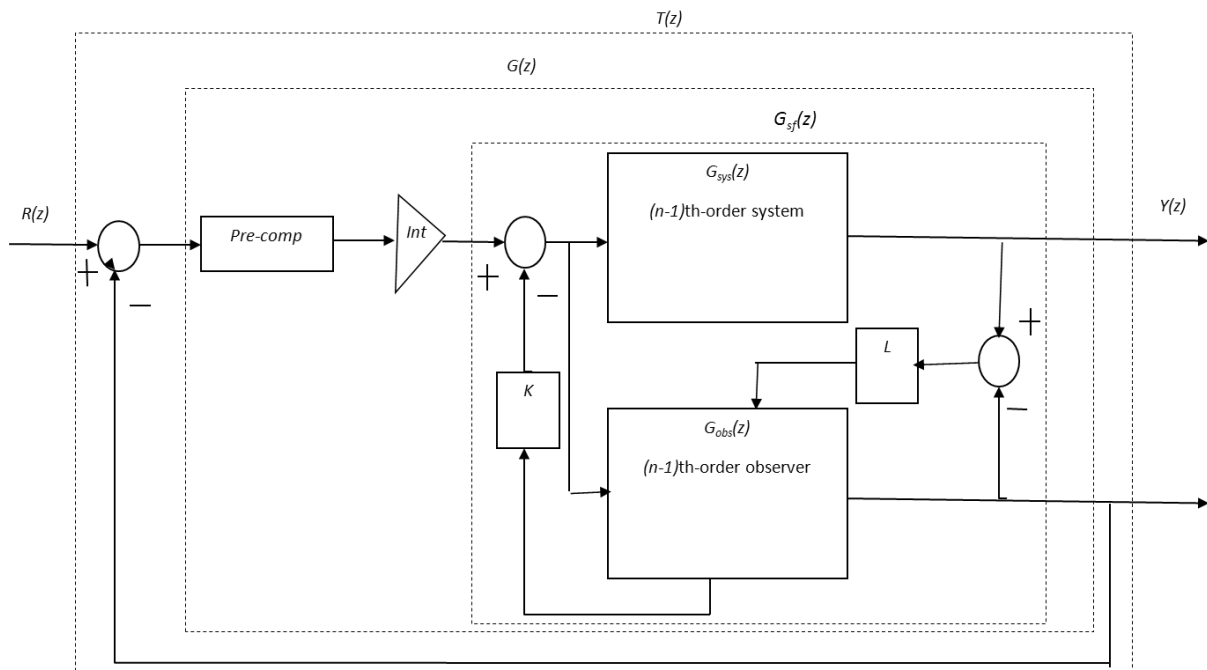


Figure 3- 2: Discrete-time neoclassical controller block diagram with an observer

For a discrete-time system with no feed forward term, which all of our systems will obey,

$$x_{k+1} = \Phi x_k + \Gamma u_k \quad (3.11a)$$

$$y_k = C x_k' \quad (3.11b)$$

a closed-loop observer can be defined as,

$$\hat{x}_{k+1} = \Phi \hat{x}_k + \Gamma u_k + L(y_k - C \hat{x}_k), \quad (3.12)$$

where \hat{x}_k represents the estimate of the state. The term $(y_k - C \hat{x}_k)$ is a measure of the difference between the output of the actual system and the output obtained from the observer. This term weighted by the observer gain, L , is used to improve the state estimate.

Defining the error of the system as,

$$e_k = x_k - \hat{x}_k \quad (3.13)$$

the error update equation can be derived from the state update equation and the estimate update equation as,

$$e_{k+1} = (\Phi - LC)e_k. \quad (3.14)$$

The solution to this error update equation is,

$$e_k = (\Phi - LC)^k e_0, \quad (3.15)$$

where e_0 is the initial error in the estimate. The purpose of observer design is to choose L such that the eigenvalues of $(\Phi - LC)$ are within the unit circle. By placing the eigenvalues of $(\Phi - LC)$ inside the unit circle, it is guaranteed that the error will be driven to zero in time, no matter how large the initial error may be.

The observer settling time is the time it takes for the estimate error to go to zero. For discrete-time systems, the observer settling time must be less than the settling time of the controller by a factor of ten. This is achieved by placing the poles of the observer in

the location to get the desired error response. The observer does not affect the overall transfer function of the closed-loop system. When the state equations in x_k and e_k are combined, the closed-loop system is represented as,

$$\begin{pmatrix} x_{k+1} \\ e_{k+1} \end{pmatrix} = \begin{bmatrix} \Phi - \Gamma K & \Gamma K \\ 0 & \Phi - LC \end{bmatrix} \begin{bmatrix} x_k \\ e_k \end{bmatrix}. \quad (3.16)$$

The challenge of implementing the neoclassical control technique on a system with an observer is to do so without losing the information from the observer. As discussed previously, transfer functions were developed as a part of classical control theory. Transfer functions only represent the input-output relation of the system; they don't provide information about the state of the system. The effect of using state-feedback on a system can be seen in a transfer function representation. To this point, the neoclassical control design procedure could be represented at every step with a transfer function. However, the observer transfer function representation will not provide any information on the state estimates. Therefore, in order to apply the controller gains, K , to the state estimated system, the design procedure must be done in state-space representation to preserve the observer information. Fortunately, modern control theory has already developed a way to integrate an observer into a controller at the same time without losing important observer information.

The observer, combined with the state-feedback model forms the state-feedback block, $G_{sf}(z)$. The integrator and the pre-compensator can be put in series with the state-feedback system. Using unity gain output feedback, the loop is closed and the overall

transfer function is equivalent to the desired transfer function and the error in the initial guess is minimized within the desired settling time.

3.5: Conclusion

The discrete-time neoclassical control theory has been described in this chapter. The pieces of the neoclassical control system block diagram have been explained. The limitations of using the ZOH to discretize the system have been noted, as has the way to control continuous-time systems with zeroes in the right half complex plane using discrete-time neoclassical control. In Chapter 4, the neoclassical control design will be applied to stable and unstable systems of various orders. In Chapter 5, the state observers that have been discussed in this chapter will also be implemented for the same systems.

Chapter 4: Case Studies without Observers

In order to test the application of the discrete-time neoclassical controller, there will be three systems used in case studies, a mass-spring-damper, a magnetic levitation, and a ball and beam. After using the design process, all three systems will have step responses which will match the three desired standard transfer function; ITAE, Bessel, and Butterworth.

4.1: The Mass-Spring Damper system

A mass on a spring and dashpot (damper) is often used as an example of harmonic motion. The differential equation comes from Newton's laws and the behavioral equations of springs and dashpots.

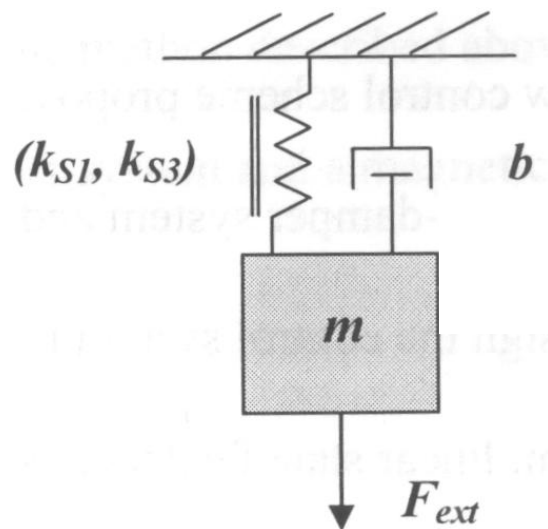


Figure 4- 1: A diagram of the Mass-Spring-Damper system with relevant force vectors (1)

The component forces can be seen in Figure 4-1 or from a free body diagram of the system. In Figure 4-1, m is the mass, k_{S1} is the linear spring constant, k_{S3} is the non-linear spring constant, b is the resistance to motion due to the dashpot, and F_{ext} is an external force applied to the system. The equation of motion for the mass on the spring with a dashpot is (1):

$$m\ddot{x} = -b\dot{x} - k_{S1}x - k_{S3}x^3 + F_{ext} . \quad (4.1)$$

The system needs to be linearized before the neoclassical control design can begin. This linearization is achieved simply by setting k_{S3} to zero. By assuming $x_1=x$ and $x_2=\dot{x}$, the state-space model is given as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{b}{m}x_2 - \frac{k_{S1}}{m}x_1 + \frac{F_{ext}}{m} . \end{aligned} \quad (4.2)$$

Equation 4.2 can be placed into vector- matrix form as,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_{S1}}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F_{ext} \quad (4.3)$$

Assuming that the displacement of the mass with respect to its rest position is the measured quantity, the output or measurement equation is

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u . \quad (4.4)$$

For the purposes of simulation, the parameters are assigned the values shown in Table 4-1. With these values, the system exhibits a slightly underdamped step response.

m	b	k_{S1}	k_{S3}
(kg)	(N·sec/m)	(N/m)	(N/m ³)
1	3	9	9

Table 4- 1: Table of parameters to be used for the Mass-Spring-Damper system

Using a sampling period of T=0.01 seconds, the discrete-time state space representation of the system can be derived via the ZOH. The discrete-time state-space equations are:

$$\begin{aligned}
 x[k+1] &= \begin{bmatrix} 0.9996 & 0.0099 \\ -0.0887 & 0.97 \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0.0099 \end{bmatrix} u[k] \\
 y[k] &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u[k]
 \end{aligned} \tag{4.5}$$

A unit step input is applied to the discrete-time system at t=0. The open-loop step response of the system is shown below in Figure 4-2.

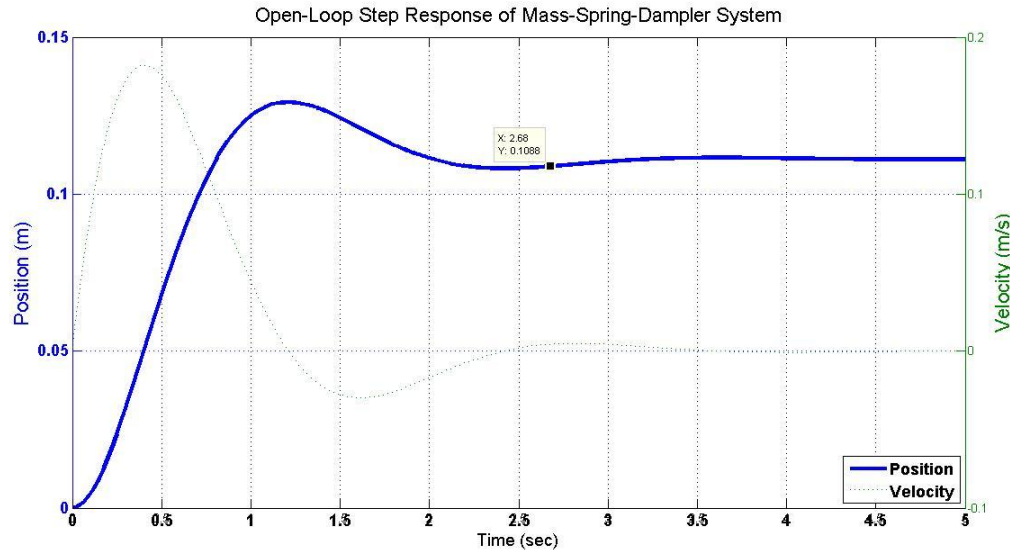


Figure 4- 2 : The open-loop Mass-Spring-Damper system for a unit step input

The open-loop step response of this system has a 2% settling time of 2.6763 seconds. Figure 4-2 shows a significant steady state error for the position vector with respect to the reference input of 1.

In order to control the position variable of the system such that the steady state error goes to zero and the settling time is reached at a specified time, neoclassical control will be used. Applying neoclassical control to the system requires that the control gains be calculated first.

Remember, the goal is displacement control of the Mass-Spring-Damper system to have a 1 meter steady state value of displacement and a settling time of 1 second for a unit step input. It is also desired to have the closed-loop system display behavior consistent with the step response for standard transfer function, whose properties are well established. Since the Mass-Spring-Damper system is a second order system, as explained before, the standard transfer function to which the overall system is matched

must be third order. State-feedback control is used to adjust the poles of the plant to the locations that give the overall desired transfer function. The appropriate discrete time optimal transfer function is chosen for the desired step response. The first of these is the ITAE 3rd order discrete transfer function,

$$T_i(z) = \frac{10^{-5} * (6.911z^2 + 26.74z + 6.469)}{(z - 0.948) * (z^2 - 1.917z + 0.9244)}, \quad (4.6)$$

which represents the desired closed-loop transfer function to be obtained using the neoclassical block diagram shown in Figure 3-1. The closed-loop system needs to be made into an open-loop system by removing the unit feedback.

$$G_i(z) = \frac{T_i(z)}{1 - T_i(z)} \quad (4.7a)$$

$$G_i(z) = \frac{10^{-5} * (6.911z^2 + 26.74z + 6.469)}{(z - 1) * (z^2 - 1.865z + 0.9764)} \quad (4.7b)$$

The next step is to remove the discrete-time integrator. In Chapter 2, it was shown that once a standard transfer function has its unit feedback removed, it will always have a pole at the origin in the s-domain. Poles at the origin in the s-domain map to poles at $z=1$ in the z-domain. Therefore, by removing the unit output feedback from the standard transfer function, there will always be a pole at $z=1$. This fact means that a discrete-time integrator can always be removed from the equation,

$$G_{sfi}(z) = \frac{G_i(z)}{1/(z-1)} \quad (4.8a)$$

$$G_{sfi}(z) = \frac{10^{-5} * (6.911z^2 + 26.74z + 6.469)}{z^2 - 1.865z + 0.9764} \quad (4.8b)$$

$G_{sfi}(z)$ represents the inner loop state-feedback block in Figure 3-1 for a system being designed for an ITAE transfer function. At this point, the poles of $G_{sfi}(z)$, the Φ matrix, and the Γ matrix are known. This is all that is needed to calculate the feedback gains necessary to match the closed-loop response to the ITAE transfer function.

MATLAB's place command or the Ackermann formula can be used to place the poles into the desired positions. The calculated state feedback gain for the ITAE neoclassical controller with a desired settling time of 1 second is,

$$K_i = [109.16 \quad 10.09] \quad (4.9)$$

Now that all the pieces are known, the system can be reassembled again by putting the discrete integrator and pre-compensator in series with $G_{sfi}(z)$. After closing the loop with unity gain feedback, the reconstructed system will have a transfer function that is equivalent to the desired ITAE transfer function.

The step response of the resulting system is shown in Figure 4-3 for the neoclassical control designed to yield the ITAE unit step optimized 3rd order transfer function.

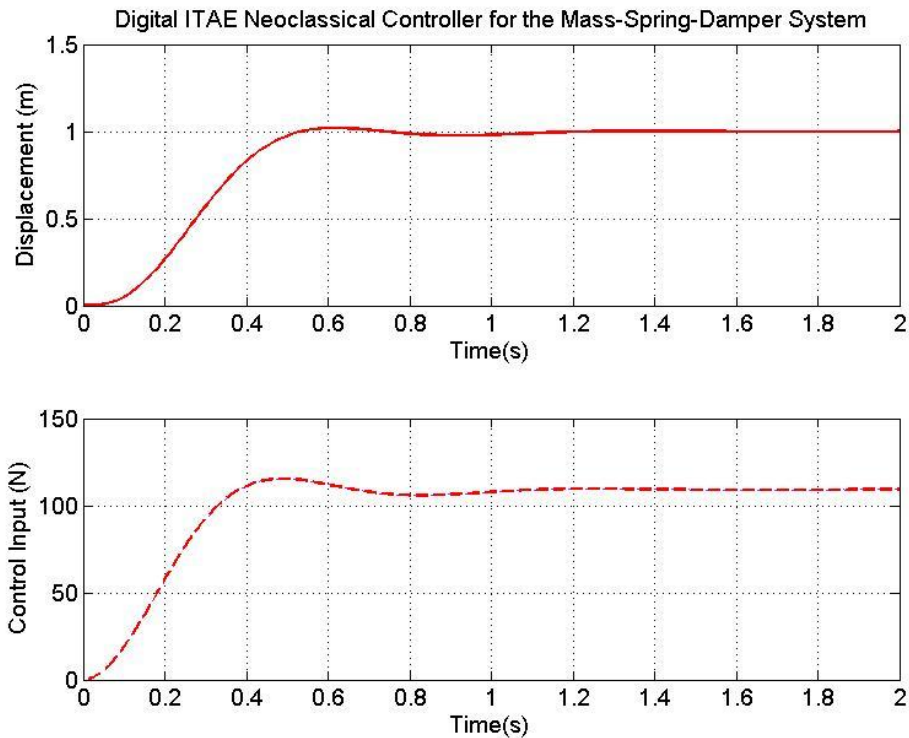


Figure 4- 3: The (a) step response and (b) control input of the neoclassical ITAE controller for the Mass-Spring-Damper system

The 2% settling time for the discrete-time ITAE neoclassically controlled Mass-Spring-Damper is 1 second as per design. The steady state value is also equal to the unit step reference input, 1 m. The success in matching the ITAE standard transfer function to the position of the mass shows that the neoclassical approach works on this stable, linearized, single-input, single-output (SISO) system. The ITAE transfer function is an arbitrary choice of transfer function. The same design procedures can be used for other standard transfer functions. This gives neoclassical control flexibility in choosing the desired transient response for a system. To further display the control capabilities of the neoclassical controller, the system has also been tuned to the Bessel and Butterworth performance criteria, as shown below in Figure 4-4 and 4-5 respectively.

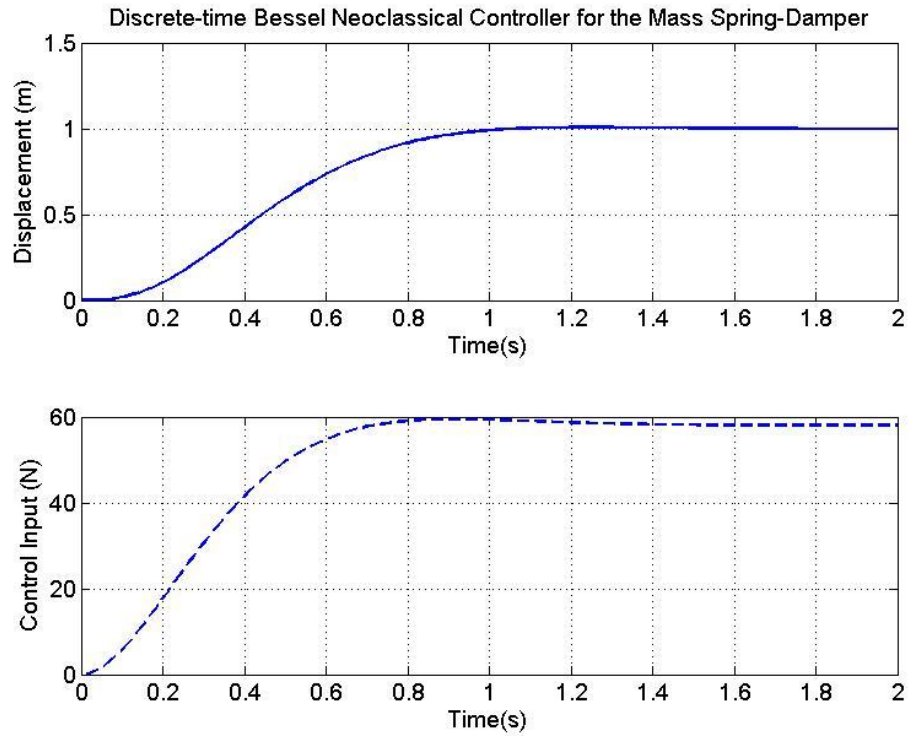


Figure 4- 4: The (a) step response and (b) control input of the neoclassical Bessel controller for the Mass-Spring-Damper system

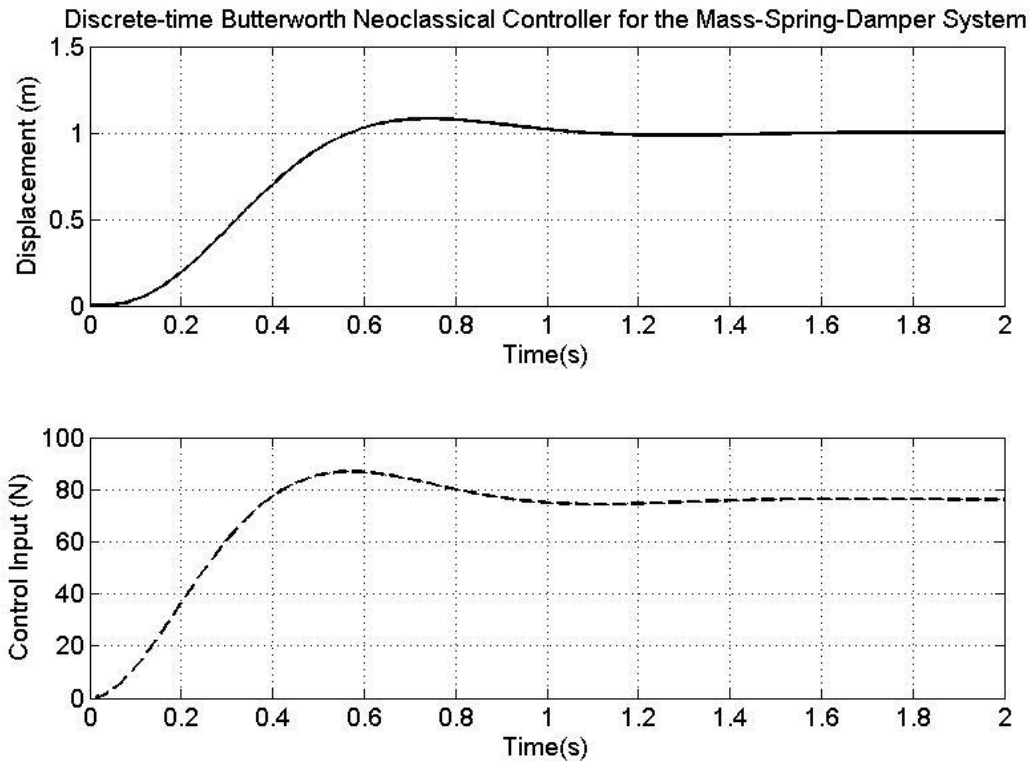


Figure 4- 5: The (a) step response and (b) control input of the neoclassical Butterworth controller for the Mass-Spring-Damper system.

Figures 4-4 and Figure 4-5 show the step response for when the neoclassical controller design procedure is repeated for the Mass-Spring-Dampers using the discrete Butterworth transfer function and the discrete Bessel transfer function. Just like the ITAE example, the desired design criteria have been attained. The desired settling time of 1 second has been achieved and the steady-state value is equal to the magnitude of the reference unit step input. In fact, the primary differences between the ITAE, Butterworth, and Bessel step responses are the transient responses and magnitude of the control input,

$$u_c = Kx \quad (4.10)$$

As Figure 4-6 shows, the control inputs are very different despite the fact that they are acting on the same system.

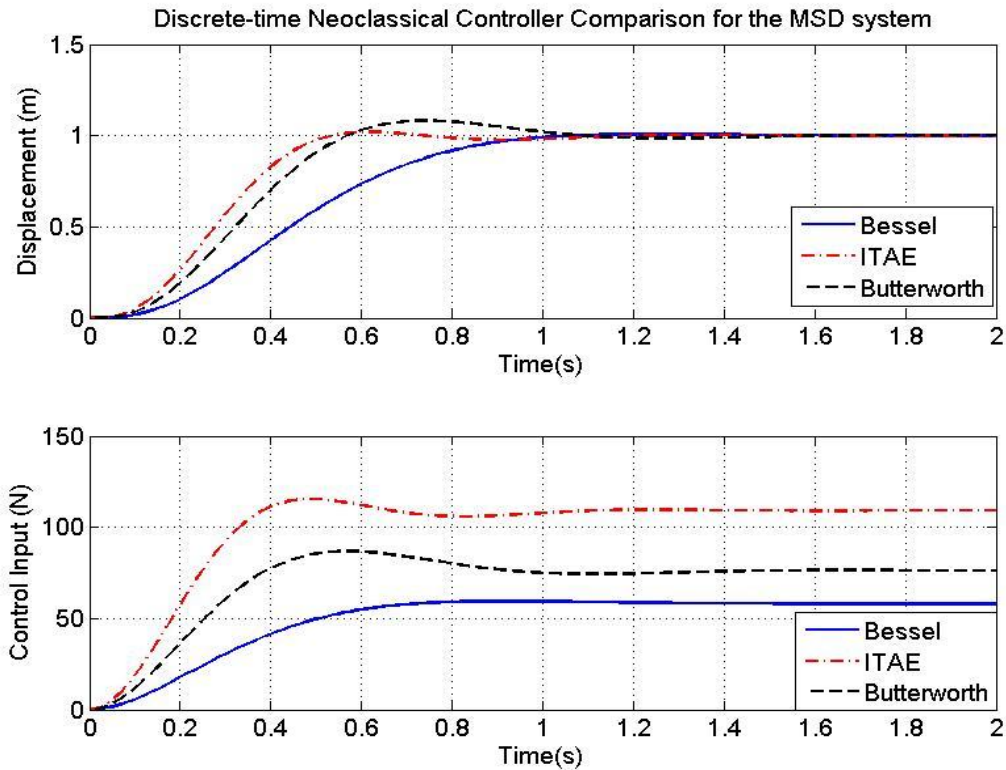


Figure 4- 6: The (a) step responses and (b) control inputs of the neoclassical controllers for the Mass-Spring-Damper system

The control inputs have very different steady-state values. The ITAE actually has the highest control input of all three performance criteria. This means that the ITAE requires more control action on the part of the controller than the others to achieve the desired performance. The steady state value of the control inputs can be calculated as,

$$u_{ss} = Kx_{ss} \quad (4.11)$$

where

$$u_{ss} = K \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (4.12)$$

for a steady state value of one for the position and zero for the velocity. Recalling that the control gain for the ITAE neoclassical controller is

$$K_i=[109.16 \quad 10.09] \quad (4.13)$$

Therefore,

$$u_{ss} = 109.16 \text{ N.} \quad (4.14)$$

This result is displayed on the red ITAE control input line in Figure 4-6 and Figure 4-3.

The results for the steady state value of the control input for the Bessel and the Butterworth transfer functions can also be calculated based on the respective control gains, K ,

$$K_b=[53.011 \quad 9.1234] \quad (4.15)$$

$$u_{ss} = 53.011 \text{ N} \quad (4.16)$$

$$K_{bw}=[76.152 \quad 9.9991] \quad (4.17)$$

$$u_{ss} = 76.152 \text{ N} \quad (4.18)$$

The steady state values for the control input match the results seen in the Figure 4-6.

4.2: Magnetic Levitation system

The discrete-time neoclassical control method has been shown to be effective for an open-loop stable 2nd order system in the previous section. The neoclassical design procedure will now be applied to higher order systems. The magnetic levitation system is a 3rd order system. The magnetic levitation system consists of a metal ball of mass, m ,

subjected to a force due to the magnetic field generated by an electromagnet with current, i , running through it. The current is controlled by the voltage applied to the coil.

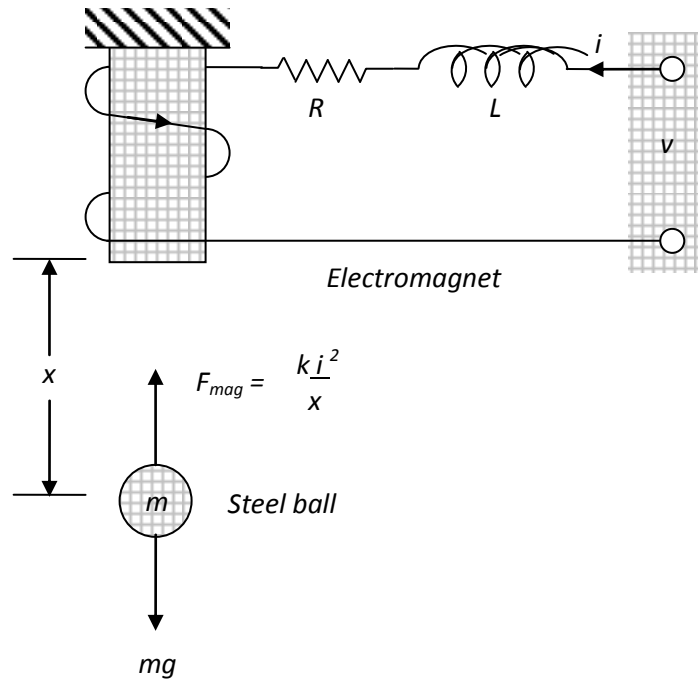


Figure 4- 7: A diagram of the Magnetic Levitation system.

The linearized state-space model for the Magnetic Levitation system can be derived from the differential equations for the system,

$$m \frac{d^2 x}{dt^2} = mg - \frac{k i^2}{x_1} \quad (4.19)$$

and

$$v = iR + L \frac{di}{dt}. \quad (4.20)$$

where m is the mass, R is the resistance, L is the inductance, and x is the position from the bottom of the magnet (1). The magnetic force constant, k , is set equal to one for all magnetic levitation case studies. For this example, the state variables, x_1 , x_2 , and x_3 represent the position of the ball, the velocity of the ball, and the current running through the electromagnet, respectively. The resulting non-linear state equations are:

$$\dot{x}_1(t) = x_2, \quad (4.21a)$$

$$\dot{x}_2(t) = g - \frac{1}{m} \left(\frac{x_3^2}{x_1} \right), \quad (4.21b)$$

and

$$\dot{x}_3(t) = \frac{1}{L} (v - Rx_3). \quad (4.21c)$$

The A and B matrices are linearized about an equilibrium position, $x_{1,0}$ (9). The resulting linear continuous-time state-space model is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g}{x_{1,0}} & 0 & -2\sqrt{\frac{g}{mx_{1,0}}} \\ 0 & 0 & -R/L \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix} v \quad (4.22a)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} v, \quad (4.22b)$$

where the position of the ball is the measured output variable.

The parameters used for simulation are shown in Table 4-2.

m	g	R	L	$x_{1,0}$
(kg)	(m/sec)	(Ω)	(mH)	(m)
0.015	9.81	0.615	143	0.125

Table 4- 2: Table of parameters to be used for the Magnetic Levitation system

A ZOH was used to discretize the magnetic levitation system. This caused an unexpected instability in the step response. An analysis of individual pieces in the design showed that the pre-compensator was the source of the instability. This implied that the numerator of the ZOH equivalent system has one or more zeroes outside of the unit circle. This was numerically confirmed upon analysis of the transfer functions. The continuous-time transfer function for the magnetic levitation system is,

$$G_{sys}(s) = \frac{-1012}{s^3 + 4.301 s^2 - 78.48 s + 337.5} = \frac{-1012}{(s-8.859)(s+8.859)(s+4.301)}. \quad (4.23)$$

The discrete-time transfer function of the magnetic levitation system, using the ZOH method is,

$$G_{sys}(z) = \frac{-0.0001669 z^2 - 0.0006606 z - 0.0001633}{z^3 - 2.966 z^2 + 2.923 z - 0.9579} = \frac{-0.00016688 (z+3.694)(z+0.265)}{(z-0.9579)(z-0.9152)(z-1.093)}. \quad (4.24)$$

The presence of one or more zeroes outside of the unit circle causes instability in the neoclassical controller due to the nature of the pre-compensator, which puts a pole at the same location as the plant's zeroes. Imperfect zero-pole cancellation will destabilize the neoclassical controller.

The continuous-time transfer function has a constant gain in the numerator. In comparison, the discrete-time transfer function not only has finite zeroes, one of the zeroes is outside of the unit circle. The Euler Approximation was chosen as an alternative

discretization method. The discrete-time transfer function of the magnetic levitation system, using the Euler Approximation is,

$$G_{\text{sys}}(z) = \frac{-0.001012}{z^3 - 2.957z^2 + 2.906z - 0.9495} = \frac{-0.001012}{(z-0.957)(z-0.9114)(z-1.089)}. \quad (4.25)$$

The Euler Approximation, unlike the ZOH, does not create any zeroes in the numerator. For this reason, the Euler approximation will be used as a method of bypassing the trouble with the numerator of the ZOH equivalent model.

The Euler Approximation of the continuous-time state-space model of the magnetic levitation system is:

$$x[k+1] = \begin{bmatrix} 1 & 0.01 & 0 \\ 0.7848 & 1 & -1.447 \\ 0 & 0 & 0.957 \end{bmatrix} x[k] + \begin{bmatrix} 0 \\ 0 \\ 0.06993 \end{bmatrix} u[k] \quad (4.26a)$$

$$y[k] = [1 \ 0 \ 0]x[k] \quad (4.26b)$$

The unit step response of this system is shown in Figure 4-8.

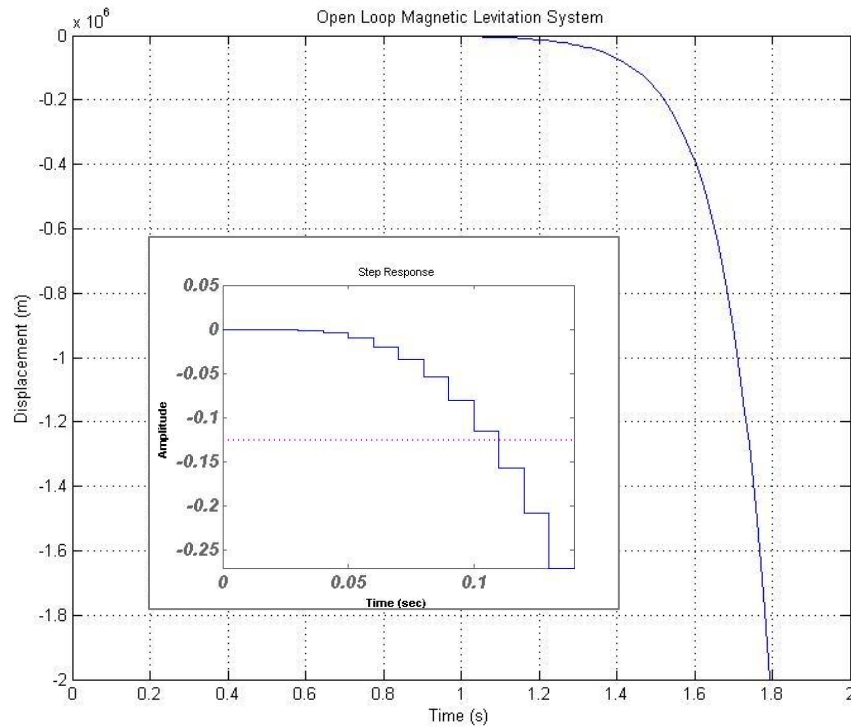


Figure 4- 8: The unit step response for the open-loop Magnetic Levitation system

The displacement goes to negative infinity, which represents the ball rising upwards towards the bottom of the magnet. The dotted line in the inset of Figure 4-8 represents the system constraint, the location of the magnet. The ball cannot move higher than the location of the magnet. The purpose of controlling the magnetic levitation system is not just to get the desired settling time and steady-state value for a step input, but also to stabilize this open-loop unstable system.

Since the magnetic levitation system is a 3rd order system, the neoclassically controlled system is matched to a 4th order standard transfer function. In the case of the ITAE performance criterion, the desired transfer function is:

$$T_{ITAE}(z) = \frac{10^{-7} * (1.7z^3 + 18.35z^2 + 18z + 1.606)}{z^4 - 3.903z^3 + 5.715z^2 - 3.722z - 0.9095} \quad (4.27)$$

Using the same procedure that was previously described for the Mass-Spring-Damper system, the state feedback gains for the magnetic levitation system are calculated to be:

$$K_{ITAE} = [-0.98146 \quad -0.14611 \quad 0.74111] \quad (4.28)$$

This state feedback gain, when used in the neoclassical control design for the magnetic levitation system, creates a closed-loop system with an overall transfer function which is equivalent to the desired ITAE optimal transfer function. The closed-loop step response for the system is shown in Figure 4-9.

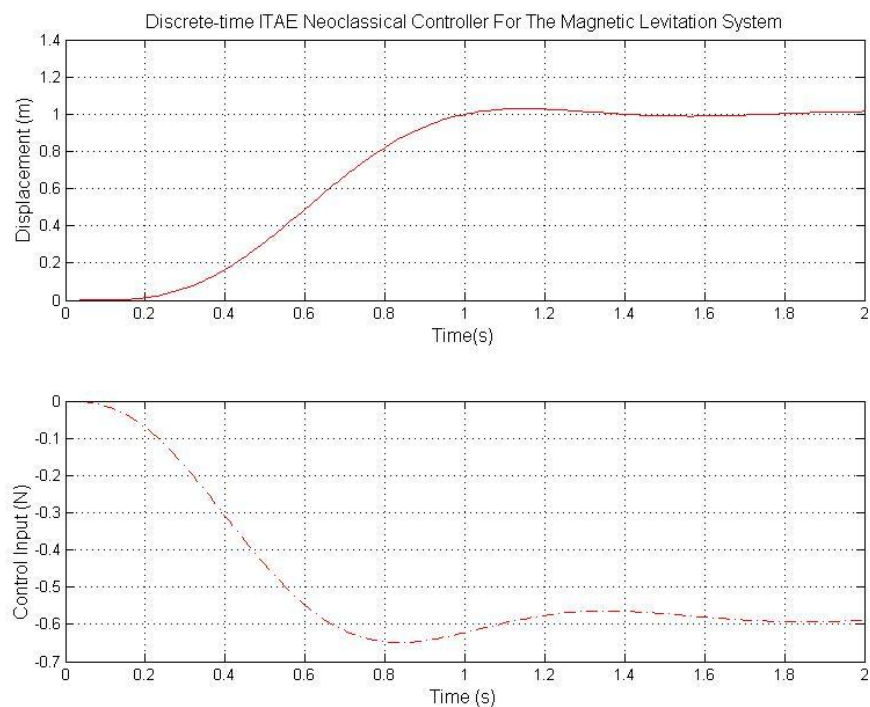


Figure 4- 9: The (a) step response and (b) control input of the neoclassical ITAE controller for the Magnetic Levitation system

As with the mass-spring-damper system, the step response shows a settling time of 1 second and a steady state value of 1 m. These are the design criteria that were originally set for the simulations. The magnetic levitation system is more complicated than the mass-spring-damper system; it is not surprising that the control input curve is more dynamic than the curve seen in the mass spring damper. After all, there are more variables involved that are being adjusted by the control input to derive the desired step response. The steady state value of the control input has a magnitude less than the control gain for the position. This is due to the fact that the steady state for the current, x_3 , is not zero, otherwise there would not be a magnetic field to hold the ball in the air.

The control gains that give the step responses that match the Bessel and Butterworth transfer functions when applied to the discrete-time neoclassical control of the magnetic levitation system are:

$$K_b = [-2.0646 \quad -0.22734 \quad 2.0089] \quad (4.29)$$

$$K_{bw} = [-4.4834 \quad -0.40627 \quad 3.0727] \quad (4.30)$$

The success of neoclassical controller on the magnetic levitation system is further highlighted Figures 4-10 and 4-11, where the Butterworth and Bessel step responses are displayed.

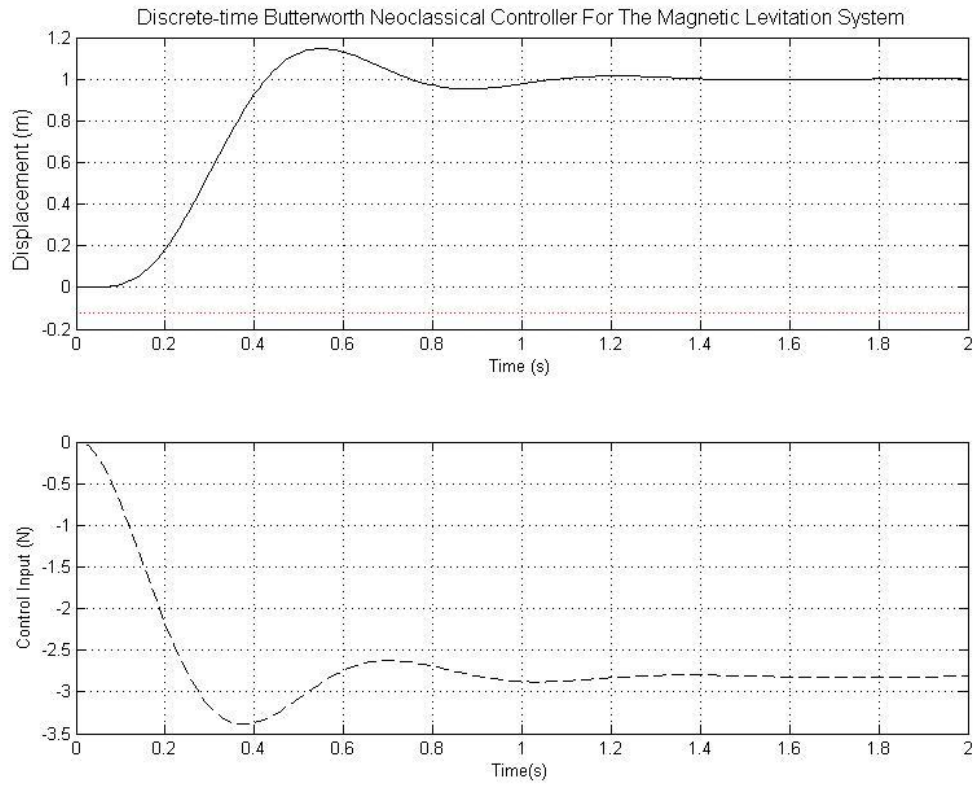


Figure 4- 10: The (a) step response and (b) control input of the neoclassical Butterworth controller for the Magnetic Levitation system

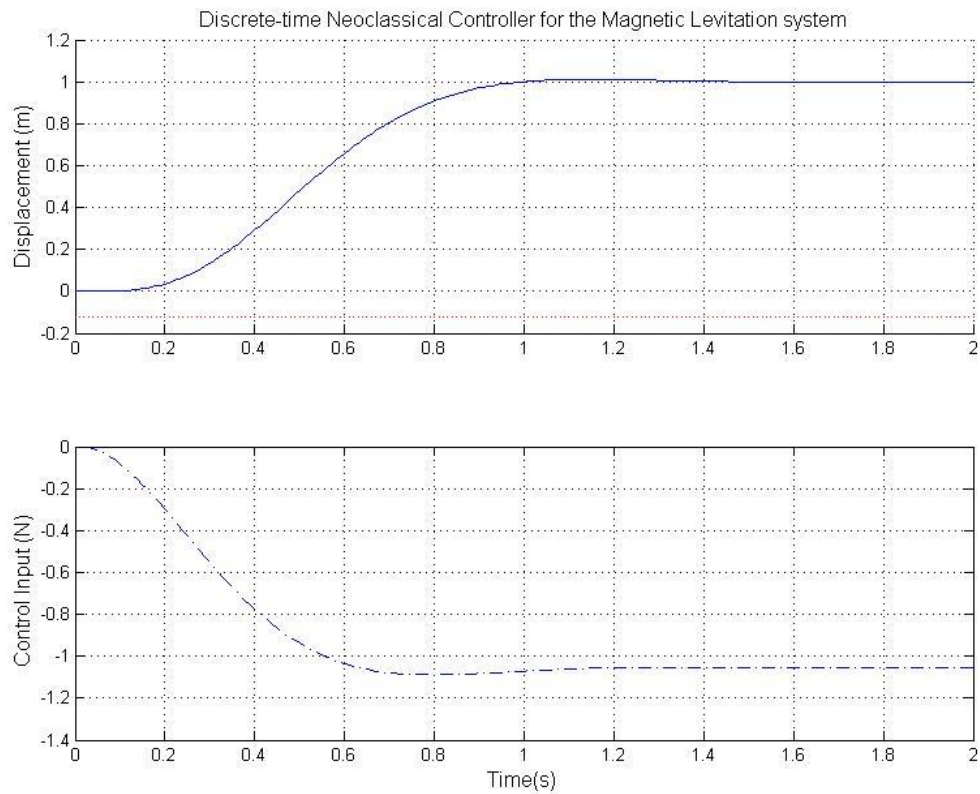


Figure 4- 11: The (a) step response and (b) control input of the neoclassical Bessel controller for the Magnetic Levitation system

The light dotted lines in the position plots are the location of the magnet. . In looking at the control inputs of the performance criterion, it is once again noted that they are very different despite the similarities in step responses, as illustrated in Figure 4-12.

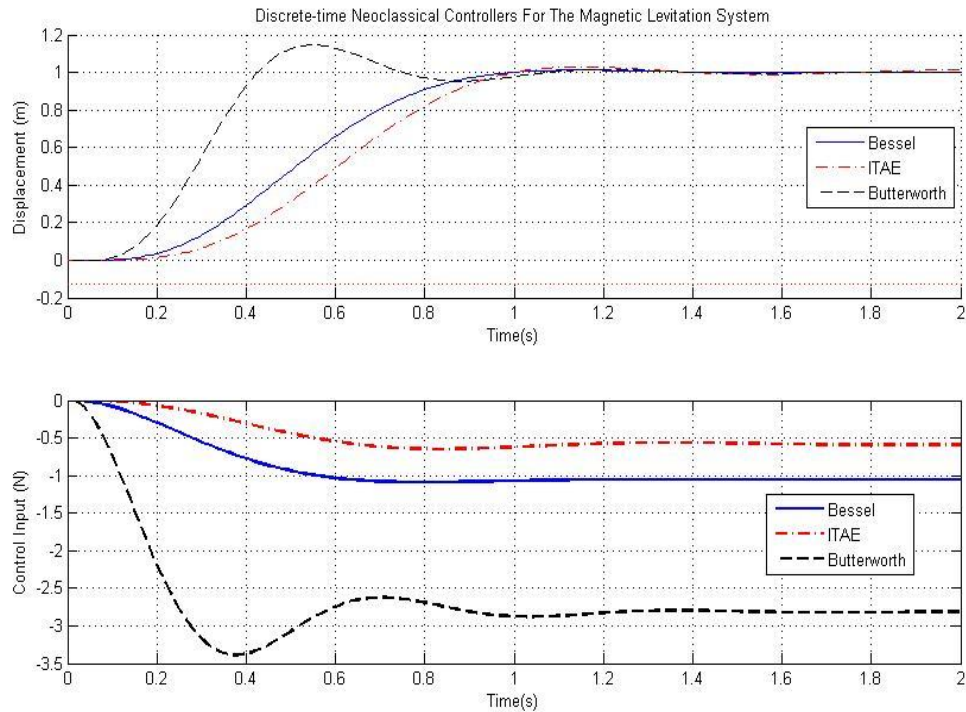


Figure 4- 12: A comparison of the ITAE, Bessel, and Butterworth (a) step responses and (b) control inputs

The differences in control inputs, displayed in Figure 4-12, are due primarily to the control input's effect on the position. Unlike the case of the Mass-Spring-Damper system, the control inputs are negative. The relative magnitudes are different from the Mass-Spring-Damper. This shows that though the standard transfer functions are similar, the amount of energy required to achieve the desired transient response differs from system to system. This is another factor in deciding which standard transfer function to use for a given system. By using neoclassical control, there is flexibility in deciding the most appropriate closed-loop transfer function for a given system.

4.3: The Ball and Beam System

The ball and beam system is unlike either of the previous systems for which the neoclassical controller has been designed. Unlike the mass-spring damper system and the magnetic levitation system, the open-loop ball and beam system has poles at the origin. While systems with at least one pole at the origin does not have steady state error for a step input, the neoclassical control design procedure can still be used to stabilize the system and design the system to have a specific transient step response.

The concept of the ball and beam system is to be able to balance a ball of mass, m , radius, R , and a moment of inertia, J_{ball} , on a beam of length, L , and mass, M , with moment of inertia, J_{beam} at an arbitrary point, d , from the pivot point. The pivot point of the beam is at its center, as shown in Figure 4-13.

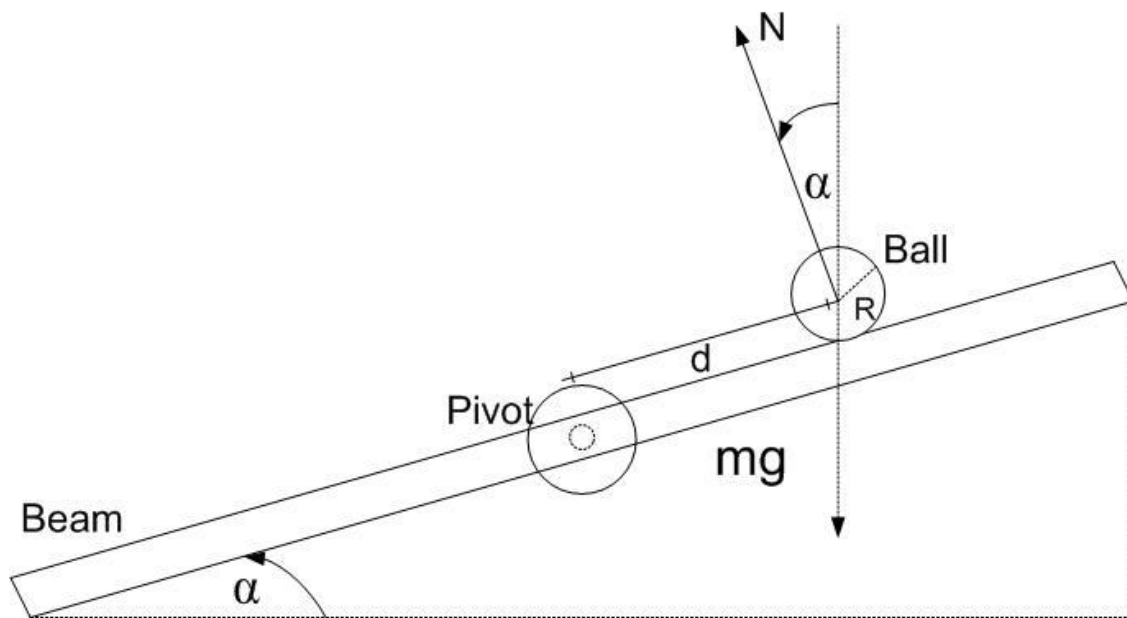


Figure 4- 13: A diagram of the Ball and Beam system dynamics

Figure 4-13 illustrates the dynamics of the ball and beam system. The angle of the beam is α , the gravitational constant is g , N is the normal Force, and d represents the distance of the ball from the pivot point. The forces acting on the ball and the beam can be described by the following differential equations (10):

$$mg\sin(\alpha) + \left(\frac{J_{ball}}{R^2} + m\right)\ddot{d} = mr\dot{\alpha}^2 \quad (4.31)$$

To linearize this system, α is restricted to values close to zero so that the term on the right side of the equation reduces to zero and $\sin(\alpha)$ may be approximated as α . When x_1 represents the position of the ball and x_2 represents the ball's velocity, the state equations are:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{mg}{\left(\frac{J_{ball}}{R^2} + m\right)} \end{bmatrix} \alpha. \quad (4.32)$$

However, for the example of this thesis, the input will not be the angle of the beam, α , but rather the angular acceleration of the beam due to an external actuator or motor. By making x_3 the angle of the beam and the x_4 the angular velocity, the system representation becomes:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{\left(\frac{J_{ball}}{R^2} + m\right)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_{beam}} \end{bmatrix} \tau_{ext}, \quad (4.33)$$

where the external torque is applied at the pivot point. The moments of inertia for the ball

and beam are,

$$J_{beam} = \frac{ML^2}{12} \quad (4.34)$$

$$J_{ball} = \frac{2mR^2}{5} \quad (4.35)$$

The output equation for the system is,

$$y = [1 \ 0 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0] \tau_{ext} \quad (4.36)$$

where the position of the ball is the measured output.

The parameters are assigned the values in Table 4-3 in order to simulate the system.

m	g	R	J_{ball}	J_{beam}
(g)	(m/sec)	(cm)	(kg·m ²)	(kg·m ²)
0.3	9.81	0.1	143	0.0125

Table 4- 3: Table of parameters to be used for the Ball and Beam simulations

Again when the ZOH was used on the Ball and Beam to discretize the system, the step response showed unstable behavior. The cause of the instability was once again identified to be the pre-compensator. An analysis of the transfer function of the plant has been done.

The continuous-time transfer function for the ball and beam system is:

$$G_{sys}(s) = \frac{-0.8541}{s^4}. \quad (4.37)$$

For a sampling interval, T, of 0.01 seconds, the ZOH equivalent discrete-time model of the ball and beam system is:

$$G_{sys}(z) = \frac{-3.5586e-010(z+9.899)(z+1)(z+0.101)}{(z-1)^4}. \quad (4.38)$$

The equivalent Euler Approximation for the discrete-time transfer function model of the ball and beam system,

$$G_{sys}(z) = \frac{-8.5407e-009}{(z-1)^4}. \quad (4.39)$$

This analysis shows that once again, the ZOH equivalent model of the ball and beam system contains a zero that is outside the unit circle. This result, combined with the instability in the implementation of the neoclassical controller on the magnetic levitation's ZOH equivalent model, implies that the unstable nature of the continuous-time model is mapped to the numerator of the ZOH equivalent system. The Euler Approximation does not show this behavior; so it is once again used for discretization. The Euler Approximation of the discrete-time ball and beam system is:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.01 & 0 & 0 \\ 0 & 1 & -5.01e-6 & 0 \\ 0 & 0 & 1 & 0.01 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 17.036 \end{bmatrix} \tau_{ext} \quad (4.40)$$

The step response resulting from this state state-space model is shown in Figure 4-14.

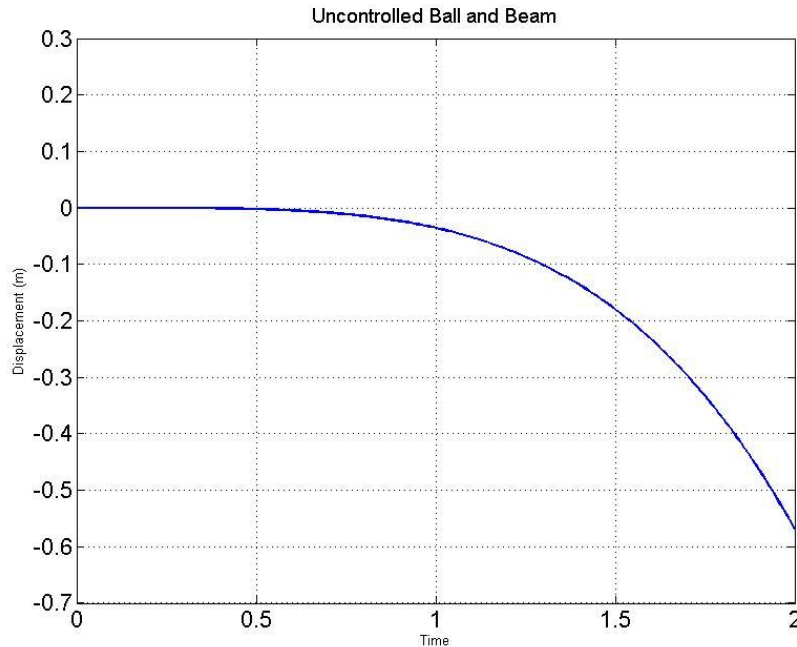


Figure 4- 14: The unit step response of the open-loop Ball and Beam system.

From Figure 4-14, we see that the uncontrolled system is unstable when subjected to a unit step input. The graph makes sense from a physical standpoint; the graph shows the ball rolling down the beam. Depending on the length of the beam, the ball will eventually roll off the beam. The graph does not show the ball falling off the beam due to the system model being an approximation due to the linearization of the system. In order to stabilize this system, the neoclassical control method will be used.

As with the previous two examples, it is important to first identify the desired step response. After applying the neoclassical method, the eigenvalues of the inner state-feedback loop is found. The gains needed to place the system poles at the desired locations to get a closed-loop transfer function equivalent for each of the optimal transfer function are:

$$K_i = [-7818.1 \quad -1899.8 \quad 0.13007 \quad 0.010941], \quad (4.41)$$

$$K_b = [-9397.7 \quad -2446.6 \quad 0.17963 \quad 0.015031], \quad (4.42)$$

$$K_{bw} = [-52268 \quad -7803.6 \quad 0.36099 \quad 0.020586]. \quad (4.43)$$

The closed-loop transfer functions are equivalent to the desired transfer functions. This fact is confirmed when examining the closed-loop step response for the transfer function designed to match the ITAE transfer function in Figure 4-15.

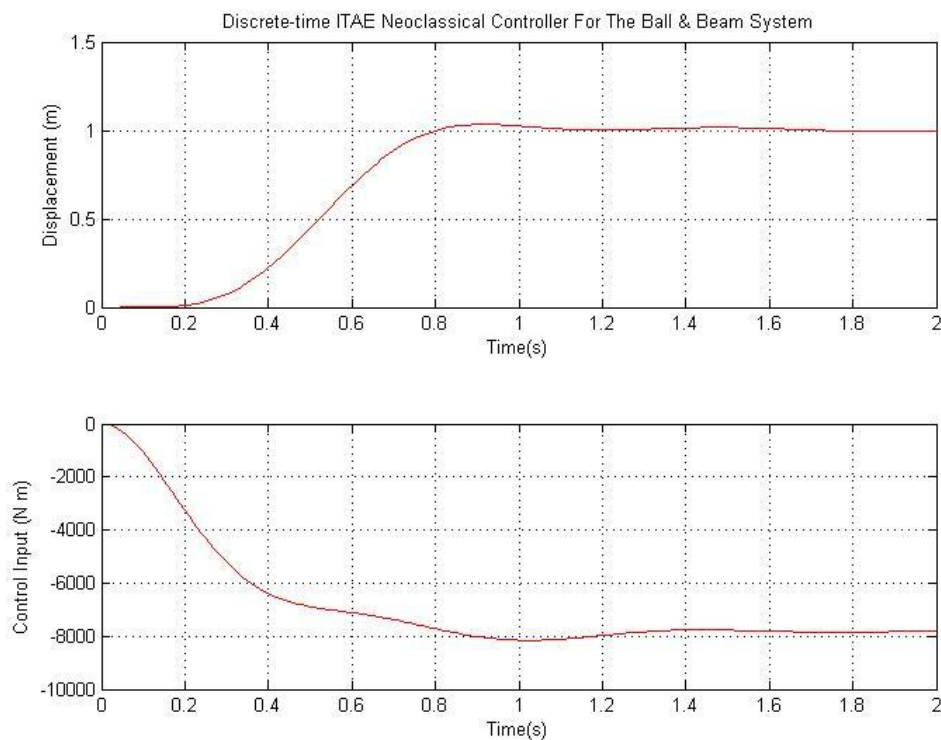


Figure 4- 15: The (a) step response and (b) control input of the neoclassical ITAE controller for the Ball and Beam system

Once again, the step response has the desired characteristics of settling time and steady state value. The control input for the ball and beam system has a very high

magnitude implying that there is a large amount of control action needed in order to stabilize the system and control it to the desired position with a particular transient response specified by the ITAE criterion. These results are also seen in step responses for the neoclassical controllers designed to match the Butterworth and Bessel transfer functions, as shown in Figure 4-16 and Figure 4-17 respectively.

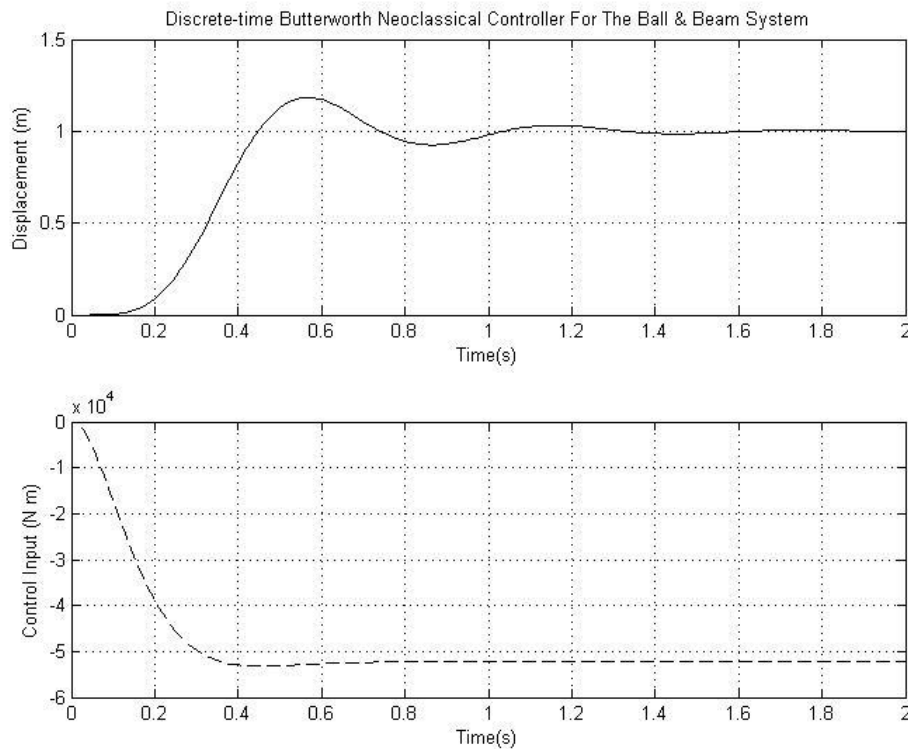


Figure 4-16: The (a) step response and (b) control input of the neoclassical Butterworth controller for the Ball and Beam system

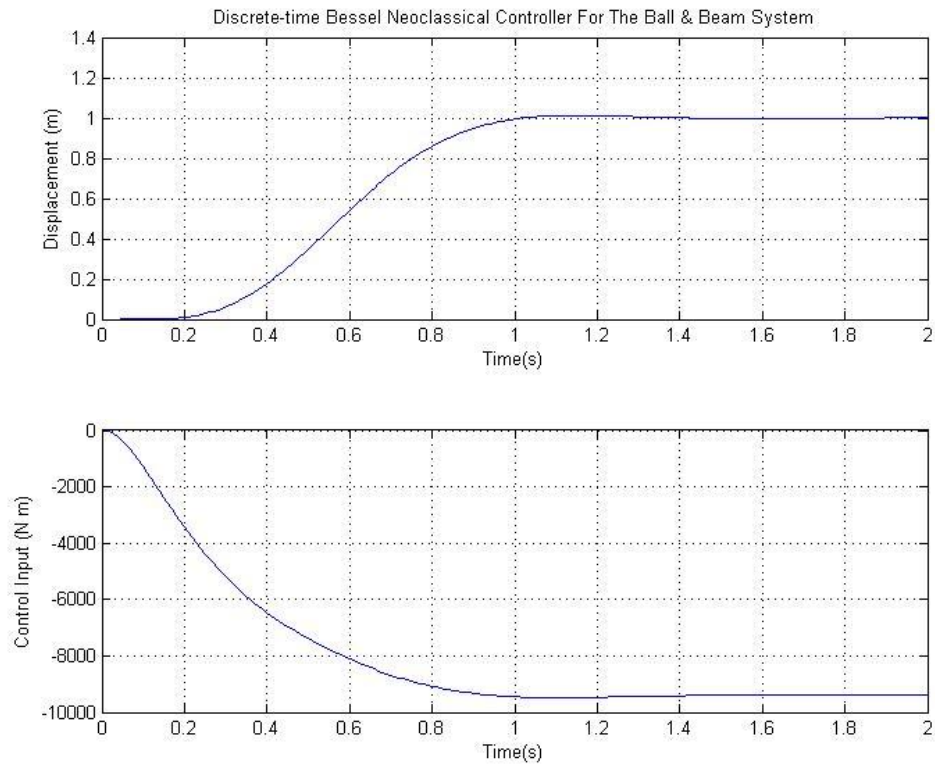


Figure 4- 17: The (a) step response and (b) control input of the neoclassical Bessel controller for the Ball and Beam system

The desired design criteria for the Ball and Beam neoclassical controller have been met. The desired settling time of 1 second has been achieved and the steady-state value is equal to the magnitude of the reference unit step input. However, the observed differences in the control input are once again evident.

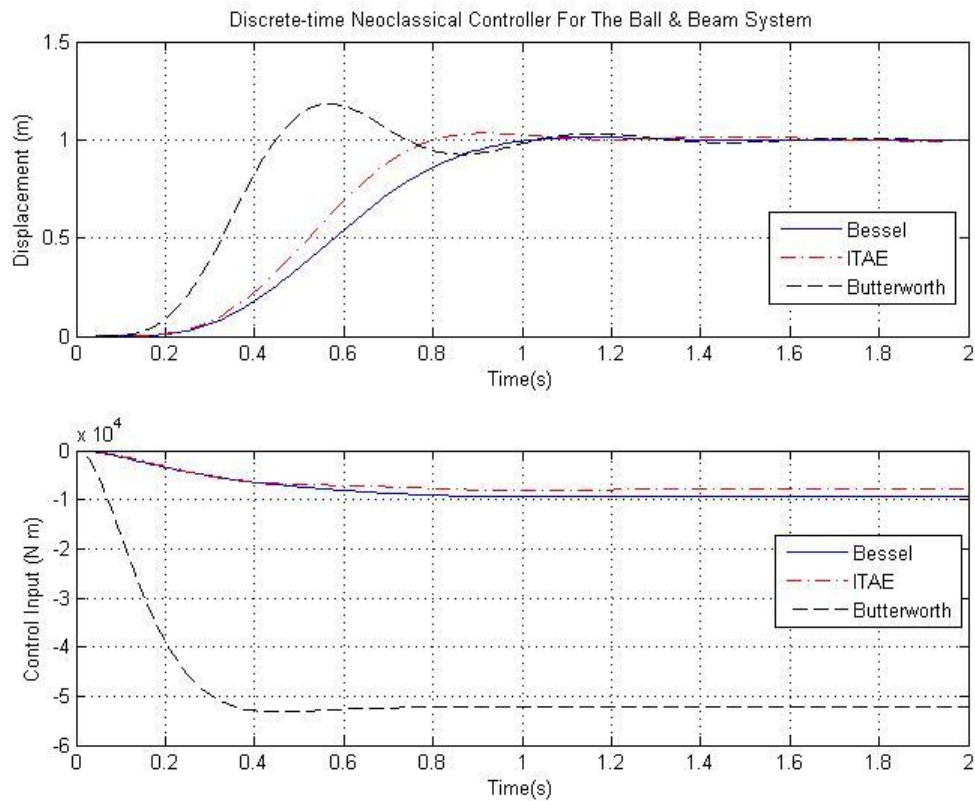


Figure 4- 18: A comparison of the ITAE, Bessel, and Butterworth (a) step responses and (b) control inputs for the Ball and Beam system

Figure 4-18 shows a co-plot of the step responses and control inputs of all three neoclassical controllers for the ball and beam system. The Butterworth neoclassical controller has a higher magnitude of control than the ITAE and Bessel neoclassical controllers. An expanded view of the control inputs is shown in Figure 4-18c to show the difference in both transient response and steady state value.

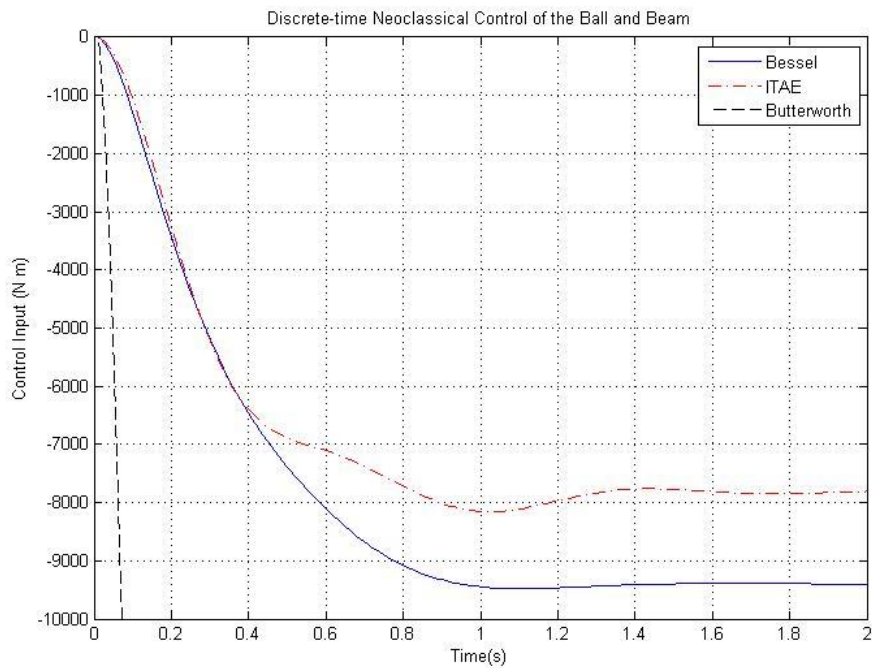


Figure 4-18c: An expanded view of the control inputs.

In implementing the neoclassical controller on the ball and beam system, the Butterworth might not be the most efficient performance criteria to use. The ITAE transfer function or Bessel transfer function would make more efficient choices for implementation on the ball and beam system. However, the neoclassical control has proven capable of stabilizing and controlling a 4th order system and can be used for higher order systems as well.

4.4: Conclusion

The discrete-time neoclassical control design has been shown to work for three systems with different open-loop properties. The step response closely matches the step response for the desired standard transfer function. The desired settling time for the examples was achieved. However, the controller has been built on the assumption that all of the states were known. In the next chapter, we will explore the effects of adding an observer to the system during the design process. We will also analyze the effect of the observer on the step response of the systems.

Chapter 5: Case Studies with Observers

The discrete-time neoclassical controller design has been successfully implemented for cases where all state information is known. In real life, there are frequently cases where state information is not available due to an insufficient number of sensors to measure every state variable. For cases where systems with an unknown state need to be controlled with incomplete state information, an observer needs to be integrated into the controller design. Once again, the mass-spring-damper, magnetic levitation, and ball and beam systems will be used to demonstrate the application of the neoclassical design model with observers incorporated in the design. The ITAE, Bessel, and Butterworth transfer functions will be used as the desired response not only for the controller gains, but also for the observer gains.

5.1: The Mass-Spring-Damper System

The mass-spring-damper system is a stable, 2nd order system. This system is controllable and observable.

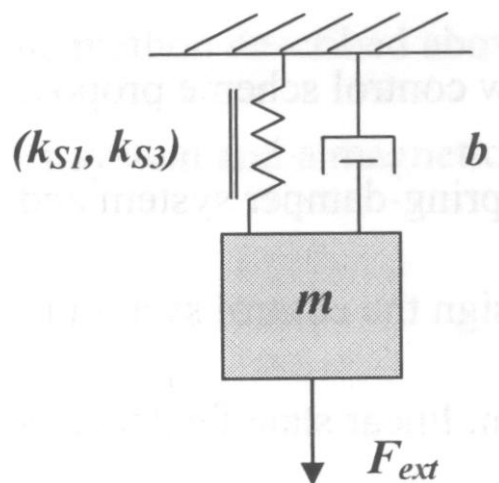


Figure 5- 1: A diagram of the Mass-Spring-Damper system (1)

In chapter 4, the state-space model for the linear Mass-Spring-Damper system was derived.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_{s1}}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} F_{ext} \quad (5.1a)$$

$$y = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (5.1b)$$

Using the same values in Table 4-1 and the ZOH, this system can be discretized using a sample period, $T=0.01s$, to give the discrete-time system matrices:

$$\begin{aligned} \Phi &= \begin{bmatrix} 0.9996 & 0.0099 \\ -0.0887 & 0.97 \end{bmatrix} \\ \Gamma &= \begin{bmatrix} 0 \\ 0.0099 \end{bmatrix} \\ C &= [1 \quad 0] \\ D &= 0 \end{aligned} \quad (5.2)$$

The procedure for adding an observer to a state space control system is well established in modern control theory. The observer gains for the Mass-Spring-Damper system will be based on the 2nd order standard transfer function of choice to calculate the pole locations. The vector matrix form of the state feedback loop is,

$$\begin{pmatrix} x_{k+1} \\ e_{k+1} \end{pmatrix} = \begin{bmatrix} \Phi - \Gamma K & \Gamma K \\ 0 & \Phi - LC \end{bmatrix} \begin{bmatrix} x_k \\ e_k \end{bmatrix} \quad (5.3)$$

The only unknowns in this equation are the controller gain, K , and the observer gain, L . Once the pole locations are scaled to the desired observer settling time, the observer gains, L , will be calculated using the Φ and C matrices. The Ackermann formula

or MATLAB's Place command can be used to find the gains for both K and L. Table 5.1 shows the controller gains and the observer gains for each standard transfer function for the desired settling times.

Observer	Controller feedback gains K	Observer gains L
Settling time	1 sec	0.1 sec
ITAE	[109.1629 10.0859]	[0.79914 23.299]
Bessel	[53.0106 9.1234]	[0.6723 12.8251]
Butterworth	[76.1516 9.9991]	[0.7726 21.2409]

Table 5- 1: Controller gains for $T_s=1$ sec and Observer gains for $T_s=0.1$ sec. for the Mass-Spring-Damper system

These gains are used in the vector-matrix form of G_{sf} , Equation 5.3. Then to complete the controller design, all that needs to be done is to place the integrator and pre-compensator in series. Then after closing the loop with unity gain output feedback, the resulting closed-loop transfer function will match the desired standard transfer function.

Figure 5-2 shows that there is a difference in the step response between the desired ITAE step response (green) and the actual step response with an initial estimate error of 0.25m.

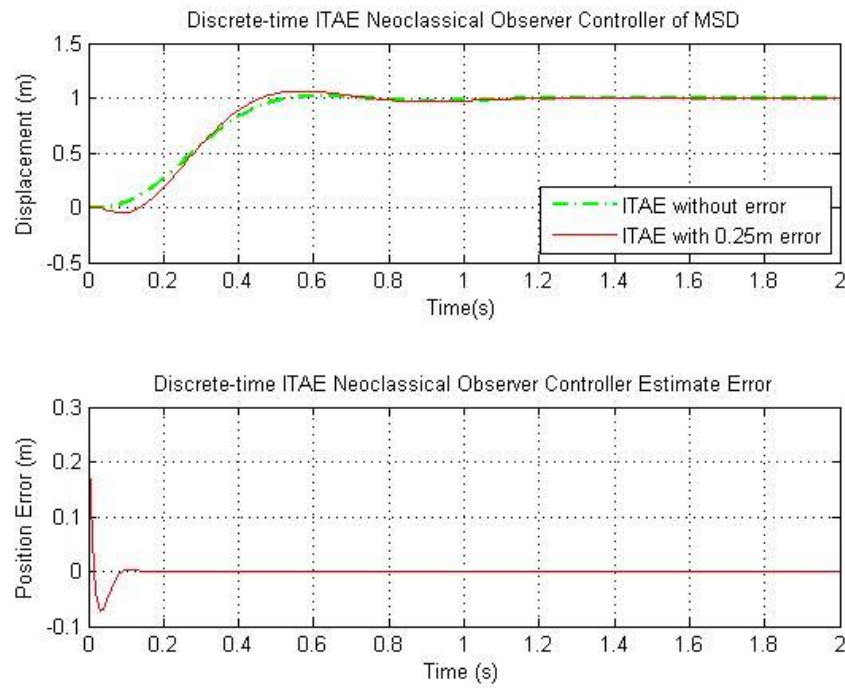


Figure 5- 2: Discrete-time ITAE Neoclassical controller (a) step response and(b) state estimate error for the Mass-Spring-Damper system

Figure 5-2 has a step response close to the desired curve. Furthermore, the error in the state estimate goes to zero in 0.1 seconds, as expected. These results are similar to the results of the neoclassical control with observer for the Bessel and Butterworth responses shown in Figures 5-3 and 5-4 respectively.

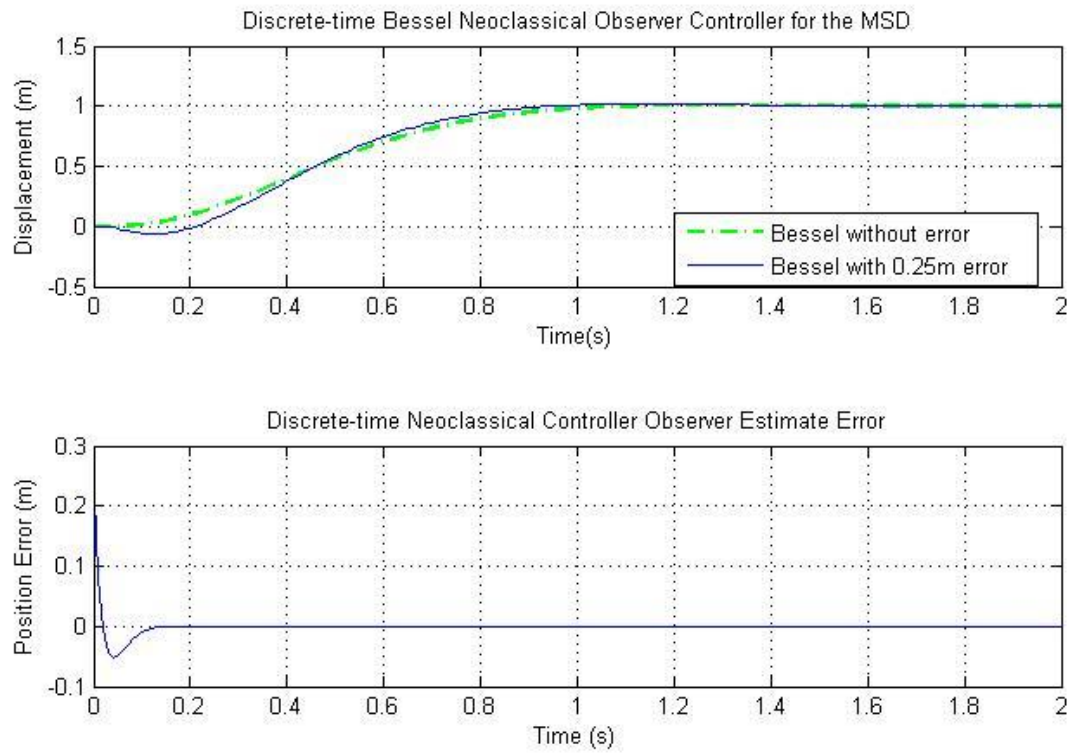


Figure 5- 3: Discrete-time Bessel Neoclassical controller (a) step response and (b) state estimate error for the Mass-Spring-Damper system

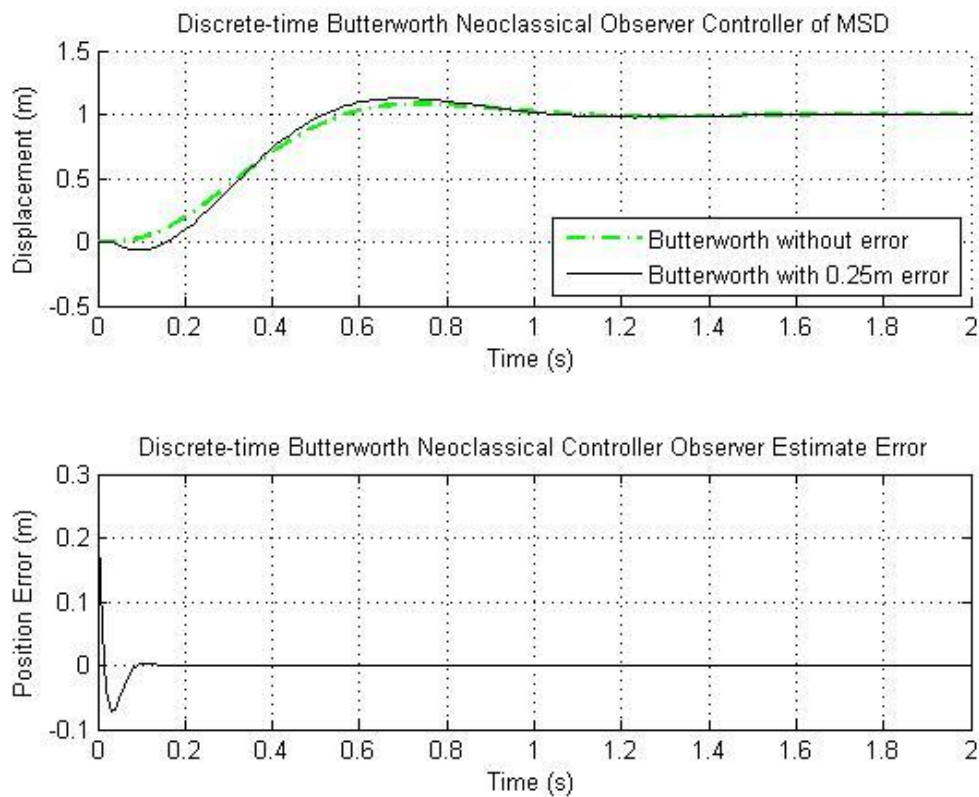


Figure 5- 4: Discrete-time Butterworth Neoclassical controller (a) step response and (b) state estimate error for the Mass-Spring-Damper system

In all three simulations, the error goes to zero in 0.1 seconds. The effect of the estimation error on the system remains long after the error goes to zero. However, differences do diminish with time and the settling time is close to the desired 1 second.

5.2: Magnetic Levitation system

The addition of the observer to the neoclassical controller has been successful when applied to a stable 2nd order system. The magnetic levitation is a more complex system with more state variables to estimate. The magnetic levitation system is both controllable and observable. Therefore, the magnetic levitation is also a good system to test the neoclassical controller design with the observer.

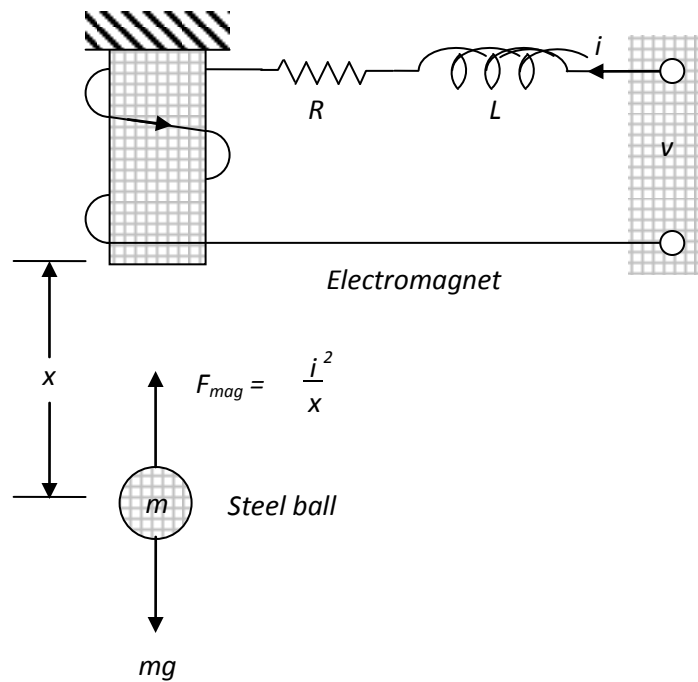


Figure 5- 5: Diagram of the Magnetic Levitation system

The linearized state-space model for the continuous-time magnetic levitation system,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g}{x_{1,0}} & 0 & -2\sqrt{\frac{g}{mx_{1,0}}} \\ 0 & 0 & -R/L \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix} v \quad (5.4a)$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (5.4b)$$

The parameters for simulation are found in Table-4-2. The sampling interval, T , is 0.01. The Euler discretization is used to give the discrete-time state-space representation of the magnetic levitation system:

$$x[k + 1] = \begin{bmatrix} 1 & 0.01 & 0 \\ 0.7848 & 1 & -1.447 \\ 0 & 0 & 0.957 \end{bmatrix} x[k] + \begin{bmatrix} 0 \\ 0 \\ 0.06993 \end{bmatrix} u[k] \quad (5.5a)$$

$$y[k] = [1 \ 0 \ 0]x[k] \quad (5.5b)$$

The vector matrix form of the state update equation and error update equation is,

$$\begin{pmatrix} x_{k+1} \\ e_{k+1} \end{pmatrix} = \begin{bmatrix} \Phi - \Gamma K & \Gamma K \\ 0 & \Phi - LC \end{bmatrix} \begin{bmatrix} x_k \\ e_k \end{bmatrix}. \quad (5.6)$$

Using the same methods to find the controller gains and the observer gains that were used in the previous example will yield a state-space representation for the $G_{sf}(z)$ block in Figure 3.2.

Table 5-2 shows the controller gains and the observer gains for each standard transfer function for the desired settling times.

Observer	Controller feedback gains K	Observer gains L
Settling time	1 sec	0.1 sec
ITAE	[-0.9815 -0.1461 0.7411]	[1.4351 90.6723 -12.2053]
Bessel	[-2.0646 -0.2273 2.0089]	[1.1011 45.8455 -4.1734]
Butterworth	[-4.4834 -0.4063 3.0727]	[1.2380 65.1320 -8.5042]

Table 5- 2: Controller gains for $T_{sc}=1$ sec. and Observer gains for $T_{so}=0.1$ sec. for the Magnetic Levitation system

After the discrete-time integrator and the precompensator are put in series with $G_{sf}(z)$ and the loop is closed with unity gain output feedback, the resulting closed-loop transfer function is equivalent to the chosen standard transfer function. However, there are small differences in the step response when there is an error in the initial estimate. Figure 5-6 through 5-8 shows the step response and observer state estimate error for the neoclassical controller with an observer with an initial estimate error of 0.075m for the ITAE, Bessel, and Butterworth transfer functions, respectively.

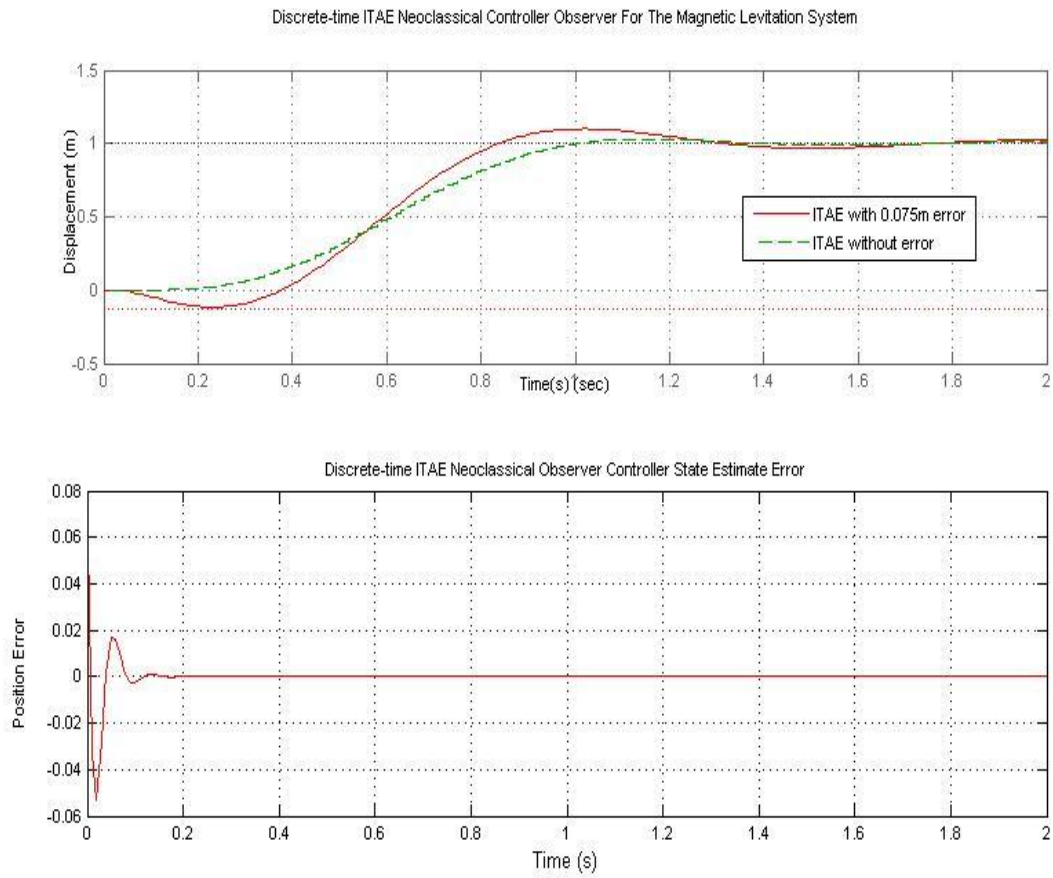


Figure 5- 6: Discrete-time ITAE Neoclassical controller (a) step response and (b) state estimate error for the Magnetic Levitation system

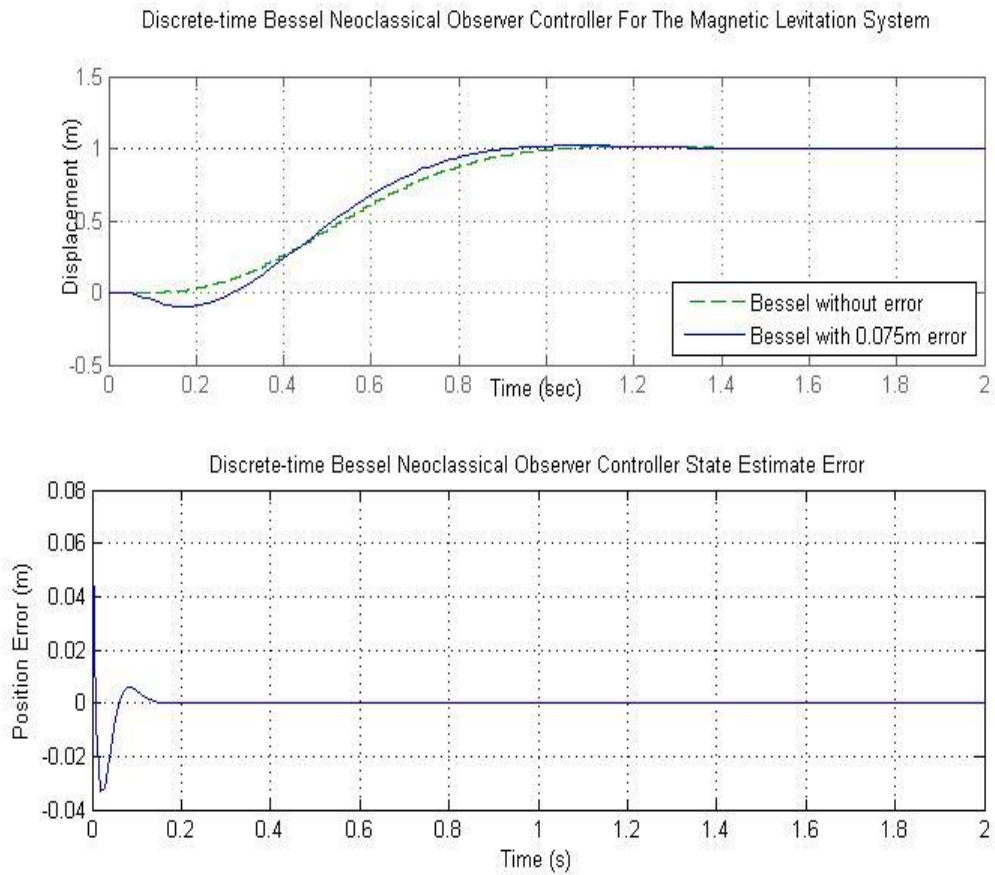


Figure 5- 7: Discrete-time Bessel Neoclassical controller (a) step response and (b) state estimate error for the Magnetic Levitation system

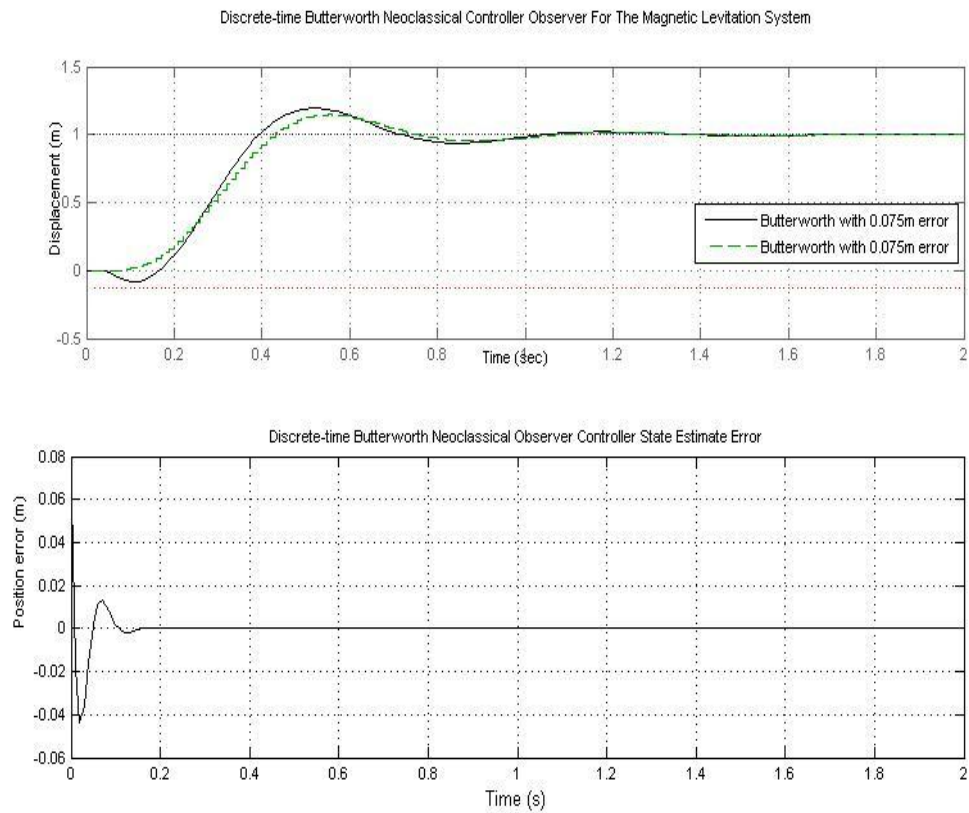


Figure 5- 8: Discrete-time Butterworth Neoclassical controller (a) step response and (b) state estimate error for the Magnetic Levitation system

In all three simulations, the error goes to zero in 0.1 seconds. But the effect of the estimation error on the system remains long after the error goes to zero. But the effect does diminish with time and the settling time is close to the desired 1 second. Due to the small error in the estimate, the systems come close to hitting the bottom of the magnet (straight dotted line). The constraint of the location of the magnet adds to the design process for the controller an extra layer of difficulty. By using the discrete-time neoclassical controller with observer, the process does not take long and even with errors in the estimate, the actual response still comes close to matching the desired step response.

5.3: The Ball and Beam System

The ball and beam system is a 4th order, observable, and controllable system.

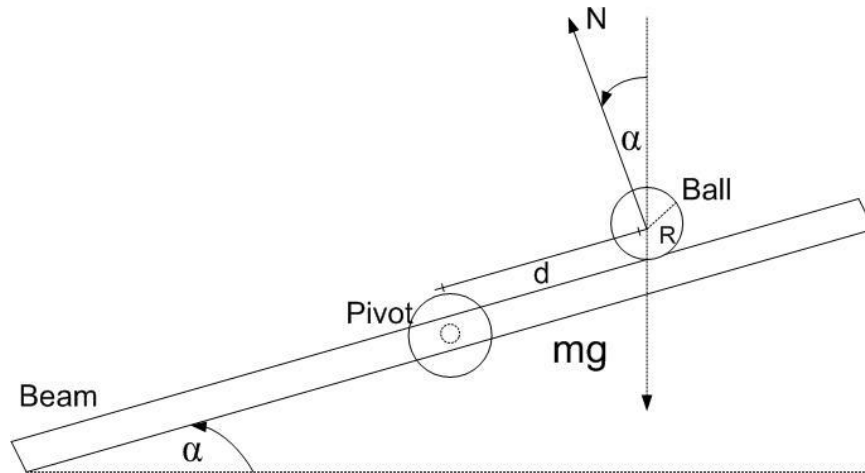


Figure 5- 9: Diagram of the Ball and Beam system

The state-space model of the ball and beam is,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-mg}{(\frac{J_{ball}}{R^2} + m)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_{beam}} \end{bmatrix} \tau_{ext} \quad (5.7a)$$

$$y = [1 \ 0 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0] \tau_{ext} \quad (5.7b)$$

Using the values from table 4-3, the system is discretized using the Euler Approximation

method.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.01 & 0 & 0 \\ 0 & 1 & -5.01e-6 & 0 \\ 0 & 0 & 1 & 0.01 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 17.036 \end{bmatrix} \tau_{ext} \quad (5.8)$$

As with the previous two examples, it is important to first identify the desired step response. Using the standard transfer functions as a basis, the controller gains and the observer gains are calculated and substituted into the observer's update equations,

$$\begin{pmatrix} x_{k+1} \\ e_{k+1} \end{pmatrix} = \begin{bmatrix} \Phi - \Gamma K & \Gamma K \\ 0 & \Phi - LC \end{bmatrix} \begin{bmatrix} x_k \\ e_k \end{bmatrix} \quad (5.9)$$

Table 5-3 shows the controller gains and the observer gains for each standard transfer function for the desired settling times.

Observer	Controller feedback gains K	Observer gains L
Settling time	1 sec	0.1 sec
ITAE	[-7.8181 -1.8998 0.0001 0.0000] $\times 10^3$	[0.0000 0.0000 -0.2441 -4.7390] $\times 10^8$
Bessel	[-9.3977 -2.4466 0.0002 0.0000] $\times 10^3$	[0.0000 0.0000 -0.3466 -7.0167] $\times 10^8$
Butterworth	[-5.2268 -0.7804 0.0000 0.0000] $\times 10^4$	[0.0000 0.0000 -0.0865 -2.2971] $\times 10^9$

Table 5- 3: Controller gains for $T_{sc}=1$ sec. and Observer gains for $T_{so}=0.1$ sec. for the Ball and Beam system

The integrator, pre-compensator, and unity gain output feedback are added in the same manner as previous systems. Figures 5-6 through 5-8 show the step response and observer state estimate error for the neoclassical controller with an observer with an initial estimate error of 0.2 m for the ITAE, Bessel, and Butterworth transfer functions, respectively.

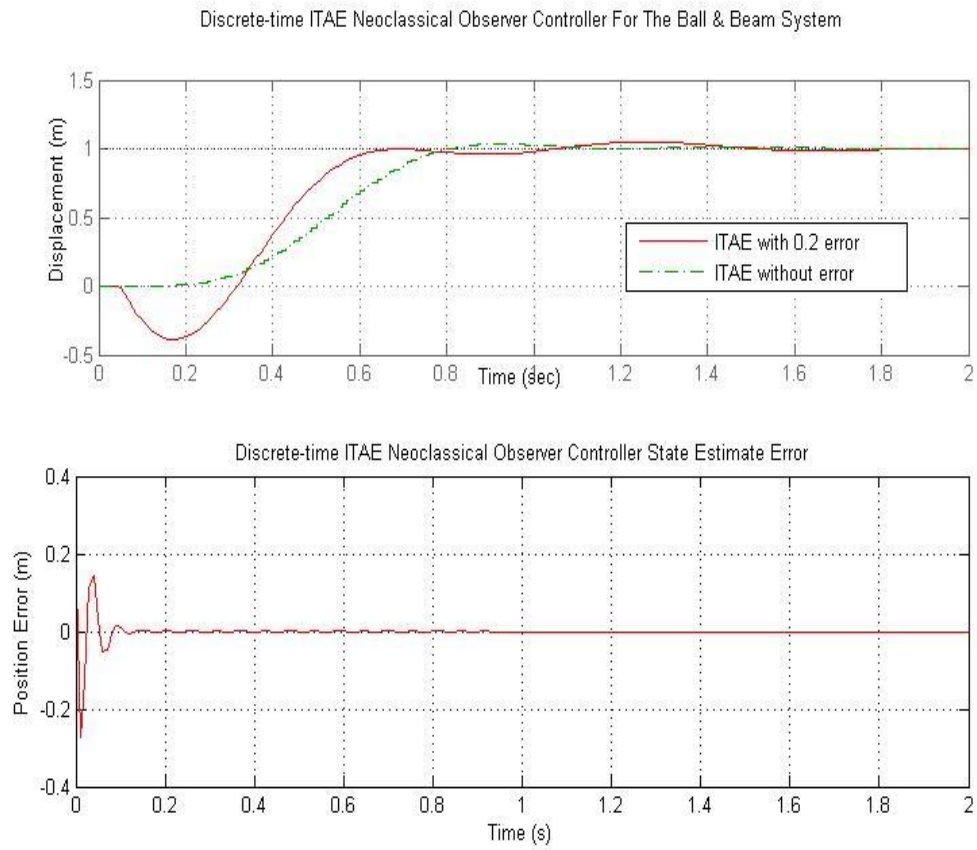


Figure 5- 10: Discrete-time ITAE Neoclassical controller (a) step response and (b) state estimate error for the Ball and Beam system

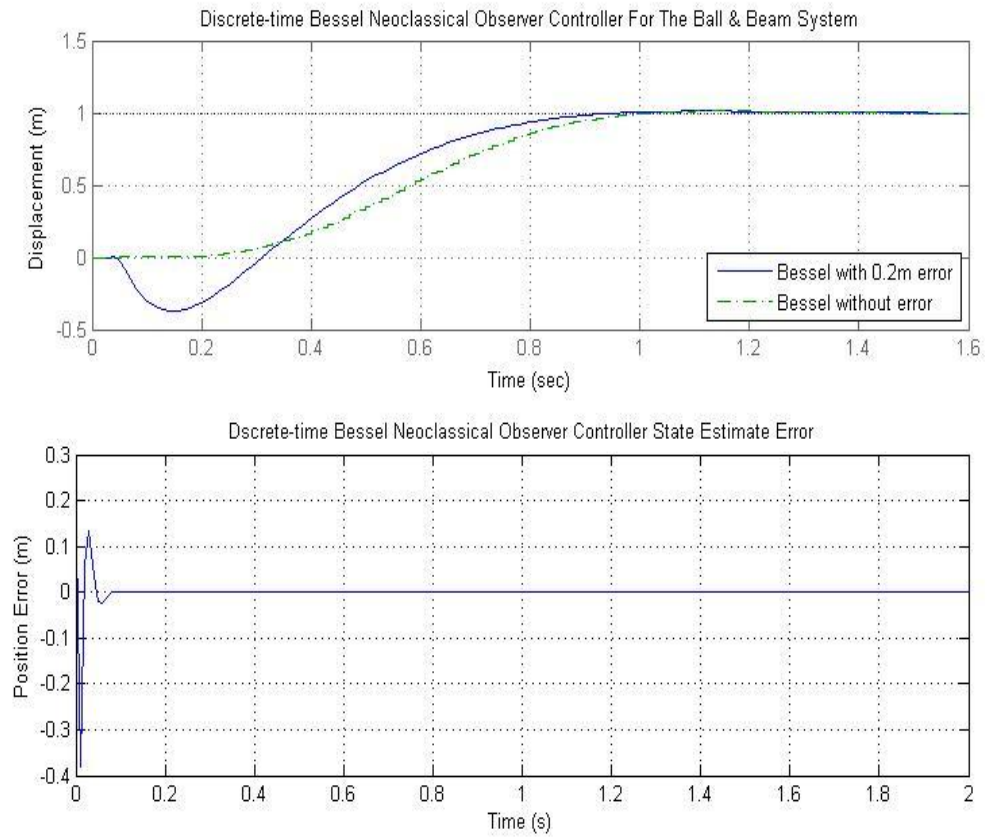


Figure 5- 11: Discrete-time Bessel Neoclassical controller (a) step response and (b) state estimate error for the Ball and Beam system

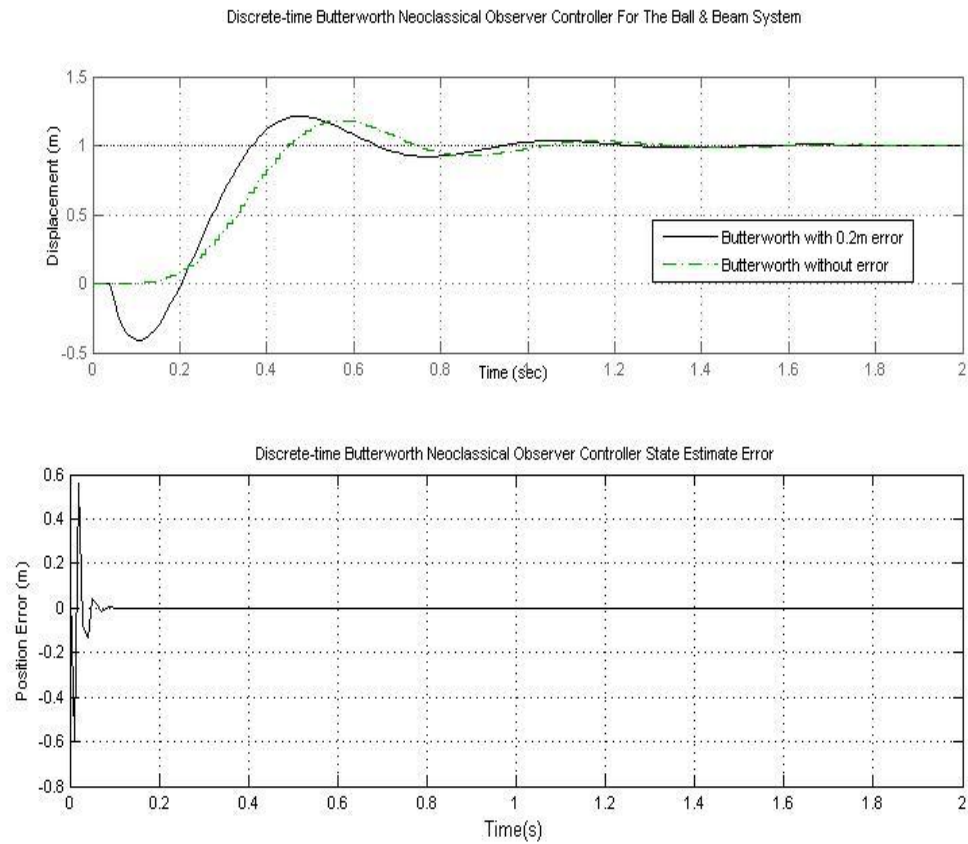


Figure 5- 12: Discrete-time Butterworth Neoclassical controller (a) step response and (b) state estimate error for the Ball and Beam system

In all three simulations, the error goes to zero in 0.1 seconds. But the effect of the estimation error on the system remains long after the error goes to zero. But the effect on the step response does diminish with time and the settling time is close to the desired 1 second.

5.4: Conclusion

The discrete-time neoclassical control design has been shown to work for three systems with different open-loop properties. The implementation of the observer into the neoclassical control design has allowed state estimates to be used when the plant's state is unknown. The error between the estimate of the state and the actual state is driven to zero in the specified observer settling time. The step response closely matches the step response for the desired standard transfer function. The settling time set by the controller is approximately 1 second in all simulations, as per design parameter. In conclusion, observers have been successfully incorporated into the discrete-time neoclassical control technique.

Chapter 6: Conclusion and Future Work

This thesis has reviewed some previous work in neoclassical control. The original work on neoclassical control was done in continuous time. This thesis has defined the limits of continuous-time neoclassical control and has introduced two new standard optimal transfer functions, the Bessel and Butterworth to be used in design. These standard transfer functions, along with the ITAE have been explained and shown to have differing properties that could be scaled to within the desired response time specification.

This thesis has taken the previous neoclassical control work and converted it to discrete-time. The pieces of the neoclassical block diagram that have significant changes in discrete-time have been examined. Two types of discretization have been discussed, the ZOH and the Euler Approximation. The likelihood of zero-pole cancellation issues due to the numerator of the plant has been discussed thoroughly. The use of the Euler Approximation as a method to bypass the zero-pole cancellation issue has also been shown. The benefits and weakness of both techniques have been discussed. The discrete-time neoclassical control theory has also incorporated an observer to estimate states that are not known or measurable.

The discrete-time neoclassical controller has been tested without an observer on the Mass-Spring-Damper system, the Magnetic Levitation system, and the Ball and Beam system. The first two systems had been used in previous work for continuous-time neoclassical control. The results of the discrete-time neoclassical control on the three systems show that the desired settling time is achieved and the steady state value is set to the reference input. Furthermore, all time response simulations for the discrete-time

neoclassical controllers designed match the step response of the specified desired transfer functions.

A full-order observer has been incorporated into the discrete-time neoclassical design and simulated. The results show that with small errors in the initial guess, the response closely matches the desired transfer functions, including having a settling time close to the desired settling time. The successful implementation of the observer into the neoclassical model is a large step towards real world applications of neoclassical control.

6.1: Future Work

Future expansion and exploration of the neoclassical control method should include systems with multiple inputs and multiple outputs (MIMO). Another important expansion of neoclassical control is to further investigate its application to systems that already have 1 or more poles at the origin in the s -domain (type-1 or higher type systems). It might be possible to use the neoclassical control technique without adding the additional integrator through partial state-feedback. Neoclassical control needs to be tested for systems subjected to different types of inputs such as ramp, parabolic and even sinusoidal. To adapt neoclassical control for these inputs, the standard transfer functions chosen will need to have time responses which are optimized for the chosen input. The introduction of zeros outside of the unit circle must be investigated more formally in the future as well. Finally, the introduction of reduced-order observers should be investigated.

Bibliography

1. **Holterman, Kathryn J.** *A Neoclassical Approach: More Efficient Controller Design*. Milwaukee, WI : Marquette University, 2008.
2. **Dorf, Richard C. and Bishop, Robert H.** *Modern Control Systems*. Tenth. Upper Saddle River, NJ : Pearson Education, Inc, 2005.
3. **Vaccaro, Richard J.** *Digital Control: A State-Space Approach*. New York : McGraw-Hill, Inc, 1995.
4. **Stefani, Raymond T., et al.** *Design of Feedback Control Systems*. Fourth. New York : Oxford University Press, 2002.
5. **D. Graham and R.C. Lathrop**, "The Synthesis of "Optimum" transient response: Criteria and Standard Forms, Part 2", Transactions of the American Institute of Electrical Engineering (AIEE) 72, Nov 1953, pp. 273-288.
6. **Williams II, Robert L. and Lawrence, Douglas A.** *Linear State-Space Control Systems*. Hoboken, NJ : John Wiley & Sons, Inc, 2007.
7. **Chitode, J. S.** *Digital Signal Processing*. First. Pune, India : Technical Publications, 2008.
8. **Kuo, Benjamin C.** *Digital Control Systems*. Second. New York : Oxford University Press, 1992.
9. **Kuo, Bengamin C. and Golnaraghi, Farid.** *Automatic Control Systems*. Eighth. New York : John Wiley & Sons, Inc, 2003.
10. **William Messner and Dawn Tilbury.** Control Tutorials for Matlab. <http://www.engin.umich.edu/class/ctms/examples/ball/ball.htm>. [Online].
11. **M. Haeri and M. S. Tavazoei**, *Comparison of the Existing Methods in Determination of the Characteristic Polynomial*. 2005, Proceedings of World Academy of Science, Engineering, and Technology, Vol. 6, 2005, pp. 130-133.

Appendix A: MATLAB Code

Mass-Spring Damper code

```

clear
s1=1; %Controller settling time
s=0.1 ; %Observer settling time
T=0.01; %Sample Time
t=0:T:2*s1; %Time Scale
xi=[0,0,.0,0]; %Initial Conditionv for Observer
u=ones(size(t))/T; %Input (Step for t>0)
int=ss(0,1,1,0); %State Space Integrator in CT
intd=c2d(int,T,'zoh'); %Digital SS integrator

% % % % System % % % %
m=1;k=9; b=3; A=[0,1;-k/m,-b/m];B=[0;1/m]; C=[1 0];D=0;
[P,G,C,D]=c2dm(A,B,C,D,T,'zoh');
[n,d]=ss2tf(P,G,C,D); %Defines the denominator of the Pre-compensator
[yk,xk]=dlsim(P,G,C,D,u);
% figure(10); plot(t,xk)
sysod=ss(A,B,C,D)
% figure(3)
% rlocus(sysod)
% % % % Bessel Poles % % % % % % % %

% Observer(nth order)
poB1=exp((1/s)*(-4.053-2.34*1i)*T); %ZOH equivalent of Bessel
Polynomial
poB2=exp((1/s)*(-4.053+2.34*1i)*T);%ZOH equivalent of Bessel Polynomial

%Bessel Transfer function
q=poly([-4.053-2.34*1i,-4.053+2.34*1i]);
w=tf(q(:,3),q)
stepinfo(w)
% Controller((n+1) order)
dpb=(1/s1)*[-5.0093;-3.9668-3.7845i;-3.9668+3.7845i]*.96;
dtf=poly(dpb); %Denominator of desired transfer function-ct
num=dtf(:,4);den=dtf;Tsb=tf(num,den); %Desired Transfer function
Tsbd=c2d(Tsb,T,'zoh');
Tssbd=ss(Tsbd); %SS Model of Desired Transfer function
[numbd,denbd]=ss2tf(Tssbd.a,Tssbd.b,Tssbd.c,Tssbd.d); %Defines
Numerator of Pre-compensator

% % % % % % ITAE zero position error Pole Polynomials % % % % % % % %

%Observer(n)
wno=5.979827684396857; %natural frequency
ITAEpoly=[1 1.4*wno wno^2];
poi=roots(ITAEpoly)';
poid=exp((T/s)*poi);

%Controller(n+1)
wn=7.54
tfi=[1 1.75*wn 2.15*wn^2 wn^3];
Pid=roots(tfi)/s1;
dtfid=poly(Pid);

```

```

numid=dtfid(:,4);denid=dtfid;
Tsi=tf(numid,denid); %Desired Transfer function
Tsid=c2d(Tsi,T,'zoh');
Tssid=ss(Tsid); %SS Model of Desired Transfer function
[nd,dd]=ss2tf(Tssid.a,Tssid.b,Tssid.c,Tssid.d);

% % % % % % % Butterworth Polynomials % % % % % % % % %

%Observer(n)
BWpoly=[1 1.4142 1];
pBW=roots(BWpoly)';
BWTs=5.9633; %Settling time of normalized butterworth polynomial
pBWd=exp((pBW/(1/BWTs))*T/s); % 2nd order butterworth

%Controller(n+1)
BWpoly3=[1 2 2 1];
pBW3=roots(BWpoly3)';
BWTs3=6.639867870220427;
pBW03d=(pBW3/(1/BWTs3))/s1;
dtfbwd=poly(pBW03d);
numbwd=dtfbwd(:,4);denbwd=dtfbwd;Tsbw=tf(numbwd,denbwd);
Tsbwd=c2d(Tsbw,T,'zoh');%Desired Transfer function
Tssbwd=ss(Tsbwd); %SS Model of Desired Transfer function
[nbwd,dbwd]=ss2tf(Tssbwd.a,Tssbwd.b,Tssbwd.c,Tssbwd.d);

% % % % % % % % % % Neoclassical controller observer % % % % % % % %

% % % % % % % Break Down % % % % % % %

%Bessel
Gsbw=minreal(Tssbw/((1-Tssbw))); %Removes feedback loop
Gssbw=minreal(Gsbw/intd); %Removes integrator
pbw=eig(Gssbw);
Kbw=place(P,G,pbw);

%ITAE
Gsid=minreal(Tssid/(1-Tssid));
Gssid=minreal(Gsid/intd);
pid=eig(Gssid);
Kid=place(P,G,pid);

%Butterworth Function
Gsbwd=minreal(Tssbwd/(1-Tssbwd));
Gssbwd=minreal(Gsbwd/intd);
pbwd=eig(Gssbwd);
Kbwd=place(P,G,pbwd);

% % % Observer Design % % %

L1=place(P',C',[poB1;poB2])';
L2=place(P',C',poid)';
L3=place(P',C',pBWd)';

```

```

Pbt = [P-G*Kbd   G*Kbd; zeros(size(P))  P - L1*C];
Pit = [P-G*Kid   G*Kid; zeros(size(P))  P - L2*C];
Pbwt = [P-G*Kbwd  G*Kbwd; zeros(size(P))  P - L3*C];
Gt = [G ;zeros(size(G))];
Ct = [ C   zeros(size(C))];

[ykcob,xkcob]=dlsim(Pbt,Gt,Ct,D,u,xi);
xcob=[xkcob(:,1) xkcob(:,2)];
ecob=[xkcob(:,3) xkcob(:,4)];
xkhatb=xcob-ecob;

[ykcoi,xkcoi]=dlsim(Pit,Gt,Ct,D,u,xi);
xcoi=[xkcoi(:,1) xkcoi(:,2)];
ecoi=[xkcoi(:,3) xkcoi(:,4)];
xkhati=xcoi-ecoi;

[ykcobw,xkcobw]=dlsim(Pbwt,Gt,Ct,0,u,xi);
xcobw=[xkcobw(:,1) xkcobw(:,2)];
ecobw=[xkcobw(:,3) xkcobw(:,4)];
xkhatbw=xcobw-ecobw;

emaxcb=max(abs(ecob(:,1)))
emaxci=max(abs(ecoi(:,1)))
emaxcbw=max(abs(ecobw(:,1)))

tecb=t*abs(ecob(:,1))
teci=t*abs(ecoi(:,1))
tecbw=t*abs(ecobw(:,1))

%%%%%%%%%Rebuild %%%%%%%%%%

Hssbd=ss(Pbt,Gt,Ct,D,T); %State-feedback Controller w/ Observer
Prebd=tf(numbd,n,T); %Pre-compensator
Hsbd=(Hssbd*minreal(intd*ss(Prebd)));
sysbd=feedback(Hsbd,1/T);

Hssid=ss(Pit,Gt,Ct,D,T); %State-feedback Controller w/ Observer
Preid=tf(nd,n,T); %Pre-compensator
Hsid=Hssid*minreal(intd*ss(Preid));
sysid=feedback(Hsid,1/T);

Hssbwd=ss(Pbwt,Gt,Ct,D,T); %State-feedback Controller w/ Observer
Prebwd=tf(nbwd,n,T); %Pre-compensator
Hsbwd=Hssbwd*minreal(intd*ss(Prebwd));
sysbwd=feedback(Hsbwd,1/T);

% % % Plots % % %
figure(11),hold on
[Ybd,Xbd]=dlsim(sysbd.a,sysbd.b,sysbd.c,sysbd.d,u,[xi,0,0]);

```

```

rbd=[[Kbd,0,0,0,0]*Xbd]';%Control Input Bessel

subplot(2,1,1),
plot(t,Xbd(:,1),'b'),hold on,step(Tsbd,'g--',t)
xlabel('Time(s)');ylabel('Displacement');grid on,title('Digital Bessel
Neoclassical Observer Controller of MSD');
% subplot(2,1,2),plot(t,Xbd(:,3),'b'),hold on;grid on,title('Digital
Bessel Neoclassical Observer Controller Error');

subplot(2,1,2),
plot(t,rbd);grid on,title('Digital Bessel Neoclassical Observer
Controller Feedback Gain');hold on
% hold on

[Yid,Xid]=dlsim(sysid.a,sysid.b,sysid.c,sysid.d,u,[xi,0,0]);
rid=[[Kid,0,0,0,0]*Xid]';%Control Input ITAE

figure(12), hold on
subplot(2,1,1),
plot(t,Xid(:,1),'r'),hold on
step(Tsid,'g--',t)
xlabel('Time(s)');ylabel('Displacement');grid on,title('Digital ITAE
Neoclassical Observer Controller of MSD');
% subplot(2,1,2),plot(t,Xid(:,3),'r'),hold on;grid on,title('Digital
ITAE Neoclassical Observer Controller Error');
subplot(2,1,2),plot(t,rid,'r-.');grid on,title('Digital ITAE
Neoclassical Observer Controller Feedback Gain');hold on

hold on
[Ybwd,Xbwd]=dlsim(sysbwd.a,sysbwd.b,sysbwd.c,sysbwd.d,u,[xi,0,0]);
rbwd=[[Kbwd,0,0,0,0]*Xbwd]';%Control Input Butterworth

figure(13), hold on
subplot(2,1,1),
plot(t,Xbwd(:,1),'k'),
hold on,step(Tsbwd,'g--',t)
ylabel('Displacement');grid on,title('Digital Butterworth Neoclassical
Observer Controller of MSD');
% subplot(2,1,2),plot(t,Xbwd(:,3),'k'),hold on;grid on,title('Digital
Butterworth Neoclassical Observer Controller Error');
subplot(2,1,2),plot(t,rbwd,'k--');grid on,title('Digital Butterworth
Neoclassical Observer Controller Feedback Gain');xlabel('Time(s)');hold
on

```


Magnetic Levitation code

```

clear all
s1=1; %Controller settling time
s=0.1 ; %Observer settling time
T=0.01; %Sample Time
t=0:T:2*s1; %Time Scale
xi=[0,0,0,0.075,0,0]; %Initial Conditionv for Observer
u=ones(size(t)); %Input (Step for t>0)
int=ss(0,1,1,0); %State Space Integrator in CT
intd=c2d(int,T,'zoh'); %Digital SS integrator

% % % % System % % % %
m=0.015;g=9.81;R=.615;L=.143;x10=0.125;
A=[0 1 0;g/x10 0 -2*sqrt(g/(m*x10));0 0 -R/L];
B=[0;0;1/L];
C=[1,0,0];
D=0;

% [P,G,C,D]=c2dm(A,B,C,D,T,'zoh');
P=(eye(3)+A*T);
G=B*T;
syso=ss(P,G,C,D,T)
[n,d]=tfdata(tf(syso),'v'); %Defines the denominator of the Pre-
compensator
[yk,xk]=dlsim(P,G,C,D,u);
% figure(10); plot(t,xk)

% % % % Bessel Poles % % % % % % % %

% Observer(nth order)
pobd=exp((T/s)*[-5.0093;-3.9668-3.7845i;-3.9668+3.7845i]);

% Controller
dpb=(1/s1)*[-4.0156-5.0723i;-5.5281-1.6553i;-4.0156+5.0723i;-
5.5281+1.6553i]*0.9613
dtf=poly(dpb); %Denominator of desired transfer function-ct
num=dtf(:,5);den=dtf;Tsb=tf(num,den); %Desired Transfer function
% Tsbd=c2d(Tsb,T,'zoh');
Tssb=ss(Tsb);
TsbA=eye(4)+Tssb.a*T;
TsbB=Tssb.b*T;
Tsbd=ss(TsbA,TsbB,Tssb.c,Tssb.d,T);
Tssbd=ss(Tsbd); %SS Model of Desired Transfer function
[numbd,denbd]=ss2tf(Tssbd.a,Tssbd.b,Tssbd.c,Tssbd.d); %Defines
Numerator of Pre-compensator
[dnb,ddb]=tfdata(Tssbd,'v');
% % % % % % ITAE zero position error Pole Polynomials % % % % % % % %

%Observer(n)
wn=7.54
tfi=[1 1.75*wn 2.15*wn^2 wn^3];
Pid=roots(tfi);
poid=exp((T/s)*Pid);

%Controller

```

```

%Desired Poles
wn=4.515840287456105;
tfi=[1 2.1*wn 3.4*wn^2 2.7*wn^3 wn^4];
Pid=roots(tfi)/s1;
dtfid=poly(Pid);
numid=dtfid(:,5);denid=dtfid;
Tsi=tf(numid,denid); %Desired Transfer function
Tssi=ss(Tsi);
TsiA=eye(4)+Tssi.a*T;
TsiB=Tssi.b*T;
Tsid=ss(TsiA,TsiB,Tssi.c,Tssi.d,T);
[dni,ddi]=tfdata(Tsid,'v');
% Tsid=c2d(Tsi,T,'zoh');
Tssid=ss(Tsid); %SS Model of Desired Transfer function
[dni,ddi]=tfdata(Tsid,'v');

% % % % % % % Butterworth Polynomials % % % % % % % % %

%Observer(n)
BWpoly3=[1 2 2 1];
pBW3=roots(BWpoly3)';
BWTs3=6.639867870220427;
pBWo=(pBW3/(1/BWTs3));
pBWd=exp(pBWo*T/s);

%Controller
BWpoly=[1 2.613 3.414 2.613 1];
pBW=roots(BWpoly)';
BWTs=9.869155358643168; %Settling time of normalized butterworth
polynomial
pBWod=(pBW/(1/BWTs))/s1;
dtfbwd=poly(pBWod)
numbwd=dtfbwd(:,5);denbwd=dtfbwd;Tsbw=tf(numbwd,denbwd);
Tssbw=ss(Tsbw);
TsbwA=eye(4)+Tssbw.a*T;
TsbwB=Tssbw.b*T;
Tsbwd=ss(TsbwA,TsbwB,Tssbw.c,Tssbw.d,T);
[dnbw,ddbwd]=tfdata(Tsbwd,'v');
% Tsbwd=c2d(Tsbw,T,'zoh');%Desired Transfer function
Tssbwd=ss(Tsbwd); %SS Model of Desired Transfer function
[dnbw,ddbwd]=tfdata(Tsbwd,'v');

% % % % % % % % % % Neoclassical controller observer % % % % % % % % %

% % % % % % % Break Down % % % % % % %

%Bessel
Gsbd=minreal(Tssbd/((1-Tssbd))); %Removes feedback loop
Gssbd=minreal(Gsbd/intd); %Removes integrator
pbd=eig(Gssbd);
Kbd=place(P,G,pbd);

%ITAE
Gsid=minreal(Tssid/(1-Tssid));
Gssid=minreal(Gsid/intd);

```

```

pid=eig(Gssid);
Kid=place(P,G,pid);

%Butterworth Function
Gsbwd=minreal(Tssbwd/(1-Tssbwd));
Gssbwd=minreal(Gsbwd/intd);
pbwd=eig(Gssbwd);
Kbwd=place(P,G,pbwd);

% % % Observer Design % % %

L1=acker(P',C',pobd)';
L2=acker(P',C',poid)';
L3=acker(P',C',pBwd)';

Pbt = [P-G*Kbd   G*Kbd; zeros(size(P))   P - L1*C];
Pit = [P-G*Kid   G*Kid; zeros(size(P))   P - L2*C];
Pbwt = [P-G*Kbwd   G*Kbwd; zeros(size(P))   P - L3*C];
Gt = [G ;zeros(size(G))];
Ct = [ C   zeros(size(C))];

[ykcob,xkcob]=dlsim(Pbt,Gt,Ct,D,u,[xi]);
xcob=[xkcob(:,1) xkcob(:,2)];
ecob=[xkcob(:,3) xkcob(:,4)];
xkhatb=xcob-ecob;

[ykcoi,xkcoi]=dlsim(Pit,Gt,Ct,D,u,[xi]);
xcoi=[xkcoi(:,1) xkcoi(:,2)];
ecoi=[xkcoi(:,3) xkcoi(:,4)];
xkhati=xcoi-ecoi;

[ykcobw,xkcobw]=dlsim(Pbwt,Gt,Ct,0,u,[xi]);
xcobw=[xkcobw(:,1) xkcobw(:,2)];
ecobw=[xkcobw(:,3) xkcobw(:,4)];
xkhatbw=xcobw-ecobw;

emaxcb=max(abs(ecob(:,1)))
emaxci=max(abs(ecoi(:,1)))
emaxcbw=max(abs(ecobw(:,1)))

tecb=t*abs(ecob(:,1))
teci=t*abs(ecoi(:,1))
tecbw=t*abs(ecobw(:,1))

%%%%%%%%% Rebuild %%%%%%%%%%%%%%%

Hssbd=ss(Pbt,Gt,Ct,D,T); %State-feedback Controller w/ Observer
[nbd,dbd]=tfdata(tf(Hssbd),'v');
Prebd=tf(dnb,nbd,T);%Pre-compensator
Hsbd=Hssbd*ss(intd*Prebd/T);
sysbd=feedback(Hsbd,1);

```

```

Hssid=ss(Pit,Gt,Ct,D,T); %State-feedback Controller w/ Observer
Preid=tf(dni,n,T);%Pre-compensator
Hsid=Hssid*intd*Preid/T;
sysid=feedback(Hsid,1);

Hssbwd=ss(Pbwt,Gt,Ct,D,T); %State-feedback Controller w/ Observer
Prebwd=tf(dnbw,n,T);%Pre-compensator
Hsbwd=Hssbwd*minreal(intd*ss(Prebwd)/T);
sysbwd=feedback(Hsbwd,1);

% % % Plots % % %
figure(11),
[Ybd,Xbd]=dlsim(sysbd.a,sysbd.b,sysbd.c,sysbd.d,u,[xi,0]);
rbd=[[Kbd,0,0,0,0]*Xbd]';%Control Input Bessel
subplot(2,1,1),hold on, plot(t,-x10,'--r'),step(Tsbd,'g--',t)
plot(t,Xbd(:,1),'b');xlabel('Time(s)');ylabel('Displacement');grid on,
title('Digital Bessel Neoclassical Observer Controller For The Magnetic
Levitation System');subplot(2,1,2),plot(t,Xbd(:,4)),grid
on,title('Digital Bessel Neoclassical Observer Controller State
Estimate Error');hold on
% title('Digital Neoclassical Controller For The Magnetic Levitation
System');
% subplot(2,1,2),plot(t,rbd,'b:');grid on,title('Digital Bessel
Neoclassical Controller Control Input');xlabel('Time(s)');hold on

figure(12)
[Yid,Xid]=dlsim(sysid.a,sysid.b,sysid.c,sysid.d,u,[xi,0]);
rid=[[Kid,0,0,0,0]*Xid]';%Control Input ITAE
subplot(2,1,1),plot(t,-x10,'--r'),hold on
plot(t,Xid(:,1),'r-'),step(Tsid,'g--',t)
% hold on,plot(t,Xid(:,2),'m:');
xlabel('Time(s)');ylabel('Displacement');grid on,
title('Digital ITAE Neoclassical Controller Observer For The Magnetic
Levitation System'); subplot(2,1,2),plot(t,Xid(:,4),'r'),hold
on,plot(t,Xid(:,4),'r');grid on,title('Digital ITAE Neoclassical
Observer Controller State Estimate Error');
% title('Digital ITAE Neoclassical Controller For The Magnetic
Levitation System'),subplot(2,1,2),plot(t,rid,'r-.');grid
on,title('Digital ITAE Neoclassical Controller Control Input');hold on

figure(13)
[Ybwd,Xbwd]=dlsim(sysbwd.a,sysbwd.b,sysbwd.c,sysbwd.d,u,[xi,0]);
rbwd=[[Kbwd,0,0,0,0]*Xbwd]';%Control Input Butterworth

subplot(2,1,1),plot(t,-x10,'--r'),hold on
plot(t,Xbwd(:,1),'k-'),step(Tsbwd,'g--',t)
% hold on,plot(t,Xbwd(:,2),'c:');
ylabel('Displacement');grid on,
title('Digital Butterworth Neoclassical Controller Observer For The
Magnetic Levitation System');subplot(2,1,2),plot(t,Xbwd(:,4),'k'), grid
on,title('Digital Butterworth Neoclassical Observer Controller State
Estimate Error');
% title('Digital Butterworth Neoclassical Controller For The Magnetic
Levitation System');subplot(2,1,2),plot(t,rbwd,'k--');grid

```

```
on,title('Digital Butterworth Neoclassical Controller Control  
input');xlabel('Time(s)');hold on
```

```
minreal(minreal(zpk(sysbd))/zpk(Tsbd))  
minreal(minreal(zpk(sysid))/zpk(Tsid))  
minreal(minreal(zpk(sysbwd))/zpk(Tsbwd))
```

Ball and Beam code

```

clear all
s1=1; %Controller settling time
s=0.05 ; %Observer settling time
T=0.01; %Sample Time
int=ss([0],[1],1,0);
intd=c2d(int,T,'zoh');

%System
m = 0.0003;R = 0.01;g = -9.81;Jball = 2.88e-9;Jbeam=0.587e-3;
H=m*g/((Jbeam/R^2)+m);
A = [0 1 0 0;
      0 0 H 0;
      0 0 0 1;
      0 0 0 0];
B = [0;0;0;1/Jbeam];
C = [1 0 0 0];
D = [0];

% [P,G,C,D]=c2dm(A,B,C,D,T,'imp');
P=(eye(4)+A*T);
G=B*T;
t=0:T:2*s1;
xi=[0,0,0,0,0.2,0,0,0]; %Initial Condition
u=ones(size(t)); %Input
[yk,xk]=dlsim(P,G,C,D,u);
figure(1)
osysd=ss(P,G,C,D,T);
% rlocus(osysd)

Co=ctrb(P,G);rank(Co)
Ob=obsv(P,C);rank(Ob)

%%%Bessel Poles
%Observer(nth order)
poB=exp((T/s)*[-4.0156-5.0723i;-5.5281-1.6553i;-4.0156+5.0723i;-
5.5281+1.6553i]);

%Controller((n+1) order)
dpb=(1/s1*[-6.448;-4.1104+6.3142i;-5.9268+3.0813i;-4.1104-6.3142i;-
5.9268-3.0813i])*0.96549;

%%%%%ITAE Polynomials
%Observer(n)
wno=4.515840287456105;
ITAEpoly=[1 2.1*wno 3.4*wno^2 2.7*wno^3 wno^4];
poi=roots(ITAEpoly)';
poid=exp((T/s)*poi);

%Controller(n+1)
wnc=6.657;
tfi=[1 2.8*wnc 5*wnc^2 5.5*wnc^3 3.4*wnc^4 wnc^5]*1.03;

%%%%%Butterworth Polynomials
%Observer(n)

```

```

BWpoly=[1 2.613 3.414 2.613 1];
pBW=roots(BWpoly)';
BWTs=9.869155358643168; %Settling time of normalized butterworth
polynomial
pBWod=exp((pBW/(1/BWTs))*T/s); % 2nd order butterworth

% Controller(n+1)
BWpoly3=[1 3.236 5.236 5.236 3.236 1];
pBW3=roots(BWpoly3)';
BWTs3=10.8375;
pBW03d=1/s1*(pBW3/(1/BWTs3));
%Neoclassical controller observer

% % % % % % % % % % % % % %Bessel Function

dtf=poly(dpb);
num=dtf(:,6);den=dtf;Tsb=tf(num,den); %Desired Transfer function
Tssb=ss(Tsb);
TsbA=eye(5)+Tssb.a*T;
TsbB=Tssb.b*T;
TsbD=ss(TsbA,TsbB,Tssb.c,Tssb.d,T);
[dnb,ddb]=tfdata(TsbD,'v');
Tssbd=ss(TsbD);
GsbD=minreal(Tssbd/(1-Tssbd));
Gssbd=minreal(GsbD/intd);
pbd=eig(Gssbd);
Kbd=place(P,G,pbd);

%Desired ITAE zero position error Pole
Pid=roots(tfi)/s1;
dtfid=poly(Pid);
numid=dtfid(:,6);denid=dtfid;Tsi=tf(numid,denid); %Desired Transfer
function
Tssi=ss(Tsi);
TsiA=eye(5)+Tssi.a*T;
TsiB=Tssi.b*T;
Tsid=ss(TsiA,TsiB,Tssi.c,Tssi.d,T);
[dni,ddi]=tfdata(Tsid,'v');
Tssid=ss(Tsid);
Gsid=minreal(Tssid/(1-Tssid));
Gssid=minreal(Gsid/intd);
pid=eig(Gssid);
Kid=place(P,G,pid);

%third order butterworth State Space model

dtfbwd=poly(pBW03d);
numbwd=dtfbwd(:,6);denbwd=dtfbwd;Tsbw=tf(numbwd,denbwd);
Tssbw=ss(Tsbw);
TsbwA=eye(5)+Tssbw.a*T;
TsbwB=Tssbw.b*T;
TsbwD=ss(TsbwA,TsbwB,Tssbw.c,Tssbw.d,T);
[dnbw,ddbW]=tfdata(TsbwD,'v');
Tssbwd=ss(TsbwD);
GsbwD=minreal(Tssbwd/(1-Tssbwd));
Gssbwd=minreal(GsbwD/intd);

```

```

pbwd=eig(Gssbwd);
Kbwd=place(P,G,pbwd);

% % % % % % % % % % % The Observer % % % % % % % % % % %
L1=place(P',C',poB)';
L2=place(P',C',poid)';
L3=place(P',C',pBWod)';

Pbt = [P-G*Kbd   G*Kbd;   zeros(size(P))   P-L1*C];
Pit = [P-G*Kid   G*Kid;   zeros(size(P))   P-L2*C];
Pbwt = [P-G*Kbwd  G*Kbwd; zeros(size(P))   P-L3*C];
Gt = [G ;zeros(size(G))];
Ct = [ C  0 0 0 0];

[stkobd,xkobd]=dlsim(Pbt,Gt,Ct,D,u,xi);
xcobd=[xcobd(:,1) xkobd(:,2)];
ecobd=[xcobd(:,5) xkobd(:,6)];
xkhatbd=xcobd-ecobd;

[stkoid,xkoid]=dlsim(Pit,Gt,Ct,D,u,xi);
xcoid=[xcoid(:,1) xkoid(:,2)];
ecoid=[xcoid(:,5) xkoid(:,6)];
xkhatid=xcoid-ecoid;

[stkbwd,xkbwd]=dlsim(Pbwt,Gt,Ct,0,u,xi);
xcobwd=[xkbwd(:,1) xkbwd(:,2)];
ecobwd=[xkbwd(:,5) xkbwd(:,6)];
xkhatbwd=xcobwd-ecobwd;

% % % % % % % Observer Analysis %%%%%%%%%%%
emaxcb=max(abs(ecobd(:,1)));
emaxcid=max(abs(ecoid(:,1)));
emaxcbwd=max(abs(ecobwd(:,1)));

tecbd=t*abs(ecobd(:,1));
tecid=t*abs(ecoid(:,1));
tecbwd=t*abs(ecobwd(:,1));

Hssbd=ss(Pbt,Gt,Ct,D,T);
[nbd,dbd]=tfdata(tf(Hssbd),'v');
Prebd=tf(dnb,nbd,T);

Hsbd=Hssbd*intd*ss(Prebd/T
sysbd=(feedback(Hsbd,1));
minreal(zpk(sysbd))/minreal(zpk(Tsbd));
[Ybd,Xbd]=dlsim(sysbd.a,sysbd.b,sysbd.c,sysbd.d,u,[xi,0]);
[Ybo,Tbo,Xbo]=step(Tsbd,2*s1);
figure(1)
subplot(2,1,1),plot(t,Xbd(:,1),'b');xlabel('Time(s)');ylabel('Displacem
ent'),hold on, step(Tsbd,'g-.');grid on
title('Digital Bessel Neoclassical Observer Controller For The Ball &
Beam System');subplot(2,1,2),plot(t,ecobd(:,1)),grid on,title('Digital
Bessel Neoclassical Observer Controller State Estimate Error');
% rbd=[[Kbd,0,0,0,0,0]*Xbd]';%Control Input

```



```

% title('Digital Bessel Neoclassical Controller For The Ball & Beam
System'); subplot(2,1,2),plot(t,rbd,'b');grid on,title('Digital Bessel
Neoclassical Controller Control Input');xlabel('Time(s)');hold on
hold on

Hssid=ss(Pit,Gt,Ct,D,T);
[nid,did]=tfdata(tf(Hssid),'v');
Preid=tf(dni,nid,T);

Hsid=Hssid*intd*ss(Preid/T);
sysid=(feedback(Hsid,1));
minreal(minreal(zpk(sysid))/zpk(Tsid))
[Yid,Xid]=dlsim(sysid.a,sysid.b,sysid.c,sysid.d,u,[xi,0]);
[Yio,Tio,Xio]=step(Tsid,2*s1);

figure(22)
subplot(2,1,1),plot(t,Xid(:,1),'R');xlabel('Time(s)');ylabel('Displacem
ent'),hold on, step(Tsid,'g-.');grid on
title('Digital ITAE Neoclassical Observer Controller For The Ball &
Beam System');subplot(2,1,2),plot(t,ecoid(:,1),'R');grid
on,title('Digital ITAE Neoclassical Observer Controller State Estimate
Error');
% rid=[[Kid,0,0,0,0,0]*Xid]';%Control Input Butterworth
% title('Digital ITAE Neoclassical Controller For The Ball & Beam
System'); subplot(2,1,2),plot(t,rid,'r');grid on,title('Digital ITAE
Neoclassical Controller Control Input');xlabel('Time(s)');hold on

Hssbwd=ss(Pbwt,Gt,Ct,D,T);
[nbwd,dbwd]=tfdata(tf(Hssbwd),'v');
Prebwd=tf(dnbw,nbwd,T);

Hsbwd=Hssbwd*intd*ss(Prebwd/T);
sysbwd=(feedback(Hsbwd,1));minreal(minreal(zpk(sysbwd))/zpk(Tsbwd))
[Ybwd,Xbwd]=dlsim(sysbwd.a,sysbwd.b,sysbwd.c,sysbwd.d,u,[xi,0]);
[Ybwo,Tbwo,Xbwo]=step(Tsbwd,2*s1);

figure(23)
subplot(2,1,1),plot(t,Xbwd(:,1),'k');xlabel('Time(s)');ylabel('Displace
ment'),hold on, step(Tsbwd,'g-.');grid on
title('Digital Butterworth Neoclassical Observer Controller For The
Ball & Beam System');subplot(2,1,2),plot(t,Xbwd(:,5),'k');grid
on,title('Digital Butterworth Neoclassical Observer Controller State
Estimate Error');
% rbwd=[[Kbwd,0,0,0,0,0]*Xbwd]';%Control Input Butterworth
% title('Digital Butterworth Neoclassical Controller For The Ball &
Beam System'); subplot(2,1,2),plot(t,rbwd,'k--');grid on,title('Digital
Butterworth Neoclassical Controller Control
Input');xlabel('Time(s)');hold on

```