HYPERPLANE DESIGN FOR DISCRETE VARIABLE STRUCTURE CONTROL WITH LOOP TRANSFER RECOVERY

By

RYAN TODD LYLE

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1998

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2000

HYPERPLANE DESIGN FOR DISCRETE

VARIABLE STRUCTURE CONTROL

WITH LOOP TRANSFER

RECOVERY

Thesis Approved:

Thesis Adviser

prabhakar pagilla

Alfred Carlozzi

Dean of the Graduate College

PREFACE

Practical digital control systems often include a variety of undesirable effects including sampling time issues, parameter variations and exogenous disturbances. There exist many powerful modern control theories and subsequent tools for controller development which address these and other issues by guarantying stability by design in some sense based on tunable parameters. Seemingly, one inherent quality of these techniques is the tuning procedure becomes tedious and time consuming to obtain a desired level of performance. This paper presents a method of combining the ideas of loop transfer recovery (LTR) with observer based discrete variable structure control (OBDVSC) in an effort to retain design freedoms of LTR and robustness properties of OBDVSC ultimately yielding an easily tuned practical compensator.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation and many thanks to Dr. Eduardo A. Misawa for his thoughts, guidance and advice throughout this work. Without his presence and expertise in overseeing the project the end would have seemingly be out of reach. To my parents and grandparents, thank you for giving me the desire, opportunity and motivation to excel and achieve within my own potential throughout the many hurdles of life. To my sister, thank you for always guiding me through the exemplary path you have set. To my wife, Mrs. Lindsay K. Lyle, thank you for the unconditional love and support you have so unselfishly extended. Thank you to my past and present lab-mates of the Advanced Control Laboratory, Thomas Rendon, May-Win Thein, Brian O'Dell, Hanz Richter, Ban Fu Chee, Ivan Blanco and Chad Stoecker. And finally, thank you to my committee members, Dr. Prabakkar Pagilla and Dr. Gary Yen for your help and guidance throughout my master studies.

TABLE OF CONTENTS

Chapter Page

1 **Introduction** **1**

1.1 Objectives and Motivations 3

1.2 Contributions 3

1.3 Limitations 4

1.4 Thesis Outline 4

2 **Variable Structure Control Overview** **6**

2.1 Sliding Mode Control Essentials 6

2.1.1 A Note on Model-Based Tracking and VSC 11

2.1.2 Observer Based Variable Structure Control (OBVSC) 13

2.2 Discrete Variable Structure Control (DVSC) 14

2.3 OBDVSC with Disturbance Observer 15

2.3.1 Bias Estimator 17

2.3.2 Disturbance State Modelling 18

2.3.3 OBDVSC with Feedforward Disturbance Compensation 19

2.4 Hyperplane Design via LQR 20

2.4.1 Eigenvalue Constraint 21

2.4.2 LQR Hyperplane Design Procedure 22

2.5 Design Example: OBDVSC with Disturbance Observer 24

2.5.1 Disk Drive Model 24

2.5.2 Disturbance Model 26

Chapter Page

2.5.3 OBDVSC with Disturbance Observer Simulations 28

2.6 Summary 30

3 Loop Transfer Recovery Overview 33

3.1 LQG/LTR 34

3.1.1 Target Filter Design 38

3.1.2 Target Filter Loop Design Methods 39

3.2 Discrete Time LQG/LTR 41

3.2.1 LTR: Continuous vs. Discrete 42

3.2.2 Observer Selection for Discrete LTR 43

3.2.3 Prediction Observer 44

3.2.4 Current Observer 44

3.3 A Tool for Measuring Loop Transfer Recovery 45

3.4 Design Example: LQG/LTR 47

3.4.1 Remotely Piloted Vehicle Model 47

3.4.2 LTR Simulations 52

3.5 Summary 53

4 Discrete Variable Structure Control with LTR Hyperplane Design 57

4.1 OBDVSC, Regulation Inside the Boundary Layer 58

4.2 Fitting OBDVSC Into a LTR Framework for Recovery 58

4.3 Recovering the Target Filter Loop 59

4.4 Obtaining Exact Recovery 63

4.5 Design Procedure 65

5 LTR Hyperplane Design Example 66

5.1 Symbolic Disk Drive Model 66

5.2 Target Filter Loop Design 69

Chapter Page

5.3 Hyperplane Design using Loop Transfer Recovery 71

5.4 Behavior Outside the Boundary Layer 73

6 Conclusions **78**

6.1 Contributions 78

6.2 Future Work 79

Bibliography **80**

A MATLAB Setup files for Example 2.1 **86**

A.1 Main Setup File 86

A.2 Setup File for Disturbance Model. 91

B MATLAB Setup files for Example 3.4 **93**

B.1 Setup File for Continuous Time LQG/LTR Example of Section 3.4 . 93

B.2 Setup File for Discrete Time LQG/LTR Example of Section 3.4 . . . 96

C MATLAB Setup files for Chapter 5 **99**

C.1 Main Setup File 99

C.2 Filter Loop Design and Recovery 101

LIST OF TABLES

Table Page

2.1 Frequency and Damping Ratios of Disturbance Model Transfer Function. 27

2.2 Simulation Conditions. 29

3.1 Plant Pole Locations for RPV Model. 49

5.1 Simulation Parameters. 68

LIST OF FIGURES

Figure Page

2.1 Fishbone of Trajectories. 7

2.2 Reaching and Sliding Motion in Phase Plane. 9

2.3 Reaching and Sliding Motion in Phase Plane with Chattering. 10

2.4 Saturation Function, $sat(\frac{S}{\phi})$. 11

2.5 Sliding Surface with Boundary Layer. 12

2.6 Observer Based VSC Block Diagram. 14

2.7 Quasi Sliding Mode in Discrete Time, [11]. 16

2.8 OBDVSC Disturbance Compensation Scheme 21

2.9 Disturbance Generation using Random Input. 27

2.10 Magnitude & Phase of the Disturbance Model Transfer Function. 28

2.11 Position Response and Disturbance Estimation for Bias plus Vibration Disturbance Input. 30

2.12 Position Response for Model Based Disturbance Input. 31

2.13 Disturbance Estimates for Model Based Disturbance Input. 31

2.14 Estimate Error of Bias Observer and Disturbance State Modelling Observer. 32

Figure Page

3.1 Typical Feedback Control System. 36

3.2 Target Loop for Loop Transfer Recovery (LTR) at Plant Output. 37

3.3 Model Based Compensator, $K(s)$. 37

3.4 Singular Value Plots of Target Filter Design. 50

3.5 Loop Transfer Recovery of Target Filter Loop Using H. 50

3.6 Loop Transfer Recovery of Target Filter Loop Using H_o. 51

3.7 Model Based Compensator Block Diagram. 52

3.8 Discrete Model Based Compensator Block Diagram. 52

3.9 Step Response of MBC using H and H_o TFL Designs. 53

3.10 Singular Value Plots of Discrete Target Filter Design. 54

3.11 Discrete Loop Transfer Recovery of Target Filter Loop Using H. 54

3.12 Discrete Loop Transfer Recovery of Target Filter Loop Using H_o. 55

3.13 Step Response for Model Based Compensator using H and H_o Discrete Target Filter Loop Designs. 55

4.1 DVSC Recovery at the Plant Output. 59

4.2 DVSC Recovery at the Plant Output for Regulation Control Objective. . 60

4.3 DVSC Mimicking the LTR Standard Control Loop. 60

5.1 Symbolic 4^{th} Order Disk Drive Model. 67

5.2 Bode Diagram Comparison of Symbolic Drive Model to Goh *et.al*. 67

Figure Page

5.3 $C\Phi(z)H$, Target Filter Loop Block Diagram (Open-Loop). 71

5.4 Compensator Loop Block Diagram (Closed-Loop). 71

5.5 $C\Phi(z)H$, Target Filter Loop Block Diagram Loop (Closed-Loop) 73

5.6 Compensator Loop Block Diagram Loop (Closed-Loop). 73

5.7 Frequency Response of the Target Filter Loop. 74

5.8 Target Filter Loop Recovery ($\rho = 10^{14}$, 10^{16}, 10^{17}). 75

5.9 Time Domain Responses. 75

5.10 Sliding Surface for Initial Conditions $[\frac{25}{TPI}\,0\,0\,0]^T$. 76

5.11 Sliding Surface for 25 Track Set Point Change. 77

NOMENCLATURE

Notations used in this thesis include:

$j := \sqrt[2]{-1}$,

$s :=$ Laplacian variable,

$z :=$ complex number, ($z = a + jb$ for $a, b \in \mathbb{R}$),

$\mathbb{R} :=$ field of real numbers,

$\mathbb{C} :=$ field of complex numbers,

$\mathbb{C}^{-} :=$ open left half complex plane, ($z \in \mathbb{C} | Re[z] < 0$),

$\mathbb{C}^{\odot} :=$ unit disk of the complex plane,

$\mathbb{C}^{\otimes} :=$ complement set of $\mathbb{C}^{\odot}$,

$\mathbb{R}^{n \times m} :=$ set of real matrices with n rows, m columns,

$Re[z] :=$ real component of complex number z,

$Im[z] :=$ complex component of complex number z,

$\text{rank}(A) :=$ the rank of matrix A,

$\det(A) :=$ the determinant of matrix A,

$A^T :=$ the transpose of matrix A,

$A^* :=$ the complex conjugate transpose of matrix A,

$\lambda(A) :=$ set of eigenvalues of matrix A, where $A \in \mathbb{R}^{n \times n}$,

$\ker[G] :=$ the kernel or null space of matrix G,

$\Phi(s) := (sI - A)^{-1}$,

$\Phi(z) := (zI - A)^{-1}$,

$\Sigma_c :=$ quadruple $\Sigma(A, B, C, D)$ constrained by ($\dot{x} = Ax + Bu$, $y = Cx + Du$),

$\Sigma_d :=$ quadruple $\Sigma_d(A, B, C, D)$ for ($x(k+1) = Ax(k) + Bu(k)$, $y = Cx(k) + Du(k)$),

Chapter 1

Introduction

Loop transfer recovery (LTR) theory has provided a powerful modern compensator design technique and at nearly the same time, variable structure control (VSC) theory has also emerged in its own right. Both theories offer desirable characteristics attractive to practical control engineering problems. For instance, the LTR technique allows for a control system to counteract disturbances at the plant input or output for either single-input single-output (SISO) or multi-input multi-output (MIMO) systems often yielding designs retaining properties associated with optimal control theory. Similarly, the VSC technique has also shown value through application to practical control problems often yielding enhanced performance and disturbance rejection.

LTR theory is generally based on three main steps; (1) formulate all design specifications (*i.e.*, robustness requirements and performance criteria) as restrictions on singular values of an open loop transfer function matrix obtained by breaking the control loop at either the input or output of the plant (2) design a target loop using optimal control theory to meet the design specifications of step (1), and (3) solve a linear quadratic regulator (LQR) problem for small control weighting to recover the loop shape of the target loop designed in step (2). The result of the aforementioned design procedure is a compensator (*i.e.*, controller/observer pair) which meets specific design requirements across a desired range of frequencies. An attractive feature of LTR theory is a single design process which combines observer and controller design.

Variable structure control (VSC) is a nonlinear, Lyapunov based technique which describes a class of control systems that allow control law structure to be changed during a given control process for enhanced system performance. For VSC systems, a switching function is generally included in the control structure, hence the name variable structure. By design, state space trajectories are attracted to a predescribed hyperplane, or switching surface. Invariance of the surface constrains trajectories to remain upon the surface once encountered. Motion along the hyperplane surface is commonly refered to as sliding mode. Often switching functions for variable structure control exhibit an undesirable chattering behavior around the switching surface. To avoid this phenomena, a boundary layer is placed around the switching surface so that motion near the boundary layer exhibits a pseudo sliding mode. For practical applications using variable structure control, a sampling process is applied and an observer is implemented to estimate unmeasurable states. Design of both the observer and sliding surface for observer based discrete variable structure control (OBDVSC) schemes becomes a separate process unlike the LTR design process.

This thesis presents a hyperplane design technique for a practical variable structure compensator. The new frequency based design methodology considers an observer based discrete variable structure control system operating under a regulation control objective within the boundary layer. The novel design technique parallels ideas from loop transfer recovery which ultimately combines the observer/controller design for discrete variable structure systems within a single process.

1.1 Objectives and Motivations

The objective of this research is to develop a compensator design technique combining observer based discrete variable structure control (OBDVSC) and the ideas behind the LTR mechanism in an effort to yield a more structured approach for tuning an OBDVSC compensator scheme. This research originally began as an attempt to enhance the disturbance rejection properties of OBDVSC while inside the boundary layer by implementing augmented prediction observer structures (*i.e.*, including a disturbance observer for extended disturbance compensation), which lead to the emphasis of one significant shortcoming of the OBDVSC technique stemming from a separate observer and sliding hyperplane design procedure. Performance of the overall OBDVSC system is sensitive to both observer and sliding surface design. One key assumption in the OBDVSC theory presented is that the error between the observer state estimate relative to the actual state vanishes sufficiently fast, so that the observer estimates are sufficiently accurate, (Misawa, [1]). Mathematically this ia a valid assumption, but in practice the task of simultaneously designing an accurate state/disturbance estimator and sliding hyperplane ultimately equates to vast amounts of tuning time. This research seeks to reduce tuning time for an OBDVSC scheme by developing a novel loop transfer recovery hyperplane design technique.

1.2 Contributions

Design examples for each individual technique are detailed in illustrative example problems. The combination of OBDVSC with LTR hyperplane design is proposed. Conditions for the exact recovery of a target loop in discrete time based on breaking the control loop at the plant output are derived for the OBDVSC system using the new hyperplane design technique. Discrete hyperplane design using loop transfer recovery is illustrated via simulation results for an OBDVSC system yielding a frequency based hyperplane design.

1.3 Limitations

Limitations of this research stem directly from the restrictions related to discrete time LTR and the LQR hyperplane design technique (see Section 2.4) used for recovery. Due to the sampling process, discrete LTR offers the choice of using either a prediction or current estimator. For the new hyperplane design technique a prediction estimator is used. Further, discrete LTR is well suited for minimum phase design plants, (see Chapter 4). Several limitations related to hyperplane design for OBDVSC using LTR include satisfying; (1) operation inside the boundary layer (2) a regulation control objective (3) solving a cheap control problem using a LQR hyperplane design technique, (see Section 4.3).

1.4 Thesis Outline

This thesis is arranged in the following order: Chapter 1 briefly describes issues related to the problem and solution formulated within this thesis through an introduction, objective, contributions and limitations sections. Background information essential for the OBDVSC hyperplane design via LTR is discussed in Chapters 2 and 3 including design examples. Chapter 2 begins with essential ideas for variable structure control theory. Continuous time and discrete time systems are discussed, with emphasis being placed on an observer based discrete variable structure compensation scheme with disturbance observer for extended disturbance compensation. Chapter 3 reviews the LTR methodology including a special model based compensator (MBC) technique known as LQG/LTR in both continuous and discrete time. A recovery error matrix useful for showing exact recovery for the

OBDVSC hyperplane design is developed in detail for recovery at the plant output. Chapter 4 investigates theoretical aspects of OBDVSC with LTR hyperplane design under a strict operating condition inside the boundary layer. Chapter 5 shows OBDVSC with LTR hyperplane design technique via simulation examples for a voice coil motor (VCM) model of a disk drive. Chapter 6 gives conclusions and ideas for future research. MATLAB codes used to simulate the design examples are given in an appendix.

Chapter 2

Variable Structure Control Overview

Variable structure control (VSC) system theory was originally developed in Russia and did not surface elsewhere until the 1970's when a book and survey paper by Itkis and Utkin, [2, 3], appeared transcribed in English. Since that time, the original ideas within variable structure control have matured and successfully extended into many fields and applications including nonlinear state estimation, adaptive systems, tracking systems, regulating systems, robot manipulators, underwater vehicles, automotive technology, disk drives and more [4, 5]. The purpose of this chapter is to briefly overview some of the fundamental ideas behind variable structure control for a practical OBDVSC compensation scheme. This chapter begins with a description of sliding mode control from a continuous-time framework leading ultimately to an application of observer based discrete-time variable structure control.

2.1 Sliding Mode Control Essentials

VSC systems possess the unique feature of a *changing* control structure, hence the name variable structure. The process contained within a VSC system can most

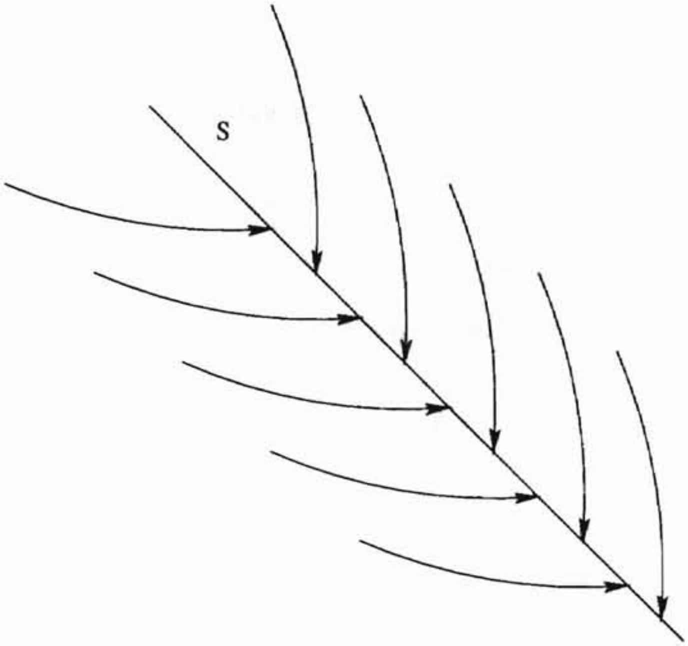

Figure 2.1: Fishbone of Trajectories.

generally be broken into two main mechanisms: (1) By design, a hyperplane or sliding surface in the state space attracts system trajectories, as illustrated in Figure 2.1. During this phase, commonly refered to as the reaching phase, all trajectories point towards a sliding surface. (2) Once on the sliding surface an invariance condition is maintained such that trajectories remain on or *slide* along a switching surface. This motion is known as the *sliding mode*.

For example, consider the explanation of variable structure control theory applied to a double integrator system (Spurgeon and Edwards, [5]),

$$\ddot{y}(t) = u(t) \tag{2.1}$$

To illustrate fundamental concepts of VSC theory, suppose trajectories are to be driven to the origin using the control law to be given by

$$u(t) = \begin{cases} -1, & S(y,\dot{y}) < 0 \\ +1, & S(y,\dot{y}) > 0 \end{cases} \tag{2.2}$$

where the surface S is defined by

$$S(y, \dot{y}) = my + \dot{y}$$

and where m is a positive design scalar[1]. The variable structure of the control law $u(t)$ is clearly seen depending on the sign of S. The control law $u(t)$ can be replaced with the $sgn(\cdot)$ or *signum* function which is equivalently stated as,

$$u(t) = -sgn(S(t)) \tag{2.3}$$

where the *sgn(·)* function is defined as

$$sgn(\cdot) = \begin{cases} +1, & sgn(\cdot) > 0 \\ -1, & sgn(\cdot) < 0 \end{cases}$$

One fundamental property of the sign function is that

$$S \cdot sgn(S) = |S| \tag{2.4}$$

Using the above definitions and considering the case for $m|\dot{y}| < 1$, if $V = \frac{1}{2}S^2$ is taken as a candidate Lyapunov function, stability of the sliding surface in the sense of Lyapunov[2] requires that $\dot{V} = S\dot{S}$ to be negative definite. Calculating $\dot{V}$ yields,

$$\begin{aligned} S\dot{S} &= S(m\dot{y} + \ddot{y}) && (2.5) \\ &= S(m\dot{y} - sgn(S)) && (2.6) \\ &< |S|(m|\dot{y}| - 1) < 0 && (2.7) \end{aligned}$$

which means the sliding surface is locally attractive. Mathematically, this behavior near the sliding surface, *i.e.*, $S \approxeq 0$, can be expressed as Equation (2.8), (Spurgeon and Edwards, [5]),

$$lim_{S\to 0^+}\dot{S} < 0 \qquad lim_{S\to 0^-}\dot{S} > 0 \tag{2.8}$$

[1] In general, for the state space $\mathbb{R}^n$, a surface can be chosen as $S(y, \dot{y}, \ddot{y}, \cdots) = (\frac{d}{dt} + m)^{n-1}y$, (Slotine and Li, [4]).

[2] For a general discussion of Lyapunov theory and stability definitions see [4], [12].

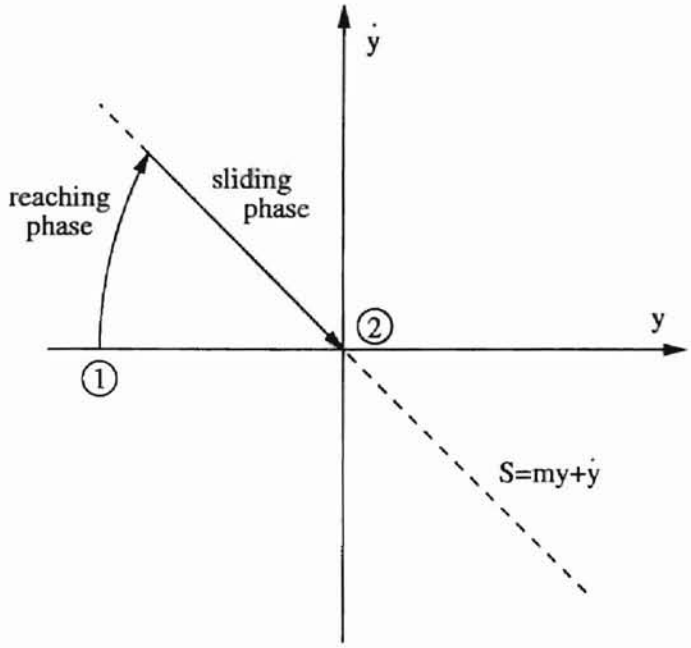

Figure 2.2: Reaching and Sliding Motion in Phase Plane.

Graphically thinking of the second order system in a state space placing position and velocity on the abscissa and ordinate axis respectively yields Figure 2.2. Recall that the sliding surface was chosen as $S = my + \dot{y}$, for m being a design parameter. Now suppose a control objective is to move the system from point 1 labeled in Figure 2.2 to the origin, or point 2. Using the control law proposed above, $u(t) = -sgn(S)$, system trajectories exhibit two behaviors, namely reaching and sliding. The reaching phase shows local system trajectories being attracted to the sliding surface. Once on the surface, the trajectories remain and *slide* to the desired set point. Alternatively, letting the reaching condition $S\dot{S} < 0$ be satisfied by

$$S\dot{S} = -\eta|S| \tag{2.9}$$

or,

$$\dot{S} = -\eta \cdot sgn(S), \quad \forall \quad \eta > 0 \tag{2.10}$$

ensures that any arbitrary initial conditions will reach the surface $S = 0$ in finite time and that $S = 0$ is an invariant subspace, (Slotine and Li, [4]).

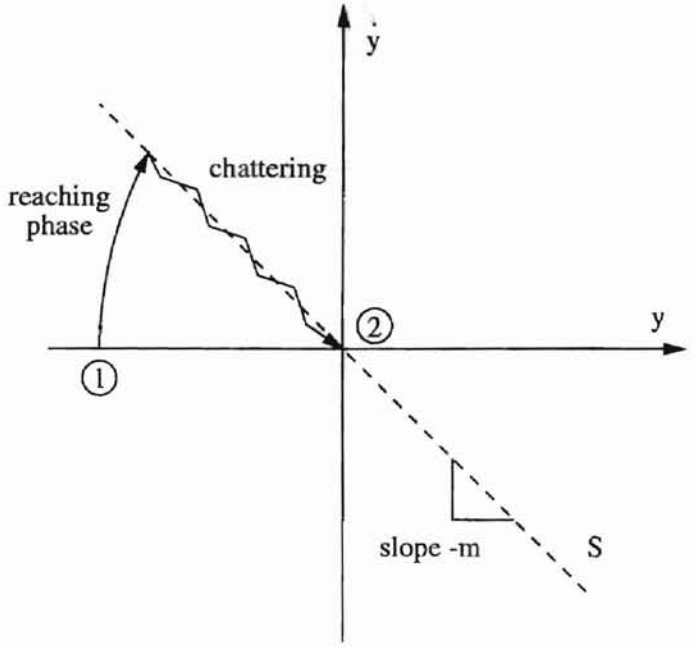

Figure 2.3: Reaching and Sliding Motion in Phase Plane with Chattering.

Figure 2.2 represents what is commonly refered to in the literature as *ideal sliding* motion. Practically speaking, ideal sliding motion is not possible because of the high frequency switching necessary to maintain system trajectories on $S = 0$. Intuitively, motion around $S = 0$ would be more accurately illustrated by Figure 2.3. Instead of smoothly riding the sliding surface, trajectories oscillate about $S = 0$. This *chattering* motion is an undesirable effect, especially when plants may include unmodeled dynamics at high frequencies. Allowing chattering around the sliding surface may possibly excite unwanted resonant vibration modes in mechanical structures. To combat the chattering effect the $sgn(\cdot)$ function can be exchanged with a $sat(\cdot)$ function and a boundary layer ϕ surrounding and parallel to $S = 0$, where the $sat(\frac{S}{\phi})$ is defined as

$$sat(\frac{S}{\phi}) = \begin{cases} +1, & S(\cdot) > \phi \\ \frac{S}{\phi}, & |S| \leq \phi \\ -1, & S(\cdot) < \phi \end{cases} \tag{2.11}$$

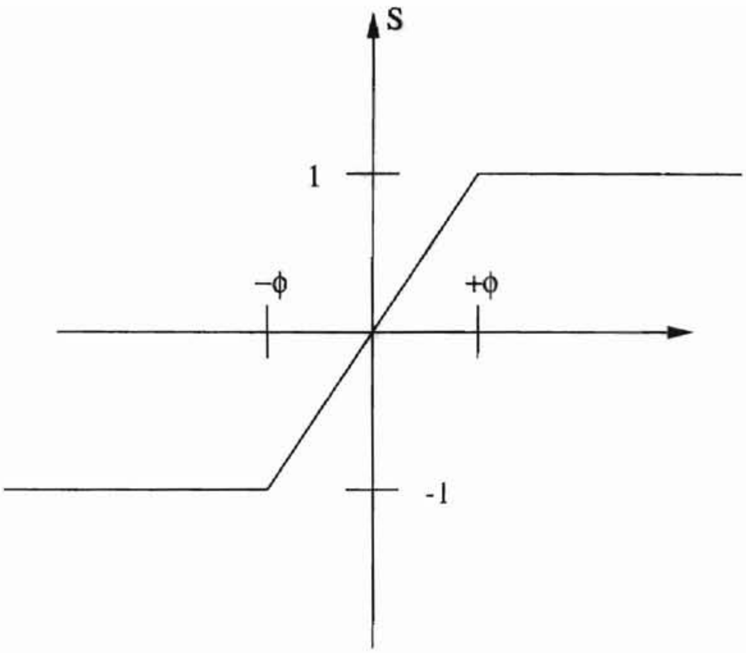

Figure 2.4: Saturation Function, $sat(\frac{S}{\phi})$.

A graphical interpretation of Equation (2.11) is shown in Figure 2.4. A possible system trajectory using a variable structure control with $sat(\frac{S}{\phi})$ may be similar to Figure 2.5.

2.1.1 A Note on Model-Based Tracking and VSC

The previous example considered a regulation control objective which corresponded to driving the system trajectory to the origin of the phase plane for a double integrator system. Representing $\ddot{y} = u(t)$ in a state space representation, $\Sigma(A, B, C)$ yields,

$$\dot{x} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} y \\ \dot{y} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B} u(t) \qquad (2.12)$$

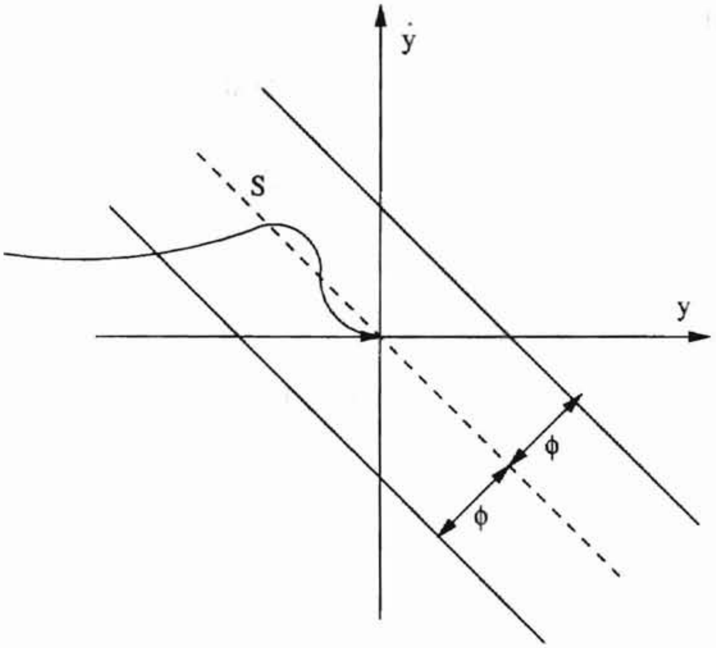

Figure 2.5: Sliding Surface with Boundary Layer.

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} y \\ \dot{y} \end{bmatrix}}_{x} \tag{2.13}$$

For the SISO second order system we have, $A \in \mathbb{R}^{2\times 2}$, $B \in \mathbb{R}^{2\times 1}$, $C \in \mathbb{R}^{1\times 2}$, $x \in \mathbb{R}^{2\times 1}$ and $u \in \mathbb{R}$. Suppose the control objective is changed from regulation to tracking and that the dynamics to track are given by quadruple $\Sigma_{track}(A_d, B_d, C_d, D_d)$ such that x_d is the desired state to track and u_d is a convenient control input. Let $\tilde{x} = x - x_d$, and define a sliding surface as,

$$S = G \cdot \tilde{x} \tag{2.14}$$

where $G \in \mathbb{R}^{1\times 2}$. As before, let $V = \frac{1}{2}S^2$ be a Lyapunov candidate function so that showing $\dot{V}$ negative definite yields stability. Calculating $S\dot{S}$ yields

$$\begin{aligned} S\dot{S} &= S \cdot G\dot{\tilde{x}} && (2.15)\\ &= S \cdot G(\dot{x} - \dot{x}_d) && (2.16)\\ &= S \cdot G(Ax + Bu - Ax_d - Bu_d) && (2.17)\\ &= S(GA\tilde{x} + GB(u - u_d)) && (2.18) \end{aligned}$$

Noting that $GB \in \mathbb{R}$ for SISO and choosing u as

$$u = -\frac{1}{GB}(GA\tilde{x} + Ksat(\frac{S}{\phi})) + u_d \qquad (2.19)$$

where $K > 0$ is an arbitrary design constant yields

$$S\dot{S} = -SKsat(\frac{S}{\phi}) \qquad (2.20)$$

which is of the form $S\dot{S} < -\frac{K}{\phi}S^2$. To track x_d we desire $x \to x_d$ as $t \to \infty$. By defining an error, the tracking objective is transformed into a regulation problem in $\tilde{x}$. This transformation of a tracking problem into a regulation regulation problem is a common result, but within a sliding mode control context serves to show the need for generating the desired trajectories, x_d. Trajectory shaping and generation can be accomplished via a x_d-generator (Richter, [6]).

2.1.2 Observer Based Variable Structure Control (OBVSC)

The tracking problem shown previously assumes full state feedback. Practically speaking, all of the states are generally not available for feedback, and thus an observer must be implemented. The effect of an observer used in VSC can be argued to be negligible if the convergence of state estimates is fast enough (Misawa, [1]). An OBVSC compensator, observer-controller pair, is illustrated in block diagram form in Figure 2.6. The control law u in Equation (2.23) for the

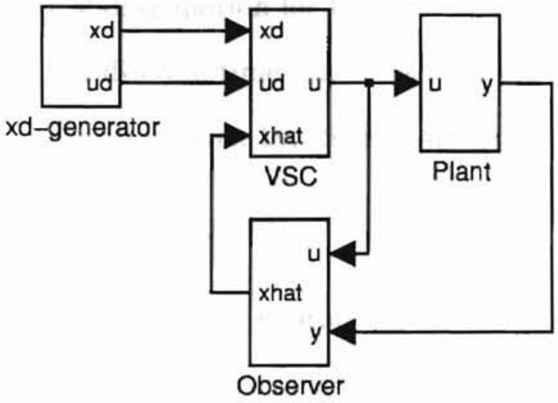

Figure 2.6: Observer Based VSC Block Diagram.

observer based compensator is taken from Equation (2.19) using two modifications given by Equation (2.21) and sliding surface given in Equation (2.22).

$$\hat{\tilde{x}} = \hat{x} - x_d \tag{2.21}$$

$$\hat{S} = G\hat{\tilde{x}} \tag{2.22}$$

$$u = -\frac{1}{GB}(GA\hat{\tilde{x}} + Ksat(\frac{\hat{S}}{\phi})) + u_d \tag{2.23}$$

where $K > 0$ is an arbitrary design constant and $\hat{x}$ is the estimate of the state vector x generated by a suitable observer, such as Luenberger observer[3]. Assuming that the error between the state estimate $\hat{x}$ and the actual state x vanishes fast enough, the observer based variable structure control essentially behaves like the variable structure control system assuming full state feedback.

2.2 Discrete Variable Structure Control (DVSC)

Variable structure control in a discrete context is conceptually different from the continuous time VSC counterpart strictly because of the sampling process. It is well

[3] For a general discussion of Luenberger observer theory in discrete-time see Franklin *et.al.*, [10].

known that the achievable sliding motion for DVSC systems can be graphically represented as Figure 2.7. A discrete time counterpart of the continuous time sliding condition, (*i.e.*, $S\dot{S} < 0$) is given by, (Sira-Ramirez, [9])

$$|S_i(k+1)| < |S_i(k)|, \quad i = 1, 2, \ldots, m \tag{2.24}$$

As opposed to the *ideal* sliding mode in continuous time, for discrete sliding mode, trajectories are allowed to lie within a *boundary* of the surface $S = 0$. Although this boundary layer is necessary to counteract chattering as noted in Section 2.1, true sliding mode in discrete time systems is unobtainable. The motion inside the boundary layer for discrete time systems is refered to as *quasi-sliding mode* because trajectories never lie exactly on $S = 0$. This quasi-sliding motion of discrete time sliding mode control can be directly attributed to the discrete time interpretation of continuous time Lyapunov stability theory.

Milosavljevic, [13] first commented on the limitations of true sliding mode in discrete time. Since that time several researchers have investigated many aspects of DVSC, (Su [14], Pieper [15], Paden [16], Furuta [17]). Research on the stability of DVSC systems is given by Sarpturk, [19], Kotta, [20]. Recently, more practical applications of DVSC have been proposed including DVSC schemes using state estimators (Misawa, [1]).

2.3 OBDVSC with Disturbance Observer

An extension of the observer based discrete variable structure control work of Misawa, [1] is given by Tang, [7]. Tang covers necessary OBDVSC theory to include an extra disturbance compensation ability for disturbances at the plant input under a matching condition. Two main components, (*i.e.*, a prediction observer and a discrete variable structure controller (DVSC)), make up the OBDVSC compensation scheme illustrated by Figure 2.8. The DVSC portion of the compensator is assigned

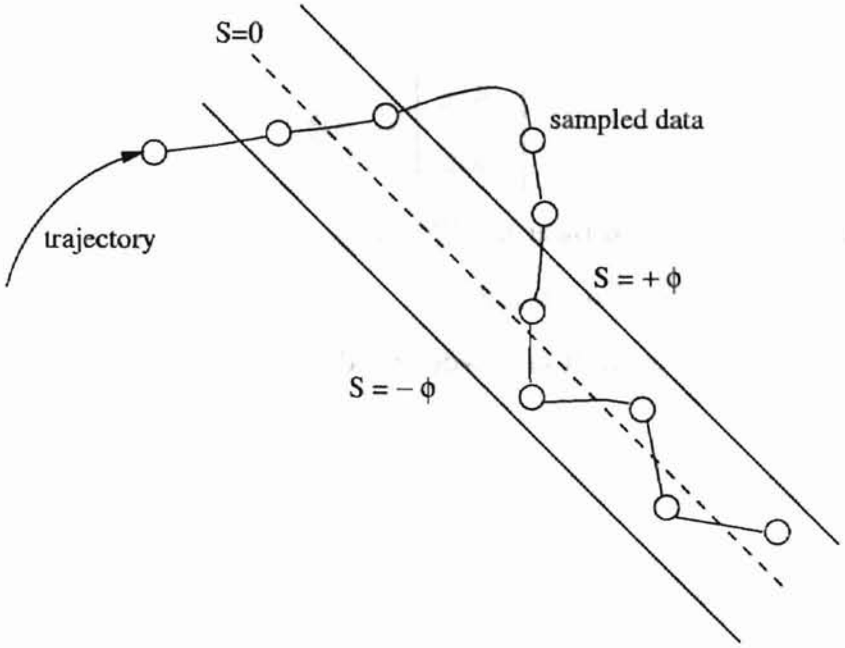

Figure 2.7: Quasi Sliding Mode in Discrete Time, [11].

the task of tracking or regulation depending on the desired control objective. The functionality of the observer is two fold. First, the observer provides necessary state estimates, $\hat{x}(k)$, to the DVSC and secondly, the observer also provides a disturbance estimate for feedforward disturbance compensation. To accomplish the tasks of supplying both state and disturbance estimates, the observer in Figure 2.8 is based on augmented dynamics including both mechanical and disturbance models.
Tang [7] gives a thorough description this disturbance observer, or so called prediction observer with uncertainty estimation, implemented within an OBDVSC framework.

The disturbance observer of Figure 2.8 is implemented by defining an augmented state vector as Equation (2.25), where the plant states are $x(k)$ and the disturbance state $d(k)$ is taken as an external disturbance at the plant input (Franklin *et.al.*, [10]).

$$\xi(k) = \begin{bmatrix} x(k) \\ d(k) \end{bmatrix} \tag{2.25}$$

From augmented dynamics of Equation (2.25), it is straightforward to compute an observer design assuming *a priori* knowledge of a disturbance model. The following section describes two types of disturbance observers that may fit into the augmented dynamics approach of Equation (2.25).

2.3.1 Bias Estimator

Given a discrete plant state space quadruple $\Sigma_d(A, B, C)$ and assuming $d(k+1) = d(k)$, using the augmented state vector of Equation (2.25) the augmented dynamics become Equation (2.26)[4],

$$\underbrace{\begin{bmatrix} x(k+1) \\ d(k+1) \end{bmatrix}}_{\xi(k+1)} = \underbrace{\begin{bmatrix} A & B \\ 0 & 1 \end{bmatrix}}_{A_{aug}} \underbrace{\begin{bmatrix} x(k) \\ d(k) \end{bmatrix}}_{\xi(k)} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_{aug}} u(k) \tag{2.26}$$

with the output equation,

$$y(k) = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{C_{aug}} \cdot \xi(k) \tag{2.27}$$

Using the augmented state dynamics in Equation (2.26), traditional linear Luenberger observer techniques of the standard form given by Equation (2.28),

$$\hat{\xi}(k+1) = (A_{aug} - H\,C_{aug})\hat{\xi}(k) + B\,u(k) + H\,y(k) \tag{2.28}$$

can estimate the plant's states and disturbance for a suitable augmented observer gain matrix H.

[4] Forming augmented observer dynamics with an augmented state vector such as Equation (2.25) with $d(k+1) = d(k)$ is a special form of disturbance observer called a *bias estimator*.

2.3.2 Disturbance State Modelling

Describing an external disturbance with an appropriate set of differential equations, or difference equations in the discrete sense, leads to a slightly different augmented structure. For instance, a *disturbance state* modelling approach allows a disturbance model based on experimental data with some measurable spectrum to be implemented within the augmented state vector $\xi(k)$, (*i.e.*, using *a priori* knowledge of a disturbance model taken as some appropriate $\Sigma(A_{ds}, B_{ds}, C_{ds}, D_{ds})$ stochastic shaping filter). Suppose the disturbance is modeled by Equation (2.29),

$$x_{ds}(k+1) = A_{ds} \cdot x_{ds}(k) + B_{ds} \cdot w_{ds} \tag{2.29}$$

where the disturbance output equation is given by Equation (2.30).

$$d(k) = C_{ds} \cdot x_{ds}(k) + D_{ds} \cdot w_{ds} \tag{2.30}$$

Taking an augmented state vector as $x_{aug}(k) = [x(k)\ x_{ds}(k)]^T$ and combining quadruple $\Sigma_d(A, B, C)$ with Equations (2.29) and (2.30) yields augmented dynamics of Equation (2.31).

$$\underbrace{\begin{bmatrix} x(k+1) \\ x_{ds}(k+1) \end{bmatrix}}_{\xi_{aug}(k+1)} = \underbrace{\begin{bmatrix} A & B \cdot C_{ds} \\ 0 & A_{ds} \end{bmatrix}}_{A_{aug}} \underbrace{\begin{bmatrix} x(k) \\ x_{ds}(k) \end{bmatrix}}_{\xi_{aug}(k)} + \underbrace{\begin{bmatrix} B & B \cdot D_{ds} \\ 0 & B_{ds} \end{bmatrix}}_{B_{aug}} \underbrace{\begin{bmatrix} u(k) \\ w_{ds} \end{bmatrix}}_{u_{aug}} \tag{2.31}$$

$$y \;=\; \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{C_{aug}} \underbrace{\begin{bmatrix} x \\ x_{ds} \end{bmatrix}}_{\xi(k)} \tag{2.32}$$

The input to the disturbance difference equation, w_{ds} is unknown, thus observers for systems with unknown inputs should be applied. Observers for linear systems with unknown inputs have been investigated by several researchers including Kudva [23], Hou [24], Meditch [25], Wang [26], Yang [27] and Hostetter [29]. A key point of using observers for systems with unknown inputs is that the observability matrix[5] remains full rank. For Equation (2.31), the observability requirement is equivalent to the observability matrix made up of matrices A_{aug} and C_{aug} of Equations (2.31, 2.32) being full rank.

Designing an observer for augmented dynamics given in Equation (2.31) relies on *a priori* knowledge of a set of differential equations describing the disturbance. Obtaining the disturbance model may be left for the designer via experimental results or statistical approximation. Guidelines for choosing an appropriate model to fit experimental data so all significant waveform modes observed in disturbance data are correctly represented see Johnson, [30].

2.3.3 OBDVSC with Feedforward Disturbance Compensation

For the OBDVSC compensator system of Figure 2.8 the disturbance input is assumed under a matching condition such that $x(k+1) = A\,x(k) + B(u(k) + d(k))$, where $x \in \mathbb{R}^n$, u and $d \in \mathbb{R}$, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times 1}$. The total control effort from the compensation system, u is comprised of u_c from the controller and u_d, the disturbance estimate from the augmented observer, such that $u = u_c - u_d$. Given a linear time-invariant plant quadruple, namely $\Sigma_d(A, B, C, D)$, the DVSC control effort u_c is given as Equation (2.33), (Misawa, [1]),

$$u_c \;=\; \tfrac{1}{GB}\left[G(I-A)\hat{x}(k) + G\Delta x_d(k) + Ksat(\tfrac{\hat{S}}{\phi})\right] \qquad (2.33)$$

[5] Recall for linear systems, the observability matrix for $\Sigma(A, B, C, D)$ is defined as $\mathcal{O} = [C\;\, CA\;\, CA^2\;\cdots\;\, CA^{n-1}]^T$.

where matrix G is of appropriate dimension, $x_d(k+1)$ and $x_d(k)$ are desired states and $\hat{x}(k)$ is the estimate of the actual state. Misawa [1] suggests the boundary layer thickness, ϕ is generally chosen as $\phi \geq \gamma + \Delta t\varepsilon$, $\forall \gamma > 0$, where Δt is the sampling period, γ represents a bound on additive uncertainty and ε is an arbitrary positive design constant. $\hat{S}$ is the sliding manifold defined by $\hat{S}{=}G\,[x_d(k) - \hat{x}(k)]$. The $sat(\frac{\hat{S}}{\phi})$ function is defined by Equation (2.34), $\Delta x_d(k)$ is defined by Equation (2.35) and the sliding gain K, is defined by Equation (2.36).

$$sat(\frac{\hat{S}}{\phi}) = \begin{cases} +1, & \hat{S} > \phi \\ \frac{\hat{S}}{\phi}, & |\hat{S}| \leq \phi \\ -1, & \hat{S} < \phi \end{cases} \tag{2.34}$$

$$\Delta x_d(k) = x_d(k+1) - x_d(k) \tag{2.35}$$

$$K = \gamma + 2\Delta t\varepsilon \tag{2.36}$$

Obtaining *acceptable* performance from the OBDVSC compensator scheme is strongly dependent upon the design of sliding surface $\hat{S}(k) = G\tilde{\hat{x}}$ and augmented observer gain, H. For an arbitrary choice of $\hat{S}$ and H, typically a trial and error process is necessary to obtain satisfactory performance.

2.4 Hyperplane Design via LQR

This section is intended to give background information on a DVSC design technique following a LQR hyperplane approach given by Tang [7, 8]. This hyperplane design technique is intended for discrete variable structure control systems as described in Chapter 2.

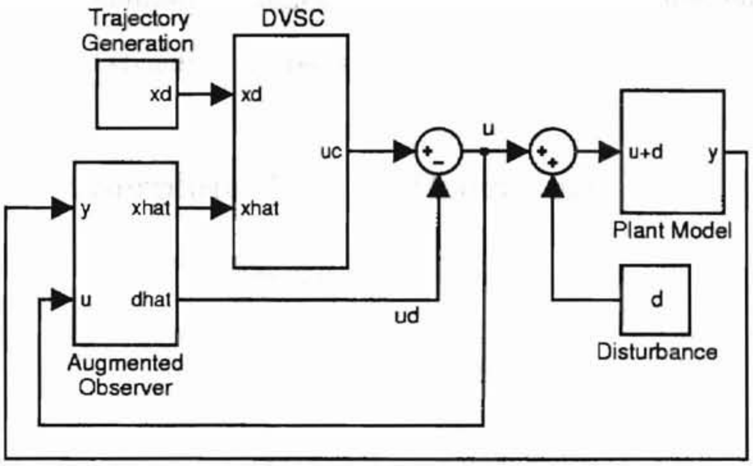

Figure 2.8: OBDVSC Disturbance Compensation Scheme

2.4.1 Eigenvalue Constraint

For single input systems it can be shown that tracking error dynamics inside the sliding boundary layer are given by,

$$A_{eq} = A - \frac{BG}{GB}(A - \alpha I) \tag{2.37}$$

where $\alpha = 1 - \frac{K}{\phi}$ is one real eigenvalue of A_{eq}, (Richter, [6] and Misawa, [21]). It is specifically shown by Tang [7] through an illustrative example that inside the boundary layer ϕ, for systems controlled by Equation (2.33) a necessary condition for desirable behavior on or inside the boundary layer is that one eigenvalue of A_{eq} must be real. This fact essentially constrains α to lie on the Real axis in the complex plane inside the unit disk (*i.e.*, $\alpha = Re[z] \in (-1, 1)$. For a more formal description of the eigenvalue constraint for variable structure systems see Richter [6]. Allowing A_{eq} to strictly have complex valued eigenvalues causes trajectories near the sliding surface to slide along the boundary layer before approaching the origin. If eigenvalues of A_{eq} are strictly complex, then an optimal sliding surface S cannot be obtained, (Furuta, [18]). Thus, α must correspond to a real eigenvalue of A_{eq}, (Richter [6] and Furuta [18]). Restricting A_{eq} to have at least one real eigenvalue is common practice. This requirement is called *eigenvalue constraint* for variable structure systems.

2.4.2 LQR Hyperplane Design Procedure

Letting the A_{eq} matrix describe tracking error dynamics yields

$$\bar{x}(k+1) = A\bar{x}(k) + B\bar{u}(k) \tag{2.38}$$

where taking $\bar{u}(k) = -\frac{G}{GB}(A - \alpha I)\bar{x}(k)$ allows the problem to be cast into a traditional optimal control form where a performance index J given by

$$J = \sum_{k=k_s}^{\infty} \bar{x}^T(k)Q\bar{x}(k) + R\bar{u}^2(k) \tag{2.39}$$

for k_s being the instant a trajectory enters the boundary layer, $Q = Q^T \geq 0$ and $R > 0$,(Tang, [7]). It is well known that $\bar{u}(k) = F\bar{x}(k)$ minimizes the performance index J for

$$F = [R + B^TPB]^{-1}B^TPA \tag{2.40}$$

where $P = P^T > 0$ is the solution to the discrete time algebraic Ricatti equation (DARE) given as

$$A^TPA - P - A^TPB[R + B^TPB]^{-1}B^TPA + Q = 0 \tag{2.41}$$

To satisfy the *eigenvalue constraint* and force Equation (2.38) to behave like a LQR regulator, Tang [7] proposes to solve the so called inverse optimal problem. In this method, a designer must prespecify a real eigenvalue α based on K, Δt, ε, ϕ and also supply a desired weighting matrix Q_d. Several constraint equations given by Equations (2.42, 2.43, 2.44, 2.45, 2.46, 2.47) must also be satisfied.

$$\alpha = (1 - \frac{K}{\phi}) \tag{2.42}$$

$$K = \gamma + 2\Delta t\epsilon \tag{2.43}$$

$$\phi = \gamma + \Delta t\epsilon \tag{2.44}$$

$$\gamma > 0 \tag{2.45}$$

$$\epsilon > 0 \tag{2.46}$$

$$\alpha \in (-1, 1),\ \alpha \neq 0 \tag{2.47}$$

Fixing the control weighting matrix as $R = 1$, an optimization problem must be solved using MATLAB to find a Q closest to Q_d using either a least squares or a convex programming approach. Then using the Q closest to Q_d, the optimal feedback matrix F is found using the **dlqr.m** MATLAB command and the sliding gain matrix G is calculated using $G =$ker$((A - BF - \alpha I)^T)^T$.

Fundamental to the procedure is leveraging use of the symmetric root locus, which stems from the LQR closed loop characteristic equation, given by

$$1 + \frac{1}{R} G^T(z^{-1}) G(z) = 0 \tag{2.48}$$

where $G(z) = C(zI - A)^{-1}B$ using C as a *fictitious* output depending on Q. Simplifying Equation (2.48) by substituting $\Phi(z) = C(zI - A)^{-1}B$, $G(z)$ yields

$$1 + \frac{1}{R} B^T[z^{-1}I - A^T]^{-1} C^T C [zI - A]^{-1} B = 0 \tag{2.49}$$

or,

$$\Phi^T(z^{-1}) Q \Phi(z) = -R \tag{2.50}$$

A key point of the inverse optimal problem given by Tang [7] is that Q_d and an α may or may not be compatible. That is to say, the Q returned via the MATLAB solution for the inverse optimal problem is guaranteed to produce one real eigenvalue of A_{eq} at α. However, fixing $R = 1$ and solving the LQR problem using Q_d does not necessarily guarantee $\lambda_i(A - BF)$ will satisfy the real eigenvalue constraint. Equally important is the scalar relationship between output and control weighting matrices, Q and R respectively, that can clearly be seen in Equation (2.50).

The scalar relationship between Q and R is necessary for the combination of LTR[6] ideas with LQR hyperplane design method of Tang [7]. Because Tang [7] fixes the control weight as $R = 1$, the recovery mechanism must be placed on Q_d. This issue is further discussed in Chapter 4.

2.5 Design Example: OBDVSC with Disturbance Observer

The following section covers a design example implementing background OBDVSC material covered in Chapter 2. Key points for the example include (1) hard disk drive model, (Goh *et. al.*, [50]) (2) discrete variable structure control with feedforward disturbance compensation, (Section 2.3) (3) Compensator based on separate iteratively tuned observer and controller designs.

2.5.1 Disk Drive Model

An ideal mechanical model for a disk drive which maps the voice coil motor voltage input into output position is $G_{plant}(s) = \frac{1}{Js^2}$, where J is the actuator inertia. This *double integrator* model is simplistic because of the neglected higher frequency resonance modes of the actuator arm and low frequency bearing and pivot frictional effects. A more realistic model for the VCM actuator which accounts for high frequency information is given in Equation (2.51), (Goh *et. al.*, [50]). A similar

[6] Loop transfer recovery (LTR) methodology is discussed in detail in Chapter 3.

model in frequency domain for a VCM actuator is given by Lee *et. al.* [51].

$$G_{plant} = \frac{4.3817 \times 10^{10}s + 4.3247 \times 10^{15}}{s^2(s^2 + 1.5962 \times 10^3 s + 9.7631 \times 10^7)} \tag{2.51}$$

The state space representation of Equation (2.51) in controller canonical from is given by $\dot{x} = A_p x + B_p u$, $y = C_p x + D_p u$ for the state space matrices given as Equations (2.52, 2.53, 2.54, 2.55).

$$A_p = \begin{bmatrix} -1.596 \cdot 10^3 & -9.763 \cdot 10^7 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.52}$$

$$B_p = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \tag{2.53}$$

$$C_p = \begin{bmatrix} 0 & 0 & 4.382 \cdot 10^{10} & 4.325 \cdot 10^{15} \end{bmatrix} \tag{2.54}$$

$$D_p = [0] \tag{2.55}$$

Discretizing the continuous state space matrices assuming an arbitrary sampling rate of $t_s = 40 \cdot 10^{-6}$ seconds, yields Equations (2.56, 2.57).

$$A = \begin{bmatrix} 8.6 \cdot 10^{-1} & -3.6 \cdot 10^3 & 0 & 0 \\ 3.7 \cdot 10^{-5} & 9.2 \cdot 10^{-1} & 0 & 0 \\ 7.7 \cdot 10^{-10} & 3.8 \cdot 10^{-5} & 1 & 0 \\ 1.0 \cdot 10^{-14} & 7.8 \cdot 10^{-10} & 4 \cdot 10^{-5} & 1 \end{bmatrix} \tag{2.56}$$

$$B = \begin{bmatrix} 3.774 \cdot 10^{-5} \\ 7.731 \cdot 10^{-10} \\ 1.042 \cdot 10^{-14} \\ 1.047 \cdot 10^{-19} \end{bmatrix} \tag{2.57}$$

Example 2.1 *OBDVSC and Disturbance Observer: Given the discrete disk drive design plant $\Sigma_d(A, B, C, D)$ from Section 2.5.1 and the disturbance model outlined in Section 2.5.2 compare the bias estimator to the state modelling disturbance observer within the OBDVSC compensation scheme of Section 2.3.*

2.5.2 Disturbance Model

Designing an observer for augmented dynamics given in Equation (2.31) relies on *a priori* knowledge of a set of differential equations describing the disturbance. Obtaining the disturbance model may be left for the designer via experimental results or statistical approximation, (Johnson [30]). A disturbance model is taken as a transfer function mapping a random white noise input into disturbance at the plant input. Figure 2.9 illustrates generation of the model-based disturbance using this idea. The magnitude and phase response of the model-based disturbance transfer function, $H(s) = \frac{P(s)}{Q(s)}$, are illustrated in Figure 2.10. The transfer function is designed with appropriately placed poles and zeros in the form of Equation (2.58).

$$H(s) = \frac{K_a \cdot (s^2 + 2\zeta_{z1} w_{z1} s + w_{z1}^2)}{(s^2 + 2\zeta_{p1} w_{p1} s + w_{p1}^2)(s^2 + 2\zeta_{p2} w_{p2} s + w_{p2}^2)} \tag{2.58}$$

where gain K_a given by Equation (2.59) guarantees 0 dB magnitude at low frequencies.

$$K_a = \frac{w_{p1}^2 w_{p2}^2}{w_{z1}} \tag{2.59}$$

The frequencies $(w_{p_{1,2}}, w_{z_1})$ and damping ratios $(\zeta_{p_{1,2}}, \zeta_{z_1})$ are given in Table 2.1 for a disturbance model emphasizing frequencies near 600 and 1200 Hz. Assuming a spindle angular velocity of 7200 RPM, the fundamental spindle frequency is located at 120 Hz, hence the emphasis of frequencies near 600 and 1200 Hz, which are multiples of 120 Hz. Substituting in values for damping ratios and natural frequencies in units of $\frac{rad}{sec}$ from Table 2.1, a numerical representation of Equation (2.58) may be found. Finding equivalent state space matrices and using a

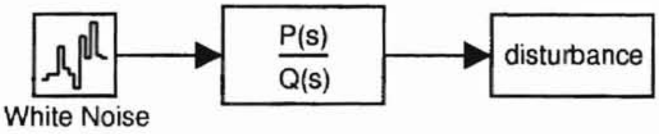

Figure 2.9: Disturbance Generation using Random Input.

sampling rate of $T_s = 40e^{-6}$ seconds yields discrete matrices (A_{ds}, B_{ds}, C_{ds}) with $D_{ds} = [0]$ for the disturbance difference Equation (2.29):

$$A_{ds} = \begin{bmatrix} 9.2 \cdot 10^{-1} & -3.3 \cdot 10^{3} & -8.2 \cdot 10^{5} & -2.9 \cdot 10^{10} \\ 3.9 \cdot 10^{-5} & 9.3 \cdot 10^{-1} & -1.3 \cdot 10^{1} & -5.8 \cdot 10^{5} \\ 7.9 \cdot 10^{-10} & 3.9 \cdot 10^{-5} & 9.9 \cdot 10^{-1} & -7.9 \cdot 10^{0} \\ 1.1 \cdot 10^{-15} & 7.9 \cdot 10^{-10} & 4.0 \cdot 10^{-5} & 9.9 \cdot 10^{-1} \end{bmatrix} \tag{2.60}$$

$$B_{ds} = \begin{bmatrix} 3.9 \cdot 10^{-5} & 7.9 \cdot 10^{-10} & 1.1 \cdot 10^{-14} & 1.1 \cdot 10^{-19} \end{bmatrix}^T \tag{2.61}$$

$$C_{ds} = \begin{bmatrix} 0 & 5.2 \cdot 10^{7} & 9.5 \cdot 10^{10} & 7.5 \cdot 10^{14} \end{bmatrix} \tag{2.62}$$

Assuming a perfect disturbance model raises a robustness question related to the disturbance modelling augmentation structure. For simulations, a disturbance model mismatch is assumed. Figure 2.10 illustrates the mismatch in disturbance model in frequency domain. The curves labeled *actual* represent the transfer function used to input the actual disturbance into the system, while the curves labeled *disturbance state* represent the transfer function used to design the augmented observer.

Table 2.1: Frequency and Damping Ratios of Disturbance Model Transfer Function.

Poles				Zeros	
w_{p1} (Hz)	w_{p2} (Hz)	ζ_{p1}	ζ_{p2}	w_{z1} (Hz)	ζ_{z1}
600	1200	0.008	0.006	750	0.2

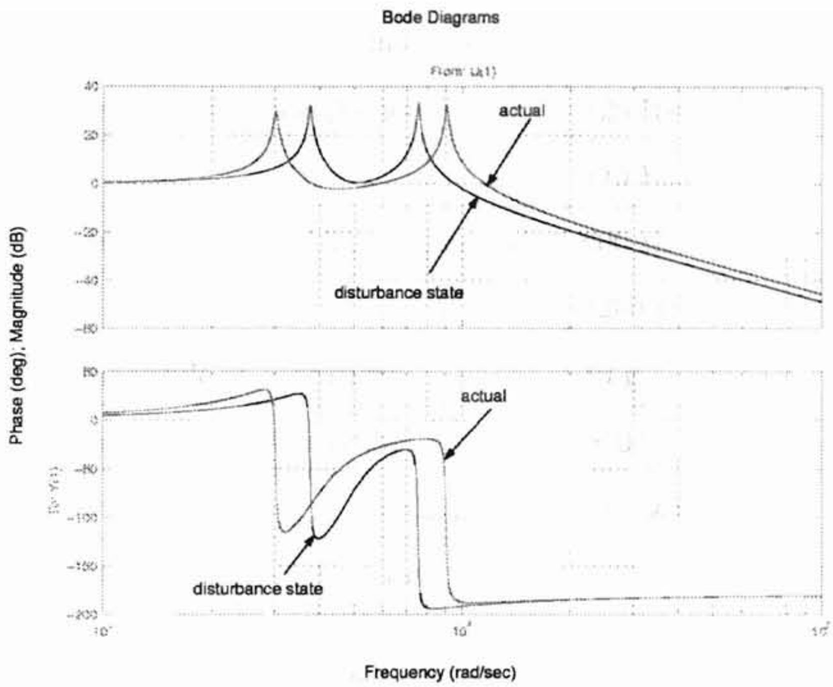

Figure 2.10: Magnitude & Phase of the Disturbance Model Transfer Function.

2.5.3 OBDVSC with Disturbance Observer Simulations

Two main sets of simulations[7] were ran to test the augmented observer structures within the OBDVSC scheme. Both sets of simulations considered track following performance with a disturbance at the plant input. The first set of simulations considered the disturbance to be made up a fictitious bias and vibration, while the second set of simulations considered the input disturbance as discussed in Section (2.5.2). The two augmented observers were simulated under conditions given in Table (2.2).

Under the bias and vibration disturbance condition, both the bias estimator and disturbance state augmentation structures responses are illustrated in Figure 2.11. The bias estimator augmented observer has the best regulation performance, less than 1% error, and almost an exact disturbance estimate. Although the bias

[7] Simulations are in discrete time, although discrete samples are not specifically shown.

Table 2.2: Simulation Conditions.

Freq. of Vibration (Periodic)	120 Hz
	240 Hz
Freq. Emphasis of Vib. (Random)	600 Hz
	1200 Hz
Magnitude of Vibrations	10 g
Magnitude of Bias (volts)	± 0.04
Sampling time	40 μsec
Track follow criteria, (track width)	± 5%

estimator is based on a constant model, (*i.e.*, $x_{bias}(k+1) = x_{bias}(k)$), a time varying disturbance can be estimated. The disturbance state augmented observer also captures the disturbance and is able to keep off track performance within 2.8% of the regulation objective. Allowing the OBDVSC to solely counteract the input bias and vibration disturbance results in nearly 5% tracking error.

Simulating the OBDVSC scheme with a disturbance given by Section (2.5.2) for both the bias estimator and disturbance state augmentation structures produces responses illustrated in Figure 2.12 and Figure 2.13. The regulation task is most accurately accomplished using the disturbance modeling augmented structure combined with OBDVSC, within 0.7% of a track width, even with mismatched disturbance models. From Figure 2.12, the bias estimator augmented structure produces a position response that is worse than not supplying a feedforward compensation term, 3% compared to 1.5% off track performance. The degradation in performance for the bias augmented observer OBDVSC scheme can be attributed to the disturbance estimate illustrated in Figure 2.13. Like the disturbance modeling augmented observer, the bias observer is again capable of capturing the input disturbance's general shape. However, the bias estimator fails to closely

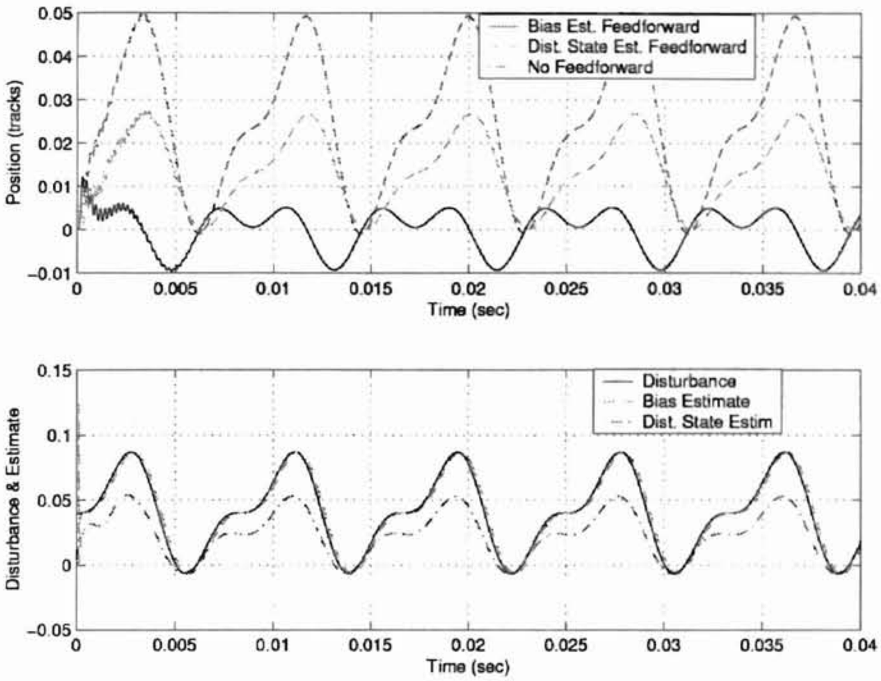

Figure 2.11: Position Response and Disturbance Estimation for Bias plus Vibration Disturbance Input.

maintain the phase information of the disturbance signal which adds extra delay to the feedforward compensation ultimately resulting in poorer tracking performance. Simulating the system with a combined disturbance of both bias and vibration as well as the disturbance described in Section (2.5.2) produces disturbance estimation errors in Figure 2.14, where $Error_1 = total\ disturbance - bias\ disturbance\ estimate$, and $Error_2 = total\ disturbance - disturbance\ modeling\ estimate$. $Error_2$ is approximately 50% of that of $Error_1$.

2.6 Summary

Chapter 2 gives a review of variable structure control systems for both continuous and discrete time. Important points from Chapter 2 to remember include:

- Variable structure systems are Lyapunov based techniques which possess the

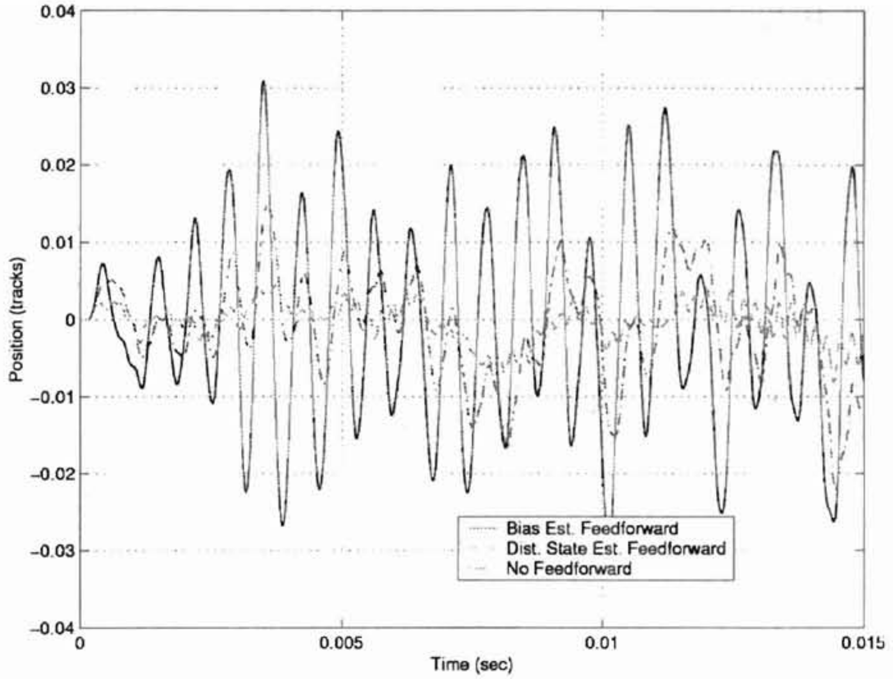

Figure 2.12: Position Response for Model Based Disturbance Input.

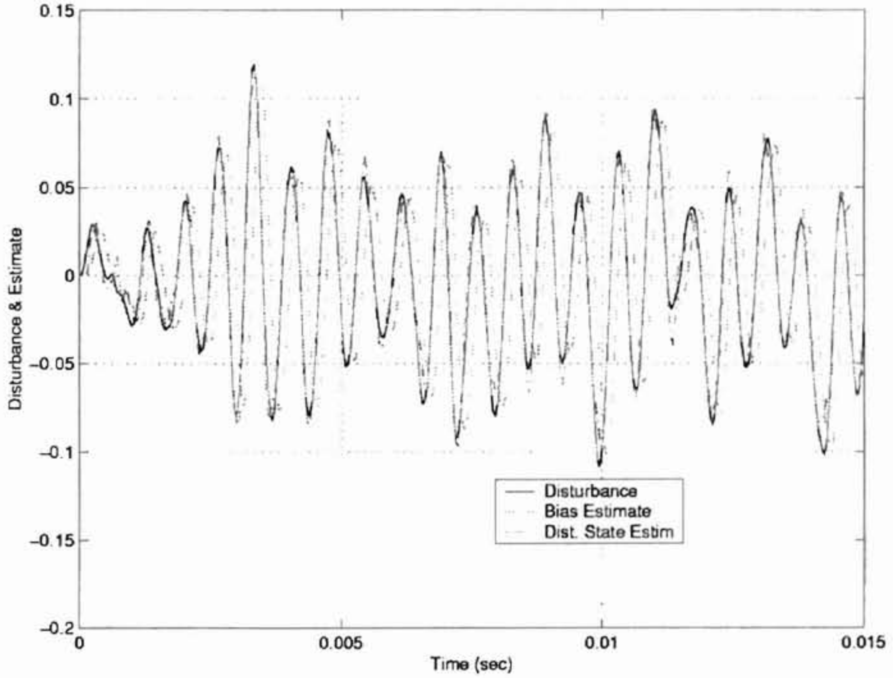

Figure 2.13: Disturbance Estimates for Model Based Disturbance Input.

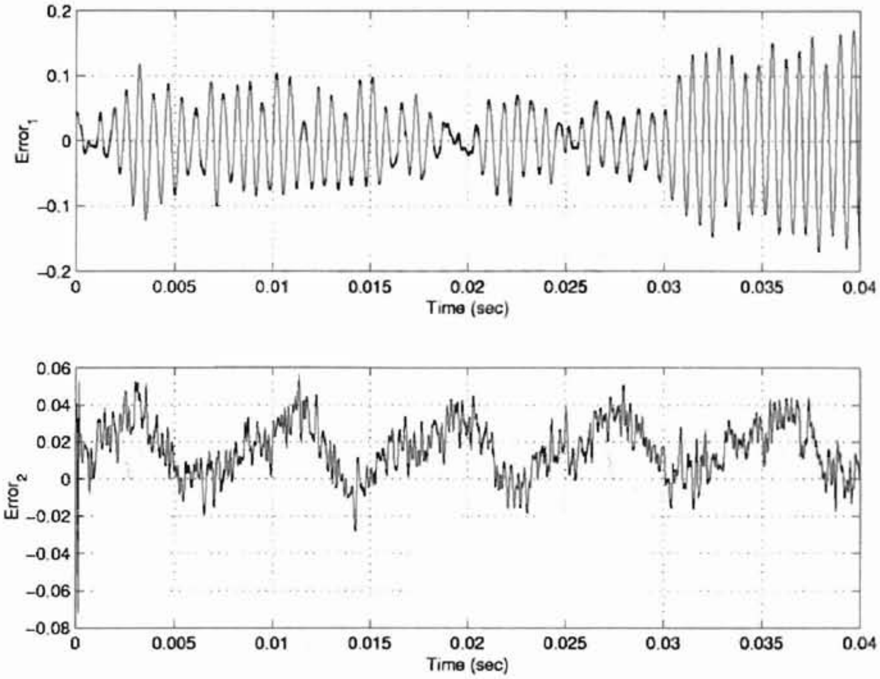

Figure 2.14: Estimate Error of Bias Observer and Disturbance State Modelling Observer.

ability to *change* control structure.

- Lyapunov theory for continuous and discrete time is different ultimately negating a true *sliding* mode in discrete time. Implementing a boundary layer about the sliding manifold to address chattering and using the discrete time sliding condition yields *quasi* sliding motion inside the boundary layer.

- A DVSC hyperplane design technique via LQR exists, (Tang, [7]) which allows a designer to satisfy mild system constraints ultimately yielding a sliding surface, $\acute{S}$.

- Observer based discrete time variable structure control can be implemented for practical applications, (Section 2.5).

Chapter 3

Loop Transfer Recovery Overview

Practically speaking no matter how powerful a specific control methodology is, several iterations are generally necessary to tune parameters for an acceptable level of performance. Providing a systematic means to minimize trial and error iterations would be beneficial for any practical control applications that require an iterative tuning process. Loop transfer recovery (LTR) methodologies are such tools which offer means for control engineers to tweak design parameters in an *educated* way effectively reducing the number of iterative steps.

This chapter covers basic concepts of the LTR methodology, specifically highlighting a technique called Linear Quadratic Gaussian Loop Transfer Recovery (LQG/LTR). First an overview of the basic ideas and properties of LTR is given in a continuous time setting. Also, loop shaping ideas for the target loop are discussed. Next the LTR methodology is covered in a broader sense encompassing discrete time systems using full order observers. Chapter 3 ends with a discussion of the recovery error matrix, a measuring tool for loop transfer recovery for both continuous and discrete time systems which will be useful for measuring the success of LTR hyperplane design theory in Chapters 5 and 6.

3.1 LQG/LTR

The fundamental idea for *recovery* driving the LTR mechanism is accredited to Doyle and Stein [42], although a prior work also by Doyle and Stein [41] alludes to the LTR concept. Since that time, LTR ideas have become a popular modern design technique for MIMO continuous and discrete time systems with some modest restrictions on design plant characteristics.

One pictorial interpretation of the LTR design procedure is cast graphically beginning with the servo control block diagram depicted in Figure 3.1. Suppose a continuous time linear time-invariant, either MIMO or SISO, system exists such that $\Sigma(A, B, C)$ is a state space representation of a minimum phase[1] design plant to be controlled. Let $G(s)$ in Figure 3.1 define the transfer function given by Equation (3.1),

$$G(s) = C\,\Phi(s)\,B \tag{3.1}$$

such that $\Phi(s)$ is taken as Equation (3.2).

$$\Phi(s) \triangleq (sI - A)^{-1} \tag{3.2}$$

Assuming that (A, B) is stabilizable and (A, C) is detectable[2], the LTR methodology seeks to define the MIMO compensator $K(s)$, (*i.e.*, controller/observer pair), such that desired stability, robustness and performance specifications are met (Athans [43]).

Typically for the LTR methodology in continuous time, a linear state feedback controller of the form $u(t) = -F\,\hat{x}(t)$ is used in tandem with a Luenberger state

[1] A design plant $G(z) = C(zI - A)^{-1}B$ is said to be minimum phase for $\Sigma_d(A, B, C, D)$ if all zeros of $G(z)$ are contained in $\mathbb{C}^{\odot}$. Similarly, a continuous time design plant $G(s)$, for $\Sigma(A, B, C, D)$, is said to be minimum phase if all the zeros of $G(s)$ are in $\mathbb{C}^-$

[2] See Zhou, [33] for definitions for stabilizable and detectable

estimator, where $\hat{x}(t)$ is an estimated state vector and F defines a controller gain matrix. Implementing this procedure requires the design of an observer gain matrix, H, and a state feedback gain matrix, F. Suppose the observer and state feedback control design are handled separately, one may design an observer using Kalman filtering techniques and then separately design a controller using linear quadratic regulator (LQR) techniques. Each individual design of H and F could be thought of as optimal, however when implemented together as a compensator, the two optimal parts may function in a sub-optimal manner. An arbitrary combination of observer and controller designs can be "arbitrarily bad", (Doyle and Stein [41]). The LTR method seeks to alleviate the arbitrary combination of observer and controller pairs in a way to find gains, H and F, that when collectively used give good stability margins, performance characteristics and robustness properties.

There are three major steps in the LTR methodology:

1. Given a design plant, first characterize design requirements as restrictions on the singular values[3] of an open loop transfer function matrix formed by breaking the control loop in Figure 3.1 at the input or output of the plant $G(s)$.

2. Next design a *target loop* to meet specifications outlined by step (1) with the intention of implementing a compensator composed of state feedback control and a state estimator. For example, breaking the control loop at the plant output neglecting disturbances d_i and d_o of Figure 3.1, the target loop would then be given as the open loop equivalent of Figure 3.2. Matrix H of appropriate dimension would be called the *filter gain matrix*[4]. The target loop in this situation is refered to as the *target filter loop* because it consists designing an observer matrix H.

3. Staying consistent with the case of breaking the control loop at the plant output, the last step is to hold H found in step (2) constant and to *recover*

[3] See Zhou, [33] for a general discussion of singular values and singular value decomposition.

[4] A dual procedure also exists for breaking the control loop at the plant input where the target loop is designed using a state feedback control.

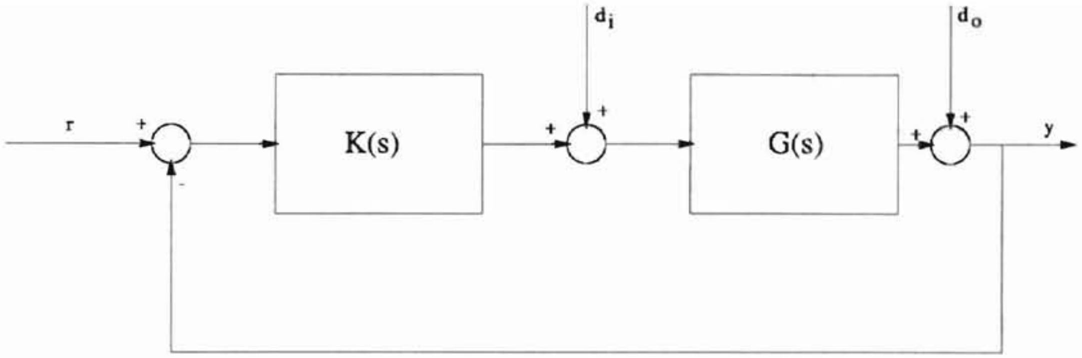

Figure 3.1: Typical Feedback Control System.

the target filter loop characteristics in the control loop using compensator $K(s)$ of Figure 3.1 in a special way such that the performance of control loop approximates that of the target loop.

One key flexibility in the LTR method is that any technique which yields a target loop in step (2) satisfying properties desired in step (1) is valid. Fulfilling step (2) of the above procedure essentially becomes that of observer design, or the selection of gain matrix H. The design of the target filter loop is an arbitrary choice for the designer. Because of the stability margins and robustness properties intrinsic to optimal control, often a Kalman-Bucy filtering technique is used to design H. When a Kalman-Bucy filter is used for the design of H along with the recovery technique of step (3), the composite technique is commonly refered to as linear quadratic Gaussian loop transfer recovery, or LQG/LTR. The LQG/LTR compensator technique belongs to a broader class of compensators known as *model based compensators*, described by Figure 3.3.

One fundamental question remains: How does the LTR recovery mechanism in step (3) work? The answer lies in the following, given a SISO continuous system, $\Sigma(A, B, C, D)$, the Linear Quadratic Regulator (LQR) problem seeks to minimize a

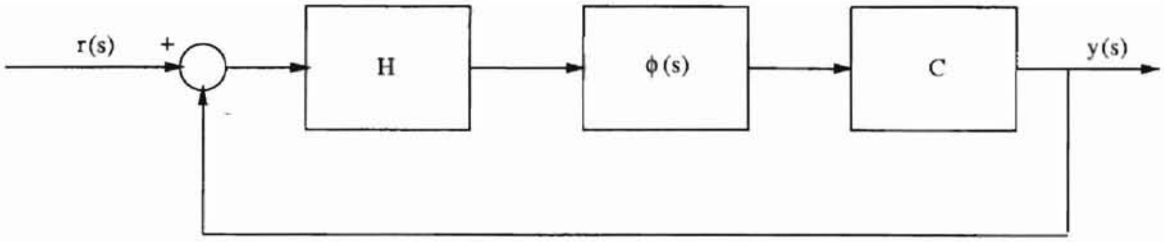

Figure 3.2: Target Loop for Loop Transfer Recovery (LTR) at Plant Output.

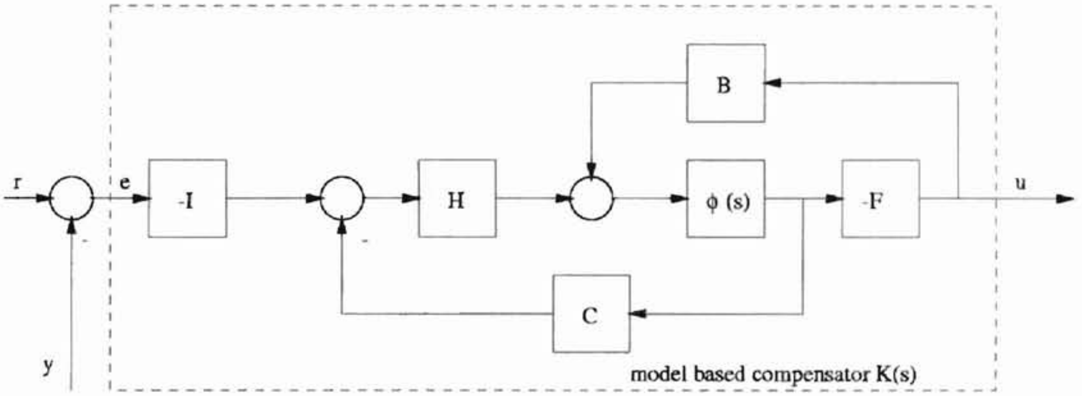

Figure 3.3: Model Based Compensator, $K(s)$.

performance index,

$$J = \sum_{t=0}^{t=\infty} x^T Q x + R u^2 \tag{3.3}$$

for weighting terms $Q = Q^T > 0$ and $R > 0$.

A special case of the LQR problem known as cheap control exists when the control weight R approaches zero. The is equivalent performance index is given by Equation (3.4).

$$J_{cheap} = \sum_{k=0}^{k=\infty} x^T(k) Q x(k) + \rho R u^2 \tag{3.4}$$

For step (3) of the major steps in LTR methodology to be valid, the following condition must hold.

$$lim_{\rho \to 0} \sqrt{\rho} F = WC,\ W^T W = I$$

Lemma 3.1 *For recovery at the plant output, given $\Sigma(A, B, C)$ and the continuous-time model based compensator of Figure 3.3 as*

$K(s) = F(sI - A + BF + HC)^{-1}H$ where $G(s) = C(sI - A)^{-1}B$ is minimum phase, if conditions $1 \rightarrow 3$ are valid, where

1. *$Re[\lambda_i(A - BF)] < 0$*
2. *$Re[\lambda_i(A - HC)] < 0$ and*
3. *$lim_{\rho \to 0} \sqrt{\rho} F = WC$, $W^T W = I$ (cheap control)*

then

$$lim_{\rho \to 0} K(s) = [C(sI - A)^{-1}B]^{-1}[C(sI - A)^{-1}H]. \tag{3.5}$$

point wise in s.
Proof. See discussion in Doyle and Stein [42].

For recovery at the plant output, Lemma 3.1 and solving the cheap control LQR problem suggest as $\rho \to 0$ for minimum phase systems, the LTR mechanism replaces the design plant dynamics with the dynamics of the target loop. The main LTR result for recovery at the output of the design plant is given by the following Lemma, (Doyle and Stein [42]):

Lemma 3.2 *If conditions $1 \rightarrow 3$ of Lemma 3.1 are valid for the given continuous-time LTI system $\Sigma(A, B, C)$, then it follows that:*

$$lim_{\rho \to 0} G(s)K(s) = \underbrace{C(sI - A)^{-1}B}_{G(s)} \cdot \underbrace{[C(sI - A)^{-1}B]^{-1}C(sI - A)^{-1}H}_{lim_{\rho \to 0} K(s)} \tag{3.6}$$

so that

$$lim_{\rho \to 0} G(s)K(s) = C(sI - A)^{-1}H \tag{3.7}$$

Proof. Simple substitution of result taken from Lemma 3.1into $G(s)K(s)$.

3.1.1 Target Filter Design

For this thesis, recovery at the plant output is assumed to be the design objective. In that light, following steps (1) and (2) of the procedure outlined in Section 3.1 coincides mainly with the design of a target filter loop, or observer design. For the continuous time MIMO case, Athans [43] gives several useful hints for designing the target feedback loop for recovery at the plant output based upon Kalman's frequency domain equality [42, 44]. For MIMO systems, the target loop design consists of suitably shaping the singular values to fit design specifications. For SISO systems, shaping the singular value plots is equivalent to shaping the Bode magnitude plot. The target filter loop design is an important part of the LTR procedure that will be necessary for hyperplane design of Chapters 4 5.

3.1.2 Target Filter Loop Design Methods

A common design requirement is for a system to have zero steady state error to step changes in set point reference. This requirement is satified by type 1 systems[5]. Thus, it is a common practice to augment state dynamics with free integrators during the loop shaping phase of the LQG/LTR procedure. Suppose plant dynamics are given by $\Sigma(A, B, C, D)$, augmenting each channel of a MIMO system with a free integrator is achieved by redefining the state vector as $x_{aug} = [u_p \; x]^T$, where $\dot{u}_p(t) = u(t)$ or $u_p(s) = \frac{1}{s}u(s)$. Defining the state in this way allows the augmented dynamics to be rewritten as

$$\underbrace{\begin{bmatrix} \dot{u}_p(t) \\ \dot{x}(t) \end{bmatrix}}_{\dot{x}_{aug}(t)} = \underbrace{\begin{bmatrix} 0 & 0 \\ B & A \end{bmatrix}}_{A_d} \underbrace{\begin{bmatrix} u_p(t) \\ x(t) \end{bmatrix}}_{x_{aug}} + \underbrace{\begin{bmatrix} I \\ 0 \end{bmatrix}}_{B_d} u(t) \tag{3.8}$$

$$y(t) = \underbrace{\begin{bmatrix} 0 & C \end{bmatrix}}_{C_d} \underbrace{\begin{bmatrix} u_p(t) \\ x(t) \end{bmatrix}}_{x_{aug}} \tag{3.9}$$

[5] Type 1 systems exhibit an integral action synonymous with containing a free integrator per input channel.

Starting with the target filter loop using the augmented dynamics $\Sigma(A_d, B_d, C_d)$ given as,

$$G_{kf}(s) = C_d(sI - A_d)^{-1}H \tag{3.10}$$

it is suggested by Athans [43] following the original procedure by Doyle and Stein [42], that solving a fictitious continuous time Kalman filter problem often leads to suitable solution for observer gain matrix H. Let $\zeta(t)$ be process noise characterized as white, zero mean with an identity (I) noise intensity matrix, and at the same time let $\theta(t)$ be white, zero mean with noise intensity matrix μI. Given the stochastic system using augmented dynamics from Equations (3.8, 3.9),

$$\dot{x}(t) = A_d x(t) + L\zeta(t) \tag{3.11}$$

$$y(t) = C_d x(t) + \theta(t) \tag{3.12}$$

it is well known that the solution of the Kalman filtering problem is given by an observer matrix selected as

$$H = (\frac{1}{\mu})\Sigma C_d^T \tag{3.13}$$

where Σ is the symmetric positive definite solution of the filter algebraic Ricatti equation (FARE)

$$0 = A_d\Sigma + \Sigma A_d^T + LL^T - (\frac{1}{\mu})\Sigma C_d^T C_d \Sigma \tag{3.14}$$

where $\mu > 0$ and L can be used as design parameters to "shape" $C_d(sI - A_d)^{-1}H$, which is guaranteed to be nominally stable due to Kalman filtering theory assuming that $[A_d, L]$ is stabilizable and $[A_d, C]$ is detectable.

Selection of L can be arbitrary or may be chosen from the frequency domain equality (FDE), which may be derived using the FARE and $C_d(sI - A_d)^{-1}H$. Define $L \triangleq [L_{low}\ L_{high}]^T$, where L_{low} and L_{high} correspond to low and high frequency. From the frequency domain equality given as,

$$\sigma_i[I + G_{kf}] = \sqrt{1 + (\frac{1}{\mu})\sigma_i^2[C_d(sI - A_d)^{-1}L]} \tag{3.15}$$

looking at the matrix $C_d(sI - A_d)^{-1}L$ reveals conditions sufficient to match singular values at low and high frequencies. Athans [43] suggests selecting L_{low} and L_{high} as Equations (3.16, 3.17), given that A^{-1} exists and using μ to govern the crossover frequency (w_{cof}) as Equation (3.18).

$$L_{low} = -[CA^{-1}B]^{-1} \tag{3.16}$$

$$L_{high} = C^T(CC^T)^{-1} \tag{3.17}$$

$$\mu = (\frac{1}{w_{cof}})^2 \tag{3.18}$$

The choice of L from Equations (3.16, 3.17) matches singular values only at low and high frequencies, however an enhanced MBC/LTR target loop design suggested by O'Dell [46] matches singular values across all frequencies by choosing L as Equation (3.19).

$$L = \begin{bmatrix} L_{low} \\ L_{high} \end{bmatrix} = \begin{bmatrix} -[CA^{-1}B]^{-1} \\ A^{-1}B(CA^{-1}B)^{-1} \end{bmatrix} \tag{3.19}$$

If A is non-invertible, then a practical solution is to artificially place fast poles within A making a numerical inverse.

3.2 Discrete Time LQG/LTR

Many authors have extended the seminal work of Doyle and Stein [41, 42] for loop transfer recovery in both continuous and discrete time. Similar to the extension of sliding mode control into discrete time, the sampling process defeats a direct comparison between continuous and discrete time LQG/LTR methodology. Despite any limitations brought about by sampling, many authors have shown discrete time LTR theory to be a viable design technique. For discrete time LTR, Maciejowski [31] shows recovery is possible for systems that are minimum phase if cheap control is applied using a current observer. Maciejowski also states that exact recovery is generally not possible for systems using a predicting observer or for non-minimum phase systems although generally a useful degree of recovery (*i.e.*, generally surpassing system bandwidth) is usually obtained. Zhang *et. al.* [32] discusses discrete time LTR for non-minimum phase systems using prediction and filtering (current) observers. Tadjine *et. al.* [34] addresses discrete time loop transfer recovery at the plant input and output using the so called delta operator formulation. Ishihara *et. al.* [35] investigates the role of current and prediction estimators in discrete LTR. Direct applications of discrete time LTR have also been reported in the literature. Lopez *et. al.* [36] uses discrete LQG/LTR for control of a ship steering autopilot. Microactuator technology in the disk drive industry has sparked research and application of discrete LTR, [37, 38, 39].

3.2.1 LTR: Continuous vs. Discrete

The general procedure for the LTR mechanism of discrete time systems is similar to continuous time procedure given in Section 3.1 with a few modifications. One major difference between continuous and discrete time LTR theory lies in observer selection described in Section 3.2.2. Another fundamental difference between continuous and discrete time LTR theory lies in the definitions for stability. For continuous time systems, conditions 1 → 3 of Lemma 3.1 requires $Re[\lambda_i(A-BF)] < 0$, $Re[\lambda_i(A-HC)] < 0$ and $lim_{\rho \to 0} \sqrt{\rho} F = WC, \forall W^T W = I$. A discrete equivalent for conditions 1 → 2 would be $\lambda_i(A-BF) < \mathbb{C}^{\odot}$ and $\lambda_i(A-HC) < \mathbb{C}^{\odot}$ as stability requirements for discrete-time linear systems suggests.

For example, suppose it is desired to perform recovery at the plant output for a discrete system $\Sigma_d(A, B, C, D)$. The procedure for LTR in discrete time parallels that given in Section 3.1 in that the procedure would follow three basic steps; (1) formulate design specifications into restrictions on singular value plots by breaking the control loop at the plant output, (2) design target filter loop. (3) holding the observer gain matrix found in step (2) constant, perform recovery using LQR with cheap control.

For continuous time systems, solving the LQR problem guarantees infinite gain margin and a minimum of 60 degrees of phase margin. Because the return difference equality[6] differs between continuous and discrete time systems, robustness results (*i.e.*, gain and phase margin bounds) for discrete LQR are less attractive than their continuous counterparts.

3.2.2 Observer Selection for Discrete LTR

As in continuous time systems, using state feedback control in discrete time requires the use of a state estimator to make up for unmeasured states. For discrete time systems, there are two versions of full order state estimators which allow the option to account for the necessary computational time. The discrete observers include: (1) A prediction observer which is based on measurements up to and including $y(k-1)$, which accounts for computational time, and (2) A filtering (or current) observer which is based on measurements up to and including the current measurement $y(k)$, which neglects computational time. Both observers use the output measurements $y(l)$ for $l \leq k$ where k is the current sample time.

[6] For a discussion of the return difference equality for both continuous and discrete time systems see Anderson, [45].

3.2.3 Prediction Observer

Given a discrete system $\Sigma_d(A, B, C)$ the *prediction observer* is described by

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + Bu_k + H_p(y_k - C\hat{x}_{k|k-1}) \tag{3.20}$$

where the observer gain H_p is chosen such that the eigenvalues of $(A - K_pC)$ are designed stable in the discrete sense. The observer based state feedback control law using the prediction observer is given by

$$u_k = -F\hat{x}_{k|k-1} \tag{3.21}$$

3.2.4 Current Observer

On the other hand, the *current observer*, given $\Sigma_d(A, B, C)$ is described by

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + Bu_k + AH_f(y_k - C\hat{x}_{k|k-1}) \tag{3.22}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + H_f(y_k - C\hat{x}_{k|k-1}) \tag{3.23}$$

where the observer gain H_f is chosen such that the eigenvalues of $(A - AK_fC)$ are designed stable in the discrete sense. Similarly, the observer based state feedback control law using the current observer is given by

$$u_k = -F\hat{x}_{k|k} \tag{3.24}$$

When Equation (3.22) is used alone, it is called the *predicting* version of the current observer because it is essentially the prediction observer with a special form of gain matrix, namely $H_p = AH_f$.

3.3 A Tool for Measuring Loop Transfer Recovery

A tool for measuring the level of recovery is given by Goodman [40] as the error recovery matrix based on achieving exact recovery at the plant input or output. Assuming recovery at the plant output, to be consistent with this thesis, starting with the fundamental idea of LTR being the *approximation* of the target loop by the compensator loop of Figure 3.1 as the overall goal (*i.e.*, LTR desires $G(jw)K(jw) \cong C\Phi(jw)H$), an error matrix, $E_o(s)$[7] is developed. Showing LTR for systems that achieve *exact* recovery is equivalent to rendering $E_o(s)$ zero for all frequency. Lemma 3.3 states the main result derived from the output error recovery matrix.

Lemma 3.3 *Let $E_o(s)$ be defined as*

$$E_o(s) \triangleq C\Phi(s)H - G(s)K(s) \tag{3.25}$$

Then

$$E_o(s) = [I + C\Phi(s)H][I + M_o(s)]^{-1}M_o(s) \tag{3.26}$$

where

$$M_o(s) \triangleq C(sI - A + BF)^{-1}H \tag{3.27}$$

Proof, [40] Substitute $K(s) = F(sI - A + BF + HC)^{-1}H$ in definition of $E_o(s)$

$$E_o(s) \triangleq C\Phi(s)H - G(s)K(s) \tag{3.28}$$

$$= C\Phi(s)H - C\Phi(s)BF(sI - A + BF + HC)^{-1}H \tag{3.29}$$

$$= C\Phi(s)[I - BF(sI - A + BF + HC)^{-1}]H \tag{3.30}$$

$$= C\Phi(s)(sI - A + HC)(sI - A + BF + HC)^{-1}H \tag{3.31}$$

$$= \Gamma(s)C(sI - A + BF + HC)^{-1}H \tag{3.32}$$

$$= \Gamma(s)C[I + (sI - A + BF)^{-1}HC]^{-1}(sI - A + BF)^{-1}H \tag{3.33}$$

$$= \Gamma(s)C[I + C(sI - A + BF)^{-1}H]^{-1}C(sI - A + BF)^{-1}H \tag{3.34}$$

$$= \Gamma(s)[I + M_o(s)]^{-1}M_o(s) \tag{3.35}$$

[7] The subscript "o" denotes recovery at the plant output. A subscript "i" denotes recovery at the plant input.

where $\Gamma(s) \triangleq [I + C\Phi(s)H]$ *and the matrix identities*

$$(A+B)^{-1} = (I + A^{-1}B)^{-1}A^{-1}$$

and

$$G(I+HG)^{-1} = (I+GH)^{-1}G$$

were used from Equation (3.32) to Equation (3.33) and from Equation (3.33) to Equation (3.34), respectively.

Three equivalent conditions for exact recovery using Equation (3.35) that are synonymous to the recovery error matrix being rendered zero (*i.e.*, exact recovery) are given by Theorem 3.1, (Goodman [40]).

Theorem 3.1 *Given a non-defective*[8] *matrix A-BF with right eigenvectors* u_i, $1 \leq i \leq n$ *and left eigenvectors* v_i, $1 \leq i \leq n$, *from* $E_o(s)$ *and* $M_(s)$ *of Lemma 3.3, the following conditions are equivalent:*

1. $E_o(s) = 0$
2. $M_o(s) = 0$
3. $Cu_i = 0$ *or* $v_i^* H = 0, \forall\, 1 \leq i \leq n$

Proof Goodman [40], For conditions $1 \Leftrightarrow 2$ *see Goodman [40]. For Condition* $2 \Leftrightarrow 3$, *Goodman [40] suggests letting* λ_i $1 \leq i \leq n$ *be the eigenvalues of (A-BF). Define*

$$\Lambda \triangleq \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \tag{3.36}$$

and

$$U \triangleq \left[\begin{array}{c|c|c|c} u_1 & u_2 & \cdots & u_n \end{array} \right] \tag{3.37}$$

$$V \triangleq \left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array} \right] \tag{3.38}$$

[8] A non-defective matrix is defined as a square matrix having a complete set of independent eigenvectors.

so that U and V are scaled $UV^* = V^*U = I$. It follows that

$$(A - BF) = U\Lambda V^* \tag{3.39}$$

From the definition of $M_o(s)$ using Equations (3.36, 3.37, 3.38)

$$\begin{aligned} M(s) &= C(sI - A + BF)^{-1}H & (3.40)\\ &= C(sUV^* - U\Lambda V^*)^{-1}H & (3.41)\\ &= CU(sI - \Lambda)^{-1}V^*H & (3.42)\end{aligned}$$

which can be written in matrix residue form as

$$M(s) = \sum_{i=1}^{n} \frac{Cu_i v_i^* H}{s - \lambda_i} \tag{3.43}$$

From Equation 3.42 it is trivial to see Condition $2 \Leftrightarrow 3$.

Theorem 3.1 is useful for showing exact recovery of the hyperplane design technique using LTR of Chapters 5 and 6.

3.4 Design Example: LQG/LTR

The following section applies the background information covered in Chapter 3 applied to a remotely pivoted vehicle (RPV). The purpose of the following example is to show the loop transfer recovery design technique in both continuous and discrete time. MATLAB scripts for the design example in discrete time can be found in Appendix B.

3.4.1 Remotely Piloted Vehicle Model

To illustrate the LQG/LTR design methodology, consider the following example based upon an MIMO remotely piloted vehicle (RPV), (Maciejowski *et. al.* [48]). A linear model for the RPV given as a two input, two output, six state representation

$\Sigma(A_p, B_p, C_p, D_p)$ has the following continuous time plant matrices:

$$A_p = \begin{bmatrix} -0.02567 & -36.6170 & -18.8970 & -32.0900 & 3.2509 & -0.76257 \\ 9.257 \cdot 10^{-5} & -1.8997 & 0.98312 & -7.256 \cdot 10^{-4} & -0.1708 & -4.965 \cdot 10^{-3} \\ 0.012338 & 11.720 & -2.6316 & 8.758 \cdot 10^{-4} & -31.6040 & 22.3960 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -30.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & -30.0000 \end{bmatrix}$$

$$B_p = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 30 & 0 \\ 0 & 30 \end{bmatrix}$$

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

$$D_p = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Checking the eigenvalues of A_p reveals an unstable plant with pole locations given in Table 3.1.

Example 3.1 *LQG/LTR: Given the design plant $\Sigma(A_p, B_p, C_p, D_p)$ and design specifications which include a closed loop system bandwidth of $10\frac{rad}{sec}$ with no steady state error for step commands and good rejection of constant disturbances at the plant output. Design a suitable observer/controller pair using LQG/LTR utilizing target filter loop shaping ideas of Section 3.1.1 and Lemma 3.1.*

Table 3.1: Plant Pole Locations for RPV Model.

-30	-30	-5.67	-0.26	$0.69 \pm 0.25j$

Continuous LQG/LTR Example

To begin the continuous time LQG/LTR design procedure for recovery at the plant output, first the target loop is designed based on a crossover frequency of $w_{cof} = 10\,\frac{rad}{sec}$, which yields $\mu = (\frac{1}{10})^2 = 0.01$. To satisfy the zero steady state error condition, the design plant is augmented with integrators as suggested in Section 3.1.1 by Equations (3.8, 3.9) yielding augmented dynamics $\Sigma(A_d, B_d, C_d, D_d)$. To match singular values at low and high frequency, Equations (3.16, 3.17) are used in the MATLAB command

```
>> [H,P,E] = lqe(Ad,eye(size(Ad)),Cd,Qlqe,Rlqe);
```

where $Q_{lqe} = LL^T$ for $L \triangleq [L_{low}\ L_{high}]^T$ and $R_{lqe} = \mu I$ are used to produce a Kalman filter gain H. Similarly, using O'Dell's method of obtaining uniform singular values given by Equation (3.19), the Kalman filter gain H_o is obtained. Plotting the singular value of both target filter loop designs, using H and H_o produces Figure 3.4 Performing recovery of the target loop using cheap control is next performed using a linear quadratic regulator implemented using the MATLAB command

```
>> [F]=lqr(Ad,Bd,Qlqr,Rlqr);
```

where state and control weighting matrices are defined as $Q_{lqr} \triangleq C_d^T C_d$ and $R_{lqr} \triangleq \rho I$ respectively. To perform recovery step of LQG/LTR method, let $\rho \rightarrow 0$ as suggested by Lemma 3.1. The recovery of the target filter loop using filter gains H and H_o are illustrated in Figures 3.5 and Figure 3.6.

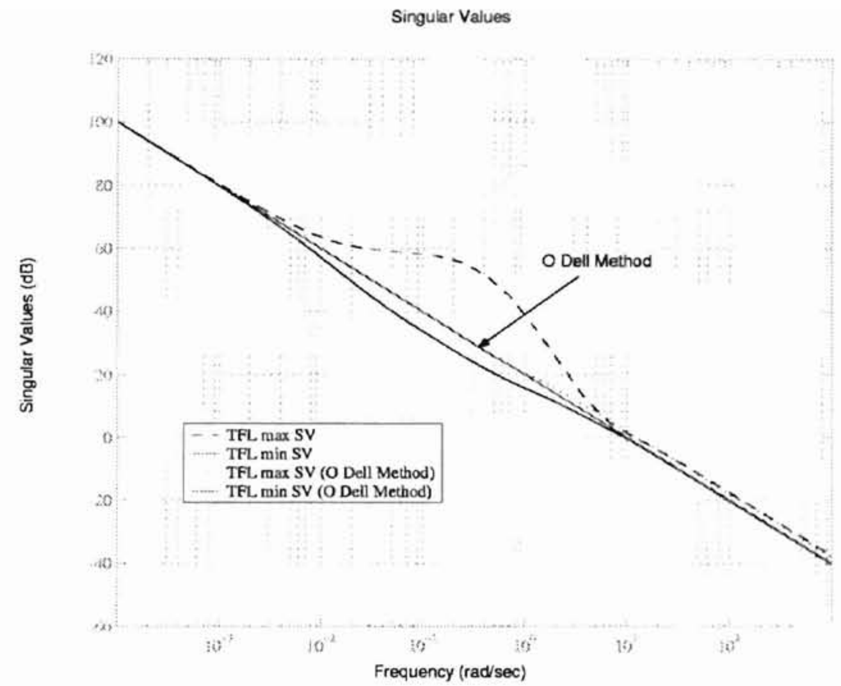

Figure 3.4: Singular Value Plots of Target Filter Design.

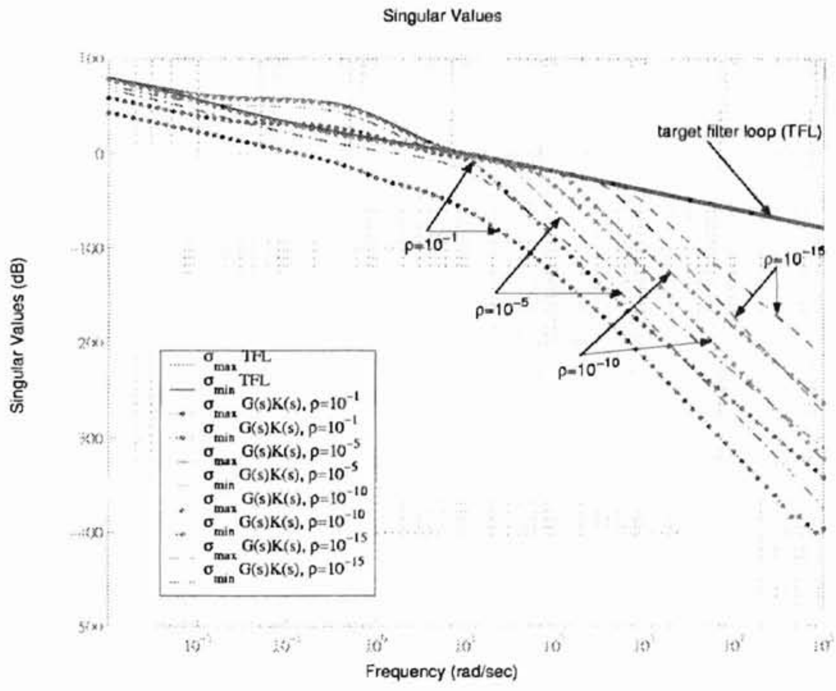

Figure 3.5: Loop Transfer Recovery of Target Filter Loop Using *H*.

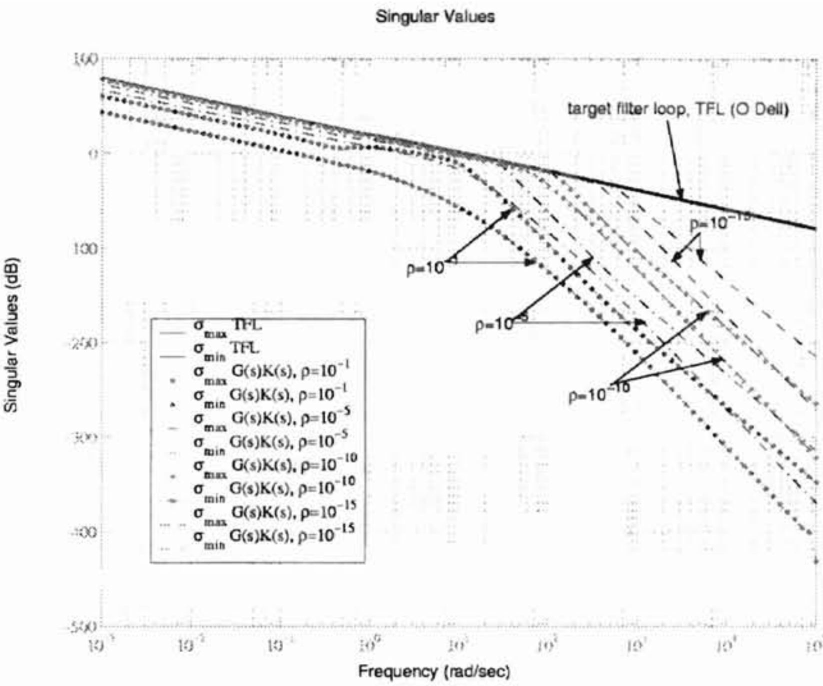

Figure 3.6: Loop Transfer Recovery of Target Filter Loop Using H_o.

Discrete LQG/LTR Example

Modifying the diagram of Figure 3.7 is to include a sampling time, t_s, produces the discrete time system of Figure 3.8. As mentioned in Section 3.2, the discrete LQG/LTR design procedure is essentially the same as the continuous time version. Two MATLAB commands, `lqed.m` and `lqrd.m`, are useful for extending the continuous time LQG/LTR method into discrete time. The `lqed.m` command calculates the discrete Kalman estimator from the desired continuous cost function, which essentially allows the loop shaping ideas presented in Section 3.1.1 to be extended directly into discrete time. Further, `lqed.m` assumes the predicting version of the filtering observer for implementation. Similarly, the `lqrd.m` command calculates the optimal discrete time LQR gain minimizing the corresponding continuous time cost function, which allows the notion of cheap control to be translated into discrete time counterpart.

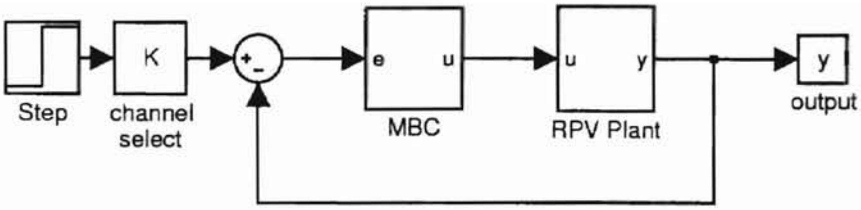

Figure 3.7: Model Based Compensator Block Diagram.

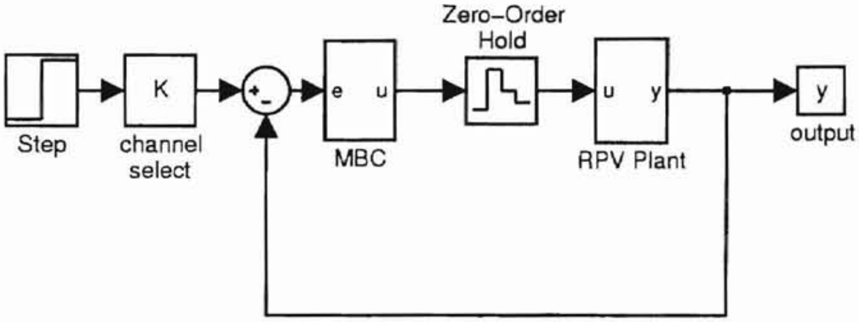

Figure 3.8: Discrete Model Based Compensator Block Diagram.

3.4.2 LTR Simulations

Continuous Time Simulation Results

A block diagram for the model based compensator system is given by Figure 3.7. For simulation, "lqr" gains F and F_o corresponding to H and H_o respectively for $\rho = 10^{-10}$ were chosen. Simulating the model based compensator designs for command reference signal ref=$[-0.1\ 0.1]^T$ using both target filter designs based on observer gain matrices H and H_o yields the step responses given in Figure 3.9. The responses for both systems to step commands are well damped with a settling time near 0.5 seconds for both channels, which approximately corresponds to a natural frequency of $10\frac{rad}{sec}$. The responses are also exhibiting a zero steady state error characteristic as desired.

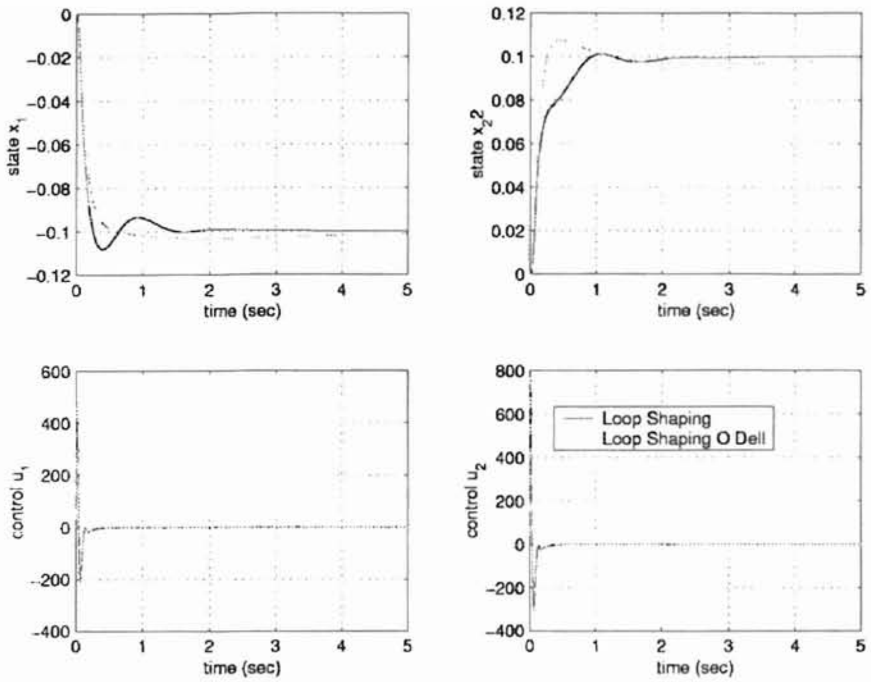

Figure 3.9: Step Response of MBC using H and H_o TFL Designs.

Discrete Time Simulation Results

Using the lqed.m and lqrd.m MATLAB commands along with the continuous time weighting matrices previously described, example 3.1 was solved using discrete LQG/LTR. A sampling time of $1 \cdot 10^{-4}$ was arbitrarily chosen to discretize the RPV model. Singular value plots for the discrete time target filter loop design are shown in Figure 3.10. Similar to the continuous time version, recovery is shown for two target filter designs in Figures 3.11 and 3.12. Simulating the compensator of Figure 3.7 for step change in command input given as $ref = [-0.1 \ 0.1]^T$, yields the response in Figure 3.13.

3.5 Summary

Chapter 3 gives a review of loop transfer recovery for both continuous and discrete time systems. Important points from Chapter 3 to remember include:

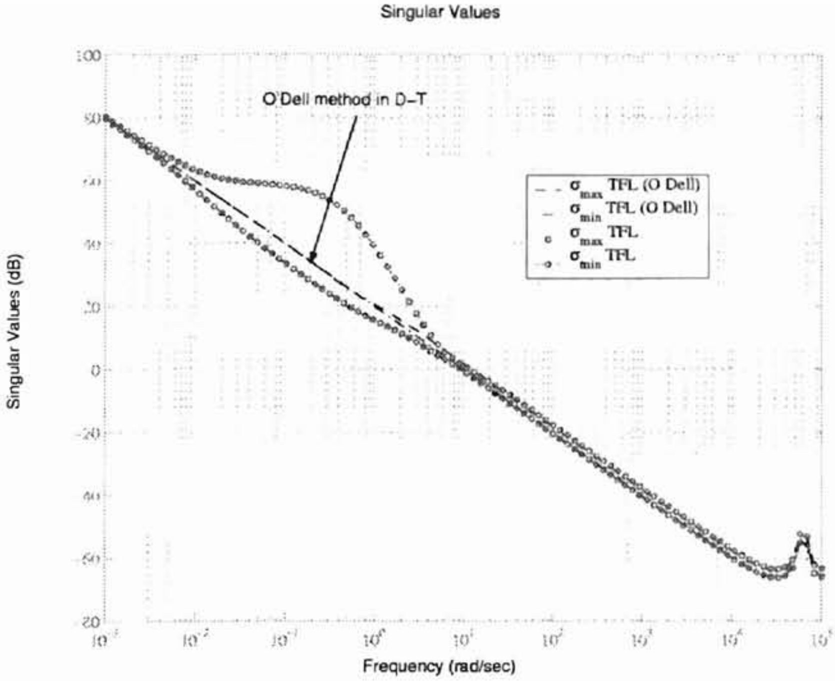

Figure 3.10: Singular Value Plots of Discrete Target Filter Design.

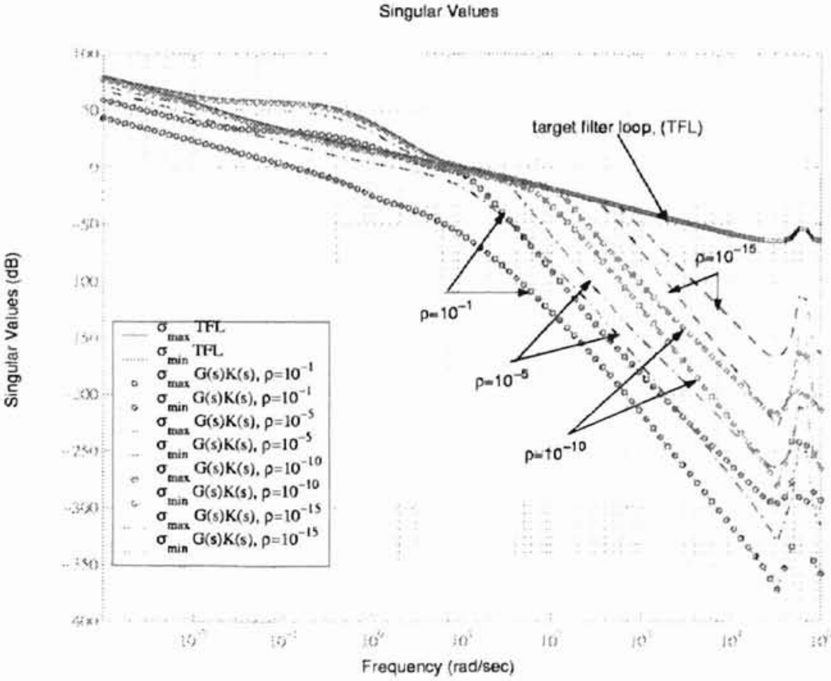

Figure 3.11: Discrete Loop Transfer Recovery of Target Filter Loop Using H.

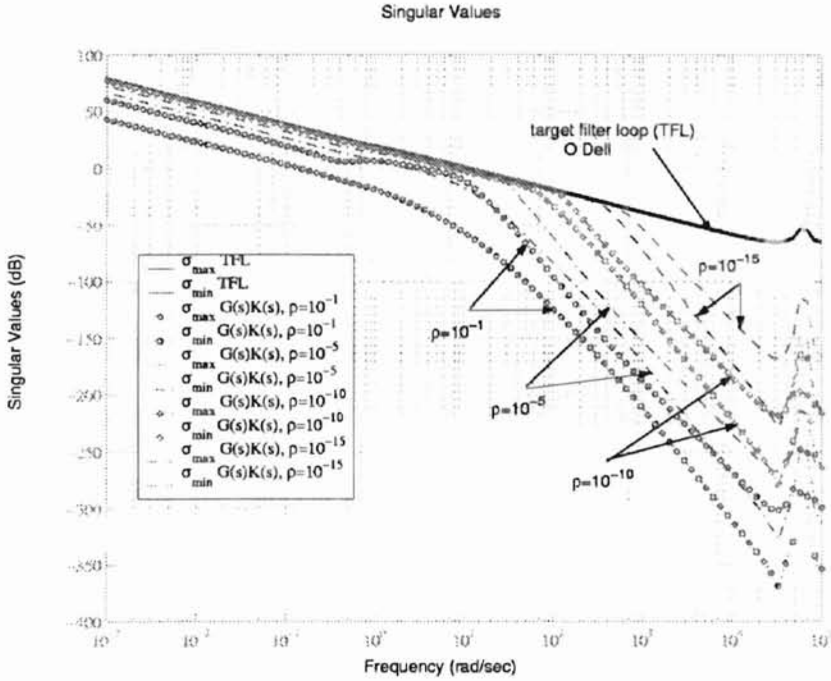

Figure 3.12: Discrete Loop Transfer Recovery of Target Filter Loop Using H_o.

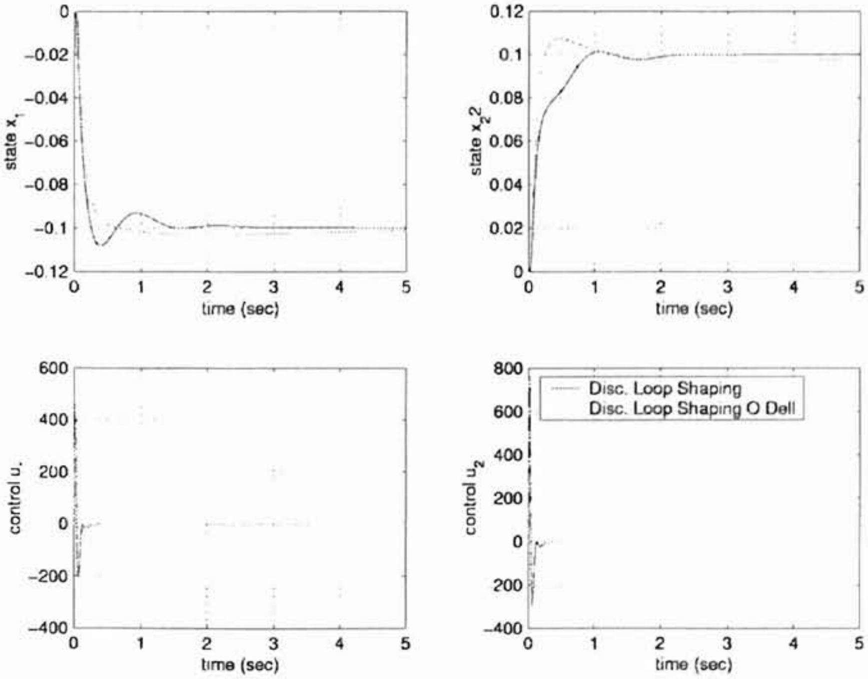

Figure 3.13: Step Response for Model Based Compensator using H and H_o Discrete Target Filter Loop Designs.

- Three basic steps encompass the LTR technique which include:

  1. Define restrictions on the singular values of an open loop transfer function matrix formed by breaking the control loop at the input or output of the plant.
  2. Design of a target loop. For recovery at the plant output, this step corresponds to observer design. Several method for designing the target filter loop exist which are based on optimal control techniques.
  3. While holding the matrix found in step (2) constant, recover the target loop by applying cheap control.

- Designing the target filter loop may be accomplished by any design technique.
- Continuous time Loop transfer recovery can be extended into discrete time.
- Discrete time loop transfer recovery is essential equivalent to continuous time loop transfer recovery except for key differences in stability criteria and observer selection.
- Discrete loop transfer recovery can be implemented for practical applications, (Section 3.4).

Chapter 4

Discrete Variable Structure Control with LTR Hyperplane Design

This section address the possible pitfalls of arbitrarily combining sliding surface and observer design by developing a new sliding hyperplane design technique tailored after loop transfer recovery. Several assumptions are necessary to make a fair comparison between the traditional LTR loop structures and those common to discrete variable structure control. Following a brief description of necessary assumptions is a fundamental outline for a hyperplane based LTR design technique including theoretical considerations of the extension of a LQR type hyperplane design previously reported by Tang [7].

4.1 OBDVSC, Regulation Inside the Boundary Layer

Consider a discrete sliding surface given by $\hat{S} = G\left[x_d(k) - \hat{x}(k)\right]$. Given a boundary layer thickness of ϕ, for the discrete variable structure control law given in Equation (2.33) the $sat(\frac{\hat{S}}{\phi})$ term reduces to $\frac{\hat{S}}{\phi}$ considering the case when $|\hat{S}| < \phi$, (Richter [6] and Tang [7]). Thus, Equation (2.33) can be rewritten as Equation (4.1).

$$u(k) \;=\; \tfrac{1}{GB}\left[G(I-A)\hat{x}(k) + G\Delta x_d(k) + \tfrac{K}{\phi}\hat{S}\right] \tag{4.1}$$

Further restricting Equation (4.1) by supposing a regulation control objective, (*i.e.*, $x_d(k+1) = x_d(k) = 0 \Rightarrow \Delta x_d = 0$), and substituting in for $\hat{S}$ yields,

$$u(k) \;=\; \underbrace{-\frac{G}{GB}\left[A - (1-\frac{K}{\phi})I\right]}_{F}\hat{x}(k) \tag{4.2}$$

Thus, inside the boundary layer under a regulation control objective the OBDVSC essentially reduces to a linear state feedback control of the traditional form $u = -F\hat{x}(k)$. Recall, the equivalent dynamics described by A_{eq} outlined in by Equation (2.37).

4.2 Fitting OBDVSC Into a LTR Framework for Recovery

Both OBDVSC and LTR schemes are similar in that both use a state estimator and state feedback for control. The major difference between DVSC and LTR based systems lies in the trajectory generation and nonlinear hyperplane contributions of the DVSC law, explicitly seen in the control law of Equation (2.33). However, as previously seen, if the DVSC controller is given the task of regulation inside the boundary layer, then the DVSC scheme reduces to a traditional linear state

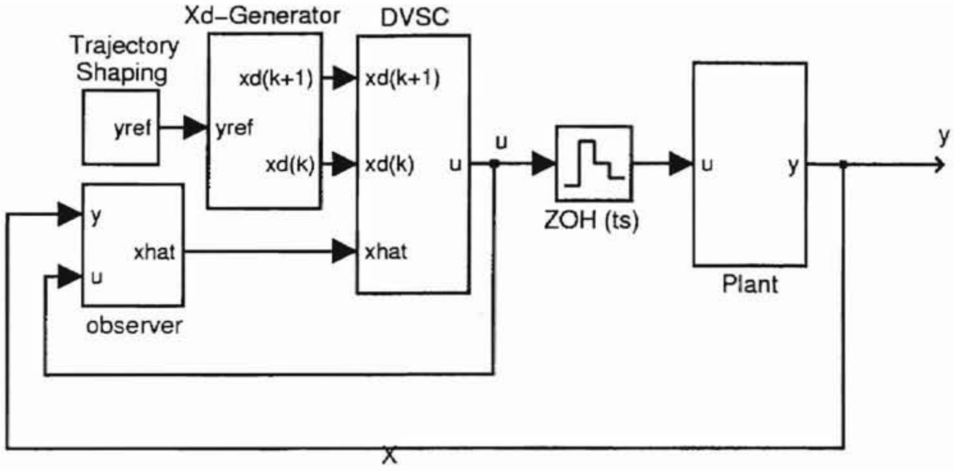

Figure 4.1: DVSC Recovery at the Plant Output.

feedback control with a special structure on control matrix F. The question now becomes where to break the DVSC servo control loop in order to make a reasonable comparison?

Following the discussion of LTR for recovery at the plant output, the DVSC control loop is broken at the plant output as shown in Figure 4.1 at the point labeled "X", which is equivalent to performing recovery at the plant output in the traditional LTR procedure. For a regulation task inside the boundary layer, Figure 4.1 can be reduced to Figure 4.2. To further mimic the standard LTR control loop shown in Figure 3.1, feedback can be placed around the compensator and plant as shown in Figure 4.3. Note the sign change in the summation block of Figure 4.3, which is an artifact of the $\hat{S}(k)$ term of Equation (2.33).

4.3 Recovering the Target Filter Loop

Fundamental to the LTR process for recovery at the plant output is the combination of two key components, namely the optimal control solution of the LQ regulator and cheap control problem as mentioned in Lemma 3.1. For discrete

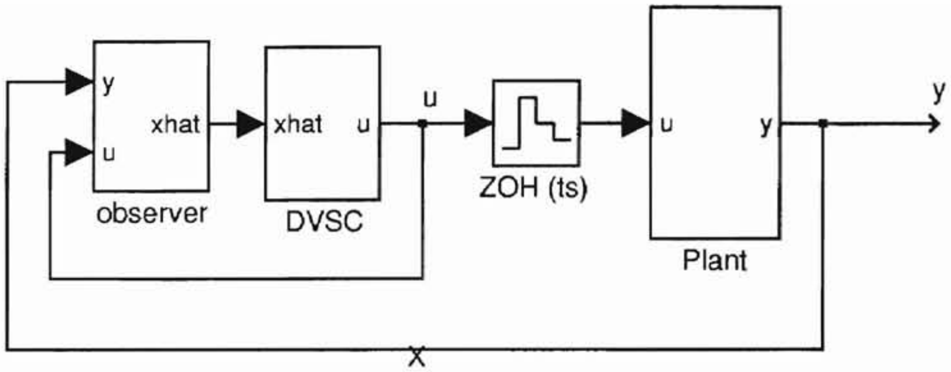

Figure 4.2: DVSC Recovery at the Plant Output for Regulation Control Objective.

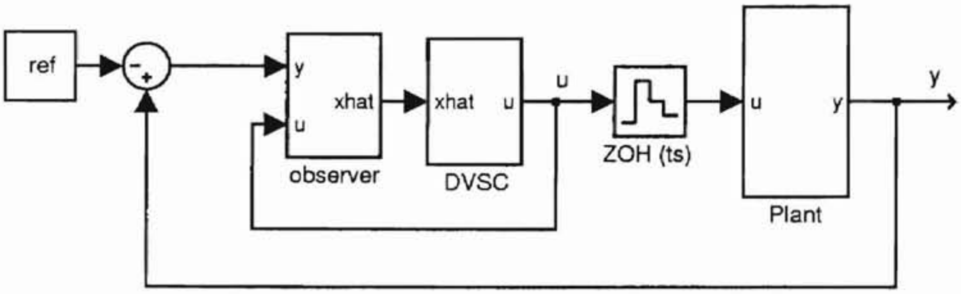

Figure 4.3: DVSC Mimicking the LTR Standard Control Loop.

variable structure control hyperplane design using quasi-LTR (qLTR) to be possible, similar key components must exist.

First addressing the component necessary for the recovery step leads to an extension of previously mentioned LQR hyperplane procedure of Tang [7]. The symmetric root locus of Equation (2.50) is vital to solving the so called inverse optimal control problem as well as incorporating the LQR hyperplane design technique into a new LTR hyperplane design methodology. In order to reproduce the traditional cheap control problem using Tang's [7] methodology, emphasis must be place on the state weighting matrix because a control weighting of $R = \mu I$ is not possible due to the hard coded assumption that $R = 1$ within Tang's MATLAB scripts **dvsclqr1.m** and **dvsclqr2.m**. Instead, a scalar weighting of ρ is placed on the state weighting matrix Q_d. Further, for the LTR hyperplane design procedure to parallel that of the traditional LTR procedure, the state weighting matrix is taken as $Q_d = C^T C$, so that the symmetric root locus for the LTR hyperplane design is given by Equation (4.3).

$$\Phi^T(z^{-1})\underbrace{\rho C^T C}_{Q_d}\Phi(z) = -R \tag{4.3}$$

It is important to note that because of the scalar relationship of Equation (2.50) the weighing scheme or Equation (4.3) is possible. Running Tang's scripts **dvsclqr1.m** and **dvsclqr2.m** for a discrete system $\Sigma_d(A, B, C, D)$ first requires inputs A, B, Q_d and α then outputs G and state feedback matrix F.

The last key component which is most fundamental to the qLTR hyperplane design is a behavior similar to that exhibited in Lemma 3.2. The following conjecture outlines the LTR recovery using the weighting scheme synonymous to Equation (4.3).

Conjecture 4.1 *For recovery at the plant output given a discrete system $\Sigma_d(A, B, C, D)$ using the discrete time model based compensator $K(z)$ of Figure 4.3 taken as $K(z) = F(zI - A + BF + HC)^{-1}H$ where H is a prediction filtering* *observer matrix of the form in Equation (3.22) and feedback matrix F has the special form $F = \frac{G}{GB}[A - (1 - \frac{K}{\phi})I]$ for a discrete variable structure control system using the sliding manifold $\hat{S}(k) = G[x_d(k) - \hat{x}(k)]$ if conditions $1 \rightarrow 3$ are valid where*

1. $\lambda_i(A - HC) < \mathbb{C}^{\odot}$ *for* $1 \leq i \leq n$
2. $\lambda_i(A - BF) < \mathbb{C}^{\odot}$ *for* $1 \leq i \leq n$
3. *The discrete variable structure control constraint Equations (2.42-2.47) are satisfied*

and the traditional LTR procedure for recovery at the plant output is performed assuming the symmetric root locus Equation (4.3) and state weighting matrix $Q_d = \rho C^T C$ using Tang's LQR hyperplane design technique, then

$$lim_{\rho\rightarrow\infty} K(z) = [C(zI - A)^{-1}B]^{-1}[C(zI - A)^{-1}H]. \tag{4.4}$$

point wise in z.

Conjecture 4.1 is left unproven, however simulation results shown in Section 5 suggest the validity of the argument. The significance of the argument can be directly seen in Conjecture 4.2.

Conjecture 4.2 *If conditions $1 \rightarrow 3$ and Equation (4.4) of Conjecture 4.1 are valid for the given discrete time LTI system $\Sigma(A, B, C, D)$, then it follows that:*

$$lim_{\rho\rightarrow\infty} G(z)K(z) = \underbrace{C(zI - A)^{-1}B}_{G(z)} \cdot \underbrace{[C(zI - A)^{-1}B]^{-1}C(zI - A)^{-1}H}_{lim_{\rho\rightarrow\infty} K(z)} \tag{4.5}$$

so that

$$lim_{\rho\rightarrow\infty} G(z)K(z) = C(zI - A)^{-1}H \tag{4.6}$$

Proof. *Simple substitution of result taken from Conjecture 4.1 into $G(z)K(z)$.*

This result is similar to the traditional LTR result by Doyle and Stein [41], in that as ρ is increased, the compensator begins to behave or recover the properties of the target filter loop.

4.4 Obtaining Exact Recovery

Consider an observer based discrete sliding mode system operating inside the boundary layer $|\hat{S}| < \phi$ such that the reduced control is given by Equation (4.2). Formulating the error recovery matrix in discrete time for recovery at the plant output paralleling Goodman [40] given a discrete state space quadruple $\Sigma_d(A, B, C, D)$, an observer gain matrix H and state feedback control matrix F with the special form of Equation (4.2) yields

$$E_o(z) \triangleq C\Phi(z)H - G(z)K(z) \tag{4.7}$$

where $G(z) = C\Phi(z)B$ and the compensator $K(z)$ is given by

$$K(z) = F(zI - A + BF + HC)^{-1}H \tag{4.8}$$

Lemma 4.1 *For discrete recovery at the plant output, $E_o(z)$ may be written as*

$$\begin{aligned} E_o(z) &\triangleq C\Phi(z)H - G(z)K(z) && (4.9) \\ &= \Gamma(z)[I + M_o(z)]^{-1}M_o(z) && (4.10) \end{aligned}$$

where $\Gamma(z) \triangleq [I + C\Phi(z)H]$ and $M_o(z) \triangleq C(zI - A + BF)^{-1}H$

Proof, See Goodman [40], it essentially follows from the proof of Lemma 3.3.

Recall, the output recovery error matrix $E_o(z)$ measures the level of recovery and that rendering $E_o(z)$ zero for all frequency is equivalent to exact recovery of the target filter loop. Theorem 4.1 outlines conditions yielding exact recovery of the target filter loop using the structure of state feedback gain matrix F in Equation (4.2).

Theorem 4.1 *Given $E_o(z)$ of Equation (4.7) constructed from an observer gain matrix H, state feedback gain matrix F of the form $F = \frac{G}{GB}\left[A - (1 - \frac{K}{\phi})I\right]$, the discrete plant quadruple $\Sigma_d(A, B, C, D)$, and the non-defective[1] matrix A_{eq} of*

[1] A non-defective matrix is defined as a square matrix having a complete set of independent eigenvectors.

Equation (2.37) with eigenvalues λ_i, $1 \le i \le n$ and corresponding right eigenvectors u_i, $1 \le i \le n$ and left eigenvectors v_i, $1 \le i \le n$, the three following conditions are equivalent:

1. $E_o(z) = 0$.

2. $M_o(z) = 0$.

3. $Cu_i = 0$ *or* $v_i^* H = 0, \forall 1 \le i \le n$

Proof, For Conditions $1 \Leftrightarrow 2$ *it follows from the discussion in Lemma 3.3 extended into discrete time. For conditions* $2 \Leftrightarrow 3$*, let* A_{eq} *be written as*

$$A_{eq} = U \Lambda V^* \tag{4.11}$$

where

$$\Lambda \triangleq \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \tag{4.12}$$

and

$$U \triangleq \left[\begin{array}{c|c|c|c} u_1 & u_2 & \cdots & u_n \end{array} \right] \tag{4.13}$$

$$V \triangleq \left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array} \right] \tag{4.14}$$

so that U *and* V *are scaled* $UV^* = V^*U = I$*. Looking at* $M_o(z)$ *reveals*

$$\begin{aligned} M(z) &= C(zI - A + BF)^{-1}H & (4.15) \\ &= C(zI - A + \frac{BG}{GB}\left[A - (1 - \frac{K}{\phi})I\right])^{-1}H & (4.16) \\ &= C(zI - A_{eq})^{-1}H & (4.17) \\ &= C(zUV^* - U\Lambda V^*)^{-1}H & (4.18) \\ &= CU(zI - \Lambda)^{-1}V^*H & (4.19) \end{aligned}$$

which can be alternatively written in matrix residue form as

$$M(z) = \sum_{i=1}^{n} \frac{Cu_i v_i^* H}{z - \lambda_i} \tag{4.20}$$

From Equation (4.19) it is trivial to see Condition $2 \Leftrightarrow 3$.

4.5 Design Procedure

Given a discrete quadruple $\Sigma_d(A, B, C, D)$ operating inside a boundary layer ϕ, the hyperplane design for discrete variable structure control procedure using quasi-loop transfer recovery for recovery at the plant output is as follows:

1. Consider recovery at the plant output, break the DVSC control loop at the output of the plant (See Section 4.2).

2. Design the target filter loop, $C\Phi(z)H$ using any tuning method. Optimal control lends several plausible techniques for designing the target filter loop, (See Section 3.1.1). For upcoming designs shown in Chapter 5, MATLAB's **dlqe.m** is used to shape the target filter loop (*i.e.*, design the filtering observer matrix H).

3. Fix the filter gain matrix H, and design the observer based hyperplane $\hat{S}(k) = G\tilde{x}(k)$ using quasi-loop transfer recovery (qLTR) as follows:

    (a) Satisfy constraint equations given by Equations (2.42, 2.43, 2.44, 2.45, 2.46, 2.47) for a fixed α. (For a general rule of thumb see Remark 5.1).

    (b) Set Q_d=$\rho C^T C$, R=1.

    (c) Run Tang's scripts **dvsclqr1.m** and **dvsclqr2.m** with operands A, B, Q_d and α for $\rho \to \infty$ to obtain desired level of recovery ultimately yielding G and F. **dvsclqr1.m** and **dvsclqr2.m** will solve the LQR hyperplane design via the inverse optimal solution and using **dvsclqr1.m** and **dvsclqr2.m** in the context of (b) coincides with a hyperplane design using loop transfer recovery.

Chapter 5

LTR Hyperplane Design Example

This chapter covers a design example following the LTR hyperplane design technique discussion in Chapter 4. A symbolic 4^{th} order disk drive model is first shown followed by parameters sufficient to match the Goh *et. al.* [50] drive model given in Section 2.5 within the frequency domain. Next, the novel compensator design technique encompassing both target filter loop and loop transfer recovery hyperplane design are covered.

5.1 Symbolic Disk Drive Model

Consider an inertial mass rotating about a single pivot point. A symbolic model for the rotating system, see Figure 5.1, includes an actuator inertia J, torque constant K_t, coefficient of friction μ, actuator arm radius r and resonance model $R(s)$ taken as Equation. (5.1). The appropriate continuous time model mapping input current (amps) into displacement (inches) is given by Equation (5.2). Substituting the numerical values of parameters from Table 5.1 and putting Equation (5.2) into continuous time observerable canonical form yields the state space representation $\Sigma(A_c, B_c, C_c, D_c)$ of Equations (5.3-5.6).

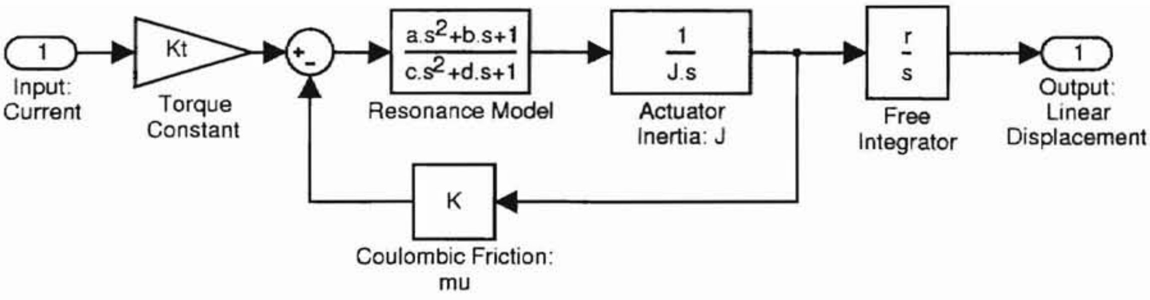

Figure 5.1: Symbolic 4^{th} Order Disk Drive Model.

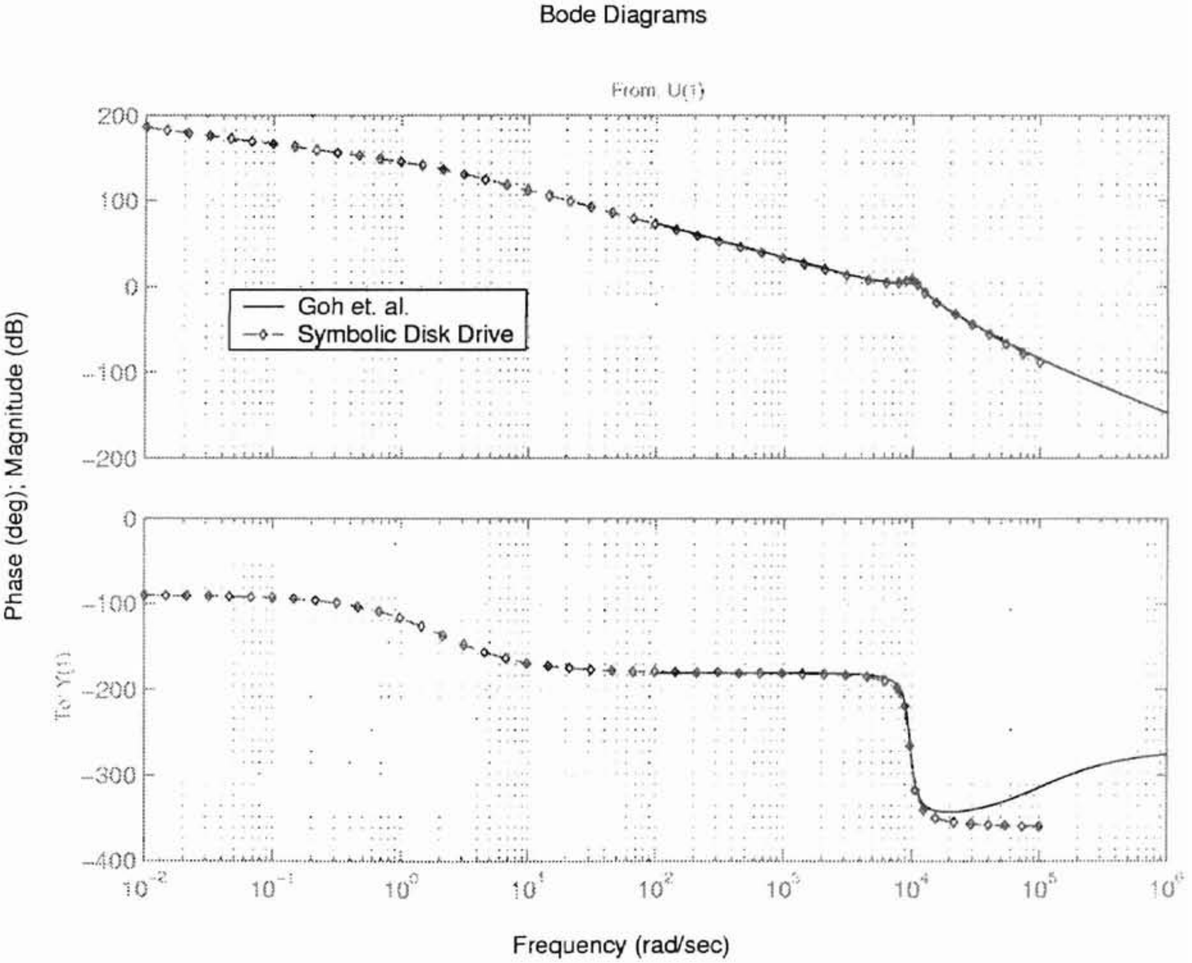

Figure 5.2: Bode Diagram Comparison of Symbolic Drive Model to Goh *et.al.*

$$R(s) = \frac{as^2 + bs + 1}{cs^2 + ds + 1} \tag{5.1}$$

$$G(s) = \frac{K_t\, r}{s(Jcs^3 + Jds^2 + Js) + \mu} \tag{5.2}$$

Table 5.1: Simulation Parameters.

Resonance Parameters	
a	0
b	0
c	$1.0243 \cdot 10^{-8}$
d	$1.6349 \cdot 10^{-5}$
Actuator Inertia, J	$5 \cdot 10^{-5}$
Actuator Arm Radius, r (in)	2.00
Coefficient of Friction, μ	0.0001
Torque Constant, K_t $(\frac{oz-in}{amp})$	1000
Sampling time (sec)	$50 \cdot 10^{-6}$
Tracks Per Inch (TPI)	17500

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1.95 \cdot 10^8 & -9.76 \cdot 10^8 & -1.59 \cdot 10^3 \end{bmatrix} \tag{5.3}$$

$$B_c = \begin{bmatrix} 0 & 0 & 0 & 3.91 \cdot 10^{15} \end{bmatrix}^T \tag{5.4}$$

$$C_c = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.5}$$

$$D_c = [0] \tag{5.6}$$

Applying a zero order hold using a sampling time of $t_s = 40 \cdot 10^{-6}$ seconds yields the discrete time state space representation $\Sigma_d(A, B, C, D)$ given in
Equations (5.7-5.10).

$$A = \begin{bmatrix} 1 & 4.00 \cdot 10^{-5} & 7.90 \cdot 10^{-10} & 1.04 \cdot 10^{-14} \\ 0 & 1.0 \cdot 10^{0} & 3.90 \cdot 10^{-5} & 7.73 \cdot 10^{-9} \\ 0 & -1.51 \cdot 10^{-1} & 9.24 \cdot 10^{-1} & 3.77 \cdot 10^{-5} \\ 0 & -7.37 \cdot 10^{3} & -3.69 \cdot 10^{3} & 8.64 \cdot 10^{-1} \end{bmatrix} \tag{5.7}$$

$$B = \begin{bmatrix} 4.09 \cdot 10^{-5} & 4.07 \cdot 10^{0} & 3.02 \cdot 10^{5} & 1.47 \cdot 10^{10} \end{bmatrix}^T \tag{5.8}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.9}$$

$$D = [0] \tag{5.10}$$

A Bode diagram comparison of the disk drive model used in Section 2.5 and the symbolic drive model substituting the parameters of Table 5.1 is given in Figure 5.2. The comparison essentially shows that the two models are equivalent for a given frequency spectrum.

5.2 Target Filter Loop Design

Following the traditional LTR procedure for recovery at the plant input, the target filter loop is designed first for a cross-over frequency near 1000 $\frac{rad}{sec}$. The MATLAB command

```
>> [T,P,Z,E] = dlqe(A,diag([.5e0 0 0 0]),C,diag([1e-2 0 0 0]),1);
>> H=A*T;
```

produces the desired target loop shaping. Note the prediction filtering structure[1] on the observer gain matrix H. The frequency response of the resulting target filter loop is shown in Figure 5.7, which is generated using the MATLAB **dlinmod.m** and **dsigma.m** commands in conjunction with the Simulink block diagram in Figure 5.3. Checking the eigenvalues of the observer dynamics, $\lambda_i(A - HC)$, reveals that condition 1 of Conjecture 4.1 is satisfied.

The next step in the LTR hyperplane design procedure is to choose the parameters within the constraint Equations (2.42-2.47).

Remark 5.1 *Satisfying the constraint Equations (2.42-2.47) can be an arduous task because of the unequal amount of constraint equations and unknowns. To satisfy the constraints, a possible rule of thumb for choosing parameters is as follows:*

1. *Fix 3 of the unknown variables using practical considerations like sampling time Δt limitations, boundary layer thickness ϕ, and perhaps α so that there are 3 equations and 3 unknowns.*

2. *Solve for the remaining unknowns using constraint equations making sure all constraints are satisfied.*

For example, suppose α, Δt and ϕ are to be fixed which seem practically feasible. The remaining unknowns from the constraint equations are K, ϵ and γ. Rewriting the constraint equations using simple algebra to solve for the unknowns leads to

$$\begin{bmatrix} \phi(1-\alpha) \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 2\Delta t \\ 1 & \Delta t \end{bmatrix} \begin{bmatrix} \gamma \\ \epsilon \end{bmatrix} \tag{5.11}$$

Solving for the unknowns γ and ϵ becomes a simple matrix inversion and multiplication.

[1] For more details on observer structures for discrete time systems see Section 3.2.2.

For the simulations shown, $\alpha = -0.92$, $\Delta t = t_s = 40 \cdot 10^{-6}$ and the boundary layer thickness $\phi = 1 \cdot t_s$ were chosen to be constant. Solving for the remaining unknowns as suggested by Remark 5.1 reveals $\gamma = 3.2 \cdot 10^{-6}$ and $\epsilon = 0.92$.

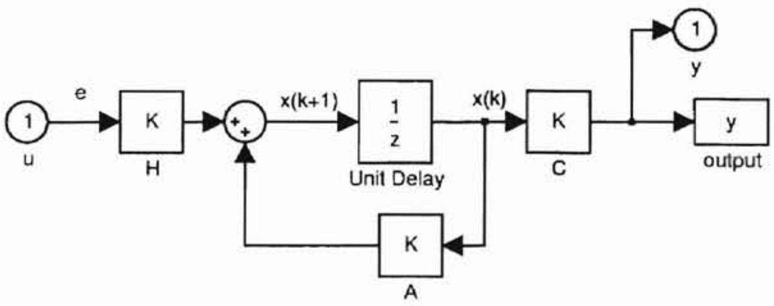

Figure 5.3: $C\Phi(z)H$, Target Filter Loop Block Diagram (Open-Loop).

5.3 Hyperplane Design using Loop Transfer Recovery

The last step in the procedure is to apply LTR as Conjecture 4.1 implies using $Q_d = \rho C^T C$. This final recovery step is carried out via MATLAB scripts, shown in Appendix C. The process includes tending the design variable ρ large and allowing the compensator loop shown in Figure 5.4 to recover the target filter loop design of

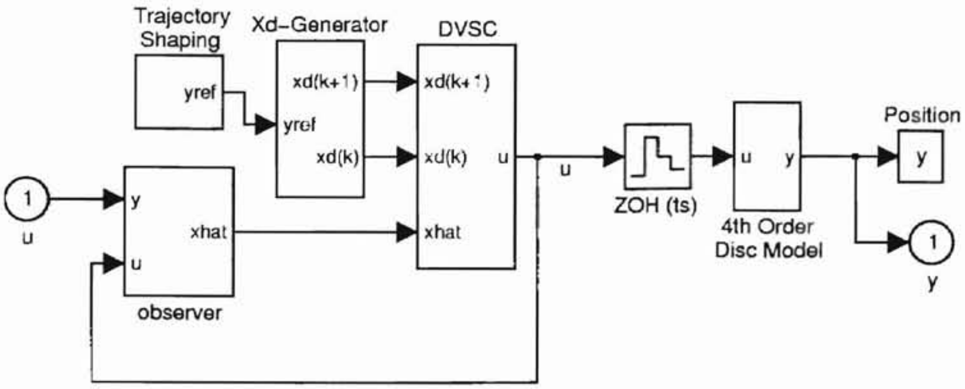

Figure 5.4: Compensator Loop Block Diagram (Closed-Loop).

Figure 5.3. Recovery of the filter loop is illustrated by Figure 5.8 for $\rho = 10^{14}, 10^{16}, 10^{17}$.

Remark 5.2 *For this simulation example it was necessary to use relatively large values for ρ. If smaller values for ρ were used in an effort to recover the target filter loop, Tang's scripts would ultimately fail during the construction of the F matrix which uses the MATLAB* **dlqr.m** *command.*

To further check the recovery design, Figure 5.9 was simulated[2] using the hyperplane design for F for $\rho = 10^{14}$ in the time domain (as compared to the frequency domain representation shown in the recovery Figure 5.8) for a step response equivalent to a single track change in set point at 17500 TPI (tracks per inch). Figures 5.5, 5.6 were used to simulate the time response in a closed loop fashion. Recall the cross over frequency of the target filter loop and the recovered compensator loop is near 1000 $\frac{rad}{sec}$ which approximately corresponds to a settling time of approximately 4.5 ms (assuming critically damped system), as seen in Figure 5.9. Also in Figure 5.9 is the control effort used during the closed loop simulation for the compensator. A *high* frequency component in evident in the control effort. Looking at the internal states reveals that the observer estimate of position not converging to the actual position. Also, although the design variable ρ is relatively large, the control effort is relatively small in magnitude. This fact contrasts the traditional LTR methodology which typically exhibits a increase in control effort for an increase in recovery effort.
As Theorem 4.1 suggests, to yield exact LTR the recovery error matrix $M_o(z)$ must vanish. Specifically paying attention to Equation (4.20) and calculating appropriate right and left eigenvectors (unscaled) via MATLAB **eig.m** command in fact reveals

[2] Simulations are in discrete time, although discrete samples are not specifically shown.

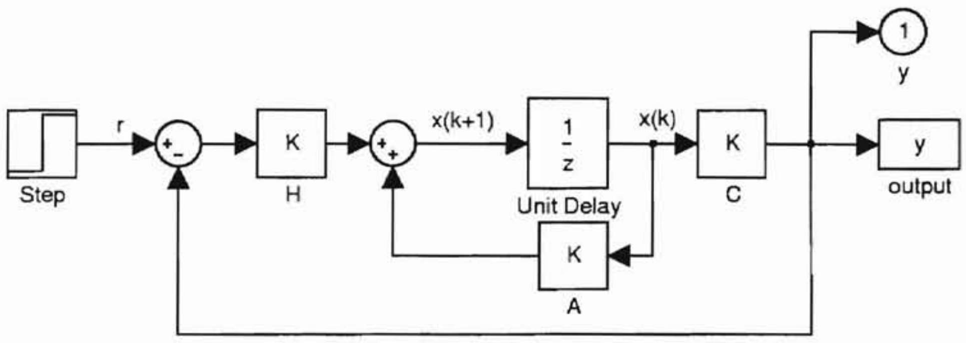

Figure 5.5: $C\Phi(z)H$, Target Filter Loop Block Diagram Loop (Closed-Loop) .

the following:

$$\sum_{i=1}^{n} v_i^* H \approx 0 \; for \; 1 \leq i \leq n \tag{5.12}$$

which also somewhat verifies the validity of the recovery procedure and the argument contained within Conjecture 4.1.

5.4 Behavior Outside the Boundary Layer

To simulate the behavior of the system outside the boundary layer thickness, two extra simulations were ran. The first simulation considered a regulation control objective with a large initial condition $[\frac{25}{TPI}\, 0\, 0\, 0]^T$ placed in the plant dynamics of

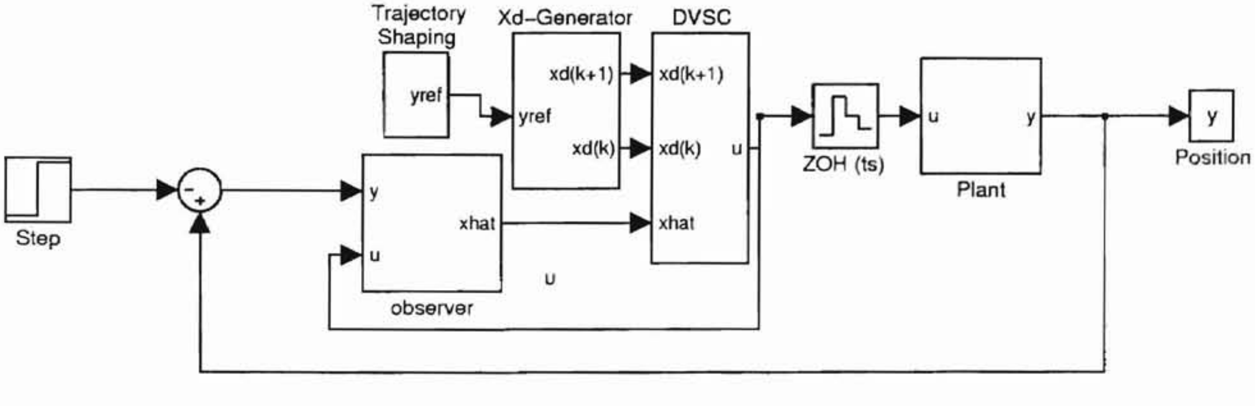

Figure 5.6: Compensator Loop Block Diagram Loop (Closed-Loop).

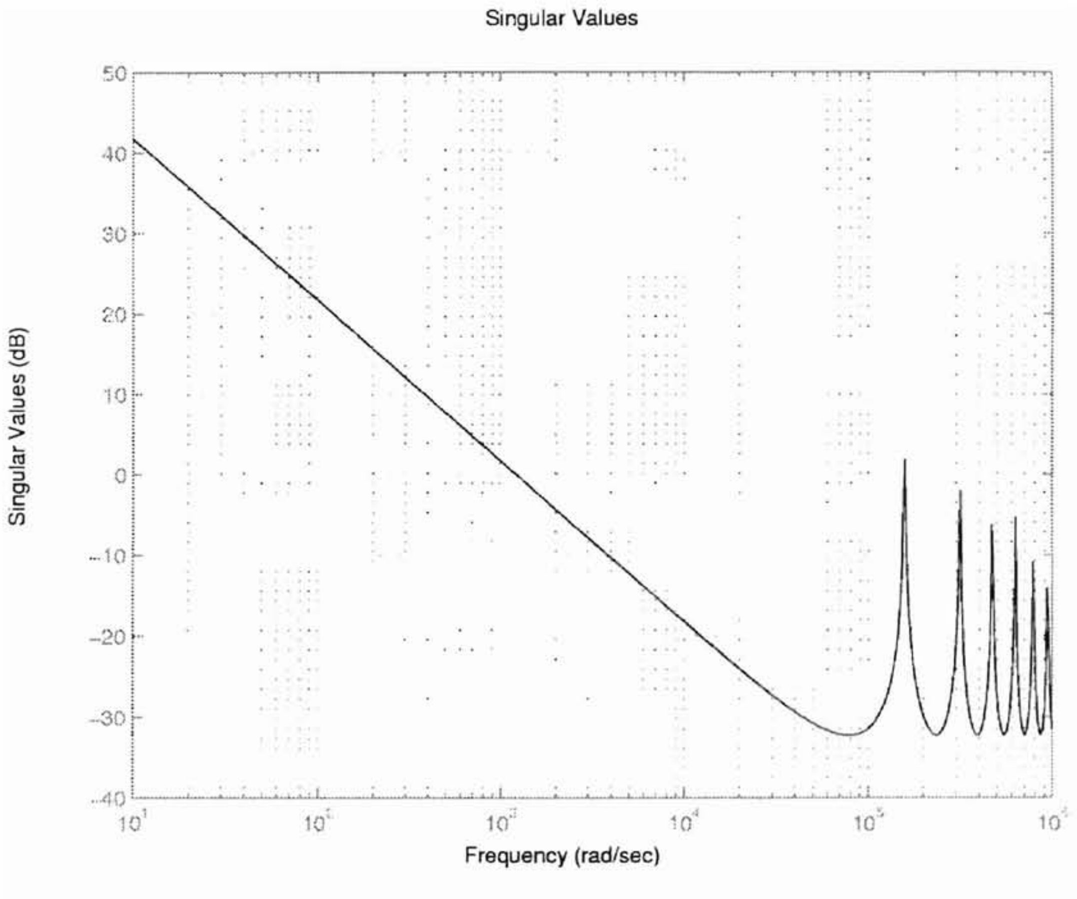

Figure 5.7: Frequency Response of the Target Filter Loop.

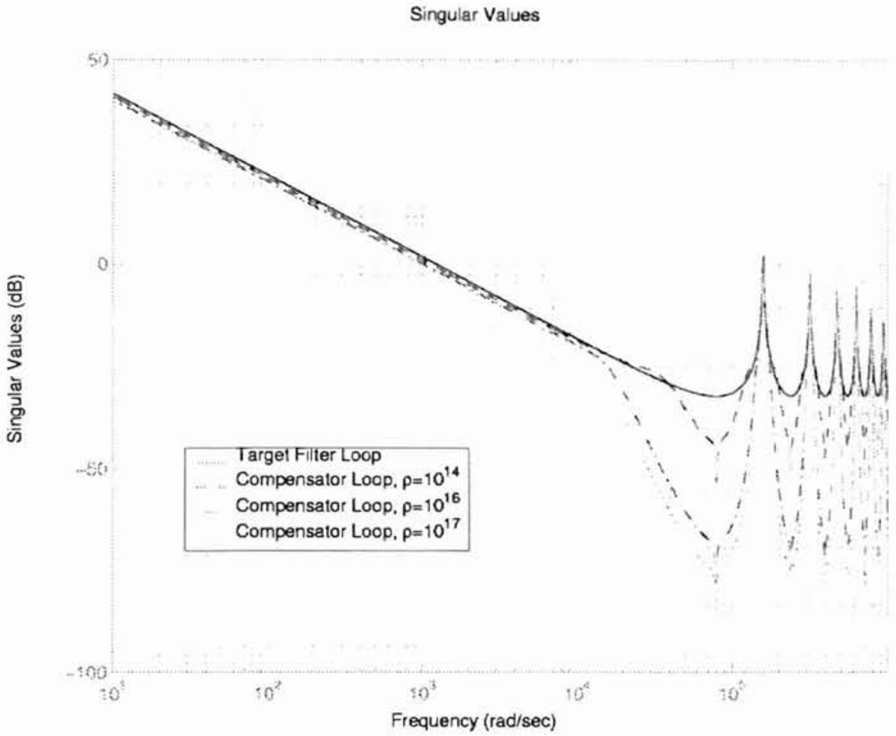

Figure 5.8: Target Filter Loop Recovery ($\rho = 10^{14}$, 10^{16}, 10^{17}).

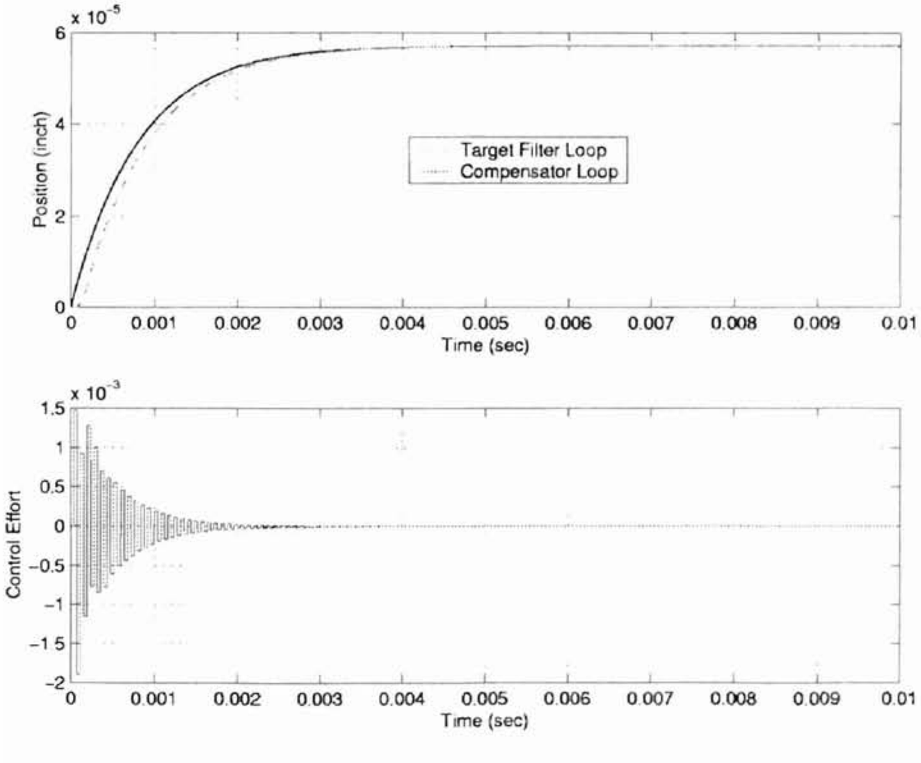

Figure 5.9: Time Domain Responses.

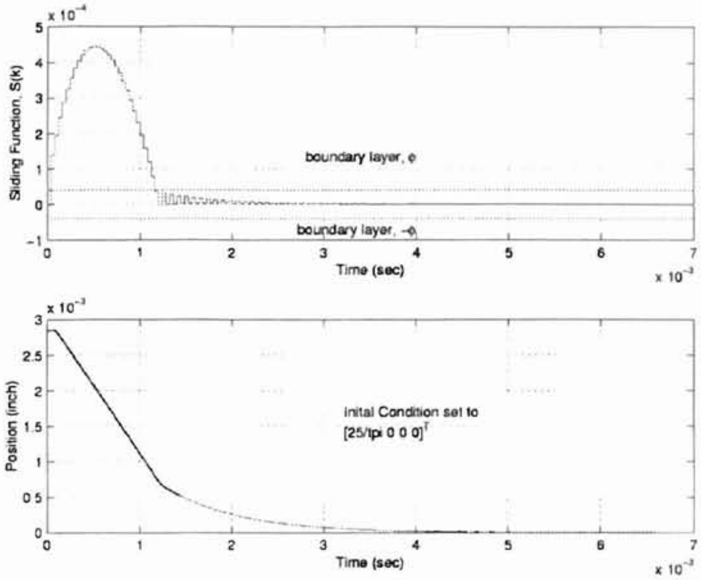

Figure 5.10: Sliding Surface for Initial Conditions $[\frac{25}{TPI}\,0\,0\,0]^T$.

Figure 5.6. The second simulation considers a change in step input of 25 tracks (*i.e.*, $\frac{25}{TPI}$ for the same system without initial conditions). The sliding function for both scenarios is given in Figure 5.10 and Figure 5.11 respectively. Both simulations show the system trajectory clearly leaving the boundary layer region due to either a set point change or initial conditions. For both simulations the trajectory returns inside the boundary layer. This result alludes to the sliding surface design by qLTR being locally attractive in neighborhood around ϕ.

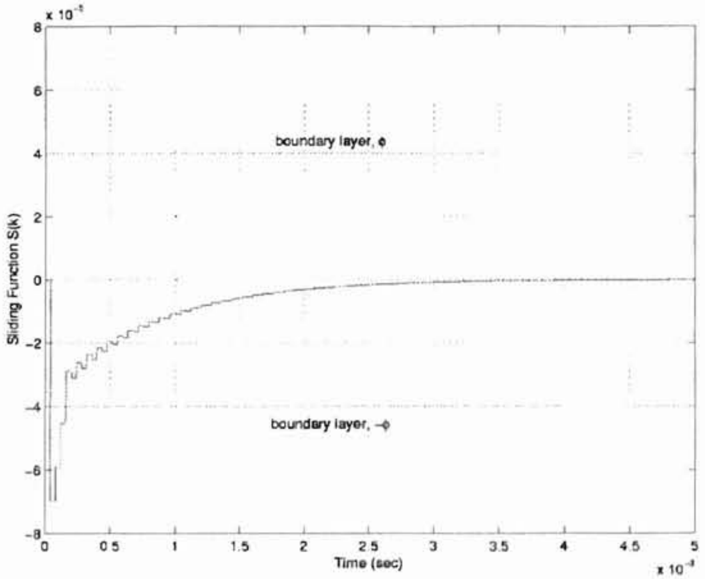

Figure 5.11: Sliding Surface for 25 Track Set Point Change.

Chapter 6

Conclusions

A new hyperplane design procedure for discrete variable structure control systems has been presented. The method seeks to combine observer and controller design for discrete variable structure systems. A brief literature review of discrete LTR and discrete variable structure control is given to lay the foundation for the hyperplane design technique. Several assumptions are necessary for the theory including that the sliding dynamics are restricted inside the boundary layer thickness ϕ, a regulation control objective, (*i.e.*, no trajectory generation from a x_d-generator) and recovering a target filter loop based on breaking the control loop at the plant output. A key conjecture suggests compensator loop behavior may approach that of a target filter loop as a *cheap control* like mechanism is applied. Simulation results show target filter loop recovery for a discrete 4^{th} order disk drive plant using the newly proposed method.

6.1 Contributions

- A novel hyperplane design procedure for discrete variable structure control systems using a quasi loop transfer recovery (qLTR) technique consisting of:

1. Necessary and sufficient conditions for exact target filter loop recovery for recovery at the plant output.
2. Conjecture for qLTR behavior as *cheap* like control is applied forecasting *recovery* of a target filter loop.
3. Simulation results verifying target filter loop recovery alluded to by qLTR conjecture statement for disc drive application.

- Design examples for both discrete variable structure control and discrete time LQG/LTR. Two different disturbance observers within discrete variable structure control simulated for a 4^{th} order disk drive model.

6.2 Future Work

Formal proof of Conjecture 4.1 is needed for stronger validation of the design technique. Several relationships among design variables need investigation. The following list of questions may give possible future work pertaining to hyperplane design for discrete variable structure control using loop transfer recovery:

- What role does the design variable α play within hyperplane design using LTR?
- What limitations are introduced from the sampling process of discrete time systems in relation to LTR hyperplane design?
- What are the significant impacts of system characteristics (*i.e.*, minimum phase versus non-minimum phase plants)?

Bibliography

[1] E.A. Misawa, "Observer-Based Discrete-Time Sliding Mode Control with Computational Time Delay: The Linear Case," *Journal of Dynamic Systems, Measurement, and Control*, pp. 819-821, 1997.

[2] U. Itkis, *Control of Systems of Variable Structure.* John Wiley and Sons, New York, 1976.

[3] V.I. Utkin, "Variable Structure Systems with Sliding Mode," *IEEE Transactions on Automatic Control*, vol. 22, pp. 212-222, 1977.

[4] J.J. Slotine and W. Li, *Applied Nonlinear Control.* Prentice Hall, New Jersey, 1991.

[5] C. Edwards and S. Spurgeon, *Sliding Mode Control: Theory and Applications.* Taylor and Francis. Pennsylvania, 1998.

[6] H. Richter, *Hyperplane Design in Observer-Based Discrete Sliding Mode Control*, Masters Thesis. Oklahoma State University, 1997.

[7] C.Y. Tang, *Discrete Variable Structure Control for Uncertain Linear Multivariable Systems*, Masters Thesis. Oklahoma State University, 1997.

[8] C.Y. Tang, "Sliding Surface Design for Discrete VSS Using LQR Technique with a Preset Real Eigenvalue," *Proceedings of the American Control Conference*, (San Diego, CA), pp. 520-524, June 1999.

[9] H. Sira-Ramirez, "Non-linear Discrete Variable Structure Systems in Quasi-Sliding Mode," *International Journal of Control*, vol. 54, no. 5, pp. 1171-1187, 1991.

[10] G. Franklin, J. Powell and M. Workman, *Digital Control of Dynamic Systems.* Addison-Wesley, Massachusetts, 1990.

[11] E.A. Misawa, Classnotes: MAE 5723 Nonlinear Control. Oklahoma State University, 1998.

[12] A. Isidori, *Nonlinear Control Systems.* Springer, New York, 1995.

[13] Milosavljevic, "General Conditions for the Existence of a Quasi-Sliding Mode on the Switching Hyperplane in Discrete Variable Structure Systems," *Automation and Remote Control*, vol. 46, no3, pp. 207-214, 1985.

[14] W. Su, S. Srakunov and U. Ozguner. "Discrete-Time Sliding Mode Control in Nonlinear Sampled-Data Systems," *Proceedings of the 31st Annual Allerton Conference on Communication, Control and Computing*, (Champaign-Urbana, IL), pp. 809-818, 1993.

[15] J.K. Pieper and K.R. Goheen, "Discrete-Time Sliding Mode Control via Input-Output Models," *Proceedings of the American Control Conference*, pp. 964-965, 1993.

[16] R. Paden and M. Tomizuka, "Variable Structure Discrete Time Position Control," *Proceedings of the American Control Conference*, pp. 959-963, 1993.

[17] K. Furuta, "Sliding Mode Control of a Discrete System," *Systems & Control Letters*, vol. 14, pp. 145-152, 1990.

[18] K. Furuta and Y. Pan, "VSS Controller Design for iscrete-Time Systems," *Control-Theory and Advanced Technology*, vol. 10, no. 4, pp. 669-687, 1994.

[19] S.Z. Sparpturk, et al., "On the Stability of Discrete-Time Sliding Mode Control," *IEEE Transactions on Automatic Control*, vol. 32, no. 10, pp. 930-932, 1987.

[20] U. Kotta, "Comments of the Stability of Discrete-Time Sliding Mode Control," *IEEE Transactions on Automatic Control*, vol. 34, no. 10, pp. 1021-1022, 1989.

[21] E.A. Misawa, "Discrete-Time Sliding Mode Control: The Linear Case," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 119, no. 3, pp. 503-512, 1997.

[22] R.T. Lyle and E.A. Misawa, "Discrete Time Variable Structure Control with Disturbance Observer for Disk Drives," 9^{th} *IEEE International Conference on Control Applications*, (Anchorage, Alaska), 2000.

[23] N. Kudva, A. Viswanadham and Ramakrishna, "Observers for Linear Systems with Unknown Inputs," *IEEE Transactions on Automatic Control*, vol. 25, pp. 113-115, 1980.

[24] P. Hou and C. Muller, "Design of Observers for Linear Systems with Unknown Inputs," *IEEE Transactions on Automatic Control*, vol. 37, pp. 871-874, 1992.

[25] G. Meditch and H. Hostetter, "Observers for Systems with Unknown and Inaccessible Inputs," *International Journal of Control*, vol. 19, pp. 473-480, 1974.

[26] W. Wang, P. Davison and P. Dorato, "Observing the States of Systems with Unmeasurable Disturbances," *IEEE Transactions on Automatic Control*, pp. 716-717, 1975.

[27] R. Yang and Wilde, "Observers for Linear Systems with Unknown Input," *IEEE Transactions on Automatic Control*, vol. 33, pp. 677-681, 1988.

[28] J. Park and Stein, "Closed-loop, State and Input Observer for Systems with Unknown Inputs," *International Journal of Control*, vol. 48, pp. 1121-1136, 1988.

[29] J. Hostetter and S. Meditch, "Observing Systems with Unmeasurable Inputs," *IEEE Transactions on Automatic Control*, pp. 307-308, 1973.

[30] C. Johnson, "Accommodation for External Disturbances in Linear Regulator and Servomechanism Problems," *IEEE Transactions on Automatic Control*, vol. 16, pp. 635-644, 1971.

[31] J.M. Maciejowski, "Asymptotic Recovery for Discrete-Time Systems," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 602-605, 1985.

[32] Z. Zhang and J.S. Freudenberg, "On Discrete Time Loop Transfer Recovery," *Proceedings of the American Controls Conference*, (Boston, MA), pp. 2214-2219, 1993.

[33] K. Zhou and J. Doyle, *Essentials of Robust Control.* Prentice Hall, NJ, 1998.

[34] M. Tadjine, M. M'Saad and L. Dugard, "Discrete-Time Compensators with Loop Transfer Recovery," *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1259-1262, 1994.

[35] T. Ishihara and H. Takeda, "Loop Transfer Recovery Techniques for Descrete-Time Optimal Regulators Using Predictions Estimators," *IEEE Transactions on Automatic Control*, vol. 31, no. 12, pp. 1149-1151, 1986.

[36] M.J. Lopez and F.R. Rubio. "LQG/LTR Control of Ship Steering Autopilots," *IEEE International Symposium on Intelligent Control*, pp. 447-450, 1992.

[37] S. Weerasooriya and D.T. Phan, "Discrete-Time LQG/LTR Design and Modeling of a Disk Drive Actuator Tracking Servo System," *IEEE Transactions on Industrial Electronics*, vol. 42, no. 3, pp. 240-247, 1995.

[38] X. Hu, W. Guo,T. Huang and B. Chen, "Discrete-Time LQG/LTR Dual Stage Control Design and Implementation for High Track Density HDD's," *Proceedings of the American Controls Conference*, (San Diego, CA), pp. 4111-4115, 1999.

[39] J-K. Chang and H. Ho. "LQG/LTR Frequency Loop Shaping to Improve TMR Budget," *IEEE Transaction on Magnetics*, vol. 35, no. 5, p. 12, 1999.

[40] G.C. Goodman, *The LQG/LTR Method and Discrete-Time Control Systems.* Masters Thesis, Massachuetts Institute of Technology, Cambridge, Massachusetts, 1984.

[41] J.C. Doyle and G. Stein, "Robustness with Observers," *IEEE Transactions on Automatic Control*, vol. 24, no. 4, pp. 607-611, 1979.

[42] J.C. Doyle and G. Stein, "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 4-16, 1981.

[43] M. Athans, "A Tutorial on the LQG/LTR Method," *Proceedings of the American Control Conference*, pp. 1289-1296, 1986.

[44] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems.* Wiley-Interscience, New York, 1972.

[45] B.D.O Anderson and J. Moore, *Optimal Control: Linear Quadratic Methods.* Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[46] B.D. O'Dell, "Obtaining Uniform Singular Values," *Proceedings of the American Controls Conference*, pp. 461-465, 1995.

[47] Y.S. Hung and A.G. MacFarlane, "Multivariable Feedback: A Quasi-classical Approach," *Lecture Notes in Control and Information Sciences*, vol. 40, Springer-Verlag, Berlin, 1982.

[48] J.M. Maciejowski, *Multivariable Feedback Design*, Addison-Wesley Publishing, Reading, MA., 1989.

[49] M.G. Safonov, A.J. Laub and G.L. Hartmann, "Feedback Properties of Multivariable Systems: The Role and Use of the Return Difference Matrix," *IEEE Transactions on Automatic Control*, vol. 26, pp. 47-65, 1981.

[50] Goh, Z. Li, B. Chen, T. Lee and T. Huang, "Design and Implementation of a Hard Disk Drive Servo System Using Robust and Perfect Tracking Approach," *IEEE Conference on Decision & Control*, (Phoenix, AZ), pp. 5247-5252, 1999.

[51] S.H. Lee, S.E. Baik and C.C. Chung, "Design of A Servomechanism with Sliding Mode for A Disk Drive Actuator," *IEEE Conference on Decision & Control*, (Phoenix, AZ), pp. 5253-5258, 1999.

Appendix A

MATLAB Setup files for Example 2.1

A.1 Main Setup File

```
% This m-file sets up the OBDVSC with Disturbance Observer Example
% for the comparison of OBDVSC with bias and dist. state modelling
%
% Design plant taken from Goh, et. al.
% m-file written by R. Todd Lyle
% Advanced Controls Laboratory
% Oklahoma State University
% Stillwater, OK
%

% Initially clear the workspace
clear, clc
disp('workspace cleared to setup for OBDVSC simulations....')

% Setup the mechanical model (gives A,B,C,D)
%
% Setup mechanical model from Goh et. al.
mnum_goh=[4.3817e10 4.3247e15];
mden_goh=[1 1.5962e3 9.7631e7 0 0];
```

```
%state space representation of Goh
[A,B,C,D]=tf2ss(mnum_goh,mden_goh);

Ap=A; Bp=B; Cp=C;

% Physical parameters
tpi=17500;              % tracks per inch
ts=40e-6;               % samples per second

% Specify the simulation conditions
rpm=7200;               % spindle angular speed in RPM
tstop=0.01;             % simulation stop time milliseconds

% Discretize the mechanical model
[A,B]=c2d(A,B,ts);

% switches for simulation conditions
ON=1; OFF=0;

dist_switch=OFF;  % disturbance switch
reg_switch=ON;    % reg_switch=ON=follow, reg_switch=OFF=1 trk seek
bode_switch=OFF;  % plot bode diagram of HDA models

%Bode diagram compare for the three models
if bode_switch==ON
w=logspace(2,6,10000);
figure, bode(mnum_goh,mden_goh,w), hold
bode(mnum_lee,mden_lee,w), bode(mnum_seg,mden_seg,w)
end

% OBDVSC design for Goh et. al.
  cpole=[4.611643777792891e-001+8.682636351410205e-001i
         4.611643777792891e-001-8.682636351410205e-001i
         9.016391355195683e-001
         7.709338644383628e-001];
  lambda=cpole(4);
  phi=.1;         % boundary layer thickness
```

```
% OBDVSMC design based on 4th order model
 K=phi*(1-lambda);      % sliding gain
 F=place(A,B,cpole);    % controller feedback gain matrix
 %F=cpole_map(A,B,F);
 I=eye(4);
 G=null((A-B*F-lambda*I)')';    % sliding surface
 G=G(size(G,1),:);
 Aeq=A-B*G/(G*B)*(A-(1-K/phi)*I);

 Ed=inv(G*B);

% Trajectory shaping & generation based on position in inches
sp=1/tpi ;
wb=366;
m=0.2;
tref=2*sqrt(sp)/wb;
zeta=0.96;
wn=4000/(zeta*m);
tau1=1/(10*wn*zeta);
tau2=1/(10*wn*zeta);
den=conv([1 2*zeta*wn wn^2],[tau1*tau2 tau1+tau2 1]);
Fd=acker(A,B,exp(roots(den)*ts));
pref=C*inv(I-A+B*Fd)*B;

% trajectory generation based on position in tracks
sp=1;     % set sp=1/tpi for position in inches

% Augmented observers
% bias estimator design

% augmented state for bias estimator
Aaug=[A B;zeros(1,4) 1];
Baug=[B;0];
Caug=[C 0];

% Maximum augmented state for scaling using scale.m
max=[2e-3 5e-8 2e-12 4e-16 2e-2]';
```

```
% scale the augmented system by max values
[As,Bs,Cs,M]=scale(Aaug,Baug,Caug,max);
L=inv(M);

% Observer gain matrix for bias estimator
% bias estimator gains using pole placement
 Hs=[3.075526465608720e+000
    1.034766225843390e+001
    1.197031939632046e+001
    1.893023985993461e+000
    3.167896383124237e+003];

% disturbance state modelling design
% Setup the windage disturbance model (call dist_state.m)
dist_state

% Discretize windage state space for observer augmentation
[Ad,Bd]=c2d(Awin,Bwin,ts);
Cd=Cwin;
Dd=Dwin;

% Define Augmented systems with disturbance model (discrete time)
Aaug=[A B*Cd;zeros(4,4) Ad];
Baug=[B B*Dd;zeros(4,1) zeros(4,1)];
Caug=[C zeros(1,4)];

% Maximum augmented state for scaling using scale.m
max2=[2e-3 5e-8 2e-12 4e-16 2e-6 2e-11 5e-16 4e-21]';

% scale the augmented system by max values
[As2,Bs2,Cs2,M2]=scale(Aaug,Baug,Caug,max2);
L2=inv(M2);

% Observer gain matrix for disturbance state
Gamma=diag([0 0 0 1 1 1 1e2 1e2]);
Q=diag([1 1 1 1e1 1e40 1e18 1e20 1e25]);
R=1;
```

```
[T,P,Z,E] = dlqe(As2,Gamma,Cs2,Q,R);
Hs2=As2*T;
```

A.2 Setup File for Disturbance Model.

```
% This m-file sets up the disturbance model for the
%  disturbance state modelling observer augmentation to be
%  internally called by setup.m
%
%  written by R. Todd Lyle

%variation of actual disturbance model (robustness)
percent_f1=.8;  % change in 1st peak freq (e.g. 1.1=10%change)
percent_f2=1.2; % change is 2nd peak freq
percent_z=1.2;  % change damping all poles and zeros

%First peak
wn1=600*2*pi*percent_f1;
zetapole1=0.008*percent_z;

h1num=[1];
h1den=[1 2*zetapole1*wn1 wn1^2];

%Between peaks

zetazero1=0.2*percent_z;
zwn1=750*2*pi*percent_f1;

hzeros1=[1 2*zetazero1*zwn1 zwn1^2];

%Second peak
wn2=1200*2*pi*percent_f2;
zetapole2=0.006*percent_z;

h2num=[1];
h2den=[1 2*zetapole2*wn2 wn2^2];

%zeros after peaks
%zwn2=1400*2*pi*percent_f2;
%zetazero2=0.7*percent_z;
%hzeros2=[1 2*zetazero2*zwn2 zwn2^2];
```

```
%Overall Transfer function and state space (C-T)
Ka=wn1^2*wn2^2/(zwn1^2);

windage_num=Ka*hzeros1;
windage_den=conv(h1den,h2den);
[Awin,Bwin,Cwin,Dwin]=tf2ss(windage_num,windage_den);

%Frequency range for bode plot of windage transfer function
ON=1;
OFF=0;
bode_plot=OFF;
if bode_plot==ON;
   w=logspace(2,5,10000);
   bode(windage_num,windage_den,w);
   hold
end
```

Appendix B

MATLAB Setup files for Example 3.4

B.1 Setup File for Continuous Time LQG/LTR Example of Section 3.4

```
%  This m-file sets up the design plant for MIMO LQG/LTR continuous
%  time example
%
%  Design plant taken from Maciejowski's appendix
%  m-file written by R. Todd Lyle
%
% Design criteria
% 1) bandwidth of 10 rad/sec
% 2) good damping of step responses
% 3) zero steady-state error with step demands & disturbances

% Initially clear the workspace
clear, clc

% Remotely Piloted Vehicle (RPV, from Maciejowski) in C-T
A=[-0.02567      -36.6170 -18.8970  -32.0900         3.2509    -0.76257;
  9.257*10^-5  -1.8997    0.98312  -7.256*10^-4  -0.1708   -4.965*10^-3;
  0.012338      11.720    -2.6316   8.758*10^-4  -31.6040    22.3960;
  0             0          1         0              0          0;
  0             0          0         0            -30.0000     0;
  0              0       0        0              0         -30.0000];

B=[zeros(4,2);30*eye(2)];
C=[0 1 0 0 0 0;
   0 0 0 1 0 0];
D=[0 0;0 0];

% Augment Design Plant with free integrators one each channel
Aaug=[zeros(2,8); B A];
Baug=[eye(2);zeros(6,2)];
Caug=[zeros(2,2) C];
Daug=[zeros(2,2)];

% Target loop design for recovery at the plant output
% Loop shaping for the target loop
Llow=-inv(C*inv(A)*B);      % matching singular values @ low freq
Lhigh=C'*inv(C*C');         % matching singular vaules @ high freq

% O'Dell matching singular values (comment out if using prior design)
Llow=-inv(C*inv(A)*B);
Lhigh=inv(A)*B*inv(C*inv(A)*B);

L=[Llow;Lhigh];
Qlqe=L*L';

mu=(1/10)^2;                % select desired crossover freq 10 rad/sec
R=mu*eye(2);
[H,P,E]=lqe(Aaug,eye(size(Aaug)),Caug,Qlqe,R);

figure, hold
w=logspace(-3,5,100);
sigma(Aaug,H,Caug,Daug,w)  % the target loop SV plot

% Loop Transfer Recovery
Qlqr=Caug'*Caug;
recovery=[1 5 10 15];
s=size(recovery);
```

```
for i=1:s(:,2)

rho=1*10^(-recovery(i));
Rlqr=rho*eye(2);
[F,S,E]=lqr(Aaug,Baug,Qlqr,Rlqr);

% Transfer function of the model based compensator
Ak=Aaug-Baug*F-H*Caug;
Bk=-H;
Ck=-F;

At=[Aaug Baug*Ck;zeros(size(Aaug)) Ak];
Bt=[zeros(size(H));Bk];
Ct=[Caug zeros(size(H'))];
Dt=zeros(2,2);
sigma(At,Bt,Ct,Dt,w)        %Singular Value plot of MBC
end
```

B.2 Setup File for Discrete Time LQG/LTR Example of Section 3.4

```
%  This m-file sets up design plant for MIMO LQG/LTR discrete time
%  example
%
%  Design plant taken from Maciejowski's appendix
%  m-file written by R. Todd Lyle
%
% Design criteria
% 1) bandwidth of 10 rad/sec
% 2) good damping of step responses
% 3) zero steady-state error with step demands and disturbances


% Initially clear the workspace
clear, clc

% Remotely Piloted Vehicle (RPV, from Maciejowski) in continuous time
A=[-0.02567      -36.6170 -18.8970  -32.0900        3.2509   -0.76257;
 9.257*10^-5  -1.8997   0.98312  -7.256*10^-4  -0.1708    -4.965*10^-3;
 0.012338     11.720   -2.6316   8.758*10^-4  -31.6040    22.3960;
 0            0         1         0             0          0;
 0            0         0         0           -30.0000     0;
 0            0         0         0             0        -30.0000];

B=[zeros(4,2);30*eye(2)];
C=[0 1 0 0 0 0;0 0 0 1 0 0];
D=[0 0;0 0];

% Discretize the plant model with sampling time of ts=0.0001
ts=.0001;              %sampling time
[A,B]=c2d(A,B,ts);


% Augment Design Plant with free integrators per channel
Aaug=[zeros(2,8); B A];
```

```
Baug=[eye(2);zeros(6,2)];
Caug=[zeros(2,2) C];
Daug=[zeros(2,2)];
[Aaug_discrete,Baug_discrete]=c2d(Aaug,Baug,ts);

% Target loop design for recovery at the plant output
% Loop shaping for the target loop
Llow=-inv(C*inv(A)*B);          % matching singular values @ low freq
Lhigh=C'*inv(C*C');             % matching singular vaules @ high freq

% O'Dell matching singular values (comment out if using prior design)
Llow=-inv(C*inv(A)*B);
Lhigh=inv(A)*B*inv(C*inv(A)*B);

L=[Llow;Lhigh];
Qlqe=L*L';

mu=(1/10)^2;                % select desired crossover freq 10 rad/sec
Rlqe=mu*eye(2);
[H,P,E]=lqed(Aaug,eye(size(Aaug)),Caug,Qlqe,Rlqe,ts);

figure, hold
w=logspace(-3,5,100);
dsigma(Aaug_discrete,H,Caug,Daug,ts,w) % the target loop SV plot

% Loop Transfer Recovery
Qlqr=Caug'*Caug;
recovery=[1 5 10 15];
s=size(recovery);
for i=1:s(:,2)

rho=1*10^(-recovery(i));
Rlqr=rho*eye(2);
[F,S,E]=lqrd(Aaug,Baug,Qlqr,Rlqr,ts);

% Transfer function of the model based compensator
Ak=Aaug_discrete-Baug_discrete*F-H*Caug;
```

```
Bk=-H;
Ck=-F;

At=[Aaug_discrete Baug_discrete*Ck;zeros(size(Aaug)) Ak];
Bt=[zeros(size(H));Bk];
Ct=[Caug zeros(size(H'))];
Dt=zeros(2,2);
dsigma(At,Bt,Ct,Dt,ts,w)        %Singular Value plot of MBC
end
```

Appendix C

MATLAB Setup files for Chapter 5

C.1 Main Setup File

```
% This is MATLAB m-file script.m
% Initialization script for DVSC LQG/LTR
% written by R. Todd Lyle, June 1, 2000
% Advanced Control Laboratory
% Oklahoma State University

% Initialization script for DVSC LQG/LTR
clear, clc

% Disk drive physical parameters
ts=40e-6;                            % Sampling period, ms
mu=0.0001;                           % Friction coefficient
J=0.00005;                           % Actuator inertia
Kt=1000.0;                           % Torque constant
r=2.0;                               % Radius inches
tpi=17500;                           % TPI, track per inch

% Setup mechanical model from Goh et. al.
mnum_goh=[4.3817e10 4.3247e15];
mden_goh=[1 1.5962e3 9.7631e7 0 0];

% Resonance parameters
wn_res=600;
zeta_res=.3;
```

```
c=inv(wn_res^2);
d=2*zeta_res/wn_res;

c=inv(9.7631*10^7); %uncomment to match Goh model in frequency
d=(1.5962*10^3)*c;  %uncomment to match Goh model in frequency

% 4th order mechanical plant model for a,b=0
Acont=[0 1 0 0; 0 0 1 0; 0 0 0 1; 0 -mu/(J*c) -1/c -d/c];
Bcont=[0 0 0 Kt*r/(J*c)]';
Ccont=[1 0 0 0];
C=Ccont;

%discretize the system with sample time of ts
[A,B]=c2d(Acont,Bcont,ts);

% Specify the simulation conditions
sp=1/tpi ;        % set point at 1/tpi (single track seek)
rpm=7200;         % spindle angular speed in RPM
tstop=0.01;       % simulation stop time milliseconds

% Simulation trajectory generation (if needed)
wb=266;
m=0.4;

% switches for simulation conditions
ON=1; OFF=0;
reg_switch=ON;

% OBDVSC design via lqg/ltr (call script.m)
 script
```

C.2 Filter Loop Design and Recovery

```
% This is MATLAB m-file script.m
% To be called internally by setup_lqgltr.m
% written by R. Todd Lyle, June 1, 2000
% Advanced Control Laboratory
% Oklahoma State University

% OBDVSC design via lqg/ltr

%Frequency vector for singular value plots
w=logspace(1,6,700);

% Target filter loop design observer using dlqe 4th order
[T,P,Z,E] = dlqe(A,diag([.5e-1 0 0 0]),C,diag([1e0 0 0 0]),1);
 H=A*T;

% Calculate and plot singular values for target filter loop
 [Atest1,Btest1,Ctest1,Dtest1]=dlinmod('test1');
 dsigma(Atest1,Btest1,Ctest1,Dtest1,ts,w), hold

% Hyperplane Design using Loop Transfer Recovery

% fixed parameters of OBDVSC
alp=-9.2e-1;      % real eigenvalue
phi=1e0*ts;       % boundary layer thickness

% calculate other constraints (K, gamma, epsilon)
K=phi*(1-alp);   % sliding gain

% Calculate gamma and epsilon (fixing alpha, ts, and phi)
% Display warning message if gamma > 0 or epsilon > 0 is violated

M=inv([1 2*ts;1 ts])*[phi*(1-alp); phi];
if M(1,:) <= 0
 disp('Gamma < 0, sliding matrix G may be invalid')
end
if M(2,:) <= 0
```

```
 disp('Epsilon < 0, sliding matrix G may be invalid')
end

%Identity Matrix size of A
I=eye(4);

%Rho values to be recovered
test_vector=[14 16 17];

% LTR procedure (loop running test2 "j" times)
for j=1:3
   rho=1*10^(test_vector(j))
   Qd=rho*C'*C;
   [q,F,G]=dvsclqr1(A,B,Qd,alp);
   Aeq=A-B*G/(G*B)*(A-(1-K/phi)*I);

  % Trajectory shaping & generation (if needed)
  tref=2*sqrt(sp)/wb;
  zeta=0.96;
  wn=4000/(zeta*m);
  tau1=1/(10*wn*zeta);
  tau2=1/(10*wn*zeta);
  den=conv([1 2*zeta*wn wn^2],[tau1*tau2 tau1+tau2 1]);
  Fd=acker(A,B,exp(roots(den)*ts));
  pref=C*inv(I-A+B*Fd)*B;

  %Calculate and plot singular value for compensator loop
  [Atest2,Btest2,Ctest2,Dtest2]=dlinmod('test2');
  dsigma(Atest2,Btest2,Ctest2,Dtest2,ts,w);

 end
```

VITA

Ryan Todd Lyle

Candidate for the Degree of

Master of Science

Thesis: HYPERPLANE DESIGN FOR DISCRETE VARIABLE STRUCTURE CONTROL WITH LOOP TRANSFER RECOVERY

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Tulsa, Oklahoma, On December 2, 1974, the son of Richard and Deborah Lyle.

Education: Attended Plantation High School, Plantation, Florida, from August 1989 until June 1992. Graduated Bixby High School, Bixby, Oklahoma in May 1993; received Bachelor of Science degree in Mechanical Engineering from Oklahoma State University, Stillwater, Oklahoma in May 1998. Completed requirements for Master of Science degree with a major in Mechanical Engineering at Oklahoma State University in December 2000.

Experience: Lew Wentz scholar as undergraduate researcher, Spring 1998; Oklahoma State University, Employed by Advanced Controls Laboratory for graduate level industrial research, Summer 1998 until Summer 2000; Oklahoma State University, School of Mechanical and Aerospace Engineering, Employed as Development Engineer, Summer/Fall 2000; Schlumberger Oilfield Services (OFS), North America, Schlumberger Reservoir Completions, Rosharon, Texas, Currently employed as Deveopment Engineer, Seagate Technologies, Oklahoma City, Oklahoma.

Professional Memberships: Pi Tau Sigma, Golden Key National Honor Society, American Society of Mechanical Engineers.