Discrete-time PID Controller Tuning Using Frequency Loop-Shaping

by

Md. Ashfaque Bin Shafique

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2011 by the
Graduate Supervisory Committee:

Konstantinos Tsakalis, Chair
Jennie Si
Armando Rodriguez

ARIZONA STATE UNIVERSITY

December 2011

ABSTRACT

Proportional-Integral-Derivative (PID) controllers are a versatile category of controllers that are commonly used in the industry as control systems due to the ease of their implementation and low cost. One problem that continues to intrigue control designers is the matter of finding a good combination of the three parameters – P, I and D of these controllers so that system stability and optimum performance is achieved. Also, a certain amount of robustness to the process is expected from the PID controllers. In the past, many different methods for tuning PID parameters have been developed. Some notable techniques are the Ziegler-Nichols, Cohen-Coon, Astrom methods etc. For all these techniques, a simple limitation remained with the fact that for a particular system, there can be only one set of tuned parameters; i.e. there are no degrees of freedom involved to readjust the parameters for a given system to achieve, for instance, higher bandwidth. Another limitation in most cases is where a controller is designed in continuous time then converted into discrete-time for computer implementation. The drawback of this method is that some robustness due to phase and gain margin is lost in the process. In this work a method of tuning PID controllers using a loop-shaping approach has been developed where the bandwidth of the system can be chosen within an acceptable range. The loop-shaping is done against a Glover-McFarlane type $\mathcal{H}_\infty$ controller which is widely accepted as a robust control design method. The numerical computations are carried out entirely in discrete-time so there is no loss of robustness due to conversion and approximations near Nyquist frequencies. Some extra degrees of freedom owing to choice of bandwidth and capability of choosing

loop-shapes are also involved and are discussed in detail. Finally, comparisons of this method against existing techniques for tuning PID controllers both in continuous and in discrete-time are shown. The results tell us that our design performs well for loop-shapes that are achievable through a PID controller.

DEDICATION

To Samiha, for her unending love, continuous support and patience during my long, incessant days of work.

ACKNOWLEDGMENTS

I owe my gratitude to several people who have advised, assisted, encouraged or inspired me during the course of this work.

First of all, I would like to sincerely thank my advisor Dr. Konstantinos Tsakalis who so patiently guided me throughout my research that culminated into this thesis. I have expended many of his office hours each day, discussing problems that I faced and cannot remember any instance where he would not take ample time in explaining to me the intricacies of control theory. His teachings have deeply impacted my knowledge and skills and I feel honored to have been under his mentorship.

I am greatly indebted to Dr. Armando Rodriguez for the effort he took to explain to me the complicated theories even after class time was over, just so I left with a better understanding of the topics he covered in class. I am thankful to Dr. Jennie Si for teaching me about neural networks, which I found profoundly invigorating, and also for serving on my thesis committee.

My research work and the writing of this thesis would not have been complete without the enormous support of my family, especially my mother and father. From the bottom of my heart, I would like to extend my love and gratitude to them for nurturing me throughout the years through many sacrifices. Last, but not least, my deepest gratitude to my friends and well-wishers for their encouragement.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 PID controllers

Proportional-Integral-Derivative controllers (PID controllers) are the most common form of control systems in use today. Owing to the fact that they have only three terms to control their input-output behavior, their implementation is very simple and can be done even without the use of sophisticated microcontrollers/microcomputers. Although simple in structure, their field of applicability is quite versatile and this is the primary reason behind their widespread use in the industry. Some examples of their use are found in control of temperatures in chemical processes, control of angular position of arm like structures (robotic arms), controlling rotation speed of motors, controlling liquid levels and pressures etc. The term PID comes from the structure of the controller itself, where the output (plant input) signal of the controller is computed as a sum of the input (error) multiplied with a proportional gain K_P, an integral of the input times the integral gain K_I and a derivative of the input times a derivative gain K_D. Fig. 1 shows a schematic of the flow of signals through a PID controller.

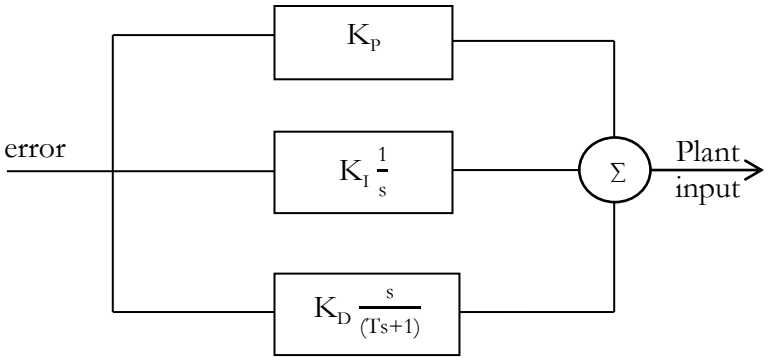

Fig 1: Schematic representation of PID controller.

$$C(s)=K_P+K_I\frac{1}{s}+K_Ds \quad (1)$$

Equation (1) shows the transfer function of the controller in continuous time. Although, for computational purposes and for practical implementation it is a general practice to put a pseudo-pole of the form Ts+1, as a denominator to the derivative term, to ensure that the frequency response rolls-off at high frequency. Thus, the transfer function takes the form shown in equation (2).

$$C(s)=K_P+K_I\frac{1}{s}+K_D\frac{s}{(Ts+1)} \quad (2)$$

The most challenging aspect in designing a PID controller is in tuning the three gains K_P, K_I and K_D. The proportional gain determines how fast the controller will react to the error input, in other words too low a value will make the controller react slowly but too high of a value may make the system unstable. The integral gain compensates for accumulated errors and thus determines overshoot. The derivative gain slows the overshoot but is very sensitive to noise and can cause the system to become unstable due to it. Given these characteristics of each of the gains it is desirable to have a controller that will make the system stable and still produce fast responses and have some robustness properties.

1.2 Literature review

Despite the recent advancements in control theory that allows for design and implementation of highly sophisticated controllers, simple PID controllers are still preferred in the industry. Added to the simplicity is the fact that computational power has grown to a point where performing numerical computations to tune the

PID parameters are no longer a matter of concern. The only important step is to define a good tuning algorithm. In the past many methods have been developed to tune PID parameters. The manual method is where each of the gains are increased and decreased individually while the operator observes the behavior of the system until they are satisfied with the performance of the controller. Other notable methods are the Ziegler-Nichols method [1], Cohen-Coon method [2] and Astrom's method [3]. See O'Dawyer [40] for a detailed set of various PID and PI controller tuning techniques. While some of these methods rely on a tuning scheme based on reduced approximations of the system model others use some form of nonlinear optimization in comparison to some performance measure of the system in question. Although Ziegler-Nichols is widely popular due its computational simplicity and almost no requirement of *a priori* knowledge of the plant, it gives up on flexibility of conditions on the plant for a successful implementation and also has a lack of proper tuning "knobs" [4]. While all the above mentioned methods work under specific circumstances, none of them provide enough degrees of freedom for the operator to be able to adjust the parameters to improve on performance based on their judgment and expertise.

The lack of a unified tuning method that fits all needs can attributed to the dependency of the performance objective of the plant on its specific requirements in different applications. Also, in the case of PIDs, having only three parameters makes thing worse by allowing minimum wiggle room for adjustment. All the existing PID tuning methods have their strengths and weaknesses in terms of time to compute, simplicity in implementation, robustness properties etc. Comparing among methods

and trying to find the best one almost seems futile. With this in mind, this work tries to focus on specific aspects of PID tuning, namely robustness to modeling error, capability of choosing bandwidth for a system (adds a degree of freedom for the operator), ability to shape the loop of the PID to a plausible extent, ease and promptness of computation and the ability to incorporate this design technique with system identification techniques to learn about the plant/process, thus getting rid of or reducing the amount of trial and error involved in many other tuning methods.

In [5], we see how Voda and Landau show how to perform an automatic PID controller tuning using Kessler's symmetrical optimum method [6]. Their method addresses the robustness and closed-loop performance issues of electrical drives. This limits the use of the methods and performance of this method in complex industrial systems is therefore unknown. Similarly, Nudelman and Kulesky in [7] have shown another PID tuning scheme for the specific purpose of controlling power–station loops. Kristiansson *et al* [8] have shown methods for optimal PI and PID parameter tuning by utilizing the fact that during optimization, to find the parameters, the high frequency pole must be incorporated for the tuned PI(D) to be robust. In [9], Malan *et al* have shown another method for robust tuning of PID controllers but with multiple performance specifications. Their method was to make use of a convergent set of inner and outer approximations of the parameters that will allow the system to perform robustly to the design. Tesi and Vicino, in [10], shows another method of designing robust controllers that are optimal and have a few degrees of freedom.

More recent work done by Tsakalis et al [11] [4] [12] [13] uses methods where the tuning is performed by comparing the loop-shape of a Linear Quadratic Regulator (LQR) based controller to that of a PID to obtain the parameters. Our work derives its roots mostly from the research done in [4]. However, in this work the loop-shaping has been done in comparison to a Glover-McFarlane $\mathcal{H}_\infty$ controller [14] instead of an LQR controller. In many instances, it is nowadays a common practice to not derive a model of the system via first principles but from system identification methods as is done in [13] and [15]. This work, however, assumes a model is available either from first principles or system identification methods and proceeds from there on. This allows our design to be easily integrated into methods that involve identifying the system computationally via input output data as was our original goal.

Some of the methods for system identification in [16] [17] [18] [19] have been studied prior to the commencement of this work. Mostly due to their relevance to what we are trying to achieve, i.e. identification of a system for use in PID controller design. But, since identification is not the primary concern of this project, we do not focus on it. Rather, we have done a thorough survey of the existing methods for designing $\mathcal{H}_\infty$ controllers both in continuous and discrete-time, since that is of a more fundamental interest to us.

In [20], Zhang *et al* have used Linear Matrix Inequalities (LMI) to find uncertainty bounds that are tolerable for robust $\mathcal{H}_\infty$ performance for discrete-time systems. Similarly, Crucius and Trofino have derived sufficient LMI conditions for $\mathcal{H}_\infty$ type output controllers [21]. In [22], we see how $\mathcal{H}_\infty$ controllers are designed for

linear discrete-time systems that have norm bounded nonlinear uncertainties. In [23], a method of designing $\mathcal{H}_\infty$ controllers with reduced orders is discussed. Glover and McFarlane also discuss a method of order reduction in [14]. Among other notable work done in discrete-time $\mathcal{H}_\infty$ design that are relevant to our work are the ones by Syrmos *et al* [24], Mirkin [25], Khargonekar *et al* [26], Francis *et al* [27] and Zhou and Doyle [28].

The most important aspect of this work lies in the fact that all the algorithms are built entirely in discrete-time. There are two methods of designing controllers:

1) First, design the control system in continuous-time (s-domain) and, then use a phase preserving Tustin transform to achieve the discrete-time (z-domain) form for computer implementation. (There are still some approximation errors in this method.)
2) Design the controller in discrete-time to start with (Begin with a plant that has been discretized either using a zero-order-hold (ZOH) method or a Bilinear transformation (Tustin) method).

Ignoring any intra-sample behavior for now, the major disadvantage of using method (1) is that some delay is introduced into the system during sampling of the plant that can lead to a loss in phase margin and gain margin, especially for closed-loop bandwidth close to Nyquist frequency. An additional advantage of the second approach is that it does not require an awkward conversion of pure discrete-time plants to continuous time first. Such is the case in the so-called "Run-to-Run" control problem that appears in semiconductor manufacturing where batch processing is the norm; see [41], [42], [43], [44]. The loop-shaping procedure is done

using the normalized coprime factorization $\mathcal{H}_\infty$ design methods of Glover and McFarlane, as shown in [14]. However, we have adopted the discrete-time version of the $\mathcal{H}_\infty$ control design, done by Walker [31].

1.3 Organization of the thesis

Chapter 2 discusses the formulation of the problem being investigated. How the need for a loop-shaping evolved and why $\mathcal{H}_\infty$ type controllers were chosen, particularly the Glover-McFarlane type. In the third Chapter the entire procedure for loop-shaping is shown, including algorithms for the discrete $\mathcal{H}_\infty$ controller, an LMI optimizer and some weight selection methods. Some results from the implementation of the $\mathcal{H}_\infty$ controller are also shown. Chapter 4 discusses the results obtained and some comparisons to other methods for PID tuning. The fifth chapter contains a recapitulation of the work done, conclusions obtained and suggestions for future work.

CHAPTER 2

PROBLEM FORMULATION

2.1 Introduction

All the existing methods for PID tuning rely on a sample input and the output the system generates in response to it; then computing the parameters based on some tuning rules set by these algorithms. However, once a set of parameters are determined for a particular system using one of these means, the control designer cannot attempt to achieve any different value using the same method if the original design did not perform well. Therefore, it results in a limitation in the degrees of freedom in the design. In this chapter we discuss the motivation behind our design idea, *viz.* a PID tuner that can be adjusted until a suitable design is found for a given plant.

2.2 Background

To get a good set of PID parameters for a process, there are some tuning methods that require knowledge of control systems to tune the parameters. This is a problem, since the operators of machinery in the industry are generally unaware of such techniques. Our goal was to design a tuner that will be operable by the everyday user thus hiding the intricate algorithms and presenting outputs that are readily usable. Essentially, the operator should put in the required bandwidth and the plant to the algorithm and it would spur out the three parameters K_P, K_I and K_D.

2.2 Formulation of the Problem

In this work our main focus is to design a PID tuner for linear, time-invariant Single-Input-Single-Output (SISO) plant/system. The assumption being held is that a model of the system has already been derived either from first principles or from system identification experiments. Fig. 2 shows the general form of a feedback control system loop.

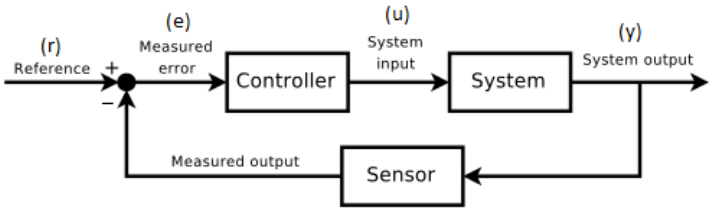

Fig 2: Schematic representation of a general feedback control system.

As depicted in the figure, the input to the controller is the error (e) between the reference signal and the output fed back through the sensor. The output of the controller is the plant input (u). So, the output "u" has signal flow:

$$U(z) = K(z) * E(z) \tag{3}$$

in discrete-time. For a system like this, the PID controller we are attempting to design will have the following transfer function:

$$K(z) = K_p + K_I \frac{T}{(z-1)} + K_D \frac{(z-1)}{Tz} \tag{4}$$

Where, K_P, K_I and K_D are the proportional, integral and derivative gains and T is the sampling time of the controller. The objective of our work is to determine the three PID gains such that the open loop transfer function (LTF) when compensated by the controller C(z) will be close in the sense of the $\mathcal{H}_\infty$ norm, to the one of a chosen

target loop transfer function. Let, the target open LTF be L(s). Alternatively, the structure for the controller can be rewritten like this:

$$K(z)= \frac{(K_1 z^2 + K_2 z + K_3)}{z(z-1)} \quad (5)$$

Where,

$$K_1 = K_P + K_D/T \quad (6)$$

$$K_2 = K_I T – K_P – 2K_D/T \quad (7)$$

$$K_3 = K_D/T \quad (8)$$

Thus, the PID parameters can be extracted from (6), (7) and (8) via the transformations:

$$K_p = K_1 – K_3 \quad (9)$$

$$K_i = 1/T(K_1 + K_2 + K_3) \quad (10)$$

$$K_d = K_3 T \quad (11)$$

Therefore, instead of tuning for the original PID parameters, we seek to tune for the coefficients of the numerator of K(z): K_1, K_2 and K_3. The alternate linearly parameterized form has the advantage that any functional of the form $\|W(PK - L)\|_{\mathcal{H}_\infty}$ is convex in the design parameters. "W" is a carefully selected weighting transfer function and "P" is the transfer function of the plant. It must be noted that to achieve internal stability "PK" should not contain any pole-zero cancellations outside the unit circle. The condition is easily met by restricting the scope of this controller design to minimum phase controllers. To ensure that minimality is observed, the following constraints are put on the coefficients of the numerator of K(z):

$K_1 > 0$, $K_2 > 0$, $K_3 > 0$; this ensures positivity, and

$K_1 + K_2 + K_3 > 0$, $K_1 - K_2 + K_3 > 0$, $K_3 < K_1$.

The last three conditions were derived from the Jury stability test criterion [29]. While, the first three ensure positivity of the gains the latter three do the same in addition to ensuring that the gains do not become large enough to render the system unstable. Before we move on to discussing how the PID tuning can be turned into a convex optimization problem we must probe a little further into computing how the closed-loop will have guaranteed stability.

Let, the error loop transfer function be $\Delta = L - PK$. Also, let us assume that PC has no pole-zero cancellations outside the unit circle. Furthermore, let us denote the nominal sensitivity of the closed loop system as $S_o \stackrel{\text{def}}{=} \frac{1}{1+L}$. Reformulating the expression of the closed-loop system in terms of L and Δ and then applying the small gain theorem [30] [39] to it; a sufficient condition for the closed-loop system can be written as

$$\|S_o \Delta\|_{\mathcal{H}_\infty} < 1, \tag{12}$$

$$\Rightarrow \left\| \frac{1}{1+L} (PK-L) \right\|_{\mathcal{H}_\infty} < 1 \tag{13}$$

The inequality (3) that follows from the application of the small-gain theorem on Δ and the sensitivity transfer function can be thought of as a cost functional for solving the weighted approximation problem of L by PK. By further inspection we can also see that

$$\|W(PK-L)\|_{\mathcal{H}_\infty} < 1 \tag{14}$$

where, W is any weighting transfer function that is stable and minimum phase given that:

$$\left|\frac{1}{1+L(z)}\right| \leq \|W(z)\|, \qquad \forall\omega \tag{15}$$

If we observe the characteristics of this expression in terms of the Nyquist plot we will see that this weighting function, W(z) can make the approximation around the crossover frequency stand out more prominently than just the stability requirement. If carefully selected, this weighting function can be made to bring about a reduction in the sensitivity transfer function peaking and subsequently add additional robustness constraints with respect to modeling errors. Therefore, the challenge of finding the PID parameters can be translated from a frequency loop-shape tuning into the optimization problem described below

$$\min_{K_1,K_2,K_3} \left\|W(PK_{K_1,K_2,K_3}-L)\right\|_{H_\infty} \tag{16}$$

For the method shown above, if we assume that P and S_o are stable and the open LTF, L, has an integrator for good command following properties, then it can easily be shown that the closed-loop stability will be guaranteed if the value of the minimum in (16) is less than 1. The only problem that remains to be solved is to find a good target LTF. In [4], the loop transfer function was selected by first designing an LQR controller for the plant then incorporating it into L. However, in this work, we will be selecting the loop based on an $\mathcal{H}_\infty$ controller designed for the plant.

2.3 Conclusion

In this chapter we saw how the problem has been developed and the proposed method for solving it. In the next chapter we will explore how the loop-shaping procedure has been carried out. We will also describe how we how we perform an LMI optimization to the loop-shape to obtain our PID parameters.

CHAPTER 3

LOOP-SHAPING AND OPTIMIZATION

3.1 Introduction

The choice of loop-shape for our algorithm could have been any open-loop that is stable when closed. That being said, a good loop-shape for most plants that are open-loop stable would be: $L(z) = \frac{Ta}{z-1}$; where, T is the sampling time and a is any constant gain. However, our goal was to make an algorithm that would be generic and should work well with plants that are even open-loop unstable. This is why our design incorporates the use of an $\mathcal{H}_\infty$ controller. Since, the numerical optimization method in equation (16) will try to minimize the difference between a norm of the loop-shape in L against the one in PK, where K is the PID controller we are searching for; the better behaved the loop-shape of L, the better behaved will be the PID controller performance found from the optimization. Although methods shown in [4] [11] [12] [13] work reliably with their use of LQR controllers for choice of loop-shape; LQR controllers are not robust to system modeling errors. Neither are they good with zeros far away from the system bandwidth; the matter can be worse with ones outside the unit circle. On the other hand, since $\mathcal{H}_\infty$ design produce output controllers, they have no such problem. In fact, the work done by McFarlane and Glover and D. Walker in [14] and [31] respectively, shows that: designed in a certain way, $\mathcal{H}_\infty$ controllers can be very robust to modeling errors up to a specified uncertainty region.

Traditional methods of $\mathcal{H}_\infty$ control design depend on an iterative procedure where the $\mathcal{H}_\infty$ norm of certain transfer functions involving the plant and some carefully selected weighting functions are minimized such that the controller, that is found as a result of the optimization in $\mathcal{H}_\infty$, is the minimum possible stabilizing solution. However, the iteration can be time consuming and so the work done by Glover and McFarlane shows methods in which the solution to the $\mathcal{H}_\infty$ optimization problem can be found in one shot. This is the primary motivation behind adapting their method into the discrete-time space [31] and using that as our loop-shape. It may be appropriate to restate now that doing an $\mathcal{H}_\infty$ control design may be very apt in this scenario; since, it is highly likely that the model of the plants we get are found from system identification methods which have a certain amount of modeling error in them. The same would be true if the plant state-space were found from first principles as well; but, in the industry it is more common for system identification methods to be used. The next section describes how the control design is achieved.

3.2 The $\mathcal{H}_\infty$ controller

All control systems are expected to have some common properties such as good tracking behavior and transient response. They are also expected to have some stability margins whilst modeling errors, parameter variations, noise and other forms of uncertainty are present in the state-space that describes the system. This has been a recognized problem and been studied for a long time by many: [12], [13], [14] and [15]. The studies have led to $\mathcal{H}_\infty$ design theory; where, unstructured model uncertainty can be described as additive or multiplicative, infinity-norm-bounded,

stable transfer functions that act on the nominal plant. Using $\mathcal{H}_\infty$ theory, it is often possible to reduce the closed-loop system's susceptibility to becoming unstable with respect to the perturbations to the system, while maintaining sufficiently satisfactory system performance.

The mechanism of incorporating modeling error suggested by Vidyasagar *et al* [30] is to represent them as a stable transfer function matrix (TFM) acting additively on each of the elements of a right or left coprime factorization on the nominal plant model.

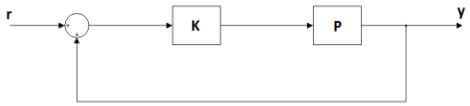

Fig 3: Schematic of unity feedback control system.

Let us first consider the feedback system shown in fig. 3. It has transfer function and state space:

$$\boldsymbol{P}(z) = \boldsymbol{D} + \boldsymbol{C}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right] \tag{17}$$

It also has the right coprime factorization

$$\mathbf{P} = \mathbf{N}\mathbf{M}^{-1} \tag{18}$$

We will assume that (**A, B**) is stabilizable and (**A, C**) is detectable in which case we have the well-known realizations for the transfer function matrices satisfying equation (18) to be:

$$\begin{bmatrix} \mathbf{M} & \mathbf{Y} \\ \mathbf{N} & \mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{A_F} & \mathbf{BZ_1} & \mathbf{-HZ_2^{-1}} \\ \mathbf{F} & \mathbf{Z_1} & \mathbf{0} \\ \mathbf{C_F} & \mathbf{DZ_1} & \mathbf{Z_2^{-1}} \end{bmatrix} \tag{19}$$

Where, **F** and **H** are the stabilizing state feedback and output injection matrices respectively; $\mathbf{A_F = A + BF}$, $\mathbf{C_F = C + DF}$ and $\mathbf{Z_1}$ and $\mathbf{Z_2}$ are arbitrary invertible matrices.

Let us focus on the case where the uncertainty in **P** is the transfer function of a stable 'perturbation' and is represented by Δ_1 and Δ_2, that act additively on each of the elements **N** and **M** of the right coprime factorization of **P**. Now, **P** is given by

$$\boldsymbol{P}_\Delta = (\boldsymbol{N} + \Delta_1)(\boldsymbol{M} + \Delta_2)^{-1} \tag{20}$$

Now, if we add the signals u_1, u_2 and y to fig. 1 and replace **P** with $\mathbf{P}_\Delta$ we get the loop shown below in fig. 4:

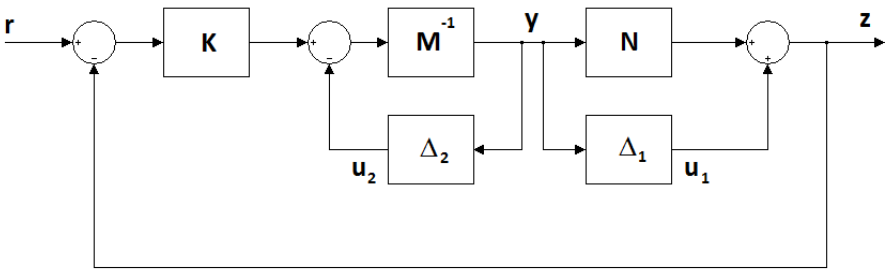

Fig 4: Schematic of unity feedback control system with **P** replaced by its uncertainty model.

The feedback loop equations for this system can be written as:

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} (M+KN)^{-1}[-K & -I] & (M+KN)^{-1}K \\ (1+PK)^{-1}[-I & P] & (I+PK)^{-1}PK \end{bmatrix} \begin{bmatrix} \{u_1 \\ u_2\} \\ r \end{bmatrix} \tag{21}$$

Where, $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \Delta y$ and $\Delta = \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix}$ (22)

Now, if we define the conjugate system of **P** as $\mathbf{P}^* = \mathbf{D}^T + \mathbf{B}^T(\mathbf{z}^{-1}\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{C}^T$. Then, for the state-space realization of P^* given in equation (23) will be all-pass if the conditions in (24) and (25) is true.

$$P^* = \left[\begin{array}{c|c} A^{-t} & A^{-t}C^t \\ \hline -B^tA^{-t} & D^t - B^tA^{-t}C^t \end{array}\right]$$, provided **A** is invertible (23)

If

$\mathbf{D}^t\mathbf{D}+\mathbf{B}^t\mathbf{Q}\mathbf{B} = \mathbf{I}$, $\mathbf{D}^t\mathbf{C}+\mathbf{B}^t\mathbf{Q}\mathbf{A} = \mathbf{0}$ and $\mathbf{Q}- \mathbf{A}^t\mathbf{Q}\mathbf{A}=\mathbf{C}^t\mathbf{C}$, then $\mathbf{P}^*\mathbf{P}=\mathbf{I}$ (24)

Similiarly, if

$\mathbf{D}\mathbf{D}^t+\mathbf{C}\mathbf{P}\mathbf{C}^t = \mathbf{I}$, $\mathbf{B}\mathbf{D}^t+\mathbf{A}\mathbf{P}\mathbf{C}^t = \mathbf{0}$ and $\mathbf{P}- \mathbf{A}\mathbf{P}\mathbf{A}^t=\mathbf{B}\mathbf{B}^t$, then $\mathbf{P}\,\mathbf{P}^* =\mathbf{I}$ (25)

In equations (24) and (25), **P** and **Q** are the solutions to the following Riccati equations:

$$\mathbf{B}\mathbf{R}_1^{-1}\mathbf{B}^t - \mathbf{P} + \mathbf{\Phi}\mathbf{P}\mathbf{\Phi}^t - \mathbf{\Phi}\mathbf{P}\mathbf{C}^t(\mathbf{R}_2+\mathbf{C}\mathbf{P}\mathbf{C}^t)^{-1}\mathbf{C}\mathbf{P}\mathbf{\Phi}^t = \mathbf{0} \quad (26)$$

$$\mathbf{C}^t\mathbf{R}_2^{-1}\mathbf{C} - \mathbf{Q} + \mathbf{\Phi}^t\mathbf{Q}\mathbf{\Phi} - \mathbf{\Phi}^t\mathbf{Q}\mathbf{B}\,(\mathbf{R}_1+\mathbf{B}^t\mathbf{Q}\mathbf{B})^{-1}\mathbf{B}^t\mathbf{Q}\mathbf{\Phi} = \mathbf{0} \quad (26)$$

Where,

$\mathbf{\Phi} = \mathbf{A} - \mathbf{B}\mathbf{R}_1^{-1}\mathbf{D}^t\mathbf{C}$,

$\mathbf{R}_1 = \mathbf{I} + \mathbf{D}^t\mathbf{D}$ and $\mathbf{R}_2 = \mathbf{I} + \mathbf{D}\mathbf{D}^t$

Given these matrices, it is straightforward to show that an optimal bound for the maximum robustly stabilizable uncertainty will be

$$\gamma_{opt} = \left[1 - \left\|\begin{matrix} N \\ M \end{matrix}\right\|_H^2\right]^{-1/2} \quad (27)$$

Following the definition of γ_{opt}, it has been shown by Walker [31] that there exists suboptimal controllers with bound $\gamma < \gamma_{opt}$ which will internally stabilize the super plant **G** as shown in fig. 5, where G has the state-space realization given by equation (28).

$$G(z) = \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array}\right] = \left[\begin{array}{cc|c} 0 & M^{-1} & M^{-1} \\ I & -P & -P \end{array}\right] \quad (28)$$

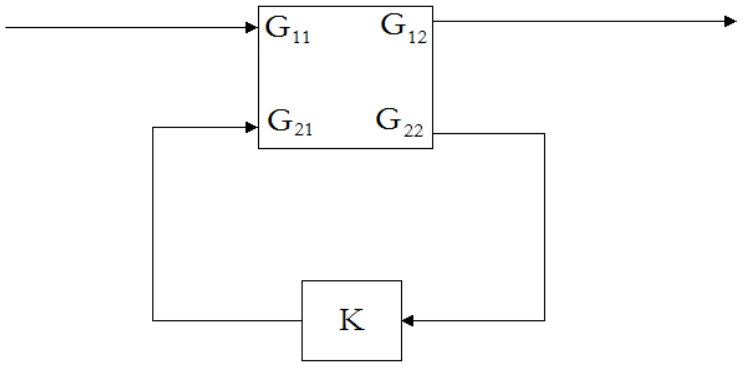

Fig 5: Feedback loop of super-plant **G** with **K**.

We are now ready to see how the structure of the controller was formulated by Limebeer et al [33]. These suboptimal controllers, with bound γ, are parameterized in terms of strong solutions to two indefinite algebraic Riccati equations. The first one in (29) has a solution $\mathbf{X}_\infty$ that is symmetric and positive definite

$$\mathbf{0} = \mathbf{C}_x^t\mathbf{J}\mathbf{C}_x+\mathbf{A}^t\mathbf{X}_\infty\mathbf{A}- \mathbf{X}_\infty-(\mathbf{C}_x^t\mathbf{J}\mathbf{D}_x+\mathbf{A}^t\mathbf{X}_\infty\mathbf{B}_x)(\widehat{\mathbf{D}}_x+\mathbf{B}_x^t\mathbf{X}_\infty\mathbf{B}_x)^{-1}(\mathbf{D}_x^t\mathbf{J}\mathbf{C}_x+\mathbf{B}_x^t\mathbf{X}_\infty\mathbf{A}) \quad (29)$$

Where,

$$\mathbf{B}_x=[\mathbf{B}_1 \quad \mathbf{B}_2],\ \mathbf{C}_x=\begin{bmatrix}\mathbf{C}_1\\ \mathbf{0}\end{bmatrix},\ \mathbf{D}_x=\begin{bmatrix}\mathbf{D}_{11} & \mathbf{D}_{12}\\ \mathbf{I} & \mathbf{0}\end{bmatrix},\ \widehat{\mathbf{D}}_x=\mathbf{D}_x^t\mathbf{J}\mathbf{D}_x,\ \mathbf{J}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & -\gamma^2\mathbf{I}\end{bmatrix} \quad (30)$$

And, the second Riccati in (31) has a solution $\mathbf{Y}_\infty$ that is also symmetric and positive definite

$$\mathbf{0} = \mathbf{C}_y^t\mathbf{J}\mathbf{C}_y+\mathbf{A}_y^t\mathbf{Y}_\infty\mathbf{A}_y- \mathbf{Y}_\infty-(\mathbf{C}_y^t\mathbf{J}\mathbf{D}_y+\mathbf{A}_y^t\mathbf{Y}_\infty\mathbf{B}_y)(\widehat{\mathbf{D}}_y+\mathbf{B}_y^t\mathbf{Y}_\infty\mathbf{B}_y)^{-1}(\mathbf{D}_y^t\mathbf{J}\mathbf{C}_y+\mathbf{B}_y^t\mathbf{Y}_\infty\mathbf{A}_y) \quad (31)$$

Where,

$$\mathbf{A}_y=\mathbf{A}^t,\ \mathbf{B}_y=[\mathbf{C}_1^t \quad \mathbf{C}_2^t],\ \mathbf{C}_y=\begin{bmatrix}\mathbf{B}_1^t\\ \mathbf{0}\end{bmatrix},\ \mathbf{D}_y=\begin{bmatrix}\mathbf{D}_{11}^t & \mathbf{D}_{12}^t\\ \mathbf{I} & \mathbf{0}\end{bmatrix},\ \widehat{\mathbf{D}}_y=\mathbf{D}_y^t\mathbf{J}\mathbf{D}_y \quad (32)$$

Using the solutions of the Riccatis above, we can invoke a result that can be derived from a modification to theorem 3 in [16] which states that

$$\mathbf{Z}_\infty = \mathbf{Y}_\infty(I - \gamma^{-2}\mathbf{X}_\infty\mathbf{Y}_\infty)^{-1} \tag{33}$$

Now, substitution of the realization of $\mathbf{G}(z)$ into the Riccati equation in (29) provides us the following Riccati equation

$$\begin{aligned}\mathbf{0} = &(\mathbf{C}'\mathbf{D}+\mathbf{A}'\mathbf{Q}\mathbf{B})(\mathbf{R}_1+\mathbf{B}'\mathbf{Q}\mathbf{B})^{-1}(\mathbf{D}'\mathbf{C}+\mathbf{B}'\mathbf{Q}\mathbf{A})+\mathbf{A}'\mathbf{X}_\infty\mathbf{A}-\mathbf{X}_\infty \\ &-(\mathbf{C}'\mathbf{D}+\mathbf{A}'\mathbf{Q}\mathbf{B}+\mathbf{A}'\mathbf{X}_\infty\mathbf{B})(\mathbf{R}_1+\mathbf{B}'\mathbf{Q}\mathbf{B}+\mathbf{B}'\mathbf{X}_\infty\mathbf{B})^{-1}(\mathbf{D}'\mathbf{C}+\mathbf{B}'\mathbf{Q}\mathbf{A}+\mathbf{B}'\mathbf{X}_\infty\mathbf{A})\end{aligned} \tag{34}$$

Upon inspection, it immediately becomes clear the the solution to this Riccati equation is $\mathbf{X}_\infty=0$. Putting this result into equation (33), it follows that

$$\mathbf{Z}_\infty = \mathbf{Y}_\infty \tag{35}$$

Thus, it follows that all the suboptimal controllers are paramterized via a single non-trivial, indefinite, Riccati equation satisfying (31). In the event that the superplant **G** is strictly proper the controller state-space has a realization as shown below

$$\mathbf{K}=\left[\begin{array}{c|c} (\mathbf{I}+\mathbf{B}\mathbf{B}'\mathbf{Q})^{-1}\mathbf{A}(\mathbf{I}+\mathbf{Y}_\infty\mathbf{C}'\mathbf{C})^{-1} & (\mathbf{I}+\mathbf{B}\mathbf{B}'\mathbf{Q})^{-1}\mathbf{A}(\mathbf{I}+\mathbf{Y}_\infty\mathbf{C}'\mathbf{C})^{-1}\mathbf{Y}_\infty\mathbf{C}' \\ \hline \mathbf{B}'\mathbf{Q}(\mathbf{I}+\mathbf{B}\mathbf{B}'\mathbf{Q})^{-1}\mathbf{A}(\mathbf{I}+\mathbf{Y}_\infty\mathbf{C}'\mathbf{C})^{-1} & \mathbf{B}'\mathbf{Q}(\mathbf{I}+\mathbf{B}\mathbf{B}'\mathbf{Q})^{-1}\mathbf{A}(\mathbf{I}+\mathbf{Y}_\infty\mathbf{C}'\mathbf{C})^{-1}\mathbf{Y}_\infty\mathbf{C}' \end{array}\right] \tag{36}$$

3.3 The Optimization Method

Now, that we have a form for computing our controller we can move on to how the loop-shaping for the PID will be done. The open-loop transfer function for the nominal plant, **P**, found using the controller, **K**, we just defined in the previous section, is **L** in equation (16). Then, the tuning of the PID controller is performed by using a convex optimization, viz. the ellipsoid algorithm mentioned, specifically the deep-cut ellipsoid method, in [37] where the optimization problem restated is

$$\min_{K\epsilon\mathsf{C}} \left\| WK_{K_1,K_2,K_3} - Z \right\|_{\infty}^{2} \quad (37)$$

Where, Z is the complementary sensitivity transfer function; K is a vector of the PID parameters over which the optimization is being performed and W is the necessary transfer function to make the product of WK similar to the complementary sensitivity transfer function. C, is the set of convex constraints for K.

The deep-cut ellipsoid algorithm in [37] is stated below

a) Initialize K(this is a vector) and A(an ellipsoid that contains feasible minimizers; if there are any). Compute the frequency responses of W and Z. The range of frequency for the frequency responses in this optimization is being chosen as two orders below and above the required bandwidth. This way, the choice of the frequency vector becomes independent of the problem.

b) Check if K satisfies the constraints C, mentioned below

$K_1 > 0$, $K_2 > 0$, $K_3 > 0$; and

$K_1 + K_2 + K_3 > 0$, $K_1 - K_2 + K_3 > 0$, $K_3 < K_1$.

If the constraints are not met, then use the active constraint sub-gradient iteration method in [17].

c) If the constraints are satisfied by K, then compute the frequency at which the objective $\left| WK_{K_1,K_2,K_3} - Z \right|^2$ attains its maximum, say ω_*. Then, we will use $h = 2Re\{WK_{K_1,K_2,K_3} - Z\}(e^{j\omega T}) \times W(e^{j\omega T})$ as a subgradient in the objective iteration. Also, we will try to see if a deep-cut may be performed to

further reduce the size of the ellipsoid in the same step. This will save a couple of iterations.

d) Repeat steps b and c until the objective function is below a threshold.

At this point, it will be relevant to mention that the performance of the ellipsoid algorithm is generally found to be adequate when the search set initialization was done with a radius of $1e^3$. But, this may deteriorate as the size of the search radius increases. However, greatly increased search radiuses may be required if the gains of the tuned PID are large. To keep the optimization quick and save programming overhead, we simply verify that the optimizing K is inside the initial search set. If not, the initialization is changed and the optimization repeated.

3.4 Conclusion

In this chapter, we have seen how the $\mathcal{H}_\infty$ controller was implemented. Also, we have taken a look at how the deep-cut ellipsoid algorithm was incorporated into our work for the convex optimization of PID parameters to obtain the loop-shape. In the next chapter, we will observe some results obtained using example plants and some comparison of the discrete-time implementation to some of the continuous-time methods.

CHAPTER 4

RESULTS

4.1 Introduction

In this chapter we will look at some of the simulation results from the $\mathcal{H}_\infty$ solver and the PID tuner programs that we created. Although the $\mathcal{H}_\infty$ solver's results are intermediary to the final output, we believe it is essential to look at them, since the PID tuner will attempt to minimize the difference between the PID controller and the $\mathcal{H}_\infty$ controller. A few specific types of SISO plants have been chosen for the purpose of analyzing simulation results; most of them are used as test-cases in various literatures. Their structures (transfer function) are shown followed by their $\mathcal{H}_\infty$ controller and PID tuned controller performances.

4.2 Results from the $\mathcal{H}_\infty$ solver

Before starting to present results a few more comments must be added about the $\mathcal{H}_\infty$ controller design algorithm that we are using. The entire program has been written in MATLAB. The input to the program is the plant, the sampling time, the required bandwidth, a pole and a zero location. Since, Glover-McFarlane method requires that a pre-shaped loop for the plant is entered into the solver, we augment the nominal plant, P_o, with a filter that has an integrator, a zero and a roll-off pole. The pole and zero location are entered as they would lie in the s-plane (continuous time), and then converted to a discrete-time filter via a Tustin transformation. The default value for the zero is half of crossover frequency and the default for the roll-off pole is the geometric mean of the crossover and Nyquist frequency. The

augmented plant, P, is then sent into the $\mathcal{H}_\infty$ controller solver as described in Section 4.2. Then the discrete filter is taken out of P and augmented into the controller to achieve the final controller structure. Having said this, we will now look into the simulation results:

a) Continuous time Plant 1:

$$P(s)=\frac{1}{s}$$

Discretized using ZOH

$$P(z)=\frac{0.01}{(z-1)}$$

The open loop plant bandwidth is 1.4137 rad/s. The results below are what we obtained for a bandwidth 0.5 times that value.

Roll-off pole at -12.1673 and Integrator zero at -0.23562 (for continuous case)

Controller Design Results	
Closed loop Bandwidth requested	0.70685 rad/s
Closed Loop Bandwidth achieved	0.70934 rad/s
Gamma_opt	1.7029
Gamma	1.7199
Step Response Characteristics	
Rise Time	2.54 Sec
Settling Time	19.85 Sec
Overshoot	14.1195 %
Undershoot	0 %
Peak	1.1412
Time to Peak	7.13 Sec

Table 1: Step response characteristics of Plant 1 for $\mathcal{H}_\infty$ controller.

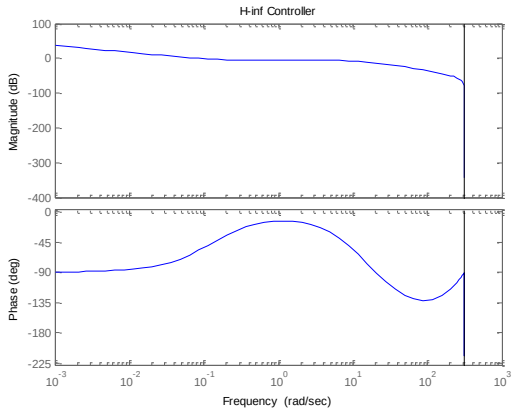

Fig 6: Bode plot of $\mathcal{H}_\infty$ controller for test plant 1.

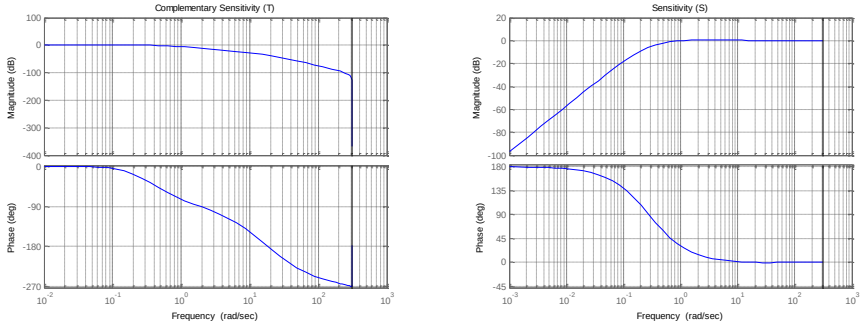

Fig 7: Complementary sensitivity and sensitivity plot of $\mathcal{H}_\infty$ controller for test plant 1.

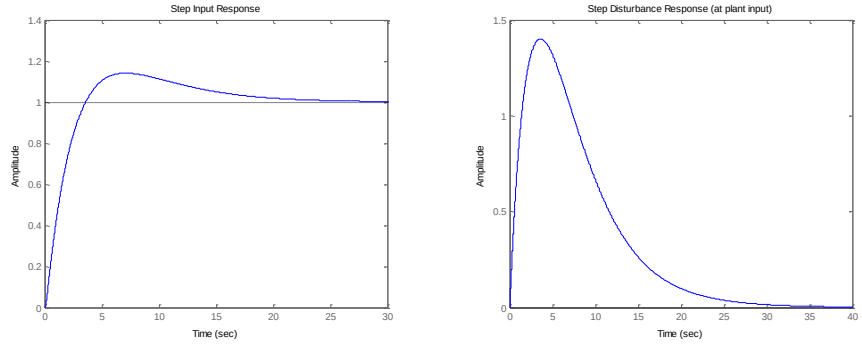

Fig 8: Step input response and step disturbance response of $\mathcal{H}_\infty$ controller for test plant 1.

Now, let us look at the results for tuner by using a closed-loop bandwidth of 5 times the plant bandwidth.

Roll-off pole at -38.4763 and Integrator zero at -2.3562 (for continuous case)

Controller Design Results	
Closed loop Bandwidth requested	7.0685 rad/s
Closed Loop Bandwidth achieved	7.3078 rad/s
Gamma_opt	1.8045
Gamma	1.8225
Step Response Characteristics	
Rise Time	0.25 Sec
Settling Time	2.01 Sec
Overshoot	15.7045 %
Undershoot	0 %
Peak	1.157
Time to Peak	0.7 Sec

Table 2: Optimization results and step response characteristics of Plant 1 for faster PID controller.

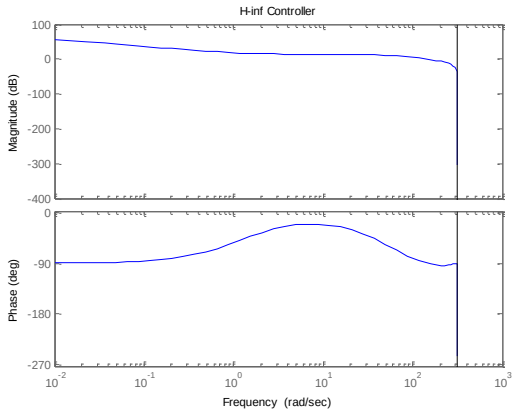

Fig 9: Bode plot of faster $\mathcal{H}_\infty$ controller for test plant 1.

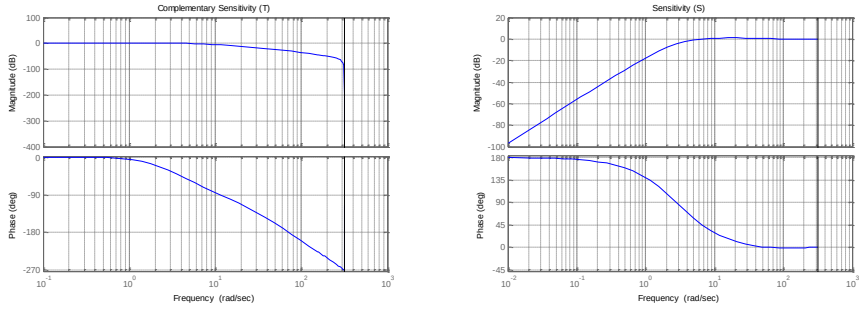

Fig 10: Complementary sensitivity and sensitivity plot of faster $\mathcal{H}_\infty$ controller for test plant 1.

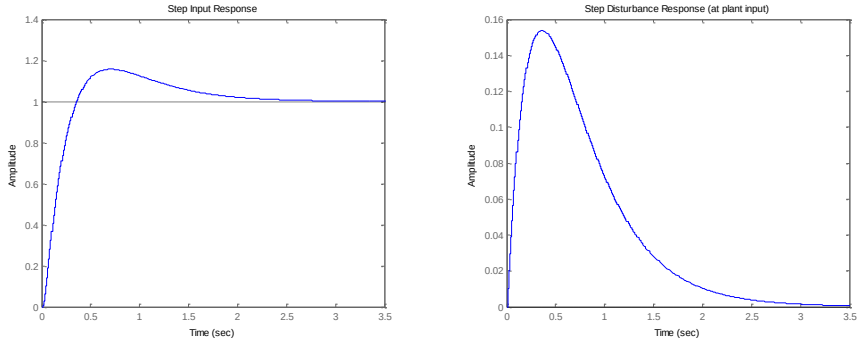

Fig 11: Step input response and step disturbance response of faster $\mathcal{H}_\infty$ controller for test plant 1.

b) Continuous time Plant 2:

$$P(s)=\frac{1}{s+1}$$

Discretized using ZOH

$$P(z)=\frac{0.00995}{(z-0.99)}$$

The open loop plant bandwidth is 0.9998 rad/s. The results below are what we obtained for a bandwidth 0.5 times that value.

Roll-off pole at -10.2322 and Integrator zero at -0.16663 (for continuous case)

Controller Design Results	
Closed loop Bandwidth requested	0.4999 rad/s
Closed Loop Bandwidth achieved	0.20613 rad/s
Gamma_opt	1.2239
Gamma	1.2362
Step Response Characteristics	
Rise Time	12.11 Sec
Settling Time	26.22 Sec
Overshoot	0 %
Undershoot	0 %
Peak	0.99734
Time to Peak	43.79 Sec

Table 3: Step response characteristics of Plant 2 for $\mathcal{H}_\infty$ controller.

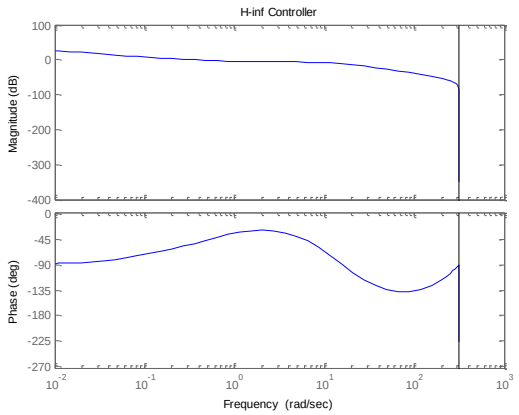

Fig 12: Bode plot of $\mathcal{H}_\infty$ controller for test plant 2.

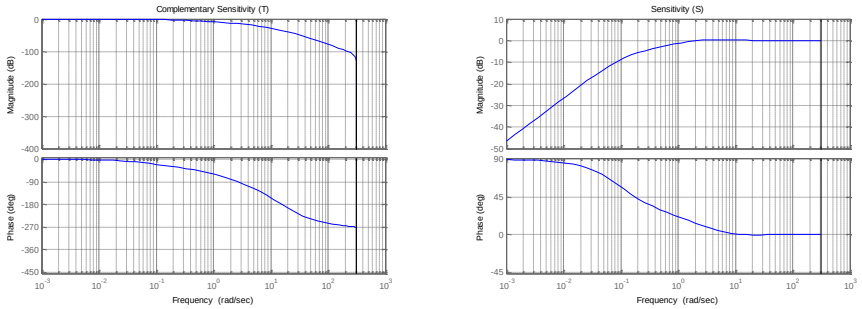

Fig 13: Complementary sensitivity and sensitivity plot of $\mathcal{H}_\infty$ controller for test plant 2.

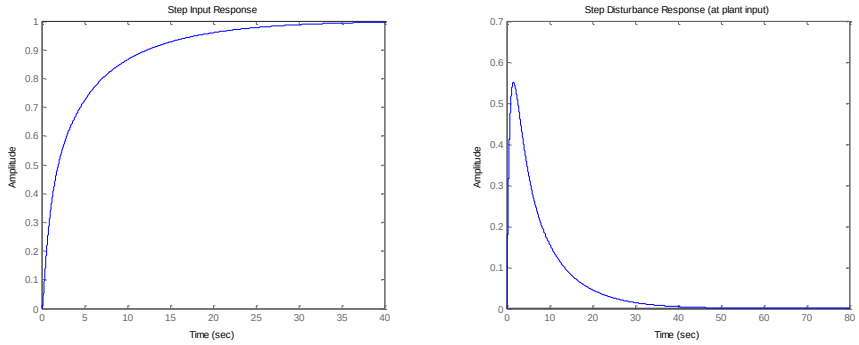

Fig 14: Step input response and step disturbance response of $\mathcal{H}_\infty$ controller for test plant 2.

Now, let us look at the results for tuner by using a closed-loop bandwidth of 5 times the plant bandwidth.

Roll-off pole at -32.3572 and Integrator zero at -1.6663 (for continuous case)

Controller Design Results	
Closed loop Bandwidth requested	4.999 rad/s
Closed Loop Bandwidth achieved	3.9469 rad/s
Gamma_opt	1.5777
Gamma	1.5935
Step Response Characteristics	
Rise Time	0.52 Sec
Settling Time	1.89 Sec
Overshoot	3.1687 %
Undershoot	0 %
Peak	1.0317
Time to Peak	1.3 Sec

Table 4: Step response characteristics of Plant 2 for faster $\mathcal{H}_\infty$ controller.

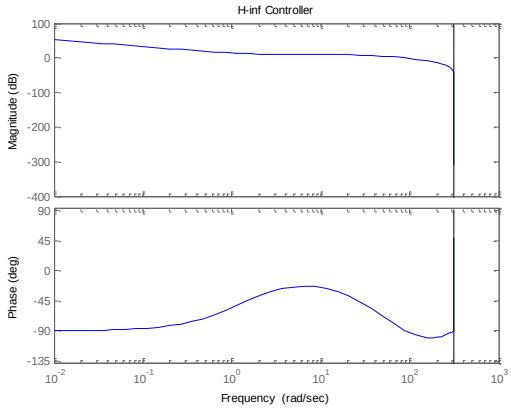

Fig 15: Bode plot of faster $\mathcal{H}_\infty$ controller for test plant 2.

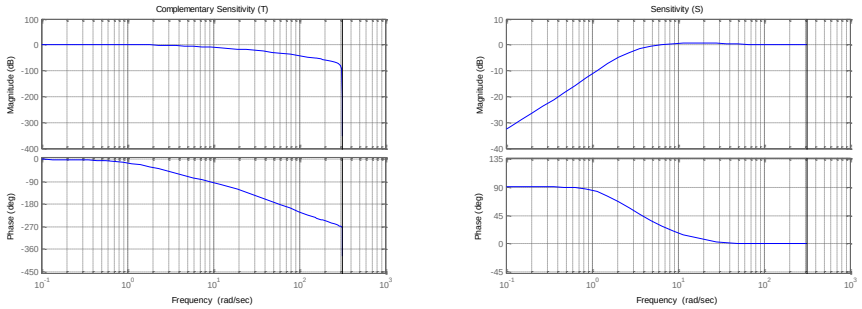

Fig 16: Complementary sensitivity and sensitivity plot of faster $\mathcal{H}_\infty$ controller for test plant 2.

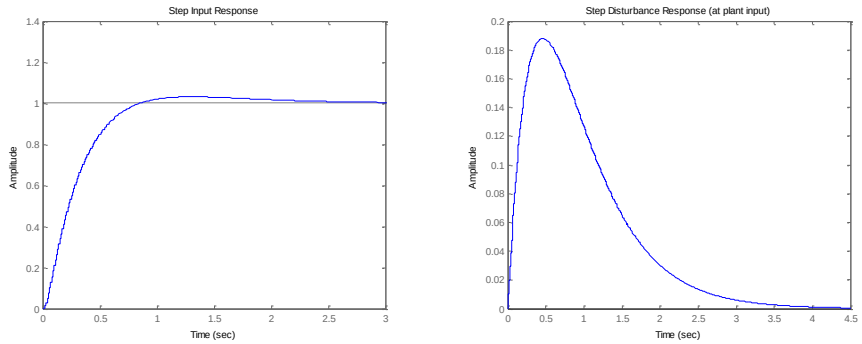

Fig 17: Step input response and step disturbance response of faster $\mathcal{H}_\infty$ controller for test plant 2.

c) Continuous time Plant 3:

In the next two test cases we will investigate the $\mathcal{H}_\infty$ controller performance for plants of type

$P(s)=\frac{-s+a}{(s+1)^2}$, where a = 0.5 in this case and a closed loop bandwidth of 1 rad/s is being used.

Putting a =0.5 and discretizing using ZOH

$$P(z)=\frac{-0.009876\,z\ +\ 0.009925}{z\text{^}2\ -\ 1.98\,z\ +\ 0.9802}$$

The results below are what we obtained for a bandwidth of 1 rad/s

Roll-off pole at -14.472 and Integrator zero at -0.33333 (for continuous case)

Controller Design Results	
Closed loop Bandwidth requested	1 rad/s
Closed Loop Bandwidth achieved	0.25315 rad/s
Gamma_opt	2.054
Gamma	2.0745
Step Response Characteristics	
Rise Time	10.81 Sec
Settling Time	19.43 Sec
Overshoot	0 %
Undershoot	38.372 %
Peak	0.99782
Time to Peak	32.19 Sec

Table 5: Step response characteristics of Plant 3a for $\mathcal{H}_\infty$ controller.

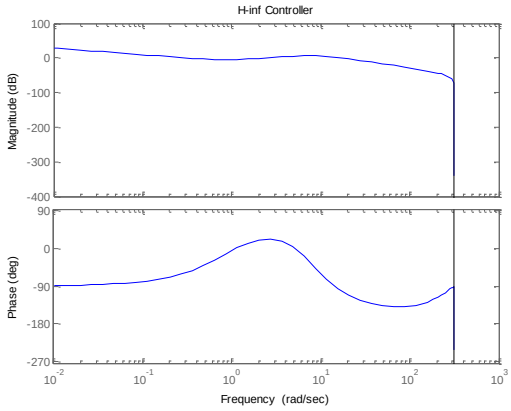

Fig 18: Bode plot of $\mathcal{H}_\infty$ controller for test plant 3a.

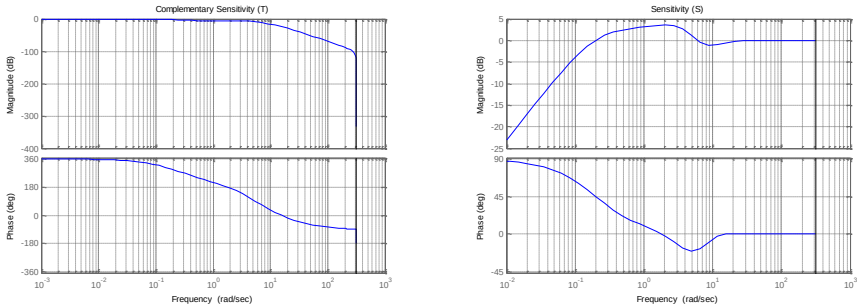

Fig 19: Complementary sensitivity and sensitivity plot of $\mathcal{H}_\infty$ controller for test plant 3a.

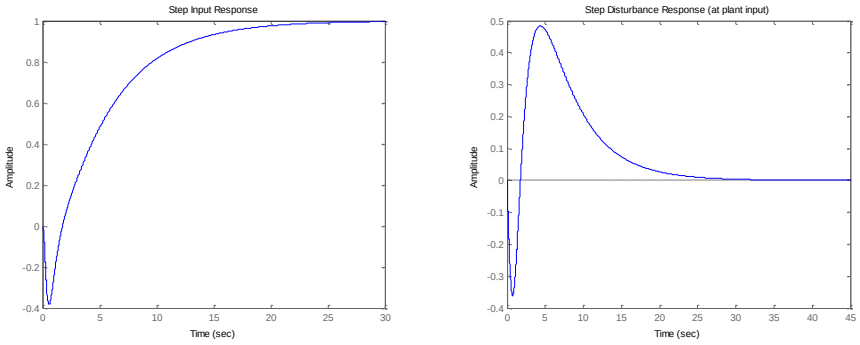

Fig 20: Step input response and step disturbance response of $\mathcal{H}_\infty$ controller for test plant 3a.

Now, let us look at the results for a= 5.

P(s)=$\frac{-s+a}{(s+1)^2}$, where a = 5 in this case and a closed loop bandwidth of 1 rad/s is being used.

Putting a =5 and discretizing using ZOH

$$P(z)=\frac{-0.009652\,z\ +\ 0.01015}{z\text{^}2\ -\ 1.98\,z\ +\ 0.9802}$$

Roll-off pole at -14.472 and Integrator zero at -0.33333 (for continuous case)

Controller Design Results	
Closed loop Bandwidth requested	1 rad/s
Closed Loop Bandwidth achieved	0.42776 rad/s
Gamma_opt	1.5798
Gamma	1.5956
Step Response Characteristics	
Rise Time	5.65 Sec
Settling Time	11.95 Sec
Overshoot	0 %
Undershoot	1.8301 %
Peak	0.99816
Time to Peak	20.59 Sec

Table 6: Step response characteristics of Plant 3b for $\mathcal{H}_\infty$ controller.

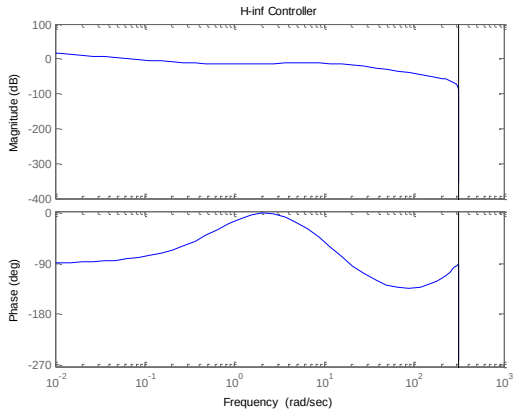

Fig 21: Bode plot of $\mathcal{H}_\infty$ controller for test plant 3b.

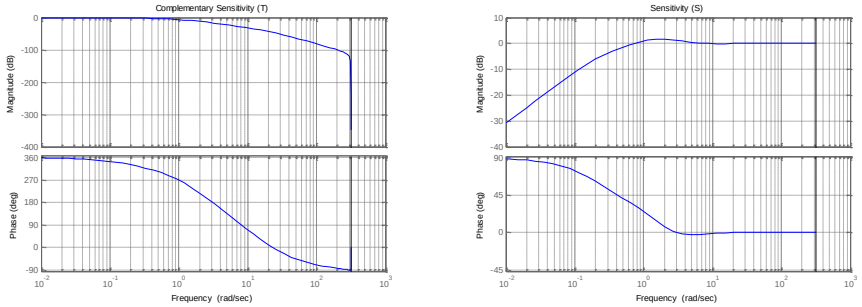

Fig 22: Complementary sensitivity and sensitivity plot of $\mathcal{H}_\infty$ controller for test plant 3b.

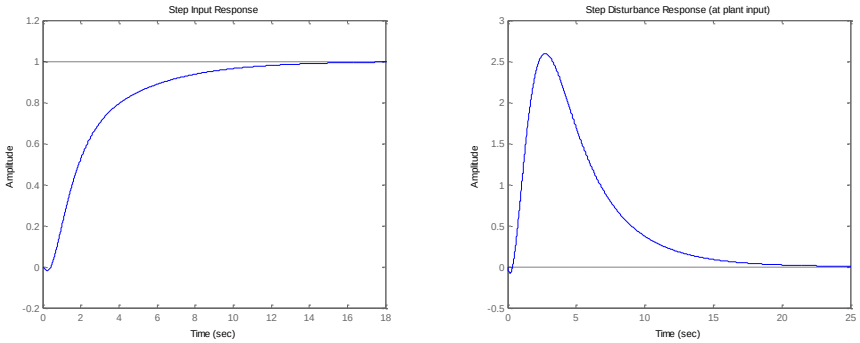

Fig 23: Step input response and step disturbance response of $\mathcal{H}_\infty$ controller for test plant 3b.

d) Continuous time Plant 4:

$P(s)=\frac{s+\varepsilon}{(s+1)^2}$, where ε is a small number, in this case ε = 0.5(1/2 bandwidth)

Discretized using ZOH

$$P(z)=\frac{0.0099253\ (z-0.995)}{(z-0.99)^2}$$

The open loop plant bandwidth is 1 rad/s. The results below are what we obtained.

Roll-off pole at -14.472 and Integrator zero at -0.33333 (for continuous case)

Controller Design Results	
Closed loop Bandwidth requested	1 rad/s
Closed Loop Bandwidth achieved	0.3327 rad/s
Gamma_opt	1.2277
Gamma	1.24
Step Response Characteristics	
Rise Time	7.38 Sec
Settling Time	15.83 Sec
Overshoot	0 %
Undershoot	0 %
Peak	0.99821
Time to Peak	27.99 Sec

Table 7: Step response characteristics of Plant 4 for $\mathcal{H}_\infty$ controller.

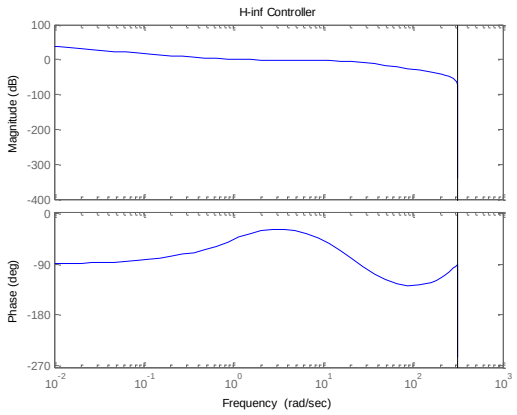

Fig 24: Bode plot of $\mathcal{H}_\infty$ controller for test plant 4.

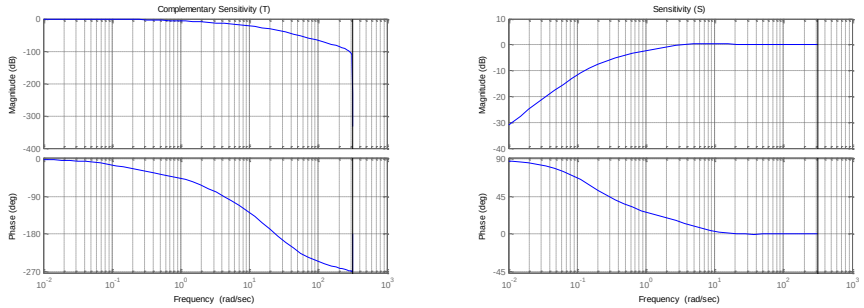

Fig 25: Complementary sensitivity and sensitivity plot of $\mathcal{H}_\infty$ controller for test plant 4.

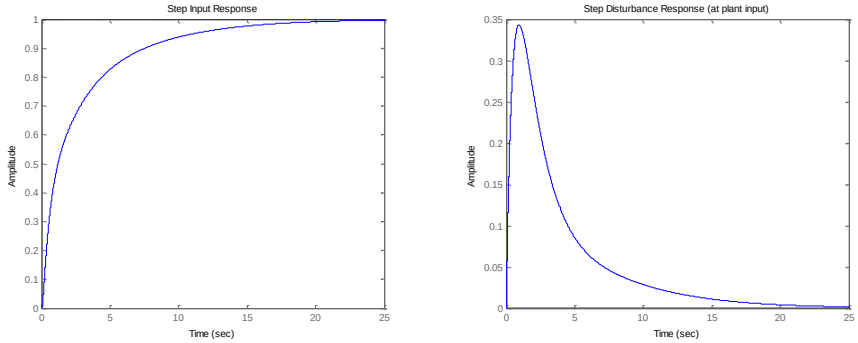

Fig 26: Step input response and step disturbance response of $\mathcal{H}_\infty$ controller for test plant 4.

4.3 Results from PID tuner

The input to the PID tuner code is the plant, its sampling rate and the requested closed-loop bandwidth. If the plant is not discretized beforehand, the code can discretize it if a suitable discretization method is specified. The tuner then performs the optimization to spur out the PID controller transfer function and the values of the parameters K_P, K_I and K_D. The same plants used in section 4.2 are tested here.

a) Continuous time Plant 1:

$$P(s)=\frac{1}{s}$$

Discretized using ZOH

$$P(z)=\frac{0.01}{(z-1)}$$

The open loop plant bandwidth is 1.4137 rad/s. The results below are what we obtained for a bandwidth of 5 times that value.

PID zeros at:

-0.999989349179974

0.984578057212416

PID Parameters : Kp = 4.6125, Ki = 7.167, Kd = -0.022878

Optimization Results	
Closed loop Bandwidth requested	7.3078 rad/s
Closed Loop Bandwidth achieved	6.5828 rad/s
Optimization took	285 iterations.
Approximation error	0.090869
Step Response Characteristics	
Rise Time	0.28 Sec
Settling Time	1.9 Sec
Overshoot	17.0546%
Undershoot	0 %
Peak	1.1705
Time to Peak	0.76 Sec

Table 8: Optimization results and step response characteristics of Plant 1 for PID controller.

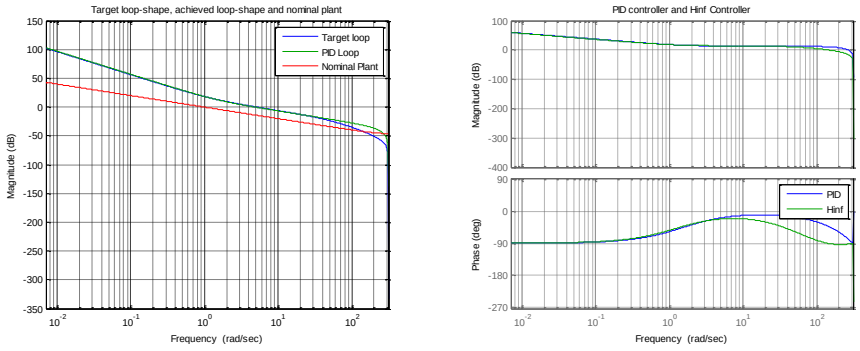

Fig 27: Magnitude plot for $\mathcal{H}_\infty$ and PID loop-shape and Bode plot of $\mathcal{H}_\infty$ and PID controller for test plant 1.

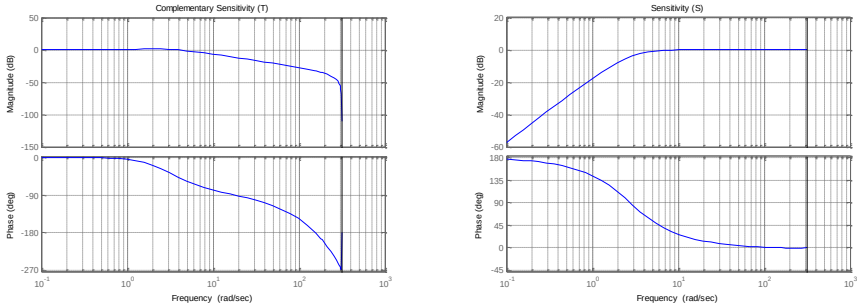

Fig 28: Complementary sensitivity and sensitivity plot of PID controller for test plant 1.

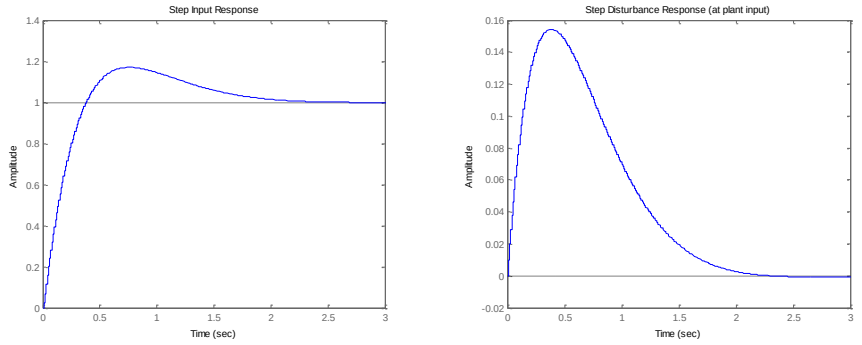

Fig 29: Step input response and step disturbance response of PID controller for test plant 1.

b) Continuous time Plant 2:

$$P(s)=\frac{1}{s+1}$$

Discretized using ZOH

$$P(z)=\frac{0.00995}{(z-0.99)}$$

The open loop plant bandwidth is 0.9998 rad/s. The results below are what we obtained for a bandwidth 5 times that value.

PID zeros at:

-0.999166221726185

0.983993523940431

PID Parameters : Kp = 2.7967, Ki = 4.5127, Kd = -0.013865

Optimization Results	
Closed loop Bandwidth requested	4.999 rad/s
Closed Loop Bandwidth achieved	3.5578 rad/s
Optimization took	282 iterations.
Approximation error	0.073511
Step Response Characteristics	
Rise Time	0.55 Sec
Settling Time	2.2 Sec
Overshoot	5.0315%
Undershoot	0 %
Peak	1.0503
Time to Peak	1.31 Sec

Table 9: Optimization results and step response characteristics of Plant 2 for PID controller.

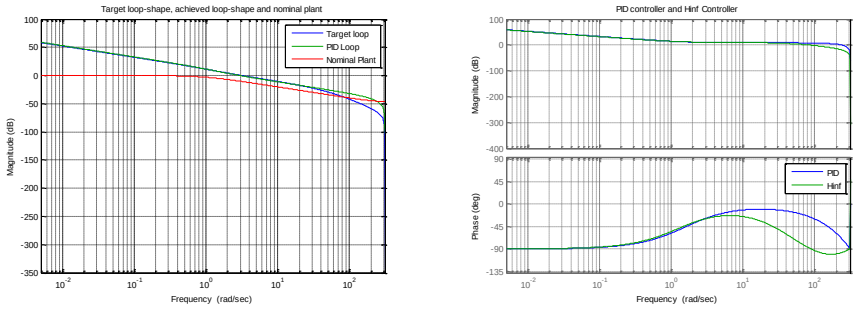

Fig 30: Magnitude plot for $\mathcal{H}_\infty$ and PID loop-shape and Bode plot of $\mathcal{H}_\infty$ and PID controller for test plant 2.

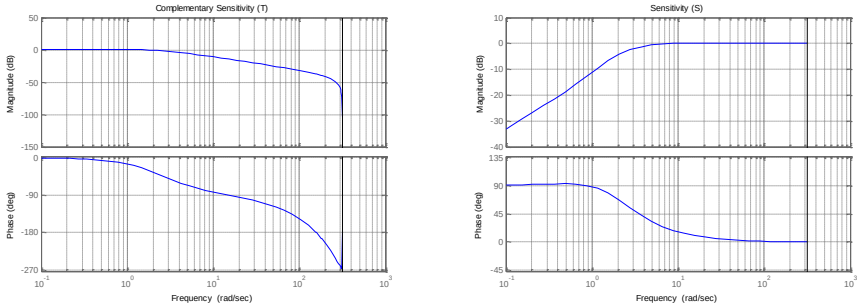

Fig 31: Complementary sensitivity and sensitivity plot for PID controller for test plant 2.

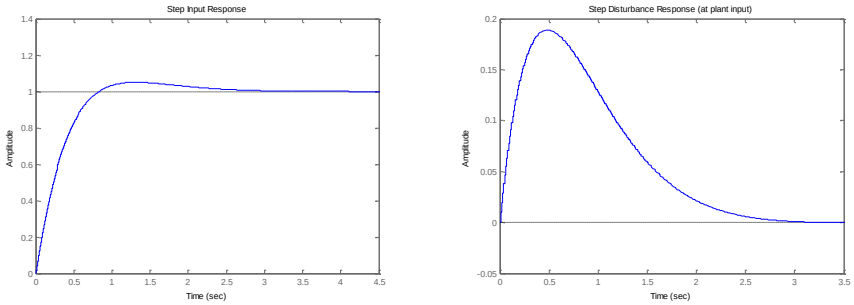

Fig 32: Step input response and step disturbance response of PID controller for test plant 2.

c) Continuous time Plant 3, taken from [5]:

In the next two test cases we will investigate the PID controller performance for plants of type

$P(s)=\frac{-s+a}{(s+1)^2}$, where a = 0.5 in this case and a closed loop bandwidth of 1 rad/s is being used.

Putting a =0.5 and discretizing using ZOH

$$P(z)=\frac{-0.0098757\ (z-1.005)}{(z-0.99)^2}$$

The results below are what we obtained for a bandwidth of 1 rad/s.

PID zeros at:

0.997305061617537

0.823274700016398

PID Parameters : Kp = 0.83727, Ki = 0.22284, Kd = 0.038417

Optimization Results	
Closed loop Bandwidth requested	1 rad/s
Closed Loop Bandwidth achieved	1.4423 rad/s
Optimization took	361 iterations.
Approximation error	0.21843
Step Response Characteristics	
Rise Time	18.17 Sec
Settling Time	33.32 Sec
Overshoot	0 %
Undershoot	33.1193%
Peak	0.99954
Time to Peak	72.79 Sec

Table 10: Optimization results and step response characteristics of Plant 3a for PID controller.

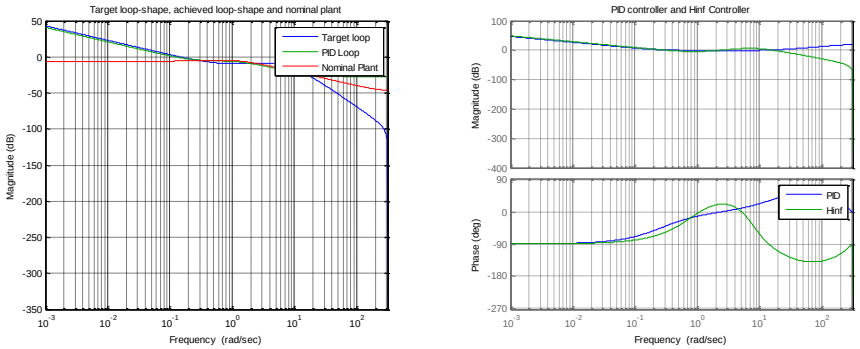

Fig 33: Magnitude plot for $\mathcal{H}_\infty$ and PID loop-shape and Bode plot of $\mathcal{H}_\infty$ and PID controller for test plant 3a.

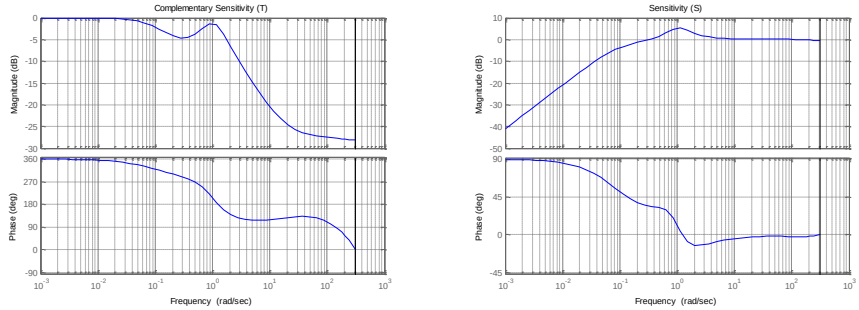

Fig 34: Complementary sensitivity and sensitivity plot for PID controller for test plant 3a.

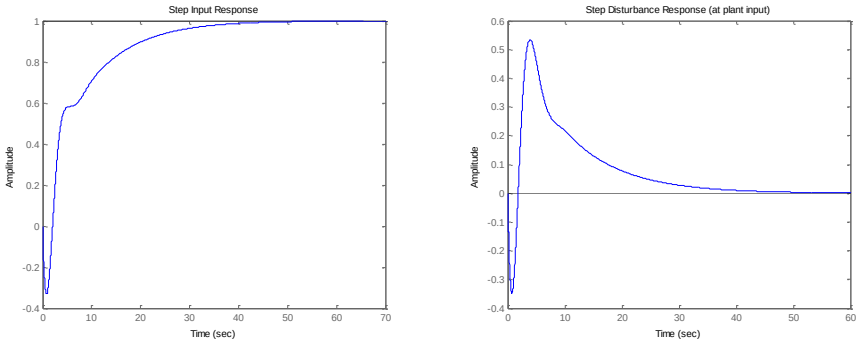

Fig 35: Step input response and step disturbance response of PID controller for test plant 3a.

Now, let us look at the results for a= 5.

P(s)=$\frac{-s+a}{(s+1)^2}$, where a = 5 in this case and a closed loop bandwidth of 1 rad/s is being used.

Putting a =0.5 and discretizing using ZOH

$$P(z)=\frac{-0.009652\,z\ +\ 0.01015}{z\text{^}2\ -\ 1.98\,z\ +\ 0.9802}$$

The results below are what we obtained for a bandwidth of 1 rad/s.

PID zeros at:

0.995948365374723

0.840016852744668

PID Parameters : Kp = 0.16555, Ki = 0.065679, Kd = 0.008477

Optimization Results	
Closed loop Bandwidth requested	1 rad/s
Closed Loop Bandwidth achieved	0.42776 rad/s
Optimization took	361 iterations.
Approximation error	0.046608
Step Response Characteristics	
Rise Time	6.45 Sec
Settling Time	13.8 Sec
Overshoot	0 %
Undershoot	1.4997%
Peak	0.99968
Time to Peak	30.99 Sec

Table 11: Optimization results and step response characteristics of Plant 3 for PID controller.

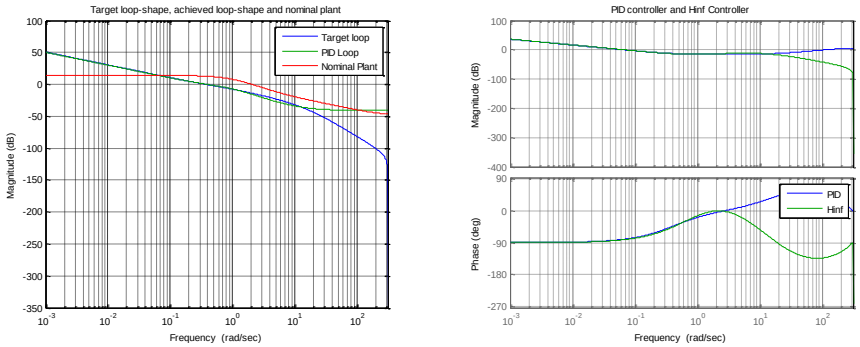

Fig 36: Magnitude plot for $\mathcal{H}_\infty$ and PID loop-shape and Bode plot of $\mathcal{H}_\infty$ and PID controller for test plant 3b.

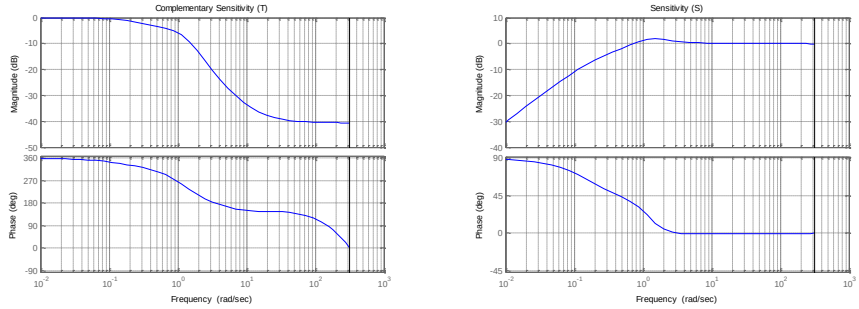

Fig 37: Complementary sensitivity and sensitivity plot for PID controller for test plant 3b.

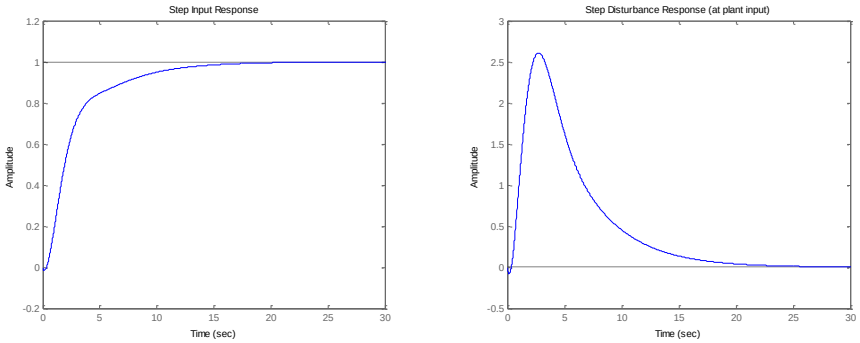

Fig 38: Step input response and step disturbance response of PID controller for test plant 3b.

d) Continuous time Plant 4:

$P(s)=\frac{s+\varepsilon}{(s+1)^2}$, where ε is a small number, in this case ε = 0.5(1/2 bandwidth)

Discretized using ZOH

$$P(z)=\frac{0.0099253\ (z-0.995)}{(z-0.99)^2}$$

The results below are what we obtained for a bandwidth of 1 rad/s

PID zeros at:

-0.999711095777509

0.989587138716866

PID Parameters : Kp = 0.71025, Ki = 0.74345, Kd = -0.0035322

Optimization Results	
Closed loop Bandwidth requested	1 rad/s
Closed Loop Bandwidth achieved	0.3327 rad/s
Optimization took	345 iterations.
Approximation error	0.053938
Step Response Characteristics	
Rise Time	7 Sec
Settling Time	13.57 Sec
Overshoot	0 %
Undershoot	0 %
Peak	0.99997
Time to Peak	39.79 Sec

Table 12: Optimization results and step response characteristics of Plant 4 for PID controller.

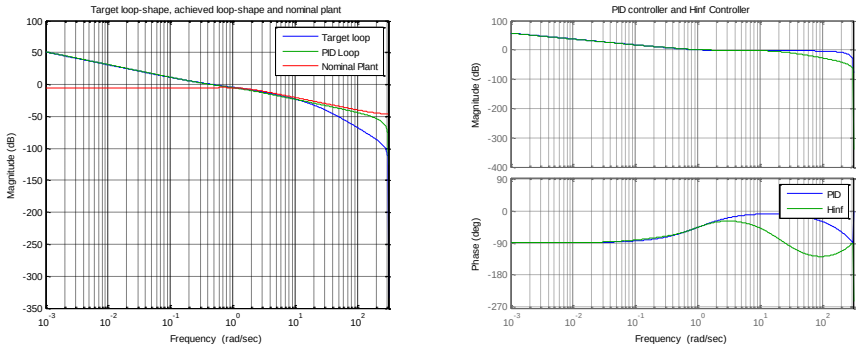

Fig 39: Magnitude plot for $\mathcal{H}_\infty$ and PID loop-shape and Bode plot of $\mathcal{H}_\infty$ and PID controller for test plant 4.

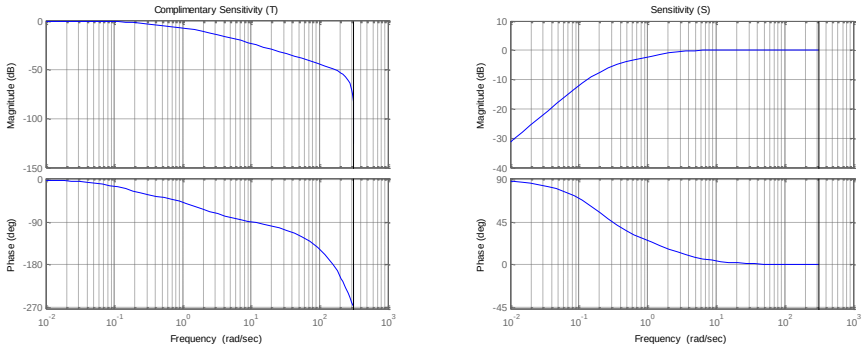

Fig 40: Complementary sensitivity and sensitivity plot of PID controller for test plant 4.

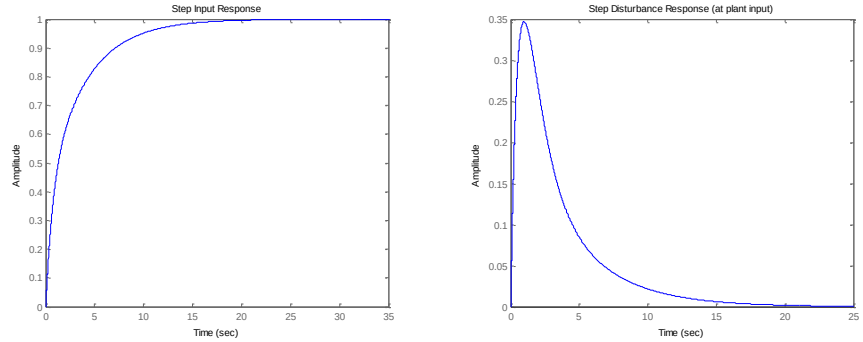

Fig 41: Step input response and step disturbance response of PID controller for test plant 4.

4.4 Comparison among tuning methods

In this section we will see how our design fares with the continuous time PID tuner described in [4]. We will also compare our results with two other methods. One is where the tuning is done against an $\mathcal{H}_\infty$ loop-shape entirely designed in continuous time; the other is where a delay equal to half the sampling time is added to the plant(using a second-order Padé) then the continuous PID tuned from the $\mathcal{H}_\infty$ loop-shape is discretized. So, we have two continuous and two discrete-time systems being compared all together.

The test plant chosen for this is

$$P(s)=\frac{1}{(s+1)^3}$$

Discretized by a sampling rate of 0.01 S/sec to

$$P(z)=\frac{1.6542e-007\ (z+3.704)\ (z+0.2659)}{(z-0.99)^3}$$

The table below summarizes the behavior of the four loops. It must be noted, that our key goal was to make sure that closed-loop bandwidth for all tuners were comparably close.

Property	Discrete $\mathcal{H}_\infty$	Discretized $\mathcal{H}_\infty$	Cont. $\mathcal{H}_\infty$	Cont. LQR.
RiseTime	0.98	0.98	0.99	0.96
SettlingTime	5.58	5.50	5.51	6.14
Overshoot	24.38	23.08	22.94	10.55
Undershoot	0.00	0.00	0.00	0.00
Peak	1.24	1.23	1.23	1.11
PeakTime	2.27	2.26	2.30	2.18
BW achieved	2.0714	2.0813	2.0763	2.2135
Approx. error	0.17213	0.13382	0.13394	0.26368

Table 13: Step response characteristics of test plant using the four tuning methods.

Another comparison we are interested in looking at is the effect of slower sampling rates on the two discrete PID tuners. Our design, and the method of converting the continuous PID tuned from an $\mathcal{H}_\infty$ loop-shape into discrete-time. It is our inference that high sampling rates will not adversely affect our design as it will the other method.

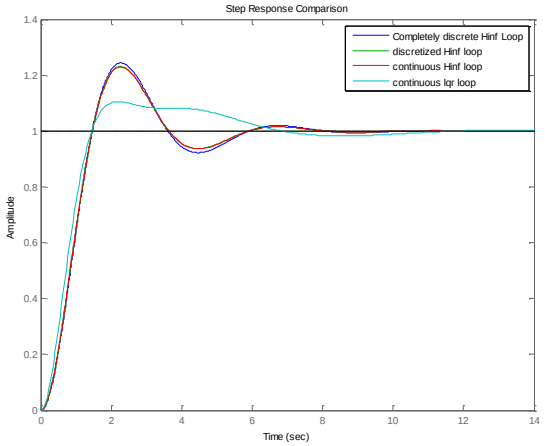

Fig 42: Step input response using four methods of tuning.

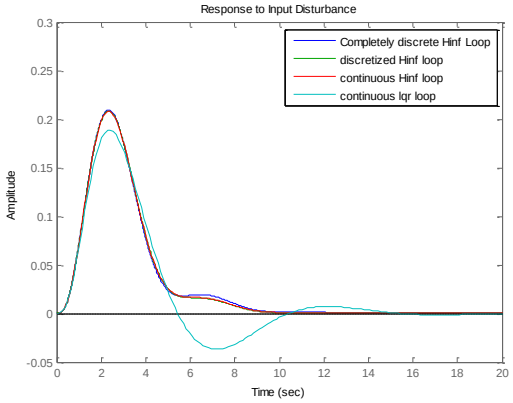

Fig 43: Step disturbance response using four methods of tuning.

CHAPTER 5

CONLUSIONS AND FUTURE WORK

5.1 Concluding remarks

In this work we have developed a method for PID controller tuning in discrete-time. An $\mathcal{H}_\infty$ controller is first designed for a given plant then the open-loop loop-shape is optimized against a PID controller structure via an LMI minimization algorithm to obtain the PID parameters.

Comparing results from the $\mathcal{H}_\infty$ solver and the PID tuner we see that the PID can work well within suitable ranges. The low approximation errors and step response plots tell us how well the PIDs perform. In some cases we have seen that the optimization may fail for certain specified closed-loop bandwidths. It is important to point out at this point that all the results shown in chapter 4 are based on default values for the pole and zero location of the filter. The user can choose to modify those locations if the default value does not work for the required bandwidth. This should, in most circumstances, produce acceptable results. If the plant is too complicated, then a PID controller may not be a suitable form of controller for the plant after-all that is where PID limitations show up due to their simple structure.

We have also shown that our method works comparably well in comparison to continuous $\mathcal{H}_\infty$ tuning and a method of discretizing the tuned $\mathcal{H}_\infty$ controller. There are some improvements to be done to the work and those are discussed in the next section.

5.1 Future Work

There are many improvements that can be added to this design of ours. The first thing to do may be to incorporate an automatic mechanism in which the tuner will find out a range of suitable closed loop bandwidths from which the user can pick one; instead of the user having to guess it themselves. This will make things easier for operators in charge of machinery. Another added functionality to work on can be the ability of the tuner to choose the range of frequency to run the optimization around; this will enable the code to further minimize approximation error. Then, we may try to find a better way of choosing the pole and zero locations for the filter so that we have better loop-shapes. Ones that can be approximated by PIDs and have good rise-time, overshoot and bandwidth properties; essentially reducing the trade-offs in those three system performance parameters.

Also, some form of controller order reduction may be incorporated to the $\mathcal{H}_\infty$ solver. During our work, we have seen that there are test cases where the optimization fails for low bandwidths. This needs to be solved by delving deeper into the LMI algorithm so as to make the tuner more generic.

Another important step to be incorporated is an added layer of iteration for the $\mathcal{H}_\infty$ solver. Right now, the solver tries to approximate the desired closed-loop bandwidth based on the open-loop shape of the augmented plant; but, what we get as closed-loop bandwidth, although close, is not exactly the number we ask for. This error in turn translates to the PID tuner and the tuner adds some more error to it. So, some iteration may be added to ensure that the controller gives us exactly the closed-loop bandwidth we want. Last of all, the $\mathcal{H}_\infty$ solver used in this work was

based on a right-coprime factorization of the plant; we may try to use a left-coprime factorization since that is what yields from system identification type experiments.

REFERENCES

[1] J. B. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *ASME Transactions*, v64, pp. 759-768, 1942.

[2] G. H. Cohen and G. A. Coon. Theoretical Consideration of Related Control. *ASME Transactions* v75, pp. 827-834, 1953.

[3] K. J. Astrom and T. Hagglund. Automatic Tuning of PID Controllers. *Instrument Society of America*, 1998.

[4] E. Grassi and K. Tsakalis. PID controller tuning by frequency loop-shaping: application to diffusion furnace temperature control. *IEEE Transactions on Control Systems Technology*, vol.8, no.5, pp.842-847, 2000.

[5] A. Voda and I. Landau. A method for Auto-Calibration of PID Controllers. *Automatica, Volume 31, Issue 1*, January 1995.

[6] C. Kessler. Das symmetrische optimum. *Regelungstetechnik*, pp. 432–436, (1958).

[7] G. Nudelman and R. Kulesky. New Approach to PID-Control Optimization for Power Station Loops. *Proceedings Of European Control Conf.*, Sept. 2001.

[8] B. Kristiansson and B. Lennartson. Robust and optimal tuning of PI and PID controllers. *Control Theory and Applications, IEE Proceedings* - , vol.149, no.1, pp.17-25, Jan 2002.

[9] S. Malan, M. Milanese and M. Taragna. Robust tuning for PID controllers with multiple performance specifications. *Proceedings of the 33rd IEEE Conference on Decision and Control,* 1994.

[10] A. Tesi and A. Vicino. Design of Optimally Robust Controllers with few Degrees of Freedom. *Proc. IFAC Symp. On Design Methods of Control Sys,* 1991.

[11] E. Grassi, K. Tsakalis, S. Das, S. Gaikwad and G. Stein. Adaptive/Self-Tuning PID Control by Frequency Loop-Shaping. *Proc. Of 39th IEEE CDC,* 2000.

[12] E. Grassi and K. Tsakalis. PID Controller Tuning by Frequency Loop-Shaping. *Proc. 35th Conference on Decision and Control,* 1996.

[13] E. Grassi, K. Tsakalis, S. Dash, S. Gaikwad, W. MacArthur and G. Stein. Integrated system identification and PID controller tuning by frequency loop-shaping. *IEEE Transactions on Control Systems Technology,* vol.9, no.2, pp.285-294, 2001.

[14] D.C. McFarlane and K. Glover. Robust Controller Design Using Normalized Coprime Factor Plant Descriptions. *Lecture Notes in Control and Information Sciences*, 1989.

[15] L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.

[16] R. Callafon and P.M.J. Hof. Control relevant identification for H_∞-norm based performance specifications. *Proceedings of the 34th IEEE Conference on Decision and Control,* 1995.

[17] M. Gevers, B. Anderson and B. Dorons. Issues in modeling for control. *Proceedings of the 1998 American Control Conference,* 1998.

[18] S. Adusumilli, D. Rivera, S. Dash and K. Tsakalis. Integrated MIMO identification and robust PID controller design through loop shaping. *Proceedings of the 1998 American Control Conference,* 1998.

[19] D. Bayard, Y. Yam and E. Mettler. A criterion for joint optimization of identification and robust control. *IEEE Transactions on Automatic Control,* 1992.

[20] Z. Zhang, J. Cao and G. Lu. Robust H_∞ controller designs for linear uncertain discrete-time systems: the LMI approach. *Control, Automation, Robotics and Vision Conference,* 2004.

[21] C. Crusius and A. Trofino. Sufficient LMI conditions for output feedback control problems. *IEEE Transactions on Automatic Control,* 1999.

[22] P. Shi and S. Shue. Robust H_∞ Control for Linear Discrete-Time Systems With Norm-Bounded Nonlinear Uncertainties. *IEEE Trans, Automat.Contr.,* 1999.

[23] D. Hoover, W. Perkins and J. Medanic. A Reduced-Order Discrete-Time H-Infinity Norm Bounding Controller. *Proc. of the 34th CDC,* 1995.

[24] Ravi K., V. Aripirala and V. Syrmos. Normalized Coprime Factorizations for Sampled-Data Systems. *Int. J. Control,* 1997. Vol. 67, No.3.

[25] L. Mirkin. On Discrete-Time H_∞ problem with a strictly proper controller. *Int. J. Control,* 1997. Vol. 66, No.5.

[26] W. Sun, K. Nagpal and P. Khargonekar. H_∞ Control and Filtering for Sampled-Data Systems. *IEEE Trans. Automatic Control,* 1993. Vol. 38. No. 8

[27] T. Chen and B. Francis. H_∞ -Optimal Sampled-Data Control: Computation and Design. *Automatica,* 1996. Vol. 32 No. 2.

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

[29] E.I. Jury. *Theory and application of the z-transform method.* Huntington, NY: R.E. Krieger Publishing Co., 1973.

[30] M. Vidyasagar. *Nonlinear Systems Analysis*, Prentice-Hall, 1987.

[31] D.J. Walker. *Robust stabilizability of discrete-time systems with normalized stable factor perturbation.* Int. J. Control, vol.52, no. 2: pg 441-455, 1990.

[32] B. Francis. *A Course in $\mathcal{H}_\infty$ control theory*. Berlin: Springer-Verlag, 1987.

[33] D. Limebeer, M. Green and D. Walker, *Discrete-time $\mathcal{H}_\infty$ control.* Proc. Of IEEE CDC, 1989.

[34] M. Vidyasagar and H. Kimura. *Robust controller for uncertain linear multivariable systems.* Automatica, 1986.

[35] K. Glover. *All Optimal Hankel Norm Approximations of Linear Multivariable Systems and Their Error Bounds.* Int. J. of Control, 1984.

[36] D. Walker. *Relationship between three discrete-time H^∞ Algebraic Riccati Equation solutions.* Int. J. of Control, 1990.

[37] S. Boyd and C. Barratt. *Linear Controller Design, Limits of Performance.* Prentice Hall, 1991.

[38] S. Harris and D. Mellichamp. Controller Tuning Using Optimization to Meet Multiple Closed Loop Criteria. *AIChE J.,* 1985. Vol. 31, No. 3,

[39] G. Zames. On the input-output stability of time-varying nonlinear feedback systems. Part I: Conditions using concepts of loop gain, conicity, and positivity. *IEEE Trans. Automat. Contr.*, 1966. Vol. AC-11, pp.228 – 238

[40] A. O'Dwyer, Handbook of PI and PID Controller Tuning Rules, Imperial College Press, (2006).

[41] E. Sachs, A. Hu, and A. Ingolfsson, "Run by Run Process Control: Combining SPC and Feedback Control," IEEE Trans. Semiconduct. Manufact., V.8, No.1, pp. 26--43, Feb. 1995.

[42] S. W. Butler and J. A. Stefani, "Supervisory Run-to-Run Control of Polysilicon Gate Etch Using In-Situ Ellipsometry," IEEE Trans. Semiconduct. Manufact., V.7, No.2, pp. 193--201, May 1994.

[43] J. S. Baras and N. Patel, "A Framework for Robust Run by Run Control with Lot Delayed Measurements," IEEE Trans. Semiconduct. Manufact., V.10, No.1, pp. 75--83, Feb. 1997.

[44] D. S. Boning, W. P. Moyne, T. H. Smith, J. Moyne, R. Telfeyan, A. Hurwitz, S. Shellman, and J. Taylor, "Run by Run Control of Chemical-Mechanical Polishing," IEEE Trans. Compon. Pack. Manufact. C, V.19, No.4, pp. 307--314, Oct. 1996.