DESIGN OF P AND PI CONTROLLERS FOR HEAD POSITIONING IN HARD DISK DRIVES WITH TIME DELAY

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

H. Buket Koyuncu

June, 2015

DESIGN OF *P* AND *PI* CONTROLLERS FOR HEAD POSITIONING IN HARD DISK DRIVES WITH TIME DELAY
By H. Buket Koyuncu
June, 2015

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Hitay Özbay (Advisor)

Prof. Dr. Ömer Morgül

Prof. Dr. Mehmet Önder Efe

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

DESIGN OF P AND PI CONTROLLERS FOR HEAD POSITIONING IN HARD DISK DRIVES WITH TIME DELAY

H. Buket Koyuncu
M.S. in Electrical and Electronics Engineering
Advisor: Prof. Dr. Hitay Özbay
June, 2015

In today's high performance positioning applications, due to stringent design objectives, it is very challenging to cope with input-output time delays. In Hard Disk Drive (HDD) servo system, the information flow between process and the controller is under a time delay. Typical state-space based modern control algorithms are not applicable to such infinite dimensional plants. In this thesis several control design objectives are considered and various types of stabilizing controllers are derived for this infinite dimensional plant. The objective of this thesis is to determine alternative simple (low order) controllers to the previously designed H_∞ controller and controllers designed from Pade approximations. For this purpose, six different P and PI controllers for the unstable infinite dimensional plant are obtained. Comparisons of the controllers with each other are done and advantages of every approach are demonstrated.

Keywords: Time Delay, Hard Disk Drive, Infinite Dimensional System, P, PI Controller, Delay Margin.

ÖZET

SABİT DİSK SÜRÜCÜLERDE ZAMAN GECİKMELİ KAFA KONUMLANMASI İÇİN P VE PI KONTROLCÜ TASARIMI

H. Buket Koyuncu
Elektrik-Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Prof. Dr. Hitay Özbay
Haziran, 2015

Günümüzde yüksek performans gerektiren pozisyon kontrolü uygulamalarında, tasarım hedeflerinin sıkılığından, girdi-çıktı zaman gecikmesiyle başa çıkmak oldukça zordur. Sabit Disk Sürücüsünün servo sistemine bilgi akışını sağlayabilmek için gerçekleşen işlem ile kontrolcü arasındaki uzaklıktan kaynaklı bir zaman gecikmesi oluşur. Tipik durum uzayı tabanlı modern kontrol algoritmaları bunun gibi sonsuz boyutlu sistemler için geçerli değildir. Bu tezde sonsuz boyutlu sistemler için çeşitli kontrol tasarım yöntemleri göz önünde bulunduruldu ve farklı tiplerde stabilize kontrolcüler türetildi. Bu tezin amacı; önceden tasarlanmış H_∞ kontrolcü ve Pade yaklaşımı ile tasarlanmış kontrolcülere alternatif, basit denetleyiciler belirlemektir. Bu amaçla, kararsız sonsuz boyutlu sistemler için altı farklı P ve PI kontrolcü elde edildi. Her bir yöntemin birbiriyle karşılaştırılması yapıldı ve her yaklaşımın avantajları gösterildi.

Anahtar sözcükler: Zaman Gecikmesi, Sabit Disk Sürücü, Sonsuz Boyutlu Sistem, $P-PI$ Kontrolcü, Gecikme Payı .

Acknowledgement

First of all, I would like to thank my supervisor Prof. Dr. Hitay Özbay for his endless guidance and support throughout my graduate studies and for revealing my enthusiasm in control theory during my undergraduate courses.

Also, I would like to express my thanks to Prof. Dr. Ömer Morgül and Prof. Dr. Mehmet Önder Efe for being a member in my thesis committee.

I am very thankful to Bilkent University EE Department for the financial support.

I would also like to thank my office mate Deniz Varol for the collaboration during my graduate studies, my best friend Başak Kepir for her encouragements to complete my thesis and my dear friend Mustafa Akın for his unlimited support whenever I need him.

Last but not least, I would like appreciate my parents Arif and Ayşegül and my sister Bilge for their endless love and support in my whole life. I believe that with this feeling, I can always motivate myself to be successful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In many feedback control applications time delay appears due to information flow between the process and the controller, or due to material transport lag and computational delays.

Thus, in real life time delays always exist in feedback systems. When neglected in the design, it may cause the system to be unstable or to make it hard to stabilize since having a time delay is similar to having infinitely many unstable zeros. If there is a time delay, system becomes infinite dimensional. Therefore, standard finite dimensional state-space methods are not applicable to such systems.

1.1 Motivation

In this thesis, by following various methods, different kinds of controllers are designed for head positioning in hard disk drives with time delay, which is an unstable infinite dimensional plant.

Applying various methods, six different controllers which are stabilizing and optimizing in terms of different performance objectives are obtained.

1.2 Related Work

One of the main focus of control systems engineering is to stabilize an unstable system using feedback. If the plant is unstable and is delay free, there are some control theory applications such as designing P, PI, PD and PID controllers, state-space methods and Linear Quadratic Regulator (LQR) controllers. For instance an LQR controller stabilizing an unstable plant is applied to the system without time delay in [4].

If the plant is unstable and also has a time delay, some approaches mentioned as a design method for unstable plants cannot be used. Since state-space methods and LQR controllers are not compatible with time delay, different types of control applications are used when plant has a time delay. Taylor or Pade approximations can be used for an unstable time-delayed plant (see [5]). But for Taylor and Pade approximations to work well, time delay amount must be relatively small. Therefore for large time delays, approximation methods cannot be applied.

Moreover, there are some methods which can be used only if the system is stable. Smith predictor controller design and Internal Model Control (IMC) methods are two of them. For instance in [6] Smith Predictor is designed and in [7] IMC method is examined. They are very popular controller design methods but there is a disadvantage of them since their extension to unstable systems is rather complicated, [8]. In this thesis, a Hard Disk Drive servo system model is considered which is unstable and has input-output time delay.

For this study the time delay is chosen as a relatively large value. Therefore Taylor and Pade approximation methods cannot be used since they are appropriate only for small time delays. Consequently, P, PI, PD, PID controllers and $H_2 - H_\infty$ controllers are more suitable for unstable and time delayed plant. In [9], [10], [11] and [12], P, PI and PID controllers are designed; and in [1], an H_∞ controller is designed.

In this study velocity type plant is considered. Therefore, a direct derivative gain cannot be used (otherwise, an unstable pole-zero cancellation would occur).

For simplicity of implementations, P and PI controller design methods for head positioning in HDD with time delays are investigated in the rest of this thesis. The results will be compared to [1]. To see other controller design examples for HDD applications, see [13] and [14], and references therein.

1.3 Contribution

In this thesis, in view of the existing work, [1], hard disk drive servo system plant is considered as a velocity controlled type plant; and a PI controller is designed. Designing an H_∞ controller is a complex issue since there are many complicated steps, that make the implementation difficult, [16]. Thus, designing P and PI controllers with various objectives, contributes to the research of controller design for HDD servo system. In our study, time delay is increased compared to the design case example in [1]. In this manner we investigate the largest tolerable time delay with respect to various performance objectives.

1.4 Organization of the thesis

The organization of the thesis is the following. In Chapter 2 some preliminaries and feedback system structure used in the rest of the thesis are explained. Then, head positioning in hard disk drive servo system plant model considered as the plant is examined.

Furthermore, in Chapter 3, there are six different design methods for P and PI controllers. Every controller has various advantages and disadvantages in view of the performance objective taken into account.

Additionally, in Chapter 4, step responses of all the six controllers are observed. Performance metrics such as percent overshoot, settling time and steady state error values are determined to find the optimal control parameters.

Finally, in Chapter 5 the thesis is concluded with a consideration of all the designed controllers, their responses and contribution to the research.

Chapter 2

Problem Definition

2.1 Feedback system with P and PI controllers

In this section the feedback system shown in Figure 2.1 is considered.

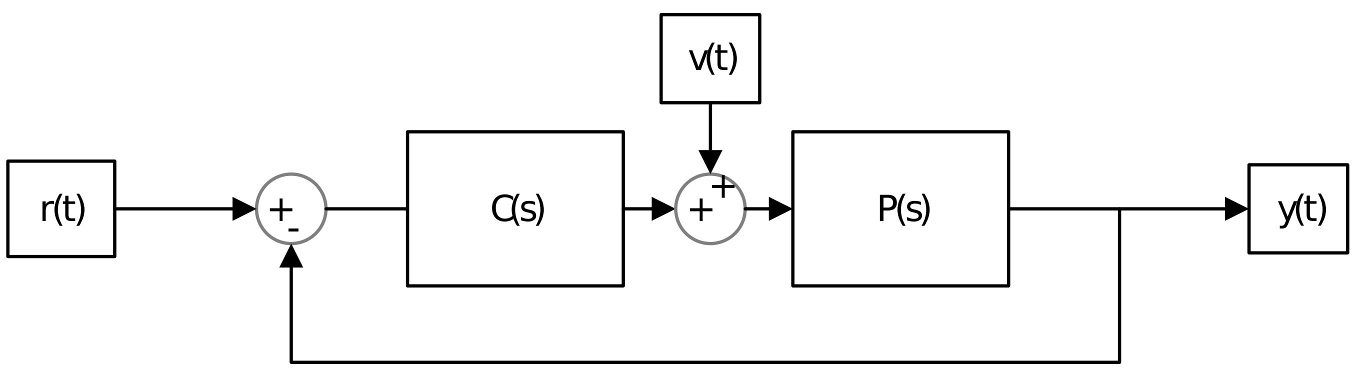

Figure 2.1: General Feedback System Representation

According to definition in [2], a feedback system formed by the controller C and the plant P is stable, if sensitivity function $S = (1 + PC)^{-1}$, as well as CS and PS are stable, i.e., these transfer functions are in H_∞. If these transfer functions are stable, it is said that the controller C is stabilizing the plant P.

In addition to this, the set of all controllers stabilizing the plant P is denoted as $\mathcal{C}(P)$.

This study involves P and PI controller type controllers. Proportional controller has a single parameter named as K_p and has a transfer function which is $C(s) = K_p$. Proportional-Integral controller has two parameters: K_p and K_i. Hence the transfer function of the PI controller becomes:

$$C(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} \tag{2.1}$$

2.2 Preliminaries

2.2.1 Delay Margin

Let $P(s) = e^{-hs}P_0(s)$, where $h \geq 0$ is the time delay, and $P_0(s)$ is a delay-free plant. Now consider a controller $C_0(s)$ which is stabilizing $P_0(s)$. If C_0 is stabilizing P, for all values of $h \in [0, h_{max})$ and the feedback system is unstable for $h = h_{max}$ then we say that h_{max} is the *delay margin* of the feedback system formed by the controller C_0 and the plant P_0.

2.2.2 Least Fragile Controller

Consider a feedback system formed by a controller C_θ and a fixed plant P, with θ representing the controller parameters in $\mathbb{R}^n$. Assume that the feedback system is stable for a fixed parameter $\theta_0 \in \mathbb{R}^n$; then by continuity the feedback system remains stable for a set of parameters θ in the neighbourhood of θ_0. The *least fragile controller* is the one where θ_0 is such that the feedback system remains stable for the largest neighbourhood around θ_0.

2.2.3 Robust Stability Condition

Consider the feedback system formed by a controller C and an uncertain plant P_Δ, where

$$P_\Delta \in \mathcal{P} := \{P + \Delta_a \ : \ |\Delta_a(j\omega)| < |W_a(j\omega)| \ \forall \ \omega\} \tag{2.2}$$

where P and P_Δ are assumed to have the same number of unstable poles. The set $\mathcal{P}$ is the set of all possible plants, P is the nominal plant, and W_a is the uncertainty bound. Given P and W_a our aim is to find a fixed controller C such that the feedback system is stable for all $P_\Delta \in \mathcal{P}$. Such a controller is called robustly stabilizing controller.

Knowing that the nominal feedback system must be stable, the Nyquist graph of the open-loop system $G_0(j\omega) = P(j\omega)C(j\omega)$ must encircle (-1) in the counter-clockwise direction as many times as number of poles of $G_0(s)$ in right half plane. When the uncertain plant is written as $P(s) + \Delta_a(s)$, $G(s)$ becomes:

$$G(s) = P(s)C(s) = (P(s) + \Delta_a(s))C(s) = G_0(s) + \Delta_a(s)C(s). \tag{2.3}$$

In Nyquist plot of $G(s)$, $G(j\omega)$ is inside a disk whose center is $G_0(j\omega)$ and radius of the disk is the following:

$$R(\omega) = |\Delta_a(j\omega)||C(j\omega)|. \tag{2.4}$$

To satisfy the robust stability, (-1) should not be inside the disk whose center is $G_0(j\omega)$ and radius is $R(\omega)$. Knowing that $W_a(j\omega)$ is the largest possible uncertainty bound. Therefore $R(\omega) = |W_a(j\omega)||C(j\omega)|$ is the largest possible radius.

Then the distance between (-1) and $G_0(j\omega)$ can be represented mathematically as:

$$|G_0(j\omega) - (-1)| \geq |W_a(j\omega)||C(j\omega)| \quad \forall\omega. \tag{2.5}$$

It can also be written as:

$$1 \geq |W_a(j\omega)C(j\omega)(1+G_0(j\omega))^{-1}| \quad \forall\omega. \tag{2.6}$$

Then, robust stability inequality(RSI) for additive uncertainty becomes:

$$||W_aC(1+PC)^{-1}||_\infty \leq 1. \tag{2.7}$$

This equation shows that as $|W_a(s)|$ increases, it becomes harder to satisfy the robust stability [3].

If multiplicative uncertainty $W_m(s)$ is used instead of additive uncertainty, plant can be written as:

$P(s) + \Delta_a(s) = P(1 + \dfrac{\Delta_a}{P}) = P(1 + \Delta_m)$ where $\Delta_m = \dfrac{\Delta_a}{P}$.

Now let $|\Delta_a(j\omega)| = |\Delta_m(j\omega)||P(j\omega)|$ and $|\Delta_m(j\omega)| < |W_m(j\omega)| \quad \forall\omega$. Then, robust stability inequality (RSI) for multiplicative uncertainty becomes:

$$||W_mPC(1+PC)^{-1}||_\infty \leq 1 \tag{2.8}$$

Since complementary sensitivity function $T(s)$ is :

$$T(s) = \frac{P(s)C(s)}{1+P(s)C(s)}. \tag{2.9}$$

RSI with multiplicative uncertainty becomes:

$$||W_mT||_\infty \leq 1. \tag{2.10}$$

2.2.4 The Small Gain Theorem

Small gain theorem is a special case of robust stability condition. If the nominal plant and the additive uncertainty are stable, to determine the robust stability inequality, small gain theorem can be used.

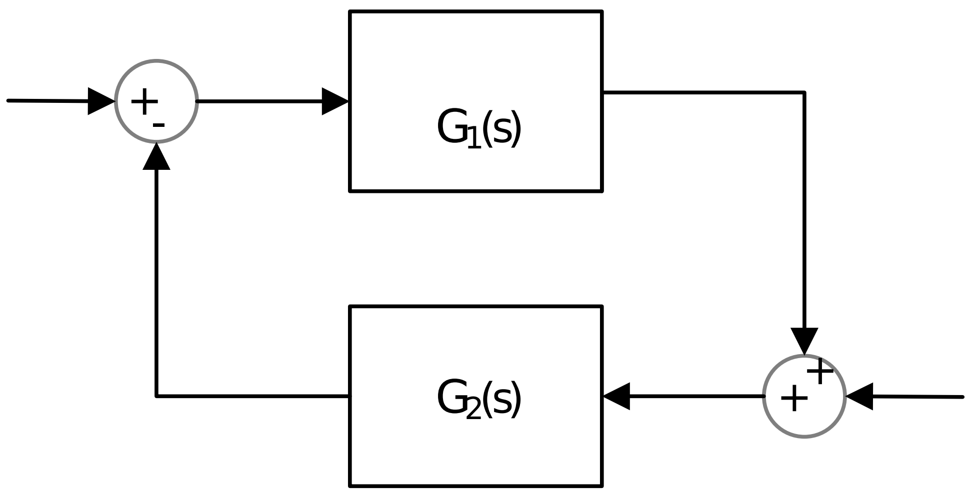

Figure 2.2: Feedback System with small gain $||G_1G_2||_\infty < 1$

Consider feedback system shown in Figure 2.2, consisting $G_1(s)$ and $G_2(s)$ which are stable linear systems. The feedback system is stable, if the small gain condition holds,

$$|G_1(j\omega)G_2(j\omega)| < 1 \quad \forall\omega. \tag{2.11}$$

Note that the small gain condition is a sufficient condition for feedback system stability. Hence, there is some conservatism: it is equivalent to having the Nyquist graph inside the unit circle implying that $G_1(j\omega)G_2(j\omega)$ does not encircle (-1).

2.3 Hard Disk Drive Plant Model

As it is widely known, Hard Disk Drive (HDD) stores digital information using rapidly rotating disks called platters.

Electronics present on HDDs are generally controlling the actuator and rotation of the disk. These rotations and movements of actuator allows the controller to read and write data on HDD. Data on disk is usually are concentrated in circles and in this sense servo systems are used by the electronics of the HDD. Servo control movements include the following modes: track-following, track seeking and seek settling.

In track-following mode, head is positioned within the track Off-Center-Limit. In this mode, many disturbances can happen and it causes track misregistration. Servo control algorithms attenuate both periodic and random disturbances. Track seeking allows to determine the trajectory of changing between different tracks. Seek settling allows finding the transition from track seeking mode to track following mode. In seek settling mode, transient behaviours can be observed.

In a single stage HDD, the servo system is composed by two components. Voice Coil Motor (VCM) allows the magnetic head to be actuated, and Position Error Signal (PES) allows the position to be measured by the readings from the servo information.

The dynamics of HDD servo system can be modelled as follows, [1], (transfer function from voltage applied to the motor to the track position)

$$P(s) = \frac{K_{DC}e^{-hs}}{s^2} T_s(s) T_m(s) \tag{2.12}$$

where $K_{DC} = K_0$ which is the nominal DC gain, h is the time delay and $T_s(s)$ is a second order term including dominant flexible modes:

$$T_s(s) = \frac{A_s(s)}{B_s(s)} = \frac{s^2 + 2\xi_{z,0}\omega_{n_{z,0}} s + \omega_{n_{z,0}}^2}{s^2 + 2\xi_{p,0}\omega_{n_{p,0}} s + \omega_{n_{p,0}}^2} \tag{2.13}$$

where $\xi_{z,0}$ and $\xi_{p,0}$ are the damping ratios of the zeros and poles, and $\omega_{n_{z,0}}$ and $\omega_{n_{p,0}}$ are the national frequencies of the zeros and poles.

The transfer function $T_s(s)$ is the first translational mode which is also known as system mode, therefore it is used in nominal plant model. For high frequency resonant modes $T_m(s)$ is modelled as:

$$T_m(s) = \prod_{i=1}^{N} \frac{\frac{1}{\omega_{n_{z,i}}^2} s^2 + 2\frac{\xi_{z,i}}{\omega_{n_{z,i}}} s + 1}{\frac{1}{\omega_{n_{p,i}}^2} s^2 + 2\frac{\xi_{p,i}}{\omega_{n_{p,i}}} s + 1} \tag{2.14}$$

where i represents different resonant modes and $\xi_{z,i}$, $\xi_{p,i}$, $\omega_{n_{z,i}}$ and $\omega_{n_{p,i}}$ are the damping ratios and natural frequencies of the ith resonant mode.

For nominal plant, only $T_s(s)$ is used,

$$P_{nominal}(s) = \frac{K_0 e^{-hs}}{s^2} \left(\frac{s^2 + 2\xi_{z,0}\omega_{n_{z,0}} s + \omega_{n_{z,0}}^2}{s^2 + 2\xi_{p,0}\omega_{n_{p,0}} s + \omega_{n_{p,0}}^2} \right) \tag{2.15}$$

When PI controller is designed velocity feedback control is used instead of position control. Therefore, for this purpose velocity feedback model of the nominal plant is,

$$P_0(s) = \frac{K_0 e^{-hs}}{s} \left(\frac{s^2 + 2\xi_{z,0}\omega_{n_{z,0}} s + \omega_{n_{z,0}}^2}{s^2 + 2\xi_{p,0}\omega_{n_{p,0}} s + \omega_{n_{p,0}}^2} \right) \tag{2.16}$$

In this thesis, the design case study considered in [1] is taken as a plant model. The HDD considered has the nominal plant model parameters shown in Table 2.1.

K_0	h	$\xi_{z,0}$	$\xi_{p,0}$	$\omega_{n_{z,0}}$	$\omega_{n_{p,0}}$
5.2269×10^8	6×10^{-5}	0.99	0.018	1.244×10^5	5.29×10^4

Table 2.1: Coefficients used in plant model

In the design case of [1], the time delay "h" is relatively small. Therefore, time delay amount can be increased. To find the amount of maximum tolerable time delay, a delay margin analysis is done. For detailed delay margin analysis, see Section 3.1. The maximum tolerable time delay is found as 0.54 sec. For the rest of this study, the nominal time delay value h is considered as 0.01 sec.

After putting the given parameters in Table 2.1, plant dynamics becomes:

$$P_0(s) = \frac{5.2269 \times 10^8 e^{-0.01s}}{s} \left(\frac{s^2 + 2 \times 0.99 \times 1.244 \times 10^5 s + (1.244 \times 10^5)^2}{s^2 + 2 \times 0.018 \times 5.29 \times 10^4 s + (5.29 \times 10^4)^2} \right) \tag{2.17}$$

$$P_0(s) = e^{-0.01s} \left(\frac{5.227 \times 10^8 s^2 + 1.287 \times 10^{14} s + 8.089 \times 10^{18}}{s^3 + 1904 s^2 + 2.798 \times 10^9 s} \right) \tag{2.18}$$

Our goal is to design P and PI controllers for this plant and examine their performances from different perspectives.

Chapter 3

Overview of Controller Design Methods

3.1 Method 1: Design of Delay Margin Maximizing PI Controller

In this section we consider the feedback system shown in Figure 3.1.

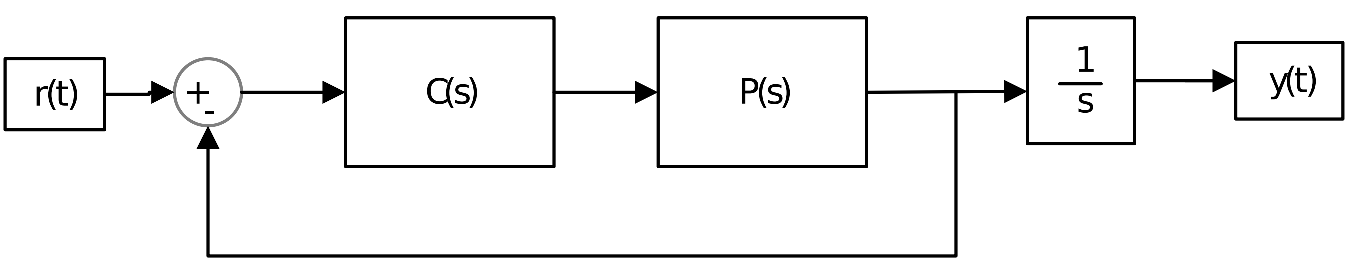

Figure 3.1: Closed loop system representation where $C(s)$ is the controller of the system and $P(s)$ is the plant of the HDD system.

First method's main objective is maximizing the delay margin. (See Section 2.2.1) Therefore at first a delay margin analysis for the nominal plant without time delay is needed. So the plant model for this section is the following:

$$P_0(s) = \left(\frac{5.227 \times 10^8 s^2 + 1.287 \times 10^{14} s + 8.089 \times 10^{18}}{s^3 + 1904 s^2 + 2.798 \times 10^9 s} \right).$$

At first stabilizing K_p interval for the above mentioned plant is found by examining the root locus plot of that plant. The root locus plot of the plant is the following:

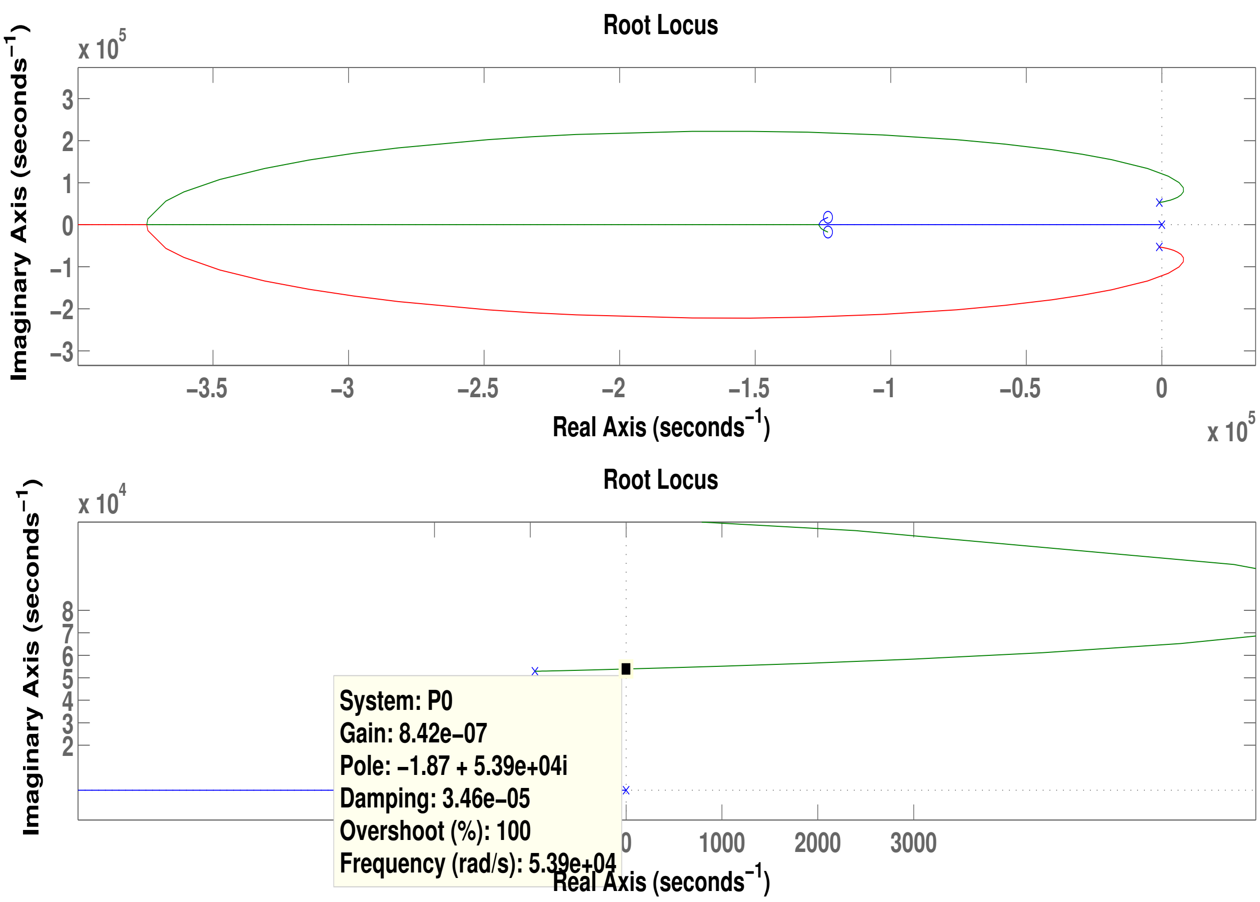

Figure 3.2: Root locus plot of the plant $P_0(s)$

As it is seen from the above graph, the stabilizing K_p interval is found as $K_p \in (0, 8.42 \times 10^{-7})$. There are other stable K_p values as seen from the root locus plot, but for high gain delay margin becomes very low. Therefore, the above region is chosen as the stabilizing proportional gain interval. Considering the PI controller as follows:

$$C(s) = \frac{K_p s + K_i}{s} = K_p \left(\frac{s + \frac{K_i}{K_p}}{s} \right) = \frac{K_p}{s}(s + \alpha)$$

where $\alpha = \dfrac{K_i}{K_p}$ is considered as a zero of the controller. By trial and error we determine that for the above K_p interval the stabilizing values of α are in the interval $\alpha \in (10^{-3}, 1)$.

Then we perform the delay margin analysis for different K_p and α values among the intervals found above. For this purpose Matlab's `allmargin` command is very helpful. It uses the open-loop system $P \times C$ as an input and gives gain margin, phase margin, delay margin, crossover frequencies and stability condition of that system. If the system is stable, stability condition resulting from the `allmargin` command is 1. Otherwise it is 0.

By checking the stability condition for all $K_p - K_i$ pairs, stabilizing region of the controller parameters are found. Afterwards, for these pairs delay margins are computed.

Delay Margin versus K_p plot is as follows:

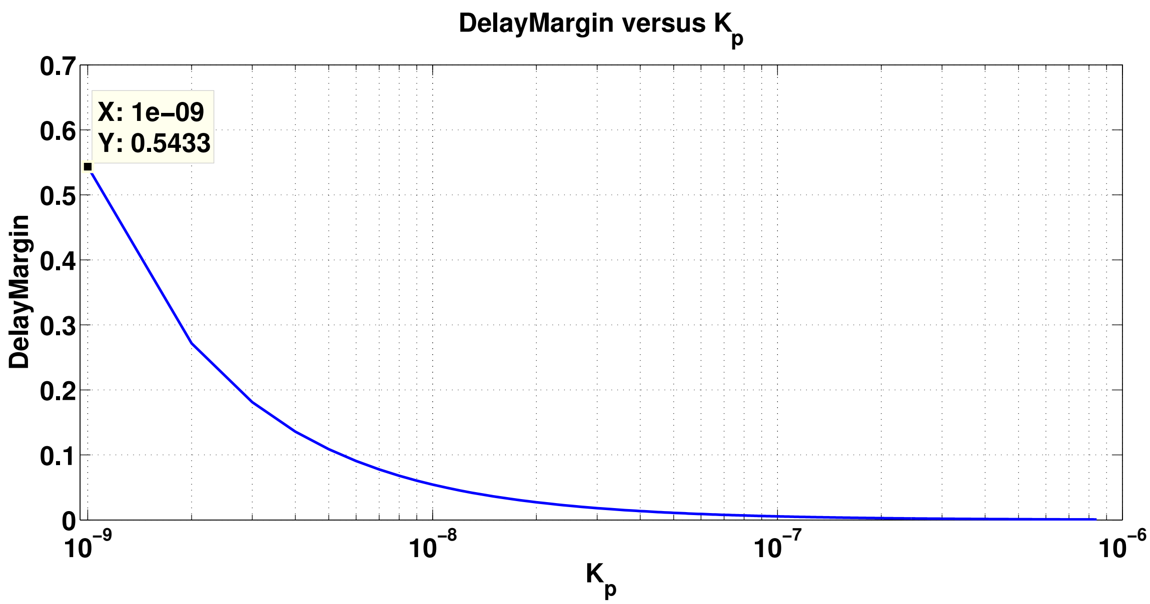

Figure 3.3: Delay Margin versus K_p

Among these computed values, maximum delay margin for the plant is 0.54 for $K_p = 10^{-9}, \alpha = 10^{-3}, K_i = 10^{-12}$. Therefore optimal controller for in terms of maximizing delay margin is as follows:

$$C(s) = \frac{10^{-9}s + 10^{-12}}{s}. \tag{3.1}$$

The best delay margin is obtained with approximately zero controller consistent with the fundamental result of robust controller which says that the robustly optimal controller for an unknown stable plant is zero.

To see this recall the robust stability condition (see Section 2.2.3),

$||WC(1 + PC)^{-1}||_\infty < 1.$

Note that $C = 0$ always solves this problem for any W if the plant is stable. But since P has a pole at $s = 0$ as seen from root locus, a small non-zero gain in C is needed to stabilize plant.

Simulations and performance results corresponding to this design can be found in Chapter 4.

3.2 Method 2: Design of Least Fragile P and PI Controllers

The second method considers design of least fragile (see Section 2.2.2) P and PI controller for the given servo system dynamics for hard disk drive.

Plant model for this section can be found in (2.18).

3.2.1 Least Fragile P Controller

In this section we consider the feedback system shown in Figure 3.4.

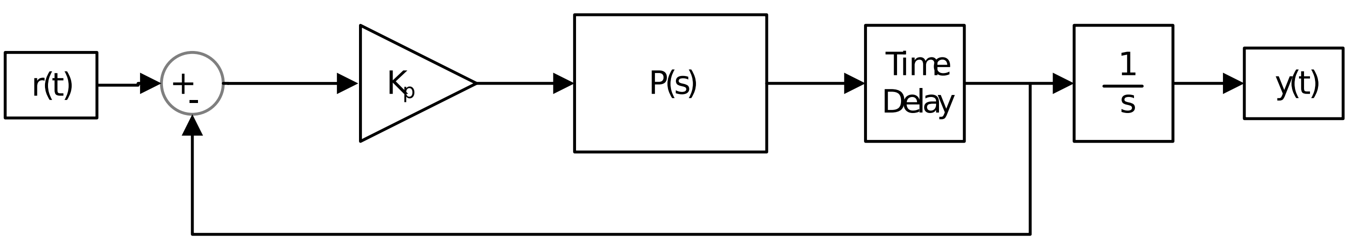

Figure 3.4: Closed loop system representation where $C(s)$ is the controller of the system and $P(s)$ is the plant of the HDD system.

In this section, velocity control type plant is used with the proportional controller $C(s) = K_p$, as shown in Figure 3.4. Again by using MATLAB's "allmargin" command, which gives us whether the closed loop system is stable or not, a stable proportional gain (K_p) interval is found for the given unstable time-delayed plant. "Allmargin" command uses open-loop transfer function as an input and gives us "1" or "0" referring stable or unstable. At first a huge K_p

interval is considered. Then among that interval stabilizing K_p interval is found by keeping the K_p values which "allmargin" gives "1" as output.

Therefore by finding that K_p interval, different kinds of P controllers stabilizing the plant are designed. The proportional gain interval is $K_p \in (10^{-9}, 5.4 \times 10^{-8})$

To find the least fragile P controller among all K_p values in the admissible region the midpoint of the interval is chosen, that is $K_p = 2.7 \times 10^{-8}$,

$$C(s) = 2.7 \times 10^{-8}. \tag{3.2}$$

3.2.2 Least Fragile PI Controller

In this section we consider the feedback system shown in Figure 3.5.

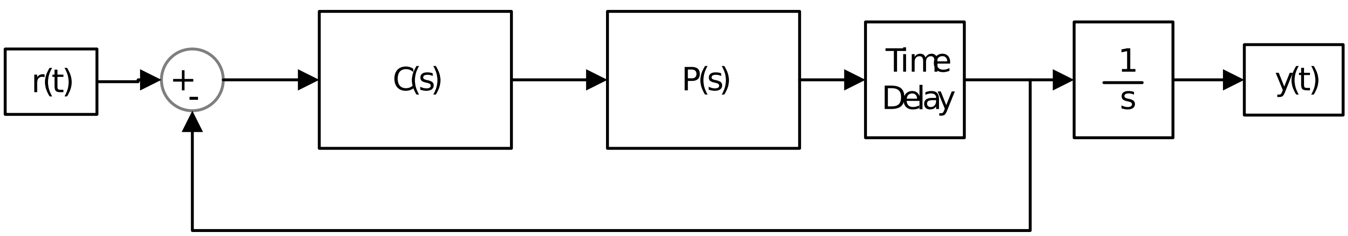

Figure 3.5: Closed loop system representation where $C(s)$ is the controller of the system and $P(s)$ is the plant of the HDD system.

After finding the allowable proportional gain interval, PI controller is designed for the velocity controlled type plant as shown in the Figure 3.5. To obtain stabilizing proportional and integral gain pairs, MATLAB's "allmargin" command is used as discussed before. Again, let us consider two large K_p and K_i intervals. For the values in this region, by using "allmargin" command, stable $K_p - K_i$ pairs are saved.

All stabilizing $K_p - K_i$ pairs are shown in the following figure as the colored area.

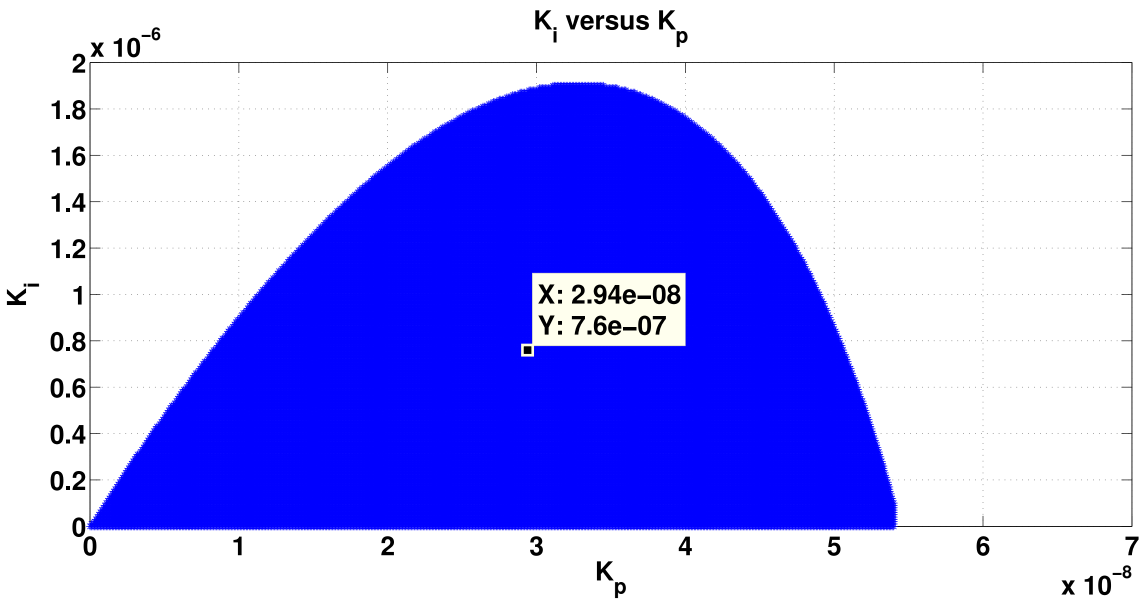

Figure 3.6: Area representing the stable $K_p - K_i$ pairs for Method 2.2

To find the least fragile PI controller, center of the shaded area (in the sense that around this center we can place the largest circle staying inside the allowable region) is chosen. The center is found as

$$(K_p, K_i) = (2.94 \times 10^{-8}, 7.6 \times 10^{-7}).$$

In conclusion the optimal PI controller is the following:

$$C(s) = \frac{2.94 \times 10^{-8}s + 7.6 \times 10^{-7}}{s}. \tag{3.3}$$

Simulations and performance results corresponding to this design can be found in Chapter 4.

3.3 Method 3: Design of Least Fragile Integral Action PI Controller

In this section we consider the feedback system shown in Figure 3.7.

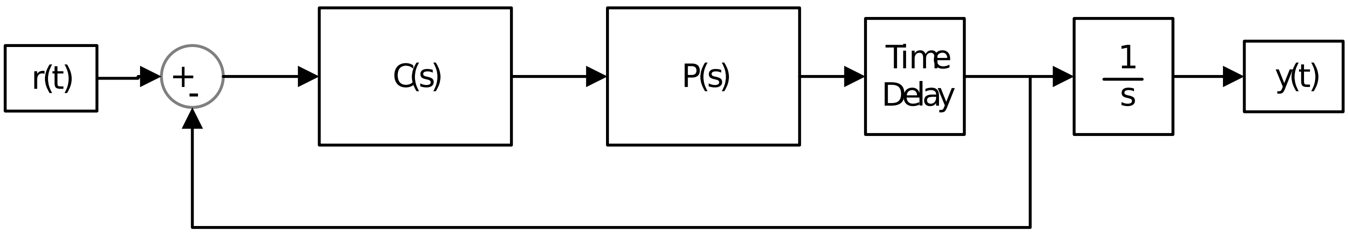

Figure 3.7: Closed loop system representation where $C(s)$ is the controller of the system and $P(s)$ is the plant of the HDD system.

Plant model for this section can be found in (2.18).

For this method least fragile (see Section 2.2.2) integral action PI controller is designed using the method of [2]. The design is applied to the velocity control type plant $P_0(s)$ as shown in Figure 3.7. This method involves designing PI controller in the form:

$$C_{pi}(s) = K_p + \frac{K_i}{s} = C_1(s) + \frac{K_i}{s} \quad \text{where} \quad C_1(s) = K_p. \tag{3.4}$$

This K_p is a proportional gain which already stabilizes the nominal plant $P_0(s)$, so that $C_1(s) = K_p \in \mathcal{C}(P)$.

Then define;

$$H_1(s) = \frac{P(s)}{1 + C_1(s)P(s)} \quad \text{which is in} \quad H_\infty. \tag{3.5}$$

For the considered design case $H_1(s)$ becomes ;

$$H_1(s) = \frac{e^{-hs}(5.227 \times 10^8 s^2 + 1.287 \times 10^{14} s + 8.089 \times 10^{18})}{2.798 \times 10^9 s + 1904 s^2 + s^3 + K_p e^{-hs}(5.227 \times 10^8 s^2 + 1.287 \times 10^{14} s + 8.089 \times 10^{18})}. \tag{3.6}$$

Assuming that designed PI controller is C_2. Characteristic equation of the system consisting controller C_2 and plant P is the following:

$$1 + C_1(s)P(s) + \frac{K_i}{s}P(s) = (1 + C_1(s)P(s))\left(1 + \frac{K_i}{s}H_1(s)\right) = 0. \tag{3.7}$$

Then defining

$$V_1(s) = \left(1 + \frac{K_i}{s}H_1(s)\right), \tag{3.8}$$

the feedback system is stable if V_1^{-1} is in H_∞.

Since $C_1(s) = K_p \in \mathcal{C}(P)$, if V_1^{-1} is in H_∞, then it can be concluded that the designed PI controller $C_2(s) = K_p \in \mathcal{C}(P)$.

Now let $b := K_i H_1(0) > 0$ where $H_1(0) = \dfrac{1}{K_p}$, then V_1 can be written as

$$V_1(s) = \left(1 + \frac{b}{s}\right) + b\left(\frac{H_1(s)H_1(0)^{-1} - 1}{s}\right). \tag{3.9}$$

$$V_1(s) = \left(1 + \frac{b}{s}\right)\left(1 + \left(1 + \frac{b}{s}\right)^{-1} b\left(\frac{H_1(s)H_1(0)^{-1} - 1}{s}\right)\right). \tag{3.10}$$

$$V_1(s) = \left(1 + \frac{b}{s}\right)\left(1 + \left(\frac{b}{s+b}\right)\left(H_1(s)H_1(0)^{-1} - 1\right)\right). \tag{3.11}$$

Assuming that $G_1(s) = \left(\frac{b}{s+b}\right)$ and $G_2(s) = \left(H_1(s)H_1(0)^{-1} - 1\right)$, by the small gain theorem explained in Section 2.2.4, if b satisfies the below inequality, it can be concluded that $V_1^{-1} \in H_\infty$ and hence $C_2 \in \mathcal{C}(P)$:

$$\| \frac{b}{s+b} \left(H_1(s)H_1(0)^{-1} - 1\right) \|_\infty < 1. \tag{3.12}$$

From Section 3.2.1, stabilizing K_p interval is found as $K_p \in (10^{-9}, 5.4 \times 10^{-8})$; and with that K_p interval and a random large b interval is taken. In this interval b values which satisfy the inequality (3.12) for fixed K_p is found for each fixed K_p value.

Then, for each K_p value, a maximum b value is found among the b values satisfying the stability condition. Therefore at the end, b_{max} versus K_p pairs are obtained for every different K_p. Accordingly, the least fragile integral action gain is defined as the following:

$$K_{i,opt} = \frac{b_{max}(K_{p,opt})}{2} H_1(0)^{-1}. \tag{3.13}$$

For this case it is mentioned that $H_1(0) = \frac{1}{K_p}$. Therefore, least fragile integral action gain is:

$$K_{i,opt} = \frac{b_{max}(K_{p,opt})}{2} K_p.$$

The above mentioned computations for the problem at hand are as follows.

Plot of the area representing the allowable $K_p - K_i$ pairs is shown in Figure 3.8.

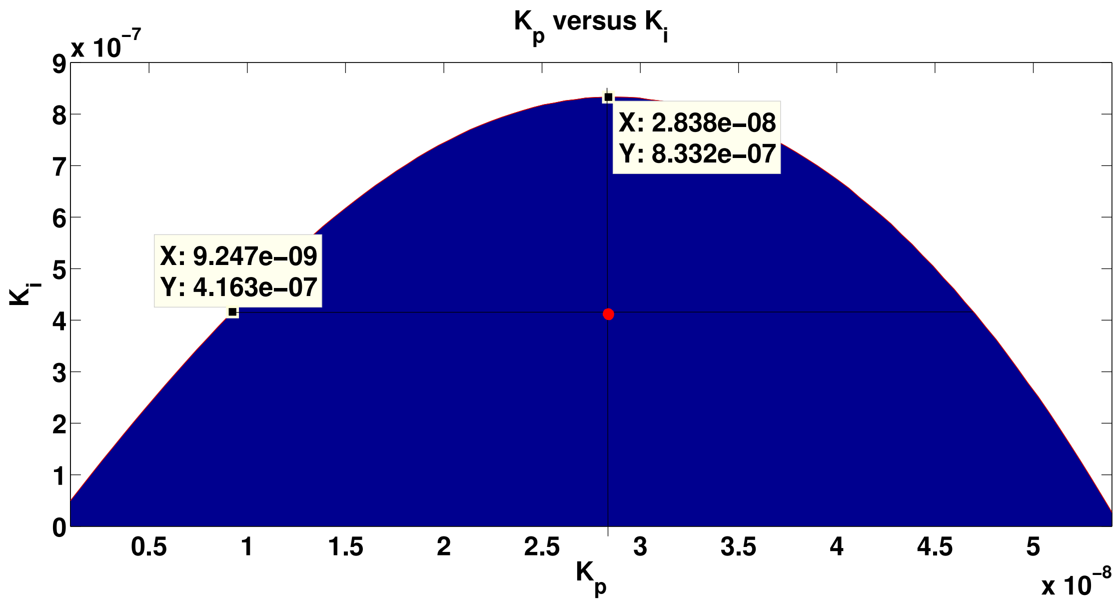

Figure 3.8: Area representing the stable $K_p - K_i$ pairs for Method 3

The largest K_i value is obtained for $K_{p,opt} = 2.83 \times 10^{-8}$. This gives $K_{i,max} = 8.332 \times 10^{-7}$, Then the least fragile integral action gain is $K_{i,max}/2 = 4.16 \times 10^{-7}$. Thus the optimal PI controller of this section has the parameters $(K_p, K_i) = (2.83 \times 10^{-8}, 4.16 \times 10^{-7})$, which gives us

$$C(s) = \frac{2.83 \times 10^{-8} s + 4.16 \times 10^{-7}}{s}. \tag{3.14}$$

Simulations and performance results corresponding to this design can be found in Chapter 4.

3.4 Method 4: Design of P and PI Controller Considering Position Tracking

Method 4 differs from the above three methods since it is concerned with performance issues for position tracking in addition to the stability issue. It mainly focuses designing a P and PI controller to a closed loop system which is already stable. Again plant model is:

$$P_0(s) = \frac{K_0 e^{-hs}}{s}\left(\frac{s^2 + 2\xi_{z,0}\omega_{n_{z,0}}s + \omega_{n_{z,0}}^2}{s^2 + 2\xi_{p,0}\omega_{n_{p,0}}s + \omega_{n_{p,0}}^2}\right) = \frac{K_0 e^{-hs}}{s}\widehat{P}(s)$$

$$where \quad \widehat{P}(s) = \frac{s^2 + 2\xi_{z,0}\omega_{n_{z,0}}s + \omega_{n_{z,0}}^2}{s^2 + 2\xi_{p,0}\omega_{n_{p,0}}s + \omega_{n_{p,0}}^2}. \tag{3.15}$$

3.4.1 Optimal P Controller

In this section the feedback system shown in Figure 3.9 is considered.

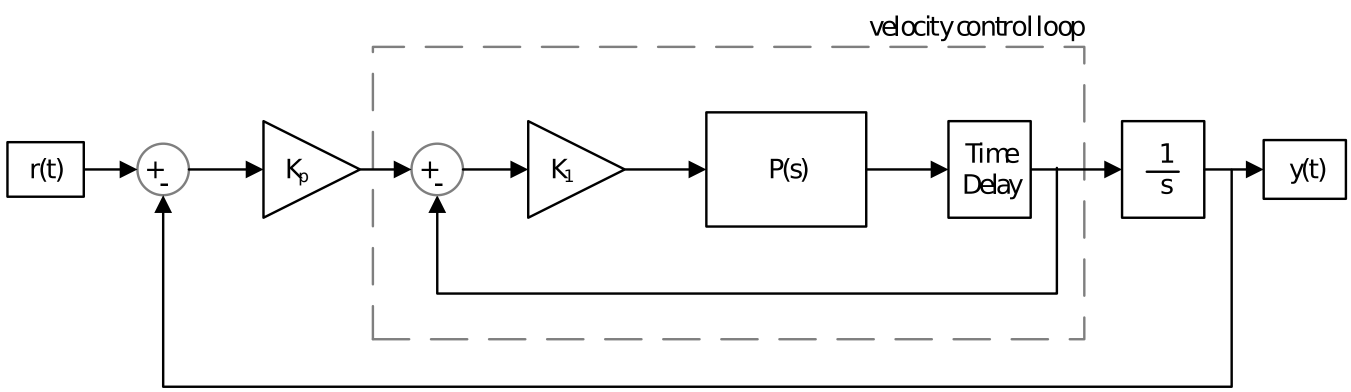

Figure 3.9: Closed loop system representation where K_p and K_1 are the controllers of the system and $P(s)$ is the plant of the HDD system.

Assuming that $K_1 = K_v K_0$ where K_v is the proportional action and $\overline{P}(s) = \frac{e^{-hs}}{s}\widehat{P}(s)$.

Then considering the above feedback system, the interval velocity loop characteristic equation can be obtained like:

$$1 + \frac{K_v K_0}{s} e^{-hs} \widehat{P}(s) = 0. \tag{3.16}$$

To determine the stabilizing K_1 range, similar to Section 2 "*allmargin*" command is used. Then finally admissible K_1 interval, $K_1 \in (K_{1min}, K_{1max}) = (0.5227, 28.2253)$ is obtained.

After finding the K_1 range, sensitivity function is computed. Knowing that sensitivity function is:

$$S(s) = \frac{1}{1 + P(s)C(s)} = \frac{1}{1 + \overline{P}(s)K_1}. \tag{3.17}$$

For this K_1 range, peak points of $|S(j\omega)|$ are found for different K_1 values and named as β which equals to $||S||_\infty$. Then, again for the known K_1 range, the natural frequency values which make the sensitivity function equals to $\frac{1}{\sqrt{2}} = 0.707$ are found and named as ω_c (bandwidth).

After computing β and ω_c values, the cost function is chosen as $\frac{\beta}{\omega_c}$ since ω_c is wanted to be maximized.

Note that in a feedback system to maximize stability robustness against combined gain and phase perturbations we need to make β small [3].

For good tracking performance we need high bandwidth ω_c. So $\frac{\beta}{\omega_c}$ is a good blend for a cost function to be minimized.

The cost $\dfrac{\beta}{\omega_c}$ versus corresponding K_1 value graph is plotted as the following:

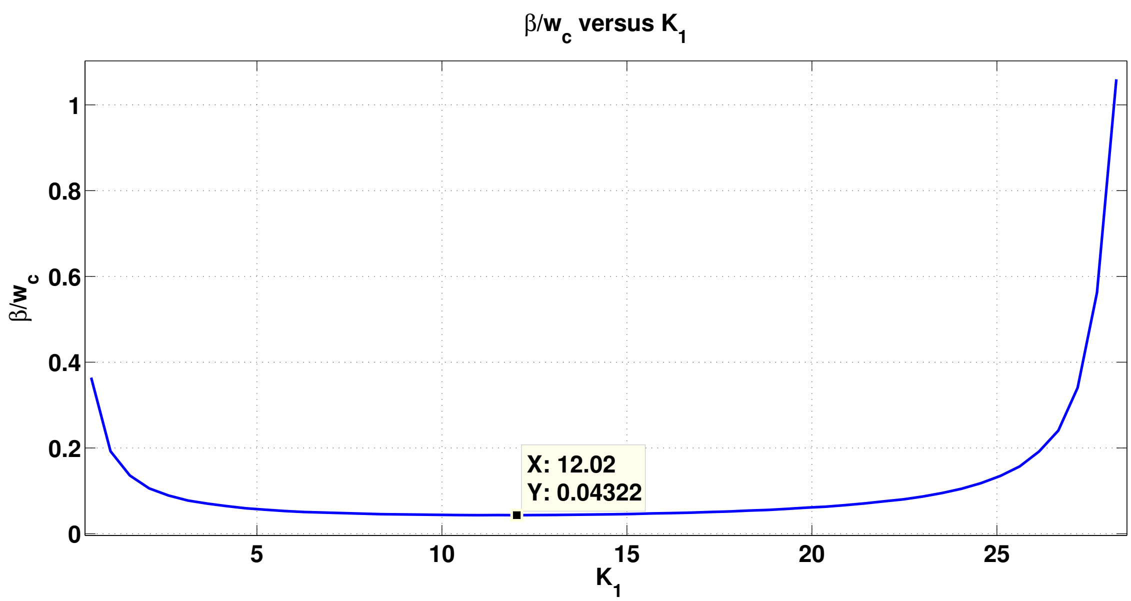

Figure 3.10: The cost $\dfrac{\beta}{\omega_c}$ versus corresponding K_1 value

According to the above graph, the minimum of $\dfrac{\beta}{\omega_c}$ is observed when $K_1 = 12.02$. Hence this K_1 value is chosen as the optimal K_1 value and named as K_{1opt}.

Sensitivity function for $K_{1opt} = 12.02$ shown in Figure 3.11.

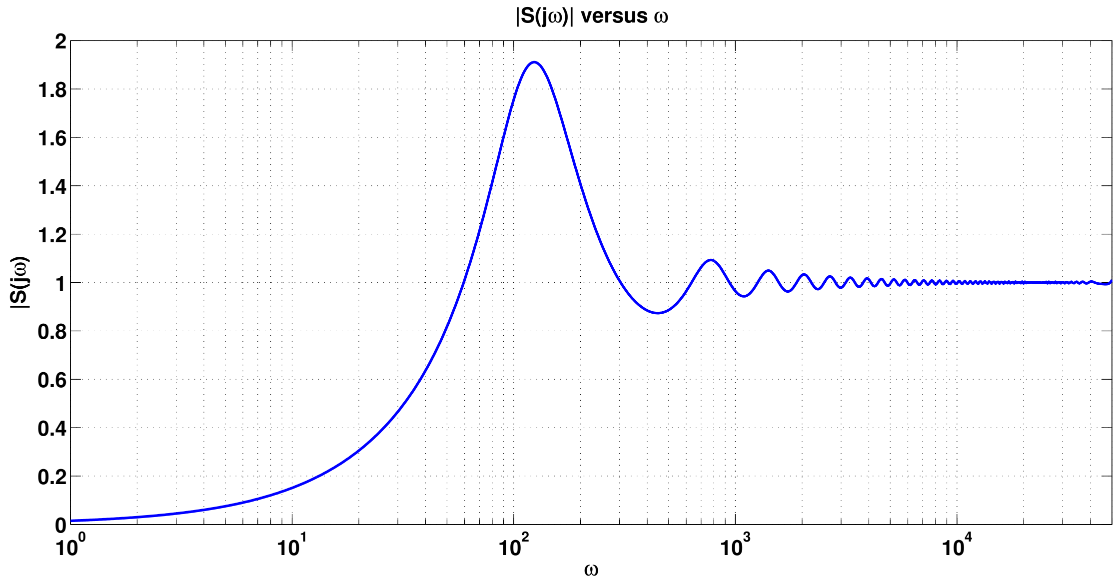

Figure 3.11: $|S(j\omega)|$ versus ω value

Afterwards, the internal closed loop transfer function becomes the velocity control type plant model of the outer closed loop feedback system. Therefore the closed loop transfer function of the velocity control loop is the following:

$$T_{velocity}(s) = \frac{K_{1opt} e^{-hs} \frac{\widehat{P}(s)}{s}}{1 + K_{1opt} e^{-hs} \frac{\widehat{P}(s)}{s}}. \tag{3.18}$$

For the outer loop, position control loop will be designed. The plant model for the outer loop is

$$P_{outer}(s) = \frac{K_{1opt} e^{-hs} \frac{\widehat{P}(s)}{s^2}}{1 + K_{1opt} e^{-hs} \frac{\widehat{P}(s)}{s}}. \tag{3.19}$$

Similar to the inner loop, same steps are followed to find the optimal K_p value. This plant has a time delay in its denominator, hence K_p values cannot be found by checking the stability with "*allmargin*" command, since "*allmargin*" command does not work for this type of plants. Therefore, gain margin analysis is needed to find the upper limit of K_p. Gain margin of the plant can be found from the Bode plot of $P_{outer}(s)$, which is shown in Figure 3.12

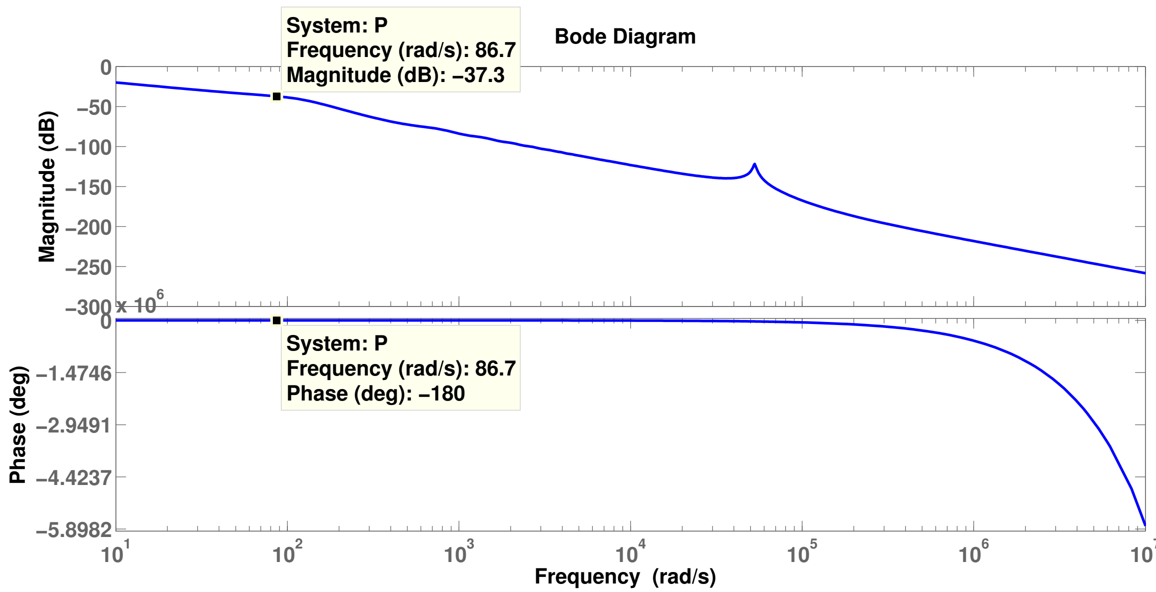

Figure 3.12: Bode plot of $P_{outer}(s)$

Crossover frequencies can be seen in the above figure. According to that plot, it crosses -180 degrees at the frequency 86.7 rad/s, where the magnitude is -37.3 dB. As a result, gain margin is computed as $37.3dB \equiv 73.28$.

Since upper limit of K_p is found as 73.28, K_p values are chosen in $(0, 73.28)$. For every K_p value, similar to the first step, sensitivity function is computed. From the sensitivity function, β and w_c values are observed.

The cost $\dfrac{\beta}{w_c}$ versus corresponding K_p value graph is shown in Figure 3.13.

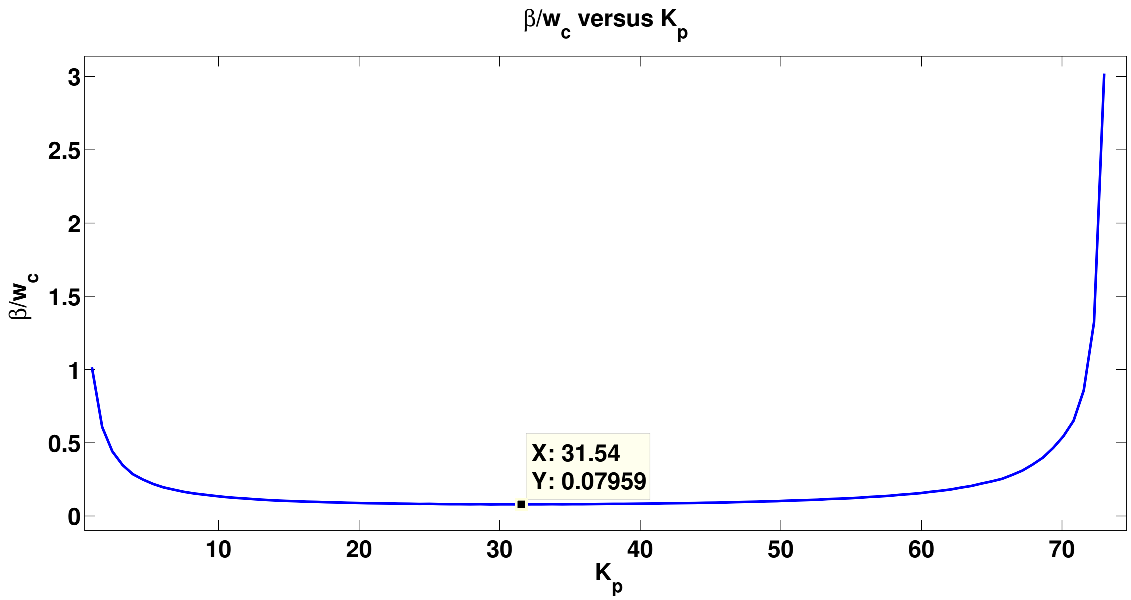

Figure 3.13: The cost $\dfrac{\beta}{\omega_c}$ versus corresponding K_p value graph

According to the above graph, the minimum of $\dfrac{\beta}{w_c}$ is observed when $K_p = 31.54$. Hence this K_p value is chosen as the optimal K_p value and named as K_{popt}.

Sensitivity function for $K_{popt} = 31.54$ is shown in Figure 3.14.

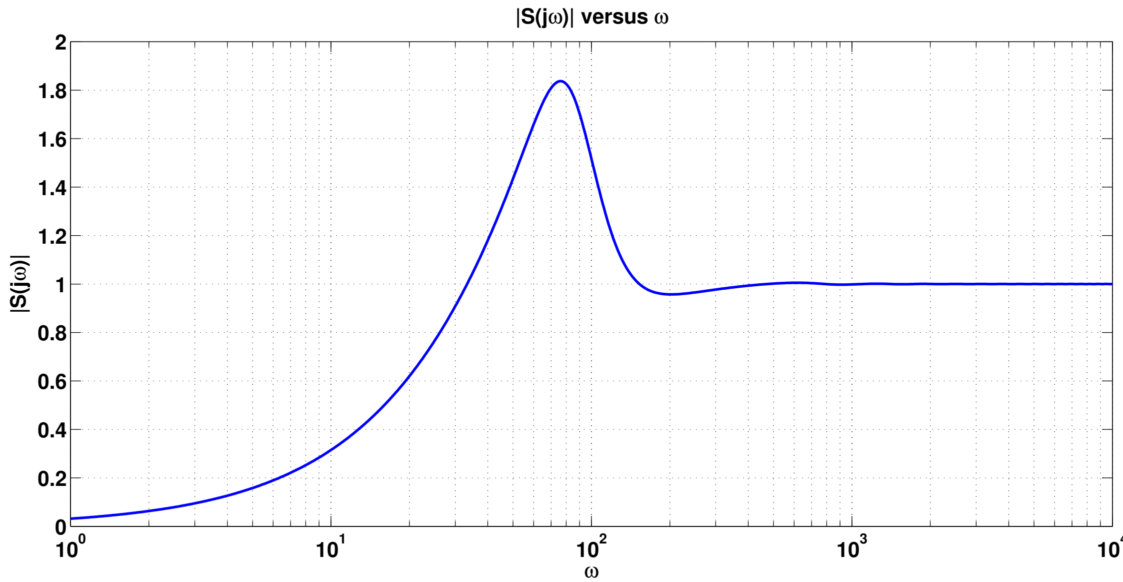

Figure 3.14: $|S(j\omega)|$ versus ω value

As a conclusion, the inner controller is chosen as

$$C_{inner} = K_{1opt} = 12.02, \tag{3.20}$$

and the outer controller is designed as

$$C_{outer} = K_{popt} = 31.54. \tag{3.21}$$

The resulting $\beta \cong 1.8$ and $\omega_c \cong 20$ rad/sec.

3.4.2 Optimal PI Controller

In this section we consider the feedback system shown in Figure 3.15.

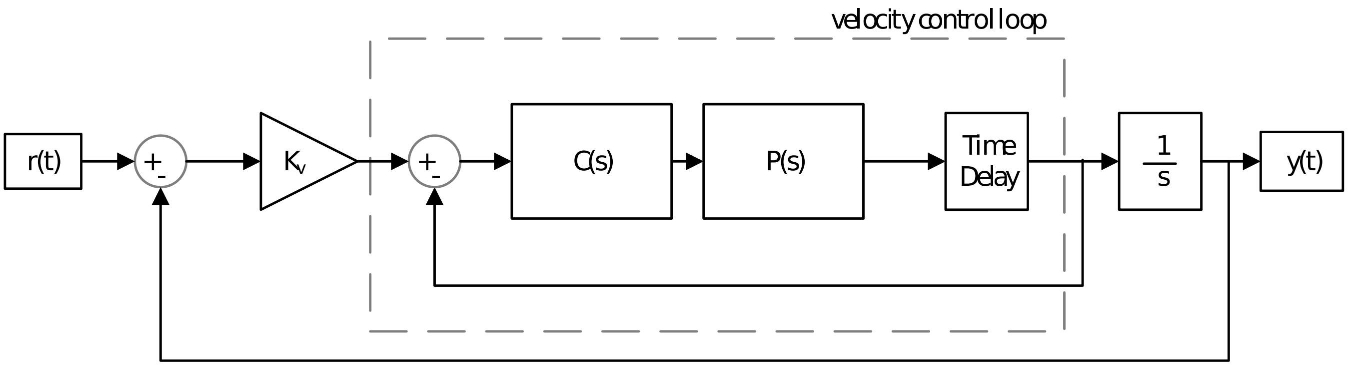

Figure 3.15: Closed loop system representation where K_v and $C(s)$ are the controllers of the system and $P(s)$ is the plant of the HDD system.

The approach taken here is similar to Section 3.4.1, but it differs with its internal closed loop system. For this method, optimal PI controller is designed for the inner velocity loop. Then, the internal closed loop system transfer function is considered as the plant and an optimal P controller is designed for that plant.

To examine the optimal PI controller parameters, stabilizing $K_p - K_i$ pairs found in Section 3.2.2 are used. By checking robust stability condition (see Section 2.2.3) determined by the uncertainty weight $W_2(s)$, sensitivity analysis can be done. In this problem the uncertainty weighting function is defined as the following (see [1]):

$$W_2(s) = 0.3125 + 9.4211 \times 10^{-6} s \tag{3.22}$$

Let us check $\| W_2 T \|_\infty$ named as γ for all stabilizing $K_p - K_i$ pairs where T is the complementary sensitivity function and defined as:

$$T = 1 - S(s) = \frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{P_0(s)C(s)}{1 + P_0(s)C(s)} \text{ where } C(s) = \frac{K_p s + K_i}{s}$$

To satisfy the robust stability inequality in Section 2.2.3, all γ values less than 1 are selected and $K_p - K_i$ pairs which make γ values less than 1 are used from now on.

Then for remaining $K_p - K_i$ pairs, complementary weighting sensitivity functions are computed and w_c values are obtained. The maximum crossover frequency (bandwidth) w_c is observed as 148.36 rad/s when $K_p = 1.9 \times 10^{-9}$ and $K_i = 3 \times 10^{-8}$. Note that this PI controller design is for the plant model $P_0(s)$ which contains a large gain named as K_0, that's why the designed K_p and K_i parameters are considerably small. But when K_0 is multiplied by these parameters, PI controller actually becomes:

$$C(s) = \frac{0.99s + 15.68}{s}. \tag{3.23}$$

Since these parameter makes w_c maximum, they are named as K_{popt} and K_{iopt}. So, the inner controller becomes $C_{inner}(s) = \dfrac{1.9 \times 10^{-9}s + 3 \times 10^{-8}}{s}$.

Afterwards, the inner velocity loop transfer function becomes the velocity control type plant model of the outer closed loop feedback system. For the outer loop, position control type plant is designed. So, the plant model transfer function is as follows:

$$P_{outer}(s) = \frac{(\frac{K_{popt}s + K_{iopt}}{s})\frac{P_0(s)}{s}}{1 + (\frac{K_{popt}s + K_{iopt}}{s})P_0(s)}. \tag{3.24}$$

The optimal K_v is determined from similar steps followed in the design method described in Section 3.4.1. Therefore, gain margin analysis is needed to find the upper limit of K_v.

Gain margin of the plant can be found from the Bode plot of $P_{outer}(s)$, shown in Figure 3.16.

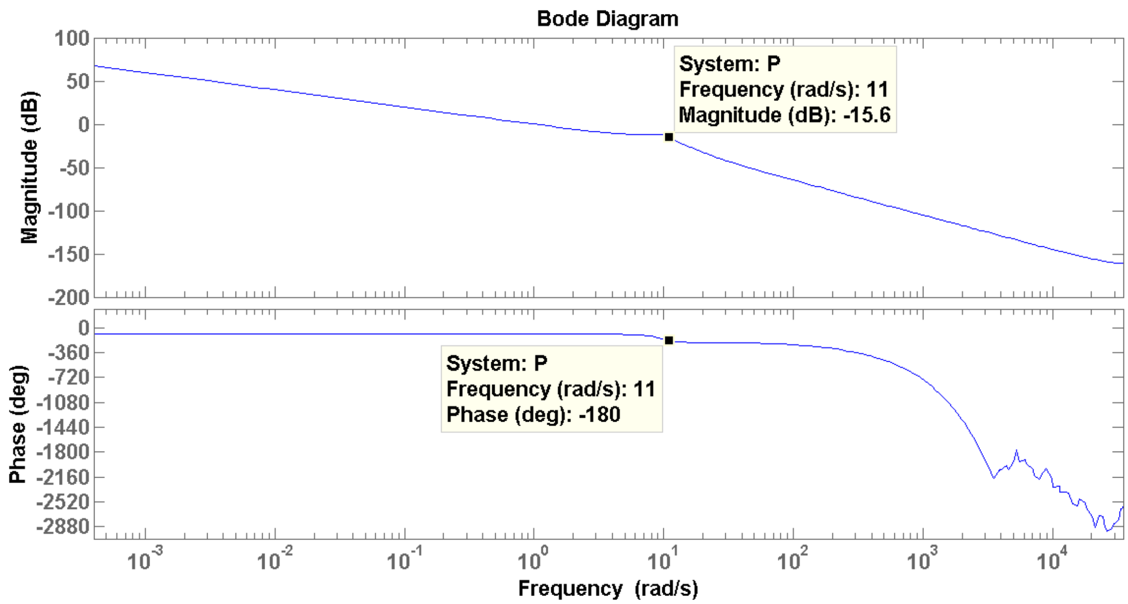

Figure 3.16: Bode plot of $P_{outer}(s)$

Crossover frequencies can be seen in the above figure. According to that plot, it crosses -180 degrees when the frequency is 11 rad/s. Then, the magnitude at that frequency is -15.6 dB. As a result, the gain margin is computed as $15.6dB \equiv 5.95$.

For every K_v value in the range $(0, 5.95)$, sensitivity function is computed. From the magnitude of the sensitivity function β and w_c values are computed.

The cost $\dfrac{\beta}{\omega_c}$ versus K_v value graph is as shown in Figure 3.17.

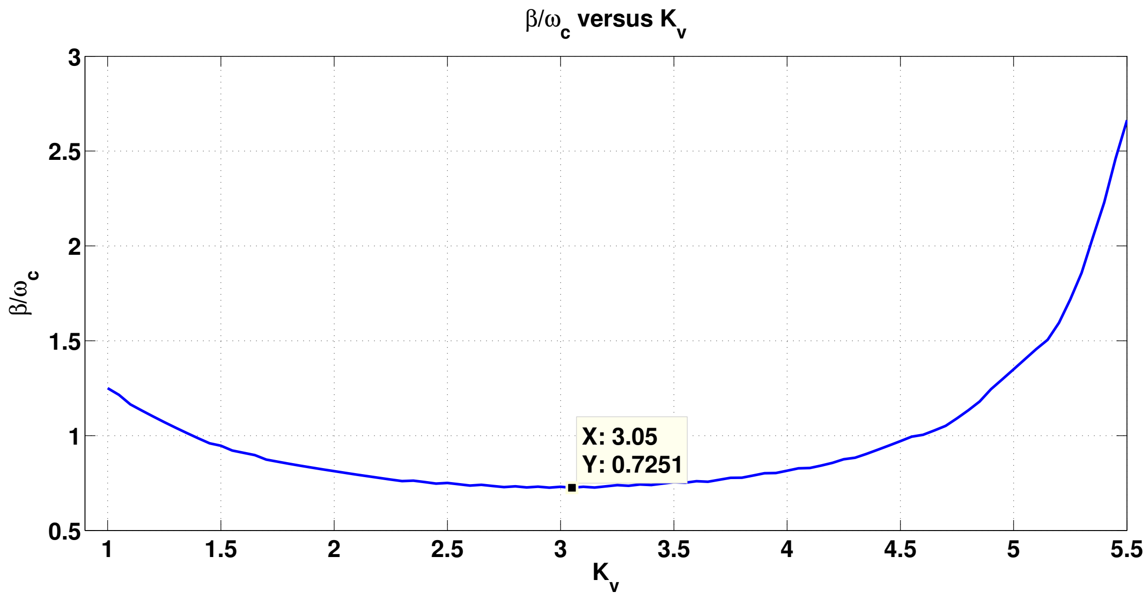

Figure 3.17: The cost $\dfrac{\beta}{\omega_c}$ versus corresponding K_v value graph

According to the above graph, the minimum of $\dfrac{\beta}{w_c}$ is observed when $K_v = 3.05$. Hence this K_v value is chosen as the optimal value and it is named as K_{vopt}.

Sensitivity function for $K_{vopt} = 3.05$ is as shown in 3.18.

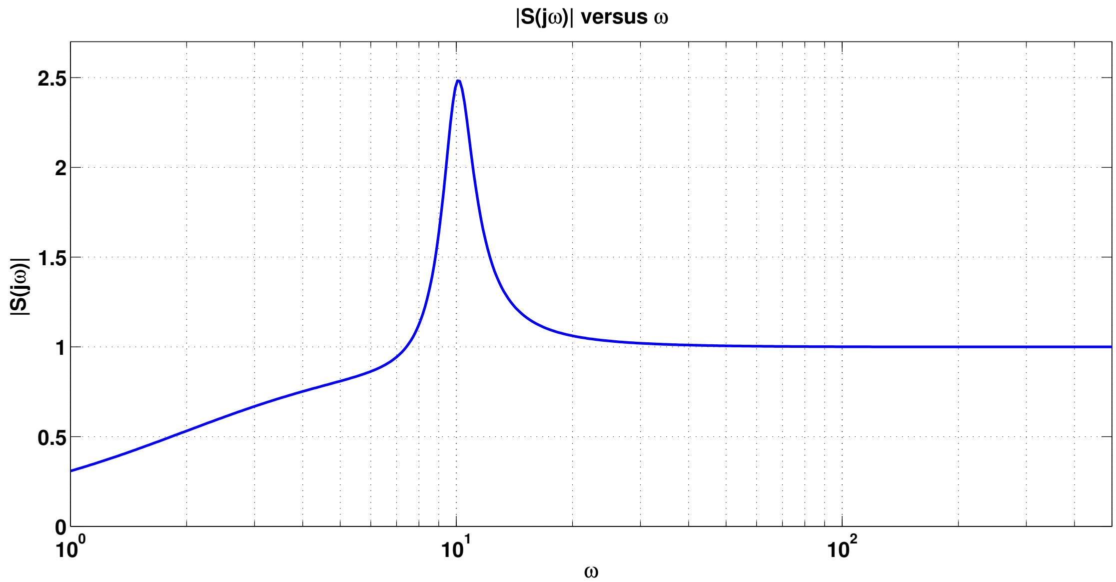

Figure 3.18: $|S(j\omega)|$ versus ω value

As a conclusion, the inner controller is designed as:

$$C_{inner}(s) = \frac{1.9 \times 10^{-9}s + 3 \times 10^{-8}}{s}, \tag{3.25}$$

and the outer controller is designed as:

$$C_{outer} = K_{vopt} = 3.05. \tag{3.26}$$

The resulting $\beta \cong 2.5$ and $\omega_c \cong 5$ rad/sec.

Simulations and performance results corresponding to this design can be found in Chapter 4.

Chapter 4

Simulations

In this Chapter, step response of the closed loop system consisting HDD servo system as the plant and six different controllers designed in Chapter 3 are obtained and time domain performances are compared.

4.1 Method 1: Design of Delay Margin Maximizing PI Controller

Step response of the feedback system for the first controller is shown in Figure 4.1.

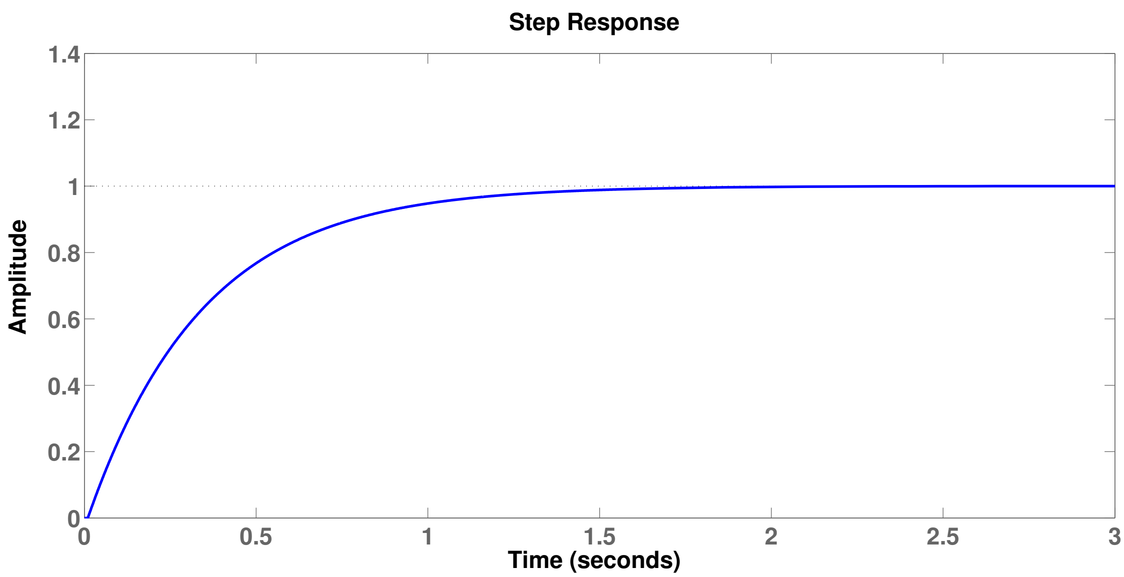

Figure 4.1: Step response of the system consisting $P_0(s)$ and $C(s) = \frac{10^{-9}s + 10^{-12}}{s}$

In this figure controller is delay margin maximizing PI controller. As can be seen there is no overshoot and settling time is nearly 1.5 sec. The steady state error is zero since the output converges to 1. This response is as expected, since controller gains are very small. How these parameters affect the stability is explained detailed in Section 3.1. This case is the best one among all, if the concern is minimizing the percent overshoot; however, settling time is large compared to other designs.

4.2 Method 2: Design of Least Fragile P and PI Controllers

4.2.1 Least Fragile P Controller

Step response of the feedback system with the least fragile P controller is shown in Figure 4.2.

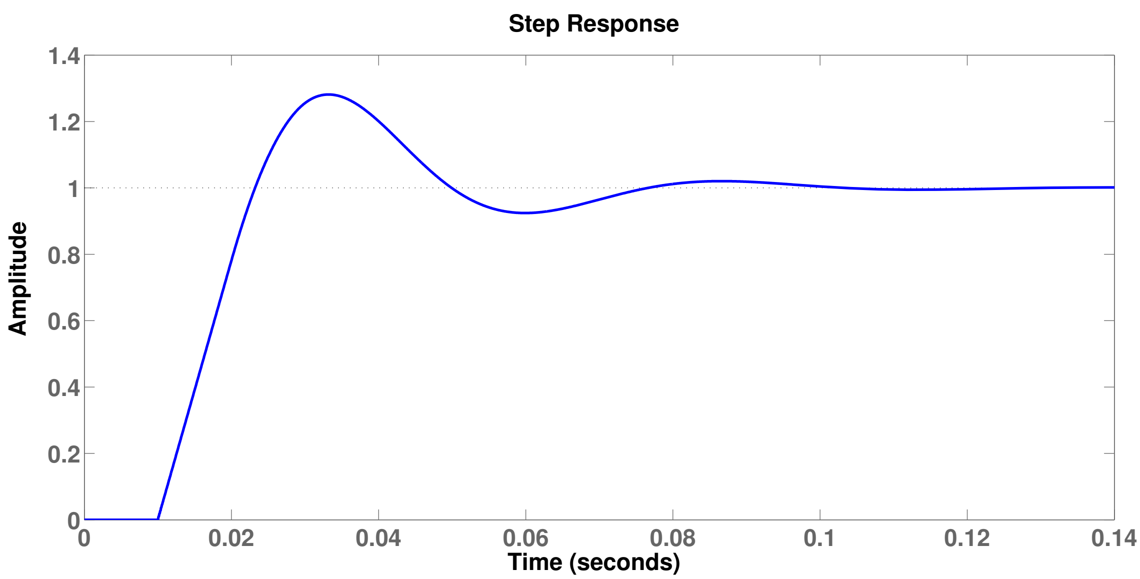

Figure 4.2: Step response of the system consisting $P_0(s)$ and $C(s) = 2.7 \times 10^{-8}$

Reminding that percent overshoot is calculated by:

$$P.O = \left(\frac{Y_{peak} - Y_{ss}}{Y_{ss}}\right) \times 100 \tag{4.1}$$

Therefore for this case $Y_{peak} = 1.3$ and $Y_{ss} = 1$ and percent overshoot becomes 30%. Also settling time is approximately 0.1. Steady state error is zero since the response converges to 1. It turns out that this controller provides the fastest response, i.e. smallest settling time; however the overshoot is very large.

4.2.2 Least Fragile PI Controller

Step response of the feedback system corresponding to the least fragile PI controller is shown in Figure 4.3.

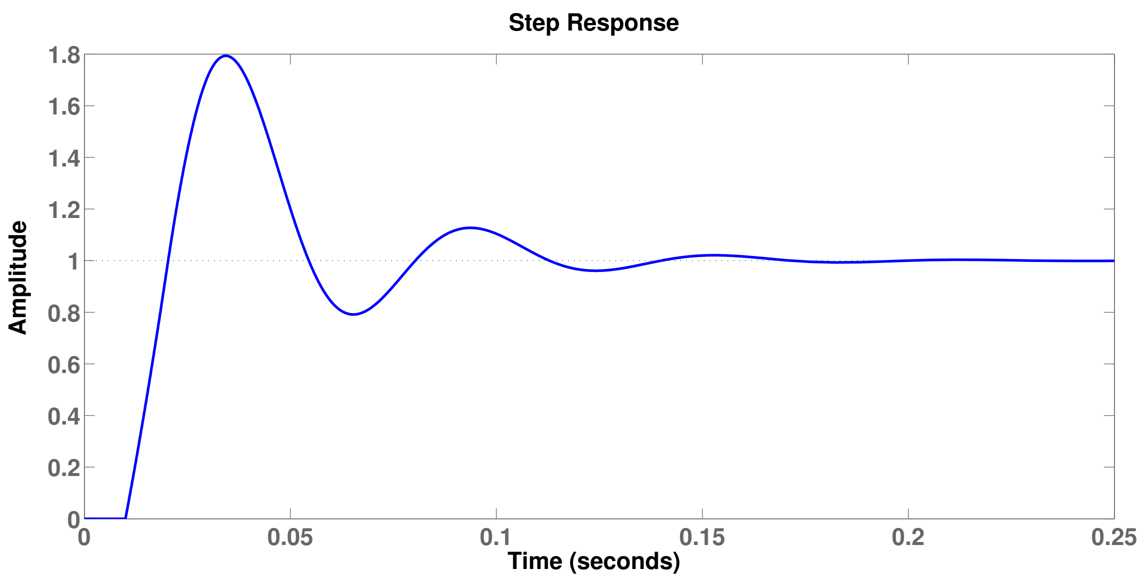

Figure 4.3: Step response of the system consisting $P_0(s)$ and $C(s) = \dfrac{2.94 \times 10^{-8}s + 7.6 \times 10^{-7}}{s}$

Percent overshoot is calculated similar to Section 4.2.1 and found as 80% and settling time is nearly 0.15. Since the controller is a PI controller, overshoot amount is relatively large and this result is expected. The steady state error is zero since it converges to 1.

Considering two least fragile controllers it can be concluded that P controller is better than PI controller, when step responses are taken into account. Since we are concerned with velocity control in this design, integral action is not necessary to get zero steady state error. This will be needed later in position control loops.

4.3 Method 3: Design of Least Fragile Integral Action PI Controller

Step response of the feedback system corresponding to least fragile integral action PI controller is shown in Figure 4.4.

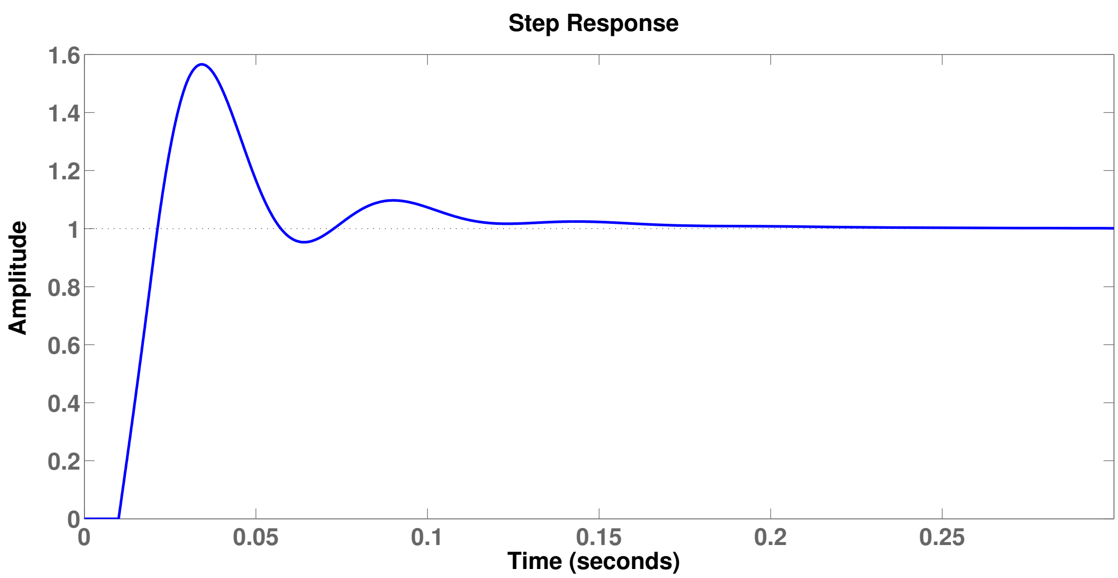

Figure 4.4: Step response of the system consisting $P_0(s)$ and $C(s) = \dfrac{2.83 \times 10^{-8}s + 4.16 \times 10^{-7}}{s}$

In this case the steady state error is zero since the response converges to 1. Percent overshoot is computed as 60% and settling time is nearly 0.12 sec. If the two designed least fragile PI controllers are compared, performance values computed in this section are better than the values obtained in Section 4.2.2. Therefore it can be concluded that Method 3 works better than Method 2.2.

4.4 Method 4: Design of P and PI Controller Considering Robust Performances

4.4.1 Optimal P Controller

Step response of the position control loop shown in Figure 3.9 is given in Figure 4.5.

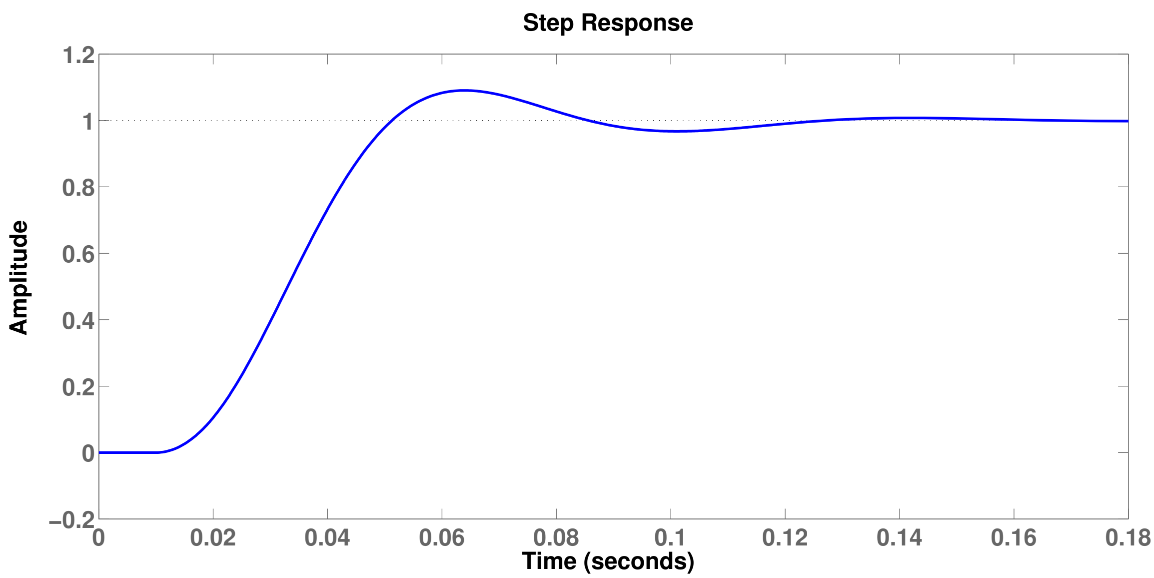

Figure 4.5: Step response of the system consisting $P_0(s)$, $C_{inner} = K_{1opt} = 12.02$ and $C_{outer} = K_{popt} = 31.54$.

According to Figure 4.5, percent overshoot is observed as 15% and settling time is found nearly 0.13 sec. The steady state error is zero since the response converges to 1. Note that there is a good balance between settling time and percent overshoot in this response compared to the other design discussed in the next section.

4.4.2 Optimal PI Controller

Step response of the position control loop shown in Figure 3.15 is illustrated in Figure 4.6.

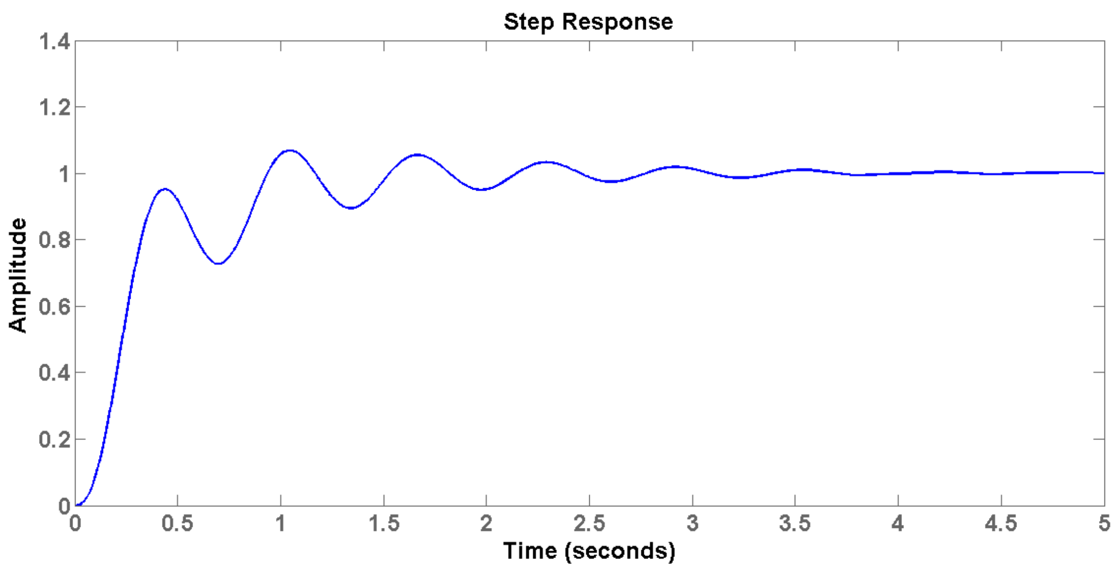

Figure 4.6: Step response of the system consisting $P_0(s)$, $C_{inner}(s) = \dfrac{1.9 \times 10^{-9}s + 3 \times 10^{-8}}{s}$ and $C_{outer} = K_{vopt} = 3.05$.

Similar to Section 4.4.1, this system includes also two cascade controllers. But for this section at first a PI controller is designed, then a P controller is designed. As can be seen from the Figure 4.6, percent overshoot is nearly 8% and settling time is 4 sec. The steady state error is zero since the response converges to 1. This case is the worst controller among all, if the concern is minimizing the settling time. Since there are most oscillations among all cases, settling time is the largest. Note that for the controller of Section 4.4.1 we found $\omega_c \cong 20$ rad/sec and in Section 4.4.2 we found $\omega_c \cong 5$ rad/sec. Therefore, it is expected that the response in Figure 4.6 is slower than the response of Figure 4.5.

For this design RSI (see Section 2.2.3) is satisfied to find the maximum w_c (crossover frequency), therefore we are forcing the performance of the system.

Because of the fact that after the large settling time, the response converges to 1.

Moreover in Section 4.4.1, the cost function $\frac{\beta}{w_c}$ is minimized two times. Therefore, with the optimal P controller designed in Section 4.4.1 we obtain better performance than with the optimal PI controller designed in Section 4.4.2.

Chapter 5

Conclusion

In this thesis, we considered an unstable infinite dimensional plant consisting of a pole at $s = 0$ (integral action) and a time delay. Therefore, observer-state feedback controller design approach cannot work for this plant. The implementation of Smith predictor and IMC methods are complicated compared to the P and PI controllers considered in this thesis from various design perspectives.

Firstly, delay margin maximizing PI controller is designed. PI controller is put in the form $C(s) = \frac{K_p}{s}(s + \alpha)$. By calculating the delay margin values in the determined K_p and α intervals, maximum delay margin and corresponding parameters are found, thus the PI controller is designed. But these parameters are so close to zero, which means the ideal controller according to RSI (Section 2.2.3). Since it is so close to be ideal controller in terms of stability robustness, overshoot is low; but settling time is large (poor performance).

The second method includes designing least fragile P and PI controller to the same plant by checking the stability condition with MATLAB's *allmargin* command. It is observed that PI controller has a greater overshoot than P controller. But settling times are approximately equal.

Thirdly, a PI controller is designed by following the procedure in [2]. If this PI controller is compared with the PI controller designed in Section 3.2.2, performance values are better in this PI controller.

For velocity control, it is concluded that if the concern is minimizing the percent overshoot, Method 1 performs best. However; if the aim is minimizing settling time, Method 2.1 is the best among others.

Finally, the last method differs from the three methods since it concerns about the robust stability conditions for the position control, after the velocity control loop is closed. There are two systems which have two closed loop systems, therefore totally four controllers are designed.

For position control, it is concluded that Method 4.1 is better than Method 4.2 in terms of the balance between settling time and percent overshoot.

To sum up, since the plant is unstable and time delayed, designing a controller is not easy. All six different designed controllers are optimal for different objectives as far as stability robustness properties are concerned. Their time domain performances are also investigated.

As a future work, in addition to this research and H_∞ controller design to position controlled type plant, another H_∞ controller can be designed to the velocity type plant.

Bibliography

[1] P. Yan and H. Özbay, "On the robust controller design for Hard Disk Drive servo systems with time delays", *European Control Conference (ECC)*, 2013.

[2] H. Özbay and A. Gündeş, "Design of first order controllers for unstable infinite dimensional plants", *Low-Complexity Controllers for Time-Delay Systems*, 2014.

[3] H. Özbay,*Introduction to Feedback Control Theory*, CRC Press LLC, Boca Raton FL, 2000.

[4] H. Li and Y.Wang, "A Unified Controller Design using two Different Types of Optimal Reset Control Laws for HDD Servo Systems ", *Int. Conf. Control, Automation, Robotics and Vision*, Singapore, 2010

[5] U. Munz, C. Ebenbauer, T. Haag and F. Allgower, "Stability analysis of time-delay systems with incommensurate delays using positive polynomials", *IEEE Transactions on Automatic Control 54*, 1019-1024, 2009.

[6] O.J. Smith, "A controller to overcome dead time", ISA J. 6 (2), 1959.

[7] X.P. Yang, Q.G. Wang, C.C. Hang and C. Lin, "IMC-based control system design for unstable processes"Ind. Eng. Chem. Res.,41:4288-94,2002

[8] U. Taşdelen and H. Özbay, "On Smith Predictor-Based Controller Design for systems with Integral Action and Time Delay", *Proc. of the 9th Asian Control conference (ASCC), İstanbul, Turkey*, June 2013; DOI: 10.1109/ASCC.2013.6606172.

[9] H.W. Fung, Q.G. Wang and T.H. Lee,“ PI Tuning in Terms of Gain and Phase Margins” *Automatica*, Vol.34, No.9, pp. 11451149, 1998.

[10] V. Venkatashankar and M. Chidambaram, “ Design of P and PI controllers for unstable first-order plus time-delay systems ”, *Int. J. Control*, 60, 137-144, 1994.

[11] P. K. Padhy and S. Majhi, “Relay based PI-PD design for stable and unstable FOPDT processes”, *Computers and Chemical Engineering*, vol. 30, pp. 790796, 2006.

[12] S. Skogestad, “Simple analytic rules for model reduction and PID controller tuning, modeling, identification and control”, 25, 291-309, 2004.

[13] T. Atsumi and W.C. Messmer, “ Mixed sensitivty problem in sampled-data positioning control systems”, *ASME Journal of Dynamic Systems, Measurement and Control*, 5(6), pp.1294-1309, 2011

[14] X. Huang and R. Horowitz, “ Robust controller design of a dual-stage disk drive servo system with an instrumented suspension”, *IEEE Transactions on Magnetics*, 41(8), pp.27-34, 2005.

[15] D. Abramovitch and G. Franklin, “ A brief history of disk drive control”, *IEEE Control Systems Magazine*, 22(3), pp.28-42, 2002.

[16] H. Özbay, “Computation of H_∞ controllers for infinite dimensional plants using linear algebra”, *Numerical Linear algebra with Applications*, vol.20, pp.327-335, 2014.

Appendix A

Code

```
%%To calculate the delay margin for stable Kp-Ki parameters
%%
clear all
clc
K0=5.2269*10^8;

epsz0= 0.99;
epsp0= 0.018;
wnz0= 1.244*10^5;
wnp0= 5.29*10^4;
eps= 0.01;
T_s=tf([1 2*epsz0*wnz0 wnz0^2],[1 2*epsp0*wnp0 wnp0^2 0 0]);
P0=minreal(T_s);

int_k = 1e-9;
int_a = 1e-1;
kmin = 0 + int_k;
kmax = 8.42e-7;
```

```
maxA = -1;
maxK = -1;
maxDM = -Inf;

aa=1e-3:int_a:1;
kkp = kmin:int_k:kmax;
vals = zeros(length(aa), length(kkp));
idx_aa = 1;
idx_kkp = 1;

for a=aa
    idx_kkp = 1;
    for kp=kkp
        G0=minreal(series(tf([1 a],1), kp * K0 *P0));
        x = allmargin(G0);
        q = -Inf;
        if ( x.Stable == 1)
            x = x.DelayMargin;
            for i=1:length(x)
               if ( x(i) > q)
                   q = x(i);
               end
               if ( x(i) > maxDM)
                  maxDM = x(i);
                  maxA = a;
                  maxK = kp;
                  fprintf(1, 'dm: %.11f a:%.4f k:%.11f\n', maxDM, maxA, maxK);
               end
            end
        end
        vals(idx_aa, idx_kkp) = q;
        idx_kkp = idx_kkp + 1;
    end
```

```
    idx_aa = idx_aa + 1;
end

%% To plot DM versus KP
%%
idx = 1;
vals2 = [];
for k=kkp
    vals2 = [vals2 max(vals(:,idx))];
    idx = idx + 1;
end
semilogx(kkp, vals2);
%%To find stable Kp values
%%
h=0.01;
P=0:1e-9:1e-6;
n=1;
K0      = 5.2269*10^8;
epsz0   = 0.99;
epsp0   = 0.018;
wnz0    = 1.244*10^5;
wnp0    = 5.29*10^4;
T_s     = tf([1 2*epsz0*wnz0 wnz0^2], [1 2*epsp0*wnp0 wnp0^2 0]);
P0      = minreal(K0*T_s);

    for j=1:length(P)
        s=tf([1 0],1);
        OL=P(j)*P0*exp(-h*s);
        x = allmargin(OL);

        if ( x.Stable == 1)
             a(n,1) = P(j);
                 n=n+1;
```

```
            end
     end

%%To find stable Kp-Ki values
%%

h=0.01;
P=0:1e-10:5.4e-8;
n=1;
K0      = 5.2269*10^8;
epsz0   = 0.99;
epsp0   = 0.018;
wnz0    = 1.244*10^5;
wnp0    = 5.29*10^4;
T_s     = tf([1 2*epsz0*wnz0 wnz0^2], [1 2*epsp0*wnp0 wnp0^2 0]);
P0      = minreal(K0*T_s);
I=0:1e-8:5.4e-6;
n=1;
for i=1:length(I)
    for j=1:length(P)
        s=tf([1 0],1);
        OL=tf([P(j) I(i)],[1 0])*P0*exp(-h*s);

        x = allmargin(OL);

        if ( x.Stable == 1)
           a(n,1) = I(i);
           a(n,2) = P(j);
                 n=n+1;

        end
     end
 end
```

```
plot(a(:,2),a(:,1),'*')
%%Applying Method 3
%%

h=0.01;
Kp=linspace(1e-9,5.4e-8,100);
b=linspace(1,100,1000);
om=logspace(1,5,3000);

values = zeros(length(b), length(Kp));

phi_norm = [];
for idx2=1:length(b)
    for idx1=1:length(Kp)
        for idx3=1:length(om)
            s=1i*om(idx3);
            y=(5.227e08*s^2 + 1.287e14*s + 8.089e18)*exp(-h*s);
            z=(s^3 + 1904*s^2 + 2.798e09*s);
            H1 = y/(z+Kp(idx1)*y);
            H10=1/Kp(idx1);
            Phi(idx3)  =(H1/H10 -1)*b(idx2)/(s+b(idx2));
        end
        values(idx2, idx1) = max(abs(Phi));
    end

end
%%To plot Kp versus Ki
%%
aq = [];
newValues = values;
for j=1:size(values,2)
    for i=1:size(values,1)
        if ( values(i,j) >= 1)
```

```
            newValues(i,j) = -Inf;
         end
     end
end
graphB = [];
for j=1:size(newValues,2)
    [C, I] = max(newValues(:,j));
    graphB = [graphB b(I)];
end
area(Kp, (Kp/2).*graphB);
%%Applying Method 4.1 to find K1opt
%%
zn=0.99;
zp=0.018;
wn=1.244e5;
wp=5.29e4;
Ko=5.2269e8;
h=0.01;
nPo=[1,2*wn*zn,wn^2];
dPo=[1,2*wp*zp,wp^2,0];
Po=tf(nPo,dPo);
omg=logspace(0,6,1000);

for kk=1:54
    for k=1:1000
    s=1i*omg(k);
    Pof(k)= (s^2 + 246312*s + 1.548e10)/(s^3 + 1904*s^2 + 2.798e09*s);
    S(k)=1/(1+exp(-h*s)*Pof(k)*Ap(kk));
    end
beta(kk)=max(abs(S));
e1=abs(abs(S)-0.707*ones(1,1000));
[err,nn]=min(e1);
omgc(kk)= omg(nn);
```

```
figure(1)
semilogx(omg,abs(S))
hold on
end
figure(2)
plot(Ap,(beta)./omgc)
%%Applying Method 4.1 to find Kpopt
%%
zn=0.99;
zp=0.018;
wn=1.244e5;
wp=5.29e4;
Ko=5.2269e8;
h=0.01;
nPo=[1,2*wn*zn,wn^2];
dPo=[1,2*wp*zp,wp^2,0];
Po=tf(nPo,dPo);
omg=logspace(0,6,1000);

K1opt=12.0219;
Kp=linspace(1,73,100);
for kk=1:100
    for k=1:1000
    s=1i*omg(k);
    Pof(k)= (s^2 + 246312*s + 1.548e10)/(s^2 + 1904*s + 2.798e09);
    P(k)= (K1opt*exp(-h*s)*Pof(k)/s^2)/(1+K1opt*exp(-h*s)*Pof(k)/s);
    S(k)=1/(1+P(k)*Kp(kk));
    end
beta(kk)=max(abs(S));
e1=abs(abs(S)-0.707*ones(1,1000));
[err,nn]=min(e1);
omgc(kk)= omg(nn);
figure(1)
```

```
semilogx(omg,abs(S))
hold on
end
figure(2)
plot(Kp,(beta)./(omgc))

%%Applying Method 4.2 to find W2T
%%

clc
om=logspace(-3,3,10000);
h=0.01;
m=1;

K0      = 5.2269*10^8;
epsz0   = 0.99;
epsp0   = 0.018;
wnz0    = 1.244*10^5;
wnp0    = 5.29*10^4;
T_s     = tf([1 2*epsz0*wnz0 wnz0^2], [1 2*epsp0*wnp0 wnp0^2 0]);
P0      = minreal(K0*T_s);


 for n=1:length(a)
for j=1:length(om)
    s=1i*om(j);
    y=(5.227e08*s^2 + 1.287e14*s + 8.089e18)*exp(-h*s);
    z=(s^3 + 1904*s^2 + 2.798e09*s);
    P=(exp(-h*s)*(y/z));
    C=a(n,2)+a(n,1)/s;
    S(j)=1/(1+P*C);
    w2(j)=0.3125+9.4211e-6*s;
    T(j)=1-S(j);
```

```
    A(j)=abs(T(j)*w2(j));
end

    if (max(A)<1)
        b(m,1)= a(n,1);
        b(m,2)=a(n,2);
        m=m+1;
    end
 end
%%Applying Method 4.2 to find Kpopt and Kiopt values
%%

om=logspace(-3,3,10000);
h=0.01;
m=1;
l=1;
K0      = 5.2269*10^8;
epsz0   = 0.99;
epsp0   = 0.018;
wnz0    = 1.244*10^5;
wnp0    = 5.29*10^4;
T_s     = tf([1 2*epsz0*wnz0 wnz0^2], [1 2*epsp0*wnp0 wnp0^2 0]);
P0      = minreal(K0*T_s);

 for n=1:length(b)
for j=1:length(om)
    s=1i*om(j);
    y=(5.227e08*s^2 + 1.287e14*s + 8.089e18)*exp(-h*s);
    z=(s^3 + 1904*s^2 + 2.798e09*s);
    P=(exp(-h*s)*(y/z));
    C=b(n,2)+b(n,1)/s;
    S(j)=abs(1/(1+P*C));
end
```

```
f=find(abs(S-1)<0.002);
wc(n)=om(min(f));
 end

%%Applying Method 4.2 to find Kvopt
%%
zn=0.99;
zp=0.018;
wn=1.244e5;
wp=5.29e4;
K0=5.2269e8;
h=0.01;
nPo=[1,2*wn*zn,wn^2];
dPo=[1,2*wp*zp,wp^2,0];
Po=tf(nPo,dPo);
omg=logspace(0,6,1000);

Kpopt=1.9e-9;
Kiopt=3e-8;
Kp=linspace(1,5.95,100);
for kk=1:100
    for k=1:1000
    s=1i*omg(k);
    Pof(k)= (s^2 + 246312*s + 1.548e10)/(s^2 + 1904*s + 2.798e09);
    P(k)= ((K0/s^2)*exp(-h*s)*Pof(k)*(Kpopt+Kiopt/s))/(1+(K0/s)*exp(-h*s)*Pof(k)*
    S(k)=1/(1+P(k)*Kp(kk));
    end
beta(kk)=max(abs(S));
e1=abs(abs(S)-0.707*ones(1,1000));
[err,nn]=min(e1);
omgc(kk)= omg(nn);
figure(1)
semilogx(omg,abs(S))
```

```
hold on
end
figure(2)
plot(Kp,(beta)./(omgc))
```