

DESIGN OF FIRST ORDER CONTROLLERS FOR A FLEXIBLE ROBOT ARM 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
Gökçe Kuralay
May 2016

Design of First Order Controllers for a Flexible Robot Arm with Time
Delay

By Gökçe Kuralay

May 2016

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.

Hitay Özbay(Advisor)

Ömer Morgül

Mehmet Önder Efe

Approved for the Graduate School of Engineering and Science:

Levent Onural
Director of the Graduate School

ABSTRACT

DESIGN OF FIRST ORDER CONTROLLERS FOR A FLEXIBLE ROBOT ARM WITH TIME DELAY

Gökçe Kuralay

M.S. in Electrical and Electronics Engineering

Advisor: Hitay Özbay

May 2016

In an earlier work, Gündes et al. (2007), stabilizing PID controllers for a class of unstable plants with time delays (I/O delays) are obtained. By utilizing this approach and methods given in Özbay and Gündes (2007, 2014) we aim to investigate appropriate PI, PD and PID controllers by finding the maximum allowable boundaries for each controller parameter. A model of a flexible robot arm which includes a time delay and an integrator is considered as an application example. It is aimed to find the optimal coefficients for the derivative and integral gains such that the designs achieve a set of performance and robustness objectives. Stability and robustness properties of the closed-loop system are also investigated. Specifically, for PD controller design, optimal derivative action gain is determined under various performance objectives. For PI controller design, optimal P (proportional) and D (derivative) gains are determined to achieve the least fragile integral action gain. Moreover, system performance is compared with other PID designs considering different types of control objectives.

Keywords: PID Controller, PD Controller, Flexible Modes, Time Delay Systems, Vector Margin, Small Gain Theorem, H_∞ Norm.

ÖZET

ZAMAN GECİKMELİ ESNEK ROBOT KOLU İÇİN BİRİNCİ DERECEDEN KONTROLÖR TASARIMI

Gökçe Kuralay

Elektrik ve Elektronik Mühendisliği, Yüksek Lisans

Tez Danışmanı: Hitay Özbay

Mayıs 2016

Girdi/çıktı ilişkisinde zaman gecikmesi bulunan bir grup kararsız tesis için kapalı döngü kararlılığı sağlayan PID kontrolörleri Gündes ve diğerleri (2007) tarafından elde edilmiştir. Özbay ve Gündes (2007, 2014)'deki yaklaşım ve metotları kullanarak her parametre için kabul edilebilir maksimum aralığa sahip PI, PD ve PID kontrolörlerin tasarımı hedeflenmiştir. Zaman gecikmesi ve bir integratöre sahip esnek bir robot kolu modeli uygulama örneği olarak ele alınmıştır. Türevsel ve integral etkiler için tasarımların performans ve gürbüzlük hedeflerini sağlayacak optimal katsayıların bulunması hedeflenmiştir. Kapalı döngü sisteminin kararlılık ve gürbüzlük özellikleri de incelenmiştir. Özellikle, PD kontrolör için uygun kontrolör etki aralığını maksimuma çıkaran optimal türevsel etki hesaplanmıştır. PI kontrolör tasarımında ise tolere edilebilecek en büyük integral katsayısının alt sınırını maksimum yapacak oransal etki değeri bulunmuştur.

Anahtar sözcükler: PID Kontrolör, PD Kontrolör, Esnek Modlar, Zaman Gecikmeli Sistemler, Vektör Marjı, Küçük Kazanç Teoremi, H_∞ Optimal Kontrol.

Acknowledgement

First and foremost, I would like to acknowledge and thank my thesis supervisor Prof. Hitay Özbay. If he hadn't had the patience and belief for me, I would never have the courage and motivation to go on when I felt lost and weak. Moreover, I would like to acknowledge my parents who endured for the last 2.5 years of my life with their support. Throughout my M. Sc. process, I have faced psychological and physical drawbacks which slowed down my progress immensely. I have been through phases which I have never experienced in my life before. However, with the help and support of my beloved ones, my efforts could bear fruit. The thesis written hereby does not only symbolize my analysis and results on the aforementioned topic but also my perseverance to overcoming the darkest time of my life. Without a doubt our angel, department secretary, Mürüvvet Parlakay tried to motivate me whenever I felt sad and overwhelmed. Her motherhood and tolerance was no different than my family. She was one of few people I could visit without shame and guilt. With patience she endured all my instabilities and tried to guide for good. Moreover, I would like to thank my colleagues Dilan Öztürk and Elvan Kuzucu who helped me to organize and edit mistakes with layout, grammar and content. Their support is irreplaceable. Among my friends, I can only think of two names who were next to me during all my struggles. Through these years, I have faced betrayal and dishonesty from the ones I trusted but there were 2 people who never ever turned their back to me. My dear friend Mehmet Bülbüldere and Ece Aydemir were my indefatigable supporters. Since I could share my fears and concerns with them, I have to thank them for listening to me without any complaints. Dr. Defne Dursunkaya and Gözde Mumcu are silent heroines in this period as well. By listening their psychological guidance, I could untie negative things I carried with me and could approach life with a different point of view. Finally, I am honored to have Prof. Dr. Ömer Morgül and Prof. Dr. M. Önder Efe in my thesis committee. I am thankful since they read and gave feedback about my thesis.

Contents

1	Introduction	1
2	Problem Definition	3
2.1	A Sufficient Condition for Feedback System Stability	5
2.2	Scope of the Controller Design	6
3	Description of the Plant and Controller Designs	7
3.1	Flexible Robot Arm with Time Delay	7
3.2	PD Controller Design	9
3.3	PI Controller Design	12
4	Performance Analysis	17
4.1	Comparison of Selected Controller Parameters	17
4.2	Unit Step Responses for the Sample Points Given	20
4.3	Performance analysis with Smith-Predictor Based Controller Design	22

4.4	Designing a controller for Smith Predictor Controller Comparison	24
4.4.1	Design for $C_{PI}(s) = 0.0556 + \frac{0.0752}{s}$	28
4.4.2	Design for $C_{PI}(s) = 0.0626 + \frac{0.0902}{s}$	30
5	Conclusion	35
A	MATLAB codes to generate controllers and evaluation of cost functions	39

List of Figures

2.1	Representation of the feedback system	4
3.1	Representation of the aforementioned flexible robot arm	8
3.2	Bode Plot Analysis for Flexible Robot Arm Plant	9
3.3	Graph of $\ \psi_a\ _\infty^{-1}$ and a	10
3.4	PD design for $K_d = 0$	11
3.5	Relation between K_p and K_i parameters	14
3.6	Vector Margin for K_p and K_i parameters	16
3.7	Vector Margin and Contour Lines for K_p and K_i parameters . . .	16
4.1	Unit Step Response for $C_{PI} = 0.0556 + \frac{0.3126}{s}$	21
4.2	Unit Step Response for $C_{PI} = 0.0626 + \frac{0.3999}{s}$	22
4.3	Unit Step Response for $C_{PI} = 0.0556 + \frac{0.0752}{s}$	23
4.4	Unit Step Response for $C_{PI} = 0.0626 + \frac{0.0902}{s}$	24
4.5	Max γ wrt α_1 and α_2 for $C_{PI} = 0.0556 + \frac{0.0752}{s}$	29

4.6	X-Y Plane for Figure 4.5	30
4.7	VM values for α_1 , α_2 and γ_{opt} for Selection 1	31
4.8	X-Y Plane for Figure 4.7	31
4.9	X-Z Plane for Figure 4.7	32
4.10	γ values wrt α_1 and α_2 for $C_{PI} = 0.0554 + \frac{0.0902}{s}$	32
4.11	X-Y Plane for Figure 4.10	33
4.12	γ values wrt α_1 and α_2 for $C_{PI} = 0.0554 + \frac{0.0902}{s}$	33
4.13	X-Y Plane for Figure 4.12	34
4.14	X-Z Plane for Figure 4.12	34

List of Tables

4.1	Performance for K_p and K_i parameters	18
4.2	Performance for K_p and K_i parameters with cost function	19

Chapter 1

Introduction

This thesis deals with the design of first order controllers for infinite dimensional high order plant with flexible modes whose transfer function contains single unstable pole. In this context, PI and PD controllers are investigated. The method consists of the usage of small gain theorem for the characteristic equation of the feedback system. For this purpose, algebraic manipulations used in [4] play a crucial role. The range of allowable controller gain parameter is estimated by using H_∞ norm of an infinite dimensional transfer function. This method is based on [4], [11] and [12]. The aim of the present thesis is to illustrate this method by applying it to a model of flexible robot arm with time delay.

It should also be noted that when there are only small number of free parameters in the controller, it is rather easy to utilize stability checks by brute force-numerical search methods. However, if the plant is unstable and infinite dimensional, this is not preferred. Considering time delay systems, there are several methods for finding low order controllers, see [3], [9] and [13].

The rest of the thesis is organized as follows. Chapter 2 introduces problem definition and sufficient conditions to satisfy feedback system stability. Moreover, controller design technique is defined. In Chapter 3 description of the flexible robot arm with time delay and controller designs for the mentioned plant

are investigated. Chapter 4 discusses the performance of the selected controller parameters when applied for the plant introduced in Chapter 3. In Chapter 5 conclusion for the overall thesis is given and possible future works are discussed.

In [6] fragility analysis of PI-controllers for single-input-single-output (SISO) systems subject to input or output delays are discussed by using a geometric approach.

PID controllers for time delay systems are investigated in [13], [9], [7] and [5]. Since time-delays are important components in dynamical systems during propagation, or transport phenomena, in [7] stabilization of dynamical systems subjected to time delay are investigated. By utilizing eigenvalue-based approach analytical methods and computational algorithms are introduced. In [9], stabilizing a SISO linear time-invariant (LTI) plant with known time delay using a low-order controller is investigated. By regarding sufficient conditions for stability, sets of P, PI and PID controllers are designed and the resultant controller design are compared to Youla parameterization of all stabilizing controllers for plants without time delay. Moreover, in [5] entire set of stabilizing PID controller parameters are computed for an arbitrary LTI system. K_p and K_d parametrization is handled such that for each fixed K_p parameter an algorithm is proposed to find stabilizing K_i and K_d parameters by utilizing convex polygons.

In addition, for flexible robots with actuator/sensor delay a Smith- Predictor based controller is designed in [1]. In [1], a new Smith predictor based controller is proposed for systems with integral action and flexible modes under input-output time-delay bu using controller parametrization to achieve specific robustness and stability objectives.

In this thesis we use the sufficient conditions derived in [4] for a model adopted from [2], [1]. By getting aid from system identification approach utilized in [2] and using a plant analogous to the one used in [1], satisfying sufficient conditions to obtain stability are aimed.

Chapter 2

Problem Definition

Finite dimensional linear time invariant (LTI) systems are sufficiently accurate models for a wide range of dynamical phenomena, however in some cases input/output delay elements cannot be ignored and should be included to make the system model more realistic. This addition makes us deal with an infinite-dimensional system. It is clear that, even for delay-free systems, not all unstable plants are stabilizable by a PID controller. Besides, right half plane pole and the time delay in the output or in the input channels of the plant impose additional restrictions on the feedback controllers. As it is mentioned above, plant discussed here has transfer functions in the form:

$$P(s) = \frac{1}{s - p} G(s) \quad (2.1)$$

where $p \geq 0$ is the unstable pole and $G(s) \in H_\infty$ is the stable part of the plant. Note that $G(s)$ can be irrational i.e. infinite dimensional (as in our case). As seen from the factorization in (2.1) the plant is strictly proper. For the flexible robot arm application considered in this paper p parameter is equal to zero i.e. the plant contains an integral action, as in many mechanical systems modeled by the Newton's law. According to [4] controllers to be designed have the following common structures:

$$C(s) = K_p + \frac{K_d s}{\tau s + 1} + \frac{K_i}{s} \quad (2.2)$$

where

$$K_p, K_i, K_d \in \mathbb{R}, \quad \tau \geq 0$$

Note that PD, PI controllers are special cases of (2.2) :

$$C_{pd}(s) = K_p (1 + \tilde{K}_d s), \quad \tilde{K}_d = \frac{K_d}{K_p}, \quad (2.3)$$

$$C_{pi}(s) = K_p (1 + \frac{\tilde{K}_i}{s}), \quad \tilde{K}_i = \frac{K_i}{K_p} \quad (2.4)$$

Definition 1. *The feedback system formed by the controller C and plant P as shown in Figure 2.1 is stable if $S := (1 + PC)^{-1}$, CS and PS are stable, i.e., they are transfer functions in H_∞ . In this case, then the controller C is said to stabilize the plant P . The set of all controllers stabilizing a given plant P is denoted by $\mathcal{C}(P)$.*

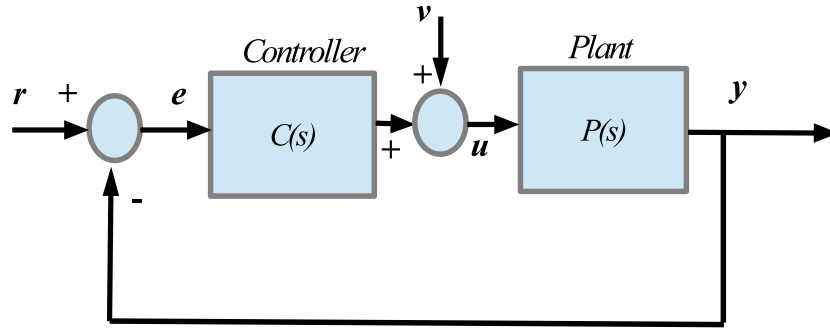


Figure 2.1: Representation of the feedback system

2.1 A Sufficient Condition for Feedback System Stability

Recall the definition of feedback system stability from Definition 1. It is assumed that the plant $P(s)$ does not have a zero at $s = 0$. Then the feedback system stability is equivalent to the stability of the sensitivity function:

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

According to [12] a controller $C(s)$ with a special structure (2.2) stabilize a plant in the form (2.1) if and only if there exists a constant $a \geq 0$ such that U_a is unimodular (i.e. $U_a, U_a^{-1} \in H_\infty$):

$$U_a = \frac{s - p}{s + a} + \frac{K_p}{s + a} G(s)C_0(s)$$

where $C_0(s) = (1 + \tilde{K}_d s)$ as in (2.3). Thus, we can define

$$K_p = (p + a)G(0)^{-1} \tag{2.5}$$

$$G_0(s) = G(s)G(0)^{-1} \tag{2.6}$$

then

$$U_a(s) = 1 + (p + a) \frac{s}{s + a} \psi_0(s)$$

$$\psi_0(s) = \frac{1}{s} (G_0(s)C_0(s) - 1)$$

Hence according to this parametrization and utilizing the fact that, $\left\| \frac{s}{s + a} \right\|_\infty \leq 1$, and the small gain theorem, U_a is unimodular if

$$(p + a) < \|\psi_0\|_\infty^{-1}. \tag{2.7}$$

For the system considered in the next section we have $p = 0$. The condition (2.7) was derived from [4], [11]. If we would like to find a smaller boundary condition for U_a to be unimodular (2.7) can be changed as,

$$(p + a) < \|\psi_a\|_\infty^{-1}, \tag{2.8}$$

where

$$\psi_a(s) = \frac{1}{s+a} (G_0(s)C_0(s) - 1).$$

If we compare the inequalities written in (2.7) and (2.8), we have:

$$\|\psi_a\|_\infty \leq \|\psi_0\|_\infty \quad \forall a > 0.$$

Hence, an allowable interval for the proportional gain K_p can be written as

$$pG(0)^{-1} < K_p < (p + a_0)G(0)^{-1}$$

where $a_0 > 0$ is the largest a satisfying (2.8). So, overall we focus on a *minmax* problem, minimizing $\|\psi_a\|_\infty$ subject to (2.8) where the norm is maximum of the frequency response magnitude.

2.2 Scope of the Controller Design

In the rest of the thesis we will use the above inequalities to obtain a set of admissible stabilizing controller parameters. In particular, to minimize $\|\psi_a\|_\infty$ we will investigate designs of $C_0(s)$, i.e. $\tilde{K}_d$. Considering the system, by following the specifications and definitions given in [11], a set of PI controllers are aimed to design at first step.

Furthermore, design analysis consists of inequalities that investigate the boundaries for parameters that determine conservativeness of the approach. In this work, auxiliary coefficients are used to span and search how wide our interval can be. Since stability is an important feature for all plants, some performance concerns like vector margin (VM), percent overshoot (%PO), settling time T_s etc. are also investigated. Comparison between values are also done and outcomes are discussed consecutively.

Chapter 3

Description of the Plant and Controller Designs

3.1 Flexible Robot Arm with Time Delay

In this study we focus on a plant which is a flexible robot arm represented in Figure 3.1 where control input is the torque applied by the motor and the angular velocity is taken to be the output. Hence from the laws of physics, the plant includes an integrator. So, $P(s)$ has a pole at $s = 0$. Due to the flexibility of the robot arm, high frequency dynamics also enter the plant transfer function. Moreover, due to sampling and signal transmission depending on the distance between the controller and the plant we encounter time delays.

Once proper approximations are done for the real-life system given above, transfer function from torque to angular velocity can be given as:

$$P(s) = \frac{K}{s} R_0(s) e^{-hs} \quad (3.1)$$

where $K > 0$ is inversely proportional to the inertia and $h > 0$ being the time delay, and $R_0(s)$ is a 12^{th} order minimum phase transfer function representing the flexible modes of the robot arm. The transfer function $R_0(s)$, gain K and

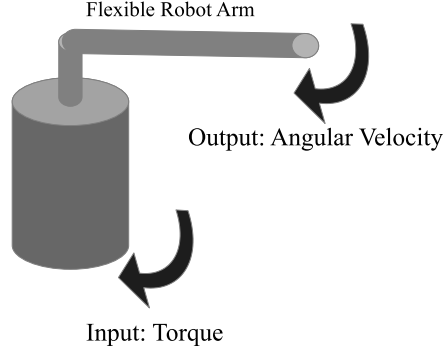


Figure 3.1: Representation of the aforementioned flexible robot arm

delay h are estimated from frequency response identification techniques e.g. [2]. The structure of $R_0(s)$ of the robot arm is:

$$R_0(s) = \frac{\frac{s^2}{\omega_{z1}^2} + \frac{2\zeta_{z1}s}{\omega_{z1}} + 1}{\left(\frac{s^2}{\omega_1^2} + \frac{2\zeta_{p1}s}{\omega_1} + 1\right)^2} \frac{\left(\frac{s^2}{\omega_{z2}^2} + \frac{2\zeta_{z2}s}{\omega_{z2}} + 1\right)^2}{\frac{s^2}{\omega_{z3}^2} + \frac{2\zeta_{z3}s}{\omega_{z3}} + 1} \frac{\frac{s^2}{\omega_{z3}^2} + \frac{2\zeta_{z3}s}{\omega_{z3}} + 1}{\left(\frac{s^2}{\omega_3^2} + \frac{2\zeta_{p3}s}{\omega_3} + 1\right)^3}.$$

Note that $R_0(0) = 1$. Parameters of the plant are listed in table below:

Plant Parameters			
Parameter	Value	Parameter	Value
K	70	h	0.0055
ω_{z1}	12	ω_{p1}	12
ω_{z2}	20	ω_{p2}	20
ω_{z3}	26	ω_{p3}	50
ζ_{z1}	0.13	ζ_{p1}	0.3
ζ_{z2}	0.4	ζ_{p2}	0.2
ζ_{z3}	0.4	ζ_{p3}	0.3

For controller design we find a coprime factorization of the delay-free plant:

$$\frac{K}{s}R_0(s) = X(s)Y(s)^{-1} \quad (3.2)$$

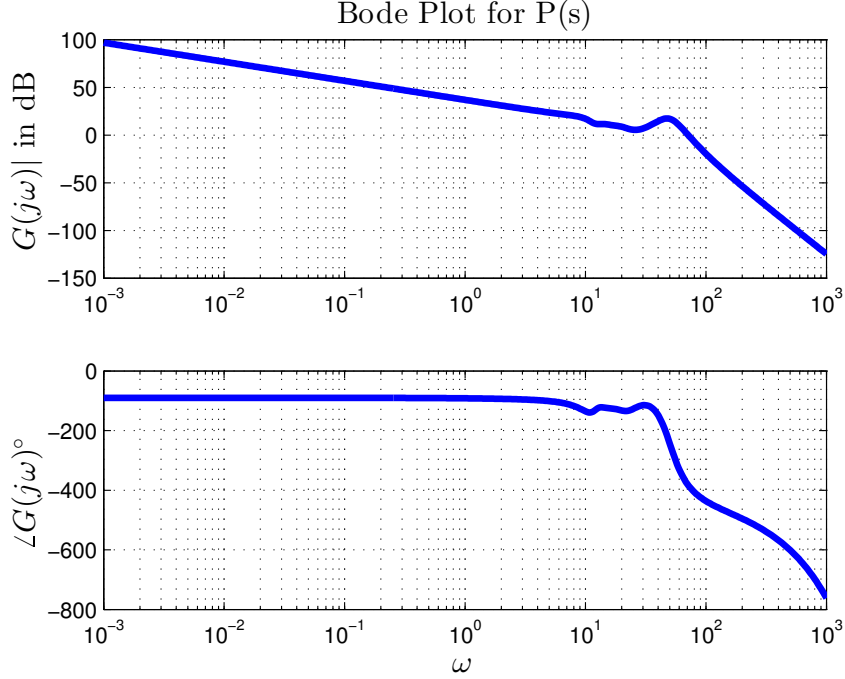


Figure 3.2: Bode Plot Analysis for Flexible Robot Arm Plant

$$X(s) = \frac{KR_0(s)}{s+a} \quad Y(s) = \frac{s}{s+a} \quad (3.3)$$

then,

$$P(s) = e^{-hs} X(s)Y^{-1}(s). \quad (3.4)$$

3.2 PD Controller Design

Considering the feedback system shown in Figure 2.1, for the plant (2.1), a PD controller is in the form $C_{pd} = K_p C_0(s)$ where $K_p = (p+a)G(0)^{-1}$ and $C_0(s) = (1 + \tilde{K}_d s)$. The largest $a > 0$ satisfying (2.8) can be found by minimizing (as also given in [12]):

$$\gamma(Q, a) = \left\| \frac{G_0(s) - 1}{s+a} + Q \frac{s}{s+a} G_0(s) \right\|_\infty \quad \text{over } Q \in \mathbb{R} \quad \text{and } a > 0, \quad (3.5)$$

where $G_0(s) = G(s)G(0)^{-1}$ and $Q = \tilde{K}_d$.

It is seen in this algorithm that the main computation involves solving a *minmax* problem. In order to maximize the gain margin of the system one should try to find the largest a satisfying (3.5). If one would like to find the optimal controllers for this plant one should start from deriving the interval for proportional gain. Let us start by choosing $C_0(s) = 1$ to understand the procedure. By using the equation in (3.5) and knowing that the plant has an unstable pole at $p = 0$, the a_0 value is found as $a_0 = 8.369$; this is determined from Figure 3.3 which shows $\|\psi_a\|_\infty^{-1}$ with respect to a . Thus an allowable interval for K_p can be written as:

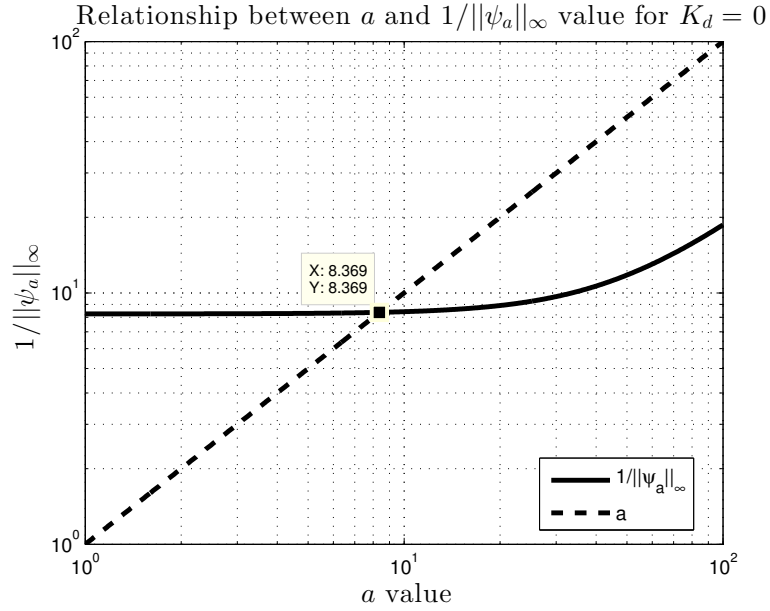


Figure 3.3: Graph of $\|\psi_a\|_\infty^{-1}$ and a

$$0 < K_p < a_0 \frac{1}{K}$$

$$0 < K_p < 0.1196.$$

Now for the PD controller design we consider $C_0(s) = (1 + \tilde{K}_d s)$, where $\tilde{K}_d$ is to be determined. Since $\tilde{K}_d$ is also variable parameter, the proposed algorithm from [12] is:

Step 1 For each fixed value of Q the equation (3.5) will be utilized and the inequality

$$(p + a) < \frac{1}{\gamma(Q, a)}$$

will be investigated such that there exists a $a_{max}(Q)$. Notice that $\forall a < a_{max}(Q)$.

Step 2 Plot Q vs $a_{max}(Q)$ find the max of $a_{max}(Q)$

$$Q_{opt} = \operatorname{argmax} \{a_{max}(Q)\}$$

Step 3 Find the range of K_p and $\tilde{K}_{d,opt} = Q_{opt}$

According to [11], since the unstable pole for our system is $p = 0$, we can choose the *the least fragile* proportional gain as

$$K_{p,LF} = \left(\frac{a_0}{2}\right) G(0)^{-1}.$$

To illustrate the computations involved in the design method described above, we provide Figure 3.4 which shows Q vs. $a_{max}(Q)$:

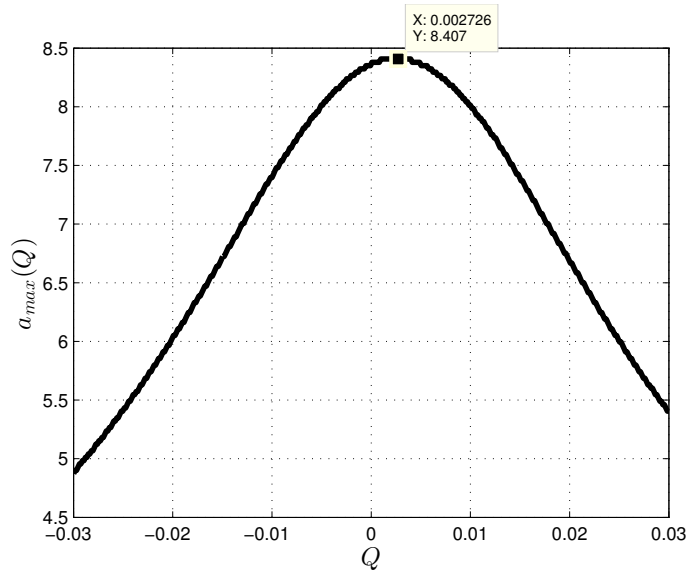


Figure 3.4: PD design for $K_d = 0$

As we see from this figure, for $\tilde{K}_d = 0$, $a_{max}(Q)$ value matches the previous case where a proportional controller is considered. The largest value of $a_{max}(Q)$ is attained at $Q_{opt} = 0.00272 = \tilde{K}_{d,opt}$ and $\max_Q a_{max}(Q) = 8.407$.

3.3 PI Controller Design

Consider the design of a PI controller in the form

$$C_{pi}(s) = C_1(s) + \frac{K_i}{s} \quad (3.6)$$

where $C_1(s) = K_p$ is such that C_1 is in the domain of all controllers that stabilize plant $P(s)$. In other words, the controller C_1 already stabilizes P and the proper integral action is sought to be added for the controller. Similarly one can apply the method given in [4] for $C_1 = C_{pd}$ as well. If that is the case, addition of the integral action will give a PID controller C_{pid} instead of a PI controller. Since C_1 is in the domain of controllers that stabilize P , the following statement holds:

$$H_1(s) = \frac{P(s)}{1 + C_1(s)P(s)} \in H_\infty.$$

Let $C(s) = C_{pi}(s)$. Then the characteristic equation is written in the form:

$$\begin{aligned} 1 + P(s)C_{pi}(s) &= 0 \\ 1 + C_1(s)P(s) + \frac{K_i}{s}P(s) &= 0 \\ (1 + C_1(s)P(s))(1 + \frac{K_i}{s}H_1(s)) &= 0 \end{aligned}$$

In this expression $H_1(s)$ is represented as:

$$\frac{P(s)}{1 + C_1(s)P(s)} \quad (3.7)$$

Below we summarize the design method from [4], [11]. Using the fact that controller C_1 is in the space of controllers that stabilize plant P , it can be concluded that new controller C_{pi} stabilizes plant P if and only if $V_1^{-1} \in H_\infty$ where transfer function for $V_1(s)$ is denoted as:

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

$$C_{pi} \in \mathcal{C}(P) \Leftrightarrow V_1^{-1} \in H_\infty$$

Let us define,

$$K_i = \frac{b}{H_1(0)}.$$

Then $V_1(s)$ can be re-written as:

$$V_1(s) = \left(1 + \frac{K_i}{s} H_1(s) + \frac{b}{s} - \frac{b}{s} \right) \quad (3.9)$$

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

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

$$= \left(1 + \frac{b}{s} \right) + b \left(\frac{H_1(s) H_1(0)^{-1} - 1}{s} \right) \quad (3.12)$$

Thus,

$$V_1(s) = \left(1 + \frac{b}{s} \right) \left(1 + \left(1 + \frac{b}{s} \right)^{-1} b \left(\frac{H_1(s) H_1(0)^{-1} - 1}{s} \right) \right). \quad (3.13)$$

Assume now that $b > 0$, then we have

$$\left(1 + \frac{b}{s} \right)^{-1} = \frac{s}{s+b} \in H_\infty$$

with $\|\frac{s}{s+b}\|_\infty = 1$. From the small gain theorem: $V^{-1} \in H_\infty$, in other words $C_{pi} \in \mathcal{C}(P)$, if b satisfies

$$0 < b < \frac{1}{\|\Phi_0\|_\infty} \quad \text{where} \quad \Phi_0(s) = \left(\frac{H_1(s) H_1(0)^{-1} - 1}{s} \right). \quad (3.14)$$

In fact, a careful examination of (3.13) shows that, rather than (3.14), the following less conservative sufficient condition on b can be used for C_{pi} to be in $\mathcal{C}(P)$:

$$0 < b < \frac{1}{\|\Phi_b\|_\infty} \quad \text{where} \quad \Phi_b(s) = \left(\frac{H_1(s) H_1(0)^{-1} - 1}{s+b} \right) \quad (3.15)$$

Note that Φ_b depends on K_p which is assumed to be in $\mathcal{C}(P)$. Thus the optimal K_p and K_i parameters which gives us the optimal PI controller $C_{pi,opt}(s) = K_{p,opt} + \frac{K_{i,opt}}{s}$ can be designed as follows. For each fixed $K_p \in \mathcal{C}(P)$ the largest allowable

$b > 0$ can be obtained with the same procedure as described for PD Controller. According to [11] and [12] the least fragile K_i parameter is obtained as:

$$K_i = \frac{b_{max}(K_p)}{2} H_1(0)^{-1} \quad (3.16)$$

Note that $H_1(0)^{-1}$ is also dependent on K_p value. Since,

$$H_1(s) = \frac{P(s)}{1 + C_1(s)P(s)}$$

and $C_1(s) = K_p$ with $P(s) = K \frac{e^{-hs} R_0(s)}{s}$, $H_1(s)$ is:

$$H_1(s) = \frac{K e^{-hs} R_0(s)}{s + K_p K e^{-hs} R_0(s)}, \quad H_1(0) = \frac{1}{K_p}$$

Thus, $H_1(0)^{-1} = K_p$. In equation (3.16), note that both b and $H_1(0)^{-1}$ parameters include K_p as a variable. Once an interval for allowable K_p is found, the corresponding allowable integral action gain K_i is computed as shown in Figure 3.5.

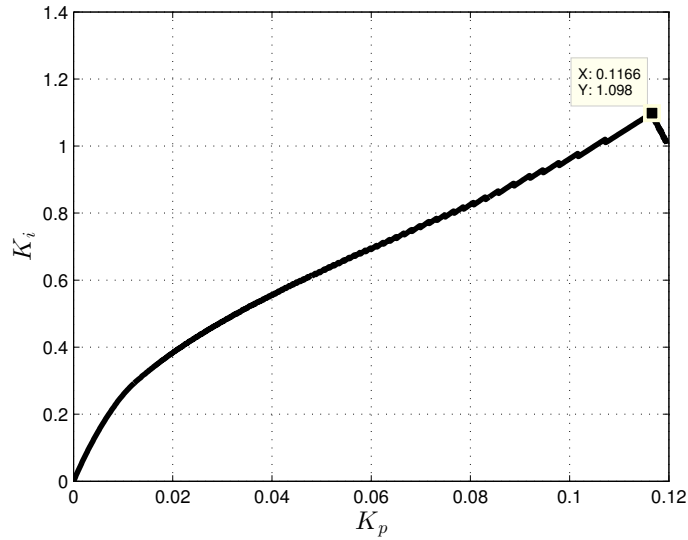


Figure 3.5: Relation between K_p and K_i parameters

According to Figure 3.5, maximum K_i value for $0 < K_p < 0.1196$ is obtained

as 1.098. Thus, the least fragile PI controller proposed in [11] is found as:

$$\begin{aligned} K_{i,opt} &= \frac{1.098}{2} = 0.5490 \\ K_{p,opt} &= 0.1166 \\ C_{PI,opt} &= K_{p,opt} + \left(\frac{K_{i,opt}}{s} \right) \\ C_{PI,opt} &= 0.1166 + \frac{0.5490}{s} \end{aligned}$$

Remark 1. *Note that this controller is designed for the least-fragile integral action gain. However, from other design perspectives this controller may be undesirable. Below we discuss other possible design criteria.*

Definition 2. *The vector margin (VM) or stability margin is the minimum distance from the Nyquist plot $P(j\omega)C(j\omega)$ to the point $(-1, 0)$ in the complex plane. The vector margin combines gain and phase margins into a single measure. In order to find the VM, sensitivity function of the closed loop system can be used:*

$$VM = \|S\|_{\infty}^{-1}.$$

Since intervals for allowable K_p and K_i parameters are found, VM can be investigated by placing allowable controllers to the system given in Figure 2.1. VM value changes with respect to K_p and K_i given as in Figure 3.6.

It is seen from Figure 3.6 that as K_p and K_i values decrease VM value increases. From the shape of the surface, we can interpret that as we reach upper boundaries, VM is vanishing. Thus, if we would like to investigate maximum K_i values for each K_p values and maximum K_p parameters corresponding to allowable K_i values we end up with dashed line and dotted-dashed line, respectively given in Figure 3.6. Boundary for both lines is $VM = 0.3$, i.e. points on both lines satisfy $VM \geq 0.3$. Note that in Figure 3.6, maximum VM values with respect to column(K_p) and row (K_i) intersect.

If one is interested in the contour field in $X - Y$ plane it is represented as in Figure 3.7.

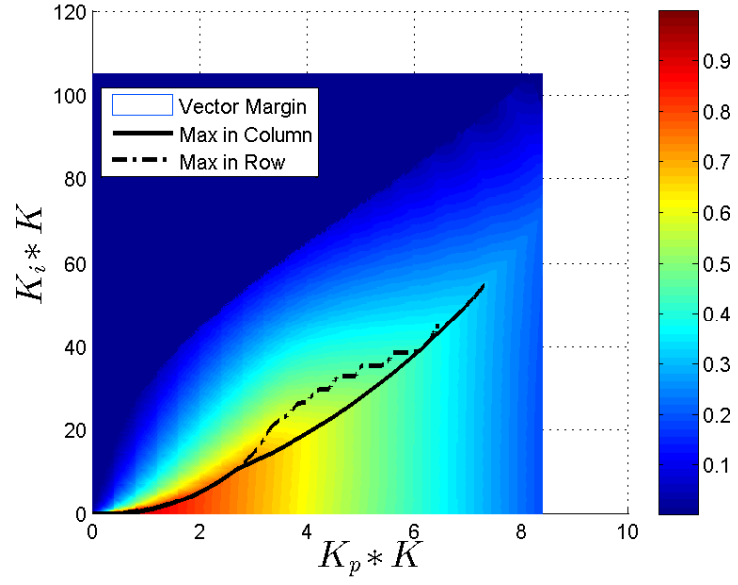


Figure 3.6: Vector Margin for K_p and K_i parameters

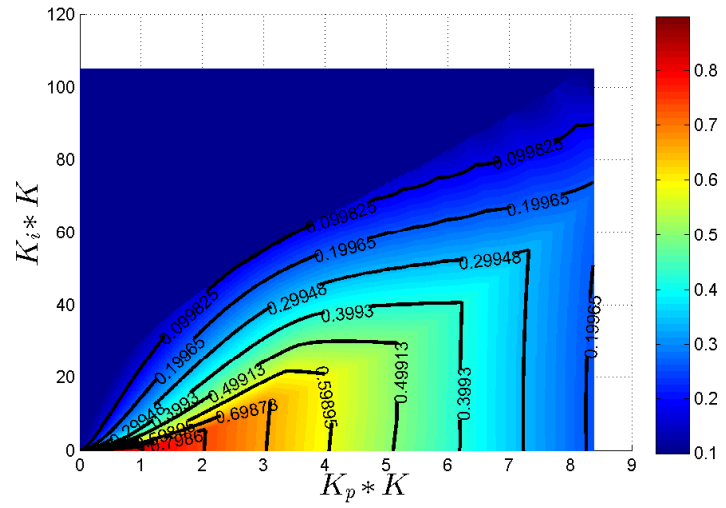


Figure 3.7: Vector Margin and Contour Lines for K_p and K_i parameters

Chapter 4

Performance Analysis

4.1 Comparison of Selected Controller Parameters

Contribution of K_p and K_i parameters to performance criteria for the aforementioned plant is investigated by selecting some controller parameters in the allowable region shown in Figure 3.6. Time domain performance comparisons of these selected controller parameters are observed by considering the percent overshoot (PO), settling time (T_s) and rise time (T_r) in unit step responses. Let us assume two cost functions as:

$$J = PO * \frac{T_s}{10} * (1 - VM)$$

and

$$\hat{J} = PO * \frac{T_s}{10} * \sqrt{(1 - VM)}.$$

Note that data written in bold in Table 4.1 samples taken from the curves shown in Figure 3.6. In Table 4.1 and 4.2 data written in the upper part are taken in the area between two curves given in Figure 3.6 while the rest are taken below those curves. According to Table 4.1 and Table 4.2 plausible controllers

$K_p * K$	$K_i * K$	VM	PO (%)	T_s (sec.)	T_r (sec.)
3.49	19.78	0.6203	57.6	2.103	0.2444
4.178	23.99	0.5643	56.3	1.898	0.2252
3.892	21.88	0.5909	53.6	1.969	0.2324
4.379	27.99	0.5213	61.3	1.387	0.2178
4.597	30.72	0.4919	64.5	1.929	0.2128
4.765	27.99	0.5175	59.7	1.894	0.2145
5.033	32.83	0.4695	65.3	2.256	0.2074
2.366	3.577	0.7663	26.1	3.321	0.472
2.668	6.944	0.7381	31.2	2.738	0.368
3.188	3.367	0.6856	18.3	2.755	0.3901
2.953	8.206	0.7108	31.8	2.491	0.3459
2.869	3.156	0.7167	19.7	2.781	0.4414
3.188	7.996	0.6878	29.1	2.441	0.3403
3.44	5.261	0.662	20.5	2.077	0.3545
3.876	6.313	0.6203	19.93	1.914	0.334
3.624	10.94	0.6407	35.4	2.117	0.2973
3.859	11.36	0.6247	35.7	2.043	0.2707
4.06	11.15	0.6052	34.3	1.32	0.2602
4.228	18.1	0.5902	47.7	1.633	0.2343
4.446	13.68	0.569	38.1	1.63	0.2395
4.698	17.04	0.5447	42.2	1.61	0.2292

Table 4.1: Performance for K_p and K_i parameters

for J and $\hat{J}$ parameters for the data in and below the area shown in 3.6 can be written as:

If we look at the cost function defined, importance of VM (vector margin) is on our scope. By taking square root of J cost function we aim to investigate the effect of VM to our plant. Once an ideal system specifications are defined, lower PO (percent overshoot), shorter T_s and higher VM are expected to be seen. The purpose of defining such J and $\hat{J}$ are fundamentally based on those three defining specifications. Hence, if one aims to obtain a system much closer to an ideal one, it is plausible to look for a cost function with the lowest outcomes. If we look at the results given in Table 4.1 and Table 4.2, we can divide them into two. As mentioned before, one given on the upper section of which the samples

$K_p * K$	$K_i * K$	VM	PO (%)	T_s (sec.)	T_r (sec.)	J	$\hat{J}$
3.49	19.78	0.6203	57.6	2.103	0.2444	4.5994	7.4642
4.178	23.99	0.5643	56.3	1.898	0.2252	4.6558	7.0534
3.892	21.88	0.5909	56.8	1.969	0.2324	4.5753	7.1533
4.379	27.99	0.5213	61.3	1.387	0.2178	4.0701	5.8826
4.597	30.72	0.4919	64.5	1.929	0.2128	6.3218	8.8688
4.765	27.99	0.5175	59.7	1.894	0.2145	5.4557	7.8542
5.033	32.83	0.4695	65.3	2.256	0.2074	7.8152	10.7299
2.366	3.577	0.7663	26.1	3.321	0.472	2.0257	4.1902
2.668	6.944	0.7381	31.2	2.738	0.368	2.2373	4.3718
3.188	3.367	0.6856	18.3	2.755	0.3901	1.5851	2.8269
2.953	8.206	0.7108	31.8	2.491	0.3459	2.2909	4.3599
2.869	3.156	0.7167	19.7	2.781	0.4414	1.5521	2.9160
3.188	7.996	0.6878	29.1	2.441	0.3403	2.2177	3.9690
3.44	5.261	0.662	20.5	2.077	0.3545	1.4392	2.4754
3.876	6.313	0.6203	19.93	1.914	0.334	1.4484	2.3506
3.624	10.94	0.6407	35.4	2.117	0.2973	2.6927	4.4921
3.859	11.36	0.6247	35.7	2.043	0.2707	2.7373	4.4681
4.06	11.15	0.6052	34.3	1.32	0.2602	1.7875	2.8448
4.228	18.1	0.5902	47.7	1.633	0.2343	3.1921	4.9864
4.446	13.68	0.569	38.1	1.63	0.2395	2.6766	4.0771
4.698	17.04	0.5447	42.2	1.61	0.2292	3.0934	4.5844

Table 4.2: Performance for K_p and K_i parameters with cost function

are taken from the area in between the curves defined as in Figure 3.6, whereas the rest are taken from below that area. If we look at the cost function defined, importance of VM (vector margin) is on our scope. By taking square root of J cost function we aim to investigate the effect of VM to our plant. Once an ideal system specifications are defined, lower PO (percent overshoot), shorter T_s and higher VM are expected to be seen.

4.2 Unit Step Responses for the Sample Points Given

It is seen from Table 4.1 and Table 4.2, values for the cost functions written for the upper part are consistently higher than those written below. It is acceptable to assume that datapoints taken below curves are more ideal and give more desirable outputs. In order to prove our assumption, it is feasible apply given K_p proportional action gain and K_i integral action gain to aforementioned plant and check if performance analysis also satisfy with cost function. Thus, for the upper part, two data points with the lowest J and $\hat{J}$ are taken and implemented, their unit step function and Bode plot analysis are investigated accordingly. In the upper part, for both J and $\hat{J}$ results, the two lowest values are obtained when:

$$K_p * K = 3.892 \quad \text{and} \quad K_i * K = 21.88$$

$$K_p * K = 4.379 \quad \text{and} \quad K_i * K = 27.99$$

Since $K = 70$, we end up with the PI controllers given as below:

$$K_p * K = 3.892 \quad \text{and} \quad K_i * K = 21.88 \quad C_{PI} = 0.0556 + \frac{0.3126}{s} \quad (4.1)$$

$$K_p * K = 4.379 \quad \text{and} \quad K_i * K = 27.99 \quad C_{PI} = 0.0626 + \frac{0.3999}{s} \quad (4.2)$$

Hence, unit step responses for the given controllers are obtained as given in Figure 4.1 and Figure 4.2:

For the lower part given in Table 4.1, two datapoints that give the lowest J and $\hat{J}$ are derived as:

$$K_p * K = 3.44 \quad \text{and} \quad K_i * K = 5.261$$

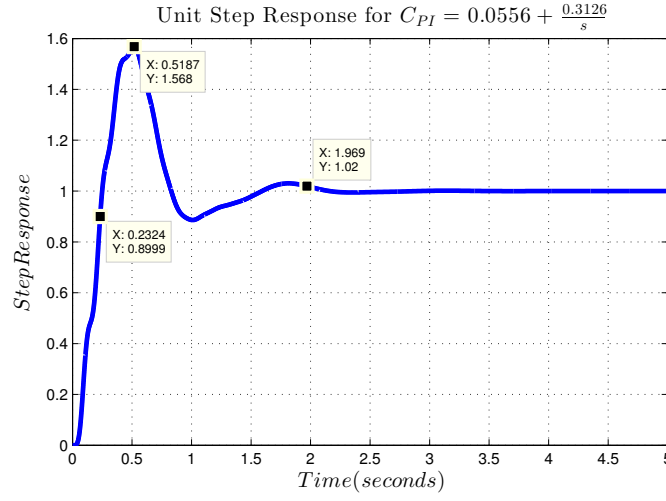


Figure 4.1: Unit Step Response for $C_{PI} = 0.0556 + \frac{0.3126}{s}$

$$K_p * K = 3.876 \quad \text{and} \quad K_i * K = 6.313$$

Since $K = 70$, we end up with the PI controllers given as below:

$$K_p * K = 3.44 \quad \text{and} \quad K_i * K = 5.261 \quad C_{PI} = 0.0491 + \frac{0.0752}{s} \quad (4.3)$$

$$K_p * K = 3.876 \quad \text{and} \quad K_i * K = 6.313 \quad C_{PI} = 0.0554 + \frac{0.0902}{s} \quad (4.4)$$

It should also be noted that even there is not much VM value difference between these point pairs, performance elements differ drastically. In Figure 4.3 and Figure 4.4, maximum points are not as sharp as the ones given in Figure 4.1 and Figure 4.2. The oscillations seen at the overshoot are occurring due to the flexible modes that are existent in our flexible robot arm plant. Although system is forced to have a singular maximum point, due to flexible modes it is not achieved.

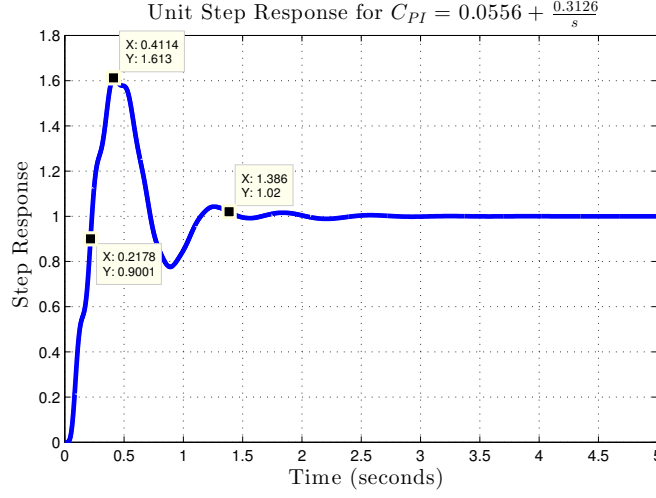


Figure 4.2: Unit Step Response for $C_{PI} = 0.0626 + \frac{0.3999}{s}$

4.3 Performance analysis with Smith-Predictor Based Controller Design

In [1] a Smith-Predictor based controller design is obtained for a plant with flexible modes. The overall controller C_1s for the aforementioned method described in [1] is defined as:

$$C_1(s) = \frac{C_0(s)}{1 + C_0(s)\hat{P}_1(s)(1 - e^{-\hat{T}_d s})} \quad (4.5)$$

When the parametrization algorithm is observed, some primary definitions should be made to make process clearer. According to [1] for our case $\hat{P}_1(s) = P_1(s)$ is accepted as the delay free part which we define as:

$$\hat{P}(s) = \frac{K * R_0(s)}{s}$$

Considering the feedback representation given in [1], closed-loop transfer function is represented as:

$$T_0(s) = \frac{C_1(s)P_1(s)e^{T_d s}}{1 + C_1(s)P_1(s)e^{T_d s}}$$

which is simplified and can be used as,

$$T_0(s) = \frac{C_0(s)P_1(s)}{1 + C_0(s)P_1(s)e^{-T_d s}} \quad (4.6)$$

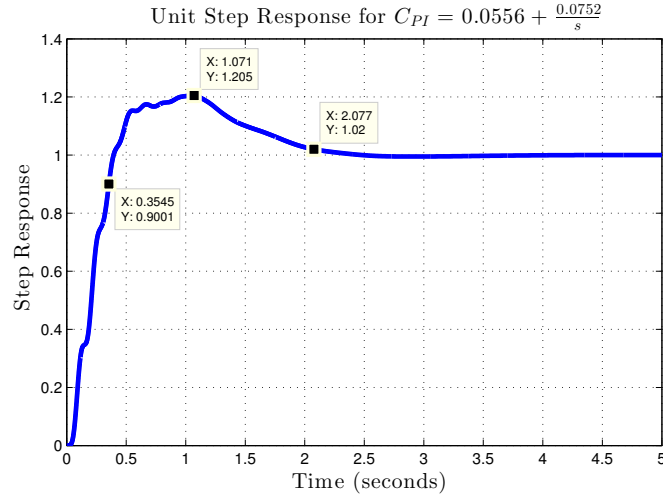


Figure 4.3: Unit Step Response for $C_{PI} = 0.0556 + \frac{0.0752}{s}$

Consequently, in order to design the overall controller for the closed loop feedback system, it is obligatory to achieve $C_0(s)$ primarily. This substitution given in (4.6) aims to remove time delay from the characteristic equation of the closed loop system and thus the controller can be designed without time delay. By the help of Smith Predictor structure, controller design methods for processes without delay can be directly implemented as also mentioned in [1].

To apply Smith Predictor based approach, controller parametrization algorithm should be applied to $C_0(s)$ in (4.6).

In [1], all stabilizing controllers for $P_1(s)$ are parametrized as:

$$C_0(s) = \frac{X(s) + D_p(s)Q(s)}{Y(s) - N_p(s)Q(s)} \quad (4.7)$$

As it is also represented in the aforementioned paper, Q is a free parameter and can be parametrized as:

$$Q(s) = \frac{(bs + c)}{(s + e)}$$

to reject constant and ramp disturbances. Once further reading is made for [1],

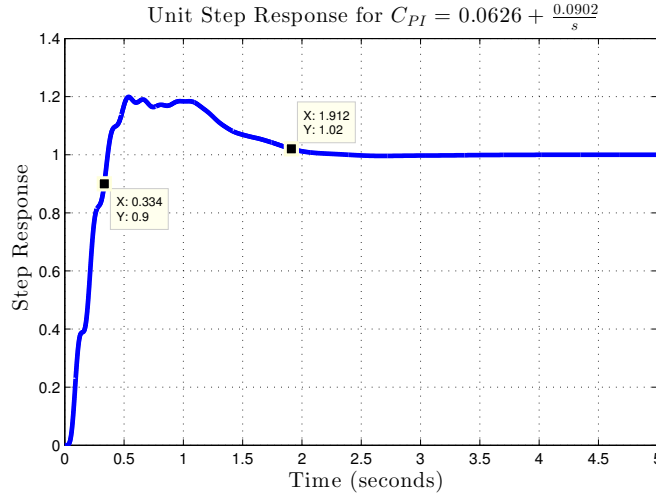


Figure 4.4: Unit Step Response for $C_{PI} = 0.0626 + \frac{0.0902}{s}$

choice for a and e is basically defining the behavior of the system.

4.4 Designing a controller for Smith Predictor Controller Comparison

In order to make right comparison with Smith Controller multiple poles are required for designing controllers for flexible robot arm system. Since a second stage controller is obligatory, by following the algorithm described in Proposition 5 given in [4], it is possible to obtain a controller design for a class of plants with multiple poles and time delay. In [4], the algorithm proposed is described elaborately as following:

Let G have no transmission zeros at $s = 0$. Define:

$$d = (a_1s + 1)(a_2s + 1) \quad \& \quad n = (s - p_1)(s - p_2) \quad (4.8)$$

where p_1, p_2 are unstable poles of the plant and $a_1, a_2 \in \mathbb{R}$ and $a_1, a_2 > 0$ and let G have a form of:

$$G = Y^{-1}X = \left[\frac{n}{d}\right]^{-1} \left[\frac{n}{d}G_{PI}\right]. \quad (4.9)$$

In this expression G represents the part with no transmission zeros where $\widehat{G}$ includes delay elements. Proposition 5 in [4] states:

Let G be defined as in (4.9), with $X = \frac{n}{d}G$ and choose any $\tau_d > 0$. Define,

$$\widehat{\Phi}_1 = \frac{1}{s} \left[\frac{n}{\tau_d s + 1} \widehat{G}(s) X(0)^{-1} - 1 \right]. \quad (4.10)$$

For PD Design: Let $p_1, p_2 \in \mathbb{R}_+$, if $0 \leq p_1 < \Omega$ where,

$$\Omega := \min \left\| \widehat{\Phi}_1 \right\|_{\infty}^{-1}, \quad (4.11)$$

then choose any $\alpha_1 \in \mathbb{R}$ satisfying

$$p_1 < \alpha_1 + p_1 < \Omega.$$

Let,

$$W = (s - p_2) \widehat{G} X(0)^{-1},$$

Define

$$\widehat{\Phi}_2 = \left[\frac{\alpha_1 (1 + \frac{\alpha_1 + p_1}{\tau_d s + 1} W)^{-1} W - 1 \right] \frac{1}{s}. \quad (4.12)$$

If $0 \leq p_2 < \Omega$ where,

$$\Omega_2 := \min \left\| \widehat{\Phi}_2 \right\|_{\infty}^{-1},$$

then choose any $\alpha_2 \in \mathbb{R}$ satisfying

$$p_2 < \alpha_2 + p_2 < \Omega_2. \quad (4.13)$$

Let

$$\widehat{K}_p = (\alpha_1 \alpha_2 - p_1 p_2) X(0)^{-1}, \quad \widehat{K}_d = (\alpha_1 + p_1)(1 + \tau_d p_2) X(0)^{-1} \quad (4.14)$$

then a PD-controller that stabilizes $\widehat{G}$ is given by

$$C_{PD}(s) = \widehat{K}_p + \frac{s \widehat{K}_d}{\tau_d s + 1}. \quad (4.15)$$

For PID Design: Let C_{PD} be given as in (4.15) and define a parameter H_{PD} which yields as:

$$H_{PD} = \frac{\widehat{G}}{1 + C_{PD}\widehat{G}}. \quad (4.16)$$

Define,

$$\Upsilon = \frac{H_{PD}(s)H_{PD}(0)^{-1} - 1}{s},$$

thus for any $\gamma \in \mathbb{R}$ satisfying (4.17), a *PID – controller* that stabilizes $\widehat{G}$ is given by (4.18).

$$0 < \gamma < \min \|\Upsilon\|_{\infty}^{-1}, \quad (4.17)$$

$$C_{PID}(s) = C_{PD}(s) + \frac{\gamma\alpha_1\alpha_2 X(0)^{-1}}{s}. \quad (4.18)$$

After describing milestones for the algorithm that is followed, it is our aim to apply the procedure aforementioned for our plant. Let us memorize our flexible robot arm plant and introduce the algorithm to be followed step by step.

Step 1:

In Section 2 and Section 3 description of the plant and first order controller designs were introduced, respectively. Given,

$$P(s) = K \frac{e^{-hs}R_0(s)}{s}$$

for a flexible robot arm where $R_0(s)$ is the transfer function for flexible modes.

Step 2:

Design of PI controller in the form

$$C_{pi}(s) = C_1(s) + \frac{K_i}{s},$$

by following [4] where $C_1(s) = K_p$ and $C_1(s)$ is in the domain of all controllers that stabilize plant $P(s)$. In Table 4.1 and Table 4.2, a group of points selected are applied to the plant and performance responses are compared. Open-loop transfer function for plant $P(s)$ and $C_{pi}(s)$ is named as $\widehat{G}_{PI}(s)$.

$$P(s) = K \frac{e^{-hs}R_0(s)}{s} \quad \text{and} \quad C_{pi}(s) = C_1(s) + \frac{K_i}{s}$$

$$\widehat{G}_{PI}(s) = \left(K_p + \frac{K_i}{s} \right) \frac{K}{s} e^{-hs} R_0(s) \quad (4.19)$$

and delay-free part is determined as

$$G_{PI}(s) = \left(K_p + \frac{K_i}{s} \right) \frac{K}{s} R_0(s) \quad (4.20)$$

Step 3:

Since it is aimed to design a second stage controller for the flexible robot arm system, (4.19) is now taken as the new plant for Proposition 5 given in [4]. Subsequently, the parameters in (4.8) and (4.9) should be determined in order to proceed. Since G_{PI} is obtained as,

$$\widehat{G}_{PI}(s) = \left(\frac{K_p s}{K_i} + 1 \right) \frac{K_i K}{s^2} e^{-hs} R_0(s)$$

where $p_1, p_2 = 0$ in (4.8) since $s = 0$ for both poles. Moreover, $a_1, a_2 = 1$ is also a plausible assumption since it satisfies both $a_1, a_2 \in \mathbb{R}$ and $a_1, a_2 > 0$. Thus, n and d parameters in (4.8) are,

$$n = s^2 \quad d = (s + 1)^2. \quad (4.21)$$

Once parameters determined for (4.21) are substituted in (4.9),

$$\begin{aligned} X(s) &= \frac{n}{d} G = \frac{s^2}{(s + 1)^2} \left(\frac{K_p s}{K_i} + 1 \right) \frac{K_i K}{s^2} R_0(s) \\ X(s) &= \frac{K_i K \left(\frac{K_p s}{K_i} + 1 \right) R_0(s)}{(s + 1)^2} \end{aligned} \quad (4.22)$$

Considering (4.22) it is appropriate to state,

$$X(0) = K_i K$$

since $R_0(0) = 1$. As we proceed, $\widehat{K}_p$ and $\widehat{K}_d$ given in (4.15) are determined as,

$$\widehat{K}_p = \alpha_1 \alpha_2 X(0)^{-1}, \quad \widehat{K}_d = \alpha_1 X(0)^{-1}$$

As we construct the equations from (4.15) to (4.18) and substitute

them in (4.18) we end up with:

$$C_{PID}(s) = C_{PD}(s) + \frac{\gamma\alpha_1\alpha_2X(0)^{-1}}{s}, \quad (4.23)$$

$$= \hat{K}_p + \frac{s\hat{K}_d}{\tau_d s + 1} + \frac{\gamma\alpha_1\alpha_2X(0)^{-1}}{s}, \quad (4.24)$$

$$= \alpha_1\alpha_2X(0)^{-1} + \frac{s\alpha_1X(0)^{-1}}{\tau_d s + 1} + \frac{\gamma\alpha_1\alpha_2X(0)^{-1}}{s}, \quad (4.25)$$

$$= (\alpha_1X(0)^{-1}) \left(\alpha_2 + \frac{s}{\tau_d s + 1} + \frac{\gamma\alpha_2}{s} \right), \quad (4.26)$$

$$= \alpha_1\alpha_2X(0)^{-1} \left(1 + \left[\frac{s/\alpha_2}{\tau_d s + 1} \right] + \frac{\gamma}{s} \right) \quad (4.27)$$

Step 4:

After determining C_{PID} in detail, apply this method for 2 most distinctive controller parameters points and find responses, i.e. first stage will be initiated with these 2 $C_{PI}(s)$ controllers and start finding required parameters starting from Step 1.

- **Selection 1:** $C_{PI}(s) = 0.0556 + \frac{0.0752}{s}$
- **Selection 2:** $C_{PI}(s) = 0.0626 + \frac{0.0902}{s}$

4.4.1 Design for $C_{PI}(s) = 0.0556 + \frac{0.0752}{s}$

In this design, we aim to apply the steps mentioned in previous sections. In other words, it is aimed to follow each and every step for $C_{PI}(s) = 0.0556 + \frac{0.0752}{s}$. For these values, α_1 and α_2 values are determined as,

$$\alpha_1 = 0.2650 \quad \text{and} \quad \alpha_2 = 0.001$$

If we look at Figure 4.5, γ values with respect to varying α_1 and α_2 can be seen. In Figure 4.5, γ values are determined by using the inequality given in (4.17). Briefly, values shown on z-axis are maximum boundaries for γ given in (4.17) which intersects with α_1 and α_2 values given in x- and y-axis, respectively.

X-Y plane graph for γ is also shown to understand behavior better. Step

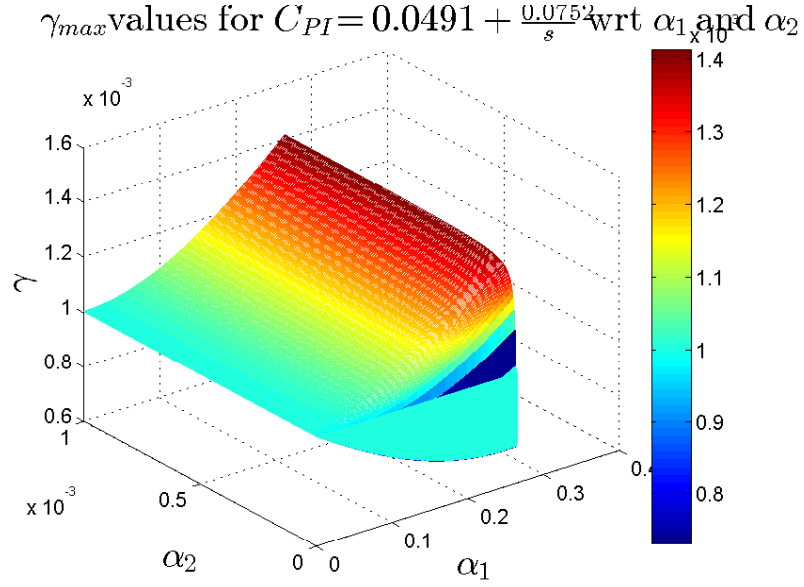


Figure 4.5: Max γ wrt α_1 and α_2 for $C_{PI} = 0.0556 + \frac{0.0752}{s}$

response of the system for the given parameters, can be achieved by following the aforementioned steps. After obtaining α_1 , α_2 and γ values, PID controllers are generated through the surface graph obtained from Figure 4.5. As stated in [12], least fragile integral gain is obtained simply by taking half of the maximum value of γ since γ is a property used for integral action gain. In other words,

$$\gamma_{opt} = \frac{\gamma_{max}}{2}.$$

Thus, VM (vector margin) can be found by implementing all possible generated $C_{PID}(s)$ with G_{PI} and obtaining open-loop sensitivity function. In Figure 4.7 vector margin values for open-loop systems with different C_{PID} are obtained.

In order to observe results better, X-Y plane and X-Z plane graphs are also added.

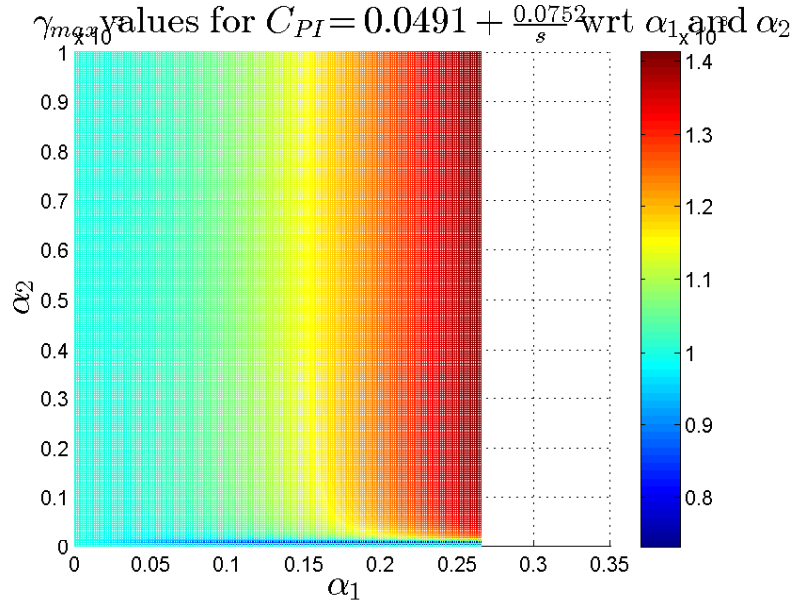


Figure 4.6: X-Y Plane for Figure 4.5

4.4.2 Design for $C_{PI}(s) = 0.0626 + \frac{0.0902}{s}$

Moreover in this sample, α_1 , α_2 and γ values are found by following the equations and inequalities written as in aforementioned section,

$$\alpha_1 = 0.2824 \quad \text{and} \quad \alpha_2 = 0.001$$

In Figure 4.10, upper boundary for γ values are shown. X-Y plane for Figure 4.10, Following similar procedures as in previous section, we aim to find $VM(\text{vectormargin})$ of the feedback system to understand its stability and In Figure 4.12 it can be seen. As also applied in previous example, X-Y and X-Z planes might give us better understanding.

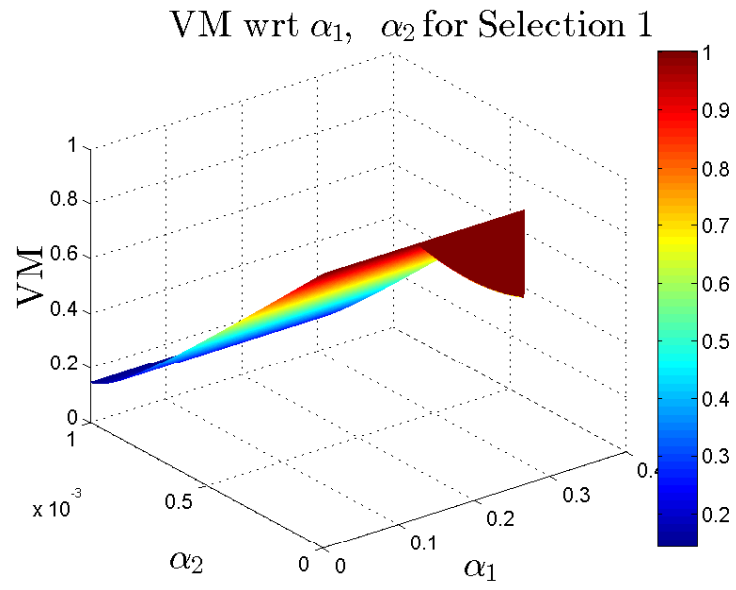


Figure 4.7: VM values for α_1 , α_2 and γ_{opt} for Selection 1

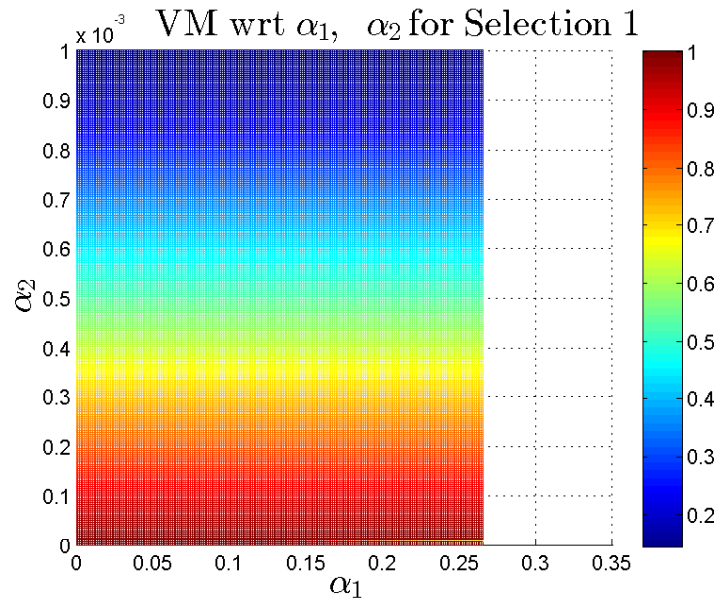


Figure 4.8: X-Y Plane for Figure 4.7

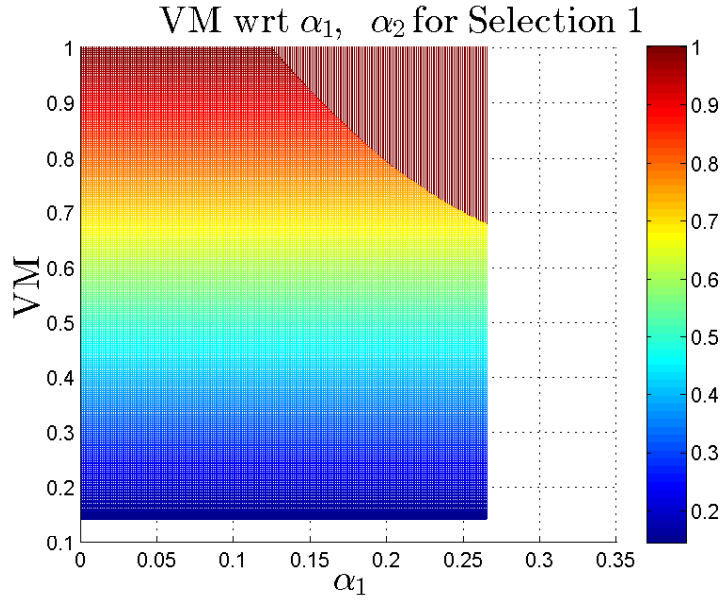


Figure 4.9: X-Z Plane for Figure 4.7

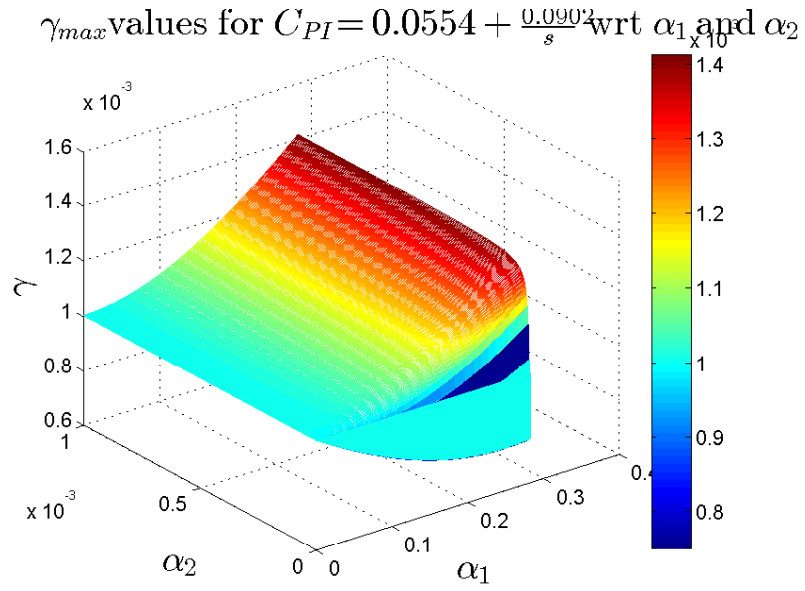


Figure 4.10: γ values wrt α_1 and α_2 for $C_{PI} = 0.0554 + \frac{0.0902}{s}$

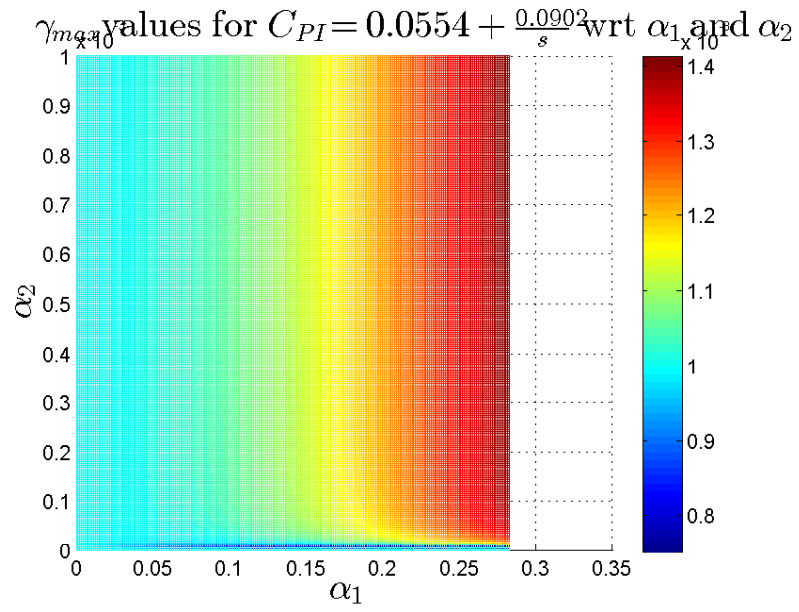


Figure 4.11: X-Y Plane for Figure 4.10

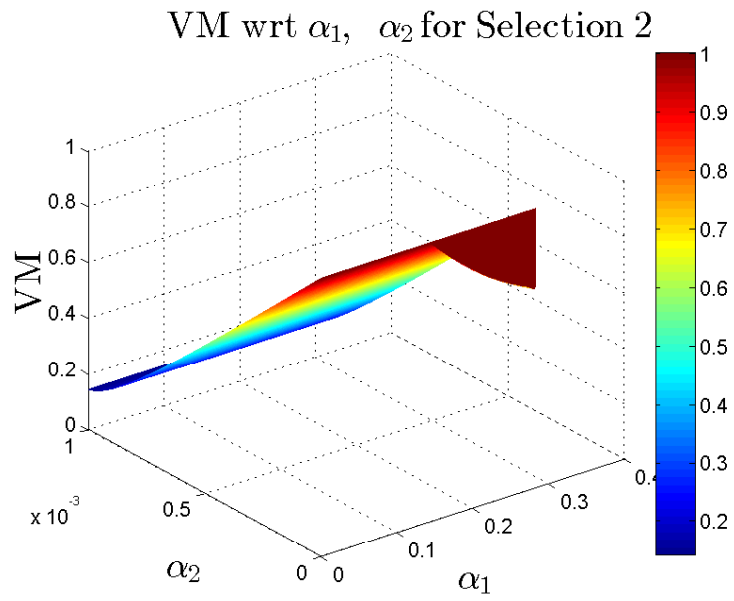


Figure 4.12: γ values wrt α_1 and α_2 for $C_{PI} = 0.0554 + \frac{0.0902}{s}$

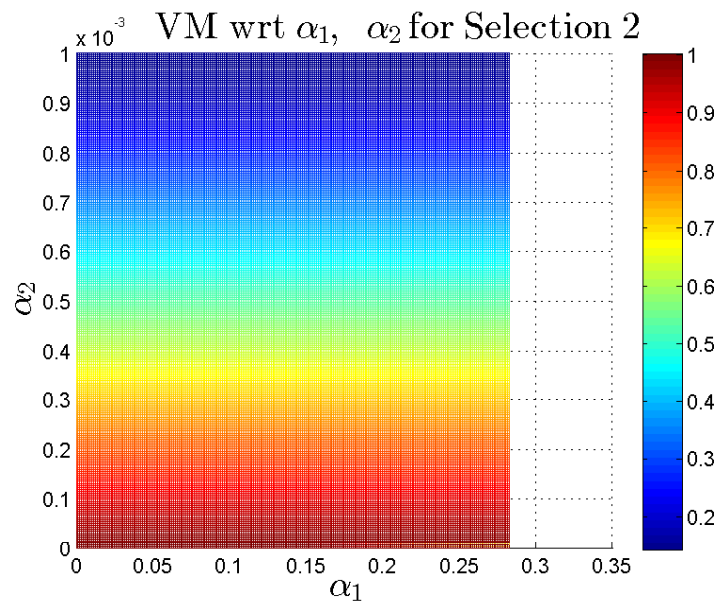


Figure 4.13: X-Y Plane for Figure 4.12

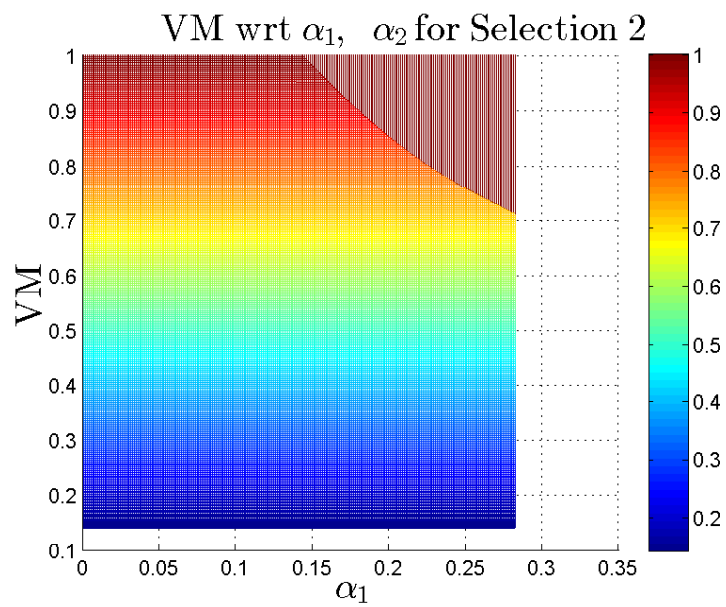


Figure 4.14: X-Z Plane for Figure 4.12

Chapter 5

Conclusion

The design of first order controllers are investigated for a flexible robot arm model which includes a time delay. Infinite dimensional flexible modes are approximated by 12th order rational transfer function using system identification techniques for a frequency response data as in [2]. However, the approach does not exclude infinite dimensional flexible modes, see [11, 12] for more details. The plant considered here contains an integral action (i.e. it is an unstable system with a pole at $s = 0$). Subset of all stabilizing PI controllers is characterized and for each element in this set, robustness and performance measures are investigated. The robustness measure is taken as the vector margin (VM); the performance is characterized by the feedback system's step response. Typically for low-order systems with time delay as the VM increases PO decreases and the settling time T_s increases as well. However, as it can be seen from the last row of Table 4.1, for this particular system with flexible modes and time delay, there is an optimal choice of the parameters of K_p and K_i such that the PO is smallest while maintaining a relatively high VM and having relatively small T_s and T_r .

Moreover, flexible robot arm plant is merged with first order PI controller and the new open-loop transfer function is considered as an expanded plant $\hat{G}_{PI}(s)$. By following the directions written in [11], second stage PID controller is designed step by step. A future study would be to compare the performances of the

controllers designed here with the Smith Predictor based approach of [1], briefly outlined in Section 4.3.

Bibliography

- [1] Taşdelen, U. and Özbay, H., “On Smith Predictor-Based Controller Design for Systems with Integral Action and Time Delay,” *Proceedings of Asian Control Conference*, 2013, Istanbul TURKEY .
- [2] Demir, O. and Özbay, H., “On Reduced Frequency Order Modeling of Flexible Structures from Frequency Response Data,” *European Control Conference*, pp. 1133–1138, Strasbourg, France, 2014.
- [3] Gümüşsoy, S., Michiels, W., “ Fixed-order H_∞ control for interconnected systems using delay differential algebraic equations,” *SIAM J., Control Opt.*, Vol. 22, pp. 2212–2238, 2011
- [4] Gündeş, A.N., Özbay, H., Özgüler B., “PID controller synthesis for a class of unstable MIMO plants with I/O delays ,” *Automatica*, Vol.43, pp.135–142, 2007.
- [5] Hohenbichler, N., “All stabilizing PID controllers for time delay Systems,” *Automatica*, Vol.45, pp.2678–2684, 2009.
- [6] Mendéz-Barrioz, C., Niculescu, S.-I., Morărescu, C.-I. and Gu, K., “On the Fragility of PI Controllers for Time-Delay SISO Systems,” *Mediterranean Conference on Control and Automation*, Ajaccio, France, 2008.
- [7] Michiels, W., Niculescu, S.-I., “Stability and Stabilization of Time-Delay Systems: An Eigenvalue-Based Approach,” *SIAM*, Philadelphia, 2007.
- [8] Niculescu, S.-I., “Delay Effects on Stability: A Robust Control Approach,” Springer, London, 2001.

- [9] Ou, L.-L., Zhang, W.-D, Yu, L., “Low-order stabilization of LTI systems with time delay,” *IEEE Trans. Automatic Control*, Vol. 54, pp. 774–787, 2009.
- [10] Özbay, H., “Introduction to Feedback Control Theory,” CRC Press LLC, Boca Raton, 2000.
- [11] Özbay, H., Gündes, A.N., “Resilient PI and PD controller designs for a class of unstable plants with I/O delays,” *Appl. Comp. Math.*, Vol.6, pp.18–26, 2007.
- [12] Özbay, H., Gündes, A.N., “Design of First Order Controllers for Unstable Infinite Dimensional Plants” in *Low-Complexity Controllers for Time-Delay Systems*, A. Seuret, et al. (eds.) Advances in Delays and Dynamics (ADD@S), Vol. 2, Springer, 2014, pp. 17–30.
- [13] Silva, G.J., Datta, A., Bhattacharyya, S., “PID Controllers for Time-Delay systems,” Birkhauser, Boston, 2005.

Appendix A

MATLAB codes to generate controllers and evaluation of cost functions

```
%% Maximum Boundary For Proportional Gain
clc
clear all
close all

%Plant Properites
load plant_parameters.mat
s=tf('s');
h=0.0055;
K=70;

%Setting Sampling Properties
a=logspace(0,2,1000);
omega=logspace(-3,3,1000);

for k=1:length(a)
```

```

for kk=1:length(omega)
    s=1i*omega(kk);
    r_1(kk)=(((5*s/omz1)^2+(2*Zetaz1*5*s/omz1)+1)/((5*s/om1)^2+...
        (2*Zetap1*5*s/om1)+1)^2);
    r_2(kk)=(((5*s/omz2)^2+(2*Zetaz2*5*s/omz2)+1)^2/((5*s/om2)^2+...
        (2*Zetap2*5*s/om2)+1));
    r_3(kk)=(((5*s/omz3)^2+(2*Zetaz3*5*s/omz3)+1)/((5*s/om3)^2+...
        (2*Zetap3*5*s/om3)+1)^3);

    %Flexible Mode Transfer Function
    R_0(kk)=r_1(kk)*r_2(kk)*r_3(kk);

    %Psi Function for a!=0 case
    psi_func_a(kk)= ((exp(-h*s)*R_0(kk))-1)/(s+a(k));

end

Psi_func_for_a(k)=max(abs(psi_func_a));

end

loglog(a,1./Psi_func_for_a,'k',a,a,'--k','LineWidth',2.5)
% Plot Properties
grid on
hold on

figure(1)
xlabel('$a$ value','Interpreter','Latex');
ylabel('$1/||\psi_a||_{\infty}$','Interpreter','Latex')
title('Relation between $a$ vs $1/||\psi_a||_{\infty}$ value for $K_d=0$',...
    'Interpreter','Latex')
hold on
legend('1/||\psi_a||_{\infty}','a','Location','Best')

```

```

%% Finding Q vs a_{max}(Q) Relation for K_{d}=0

clc
clear all
%Plant Properties
load plant_parameters.mat
s=tf('s');
h=0.0055;
K=70;

%Interval Boundaries
Kd=linspace(-0.03,0.03,400);
a=linspace(1,15,500);
omega=logspace(-3,3,500);

for j=1:length(Kd)
    for k=1:length(a)
        for kk=1:length(omega)
            s=1i*omega(kk);
            r_1(kk)=(((5*s/omz1)^2+(2*Zetaz1*5*s/omz1)+1)/((5*s/om1)^2+...
                (2*Zetap1*5*s/om1)+1)^2);
            r_2(kk)=(((5*s/omz2)^2+(2*Zetaz2*5*s/omz2)+1)^2/((5*s/om2)^2+...
                (2*Zetap2*5*s/om2)+1));
            r_3(kk)=(((5*s/omz3)^2+(2*Zetaz3*5*s/omz3)+1)/((5*s/om3)^2+...
                (2*Zetap3*5*s/om3)+1)^3);

            %Flexible mode transfer function
            R_0(kk)=r_1(kk)*r_2(kk)*r_3(kk);

            %Psi function for a!=0 case
            psi_func_a(kk)= (((exp(-h*s)*R_0(kk))*(1+Kd(j)*s))-1)/(s+a(k));

        end
    end
end

```

```

        gamma_of(k)=max(abs(psi_func_a));
        test1(k)=abs(gamma_of(k)*a(k)-1);
    end
    [min_test1,ind_test1]=min(test1);
    a_max(j)=a(ind_test1);
end

%Plot Properties
plot(Kd,a_max,'k','LineWidth',2.5)
xlabel('$Q$', 'Interpreter','Latex');
ylabel('$a_{\max}(Q)$', 'Interpreter','Latex')
grid on

```

```

%% PI Design for the optimal controller
clc
close all
clear all

load plant_parameters.mat
s=tf('s');
h=0.0055;
K=70;

% Finding  $K_{\{i\}}$  for  $0 < K_{\{p\}} < a_{\{0\}}/K$ 
eps=10-5;
K_p=linspace(0+eps,0.1196-eps,500);
b=linspace(0,50,500);
omega=logspace(-3,3,500);

for j=1:length(K_p)
    for k=1:length(b)
        for kk=1:length(omega)
            s=1i*omega(kk);
            r_1(kk)=(((5*s/omz1)2+(2*Zetaz1*5*s/omz1)+1)/((5*s/om1)2+...
                (2*Zetap1*5*s/om1)+1)2);
            r_2(kk)=(((5*s/omz2)2+(2*Zetaz2*5*s/omz2)+1)2/((5*s/om2)2+...
                (2*Zetap2*5*s/om2)+1));
            r_3(kk)=(((5*s/omz3)2+(2*Zetaz3*5*s/omz3)+1)/((5*s/om3)2+...
                (2*Zetap3*5*s/om3)+1)3);

            %Flexible mode transfer function
            R_0(kk)=r_1(kk)*r_2(kk)*r_3(kk);

            P(kk)=(K*exp(-h*s)*R_0(kk))/s;
            H_1(kk)=P(kk)/(1+K_p(j)*P(kk));
        end
    end
end

```

```

        phi_func_b(kk)= (H_1(kk)*K_p(j)-1)/(s+b(k));

    end
    phi_of(k)=max(abs(phi_func_b));
    test1_b(k)=abs(phi_of(k)*b(k)-1); % Points that satisfy in equality
    %are selected
end
[min_test1,ind_test1]=min(test1_b);% Detecting points and their indices
b_max(j)=b(ind_test1);
end

%Plot Properties
figure()
plot(K_p,b_max.*K_p,'k','LineWidth',3.0)
xlabel('$K_p$','Interpreter','Latex');
ylabel('$K_i$','Interpreter','Latex')
grid on

%%

% Performance Effect for K_p and K_i parameters are analyzed
% from the results we obtained from K_p vs K_i graph

%open('C:\Users\Asus\Desktop\Thesis-Dec2015-2\Thes_Images\...
%K_p-vs_K_i_co.fig')
% From this figure max point gives us b_max*K_p

increment=max(b_max.*K_p)/length(K_p);
VM_all=zeros(length(K_p));
K_i=b_max.*K_p;

%Surface Plot for the Kp vs. Ki relation
% vector margin

```

```

load plant_parameters.mat
s=tf('s');
h=0.0055;
K=70;

omega=logspace(-3,3,500);
for j=1:length(K_p)
    K_i_inc=0:increment:K_i(j)-increment;
    for k=1:length(K_i_inc)
        K_i_array(j,k)=K_i_inc(k);
        for kk=1:length(omega)
            s=1i*omega(kk);
            r_1(kk)=(((5*s/omz1)^2+(2*Zetaz1*5*s/omz1)+1)/((5*s/om1)^2+...
                (2*Zetap1*5*s/om1)+1)^2);
            r_2(kk)=(((5*s/omz2)^2+(2*Zetaz2*5*s/omz2)+1)^2/((5*s/om2)^2+...
                (2*Zetap2*5*s/om2)+1));
            r_3(kk)=(((5*s/omz3)^2+(2*Zetaz3*5*s/omz3)+1)/((5*s/om3)^2+...
                (2*Zetap3*5*s/om3)+1)^3);

            %Flexible mode transfer function
            R_0(kk)=r_1(kk)*r_2(kk)*r_3(kk);

            P(kk)=(K*exp(-h*s)*R_0(kk))/s;

            %optimal PI controller
            % C_pi(kk)=0.012+0.05/s;
            C_pi(kk)=K_p(j)+K_i_array(j,k)/s; % PI Controller
            OLTF_pi(kk)=P(kk)*C_pi(kk);
            S_pi(kk)=1/(1+OLTF_pi(kk));

        end
        VM_all(k,j)=1/max(abs(S_pi));
    end
end

```

```
    end  
end
```

```

clear all

load VM_all_gokce.mat

eps=10^-5;
K_p=linspace(0+eps,0.1196-eps,500);
K_i=linspace(0,1.5,500);
mesh(K_p,K_i,VM_all)
hold on

%Finding Maximum in Columns
hold on
[max_c_VM ind_max_VM]=max(VM_all);
for i=1:length(VM_all)
    curv(i)=K_i(ind_max_VM(i));
end

z=ones(500);
plot3(K_p,curv,z,'LineWidth',2)

% Finding the maximum in Rows
hold on
for i=1:length(VM_all)
    [max_c_VM_r ind_max_VM_r]=max(VM_all,[],2);
    curv_r(i)=K_p(ind_max_VM_r(i));
end
plot3(curv_r,K_i,z,'LineWidth',2)

xlabel('$K_p$', 'Interpreter', 'Latex')
ylabel('$K_i$', 'Interpreter', 'Latex')
zlabel('Vector Margin', 'Interpreter', 'Latex')

```

```

%%
%We will be generating column and row max values lines that are limited
% with VM=0.3 (Limit is adjustable)
clear all
load VM_all_gokce.mat
eps=10^-5;
K_p=linspace(0+eps,0.1196-eps,500);
K_i=linspace(0,1.5,500);

%Upper VM limit
VM_limit=0.3;

%Finding Maximum in Columns
% A z-axis array that will only show upto .3 VM
[max_c_VM ind_max_VM]=max(VM_all,[],1);
for i=2:length(VM_all)
    while (max_c_VM(i)> VM_limit)==1
        curv(i)=K_i(ind_max_VM(:,i));
        z(i)=1;
        i=i+1;
    end
end
z(1)=1;
up_to=length(z); % # of columns that are greater than VM=0.4

mesh(K_p*K,K_i*K,VM_all); %3D plot
hold on
plot3(K_p(1:(up_to-1))*K,K_i(ind_max_VM(1:(up_to-1)))*K,z(1:(up_to-1)),...
    '-k','LineWidth',2.5)
% Finding the maximum in Rows
hold on

[max_c_VM_r ind_max_VM_r]=max(VM_all,[],2);

```

```

for i=1:length(VM_all)

    if (max_c_VM_r(i) >= VM_limit)
        curv_r(i)=K_p(ind_max_VM_r(i,:));
        z_r(i)=1;
    else
        up_to_row=i;
        break
    end
end
plot3(curv_r(:,1:(up_to_row-1))*K,K_i(:,1:(up_to_row-1))*K,...
      z_r(:,1:(up_to_row-1)),'-k','LineWidth',2.5)

xlabel('$K_p*K$', 'Interpreter', 'Latex')
ylabel('$K_i*K$', 'Interpreter', 'Latex')
zlabel('Vector Margin', 'Interpreter', 'Latex')
hleg1 = legend('Vector Margin', 'Max in Column', 'Max in Row');
%%
% Finding VM_lines
% clc
% clear all
%
load VM_all_gokce.mat
eps=10^-5;
K_p=linspace(0+eps,0.1196-eps,500);
K_i=linspace(0,1.5,500);
mesh(K_p*K,K_i*K,VM_all) %3D plot

hold on
% Calculating contou lines
[C,h] = contour(K_p*K,K_i*K,VM_all,9);
clabel(C,h)

```

```

% Loop through each contour line
for i = 1:length(h)
    % Set the line width
    set(h(i),'LineWidth',2,'LineColor','k','ShowText','on')
end

hleg2=legend('Vector Margin','Contour Lines wrt. VM values','Location'...
    , 'Best')
xlabel('$K_p$', 'Interpreter', 'Latex')
ylabel('$K_i$', 'Interpreter', 'Latex')
zlabel('Vector Margin', 'Interpreter', 'Latex')

```

```

%PI Parameters from Workspace to Simulink
Kp_K= 4.178;
Ki_K= 23.99;
%%
figure()
plot(simout)
grid on
%%

PO_a=[57.6 56.3 56.8 61.3 64.5 59.7 65.3];
VM_a=[0.6203 0.5643 0.5909 0.5213 0.4919 0.5175 0.4695];
Ts_a=[2.103 1.898 1.969 1.387 1.929 1.894 2.256];

J=(PO_a).*(Ts_a./10).*(1-VM_a)
J_h=(PO_a).*(Ts_a./10).*sqrt(1-VM_a)
%%
PO_ar=[26.1 31.2 18.3 31.8 19.7 29.1 20.5 19.93 35.4 35.7 34.3 47.7 38.1...
42.2];
VM_ar=[0.7663 0.7381 0.6856 0.7108 0.7167 0.6878 0.662 0.6203 0.6407 ...
0.6247 0.6052 0.5902 0.569 0.5447];
Ts_ar=[3.321 2.738 2.755 2.491 2.781 2.441 2.077 1.914 2.117 2.043 ...
1.32 1.633 1.63 1.61];
J=(PO_ar).*(Ts_ar./10).*(1-VM_ar)
J_h=(PO_ar).*(Ts_ar./10).*sqrt(1-VM_ar)

```