

**COMPUTATION OF THE OPTIMAL \mathcal{H}^∞
CONTROLLER FOR A FRACTIONAL
ORDER SYSTEM**

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND
ELECTRONICS ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Abidin Erdem Karagül

September, 2014

I certify that I have read this thesis and that in my 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)

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

Prof. Dr. Ömer Morgül

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

Prof. Dr. Mehmet Önder Efe

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

COMPUTATION OF THE OPTIMAL \mathcal{H}^∞ CONTROLLER FOR A FRACTIONAL ORDER SYSTEM

Abidin Erdem Karagül

M.S. in Electrical and Electronics Engineering

Supervisor: Prof. Dr. Hitay Özbay

September, 2014

This work investigates the \mathcal{H}^∞ optimal controller design for a fractional order system with time delay. For illustrative purposes, a magnetic suspension system model, derived by Knospe and Zhu is considered. The transfer function is infinite dimensional including e^{-hs} and a rational function of \sqrt{s} , where $h > 0$ represents the delay. Recently in a paper by Özbay, a formulation is given to design the \mathcal{H}^∞ optimal controller for the mixed sensitivity minimization problem for unstable infinite dimensional plants with low order weights. This formulation is used to design the \mathcal{H}^∞ optimal controller for the fractional order system considered, and it is compared to alternative computation methods for \mathcal{H}^∞ control of infinite dimensional systems. To implement the controller, approximation methods are also investigated. Furthermore, finite dimensional rational approximation techniques of the fractional order integrator are evaluated for simulation purposes.

Keywords: Fractional Order Systems, \mathcal{H}^∞ Optimal Control, Approximation of Fractional Order Systems, Time Domain Simulation of Fractional Order Systems.

ÖZET

KESİRLİ DERECEDEDEN BİR SİSTEM İÇİN \mathcal{H}^∞ DENETLEÇ TASARIMI

Abidin Erdem Karagül

Elektrik ve Elektronik Mühendisliği, Yüksek Lisans

Tez Yöneticisi: Prof. Dr. Hitay Özbay

Eylül, 2014

Bu çalışma zaman gecikmeli kesirli dereceden bir sistem için \mathcal{H}^∞ optimal denetleç tasarımı incelemektedir. Knospe ve Zhu tarafından modellenmesi yapılmış olan lamine edilmemiş manyetik süspansiyon sistemi örnek olarak alınmıştır. Modelin transfer fonksiyonu sonsuz boyutludur, e^{-hs} ve \sqrt{s} gibi rasyonel olmayan terimler içermektedir. Burada $h > 0$ zaman gecikmesini göstermektedir. Yakın zamanda Özbay tarafından kararsız sistemler için düşük dereceli ağırlık fonksiyonlarıyla kurulmuş olan karışık ağırlıklı minimizasyon problemini çözen bir yöntem geliştirilmiştir. Bu tezde \mathcal{H}^∞ optimal denetleci tasarlamak için bu yöntem kullanılmış ve daha önce raporlanmış olan sonsuz boyutlu \mathcal{H}^∞ optimal denetleyici yöntemleriyle karşılaştırılmıştır. Kontrolcü tasarımına ek olarak, gerçekleştirme amacıyla, bu kontrolcünün daha düşük dereceli veya gerçeklemeye daha uygun yaklaşımları araştırılmıştır. Elde edilen denetleyicinin kapalı döngüdeki simülasyon sonuçlarının görülebilmesi için $1/\sqrt{s}$ 'in rasyonel yaklaşımları denenmiştir.

Anahtar sözcükler: Kesirli Dereceden Sistemler, \mathcal{H}^∞ Optimal Kontrol, Kesirli Dereceden Sistemlerin Yakınsaması, Kesirli Dereceden Sistemlerin Simülasyonu.

Acknowledgement

First of all, I am grateful to my family for creating an excellent opportunity which allowed me to study in this perfect environment. Without the resources they provided, it would be almost impossible to achieve anything that is already achieved with ease. Their endless love and support have always been a key source to relief. I would also like to thank them for their patience.

I want to express my gratitude to my supervisor Prof. Hitay Özbay. The completion of this work is important but the scope of his support since my undergraduate years is much wider than just guiding this thesis. I am also thankful for his excellent teaching; it would be impossible to grasp these difficult topics otherwise.

I would like to thank Prof. Ömer Morgül for his support and excellent teaching.

I would also like to thank the Bilkent University and the Electrical and Electronics Engineering Department for providing an excellent research environment.

I would also like to thank Roketsan for encouraging me to continue my graduate studies and to complete this thesis.

I would like to acknowledge Prof. Mehmet Önder Efe for reading and commenting on this thesis.

Finally, I would like to take this opportunity to refer to a beautiful river analogy which summarizes our daily talks and expresses our friends' support in an eloquent manner. I would like to thank my friends for sharing thought-provoking issues, long studying hours, delight, and of course amazement.

Contents

1	Introduction	1
2	Mathematical Model of the Plant	10
3	Optimal \mathcal{H}^∞ Controller Design	14
3.1	Factorization of the Plant	14
3.2	Toker-Özbay Formula	16
3.3	Simplified Method Given in [1]	17
3.4	The Optimal \mathcal{H}^∞ Controller	18
4	Implementation of the Fractional Order Terms	25
4.1	FIR Implementation of Fractance Device	33
4.2	Continuous Time Approximation Methods	35
4.3	Frequency Response Based Approximation Method	37
5	Time Domain Simulation Results	40
6	Conclusion	47

CONTENTS

vii

A Code

55

List of Figures

1.1	Standard feedback system	6
1.2	Magnitude and phase plots of Bode's ideal open loop system, with $\alpha = 1.1$ and $k = 6$	8
1.3	Step response of the closed loop system with $P(s)C(s) = 6/s^{1.1}$	8
2.1	Locations of the poles of the plant in [2] for $b = 1$, $10^{-5} < c < 10^5$, solid line shows the stability region in ζ -domain.	11
2.2	Locations of the poles of the plant in [2] for $b = 0.5$, $10^{-5} < c < 10^5$, solid line shows the stability region in ζ -domain.	11
2.3	Bode plots of $G(s^\alpha)$ for $c = 5$	13
3.1	γ vs. $\min(\text{svd}(\mathcal{R}_\gamma))$ (solid line) and \mathcal{P}_γ (dashed line); consistency is verified, $\gamma_{opt} = 1.463$ for $h = 0.15$ and $c = 5$	18
3.2	Magnitude and phase diagrams of C_{opt}	19
3.3	Performance level γ_{opt} versus time delay ($c = 5$).	20
3.4	Suboptimal controller with low pass filter ($c = 5$).	21
3.5	$ W_1S/\gamma $ and $ W_2T/\gamma $ for $h = 0.25$ (red) and $h = 0.15$ (blue).	22

3.6	$\sqrt{ W_1 S ^2 + W_2 T ^2}$ for $h = 0.25$ and $h = 0.15$	22
3.7	Nyquist plots of the open loop system for $k = 0.03$, $k = 0.3$ and $k = 3$, where $h = 0.15$, and $c = 5$	24
4.1	A possible block diagram representation of the plant $P(s)$	26
4.2	Block diagram representations for $K_1(s)$ and $K_2(s)$	26
4.3	Bode plots of C_1 and C_{11} , $h = 0.15$	27
4.4	Bode plots of C_1 and C_{11} , $h = 0.25$	28
4.5	Bode plots of C_1 and C_{12} , $h = 0.15$	28
4.6	Bode plots of C_1 and C_{12} , $h = 0.25$	29
4.7	Bode plots of C_1 and C_{13} , $h = 0.15$	29
4.8	Bode plots of C_1 and C_{13} , $h = 0.25$	30
4.9	Bode plots of C_1 and its approximation by YALTA, for $h = 0.15$	31
4.10	Bode plots of C_1 and its approximation by YALTA, for $h = 0.25$	31
4.11	Error between frequency responses of C_1 and its approximation by YALTA, for $h = 0.15$	32
4.12	Error between frequency responses of C_1 and its approximation by YALTA, for $h = 0.25$	32
4.13	Frequency response of the error $E_1(s)$, $N = 2000$	34
4.14	Frequency response of the error $E_2(s)$, $N = 80$	36
4.15	Frequency response of the error $E_3(s)$, $N = 15$. (Matsuda)	37
4.16	Frequency response of the error $E_4(s)$, $N = 15$. (Carlson)	38

4.17	Frequency response of the error $E_5(s)$, $N = 18$ (<code>invfreqs</code>)	39
5.1	Step responses of the closed loop system for $h = 0.15$, without any disturbance.	41
5.2	Step response of the closed loop system for $h = 0.25$, without any disturbance.	42
5.3	$ T_{yd} $, for $h = 0.15$	43
5.4	$ T_{yd} $, for $h = 0.25$	43
5.5	Maximum disturbance observed at the output for $h = 0.15$, $d(t) = \sin(\omega_d t)$	44
5.6	Maximum disturbance observed at the output for $h = 0.25$, $d(t) = \sin(\omega_d t)$	44
5.7	Output to step disturbance for $h = 0.15$	45
5.8	Output to step disturbance for $h = 0.25$	45
5.9	$ T_{yr}(s) ^{-1}$, for $h = 0.15$	46
5.10	$ T_{yr}(s) ^{-1}$, for $h = 0.25$	46

List of Tables

2.1	<i>Locations and the Phases of the Roots of $\zeta^5 + \zeta^4 - 5 = 0$</i>	12
2.2	<i>Locations and the Phases of the Roots of $\zeta^5 + 0.5\zeta^4 - 5 = 0$</i>	12
3.1	<i>Calculated γ_{opt} and a values for $k = 0.03$ and $k = 3$</i>	23
3.2	<i>Lower, Upper Gain Margins, Phase Margins, and Vector Margins for $k = 0.03$, $k = 0.3$, and $k = 3$</i>	23

Chapter 1

Introduction

This work investigates the \mathcal{H}^∞ optimal controller design for a fractional order system with time delay. For illustrative purposes, a magnetic suspension system model derived in [2] is considered. The \mathcal{H}^∞ control has been studied since 1980s, and resulted in a well developed theory. The version of the \mathcal{H}^∞ optimal control, studied here is the mixed sensitivity minimization problem. This problem seeks the controller satisfying closed loop stability in addition to achieving certain performance with plant uncertainty while minimizing the infinity norm of the two block. Plant uncertainty is modeled by the weight function $W_2(s)$, whereas the class of reference signals to be tracked is denoted by $W_1(s)$. Multiplication of $W_1(s)$ with the sensitivity function, $S(s)$, constitutes one of the blocks, the other one is the multiplication of $W_2(s)$ with the complementary sensitivity function, $T(s)$. For finite dimensional systems, solutions can be obtained through the state space methods. In the infinite dimensional case a frequency domain approach can be used and the optimal controller can be found directly without going through any approximation, see [3] and references therein for detailed discussions. The main contribution of this work is the application of this theory to the new class of systems, called fractional order systems.

In simplest terms, an equation relating a variable to its derivatives is called as a differential equation. The nice thing about differential equations is that they can be used as the mathematical models of physical systems, as in the case of

Newton's second law. When the system itself is deterministic, the solution of the differential equation reveals the course of events for all time.

"Given for one instant an intelligence which could comprehend all the forces by which nature is animated and the respective situation of the beings who compose it an intelligence sufficiently vast to submit these data to analysis it would embrace in the same formula the movements of the greatest bodies of the universe and those of the lightest atom; for it, nothing would be uncertain and the future, as the past, would be present to its eyes." [4]

In general, when the law of a physical system is described as a differential equation the order of the derivatives are integer. In other words, a describing equation will involve first, second or higher order derivatives of variables. However, although not easy to visualize, it is possible to have orders of differentiation that are not integers. Surprisingly, its history goes back to the integer order calculus. In a letter to L'Hôpital, Leibniz asked: "Can the meaning of derivatives with integer order be generalized to derivatives with non-integer orders?" [5].

When the derivative order is integer, it is easy to understand what it represents geometrically. For example, first derivative is the slope of the curve at the differentiation point. However, when the order is non-integer, differentiation seem to loose its interpretation as the rate of change in a variable. To create an insight for the non-integer case, an n -fold integral can be considered:

$$\underbrace{\int \dots \int_0^t}_{n} f(y) \underbrace{dy \dots dy}_n = \int_0^t \frac{f(y)(t-y)^{n-1}}{(n-1)!} dy.$$

In other words, an n -fold integral can be expressed as a one dimensional integral. When the order is generalized to $n \in \mathbb{R}^+$, replacing the factorial function with the gamma function can be used as a way to interpret the Riemann-Liouville definition for the fractional order integral [6]:

$$\int_0^t \frac{f(y)(t-y)^{n-1}}{(n-1)!} dy = \frac{1}{\Gamma(n)} \int_0^t f(y)(t-y)^{n-1} dy.$$

Another definition of fractional order differentiation is given by Grünwald-Letnikov [7]. This definition can be interpreted in a similar way, instead of starting

with an n -fold integral, derivation should be started with differentiation.

When Riemann-Liouville definition is used for a fractional order differential equation, it is necessary to have initial conditions expressed in terms of initial values of fractional derivatives of the unknown function. Caputo's definition [8]:

$$D^\alpha f(t) = \frac{1}{\Gamma(1-\delta)} \int_0^t \frac{f^{\Delta+1}(y)}{(t-y)^\delta} dy$$

with

$$\alpha = \Delta + \delta, \quad \Delta \in \mathbb{Z}, \quad \text{and } 0 < \delta \leq 1,$$

on the other hand, requires initial values of integer order derivatives. When the initial conditions are taken to be zero, all three definitions coincide [9].

Applications of fractional order calculus theory in system modeling increased widely in the last decades, since, they offer better fit to some physical systems. For example, heat conduction, mass transportation, viscoelasticity are described better with fractional order differential equations [7]. As in the case of this work, the mathematical model of the non-laminated magnetic suspension system, derived in [2], is represented with a fractional order differential equation.

Not only in system modeling, fractional order calculus is also studied in feedback control theory. Stability results for finite dimensional linear fractional differential systems are given in [10]. Internal and external stabilities are guaranteed if and only if roots of a polynomial lie outside the angular sector $|\arg(\sigma)| \leq \alpha\pi/2$, with α denoting the fractional order and σ denoting the roots of the polynomial. \mathcal{H}_2 -norm of a system is defined as the energy of the impulse response. Analytical computation of the \mathcal{H}_2 -norm of fractional commensurate order transfer functions can be done with the method given in [11]. It is stated that unlike integer order systems \mathcal{H}_2 -norm can be infinite although the fractional order system is BIBO stable. The \mathcal{H}_∞ -norm is defined as the largest energy among output signals resulting from all inputs of unit energy, and this definition also holds for the fractional order systems. Hamiltonian matrix definition for fractional order systems and two separate methods to calculate the \mathcal{H}_∞ -norm based on this matrix can be found in [12].

Assuming zero initial conditions the input/output behavior of a system can be represented by the following equation in time domain:

$$\begin{aligned} & a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \dots + a_0 D^{\alpha_0} y(t) \\ = & b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \dots + b_0 D^{\beta_0} u(t), \end{aligned}$$

here $D^\gamma f(t)$ denotes the Caputo derivative of order γ , and α_i, β_j for $i \in \{0, 1, \dots, n\}$; $j \in \{0, 1, \dots, m\}$ are rational numbers; and $\{a_0, a_1, \dots, a_n\}$, $\{b_0, b_1, \dots, b_m\}$ are real constants.

Then, the same system can be represented, in the frequency domain, by the transfer function:

$$G(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \dots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + a_0 s^{\alpha_0}}, \quad \text{where}$$

β_k and α_j are rational numbers, for $k = 0, 1, \dots, m$; and $j = 0, 1, \dots, n$.

The Laplace transform of the fractional order integral, with Caputo's definition, is [8]:

$$\int_0^\infty \left(\frac{1}{\Gamma(\alpha)} \int_0^t f(y) (t-y)^{\alpha-1} dy \right) e^{-st} dt = s^{-\alpha} F(s), \quad \text{where } \alpha > 0.$$

In control theory, Bode is one of the first researchers realizing the importance of the application of the fractional order calculus. He has called the following as the ideal open loop transfer function [7], [13]:

$$F(s) = \frac{A}{s^\alpha}. \quad (1.1)$$

The gain of the open loop transfer function, A , is greater than zero. And for phase margin to be greater than $\pi/2$, α should be in between 0 and 1. In fact, with smaller values of α , it is possible to have large values of the phase margin. This is one of the main principles of Bode's loop shaping which says that at the gain crossover frequency the magnitude drop should be small to have a large phase margin [14]. Phase shift of the ideal transfer function, (1.1), is

constant for all frequency values, so the phase margin of the feedback loop is invariant to the changes in the amplifier gain. Note that s^α , is an irrational function of the Laplace variable s , so (1.1) is an infinite dimensional transfer function with unlimited memory. Having infinite memory enables the fractional order integral action to consider the whole history of the input signal [15]. With this property fractional order is different from integer order integral or derivative, creating possible benefits for some applications. Advantages of application of the fractional order calculus to the feedback control theory increased the number of papers in this field.

In 1990s a non-integer order robust control method, CRONE is proposed, [16]. Superior performance of the fractional order controller $PI^\lambda D^\mu$ over the classical PID controller is shown, [17]. In [18], an algorithm for the co-design of gain and phase margins using fractional order PI^λ controllers is presented. Besides $PI^\lambda D^\mu$ and CRONE algorithms, \mathcal{H}_2 and \mathcal{H}_∞ control strategies are also proposed for fractional order systems [19], [20]. A PID controller design algorithm for fractional order systems with time delays is presented in [21]. Variable order fractional controllers are proposed in [22], aiming a constant phase margin. For fractional order systems with time delay a stabilization algorithm by using $PI^\lambda D^\mu$ controllers is given in [23]. A set point weight rules for $PI^\lambda D^\mu$ controllers is addressed in [24]. For fractional order nonlinear discrete time systems, switched state model predictive control is provided in [25]. A second order D^α type iterative learning control scheme for fractional order systems is presented in [26]. Other than the methodologies to design controllers, analysis methods of fractional order systems are also researched. Stability windows, for fractional systems with time delays, can be determined by using the numerical procedure outlined in [27]. A MATLAB toolbox, YALTA, which is based on [28], [29], [30], [31] can be used for the stability analysis of classical and fractional systems with commensurate delays [32]. The bounded input bounded output stability condition for distributed-order systems over integral interval (0,1) has been established in [33]. Some examples of the fractional order controllers can also be found in the literature. In [34], a fractional order controller is designed for a DC-DC buck converter by using frequency domain techniques. Other applications involve controller designs for a hydraulic

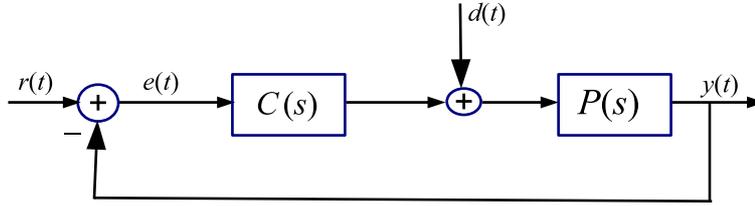


Figure 1.1: Standard feedback system

actuator [35], flexible transmission [36], a robot manipulator [37], a lightweight flexible manipulator [38]. It is widely known that there are physical devices that can perform fractional order integration or differentiation [39]. However, due to the infinite memory property of the fractional order systems, their simulation is complex. Approximate rational transfer functions can be found by frequency domain fitting techniques in continuous or discrete time. Carlson's method [40], Matsuda's method [41], and Oustaloup's method [42] are based on continuous time. For discrete equivalences, FIR implementation can be performed. For alternative discretization methods see [43], [44]. Furthermore, MATLAB's `invfreqs` command can be used for approximations of fractional order transfer functions from the frequency response data. Various methods for analog circuit realizations of fractional order systems can also be found in [45], [39].

This thesis considers the standard feedback system, depicted in Figure 1.1.

A feedback loop involving fractional order systems can be classified in two groups:

- (i) a fractional order controller used for a finite dimensional plant,
- (ii) an integer or fractional order controller used for a fractional order plant.

To our knowledge, most of the existing literature on this subject addresses feedback loops of type (i), in other words, fractional order controllers are designed for rational plants.

As an example, consider a minimum phase plant $P(s)$. By virtue of Bode's loop shaping choose a fractional order controller so that open loop system has the following transfer function:

$$C(s) = \frac{k}{s^\alpha P(s)}, \Rightarrow P(s)C(s) = \frac{k}{s^\alpha}, \text{ with } k > 0, \text{ and } \alpha > 0.$$

Phase margin of the closed loop system is $(\pi - (\pi/2)\alpha)$. Crossover frequency is given by $k^{1/\alpha}$. As seen, phase margin only depends on α , so through the choice of α phase margin can be specified. System stability can be achieved with a positive gain value, $k > 0$, for $0 < \alpha < 2$. With given requirements, a pre-design can be conducted. For example, for $PM = 80^\circ$, α is equal to 1.1 and for cut-off frequency (ω_c) of 5 *rad/sec*, k should be equal to 6, since $k = \omega_c^\alpha$. Figure 1.2 illustrates the magnitude and phase plots of the Bode's ideal open loop system with $k = 6$ and $\alpha = 1.1$. The unit step response is shown in Figure 1.3 This discussion is valid for minimum phase plants. In case of a non-minimum phase plant, the methods presented in [34] can be followed, to compensate the effects of right half plane zeros.

This work concentrates on a feedback system of type (ii). In other words, a specific fractional order plant is considered and an \mathcal{H}^∞ controller design method for this plant is illustrated. A mathematical model of the non-laminated magnetic suspension system is derived in a series of papers [2], [46]. The transfer function of the system is in the form of a rational function of \sqrt{s} followed by a time delay term e^{-hs} , where $h > 0$ represents the time delay. The method proposed in [47], can be used to compute the \mathcal{H}^∞ optimal controller for such infinite dimensional systems. Later in [1], the method in [47] has been simplified for the case where the sensitivity weight is low-order. In this thesis, the mixed sensitivity minimization problem will be solved for the unstable fractional model developed in [2], [46]; first by using the method in [1], and then the result will be verified by the old design procedure of [47]. This thesis also concentrates on the implementation and simulation of the closed loop system. For implementation purposes approximation of the controller is investigated. To see the dynamic behavior of the closed loop system in a simulation environment, methods to obtain integer order transfer functions approximating fractional order terms in the closed loop system, are

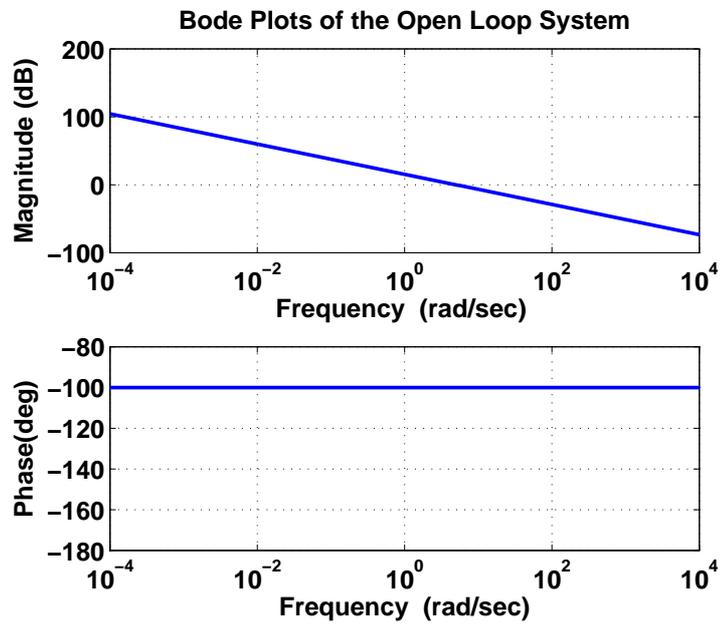


Figure 1.2: Magnitude and phase plots of Bode's ideal open loop system, with $\alpha = 1.1$ and $k = 6$.

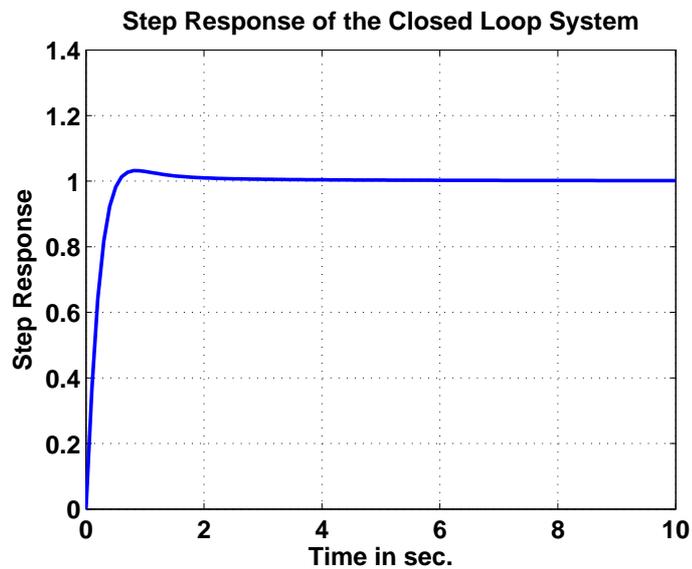


Figure 1.3: Step response of the closed loop system with $P(s)C(s) = 6/s^{1.1}$.

evaluated.

This thesis is based on our papers published recently, [48], [49], [50], [51]. The thesis is organized as follows. In Chapter 2, the mathematical model of the plant is described. In Chapter 3 a detailed discussion on the computation of the \mathcal{H}^∞ optimal controller is given. Chapter 4 outlines some recent techniques for approximation of fractional order terms in the optimal \mathcal{H}^∞ controller expression. Time domain simulation results are given in Chapter 5. Finally, concluding remarks are made in Chapter 6.

Chapter 2

Mathematical Model of the Plant

In this section, the fractional order plant model of a non-laminated magnetic suspension system given by [2] is investigated. Due to real time data acquisition and transmission actuator, sensor time delays may be present in the system. This results in the following transfer function:

$$P(s) = \frac{K_o e^{-hs}}{(s^\alpha)^5 + b(s^\alpha)^4 - c}. \quad (2.1)$$

In (2.1), s is the Laplace variable, $h > 0$ denotes the time delay and α is a rational number and $\alpha \in (0, 1)$. For the plant under consideration $\alpha = 0.5$, $b = 1$ and $K_o = 1$ is fixed, [2]. For finding the locations of the poles in the ζ -plane

$$\zeta = s^\alpha .$$

transformation can be used. With this transformation, the stability region in the ζ -plane is as follows [27]:

$$|\angle \zeta| > \alpha \frac{\pi}{2}, \text{ where } \angle \zeta \text{ denotes the phase of } \zeta \text{ and it is taken in } [-\pi, \pi].$$

It is shown that for all $c > 0$ and $b = 1$ the plant has only one unstable real pole and 4 complex stable poles, [2]. Locations of the poles with respect to c , are given in Figure 2.1 and Figure 2.2 for $b = 1$ and $b = 0.5$.

A specific value of $c = 5$, results in the poles in the ζ plane, given in Tables 2.1, 2.2 for $b = 1$ and $b = 0.5$.

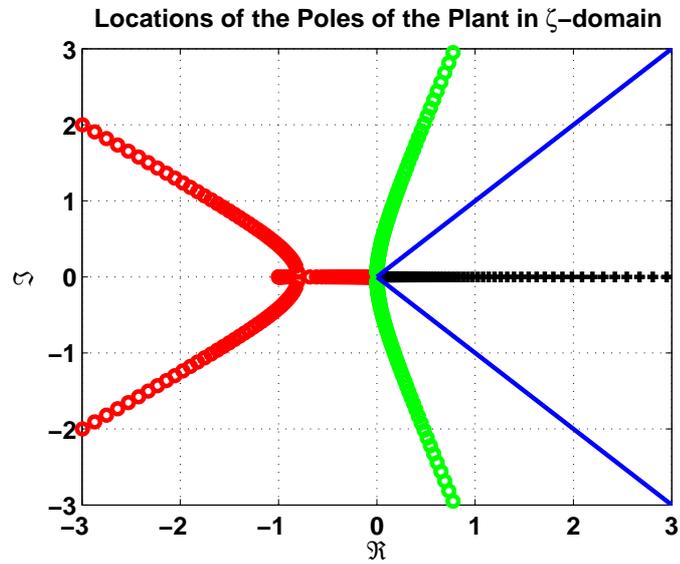


Figure 2.1: Locations of the poles of the plant in [2] for $b = 1$, $10^{-5} < c < 10^5$, solid line shows the stability region in ζ -domain.

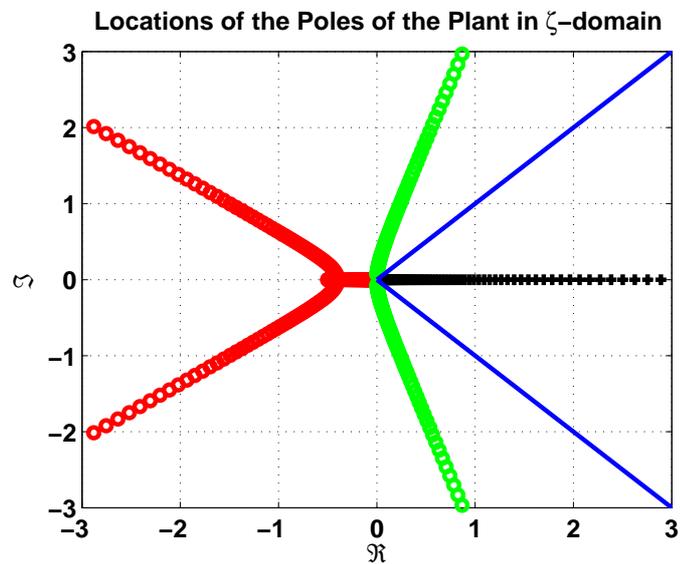


Figure 2.2: Locations of the poles of the plant in [2] for $b = 0.5$, $10^{-5} < c < 10^5$, solid line shows the stability region in ζ -domain.

Table 2.1: *Locations and the Phases of the Roots of $\zeta^5 + \zeta^4 - 5 = 0$*

Locations of the Roots	the Phases
$p_1 = -1.3665 + j0.7563$	$150^\circ = 2.61 \text{ rad}$
$p_2 = -1.3665 - j0.7563$	$-150^\circ = -2.61 \text{ rad}$
$p_3 = 0.2542 + j1.2687$	$78^\circ = 1.36 \text{ rad}$
$p_4 = 0.2542 - j1.2687$	$-78^\circ = -1.36 \text{ rad}$
$p = 1.2244$	$0^\circ = 0 \text{ rad}$

Table 2.2: *Locations and the Phases of the Roots of $\zeta^5 + 0.5\zeta^4 - 5 = 0$*

Locations of the Roots	the Phases
$p_1 = -1.2285 + j0.8001$	$147^\circ = 2.56 \text{ rad}$
$p_2 = -1.2285 - j0.8001$	$-147^\circ = -2.56 \text{ rad}$
$p_3 = 0.3323 + j1.2997$	$76^\circ = 1.33 \text{ rad}$
$p_4 = 0.3323 - j1.2997$	$-76^\circ = -1.33 \text{ rad}$
$p = 1.2923$	$0^\circ = 0 \text{ rad}$

Then, the transfer function of the plant can be re-written so that $G(s^\alpha)$ shows the stable and minimum phase part of the system:

$$P(s) = e^{-hs} G(s^\alpha) \frac{1}{(s^\alpha - p)}, \quad G(s^\alpha) = \frac{1}{(s^\alpha - p_1)(s^\alpha - p_2)(s^\alpha - p_3)(s^\alpha - p_4)},$$

Bode plots of the $G(s^\alpha)$ are given in the Figure 2.3.

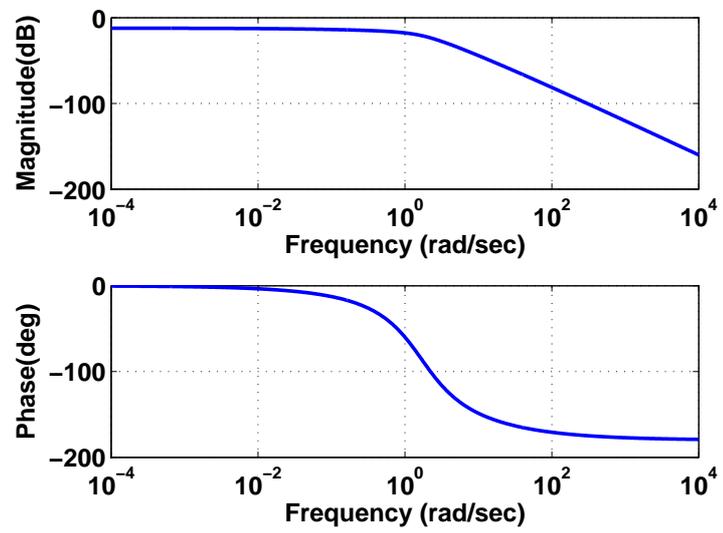


Figure 2.3: Bode plots of $G(s^\alpha)$ for $c = 5$.

Chapter 3

Optimal \mathcal{H}^∞ Controller Design

The optimal \mathcal{H}^∞ controller for the fractional order plant model, analyzed in the previous section, is designed in this section. The plant involves infinite dimensional terms like e^{-hs} and a rational function of \sqrt{s} . The technique proposed in [47] solves the mixed sensitivity minimization problem for unstable, infinite dimensional plants. In [1], it is shown that when the sensitivity weight is low-order, the method proposed in [47], can be simplified. To compute the optimal \mathcal{H}^∞ controller, these two methods are applied separately. This section is divided into four parts. In the first section, factorization of the plant is given, then in Subsection 3.2 and 3.3 two methods are applied separately to compute the optimal controller. In Subsection 3.4, the optimal \mathcal{H}^∞ controller is given.

3.1 Factorization of the Plant

Mixed sensitivity minimization problem tries to find the optimal controller resulting in the optimum performance level:

$$\gamma_{opt} := \min_{C \in \mathcal{C}(P)} \left\| \begin{bmatrix} W_1(1 + PC)^{-1} \\ W_2PC(1 + PC)^{-1} \end{bmatrix} \right\|_\infty = \left\| \begin{bmatrix} W_1(1 + PC_{opt})^{-1} \\ W_2PC_{opt}(1 + PC_{opt})^{-1} \end{bmatrix} \right\|_\infty,$$

Here set of all controllers, stabilizing the standard feedback loop formed by the plant P and the controller C is denoted by $\mathcal{C}(P)$. Feedback system stability is

satisfied if and only if all transfer functions from all external signals to internal signals are stable. That is, the closed loop system is stable if and only if $(1 + PC)^{-1}$, $P(1 + PC)^{-1}$, and $C(1 + PC)^{-1}$ are all stable. In the mixed sensitivity minimization problem, W_1 and W_2 are rational weighting functions. $W_1(s)$ denotes the reference signal generator and $W_2(s)$ represents an upper bound on the multiplicative plant uncertainty. Typically, $W_1(s)$ is a low-order, low pass filter and $W_2(s)$ is a high pass filter. More detailed discussion on the motivation of the \mathcal{H}_∞ controller design for infinite dimensional systems can be found in references of [52].

The plant under consideration given by (2.1) can be factorized as follows [53]:

$$P(s) = \frac{M_n(s)N_o(s)}{M_d(s)}. \quad (3.1)$$

In (3.1), $N_o(s)$ is an outer function, $M_n(s)$ is an inner function, and $M_d(s)$ is an inner function whose zeroes $\alpha_1, \dots, \alpha_l \in \mathbb{C}_+$ are the unstable poles of the system. $M_d(s)$ should be a rational function of s to be able to use the method given in [47]. Therefore, to find $M_d(s)$ the fact that $\alpha = 0.5$ and $(s^\alpha - p)(s^\alpha + p) = (s - p^2)$ is used. Resulting in the following factorization of the plant (2.1):

$$M_n(s) = e^{-hs}, \quad M_d(s) = \frac{(s - p^2)}{(s + p^2)}, \quad \text{and}$$

$$N_o(s) = \frac{(s^\alpha + p)}{(s + p^2)(s^\alpha - p_1)(s^\alpha - p_2)(s^\alpha - p_3)(s^\alpha - p_4)}.$$

In the specific case considered here, we have $l = 1$ and $\alpha_1 = p^2$. In this work, for simplicity of exposition low-order weights are chosen:

$$W_1(s) = \frac{1}{s}, \quad W_2(s) = ks, \quad \text{where} \quad k = 0.3$$

and the notation $W_1 = nW_1/dW_1$ is used with $nW_1(s) = 1$ and $dW_1(s) = s$. The choice of $W_1(s)$ guarantees zero steady state error for step like reference signals. Note that representation of $N_o(s)$ is not minimal in s^α .

3.2 Toker-Özbay Formula

The optimal \mathcal{H}^∞ controller, for the factorized plant, (3.1) has the form given in (3.2), see [47].

$$C = E_\gamma M_d \frac{N^{-1} F_\gamma L}{1 + M_n F_\gamma L} \quad (3.2)$$

with

$$E_\gamma(s) = \frac{W_1(-s)W_1(s)^2}{\gamma} - 1, \quad F_\gamma(s) = \frac{dW_1(-s)}{nW_1(s)} \gamma G_\gamma(s),$$

where the stable $G_\gamma(s)$ is to be found from the spectral factorization:

$$G_\gamma(s)G_\gamma(-s) = \left(1 + \frac{W_2(-s)W_2(s)}{W_1(-s)W_1(s)} - \frac{W_2(-s)W_2(s)}{\gamma^2}\right)^{-1}.$$

The optimum performance level, denoted by γ_{opt} , is found by using the parametrized matrix (3.5). And $L(s)$ is obtained from the set of linear equations:

Define $L(s)$:

$$L(s) = \frac{\begin{bmatrix} 1 & s & \dots & s^{n-1} \end{bmatrix} \Psi_2}{\begin{bmatrix} 1 & s & \dots & s^{n-1} \end{bmatrix} \Psi_1} \quad (3.3)$$

where $n := n_1 + l$, with $n_1 = \deg(dW_1)$. The unknown coefficients Ψ_1 and Ψ_2 are defined as: $\Psi_1 = [\psi_{10} \dots \psi_{1(n-1)}]^T$, $\Psi_2 = [\psi_{20} \dots \psi_{2(n-1)}]^T$. Ψ_1 and Ψ_2 satisfies:

$$\Psi_1 = \pm \mathcal{J}_n \Psi_2, \quad \mathcal{J}_n \Psi_2 =: \Phi,$$

where \mathcal{J}_n is $n \times n$ diagonal matrix, whose i^{th} diagonal entry is equal to $(-1)^{i+1}$. To determine $L(s)$, Φ is used and it is the singular vector of the parametrized matrix, \mathcal{R}_γ , corresponding to zero singular value obtained by the largest feasible $\gamma > 0$. This γ value denotes the optimum performance level, γ_{opt} .

$$\mathcal{R}_\gamma \Phi = 0 \quad (3.4)$$

with the parametrized matrix \mathcal{R}_γ given as:

$$\mathcal{R}_\gamma = \begin{bmatrix} \mathcal{V}_\alpha^l & \mathcal{D}_\alpha \mathcal{V}_\alpha^{n_1} \\ \mathcal{V}_\beta^l & \mathcal{D}_\beta \mathcal{V}_\beta^{n_1} \end{bmatrix} \pm \begin{bmatrix} \mathcal{D}_l & 0 \\ 0 & \mathcal{D}_{n_1} \end{bmatrix} \begin{bmatrix} \mathcal{V}_\alpha^l & \mathcal{D}_\alpha \mathcal{V}_\alpha^{n_1} \\ \mathcal{V}_\beta^l & \mathcal{D}_\beta \mathcal{V}_\beta^{n_1} \end{bmatrix} \mathcal{J}_n. \quad (3.5)$$

Where:

$$\mathcal{D}_l = \text{diag}(M_n(\alpha_1)F_\gamma(\alpha_1), \dots, M_n(\alpha_l)F_\gamma(\alpha_l))$$

$$\mathcal{D}_{n_1} = \text{diag}(M_n(\beta_1)F_\gamma(\beta_1), \dots, M_n(\beta_{n_1})F_\gamma(\beta_{n_1}))$$

$$\mathcal{D}_n = \text{blockdiag}(\mathcal{D}_l, \mathcal{D}_{n_1})$$

\mathcal{V}_x^m denotes $k \times m$ dimensional Vandermonde matrix, constructed from a given vector $x = [x_1 \dots x_k]^T \in \mathbb{C}^k$ and $\beta_1, \dots, \beta_{n_1} \in \mathbb{C}_+$ are the zeros of $E_\gamma(s)$. With those set of equations, the parametrized matrix, \mathcal{R}_γ , is obtained.

With the above equations, it is possible to obtain the parametrized matrix, \mathcal{R}_γ . The optimal performance level γ_{opt} is the largest value of γ which makes \mathcal{R}_γ singular with '+' or '-' sign used in (3.5). Then corresponding, nonzero Φ , satisfying (3.4) is found, yielding Ψ_2 and Ψ_1 . With Ψ_2 and Ψ_1 , $L(s)$ is obtained.

3.3 Simplified Method Given in [1]

A step like reference inputs yields a first order $W_1(s)$, giving $\beta_1 = j/\gamma$. And for the plant considered here, analyzed in the previous section, there is only one unstable pole, $\alpha_1 = p^2$. As shown in [1], the matrix equation involving \mathcal{R}_γ , given in (3.4), reduces to a scalar equation, $\mathcal{P}_\gamma = 0$.

$$\mathcal{P}_\gamma = j\gamma p^2 \left(1 \pm M_n\left(\frac{j}{\gamma}\right)F_\gamma\left(\frac{j}{\gamma}\right)\right) \frac{(1 \mp M_n(p^2)F_\gamma(p^2))}{(1 \pm M_n(p^2)F_\gamma(p^2))} + (1 \mp M_n\left(\frac{j}{\gamma}\right)F_\gamma\left(\frac{j}{\gamma}\right)), \quad (3.6)$$

The largest γ value making \mathcal{R}_γ singular is γ_{opt} , where '-' sign used in (3.5) and this is also the largest γ satisfying $\mathcal{P}_\gamma = 0$ with '-' sign used in (3.6). Therefore, both (3.5) and (3.6) can be used to find γ_{opt} . Figure 3.1 illustrates this point for some particular choices of h and c .

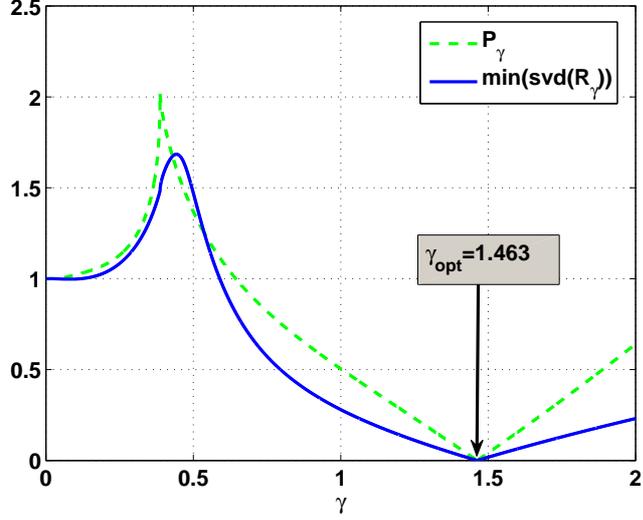


Figure 3.1: γ vs. $\min(\text{svd}(\mathcal{R}_\gamma))$ (solid line) and \mathcal{P}_γ (dashed line); consistency is verified, $\gamma_{opt} = 1.463$ for $h = 0.15$ and $c = 5$

Figure 3.1 shows that both of the algorithms given in [1] and [47] reach the same optimum performance level: $\gamma_{opt} = 1.463$. Now, $L(s)$ is computed from the set of linear equations given above, resulting in the optimal controller:

$$C_{opt} = E_{\gamma_{opt}} M_d \frac{N_o^{-1} F_{\gamma_{opt}} L}{1 + M_n F_{\gamma_{opt}} L}. \quad (3.7)$$

3.4 The Optimal \mathcal{H}^∞ Controller

Once γ_{opt} is computed as above, corresponding \mathcal{R}_γ is determined as

$$\mathcal{R}_\gamma = \begin{bmatrix} 1.6268 & 0.5596 \\ 1.5960 + j0.80 & 0.5490 + j0.3 \end{bmatrix}$$

with singular vector:

$$\Psi_2 = \begin{bmatrix} -0.3253 & -0.9456 \end{bmatrix}$$

yielding:

$$L(s) = \frac{0.9456s + 0.3253}{0.9456s - 0.3253} = \frac{as + 1}{as - 1}, \text{ with } a = 2.9.$$

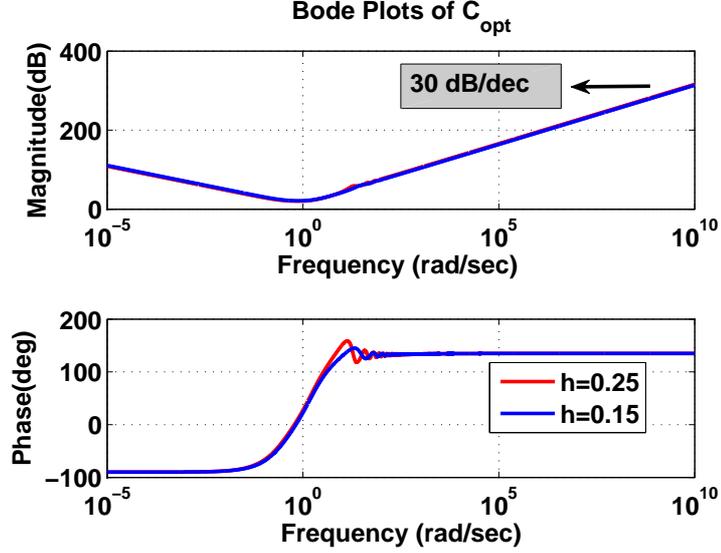


Figure 3.2: Magnitude and phase diagrams of C_{opt}

Now with the γ_{opt} value computed, numerical values of the functions $E_{\gamma_{opt}}(s)$ and $F_{\gamma_{opt}}(s)$ can be obtained:

$$E_{\gamma_{opt}}(s) = \frac{1 + \gamma_{opt}^2 s^2}{-\gamma_{opt}^2 s^2} = \frac{1 + 2.14s^2}{-2.14s^2}$$

$$F_{\gamma_{opt}}(s) = \frac{-\gamma_{opt}s}{ks^2 + k_a s + 1}; \quad \text{with} \quad k_a = \sqrt{2k - k^2/\gamma_{opt}^2},$$

$$F_{\gamma_{opt}}(s) = \frac{-1.463s}{0.3s^2 + 0.75s + 1}.$$

Finding above functions enables the computation of the optimal controller, C_{opt} , given by 3.7. Note that, in this work for illustration purposes optimal controller is computed for different time delay values, namely $h = 0.15$, and $h = 0.25$. Figure 3.2 illustrates the frequency response of the optimal controller for these two cases.

From Figure 3.2, the effect of time delay on the frequency response of the optimum controller can be seen. To investigate the effect of time delay and the value of c on the achievable performance level, γ_{opt} is computed for different values of time delay, for the cases: $c = 0.1, 1, 5, 10$ and the result is depicted in

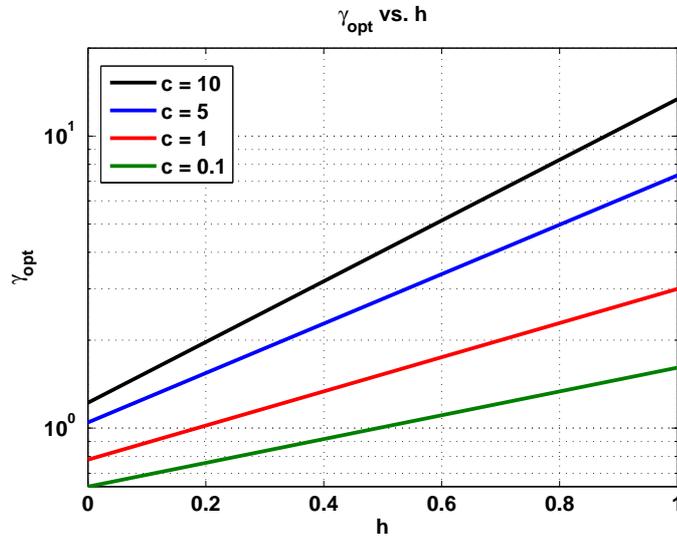


Figure 3.3: Performance level γ_{opt} versus time delay ($c = 5$).

Figure 3.3. As seen from this plot, γ_{opt} increases exponentially with the time delay h .

From Figure 3.2, it is also possible to observe that the optimum controller is improper, the magnitude of the frequency response increases with the increasing frequency. In other words, the degree of the nominator of the transfer function of the optimal controller is greater than its denominator. A suboptimal proper controller is required for practical reasons. Connecting a low pass filter in series with the optimal controller solves this problem. However, introduction of a new system may result in the instability of the feedback loop. Therefore, this low pass filter should be chosen in a way that the resultant closed loop system is still stable. Choice of a filter in the form: $1/(1+\epsilon s)^v$ with values, $v = 2$ and $\epsilon = 0.005$ guarantees the feedback stability, and the choice of this specific v value results in a strictly proper suboptimal controller. That is, the degree of the denominator is strictly greater than the degree of the numerator. The parameter ϵ is chosen in a way that a roll off occurs in the magnitude of the frequency response of C_{subopt} for $\omega \geq 200$ rad/sec. Again, to investigate the effect of time delay on the frequency response of the suboptimal controller C_{subopt} , frequency response is plotted for different time delay values, $h = 0.15$ and $h = 0.25$. The plot is

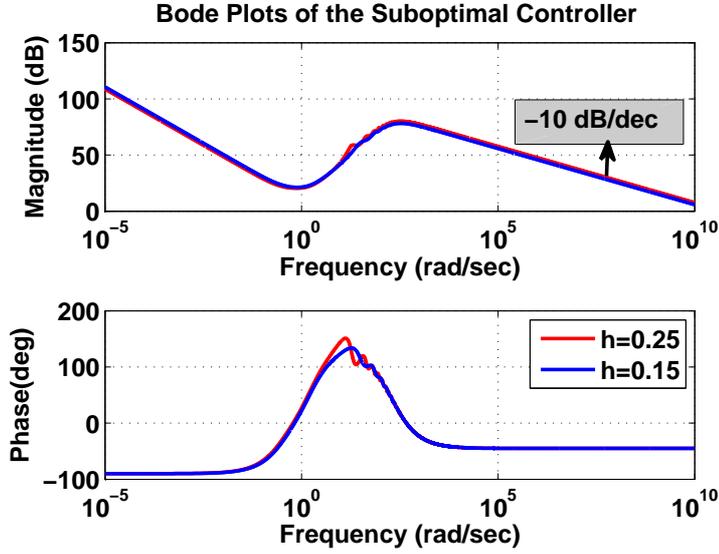


Figure 3.4: Suboptimal controller with low pass filter ($c = 5$).

depicted in Figure 3.4.

Use of the proper suboptimal controller instead of the optimal controller results in deviation of the achievable performance level from the optimal level. To see the effect of introduction of the low pass filter on the weighted sensitivity and complementary sensitivity W_1S , W_2T , the plots of these functions for $h = 0.15$ and $h = 0.25$ are shown in Figure 3.5.

Figure 3.6 depicts the performance level corresponding to C_{subopt} . As stated, addition of the low pass filter caused deviation from the optimal performance level.

Effects of time delay and addition of the low pass filter on performance level are discussed above. However, the achievable performance level is also affected by the multiplicative plant uncertainty, $W_2(s)$. In the above discussion, results are obtained for $W_2(s) = ks$ with $k = 0.3$. To see the effect of multiplicative plant uncertainty, two other different k values are investigated, namely $k = 0.03$ and $k = 3$, and \mathcal{H}^∞ optimum controllers corresponding to those values

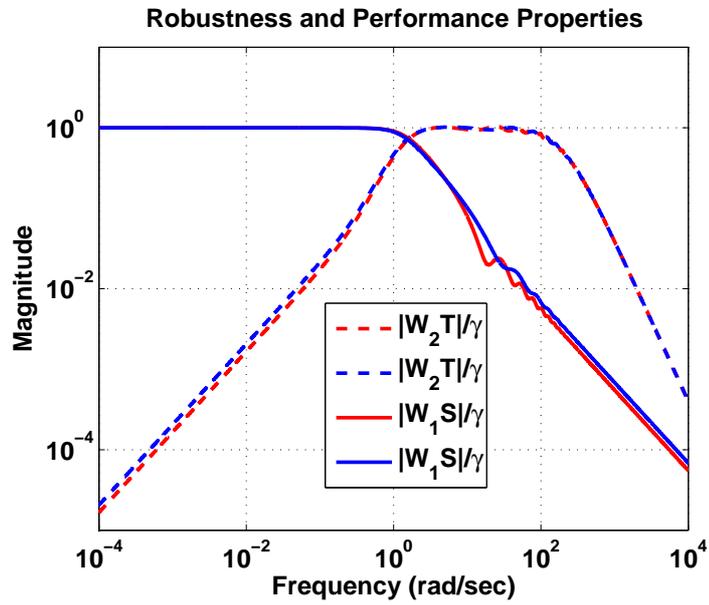


Figure 3.5: $|W_1S/\gamma|$ and $|W_2T/\gamma|$ for $h = 0.25$ (red) and $h = 0.15$ (blue).

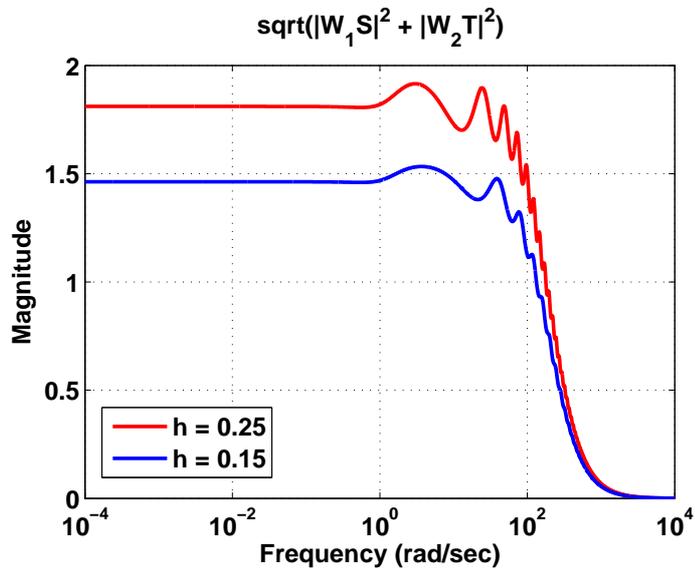


Figure 3.6: $\sqrt{|W_1S|^2 + |W_2T|^2}$ for $h = 0.25$ and $h = 0.15$.

Table 3.1: Calculated γ_{opt} and a values for $k = 0.03$ and $k = 3$

			γ_{opt}	a
$c = 5$	$k = 3$	$h = 0.15$	8.466	11.584
		$h = 0.25$	9.981	13.137
	$k = 0.03$	$h = 0.15$	0.401	1.352
		$h = 0.25$	0.560	1.574

Table 3.2: Lower, Upper Gain Margins, Phase Margins, and Vector Margins for $k = 0.03$, $k = 0.3$, and $k = 3$

			Lower GM	Upper GM	PM	VM
$c = 5$	$k = 0.03$	$h = 0.15$	0.32	2.00	33°	0.48
		$h = 0.25$	0.40	1.56	25°	0.34
	$k = 0.3$	$h = 0.15$	0.53	3.13	33°	0.57
		$h = 0.25$	0.58	2.13	25°	0.43
	$k = 3$	$h = 0.15$	0.74	4.45	28.5°	0.33
		$h = 0.25$	0.78	2.94	22°	0.27

are computed. In Table 3.1, new γ_{opt} values, and $L(s) = (as + 1)/(as - 1)$ functions for $k = 3$ and $k = 0.03$ are given. Figure 3.7 show the Nyquist plots for the open loop system formed by the optimum controller and the plant for $k = 0.03$, $k = 0.3$ and $k = 3$. Effect of multiplicative uncertainty on the resulting upper gain margin, lower gain margin, phase margin and vector margin values are also investigated and Table 3.2 gives these values. It is observed that high and low values of k lead to smaller stability margins. Effect of k on step response, and allowable uncertainty will be discussed later in Chapter 5.

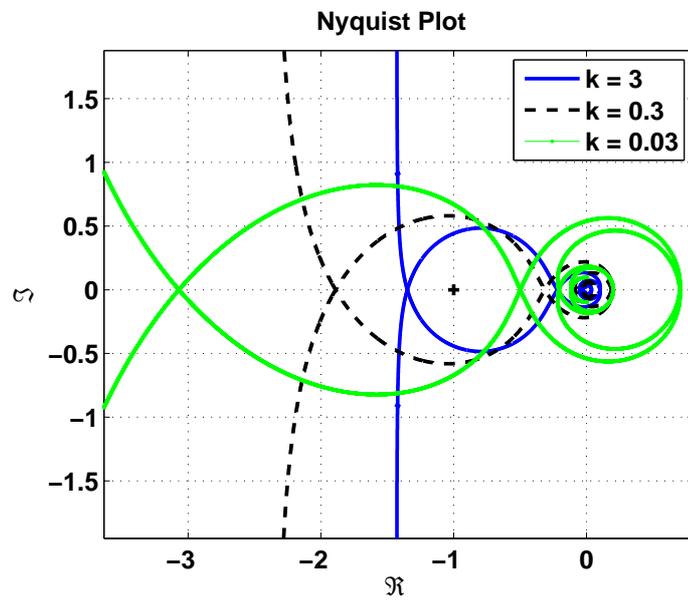


Figure 3.7: Nyquist plots of the open loop system for $k = 0.03$, $k = 0.3$ and $k = 3$, where $h = 0.15$, and $c = 5$.

Chapter 4

Implementation of the Fractional Order Terms

The previous section dealt with the design of the optimal controller for the non-laminated magnetic suspension system model, derived in [2]. The resulting controller and the plant include fractional order terms. Fractional order terms have infinite memory so integer order approximations to those terms are required for implementations purposes. In this section, continuous, discrete, and frequency response based approximation techniques are investigated. For an extensive review of these methods see [54]. It is observed that both the controller and the plant can be written in a form such that the fractional device ($1/s^\alpha$) where $0 < \alpha < 1$ can be separated from the integer order part. Therefore, to implement the suboptimal controller and the plant it is enough to approximate the fractional device.

Figure 4.1 depicts a possible block diagram for the realization of the plant (with $c = 5$).

The suboptimal controller can be written in the form:

$$C_{opt,\epsilon} = C_o(s)C_1(s)C_2(s), \quad (4.1)$$

with

$$C_o(s) = \frac{c(1 + as)}{\gamma_{opt}s}, \quad a = 2.9, \quad c = 5, \quad \gamma_{opt} = 1.463 \text{ for } h = 0.15$$

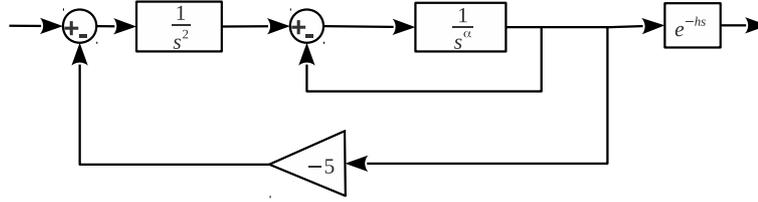


Figure 4.1: A possible block diagram representation of the plant $P(s)$.

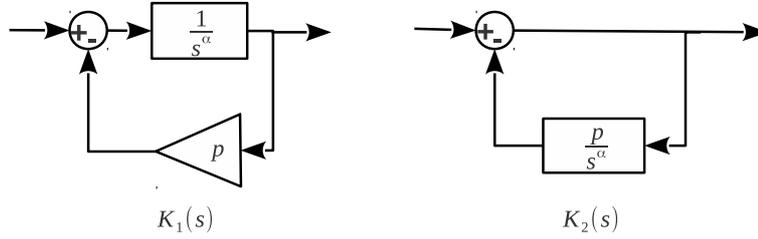


Figure 4.2: Block diagram representations for $K_1(s)$ and $K_2(s)$

$$C_1(s) = \frac{(1 + \gamma_{opt}^2 s^2)(1 - s/p^2)}{(ks^2 + k_a s + 1)(1 - as) + \gamma_{opt} s(as - 1)e^{-hs}}, \quad (4.2)$$

and

$$C_2(s) = \frac{(1 - s^\alpha/p_1)(1 - s^\alpha/p_2)(1 - s^\alpha/p_3)(1 - s^\alpha/p_4)}{(1 + s^\alpha/p)(1 + \epsilon s)^2}. \quad (4.3)$$

In this form, only $C_2(s)$ includes fractional order terms; and it can be rewritten as:

$$C_2(s) = \frac{1}{p_1 p_2 p_3 p_4} \left(K_1(s) \frac{s^2 + a_2 s + a_4}{(1 + \epsilon s)^2} + K_2(s) \frac{a_1 s + a_3}{(1 + \epsilon s)^2} \right)$$

where a_i for $i = 1, 2, 3, 4$ follow from the polynomial multiplication and K_1, K_2 are as defined in Figure 4.2.

Before applying approximation techniques to the fractional term in $C_2(s)$, $C_1(s)$ will be investigated for implementation purposes. Internal unstable pole

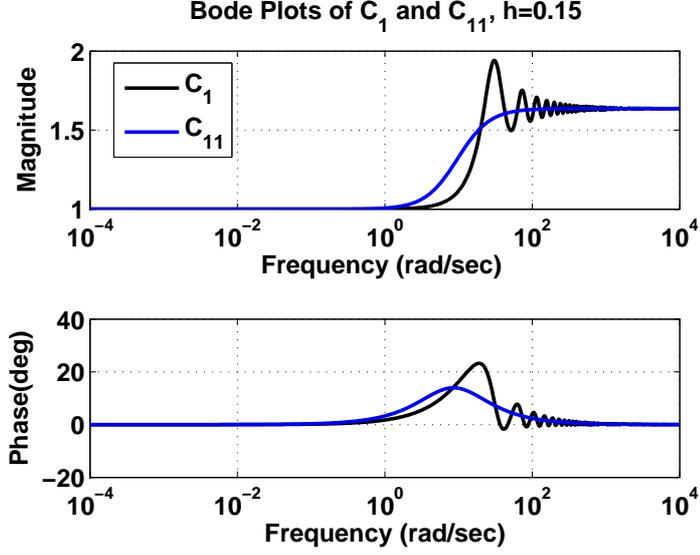


Figure 4.3: Bode plots of C_1 and C_{11} , $h = 0.15$

zero cancellations at $s = p^2$ and $s = \pm j/\gamma$ occur in $C_1(s)$, and the approximate transfer function can be written in the form:

$$C_1(s) \approx C_{11}(s) = \frac{1 + (h/2)s}{1 + (h/2)\tau_c s}, \text{ where } \tau_c = \lim_{s \rightarrow \infty} C_1(s) = \frac{k a p^2}{\gamma_{opt}^2};$$

resulting in the following frequency responses given in Figure 4.3 and 4.4.

For illustrative purposes these frequency responses are obtained for two different time delay values $h = 0.15$, and $h = 0.25$. Bode plots of a second degree approximation is depicted in Figure 4.5 and Figure 4.6.

$$C_1(s) \approx C_{12}(s) = \frac{1 + (h/2)s + (h^2/12)s^2}{1 + (h/2)\tau_c s + (h^2/12)\tau_c^2 s^2}.$$

Third order approximation to C_1 can also be found:

$$C_1(s) \approx C_{13}(s) = \frac{1 + (h/2)s + (h^2/10)s^2 + (h^3/120)s^3}{1 + (h/2)\tau_c s + (h^2/10)\tau_c^2 s^2 + (h^3/120)\tau_c^3 s^3}.$$

Frequency response of this approximation is given in Figure 4.7 and Figure 4.8.

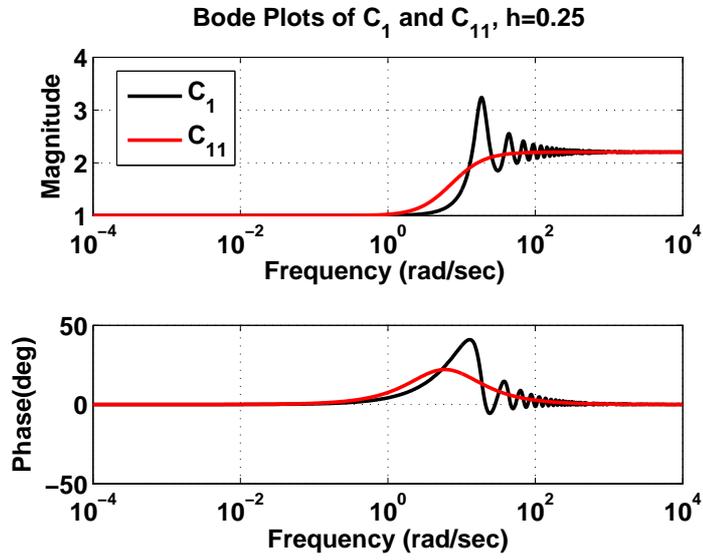


Figure 4.4: Bode plots of C_1 and C_{11} , $h = 0.25$

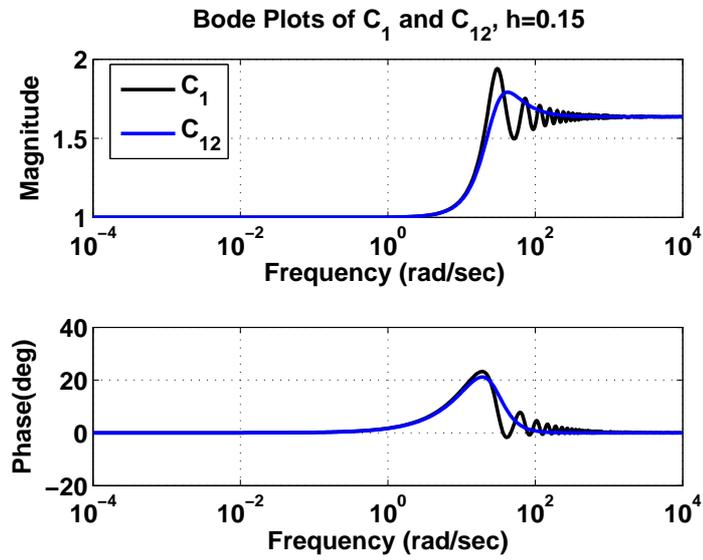


Figure 4.5: Bode plots of C_1 and C_{12} , $h = 0.15$

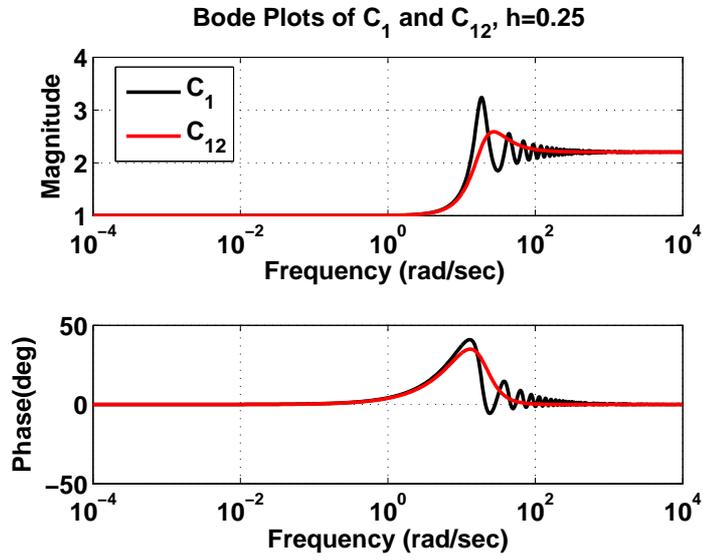


Figure 4.6: Bode plots of C_1 and C_{12} , $h = 0.25$

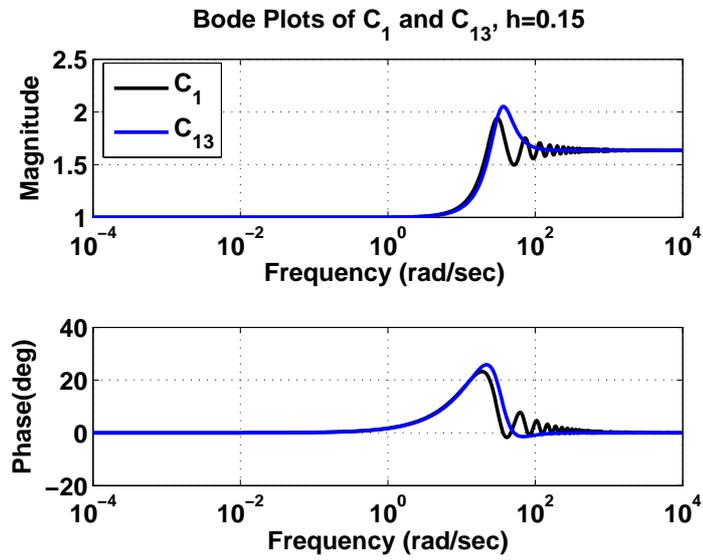


Figure 4.7: Bode plots of C_1 and C_{13} , $h = 0.15$

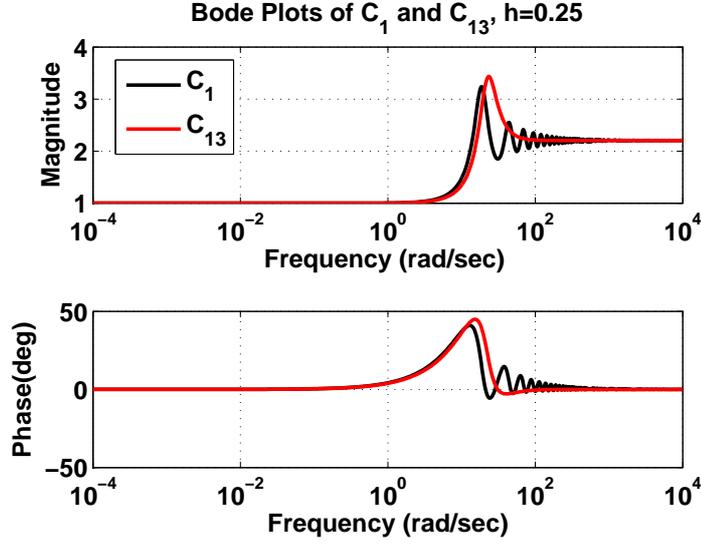


Figure 4.8: Bode plots of C_1 and C_{13} , $h = 0.25$

MATLAB toolbox YALTA can also be used to approximate $C_1(s)$, [32]. For $h = 0.15$, a 16th order rational transfer function is obtained. With this approximation $\|C_1(s) - C_{1appr}(s)\|_\infty < 0.083$. Bode plots of this approximation is given in Figure 4.9

When the time delay increases the error between frequency responses of the rational approximation and $C_1(s)$ also increases. Therefore, for the larger time delay case a higher order approximation is necessary. For $h = 0.25$ a 36th order transfer function is used to approximate C_1 . Figure 4.10 illustrates its frequency response; the resulting error is $\|C_1 - C_{1appr}\|_\infty < 0.13$ for $h = 0.25$.

In the literature, continuous, discrete, and frequency response based approximation techniques for fractional order terms are available. In the remaining parts of this section some of these techniques are reviewed. Next part investigates the FIR representation of a fractance device ($1/s^\alpha$). Then, continuous approximation techniques, namely Matsuda's method and Carlson's method will be performed. As a frequency response data approximation technique MATLAB's `invfreqs` command will be used.

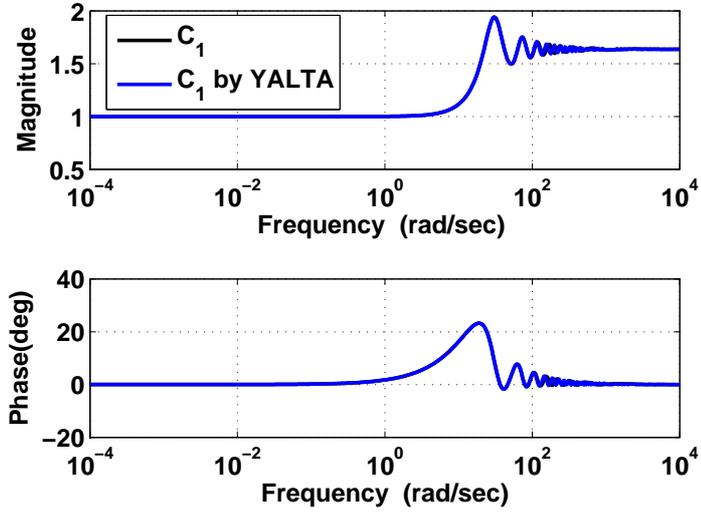


Figure 4.9: Bode plots of C_1 and its approximation by YALTA, for $h = 0.15$.

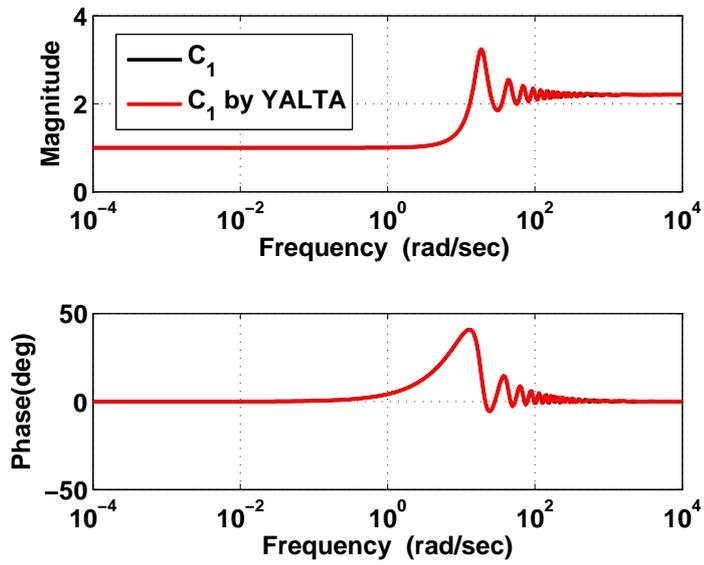


Figure 4.10: Bode plots of C_1 and its approximation by YALTA, for $h = 0.25$.

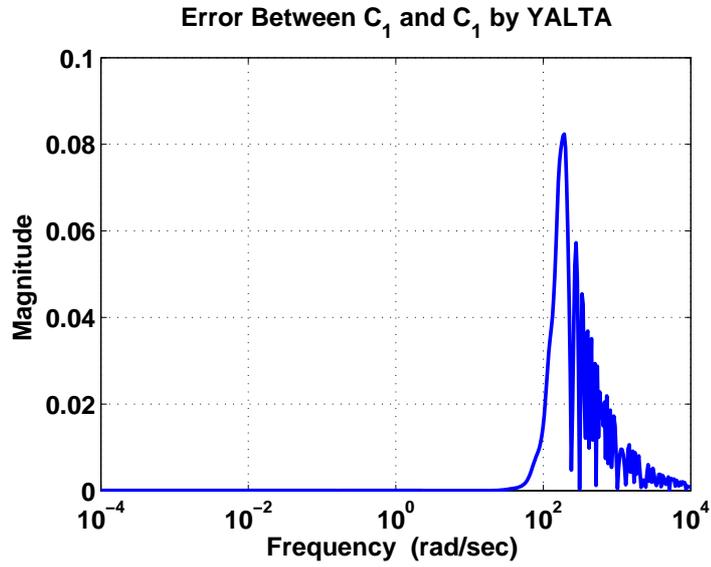


Figure 4.11: Error between frequency responses of C_1 and its approximation by YALTA, for $h = 0.15$.

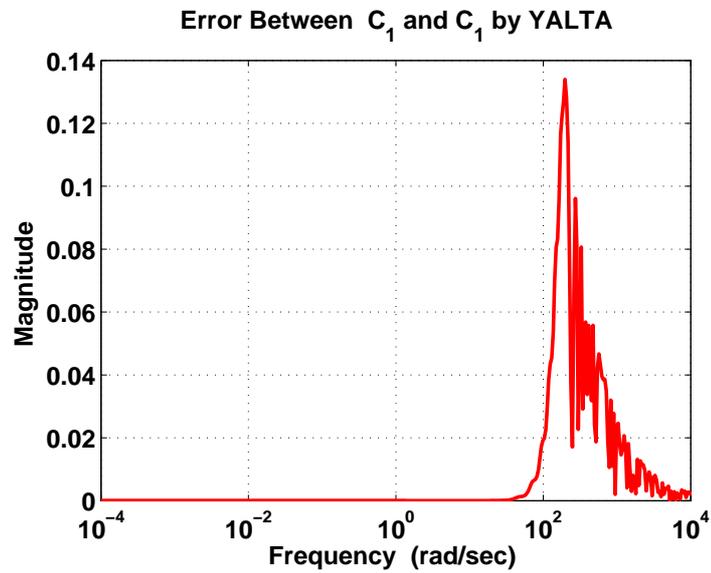


Figure 4.12: Error between frequency responses of C_1 and its approximation by YALTA, for $h = 0.25$.

4.1 FIR Implementation of Fractance Device

In this part FIR implementation of the fractance device ($1/s^\alpha$) where $0 < \alpha < 1$ is obtained through the impulse invariance method. Inverse Laplace transform of the fractance device:

$$\mathcal{L} \left\{ \frac{1}{s^\alpha} \right\}^{-1} = \frac{1}{\sqrt{\pi t}}.$$

Then the corresponding FIR filter, through $z^{-1} = e^{-sT}$ transformation is:

$$H_1(s) = \mathcal{Z} \left\{ K \sqrt{\frac{T}{\pi}} \left[h_o, \frac{1}{n_1}, \frac{1}{n_2}, \dots, \frac{1}{n_N} \right] \right\} \text{ where } n_i = \sqrt{i}$$

K and h_o are constant, $i = 1, 2, \dots, N$;

we take $N = 2000$ for good precision in the approximation, and sampling period $T = 0.01 \text{ sec}$.

In this filter, K and h_o are used to minimize the error between the feedback loops formed by the fractance and its impulse invariant equivalent, [55]. For this purpose we minimize:

$$\|(R(s) - R_1(s)) s^\alpha\|_\infty, \text{ where } R_1(s) = \frac{H_1(s)}{1 + H_1(s)}, \text{ and } R(s) = \frac{1}{s^\alpha + 1}$$

Then the computed values are: $K = 1$ and $h_o = 1.4$. With those values the frequency response of the error between the feedback loops, $E_1(j\omega) := R(j\omega) - R_1(j\omega)$, formed by the fractance and its discrete equivalent is shown in Figure 4.13

It is also possible to obtain continuous time representation of the FIR filter. Define state space representation of the FIR filter:

$$x[n + 1] = Ax[n] + Bu[n]$$

$$y[n] = Cx[n] + Du[n]$$

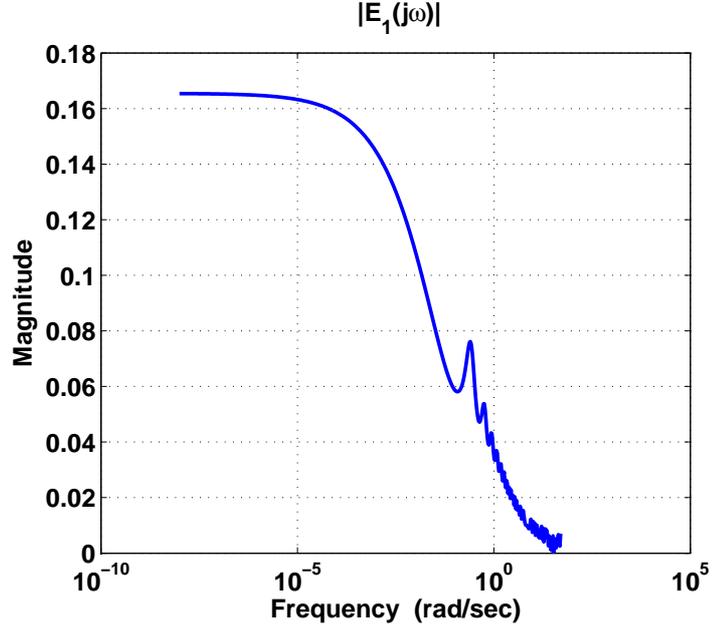


Figure 4.13: Frequency response of the error $E_1(s)$, $N = 2000$.

Where A , B , C , D are:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{N \times N} \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}_{N \times 1}$$

$$C = \sqrt{\frac{T}{\pi}} K \begin{bmatrix} h_N & h_{N-1} & \cdots & h_1 \end{bmatrix}_{1 \times N}$$

$$D = \sqrt{\frac{T}{\pi}} h_0.$$

Define $H_2(s)$ to be:

$$H_2(s) = C_c(I - e^{-h(sI - A_c)})(sI - A_c)^{-1}B_c + D_c,$$

then, $H_2(s)$ is a transfer function with FIR filter behavior. To get A_c , B_c , C_c , D_c from A , B , C , D , bilinear transformation is used:

$$zX = AX + BU$$

$$Y = CX + DU$$

instead of z ; put $\frac{1+sT/2}{1-sT/2}$, then

$$\begin{aligned} A_c &= \frac{2}{T}(A - I)(A + I)^{-1} \\ B_c &= \frac{1}{\sqrt{T}}(B - (A - I)(A + I)^{-1}B) \\ C_c &= \frac{1}{\sqrt{T}}C(A + I)^{-1} \\ D_c &= D - C(A + I)^{-1}B. \end{aligned}$$

Model reduction is required for implementation purposes. For this purpose the technique given in [56] is used. Here, order of the transfer function is reduced from 2000 to 80.

Let $E_2(s) = R(s) - R_2(s)$ where $R_2(s) = H_2(s)/(1 + H_2(s))$. The error $E_2(j\omega)$ is depicted in Figure 4.14.

4.2 Continuous Time Approximation Methods

Fractional order term in the controller, $C_2(s)$, recall (4.3) contains irrational functions of the Laplace variable s . In the previous part a discrete time based approximation method is applied. In this part, continuous time based methods: Matsuda's Method, and Carlson's Method are performed to find rational approximations. Both of the methods are evaluated in such a way that they both produce a transfer function where the degree of the approximate is 15.

Matsuda's method, [41] uses continued fraction expansion, and logarithmically spaced ω values to obtain a rational function approximating an irrational one.

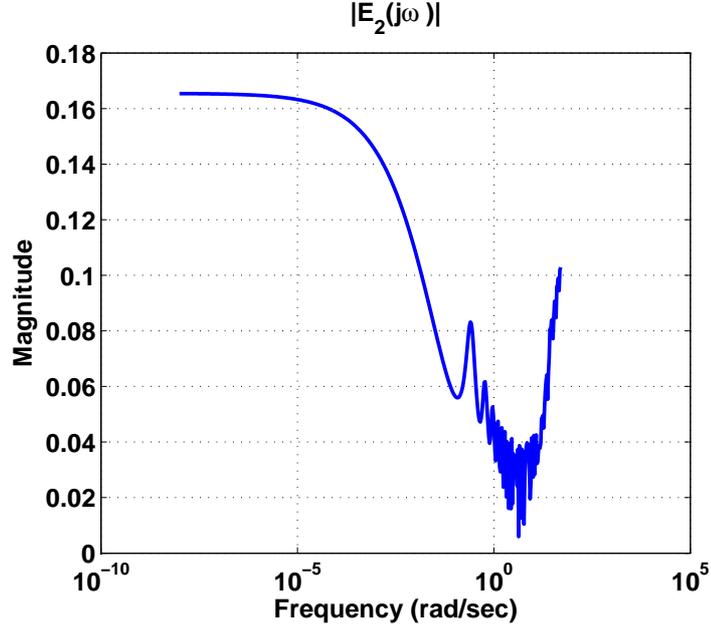


Figure 4.14: Frequency response of the error $E_2(s)$, $N = 80$.

$$\frac{1}{\sqrt{s}} \approx H_3(s) = a_o + \frac{s - \omega_0}{a_1 + \frac{s - \omega_1}{a_2 + \frac{s - \omega_2}{a_3 + \dots}}}$$

with

$$v_o(s) := H(s), \quad a_i = v_i(\omega_i), \quad v_{i+1} = \frac{s - \omega_i}{v_i(s) - a_i}.$$

Here ω_k for $k = 0, 1, 2, \dots$ are logarithmically spaced frequency values.

We performed this method over the interval $\omega \in (10^{-5}, 10^5)$ with a transfer function degree constraint of 15. This gives the third alternative approximation $H_3(s)$. Then we define $E_3(s) = R(s) - R_3(s)$ with $R_3(s) = H_3(s)/(1 + H_3(s))$, Figure 4.15 illustrates the frequency response of $E_3(s)$.

To approximate the fractional order terms (s^α), Carlson's method can also be used [40]. Defining the function $H_4(s) = (G(s))^\alpha$, in this case $G(s) = 1/s$,

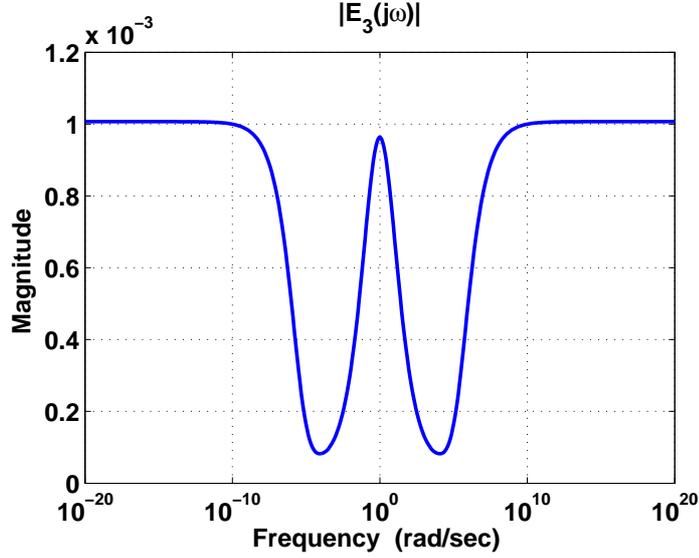


Figure 4.15: Frequency response of the error $E_3(s)$, $N = 15$. (Matsuda)

starting with $F_o(s) = 1$, iteration follows:

$$F_i(s) = F_{i-1}(s) \frac{(q - m)(F_{i-1}(s))^2 + (q + m)G(s)}{(q + m)(F_{i-1}(s))^2 + (q - m)G(s)}, \text{ where } \alpha = 1/q, \quad m = q/2.$$

With this method an approximate transfer function $H_4(s)$ with degree 15 is obtained, Figure 4.16 shows the error $|E_4(j\omega)|$ between the feedback loops, where $E_4(s) = R(s) - R_4(s)$, and $R_4(s) = H_4(s)/(1 + H_4(s))$.

4.3 Frequency Response Based Approximation Method

Other than discrete and continuous time based methods frequency response based methods can also be used to approximate the fractional terms. This part applies MATLAB's built in function `invfreqs` to the to the frequency response data of the actual feedback loop with the weighting function:

$$\widetilde{W}(s) = (1 + s/\tau_1)/(1 + s/\tau_2)^2, \text{ where } \tau_1 = 3 \times 10^{-5} \text{ and } \tau_2 = 1.$$

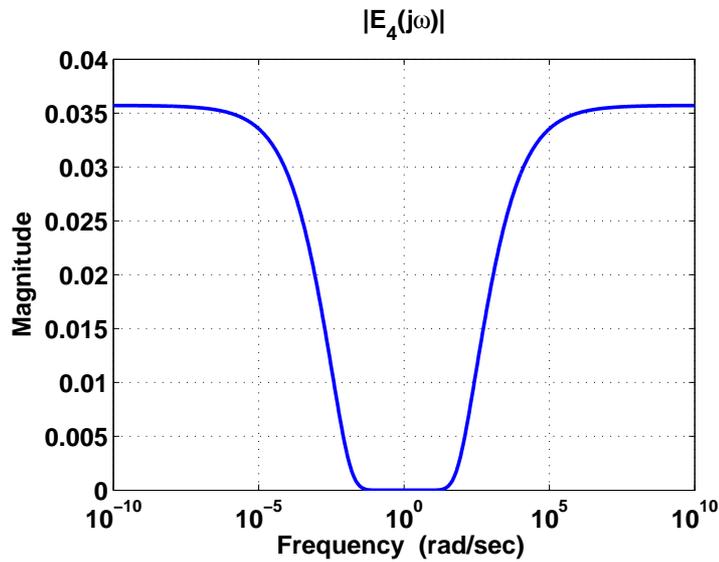


Figure 4.16: Frequency response of the error $E_4(s)$, $N = 15$. (Carlson)

Now consider:

$$\tilde{R}(s) = R(s) \frac{1 + s/\tau_1}{(1 + s/\tau_2)^2} = \frac{1/s^\alpha}{1 + 1/s^\alpha} \frac{1 + s/\tau_1}{(1 + s/\tau_2)^2}$$

and obtain $\tilde{R}(j\omega)$ for ω values in $(10^{-5}, 10^5)$. Now apply `invfreqs`, this gives an approximation $H_5(s)$ of degree 18. Define $E_5(s) = R(s) - H_5(s)$. Magnitude of $E_5(s)$ is given in Figure 4.17

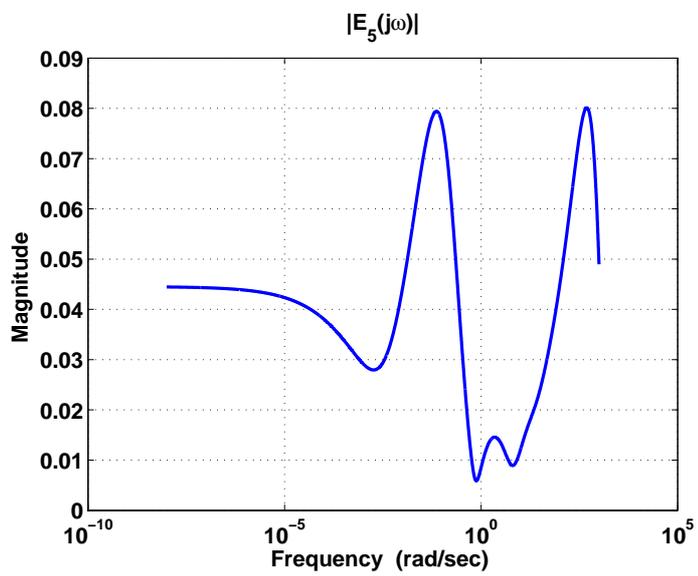


Figure 4.17: Frequency response of the error $E_5(s)$, $N = 18$ (invfreqs)

Chapter 5

Time Domain Simulation Results

In the previous chapters the optimal \mathcal{H}^∞ controller is designed for the non-laminated magnetic suspension system plant model, derived in [2]. Then a suboptimal proper controller (4.1) is obtained. The controller $C_1(s)$ is a retarded time-delay system, with internal pole zero cancellations at $s = p^2$, $s = \pm j/\gamma$. Less complex forms of $C_1(s)$ are investigated through the approximation techniques with different orders. Suboptimal controller, like the plant, includes fractional order terms, expressed in $C_2(s)$. These terms have infinite memory so to simulate the dynamic behavior of the system rational approximations to those terms are necessary. As stated in the previous section, it is enough to find an approximation for the fractional order integrator, since both of the plant and $C_2(s)$ can be realized through integer order transfer functions and the transfer function of the feedback loop formed by the fractional order integrator. In the previous section, discrete, continuous, and frequency response based approximation methods to those terms are presented.

This section illustrates the time domain simulation results for the feedback loop formed by the suboptimal controller and the non-laminated magnetic suspension system plant model by using the presented approximation and implementation techniques. To approximate $C_1(s)$, YALTA toolbox is used, for $h = 0.15$, a 16th degree transfer function, and for $h = 0.25$ a 34th degree approximate is obtained. For the fractional order terms in the plant and in $C_2(s)$, a 17th

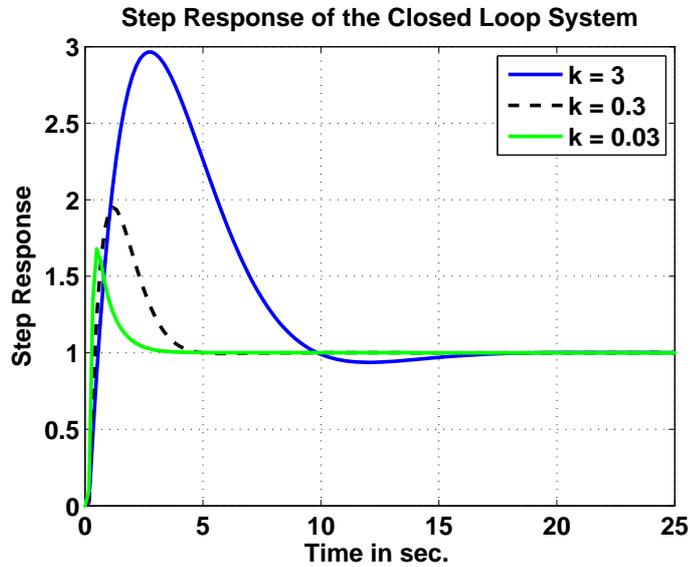


Figure 5.1: Step responses of the closed loop system for $h = 0.15$, without any disturbance.

degree approximation, obtained from the Matsuda's method is used. Simulations are done using MATLAB. To see the effect of multiplicative uncertainty bound on the designed controller and the resulting closed loop, different k values is used. Recall that multiplicative uncertainty bound is given by the function: $W_2(s) = ks$.

The step responses of the closed loop systems are depicted in Figure 5.1 and Figure 5.2 for $h = 0.15$ and for $h = 0.25$. These figures are obtained for the cases: $k = 0.03$, $k = 3$, $k = 0.3$.

The effect of disturbance on the output can be seen through the frequency response of the transfer function from disturbance (d) to output (y), T_{yd} , see Figure 5.3, Figure 5.4. Let (ω_d) denote the frequency value where the magnitude of the frequency response of T_{yd} gets its highest value. So, when a signal in the form $\sin(\omega_d t)$ is applied to the system, the maximum disturbance at the output is observed.

Figure 5.5 and Figure 5.6 show the maximum disturbance that can be observed

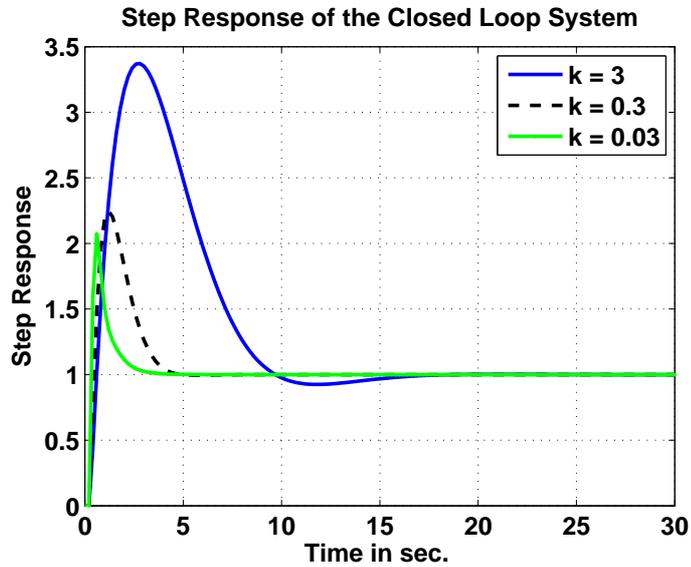


Figure 5.2: Step response of the closed loop system for $h = 0.25$, without any disturbance.

at the output, for the cases $h = 0.15$ and $h = 0.25$. Furthermore, step disturbance is also applied to the system and the resulting output is depicted in Figure 5.7, Figure 5.8.

To see the effect of the multiplicative plant uncertainty bound on the robustness, [57] the graph of $|1/T_{yr}(j\omega)|^{-1}$ versus ω are given in Figure 5.9 and Figure 5.10. It can be observed that for fast step responses and good disturbance attenuation k should be small. However, as k gets small robustness levels to unmodeled high frequency dynamics shrink (especially in the frequency range $2 \text{ rad/sec} \leq \omega \leq 100 \text{ rad/sec}$). So, $k = 0.3$ provides a good balance between performance and robustness to dynamic uncertainty.

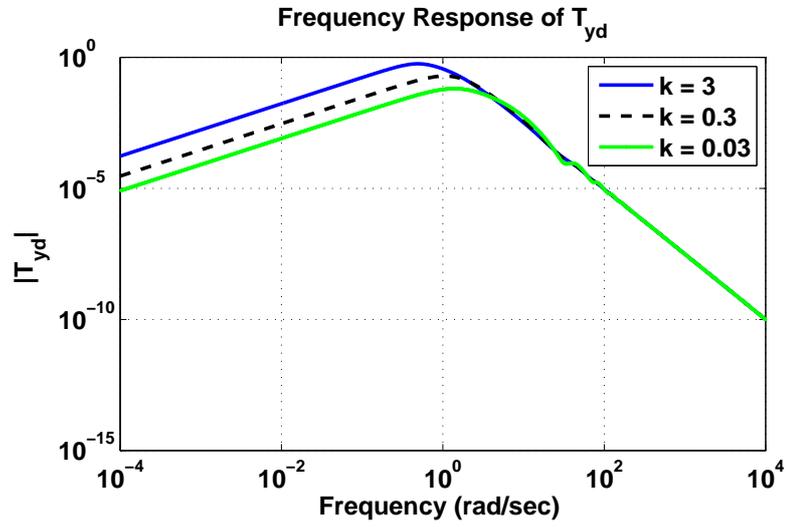


Figure 5.3: $|T_{yd}|$, for $h = 0.15$.

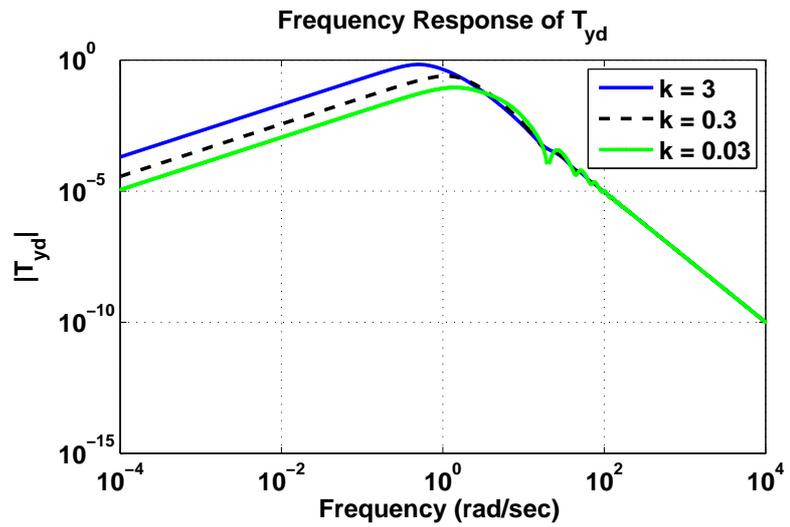


Figure 5.4: $|T_{yd}|$, for $h = 0.25$.

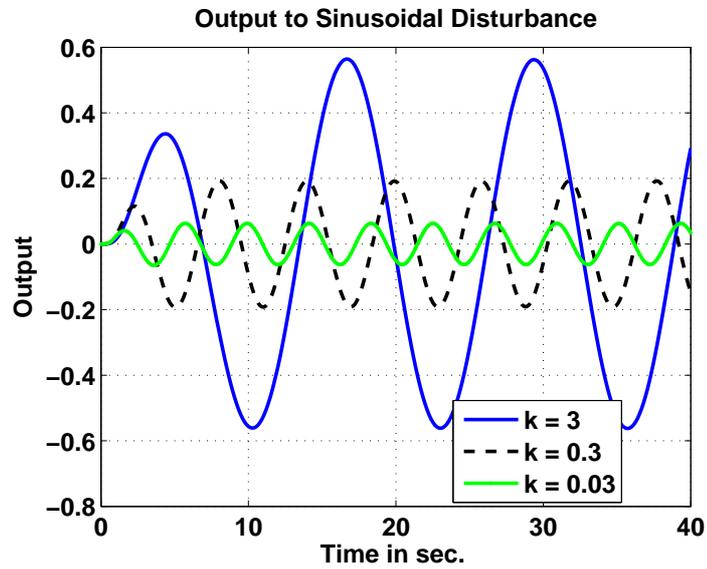


Figure 5.5: Maximum disturbance observed at the output for $h = 0.15$, $d(t) = \sin(\omega_d t)$.

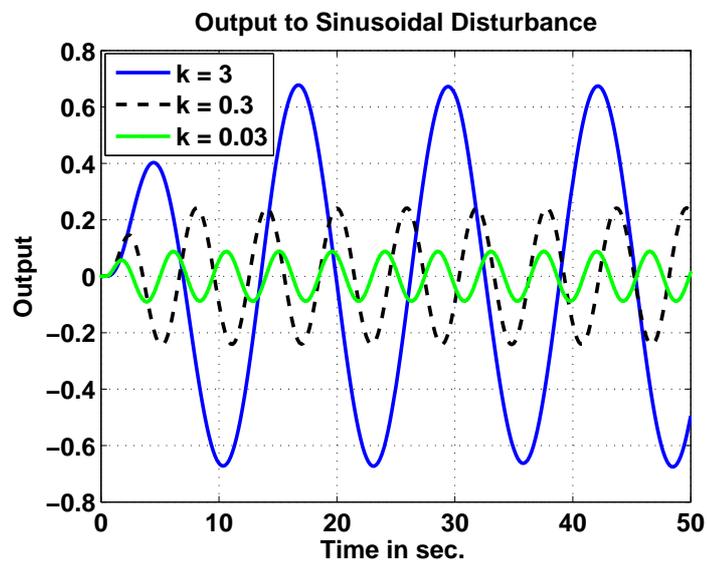


Figure 5.6: Maximum disturbance observed at the output for $h = 0.25$, $d(t) = \sin(\omega_d t)$.

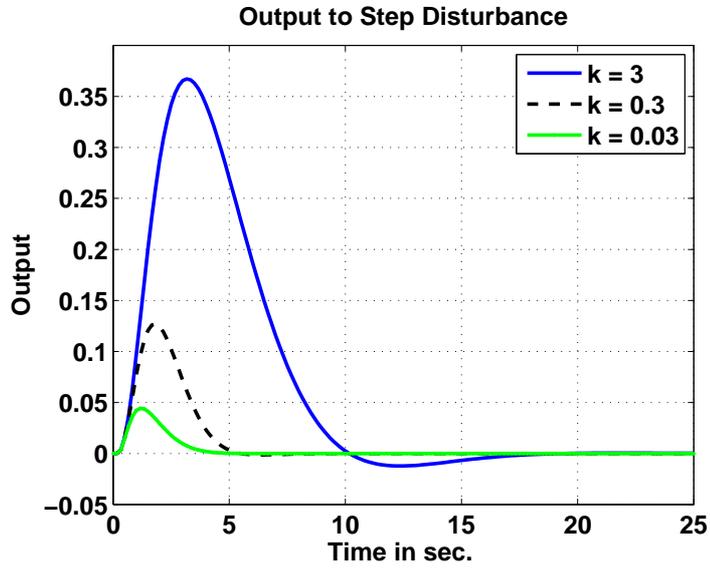


Figure 5.7: Output to step disturbance for $h = 0.15$.

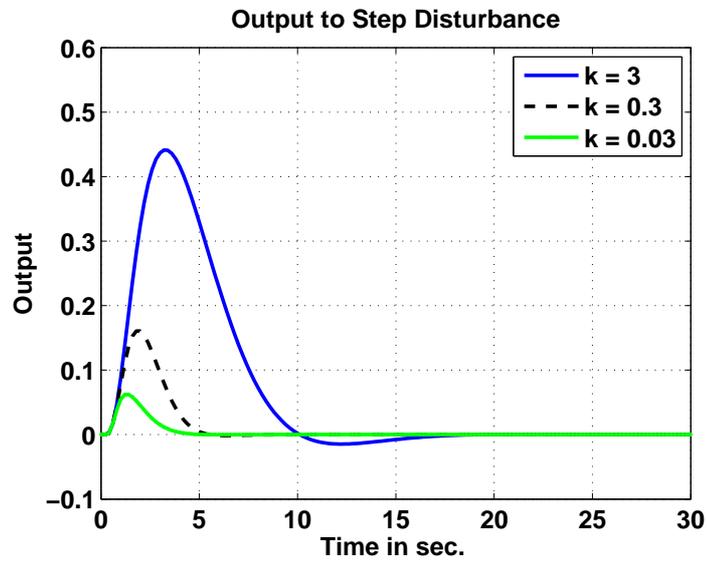


Figure 5.8: Output to step disturbance for $h = 0.25$.

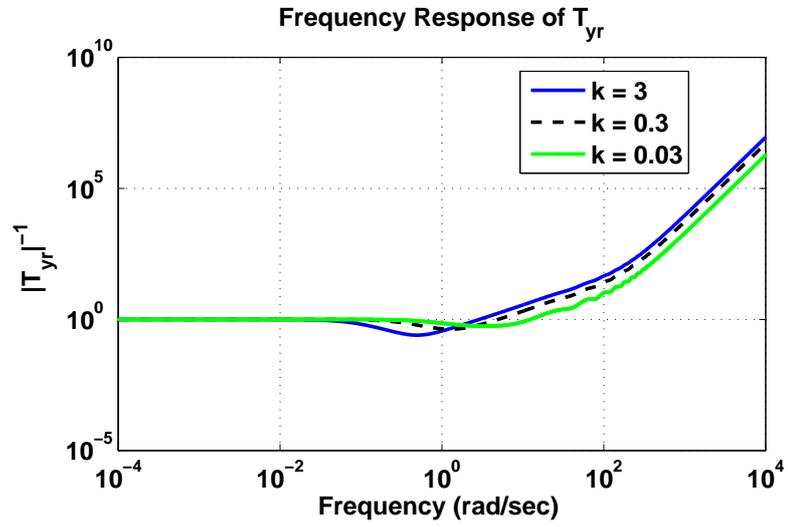


Figure 5.9: $|T_{yr}(s)|^{-1}$, for $h = 0.15$.

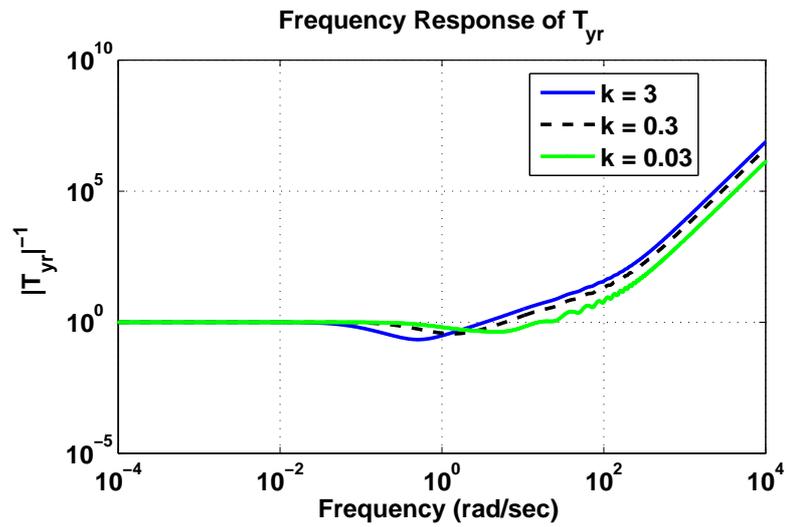


Figure 5.10: $|T_{yr}(s)|^{-1}$, for $h = 0.25$.

Chapter 6

Conclusion

This thesis reviews the recent techniques in the field of fractional order systems. To create insight, definitions of fractional order integral and derivative are presented. Bode's ideal open loop transfer function with an example is given to illustrate the possible benefits of fractional order systems. The mixed sensitivity minimization problem with rational weights for a fractional order system is solved and the \mathcal{H}^∞ optimal controller is obtained. Effect of time delay on the achievable performance level is illustrated. For implementation purposes approximation to the controller is investigated. To simulate the system, integer order approximation techniques are evaluated, and an approximate transfer function for fractional terms is found. Simulation results of the closed loop system, with approximated plant and the controller are presented.

In the last decades it is shown that some natural phenomena can be described better with fractional order differential equations, like viscoelasticity, diffusion. This increased the popularity of fractional order systems. In this thesis, the \mathcal{H}_∞ optimal controller is designed for a fractional order plant, the mathematical model of the non-laminated magnetic suspension system, derived in [2], [46]. Fractional order systems possess infinite memory and are infinite dimensional. For infinite dimensional systems the technique presented in [47] can be used to design the optimal \mathcal{H}_∞ controller. Later in [1], it is shown that when the weights are low order this formula can be simplified. These two methods are applied separately

and it is shown that simplified formula of [1] produces the same results with the [47]. The effect of multiplicative uncertainty bound $W_2(s)$ on the performance level is also investigated. Simulation results of closed loop systems for various time delays and different uncertainty bounds are presented.

Bibliography

- [1] H. Özbay, “Computation of \mathcal{H}_∞ controllers for infinite dimensional plants using numerical linear algebra,” *Numerical Linear Algebra with Applications*, vol. 20, pp. 327–335, 2013.
- [2] C. Knospe and L. Zhu, “Performance limitations of nonlaminated magnetic suspension systems,” *IEEE Trans. on Control Systems Technology*, vol. 19, pp. 327–336, 2011.
- [3] H. Özbay, “ \mathcal{H}_∞ optimal controller design for a class of distributed parameter systems,” *International Journal of Control*, vol. 58, pp. 739–782, 1993.
- [4] P. S. Marquis De Laplace, *A Philosophical Essay on Probabilities*. John Wiley & Sons, 1902.
- [5] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and B. V. Feliu, *Fractional-order systems and controls: fundamentals and applications*. Springer, 2010.
- [6] K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*. John Wiley & Sons, 1993.
- [7] I. Podlubny, *Fractional Differential Equations*. Mathematics in Science and Engineering, Elsevier Science, 1998.
- [8] M. Caputo, “Linear models of dissipation whose q is almost frequency independent,” *Geophysical Journal International*, vol. 13, pp. 529–539, 1967.
- [9] N. Heymans and I. Podlubny, “Physical interpretation of initial conditions for fractional differential equations with riemann-liouville fractional derivatives,” *Rheologica Acta*, vol. 45, pp. 765–771, 2006.

- [10] D. Matignon, “Stability results for fractional differential equations with applications to control processing,” in *Computational Engineering in Systems Applications*, vol. 2, pp. 963–968, 1996.
- [11] R. Malti, M. Aoun, F. Levron, and A. Oustaloup, “Analytical computation of the h_2 norm of fractional commensurate transfer functions,” *Automatica*, vol. 47, pp. 2425–2432, 2011.
- [12] L. Fadiga, C. Farges, J. Sabatier, and M. Moze, “On computation of \mathcal{H}_∞ norm for commensurate fractional order systems,” in *50th IEEE Conference on Decision and Control and 2011 European Control Conference*, pp. 8231–8236, 2011.
- [13] K. Åström, “Limitations on control system performance,” *European Journal of Control*, pp. 2–20, 2000.
- [14] J. C. Doyle, B. A. Francis, and A. Tannenbaum, *Feedback Control Theory*. Macmillan, 1992.
- [15] M. S. Tavazoei, M. Haeri, S. Bolouki, and M. Siami, “Stability preservation analysis for frequency-based methods in numerical simulation of fractional order systems,” *SIAM J. Numerical Analysis*, pp. 321–338, 2008.
- [16] A. Oustaloup, “Complex non-integer derivation in robust control through the crone control,” *Analysis of Controlled Dynamical Systems*, vol. 8, pp. 326–336, 1991.
- [17] I. Podlubny, “Fractional-order systems and $PI^\lambda D^\mu$ controllers,” *IEEE Transactions on Automatic Control*, vol. 44, pp. 208–214, 1999.
- [18] D. J. Wang and X. L. Gao, “Stability margins and \mathcal{H}_∞ co-design with fractional-order pi^λ controllers,” *Asian Journal of Control*, vol. 15, pp. 691–697, 2013.
- [19] R. Malti, M. Aoun, O. Cois, A. Oustaloup, and F. Levron, “ h_2 norm of fractional differential systems,” in *Proceedings of DETC’03 ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2003.

- [20] I. Petráš and M. Hypiusova, “Design of fractional-order controllers via \mathcal{H}_∞ norm minimisation,” in *IFAC Conference, Control Systems Design*, pp. 454–457, 2000.
- [21] H. Özbay, C. Bonnet, and A. R. Fioravanti, “PID controller design for fractional-order systems with time delays,” *Systems & Control Letters*, vol. 61, pp. 18–23, 2012.
- [22] D. Valério and J. Sá da Costa, “Variable order fractional controllers,” *Asian Journal of Control*, vol. 15, pp. 648–657, 2013.
- [23] S. Hamamci, “An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers,” *IEEE Transactions on Automatic Control*, vol. 52, pp. 1964–1969, Oct 2007.
- [24] F. Padula and A. Visioli, “Set-point weight tuning rules for fractional-order PID controllers,” *Asian Journal of Control*, vol. 15, pp. 678–690, 2013.
- [25] S. Domek, “Switched state model predictive control of fractional-order non-linear discrete-time systems,” *Asian Journal of Control*, vol. 15, pp. 658–668, May 2013.
- [26] Y. H. Lan and Y. Zhou, “ D^α type iterative learning control for fractional-order linear time-delay systems,” *Asian Journal of Control*, vol. 15, pp. 669–677, 2013.
- [27] A. R. Fioravanti, C. Bonnet, H. Özbay, and S. I. Niculescu, “Stability windows and unstable root-loci for linear fractional time-delay systems,” in *Proceedings of the 18th IFAC World Congress*, vol. 18, pp. 12532–12537, 2011.
- [28] C. Bonnet, A. R. Fioravanti, and J. R. Partington, “Stability of neutral systems with commensurate delays and poles asymptotic to the imaginary axis,” *SIAM Journal on Control and Optimization*, vol. 49, pp. 498–516, 2011.
- [29] A. Fioravanti, C. Bonnet, and H. Özbay, “Stability of fractional neutral systems with multiple delays and poles asymptotic to the imaginary axis,”

in *49th IEEE Conference on Decision and Control (CDC 2010)*, pp. 31–35, Dec 2010.

- [30] A. R. Fioravanti, C. Bonnet, H. Özbay, and S. I. Niculescu, “A numerical method to find stability windows and unstable poles for linear neutral time-delay systems,” in *9th IFAC Workshop on Time Delay Systems*, vol. 9, pp. 183–188, 2010.
- [31] A. R. Fioravanti, C. Bonnet, H. Özbay, and S. I. Niculescu, “A numerical method for stability windows and unstable root-locus calculation for linear fractional time-delay systems,” *Automatica*, vol. 48, pp. 2824 – 2830, 2012.
- [32] D. Avanesoff, C. Bonnet, A. Fioravanti, and L. Nguyen, “User document yalta.” <https://gforge.inria.fr/projects/yalta-toolbox>, 2013.
- [33] Z. Jiao, Y. Chen, and Y. Zhong, “Stability analysis of linear time-invariant distributed-order systems,” *Asian Journal of Control*, vol. 15, no. 3, pp. 640–647, 2013.
- [34] A. Calderón, B. Vinagre, and V. Feliu, “Linear fractional order control of a dc-dc buck converter,” in *European Control Conference*, 2003.
- [35] V. Pommier Budinger, R. Musset, P. Lanusse, and A. Oustaloup, “Study of two robust controls for an hydraulic actuator,” *European Control Conference*, 2003.
- [36] A. Oustaloup, B. Mathieu, and P. Lanusse, “The crone control of resonant plants: application to a flexible transmission,” *European Journal of Control*, vol. 1, pp. 113–121, 1995.
- [37] N. F. Ferreira and J. T. Machado, “Fractional-order hybrid control of robotic manipulators,” in *Proceedings of the 11th International Conference on Advanced Robotics*, 2003.
- [38] C. Monje, F. Ramos, V. Feliu, and B. Vinagre, “Tip position control of a lightweight flexible manipulator using a fractional order controller,” *IET Control Theory Applications*, vol. 1, pp. 1451–1460, 2007.

- [39] G. Bohannan, “Analog realization of a fractional control element - revisited,” *IEEE CDC 2002 Tutorial Workshop*, 2002.
- [40] G. Carlson and C. Halijak, “Approximation of fractional capacitors $(1/s)^{1/n}$ by a regular newton process,” *IEEE Transactions on Circuit Theory*, pp. 210–213, 1964.
- [41] K. Matsuda and H. Fujii, “ \mathcal{H}_∞ optimized wave-absorbing control analytical and experimental results,” *AIAA Journal of Guidance, Control and Dynamics*, pp. 1146–1153, 1993.
- [42] A. Oustaloup, F. Levron, B. Mathieu, and F. M. Nanot, “Frequency-band complex noninteger differentiator: characterization and synthesis,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 1, pp. 25–39, 2000.
- [43] D. Valerio, “Fractional robust system control,” *Instituto Superior Técnico, Universidade Técnica de Lisboa, Ph.D. Thesis*, 2005.
- [44] Y. Q. Chen and K. Moore, “Discretization schemes for fractional-order differentiators and integrators,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, pp. 363–367, Mar 2002.
- [45] I. Podlubny, I. Petraš, B. M. Vinagre, P. O’leary, and L. Dorčák, “Analogue realizations of fractional order controllers,” *Nonlinear Dynamics*, vol. 29, pp. 281–296, 2002.
- [46] L. Zhu and C. R. Knospe, “Modeling of nonlaminated electromagnetic suspension systems,” *IEEE/ASME Transactions on Mechatronics*, vol. 15, pp. 59–69, 2010.
- [47] O. Toker and H. Özbay, “ \mathcal{H}_∞ optimal and suboptimal controllers for infinite dimensional siso plants,” *IEEE Trans. on Automatic Control*, vol. 40, pp. 751–755, 1995.
- [48] A. E. Karagül and H. Özbay, “On the \mathcal{H}_∞ controller design for a magnetic suspension system model,” in *4th Conference on Control and Optimization with Industrial Applications*, (Borovets, Bulgaria), 2013.

- [49] A. E. Karagül and H. Özbay, “On the h-infinity controller design for a magnetic suspension system model,” in *11th IFAC Workshop on Time-Delay Systems*, (Grenoble, France), pp. 320–324, 2013.
- [50] A. E. Karagül, O. Demir, and H. Özbay, “Computation of optimal \mathcal{H}_∞ controllers and approximations of fractional order systems: a tutorial review,” *Applied and Computational Mathematics*, vol. 12, pp. 261–288, 2013.
- [51] A. E. Karagül and H. Özbay, “Kesirli bir manyetik süspansiyon sistem modeli için \mathcal{H}_∞ denetleç tasarımı,” in *Otomatik Kontrol Türk Milli Komitesi 2012 Ulusal Toplantısı*, (Niğde), 2012.
- [52] H. Özbay, “An operator theoretic approach to robust control of infinite dimensional systems,” *TWMS J. Pure Appl. Math.*, pp. 3–19, 2013.
- [53] F. Padula, S. Alcántara, R. Vilanova, and A. Visioli, “ \mathcal{H}_∞ control of fractional linear systems,” *Automatica*, vol. 49, pp. 2276–2280, 2013.
- [54] B. M. Vinagre, I. Podlubny, A. Hernandez, and V. Feliu, “Some approximations of fractional order operators used in control theory and applications,” *Fractional Calculus and Applied Analysis*, pp. 231–248, 2000.
- [55] L. B. Jackson, “A correction to impulse invariance,” *IEEE Signal Processing Letters*, pp. 273–275, 2000.
- [56] G. Gu, P. Khargonekar, and E. Lee, “Approximation of infinite dimensional systems,” *IEEE Transactions on Automatic Control*, pp. 610–618, 1989.
- [57] H. Özbay, *Introduction to Feedback Control Theory*. CRC Press LLC, 2000.

Appendix A

Code

MATLAB[®] code: plantrootlocuswrtc.m

```
% This m file shows the locations of the poles according  
to c.  
% Plant is obtained in Knospe Zhu, Perf. Limitations  
% Non Laminated Magnetic Suspension ...  
% They model the plant as  $P(s) = e^{-hs}/(s^{2.5}+s^2-c)$ .  
And suggest that  
% this plant has only one unstable pole and it is real  
%  
% In this m file I get locations of the poles according to  
c, in \zeta  
% domain.  
% I also plot stability regions with blue lines, this is  
stability region  
% in \zeta domain  
  
clc  
clear  
c = logspace(-5,5,1e2);  
r = zeros();
```

```

for i = 1 : length(c)
    r(1:5,i) = roots([1 1 0 0 0 -c(i)]);
end
a = linspace(0,10, 2);
stabilityline = a + a*1i;% for alpha = 0.5;
stabilitylinecnj = a - a*1i;

plot(real(r(1,:)),imag(r(1,:)), 'ro')
hold on
plot(real(r(2,:)),imag(r(2,:)), 'ro')
hold on
plot(real(r(3,:)),imag(r(3,:)), 'go')
hold on
plot(real(r(4,:)),imag(r(4,:)), 'go')
hold on
plot(real(r(5,:)),imag(r(5,:)), 'k+')
hold on
plot(real(stabilityline),imag(stabilityline))
hold on
plot(real(stabilitylinecnj),imag(stabilitylinecnj))
title('Locations of the poles of the plant in \zeta-domain
      for  $10^{-5} < c < 10^5$ ')
xlabel('\Re')
ylabel('\Im')

```

MATLAB[®] code: Eqn18_computuation_Hinf_ozbay.m

%A.Erdem Karagl

*%This code computes the minimum singularity points of the
parametrized*

%matrix in equation 18, in the paper zbay, H. (2011),

*%Computation of H_{∞} controllers for infinite dimensional
plants using numerical linear algebra. Numer. Linear
Algebra Appl.. doi: 10.1002/nla.1809*

```

%After computing the minimum singularity points gamma opt
    is chosen and
%related interpolation condititons are found. This code
    uses the formulas
%(7) to (18) in the given paper.

%h=0.1,c=5,k=0.3
clc
clear
gamma = linspace(0.4,0.41,100000); % creating the gamma
    vector
%vectors in the eqn 18
beta = 1i./gamma;
k = 0.03; %  $W2(s)=ks$ 
p = 1.2244;%for the plant in [4] for c=5;
psqr = p^2;
h = 0.15;%delay
Mnbeta = zeros(1,length(gamma));
Fgammabeta = zeros(1,length(gamma));
MnAd = zeros(1,length(gamma));
FgammaAd = zeros(1,length(gamma));
minsvdplus = zeros(1,length(gamma));
minsvdminus = zeros(1,length(gamma));
e = 3e-7;%numerical error this should be zero theoritacly

% for loop calculating the Mn(beta) and Fgamma(beta) to
    use in eqn 18
for ii=1:length(gamma)

    Mnbeta(ii)=exp(-h * beta(ii));

    Fgammabeta(ii)= (-gamma(ii) * beta(ii)) / ( k*beta(ii)
        ^2 + sqrt(2*k-(k^2)/gamma(ii)^2) * beta(ii) + 1);

```

end

*% for loop calculating the Mn(psqr) and Fgamma(psqr) to
use in eqn 18*

for iii=1:length(gamma)

MnAd(iii) = **exp**(-h * psqr);

FgammaAd(iii) = (-gamma(iii)* psqr) / (k*psqr^2 +
sqrt(2*k-(k^2)/gamma(iii)^2) * psqr +1);

end

*%for loop calculating the matrix eqn in 18 and finding
gamma opt by using*

*%if statement, if the min singularity value gets smaller
than numerical*

*%error value, computer stores the gamma optimum value as
gammaopt and gets*

*%the matrix at that gamma value and stores it as desired
Ry.*

for i = 1 : length(gamma)

Ryplus = [1 psqr;1 **beta**(i)]+[MnAd(i)*FgammaAd(i) 0; 0
Mnbeta(i)*Fgammabeta(i)]*[1 psqr;1 **beta**(i)]*[1 0;0
-1];

Ryminus = [1 psqr;1 **beta**(i)]-[MnAd(i)*FgammaAd(i) 0; 0
Mnbeta(i)*Fgammabeta(i)]*[1 psqr;1 **beta**(i)]*[1 0;0
-1];

minsvdplus(i) = **min**(**svd**((Ryplus)));*%%%%% abs olamal
m ?*

minsvdminus(i) = **min**(**svd**((Ryminus)));

```

    if e > minsvdminus(i)
        gammaopt = gamma(i)
        indice = i;
        desiredRy = Ryminus
    end
end

%plotting minimum singularity values versus gamma for the
both + and - sign
%in eqn 18
%plot(gamma, minsvdplus);
hold on
plot(gamma, minsvdminus);

% by using 17, reaching the values of interpolation
conditions
[A,B,C] = svd(desiredRy);
soln = C(:,end);
disp('solution of the system: ');
disp(soln)
disp('DesiredRy * x vector: ');
disp(norm(desiredRy*soln,2));
psi20 = soln(1,1);
psi21 = -soln(2,1);
psi2 = [psi20;psi21]

%to compute eqn 15
betaatopt = 1i/gammaopt;
kgammaopt = sqrt(2*k - (k^2 / gammaopt^2 ) );

Fgammaoptpsqr = (-gammaopt*psqr)/(k*psqr^2 + kgammaopt*
    psqr + 1);

```

```

Fgammaoptbeta = (-gammaopt*betaatopt)/(k*betaatopt^2 +
    kgammaopt*betaatopt + 1);
Mnbetaatopt = exp(-h*betaatopt);
%eqn 15
sign = -((([1 0]*psi2)/([1 0]*([1 psqr;1 betaatopt]^ -1)*([
    MnAd(1)*Fgammaoptpsqr 0;0 Mnbetaatopt*Fgammaoptbeta]))
    *[1 psqr;1 betaatopt] *psi2));
disp('result of equation 15:');
disp(sign); %determining sign of L(s) by eqn 15
%L= - (psi20 + psi21*s)/(psi20 - psi21*s);

```

MATLAB[®] code: Eqn_30_computation_Hinf_ozbay.m

```

% Equation 30 from Ozbay 2011 to compute gammaopt and L
%plots (30)

```

```

clc
clear
gamma=linspace(0.001,20,100000);
beta=1i./gamma;
b=-1./beta;
k=0.3;%%%
p=1.2244;
psqr=p^2;
h=1;
Mnbeta=zeros(1,length(gamma));
Fgammabeta=zeros(1,length(gamma));
MnAd=zeros(1,length(gamma));
FgammaAd=zeros(1,length(gamma));
Xgammaplus=zeros(1,length(gamma));
Xgammaminus=zeros(1,length(gamma));

for ii=1:length(gamma)
    Mnbeta(ii)=exp(-h*beta(ii));

```

```

    Fgammabeta(ii) = (-gamma(ii)*beta(ii)) / ( k*beta(ii)^2
        + sqrt(2*k-(k^2)/gamma(ii)^2)*beta(ii) + 1);
end

for iii = 1:length(gamma)
    MnAd(iii) = exp(-h*psqr);
    FgammaAd(iii) = (-gamma(iii)*psqr) / ( k*psqr^2 + sqrt
        (2*k-(k^2)/gamma(iii)^2)*psqr + 1);
end

for i = 1 : length(gamma)
    Xgammaplus(i) = b(i)*(1+Mnbeta(i)*Fgammabeta(i))*((1+
        MnAd(i)*FgammaAd(i))^-1)*(1+MnAd(i)*FgammaAd(i)
        *(-1))*(+psqr)+(1+Mnbeta(i)*Fgammabeta(i)*(-1));
    Xgammaminus(i) = b(i)*(1-Mnbeta(i)*Fgammabeta(i))*((1-
        MnAd(i)*FgammaAd(i))^-1)*(1-MnAd(i)*FgammaAd(i)
        *(-1))*(psqr)+(1-Mnbeta(i)*Fgammabeta(i)*(-1));
end
%plot(gamma, (abs(Xgammaplus)));

figure
plot(gamma, (abs(Xgammaminus)), 'g—');
grid on

% gammaopt=1.3634;
% kgammaopt=sqrt(2*k - (k^2 / gammaopt^2) );
% betaatopt=1i/gammaopt;
% Fgammaoptpsqr=(-gammaopt*psqr)/(k*psqr^2 + kgammaopt*
    psqr + 1);
% Fgammaoptbeta=(-gammaopt*betaatopt)/(k*betaatopt^2 +
    kgammaopt*betaatopt + 1);
% Mnbetaatopt=exp(-h*betaatopt);

```

```

% Ryplus=[1 psqr;1 betaatopt]+[MnAd(1)*Fgammaoptpsqr 0;0
    Mnbetaatopt*Fgammaoptbeta]*[1 psqr; 1 betaatopt]*[1 0;0
    -1];
% Ryminus=[1 psqr;1 betaatopt]-[MnAd(1)*Fgammaoptpsqr 0;0
    Mnbetaatopt*Fgammaoptbeta]*[1 psqr; 1 betaatopt]*[1 0;0
    -1]

```

MATLAB[®] code: Controllerdelay015c5.m

```

% A. Erdem Karagl
% Compute optimal and suboptimal controllers and plot
    robustness performance graphs
% h=0.25,k=0.3,c=10
% clc
% clear
%eps = linspace(1e-4,1e-2,20);

eps = linspace(1e-4,1e-2,20);
w=logspace(-4,4,1e4);
s=1i*w;
gamma=1.462653626536265;%1.6047;%2.088375883758838;%1.6047
    h=0.1;
k=0.3;
roo=[1 1 0 0 0 -5];
pls=roots(roo);
h=0.15;%0.2;%0.1
p1 = pls(1);
p1cnj =pls(2);
p2 = pls(3);
p2cnj =pls(4);
p = pls(5);
a=0.945622101602222;%0.9442;%0.959340025428322;%0.9442;%h
    =0.1 i in , (changing h, changes those values do not
    forget!!!)

```

```

b=0.325267337679942;%0.282252928436864;%0.3295;%h=0.1  i i n
ka=sqrt(2*k-(k^2)/(gamma^2));
Egamma=zeros(1,length(w));
Ggamma=zeros(1,length(w));
Fgamma=zeros(1,length(w));
Mn=zeros(1,length(w));
OneoverNo=zeros(1,length(w));
Md=zeros(1,length(w));
Copt=zeros(1,length(w));
L=zeros(1,length(w));

P=zeros(1,length(w));
%SS=zeros(length(eps),length(w));
W1=zeros(1,length(w));
%TT=zeros(length(eps),length(w));
W2 = zeros(1,length(w));
%W2T = zeros(length(eps),length(w));
%W1S = zeros(length(eps),length(w));
SS=zeros(1,length(w));
TT=zeros(1,length(w));
W2T=zeros(1,length(w));
W1S = zeros(1,length(w));
psi = zeros(length(eps),length(w));
psi_sup = zeros(1,length(eps));
lpf = zeros(1,length(w));
CoptwithLpf = zeros(1,length(w));
%factorized plant but P=(s^(0.5)-p)/G(s^0.5)
for l = 1:length(w)
    Md(l)=(s(l)-(p^2))/(s(l)+(p^2));
    OneoverNo(l)=1/((sqrt(s(l))+p)/((s(l)+p^2)*(sqrt(s(l))
        -p1)*(sqrt(s(l))-p1cnj)*(sqrt(s(l))-p2)*(sqrt(s(l))
        -p2cnj)));

```

```

Mn(1)=exp(-h*s(1));
L(1)=(a*s(1)+b)/(a*s(1)-b);
end

for l2=1:length(w)
    Fgamma(l2)=( (-s(l2)*gamma) / ( ((s(l2)^2)*k) + ( s(l2)
        )*ka ) + 1 ) );
end

for l1=1:length(w)
    Egamma(l1)=( 1+ ((gamma^2)*(s(l1)^2)) ) / ( -(gamma^2)*(
        s(l1)^2) );
end

for r=1:length(w)
    Copt(r)=(Egamma(r)*Md(r)*Fgamma(r)*OneoverNo(r)*L(r))
        / ( 1+Mn(r)*Fgamma(r)*L(r) ) ;
end

%plant as total
for l3=1:length(w)
    P(l3)=Mn(l3)/(OneoverNo(l3)*Md(l3));
end

%weight functions 1/s and ks
for l5 = 1 :length(w)
    W2(l5) = k*s(l5);
    W1(l5) = 1/s(l5);
end

% for eps_i = 1 : length(eps)
%     for l4 = 1 : length(w)

```

```

%      SS(eps_i , l4) = 1/(1+P(l4)*Copt(l4)*(1/(eps(eps_i)*s(
      l4) + 1)^3));%l=3
%      TT(eps_i , l4) = 1-SS(eps_i , l4);
%      W1S(eps_i , l4) = SS(eps_i , l4)*W1(l4);
%      W2T(eps_i , l4) = TT(eps_i , l4) * W2(l4);
%      psi(eps_i , l4) = sqrt(abs(W1S(eps_i , l4))^2 + abs(W2T(
      eps_i , l4))^2);
%      end
%      psi_sup(eps_i) = max((psi(eps_i , :)));
% end
%add 1/((es+1)^3) to controller as lpf e = 0.001;
epso=0.005;%
for l9 = 1 : length(w)
    lpf(l9) = 1 /((epso*s(l9)+1)^2);%l=2
end

for l9 = 1 : length(w)
    CoptwithLpf(l9) = Copt(l9)*lpf(l9);
end

for l4 = 1 : length(w)
    SS(l4) = 1/(1+P(l4)*CoptwithLpf(l4));%l=2
    TT(l4) = 1-SS(l4);
    W1S(l4) = SS(l4)*W1(l4);
    W2T(l4) = TT(l4) * W2(l4);
    %psi(l4) = sqrt(abs(W1S(eps_i , l4))^2 + abs(W2T(eps_i ,
      l4))^2);
end

% %plot of eps vs psi sup
% figure(2)
% semilogy(eps , psi_sup)
%%

```

```

% bodes of Copt
Coptabs=abs(Copt);
Coptangle=angle(Copt);
figure(1);
hold on
subplot(2,1,1);
hold on
semilogx((w),20*log10(Coptabs));
grid on
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude(dB) ');
hold on
subplot(2,1,2);
semilogx((w),(Coptangle)*180/pi);
grid on
xlabel('Frequency (rad/sec) ');
ylabel('Phase(deg) ');

% bodes of CoptwithLpf
CoptwithLpfabs=abs(CoptwithLpf);
CoptwithLpfangle=angle(CoptwithLpf);
figure(2);
hold on
subplot(2,1,1);
hold on
semilogx((w),20*log10(CoptwithLpfabs));
grid on
xlabel('Frequency ');
ylabel('Magnitude(dB) ');
hold on
subplot(2,1,2);
semilogx((w),(CoptwithLpfangle)*180/pi);
grid on

```

```
xlabel('Frequency (rad/sec)');  
ylabel('Phase(deg)');
```

```
% eps_o=707;
```

```
figure(3);  
hold on  
loglog(w,(1/gamma)*(abs(W1S)));  
grid on;  
xlabel('Frequency (rad/sec)');  
ylabel('Magnitude');  
axis([1e-4 1e4 1e-5 1e1])
```

```
figure(4);  
hold on  
loglog(w,(1/gamma)*(abs(W2T)));  
grid on;  
xlabel('Frequency (rad/sec)');  
ylabel('Magnitude');  
axis([1e-4 1e4 1e-5 1e1])
```

```
figure(5)  
hold on  
semilogx(w,sqrt(abs(W1S).^2+abs(W2T).^2))  
grid on;  
title('sqrt(|W1S|^2 + |W2T|^2)')  
xlabel('Frequency (rad/sec)');  
ylabel('Magnitude');
```

MATLAB[®] code: controllerapproximation015delaysecondthird.m

```
clc  
clear
```

```

w=logspace(-4,4,800);
s=1i*w;
Capprgren02 = zeros(1,length(w));
epso=0.005;%
lpf=zeros(1,length(w));
gamma=1.462653626536265;%1.6047;%2.088375883758838;%1.6047
    h=0.1;
k=0.3;
roo=[1 1 0 0 0 -5];
pls=roots(roo);
h=0.15;%0.2;%0.1
p1 = pls(1);
p1cnj =pls(2);
p2 = pls(3);
p2cnj =pls(4);
p = pls(5);
a=-0.945622101602222;%0.9442;%0.959340025428322;%0.9442;%h
    =0.1 i in , (changing h, changes those values do not
    forget!!!)
b=-0.325267337679942;0.282252928436864;%0.3295;%h=0.1 i in
ab=a/b;
ax=0;

%ka=sqrt(2*k-(k^2)/(gamma^2));

c4=abs(p1)^2*abs(p2)^2;
c1=-(p1+p1cnj+p2+p2cnj);
c2=abs(p1)^2+abs(p2)^2+(p1cnj+p1)*(p2cnj+p2);
c3=-((p1cnj+p1)*abs(p2)^2+(p2cnj+p2)*abs(p1)^2);
x3=c1/c4;
x2=c2/c4;
x1=c3/c4;
cnstnt=(k*ab*p^2)/(gamma^2);

```

```

ka=sqrt(2*k-(k^2)/(gamma^2));

for l9 = 1 : length(w)
    lpf(l9) = 1 / ( (epso*s(l9)+1)^2 );%l=2
end

for kk = 1:length(w)
    %Capprgren02(kk) = (5/gamma)*(1/s(kk))*(1+s(kk)*ab)
        *((1+h/2*s(kk))/(1+h/2*cnstnt*s(kk)))*( (s(kk)^2/c4
        ) + (x3*s(kk)^1.5) + (x2*s(kk)) + (x1*sqrt(s(kk)))
        +1)/(1+sqrt(s(kk))/p)*lpf(kk);
    %Happr3(kk)=(1+(h/2)*s(kk)+((h^2)/12)*s(kk)^2)/(1+(h
        /2)*sqrt(cnstnt)*s(kk)+((h^2)/12)*cnstnt*s(kk)^2);
    Happr2(kk)=(1+(h/2)*s(kk)+((h^2)/12)*s(kk)^2)/(1+(ax+(
        h/2)*(cnstnt))*s(kk)+((h^2)/12)*cnstnt*s(kk)^2);
    Happr3(kk)=(1 + (h/2)*s(kk) + ((h^2)/10)*s(kk)^2 + ((h
        ^3)/120)*s(kk)^3 )/( 1 + (h/2)*cnstnt*s(kk) + ((h
        ^2)/10)*cnstnt*s(kk)^2 + ((h^3)/120)*cnstnt*s(kk)^3
        );
    %Happr2(l)=(1+(h/2)*s(l)+((h^2)/12)*s(l)^2)/(1+(h/2)*
        cnstnt*s(l)+((h^2)/12)*cnstnt*s(l)^2);
    Happr(kk)=(1+(h/2)*s(kk))/(1+(h/2)*cnstnt*s(kk));
    Capprgren02(kk)=(5/gamma)*(1/s(kk))*(1+s(kk)*ab)*
        Happr3(kk)*( (s(kk)^2/c4) + (x3*s(kk)^1.5) + (x2*s(
        kk)) + (x1*sqrt(s(kk)))+1)/(1+sqrt(s(kk))/p)*lpf(kk
        );
    %Capprgren02(k) = 4.7863*(1/s(k))*(1+s(k)*1.222)*(1+s(
        k)*1.722)*(1+s(k)*0.3)^0.5*lpf(k);%4.7863
    %Capprgren02(k) =
    %4.7863*(1/s(k))*(1+s(k)*1.5)^2*(1+s(k)*0.3)^0.5*lpf(k
        );%4.7863%hitay
    %hoca en son g r d
    %1.2, 1.8 seems to be fine

```

end

%%

%%%

%h=0.1, k=0.3, c=10

% clc

% clear

%eps = linspace(1e-4,1e-2,20);

Egamma=zeros(1,length(w));

Ggamma=zeros(1,length(w));

Fgamma=zeros(1,length(w));

Mn=zeros(1,length(w));

OneoverNo=zeros(1,length(w));

Md=zeros(1,length(w));

Copt=zeros(1,length(w));

L=zeros(1,length(w));

P=zeros(1,length(w));

%SS=zeros(length(eps),length(w));

Wl=zeros(1,length(w));

%TT=zeros(length(eps),length(w));

W2 = zeros(1,length(w));

%W2T = zeros(length(eps),length(w));

%W1S = zeros(length(eps),length(w));

SS=zeros(1,length(w));

TT=zeros(1,length(w));

W2T=zeros(1,length(w));

W1S = zeros(1,length(w));

psi = zeros(length(eps),length(w));

psi_sup = zeros(1,length(eps));

```

lpf = zeros(1,length(w));
CoptwithLpf = zeros(1,length(w));
H=zeros(1,length(w));
Happr1=zeros(1,length(w));
Happr2=zeros(1,length(w));
Herr1=zeros(1,length(w));
Herr2=zeros(1,length(w));
%factorized plant but P=(s^(0.5)-p)/G(s^0.5)
for l = 1:length(w)
    Md(l)=(s(l)-(p^2))/(s(l)+(p^2));
    OneoverNo(l)=1/((sqrt(s(l))+p)/((s(l)+p^2)*(sqrt(s(l))
        -p1)*(sqrt(s(l))-p1cnj)*(sqrt(s(l))-p2)*(sqrt(s(l))
        -p2cnj)));
    Mn(l)=exp(-h*s(l));
    L(l)=(a*s(l)+b)/(a*s(l)-b);
end

%plant as total
for l3=1:length(w)
    P(l3)=Mn(l3)/(OneoverNo(l3)*Md(l3));
end

%weight functions 1/s and ks
for l5 = 1 :length(w)
    W2(l5) = k*s(l5);
    W1(l5) = 1/s(l5);
end

```

```

for l = 1 : length(w)
    H(l)=1/(((1-ab*s(l))*((s(l)^2)*k) + (s(l)*ka) + 1
        )+(gamma*s(l)*(1+ab*s(l))*exp(-h*s(l))))/(1+ ((
        gamma^2)*(s(l)^2))*(1-s(l)/(p^2)));
    Happr1(l)=(1+(h/2)*s(l))/(1+(h/2)*cnstnt*s(l));
    Happr2(l)=(1+(h/2)*s(l)+((h^2)/12)*s(l)^2)/(1+(ax+(h
        /2)*cnstnt)*s(l)+((h^2)/12)*cnstnt*s(l)^2);
    %Happr3(l)=(1+(h/2)*s(l)+((h^2)/12)*s(l)^2)/(1+(h/2)*
        sqrt(cnstnt)*s(l)+((h^2)/12)*cnstnt*s(l)^2);
    Happr3(l)=(1+(h/2)*s(l)+((h^2)/10)*s(l)^2+((h^3)
        /120)*s(l)^3)/(1+(h/2)*s(l)*cnstnt+((h^2)/10)
        *cnstnt*s(l)^2+((h^3)/120)*cnstnt*s(l)^3);
    Herr2(l)=abs(H(l)-Happr2(l));
    Herr1(l)=abs(H(l)-Happr1(l));
    Herr3(l)=abs(H(l)-Happr3(l));
end

for l2=1:length(w)
    Fgamma(l2)=(-s(l2)*gamma / ((s(l2)^2)*k) + (s(l2)
        )*ka) + 1);
end

for l1=1:length(w)
    Egamma(l1)=(1+((gamma^2)*(s(l1)^2))) / (-(gamma^2)*(
        s(l1)^2));
end
for r=1:length(w)
    Copt(r)=(Egamma(r)*Md(r)*Fgamma(r)*OneoverNo(r)*L(r))
        /(1+Mn(r)*Fgamma(r)*L(r));
end
%epso=0.005;%
for l9 = 1 : length(w)
    lpf(l9) = 1 / (epso*s(l9)+1)^2;%l=2

```

```

end

for l9 = 1 : length(w)
    CoptwithLpf(l9) = Copt(l9)*lpf(l9);
end
Coptabs=abs(CoptwithLpf);
Coptangle=angle(CoptwithLpf);

figure(1);
subplot(2,1,1);
semilogx((w),20*log10(Coptabs),'k',(w),20*log10(abs(
    Capprgren02)),'b—');
grid on
title('Bode Plots of C_{subopt} and C_{appr3}, h=0.15')
xlabel('Frequency (rad/sec)');
ylabel('Magnitude(dB)');
subplot(2,1,2);
semilogx((w),(Coptangle)*180/pi,'k',(w),angle(Capprgren02)
    *180/pi,'b—');
grid on
xlabel('Frequency (rad/sec)');
ylabel('Phase(deg)');

% eps_o=707;

% figure(2);
% loglog(w,(1/gamma)*(abs(WIS)),'r');
% grid on;
% xlabel('Frequency (rad/sec)');
% ylabel('Magnitude');
% axis([1e-4 1e4 1e-5 1e1])

```

```

% figure (3);
% loglog (w, (1/gamma)*(abs (W2T)), 'r ');
% grid on;
% xlabel('Frequency (rad/sec) ');
% ylabel('Magnitude ');
% axis ([1e-4 1e4 1e-5 1e1])

% clc
% close all
% gammaappr=(1+epso)*gamma;%2.098
% gammagraph=psi1 (1);%2.088
% disp (gammaappr)
% disp (gammagraph)

Hoptabs=abs(H);
Hoptangle=angle(H);
Happrabs=abs(Happr3);%by adjusting Happr you can see H vs
    Happr
Happrangle=angle(Happr3);

figure (5);
%hold on
%subplot (2,1,1);
%hold on
%semilogx ((w), (Hoptabs), (w), (Happrabs), 'k ');
semilogx ((w), (Herr2), 'r ', (w), (Herr1), 'k ', (w), (Herr3), 'b ');
grid on
title ('Approximation errors (black: first, red: second,
    blue: third order) , h=0.2')
xlabel ('Frequency (rad/sec) ');
ylabel ('Magnitude ');
axis ([1 1e4 0 1.6])

```

```

% %hold on
% subplot(2,1,2);
% semilogx((w),(Hoptangle)*180/pi,(w),(Happrangle)*180/pi
    , 'k');
% grid on
% xlabel('Frequency (rad/sec)');
% ylabel('Phase(deg)');
for l4 = 1 : length(w)
    SS(l4) = 1/(1+P(l4)*Capprgren02(l4));%l=2
    TT(l4) = 1-SS(l4);
    W1S(l4) = SS(l4)*W1(l4);
    W2T(l4) = TT(l4) * W2(l4);
    %psi(l4) = sqrt(abs(W1S(eps_i,l4))^2 + abs(W2T(eps_i,
        l4))^2);
end
figure(4)
psi1=sqrt(abs(W1S).^2+abs(W2T).^2);
semilogx(w,psi1,'r')
grid on;
title('sqrt(|W1S|^2 + |W2T|^2) with C_{appr3} h=0.2')
xlabel('Frequency (rad/sec)');
ylabel('Magnitude');

```

MATLAB[®] code: bode_openloop_and_nyquist_nominal_c5_h015and025_k03.m

```

% plot bode and nyquist for stability nominal plant and
    the controller

```

```

clc
clear
close all
om = logspace(-2,2,1e5);
PC = zeros(1,length(om));
eps = 0.005;%0.002 for lower order appr

```

```

c=5;
gamma=1.462653626536265;%1.6047;%2.088375883758838;%1.6047
    h=0.1;
k=0.3;
roo=[1 1 0 0 0 -c];
pls=roots(roo);
h=0.15;%0.2;%0.1
p1 = pls(1);
p1cnj =pls(2);
p2 = pls(3);
p2cnj =pls(4);
p = pls(5);
a=0.945622101602222;%0.9442;%0.959340025428322;%0.9442;%h
    =0.1 i in , (changing h, changes those values do not
    forget!!!)
b=0.325267337679942;%0.282252928436864;%0.3295;%h=0.1 i in
ka=sqrt(2*k-(k^2)/(gamma^2));
a=a/b;
% c = 10;
% h = 0.2;
% k = 0.3;
% gamma = 2.088375883758838;
% a = 0.959340025428322;
% b = 0.282252928436864;
% a = a/b;
% ka = sqrt(2*k-(k^2)/(gamma^2));

for i = 1 : length(om)
    s = 1i*om(i);

```

```

P = exp(-h*s)*(1+a*s)*(1+gamma^2*s^2)/((s-p^2)*(gamma*
    s)*(1+eps*s)^2);
C = (p^2-s)/((k*s^2+ka*s+1)*(1-a*s)+gamma*s*(1+a*s)*
    exp(-h*s));
PC(i) = P*C;
end
figure();
subplot(2,1,1);
semilogx((om),20*log10(abs(PC)));
grid on
title('Bode Plots of Open loop ');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude(dB) ');
hold on
subplot(2,1,2);
semilogx((om),unwrap(angle(PC))*180/pi,om,180*ones(1,
    length(om)), 'g—',om,-180*ones(1,length(om)), 'g—');
grid on
xlabel('Frequency (rad/sec) ');
ylabel('Phase(deg) ');

sig = linspace(0,pi/2,150);

for i = 1 : length(om) + length(sig)
    if (i < length(sig)+1)
        s = 1e-3*exp(1i*sig(i));
        P = exp(-h*s)*(1+a*s)*(1+gamma^2*s^2)/((s-p^2)*(
            gamma*s)*(1+eps*s)^2);
        C = (p^2-s)/((k*s^2+ka*s+1)*(1-a*s)+gamma*s*(1+a*s
            )*exp(-h*s));
        PC(i) = P*C;
    else

```

```

    s = 1i*om(i-length(sig));
    P = exp(-h*s)*(1+a*s)*(1+gamma^2*s^2)/((s-p^2)*(gamma*
        s)*(1+eps*s)^2);
    C = (p^2-s)/((k*s^2+ka*s+1)*(1-a*s)+gamma*s*(1+a*s)*
        exp(-h*s));
    PC(i) = P*C;
    end
end

figure()
plot(real(PC), imag(PC), real(PC), -imag(PC), 'r', -1, 0, 'k+');
title('Nyquist Plot');
xlabel('\Re');
ylabel('\Im');
grid on

VMh015k03 = min(abs(1+PC));
%% h = 0.25
om = logspace(-2, 2, 1e5);
PC = zeros(1, length(om));
eps = 0.005; %0.002 for lower order appr
c=5;
gamma=1.810669106691067; %1.6047; %2.088375883758838; %1.6047
    h=0.1;
k=0.3;
roo=[1 1 0 0 0 -c];
pls=roots(roo);
h=0.25; %0.2; %0.1
p1 = pls(1);
p1cnj = pls(2);
p2 = pls(3);
p2cnj = pls(4);
p = pls(5);

```

```

a=0.957221783541320;%0.9442;%0.959340025428322;%0.9442;%h
    =0.1 i in , (changing h, changes those values do not
    forget!!!)
b=0.289355243798994;%0.282252928436864;%0.3295;%h=0.1 i in
ka=sqrt(2*k-(k^2)/(gamma^2));
a=a/b;
% c = 10;
% h = 0.2;
% k = 0.3;
% gamma = 2.088375883758838;
% a = 0.959340025428322;
% b = 0.282252928436864;
% a = a/b;
% ka = sqrt(2*k-(k^2)/(gamma^2));

for i = 1 : length(om)
    s = 1i*om(i);
    P = exp(-h*s)*(1+a*s)*(1+gamma^2*s^2)/((s-p^2)*(gamma*
        s)*(1+eps*s)^2);
    C = (p^2-s)/((k*s^2+ka*s+1)*(1-a*s)+gamma*s*(1+a*s)*
        exp(-h*s));
    PC(i) = P*C;
end
figure();
subplot(2,1,1);
semilogx((om),20*log10(abs(PC)));
grid on
title('Bode Plots of Open loop ');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude (dB) ');

```

```

hold on
subplot(2,1,2);
semilogx((om),unwrap(angle(PC))*180/pi,om,180*ones(1,
    length(om)), 'g—',om,-180*ones(1,length(om)), 'g—');
grid on
xlabel('Frequency (rad/sec)');
ylabel('Phase(deg)');

sig = linspace(0,pi/2,150);

for i = 1 : length(om) + length(sig)
    if (i < length(sig)+1)
        s = 1e-3*exp(1i*sig(i));
        P = exp(-h*s)*(1+a*s)*(1+gamma^2*s^2)/((s-p^2)*(
            gamma*s)*(1+eps*s)^2);
        C = (p^2-s)/((k*s^2+ka*s+1)*(1-a*s)+gamma*s*(1+a*s)
            )*exp(-h*s);
        PC(i) = P*C;
    else
        s = 1i*om(i-length(sig));
        P = exp(-h*s)*(1+a*s)*(1+gamma^2*s^2)/((s-p^2)*(gamma*
            s)*(1+eps*s)^2);
        C = (p^2-s)/((k*s^2+ka*s+1)*(1-a*s)+gamma*s*(1+a*s)*
            exp(-h*s));
        PC(i) = P*C;
    end
end

figure()
plot(real(PC),imag(PC),real(PC),-imag(PC),'r',-1,0,'k+');
title('Nyquist Plot ');
xlabel('\Re ');

```

```

ylabel( '\Im' );
grid on
VMh025k03 = min(abs(1+PC));

```

MATLAB[®] code: C1fromYALTAc5h015.m

```

clc
clear

a=0.945622101602222;%0.9442;%0.959340025428322;%0.9442;%h
    =0.1 i in , (changing h, changes those values do not
    forget!!!)
b=0.325267337679942;%0.282252928436864;%0.3295;%h=0.1 i in
a=a/b;% to write L as abs+1/abs-1;
gamma=1.462653626536265;%1.6047;%2.088375883758838;%1.6047
    h=0.1;
k=0.3;% W2=ks;
ka=sqrt(2*k-(k^2)/(gamma^2));
roo=[1 1 0 0 0 -5];%denominator of the plant in knospe and
    zhu with c=10;
pls=roots(roo);
h=0.15;%0.2;%0.1% time delay
p1 = pls(1);
p1cnj =pls(2);
p2 = pls(3);
p2cnj =pls(4);
p = pls(5);
q = [-a*k k-a*ka ka-a 1; 0 a*gamma gamma 0];
%q = [1 ka-a k-a*ka -a*k; 0 gamma a*gamma 0];
iPolyMatrix=q;
iDegree = 4;
iDelay = h;
iDelayVector = 1;
iModArg = 1e-4;

```

```

iMode = 'NORM';
iRootsOption = 1;

%T = delayFrequencyAnalysisMin(iPolyMatrix, iDelayVector, 1,
    iDelay, 1);
%sqrt(T.UnstablePoles)
PadeStruct = computePade( iPolyMatrix, iDegree, iDelay,
    iDelayVector, iModArg, iMode, iRootsOption );
C_tilda_appr = computeTF(PadeStruct.Numerator, PadeStruct.
    Denominator);

s = tf('s');
C1den_appr_yalta = minreal(C_tilda_appr*(s+1)^4, 1e-3);

C1_appr_yalta = minreal((1 + gamma^2*s^2)*(1-s/p^2)/
    C1den_appr_yalta, 1e-3); %% C1 approximant
pole(C1_appr_yalta);
%%
om = logspace(-4, 4, 5e2);
C1_appr_yalta_f = freqresp(C1_appr_yalta, om);

C1 = zeros(1, length(om));
%C1den = zeros(1, length(om));
for i = 1 : length(om)
    s=1i*om(i);
    C1(i) = (1 + gamma^2*s^2)*(1-s/p^2)/((k*s^2 + ka*s +1)
        *(1-a*s)+gamma*s*(1+a*s)*exp(-iDelay*s) );
    %C1den(i) = ((k*s^2 + ka*s +1)*(1-a*s)+gamma*s*(1+a*s)
        *exp(-iDelay*s) );
end

```

```

figure ()
subplot (2,1,1);
semilogx((om),(abs(C1_appr_yalta_f(1,:))), 'k',(om),(abs(C1
    )), 'b');
grid on
title('Frequency Responses of C1 and Its Appr. by YALTA')
    ;
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude ');
subplot (2,1,2);
semilogx((om),unwrap(angle(C1_appr_yalta_f(1,:)))*180/pi, '
    k', (om),unwrap(angle(C1))*180/pi, 'b');
grid on
xlabel('Frequency (rad/sec) ');
ylabel('Phase(deg) ');

```

```

figure ()
semilogx(om,abs(C1-C1_appr_yalta_f(1,:)))
grid on
title('Error Between Frequency Responses of C1 and Its
    Approximation_{YALTA} ');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude ');
% figure ()
% semilogx(om, abs(C1den))

```

MATLAB[®] code: carlson_method_cont_appr_of_fractance.m

```

clc
clear

```

```

Ho=1;

```

```

s=tf('s');
G = 1/s;
H = minreal((Ho^2 + 3*G)/(3*Ho^2 + G));
for i=1:2
    H = minreal(H*(H^2 + 3*G)/(3*H^2 + G));
end

om = logspace(-10,10,1e3);
P = 1./sqrt(1i*om);
Hf = freqresp(H,om);
Hf= Hf(1,:);
figure(3)
semilogx(om,abs((Hf./(1+Hf))-(P./(1+P))));
title('E-4(j\omega)');
xlabel('Frequency (rad/sec)');
ylabel('Magnitude');
grid on

```

```

%%

%% Carlson's Method
%T = 1e-3;

Pappr = tf([1 36 126 84 9],[9 84 126 36 1]);
%Pappr = tf([0.0859 4.877 20.84 12.995 1],[1 13 20.84
    4.876 0.8551]);
om = logspace(-5,5,1e3);
P = 1./sqrt(1i*om);

Papprf = freqresp(Pappr,om);

figure(1)
subplot(2,1,1);
semilogx((om),20*log10(abs(Papprf(1,:))), 'r',(om),20*log10
    (abs(P)));
grid on
title('Frequency Responses of Fractance and Its
    Approximation from Carlsons Method');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude dB');

```

```

subplot(2,1,2);
semilogx((om),unwrap(angle(Papprf(1,:)))*180/pi,'r',(om),
    unwrap(angle(P))*180/pi);
grid on
xlabel('Frequency (rad/sec)');
ylabel('Phase(deg)');

```

```

Ros = feedback(Pappr,1);
Pfdbck = P./(1+P);

```

```

Rosf = freqresp(Ros,om);

```

```

figure(2)
subplot(2,1,1);
semilogx((om),20*log10(abs(Rosf(1,:))),'r',(om),20*log10(
    abs(Pfdbck)));
grid on
title('Frequency Responses of Original Feedback loop and
    Its Approximation from Carlsons Method');
xlabel('Frequency (rad/sec)');
ylabel('Magnitude dB');
subplot(2,1,2);
semilogx((om),unwrap(angle(Rosf(1,:)))*180/pi,'r',(om),
    unwrap(angle(Pfdbck))*180/pi);
grid on
xlabel('Frequency (rad/sec)');
ylabel('Phase(deg)');

```

```

figure(3)
semilogx(om,abs(Rosf(1,:)-Pfdbck));
title('Error Btw. Frequency Responses of Original Feedback
    loop and Its Approximation from Carlsons Method');
xlabel('Frequency (rad/sec)');

```

```
ylabel('Magnitude');
```

MATLAB[®] code: fdbckanalytc.m

```
%% This function gives analytical expression for the step
   response of the feedback loop
% formed by 1/sqrt(s)
% clc
% clear

T = 1e-3;
N = 5e4;
n = 1:N;
h = zeros(1,N);
t = linspace(0,10,1e3);
% for i = 1 : N
%     h(i) = T*(1/sqrt(pi*n(i)*T) - exp(n(i)*T)*erfc(sqrt(
%         n(i)*T)));
% end
% h = [1 h];
%sysfz = tf(h,[1 zeros(1,length(h)-1)],T);
stepResp=zeros(1,length(t));
for i = 1 : length(t);
    stepResp(i) = 1 - exp(t(i))*erfc(sqrt(t(i)));
end
figure()
plot(t,stepResp,'k');
xlabel('Time in sec');
ylabel('Step Response');
title('Step Response of the Feedback Loop of 1/s^\alpha
and R_o(z)');
```

MATLAB[®] code: firststatespace_tustin_with_exp_star_lyap.m

```
clc
```

```
clear
```

```
%% create discrete state space rep. of FIR filter  
% N : number of coefficients  
% A,B,C,D are the matrices  
% T, sampling period  
N = 2e3;  
A = [zeros(N-1,1),eye(N-1)];  
A = [A ; zeros(1,N)];  
T = 1e-2;  
B = [zeros(N-1,1);1];  
% gain and h0 values are found by brute force,  
syszcforgainandho.m  
% gain = 1.5;  
% h0 = 0.5;  
gain = 1;  
h0 = 1.4;  
  
coef = N : -1 :1;  
  
C = sqrt(T/pi)*gain*1./sqrt(coef);  
D = sqrt(T/pi)*h0;  
% H is a discrete system, representing FIR filter.  
H = ss(A,B,C,D,T);  
%H_fdbck = feedback(H,1);% stable: sum(abs(pole(H_fdbck))  
>1) = 0;  
  
%% Model Reduction by using the method given by  
Approximation Of Infinite  
% Dimensional Systems Gu, Khargonekar mainly equation 4.6  
and 4.7  
Q = dlyap(A,B*B');
```

```

P = dlyap(A,C'*C);
[U,E,V] = svd(P);
Tm = U';
Ab = Tm*A*Tm';
Bb = Tm*B;
Cb = C*Tm';
Db = D;

% Equation 4.7 for getting 'rsize' elements of larger
% matrices.
rsize = 150;

Ab_t = [eye(rsize) zeros(rsize,N-rsize)]*Ab*[eye(rsize);
zeros(N-rsize,rsize)];
Bb_t = [eye(rsize) zeros(rsize,N-rsize)]*Bb;
Cb_t = Cb*[eye(rsize); zeros(N-rsize,rsize)];

%[n3,d3] = ss2tf(Ab_t,Bb_t,Cb_t,Db);

%Hrdcd = ss(Ab_t,Bb_t,Cb_t,Db,1e-3);

%% From discrete state space to Continuous State Space
% Conversion
[m,n] = size(Ab_t);
Ac = 2/T*(Ab_t-eye(m))*pinv(Ab_t+eye(m));
Bc = 1/sqrt(T)*(Bb_t-(Ab_t-eye(m))*pinv(Ab_t+eye(m))*Bb_t)
;
Cc = 2/sqrt(T)*Cb_t*pinv(Ab_t+eye(m));
Dc = Db - Cb_t*pinv(Ab_t+eye(m))*Bb_t;

%% FIR implementation
P1 = ss(Ac,Bc,Cc,Dc);
h = (N-1)*T;

```

```

Ac2 = Ac;
Bc2 = Bc;
Dc2 = 0;
Cc2 = Cc*expm(h*Ac);
P2 = ss(Ac2,Bc2,Cc2,Dc2,'InputDelay',h);

% Happ is the continuous tr fnc of FIR.
Happ = (((P1)-(P2)));

%% Getting Frequency Response
om = logspace(-8,log10(1/(2*T)),5e2);
Hf = freqresp(H,om); % discrete system's freq resp
Happf = freqresp(Happ,om); % continuous app's freq resp
Happf = (Hf(1,+)/Happf(1,))*Happf; % to get same DC gain
Rof = Happf(1,+)/.(1+Happf(1,)); % freq resp of feedback
    loop by cont. tr fnc.

%Rosys = feedback(Happ,1); % stable: sum(real(pole(Rosys))
    >0) = 0; % ss form
Po = 1./sqrt(1i*om);
Po_fdbck = Po./(1+Po);

figure(1)
subplot(2,1,1);
semilogx((om),20*log10(abs(Hf(1,)))),(om),20*log10(abs(
    Happf(1,))), 'r—');
grid on
title('Frequency Response of h[n]');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude dB');
subplot(2,1,2);

```

```

semilogx((om),unwrap(angle(Hf(1,:)))*180/pi,(om),unwrap(
    angle(Happf(1,:)))*180/pi,'r—');
grid on
xlabel('Frequency (rad/sec)');
ylabel('Phase(deg)');

```

```

%Rosysf = freqresp(Rosys,om);
figure(2)
semilogx((om),abs(Rof-Po_fdbck));
title('Error Between Feedback Loops');
xlabel('Frequency (rad/sec)');
ylabel('Magnitude');
% disp('Stability of discrete feedback loop:')
% sum(abs(pole(H_fdbck))>1)
% disp('Stability of continuous feedback loop:')
% sum(real(pole(Rosys))>0)
% figure(3)
% step(Rosys,N*T);

```

MATLAB[®] code: firststatespace_tustin_with_exp_star_no_rdctn.m

```
clc
```

```
clear
```

```

%% create discrete state space rep. of FIR filter
% N : number of coefficients
% A,B,C,D are the matrices
% T, sampling period
N = 2e3;
A = [zeros(N-1,1),eye(N-1)];
A = [A ; zeros(1,N)];

```

```

T = 1e-2;
B = [zeros(N-1,1);1];
% gain and h0 values are found by brute force ,
    syszcforgainandho.m
% gain = 1.5;
% h0 = 0.5;
gain = 1;
h0 = 1.4;

coef = N : -1 :1;

C = sqrt(T/pi)*gain*1./sqrt(coef);
D = sqrt(T/pi)*h0;
% H is a discrete system , representing FIR filter .
H = ss(A,B,C,D,T);
%H_fdbck = feedback(H,1);% stable : sum(abs(pole(H_fdbck))
    >1) = 0;

%% Model Reduction by using the method given by
    Approximation Of Infinite
% Dimensional Systems Gu, Khargonekar mainly equation 4.6
    and 4.7
% Q = dlyap(A,B*B');
% P = dlyap(A,C'*C);
% [U,E,V] = svd(P);
% Tm = U';
% Ab = Tm*A*Tm';
% Bb = Tm*B;
% Cb = C*Tm';
% Db = D;
%
```

```

%% Equation 4.7 for getting 'rsize' elements of larger
    matrices.
rsize = 150;
%
% Ab_t = [eye(rsize) zeros(rsize, N-rsize)]*Ab*[eye(rsize);
    zeros(N-rsize, rsize)];
% Bb_t = [eye(rsize) zeros(rsize, N-rsize)]*Bb;
% Cb_t = Cb*[eye(rsize); zeros(N-rsize, rsize)];
%
% [n3, d3] = ss2tf(Ab_t, Bb_t, Cb_t, Db);
%
% %Hrdcd = ss(Ab_t, Bb_t, Cb_t, Db, 1e-3);

%% From discrete state space to Continuous State Space
    Conversion
[m, n] = size(A);
Ac = 2/T*(A-eye(m))*pinv(A+eye(m));
Bc = 1/sqrt(T)*(B-(A-eye(m))*pinv(A+eye(m))*B);
Cc = 2/sqrt(T)*C*pinv(A+eye(m));
Dc = D - C*pinv(A+eye(m))*B;

%% FIR implementation
P1 = ss(Ac, Bc, Cc, Dc);
h = (N-1)*T;
Ac2 = Ac;
Bc2 = Bc;
Dc2 = 0;
Cc2 = Cc*expm(h*Ac);
P2 = ss(Ac2, Bc2, Cc2, Dc2, 'InputDelay', h);

% Happ is the continuous tr fnc of FIR.
Happ = (P1-P2);

```

```

%% Getting Frequency Response
om = logspace(-8,log10(1/(2*T)),5e2);
Hf = freqresp(H,om); % discrete system's freq resp
Happf = freqresp(Happ,om); % continuous app's freq resp
Happf = (Hf(1)/Happf(1))*Happf; % to get same DC gain
Rof = Happf(1,:)./(1+Happf(1,:)); % freq resp of feedback
    loop by cont. tr fnc.

%Rosys = feedback(Happ,1); % stable: sum(real(pole(Rosys))
    >0) = 0; % ss form
Po = 1./sqrt(1i*om);
Po_fdbck = Po./(1+Po);

figure(1)
subplot(2,1,1);
semilogx((om),20*log10(abs(Hf(1,:))), (om),20*log10(abs(
    Happf(1,:))), 'r—');
grid on
title('Frequency Response of h[n]');
xlabel('Frequency (rad/sec)');
ylabel('Magnitude dB');
subplot(2,1,2);
semilogx((om),unwrap(angle(Hf(1,:)))*180/pi, (om),unwrap(
    angle(Happf(1,:)))*180/pi, 'r—');
grid on
xlabel('Frequency (rad/sec)');
ylabel('Phase(deg)');

```

figure(2)

```

semilogx((om),abs(Rof-Po_fdbck));
title('Error Between Feedback Loops');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude');
% disp('Stability of discrete feedback loop:')
% sum(abs(pole(H_fdbck))>1)
% disp('Stability of continuous feedback loop:')
% sum(real(pole(Rosys))>0)
% figure(3)
% step(Rosys,N*T);

```

MATLAB[®] code: inverse_laplace_of_freq_data_of_feedback_loop.m

```
clc
```

```
clear
```

```
om = logspace(-8,3,5e3);
```

```
s = 1i*om;
```

```
Pfdbck = 1./(sqrt(s)+1);
```

```
iter=1e2;
```

```
tau1 = 3e-5;
```

```
tau2 = 1;
```

```
weight = (1 + s./tau1)./(1 + s./tau2).^2;
```

```
[num,den] = invfreqs(Pfdbck,om,9,18,weight,iter,1e-2);%% 9
```

```
18 24 12 10 30
```

```
Ros_m = tf(num,den);
```

```
Rosmf = freqresp(Ros_m,om);
```

```
% N = 18;
```

```
% [A,B,C,D] = tf2ss(num,den);
```

```
% Q = lyap(A,B*B');
```

```
% P = lyap(A,C'*C);
```

```

% [U,E,V] = svd(P);
% Tm = U';
% Ab = Tm*A*Tm';
% Bb = Tm*B;
% Cb = C*Tm';
% Db = D;
% rsize = 10;
%
% Ab_t = [eye(rsize) zeros(rsize,N-rsize)]*Ab*[eye(rsize);
        zeros(N-rsize,rsize)];
% Bb_t = [eye(rsize) zeros(rsize,N-rsize)]*Bb;
% Cb_t = Cb*[eye(rsize); zeros(N-rsize,rsize)];
%
% [na,da] = ss2tf(Ab_t,Bb_t,Cb_t,Db);
Ros_m_rdc = balred(Ros_m,4);

Ros_m_rdc = minreal(Ros_m_rdc,0.03);
Rf = freqresp(Ros_m_rdc,om);

figure(1)
subplot(2,1,1);
semilogx((om),20*log10(abs(Rf(1,:))), 'g.',(om),20*log10(
    abs(Rosmf(1,:))), 'r',(om),20*log10(abs(Pfdbck)));
grid on
title('Frequency Responses of Original Loop and Its
    Approximation from Frequency Data and Reduced Model');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude dB');
subplot(2,1,2);
semilogx((om),unwrap(angle(Rf(1,:)))*180/pi, 'g.',(om),
    unwrap(angle(Rosmf(1,:)))*180/pi, 'r',(om),unwrap(angle(
    Pfdbck))*180/pi);

```

```

grid on
xlabel( 'Frequency (rad/sec) ');
ylabel( 'Phase(deg) ');

figure(2)
semilogx(om, abs(Rosmf(1, :)-Pfdbck), 'r')
title( 'Error Between Actual feedback loop and Its Appr.
    invfreqs ');
xlabel( 'Frequency (rad/sec) ');
ylabel( 'Magnitude ');

figure(3)
semilogx(om, abs(Rf(1, :)-Pfdbck), 'g')
title( 'Error Between Actual feedback loop and reduced
    model of its Appr. invfreqs ');
xlabel( 'Frequency (rad/sec) ');
ylabel( 'Magnitude ');

zpk(Ros_m_rdc)
s=tf('s')
Papp_invfreq=feedback(series(1/s^2, Ros_m_rdc), 10, +1);
pole(Papp_invfreq)
%% brute to order
% tau1 = 3e-5;
% tau2 = 1;
% weight = (1 + s./tau1)./(1 + s./tau2).^2;
%
% N = 30;
% M = 30;
% e = zeros(N,M);
% for m = 1 : N
%     for n = 1 : M
%         [num, den] = invfreqs(Pfdbck, om, n, m, weight);

```



```

a = zeros(1,N);
a(1) = (subs(v,1i*om(1)));
s = tf('s');
for i = 1 : N-1
    v = (x - 1i*om(i))/(v-a(i));
    a(i+1) = (subs(v,1i*om(i+1)));
end
fnc = tf([a(end)],1);
for i = N : -1 : 2
    fnc = a(i-1) + (s-1i*om(i-1))/fnc;
end

ome =logspace(-20,20,5e2);
fnc=tf(abs(fnc.num{1}),abs(fnc.den{1}));
Ros_matsuda_block = feedback(fnc,1);
Ros_m_bf = freqresp(Ros_matsuda_block,ome);
Pfdbck = 1./(sqrt(1i*ome)+1);
e=abs(Ros_m_bf(1,:)-Pfdbck);
figure
semilogx(ome,e)
title('E-3(j\omega)');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude ');
%fnc = tf(abs(fnc.num{1}),abs(fnc.den{1}));
figure
some = tf([0.08549 4.877 20.84 12.995 1], [1 13 20.84
    4.876 0.08551]);
bode(some,fnc,ome);
[d,t]=step(Ros_matsuda_block,300);
figure
plot(t,d,'r',t,1 - exp(t).*erfc(sqrt(t)));

```

```

title('Step Responses of Original Loop and Its
Approximation _Matsuda');
xlabel('Time (sec) ');
ylabel('Step Response');
Papp_matsuda_block=feedback(series(1/s^2,Ros_matsuda_block
),10,+1);
pole(Papp_matsuda_block)
figure()
subplot(2,1,1);
semilogx((ome),20*log10(abs(Ros_m_bf(1,:))), 'r',(ome),20*
log10(abs(Pfdbck)));
grid on
title('Frequency Responses of Original Loop and Its
Approximation _Matsuda');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude dB');
subplot(2,1,2);
semilogx((ome),unwrap(angle(Ros_m_bf(1,:)))*180/pi, 'r', (
ome),unwrap(angle(Pfdbck))*180/pi);
grid on
xlabel('Frequency (rad/sec)');
ylabel('Phase(deg)');
%%

clc
clear
syms x;
%T = 5e-2;log10(1/(2*T));
%log10(1/(2*T))
N = 31;
om = logspace(-5,5,N);

```

```

v = 1 / (sqrt(x)+1);
a = zeros(1,N);
a(1) = (subs(v,1i*om(1)));
s = tf('s');
for i = 1 : N-1
    v = (x - 1i*om(i))/(v-a(i));
    a(i+1) =( subs(v,1i*om(i+1)));
end
fnc = tf((a(end)),1);
for i = N : -1 : 2
    fnc =minreal((a(i-1)) + (s-(1i*om(i-1)))/fnc);
end
fnc = tf(abs(fnc.num{1}),abs(fnc.den{1}));%
    %%%%%%%%%% hocaya sor
Ros_matsuda=fnc;

om = logspace(-5,5,1e3);
Ros_matsuda_f = freqresp(fnc,om);
%Ros_matsuda_r_f = freqresp(Ros_matsuda_r,om);
Pfdbck=1./(sqrt(1i*om)+1);

figure(1)
subplot(2,1,1);
semilogx((om),20*log10(abs(Ros_matsuda_f(1,:))), 'r', (om)
    ,20*log10(abs(Pfdbck)));
grid on
title('Frequency Responses of Original Loop and Its
    Approximation _Matsuda');
xlabel('Frequency (rad/sec) ');
ylabel('Magnitude dB');
subplot(2,1,2);
semilogx((om),unwrap(angle(Ros_matsuda_f(1,:)))*180/pi, 'r'
    , (om),unwrap(angle(Pfdbck))*180/pi);

```

```

grid on
xlabel( 'Frequency (rad/sec) ');
ylabel( 'Phase(deg) ');

figure(2)
semilogx(om, abs(Ros_matsuda_f(1,:) - Pfdbck), 'r')
title( 'Error Between Actual feedback loop and reduced
        model of its Appr. invfreqs ');
xlabel( 'Frequency (rad/sec) ');
ylabel( 'Magnitude ');

[naa, daa]=tfdata(fnc);
if( length(daa{1,1}) - length(naa{1,1}) >= 0)
    disp( 'proper ');
    disp(length(daa{1,1}))
    disp(-length(naa{1,1}))
end
if(sum(real(pole((Ros_matsuda)))) > 0)
    disp( 'Model is unstable ');
end

```