ON THE STRUCTURE AND IMPLEMENTATION OF THE OPTIMAL Q-PARAMETER IN THE ONE-BLOCK H-INFINITY CONTROL PROBLEM

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

By
Çetin Taştekin
November 2020

On the Structure and Implementation of the Optimal Q-Parameter in the One-Block H-Infinity Control Problem

By Çetin Taştekin

November 2020

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

Hitay Özbay (Advisor)

Ömer Morgül

Mehmet Önder Efe

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

ON THE STRUCTURE AND IMPLEMENTATION OF THE OPTIMAL Q-PARAMETER IN THE ONE-BLOCK H-INFINITY CONTROL PROBLEM

Çetin Taştekin
M.S. in Electrical and Electronics Engineering
Advisor: Hitay Özbay
November 2020

In the robust control theory, H_∞ methods are carried out to procure solutions to sensitivity minimization (nominal performance) and robust stability problems in general. These one – block control problems can be analyzed separately or in a combined fashion. In the general sense, the design of the robust controller is made to achieve stability and performance objectives for a plant, whose nominal model and uncertainty bounds can be determined from the experimental data.

This study aims to present a solution to infinite dimensional one – block H_∞ control problem and provide a general structural result for Q – Parameter. The methods like Nevanlinna – Pick interpolation will not work for infinite dimensional control problems whereas Sarason's Theorem provides solution to infinite dimensional control problems. In this thesis, Sarason's Theorem is analyzed and examined extensively and applied on infinite dimensional one – block H_∞ control problem.

The detailed structural analysis of the resulting Q – Parameter is made as the main contribution of the thesis. Various examples are given to illustrate the computational issues. In this analysis, stability status of Q – Parameter is demonstrated by examining stable terms and FIR part of its sub – blocks.

Keywords: Robust control, one - block H_∞ control problem, H_∞ controller, infinite dimensional systems, Sarason's Theorem, Q - Parameter.

ÖZET

TEK-BLOK H-SONSUZ KONTROL PROBLEMİNDE OPTİMAL Q-PAREMETRESİNİN YAPISI VE UYGULAMASI

Çetin Taştekin
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Hitay Özbay
Kasım 2020

Gürbüz kontrol teorisinde, H_∞ yöntemleri, genel olarak hassaslık minimizasyonu (nominal performans) ve gürbüz kararlılık problemleri için çözümler sağlamak amacıyla uygulanır. Bu tek bloklu kontrol problemleri, ayrı ayrı analiz edilebilir veya birlikte incelenebilir. Genel anlamda, deneysel verilerden belirlenebilen nominal bir sistem ve belirsizlik altında kararlılık ve performans hedeflerine ulaşmak için gürbüz kontrolcü tasarımı yapılır.

Bu çalışma, sonsuz boyutlu tek bloklu H_∞ kontrol problemine bir çözüm ortaya koymayı ve Q – Parametresi için genel bir yapısal sonuç sunmayı amaçlamaktadır. Nevanlinna - Pick interpolasyonu gibi yöntemler sonsuz boyutlu kontrol problemlerinde çalışmazken, Sarason Teoremi sonsuz boyutlu kontrol problemlerine çözüm sağlar. Bu bağlamda, Sarason Teoremi kapsamlı bir şekilde analiz edilip incelenerek, sonsuz boyutlu tek bloklu H_∞ kontrol problemine uygulanmıştır.

Tezin ana katkısı, elde edilen Q - Parametresi'nin detaylı yapısal analizidir. Hesaplama meselelerinin canlandırması için çeşitli örnekler verilmiştir. Bu analizde, Q - Parametresinin kararlılık durumu, yapısının kararlı terimleri ve FIR kısmı incelenerek gösterilmiştir.

Anahtar sözcükler: Gürbüz kontrol, tek - blok H_∞ kontrol problemi, H_∞ kontrolcüsü, sonsuz boyutlu sistemler, Sarason Teoremi, Q - Parametresi.

Acknowledgement

First of all, I would like to express my endless gratitude to my advisor Professor Hitay Özbay. In the most hopeless period of my graduate education, he accepted to be my advisor. He just believed in me and helped me, just like a father. There are no simple words to tell how I respect him. He will always remain as a father and the greatest role model for me in my whole life. His understanding, patience, absolute knowledge and immense support are priceless and most precious for me and my graduate education. Today, if this work exists, it is because of him. It was an honor to be his M.Sc student and have a chance to work with him. I will be always appreciated.

I would like to thank Professor Mehmet Önder Efe and Professor Ömer Morgül for reading my thesis and guiding me in the thesis committee.

I also want to thank Professor Süleyman Serdar Kozat for supervising me at the beginning process of my graduate education. As my undergraduate advisor and respected instructor, his assistance was invaluable to shape my graduate study. Special thanks to Professor Arif Bülent Özgüler for accepting me as M.Sc student in the first place and constant support during my studies.

I would like to state my sincerest thanks and gratefulness to my parents, Gülen and İsmet Kemal Taştekin, and also my brother Osman Taştekin. Besides, I am grateful and beholden to my precious grandmother, Kadriye Soysal. With their endless support, I become who I am right now. I also want to express my special thanks and appreciation to my mother and father in law, Ferhan and Sadık Sezai Karaköse. They always backed me up and encourage me at the all stages of my studies.

I want to thank Ali Devran Kara who was always with me since the beginning of my undergraduate studies. We always supported and encouraged each other even if we live in different countries now. I also would like to thank Fatih Aslan for always being on my side since the days of Germany.

Also special thanks to my schoolfellows Giray İlhan and Serkan İslamoğlu for helping and supporting me all the time.

Last but foremost, I would like to express my warmest and adoring gratitude to my beloved wife, Pınar for always supporting, advising and encouraging me during my studies. Whenever I lost my hope and courage about my studies, she was the one always being next with me. My thesis would not exist at all if she had not been there for backing me up. This accomplishment also belongs to her since she stands out against all hard times on my side. As final words, I just want to say to her, “Pınar, we did it!”.

Contents

List of Figures

to my beloved wife, Pınar

Chapter 1

Introduction

The robust control theory consists H_∞ methods in order to acquire solution for sensitivity minimization (nominal performance) and robust stability problems. Both of these control problems are handled separately or combinedly. In the collective manner, we seek for a design of robust controller that provides stability and performance objectives under an actual plant, which can be determined from the experimental data.

Various researchers worked on different types of H_∞ - based robust control problems and contribute different approaches to the robust control theory such as finite dimensional control problems, time-delay systems control problems, stochastic control etc. [4], [7-9], [11-22].

To design H_∞ robust controllers, following feedback system (C, P) is considered for single-input-single-output systems [1]:

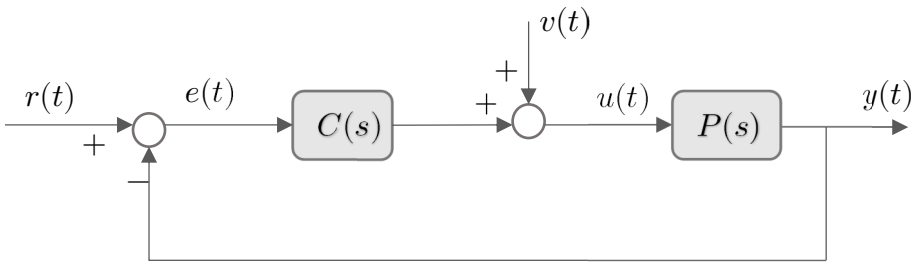

Figure 1.1 Feedback system (C, P)

Here in this configuration, it is beneficial to define two closed loop transfer functions such as following:

r to e : $S(s) = \frac{1}{1+P(s)C(s)}$ which is the sensitivity function

r to y : $T(s) = \frac{P(s)C(s)}{1+P(s)C(s)} = 1 - S(s)$ the complementary sensitivity

In general robust control problems, we have the plant model as following, where $P(s)$ is nominal fixed transfer function:

$$P_{\Delta}(s) = P(s) + \Delta(s)$$

where Δ(s) is the uncertainty; it is represented by an upper bound function W(s) such that $|\mathrm{W(s)}| > |\Delta(\mathrm{s})|$ where s = jw and w ≥ 0.

Commonly, there are additive (Δ_a) and multiplicative (Δ_m) uncertainties which define all possible plants $P_{\Delta}(s)$ as following respectively:

$$\mathcal{P}_a = \{P_{\Delta} = P + \Delta_a, |W_a(jw)| > |\Delta_a(jw)| for\ all\ w \geq 0\}$$

$$\mathcal{P}_m = \{P_{\Delta} = P + P\Delta_m, |W_m(jw)| > |\Delta_m(jw)|\ for\ all\ w \geq 0\}$$

Then, robust stability condition for the feedback system (C, P_{Δ}) can be defined as at below:

$$||W_a CS||_{\infty} \leq 1 \quad for\ additive\ uncertainty\ bound, i.e\ P_{\Delta} \epsilon \mathcal{P}_a$$

or

$$||W_m T||_\infty \leq 1 \quad \textit{for multiplicative uncertainty bound, i. e } P_\Delta \epsilon \mathcal{P}_m$$

For proof, detailed derivations and discussions, [1][3][4][23] can be seen.

In robust control problems, sensitivity minimization is also an important issue in order to investigate. For the feedback system (C, P) at the above figure, the tracking error is $e(t) \equiv r(t) - y(t)$ and we want $e(t)$ as small as possible. Also, we can define set of all reference signals as following:

$$\mathcal{R} = \{R(s) = W_r(s)R_o(s) \textit{ where } r_o(t) \textit{ is finite energy signal}\}$$

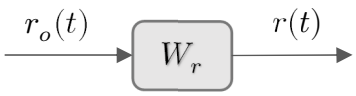

Figure 1.2 Reference Signal

Here, $W_r(s)$ can be seen as a reference generator filter or sensitivity weight which is generally a low – pass filter, [2]. Thus, we have performance indicator which is the cost function to be minimized is at below:

$$\underset{r_o(t)\epsilon\mathcal{R}}{sup} \frac{||e(t)||_2}{||r_o(t)||_2}$$

The transfer function from $r_o(t)$ to $e(t)$ is equal to $W_r S = \frac{W_r}{1+PC}$. Hence, our cost function is defined as following:

$$cost \equiv \underset{r_o(t)\epsilon\mathcal{R}}{sup} \frac{||e(t)||_2}{||r_o(t)||_2} = ||W_r S||_\infty = ||\frac{W_r}{1+PC}||_\infty$$

Achieving $||W_r S||_\infty \leq \gamma_r$, where $\gamma_r > 0$ as small as possible, is called sensitivity minimization or nominal performance problem.

Both sensitivity minimization and robust stability are important and essential in robust control theory. They are called one – block H_∞ control problems which is our main interest to provide a general structure.

On the other side, we define a condition, which is called robust performance, in order to ensure sensitivity minimization for all possible $P_\Delta(s)$ and also robust stability of the system. Thereby, robust performance condition is defined as at below: [1]

Given the nominal plant $P(s)$ and two weights $W_1(s)$and $W_2(s)$, design $C(s)$ such that feedback system is stable and $|W_1S| + |W_2T| \leq 1$ for $\forall w$

Herein, $W_1(s)$ is the sensitivity weight and $W_2(s)$ is the robustness weight. It is straightforward that when $W_1(s) = 0$, sensitivity minimization condition remains. Similarly, when $W_2(s) = 0$, robust stability condition remains. As can be understood, we can use different weights so that we can adjust sensitivity or robustness.

By using robust performance inequality, we can define mixed sensitivity inequality as following:

$$|W_1S| + |W_2T| \leq 1 \ <= \ |W_1S|^2 + |W_2T|^2 \leq \tfrac{1}{2} \ <=> \ || \begin{bmatrix} W_1S \\ W_2T \end{bmatrix} ||_\infty \ \leq \tfrac{1}{\sqrt{2}}$$

As long as the resultant controller stabilize the system, it also satisfies the robust performance condition automatically. Though, if we want to obtain one optimal controller that robustly stabilize the system and satisfies the robust performance condition, following inequality should be considered:

$$|| \begin{bmatrix} W_1S \\ W_2T \end{bmatrix} ||_\infty \ \leq \gamma$$

where $\gamma > 0$ is to be made as small as possible for given W_1, W_2 and the nominal plant P.

Here, minimizing γ provide us opportunity to maximize margins of weights and minimize the cost. In robust control theory, this problem is called two – block H_∞ control problem or mixed sensitivity minimization and general definition is at below:

$$\gamma_{opt} := \inf_{(C,P)\ stable} || \begin{bmatrix} W_1 S \\ W_2 T \end{bmatrix} ||_\infty$$

As in the above equation, H_∞ - based robust control problems can have the combination of sensitivity minimization and robust stability problems which brings us two blocks H_∞ problems to solve. So, the way to solve two – block H_∞ control problem is reducing it to one – block problem. The main idea in here is applying spectral factorization from the weights. Detailed calculations can be found on [4].

In this respect, providing general condition and structure is crucial for one – block H_∞ control problem. Thus, many efforts and works about H_∞ - based robust control problems are focused on one – block control problem.

In order to reveal general condition for one – block control problem, we firstly define parametrization of all controllers $\mathcal{C}$ stabilizing the above feedback system (C, P) as follows [3]:

$$\mathcal{C}(P) = \left\{ C(s) = \frac{X(s) + D(s)Q(s)}{Y(s) - N(s)Q(s)} \ : \ Q \epsilon H_\infty, \quad Y(\infty) - N(\infty)Q(\infty) \neq 0 \right\}$$

where $P(s) = \frac{N(s)}{D(s)}$ which is obtained with coprime factorization such that $N, D \ \epsilon \ H_\infty, N(s)$ and $D(s)$ has no common zeros in RHP and we assume $D(s)$ is inner function and,

$$N(s)X(s) + D(s)Y(s) = 1 \ , \quad N, D, X, Y \epsilon H_\infty$$

$$S(s) = D(s)\big(Y(s) - N(s)Q(s)\big), \qquad T(s) = N(s)\big(X(s) + D(s)Q(s)\big)$$

Then we have robust stability and nominal performance problem as following:

Robust stability problem;

$$||W_a CS||_\infty \leq 1 \quad => \quad ||W_a D(X + DQ)||_\infty \leq 1$$
$$\overset{(D \text{ is inner})}{=>} \quad ||W_a X - (-W_a DQ)||_\infty \leq 1$$

or

$$||W_m T||_\infty \leq 1 \quad => \quad ||W_m N(X + DQ)||_\infty \leq 1$$
$$=> \quad ||W_m NX - (-W_m NDQ)||_\infty \leq 1$$

Nominal Performance problem;

$$||W_r S||_\infty \leq \gamma \quad => \quad ||W_r D(Y - NQ)||_\infty \leq \gamma$$
$$\overset{(D \text{ is inner})}{=>} \quad ||W_r Y - W_r NQ)||_\infty \leq \gamma$$

Then, by absorbing the outer factors of $(W_r N)$ or $(W_m N)$ into the free parameter $Q \epsilon H_\infty$, we can convert both robust stability problem and sensitivity minimization problem to one – block H_∞ control problem as at below [2]:

$$\gamma_{opt} = \inf_{Q \epsilon H_\infty} ||W - MQ||_\infty = ||F_{opt}||_\infty$$

where W *and* M are given and $W, M \epsilon H_\infty$ and M is an inner function

Throughout the thesis, we consider W to be a rational outer function and M can be infinite dimensional, a typical example is the time delay $M(s) = e^{-hs}$ where h is the delay amount in the plant.

From the above equation, $\gamma_{opt} = ||W - MQ_{opt}||_\infty = ||F_{opt}||_\infty$ yields $Q_{opt} \epsilon H_\infty$ which is needed to obtain C_{opt} for the solution of the one – block H_∞ control problem.

As indicated before, many works and efforts are about to provide solution to various types of one – block H_∞ control problem such as finite or infinite

dimensional problems. This is called the model – matching and based on the mathematical interpolation theories [3].

One of the well – known techniques is Nevanlinna – Pick interpolation method. It is based on interpolating the distinct zeros of $M(s)$, which are $\propto_1, \ldots, \propto_n \ \epsilon \ \mathbb{C}_+$, to $F(s) = \ W(s) - M(s)Q(s)$ and get $W(\propto_i) = \beta_i$. Then, by defining $[A]_{i,j} = \frac{1}{\bar{\propto}_i + \propto_j}$ and $[B]_{i,j} = \frac{\bar{\beta}_i \beta_j}{\bar{\propto}_i + \propto_j}$ matrices with β_i and $\propto_i$ values γ_{opt} is obtained as $\gamma_{opt} = \sqrt{\lambda_{max}(A^{-1}B)}$ where λ_{max} is the largest eigenvalue. As a result, $F_{opt}(s) = \lambda \frac{[s^{n-1} \ldots \ s \ \ 1] J \phi}{[s^{n-1} \ldots \ s \ \ 1] \phi}$ where $J_{ixi} = (-1)^{i-1}, i = 1 \ldots . n$ and ϕ is an eigenvector of $JV_{\propto}^{-1} D_{\beta} V_{\propto}$ where $V_{\propto}$ is Vandermonde matrix constructed with $\propto_i$'s and D_{β} is diagonal matrix defined by $W(\propto_i) = \beta_i = [D_{\beta}]_{i,i}$ such that $\phi = [\phi_{n-1} \ldots \ \phi_0]^T \epsilon \ \mathbb{R}^n$ and $|\lambda| = \gamma_{opt}$. As a result, optimal $Q_{opt} \epsilon H_{\infty}$ is derived from the obtained F_{opt}. For more detailed computations, Nevanlinna – Pick Interpolation derivations, see [2],[4] and [24]. Nevanlinna – Pick approach is useful only when $M(s)$ is finite dimensional.

The other approach to solve one – block H_{∞} control problem is Nehari's Theorem which works for infinite dimensional $M(s)$ as well. In this theorem one – block problem is redefined as:

$$\gamma_{opt} = \inf_{Q \epsilon H_{\infty}} ||M^* W - Q||_{\infty}$$

since $M(s)$ is inner function as mentioned before. Then, $R(s) \coloneqq M^*(s) W(s)$ is defined such that $R(s) \epsilon L_{\infty}$ and $\gamma_{opt} = ||\Gamma_R||$ where Γ is the Hankel operator [10][25]. As a result, optimal $Q_{opt} \epsilon H_{\infty}$ is derived from the largest singular vector of the defined Hankel operator. The details of the Nehari's Theorem can be seen and examined from [2], [4] and [26].

The last one that should be mentioned is Sarason's Theorem which we are interested in this thesis to provide structural result of Q_{opt}. In fact, Nehari's

Theorem and Sarason's approach lead to the same computations for infinite dimensional systems. So, we focus only on Sarason's result.

In basic definition, by using Sarason operator $W(T)$, we define $\gamma_{opt} = ||W(T)||$ which is the norm of the Sarason operator. Then, optimal $Q_{opt} \epsilon H_\infty$ is derived from the singular vector of the Sarason operator corresponds to the largest singular value which is γ_{opt}. The details are given and discussed in the following chapters with theoretic derivations and different examples.

In this thesis, we are trying to find solution to infinite dimensional one – block H_∞ control problem and provide a general structural result for Q_{opt}. The Nevanlinna – Pick method will not work on infinite dimensional control problems. As indicated in [2], there is a lack of software tool for the infinite dimensional H_∞ - based robust control problems, which occasionally consists time delays, whereas the finite dimensional H_∞ - based robust control problems has practical and useful software tool such as Robust Control Toolbox of MATLAB [29]. In this manner, we are interested to use Sarason's theorem to solve infinite dimensional one – block H_∞ control problems, provide a general structure of Q_{opt} and present a computational tool to use in the implementation of Q_{opt} leading to optimal controller C_{opt}.

The rest of the thesis is organized as follows. In Chapter 2, the Sarason's operator and associated theorem is given. Chapter 3 contains a detailed structural analysis of the resulting Q_{opt}; that is the main contribution of the thesis. In Chapter 4 several examples are given to illustrate the computational issues. In Chapter 5, we make some concluding remarks.

Chapter 2

Sarason's Theorem

In the field of Robust Control Theory, it is important to reduce the robust control problem to one – block H_∞ control problem. As indicated in Chapter 1, there are methods such as Nevanlinna-Pick interpolation and Nehari solution to investigate a way to solve the one – block problem. The one – block H_∞ control problem is defined as given, $M(s)$ and $W(s)$ where $M \epsilon H_\infty$ is inner and $W \epsilon H_\infty$ finding γ_{opt} and $F_{opt} \epsilon H_\infty$ satisfying: [2]

$$\gamma_{opt} = \inf_{Q \epsilon H_\infty} ||W - MQ||_\infty = ||F_{opt}||_\infty$$

Here, it is assumed W is outer which automatically implies that W and M have no common zero on complex right half plane.

In this thesis, we will use Sarason's Theorem to solve γ_{opt}, F_{opt} and Q_{opt}. First, some notations and operators have to be defined as following:

$L_2(\mathbb{R}_+)$: Finite energy signals defined on $[0, \infty)$

$L_2(\mathbb{R}_-)$: Finite energy signals defined on $(-\infty,\ 0]$

$$L_2(\mathbb{R}) = L_2(\mathbb{R}_+) \oplus L_2(\mathbb{R}_-)$$

L_2 : The set of functions square integrable on imaginary axis. It can also be seen as the Fourier transforms of all functions in $L_2(\mathbb{R})$

C_+ : Complex right half plane

C_- : Complex left half plane

$H_\infty(C_+)$ *or shortly* H_∞ : Bounded analytic functions on C_+ which mean the set of all stable transfer functions

Also, if function g is in $L_2(R_+)$ then,

$$\mathcal{L}\{g\} = G(s)\epsilon H_2(C_+) \textit{ or shortly } H_2$$

Similarly, if function g is in $L_2(R_-)$ then,

$$\mathcal{L}\{g\} = G(s)\epsilon\, H_2(C_-) \textit{ or sortly } H_2^\perp$$

For any given $K \epsilon L_2$, we have a unique orthogonal decomposition in the form $K = K_+ + K_-$ where $K_+ \epsilon H_2$ and $K_- \epsilon H_2^\perp$. Then we can define projection operators as below:

$$\Pi_+(K) = K_+$$

$$\Pi_-(K) = K_-$$

Also, for an inner function $M \epsilon H_\infty$ we define the subspace $H(M) = H_2 \ominus \mathrm{M}H_2$, hence for any $K \epsilon H_2$, the projection onto $H(M)$ is defined as:

$$\Pi_{H(M)}(K) = G \text{ where } G \epsilon H(M) \textit{ and } G = K - M\Pi_+ M^* K$$

Thus, the Sarason's Operator is defined as:

$$W(T) : H(M) \to H(M) \qquad (1)$$

$$W(T)(R) = \Pi_{H(M)} WR$$

where $R \epsilon H(M)$

For more detailed treatment of the above definition we refer to [2] and [4]

Since the infinite dimensional systems are the main scope of this paper, the Sarason's theorem is needed to be examined as below:

Sarason's theorem[5]***:*** *Let W(T) be the Sarason operator defined in **(1)** and H(M) is the orthogonal complement of MH_2 in H_2, then there is a function F_{opt} in H_∞ such that* $||F_{opt}||_\infty = ||W(T)|| = \gamma_{opt}$

Assuming that the essential norm of $W(T)$ is strictly less than γ_{opt}, the norm is achieved at the discreate spectrum, meaning there exists a nonzero function $G \,\epsilon\, H(M)$ satisfying;

$$\gamma_{opt}^2 G = W(T)^* W(T) G$$

$$\left(\gamma_{opt}^2 I - W(T)^* W(T)\right) G = 0$$

which is the singular-value-singular vector equation

Then,

$$F_{opt} \ = (W - MQ_{opt}) = \frac{W(T)G}{G}$$

Moreover, F_{opt} is an inner function. Thus, γ_{opt} is equal to norm of Sarason operator and calculation of γ_{opt}, which is crucial for the solution of the one-block H_∞ control problem, can be done by finding largest singular value of $W(T)$. In this manner, Q_{opt} is calculated with the singular vector of $W(T)$ which corresponds to γ_{opt}.

Note that, $W(T)$ is finite size matrix for finite dimensional systems and interpolation problems, but in our case, $M(s)$ is infinite dimensional. In this case, $W(T)$ is not a finite size matrix since $H(M)$ is infinite dimensional. Nevertheless, a finite set of equations can be derived to compute γ_{opt} and the resulting F_{opt} and Q_{opt}. This will be illustrated in the next chapter.

Chapter 3

Main Results

In this section, we examine the singular-value-singular vector equation in more detail:

$$\left(\gamma_{opt}^2 I - W(T)^* W(T)\right) G = 0$$

This analysis has been done in the literature, [4],[10],[24], but we will investigate the structure and implementation of Q_{opt} when $M(s)$ is a delay term.

First order of business is the computation of $W(T)^* W(T)$ applied to a candidate singular vector $G \, \epsilon \, H(M)$. Note that:

$$W(T)G = \Pi_{H(M)} WG$$

$$\Pi_{H(M)} WG = WG - M\Pi_{+} M^* WG \tag{2}$$

where $M^*G \, \epsilon \, H_2^{\perp}$ and $W \, \epsilon \;\; H_2$

Consider a strictly proper $W(s)$ and assume that $W(s) = \sum_{i=1}^{n} \frac{\propto_i}{s+p_i}$, where p_i's are distinct. Then,

$$\Pi_{+}M^{*}WG = \sum_{i=1}^{n} \frac{\propto_i}{s+p_i}\phi_i \quad ,$$

where $\phi_i := (M^{*}G)(-p_i)$ and we define:

$$\psi_1(s) = \sum_{i=1}^{n} \frac{\propto_i}{s+p_i}\phi_i \quad ,$$

After that, (2) becomes:

$$W(T)G = WG - M\psi_1 \ .$$

Now we apply $W(T)^{*}$ to $W(T)G$ which is derived above:

$$\begin{aligned} W(T)^{*}W(T)G &= \Pi_{H(M)}W^{*}(WG - M\psi_1) \\ &= W^{*}(WG - M\psi_1) - \Pi_{-}W^{*}(WG - M\psi_1) \\ &= W^{*}(WG - M\psi_1) - \Pi_{-}W^{*}WG + \Pi_{-}W^{*}M\psi_1 \ . \end{aligned} \quad (3)$$

Let's define:

$$\Pi_{-}W^{*}WG : \textcircled{1}$$
$$\Pi_{-}W^{*}M\psi_1 : \textcircled{2}$$

Then these parts are analyzed one by one as following:

It is clear that ① is in the form:

$$\Pi_{-}W^{*}WG = \sum_{i=1}^{n} \frac{\propto_i}{p_i - s}\theta_i \quad ,$$

where $\theta_i := (WG)(p_i)$ and we define:

$$\psi_2(s) = \sum_{i=1}^{n} \frac{\propto_i}{p_i - s}\theta_i \quad .$$

Similarly, ② is in the form:

$$\Pi_{-}W^{*}M\psi_1 = \sum_{i=1}^{n} \frac{\propto_i}{p_i - s}\eta_i \ ,$$

where $\eta_i := (M\psi_1)(p_i)$ and we also define:

$$\psi_3(s) = \sum_{i=1}^{n} \frac{\propto_i}{p_i - s} \eta_i \quad .$$

After this procedure, ψ_2 and ψ_3 are put into (3):

$$\begin{aligned} W(T)^* W(T) G &= W^*(WG - M\psi_1) - \psi_2 + \psi_3 \\ &= W^* WG - W^* M\psi_1 - \psi_2 + \psi_3 \quad . \end{aligned} \tag{4}$$

Then by putting (4) to the singular-value-singular vector equation, following equation is obtained:

$$\gamma^2 G - W^* WG + W^* M\psi_1 + \psi_2 - \psi_3 = 0 \quad .$$

By re-arranging the terms, we get:

$$G(\gamma^2 - W^* W) = -W^* M\psi_1 - \psi_2 + \psi_3 \quad ,$$

which implies:

$$G(s) = \frac{-W^* M\psi_1 - \psi_2 + \psi_3}{\gamma^2 - W^* W} \tag{5}$$

where $G \neq 0$ and $G \, \epsilon \, H(M)$

As straightforward as it, if denominator of G (which is denoted by D_G) becomes zero in closed right half plane, then at that point numerator of G (which is denoted by N_G) should be also zero since $G \, \epsilon \, H(M)$. Thus,

$$D_G = \gamma^2 - W^* W = 0$$

at β_k where $\beta_k : \ \beta_1, \ldots, \beta_n, -\beta_1, \ldots, -\beta_n, k : 1, \ldots .,2n$ (note that $W^* W$ is symmetric)

In this respect, $N_G = 0 \ at \ \beta_k{}'s$:

$$N_G(\beta_k) = (-W^* M\psi_1 - \psi_2 + \psi_3)|_{s=\beta_k} = 0$$

Since,

$$N_G(s) = -W^* M \sum_{i=1}^{n} \frac{\propto_i}{s+p_i}\phi_i - \sum_{i=1}^{n} \frac{\propto_i}{p_i - s}\theta_i \; + \sum_{i=1}^{n} \frac{\propto_i}{p_i - s}\left[M(p_i) \sum_{j=1}^{n} \frac{\propto_j}{p_i + p_j}\phi_j \right] \quad (6)$$

$$= \sum_{i=1}^{n} [-\frac{\propto_i}{p_i - s}\theta_i - W^* M \frac{\propto_i}{s+p_i}\phi_i \quad + M(p_i)\frac{\propto_i}{p_i - s}\sum_{j=1}^{n} \frac{\propto_j}{p_i + p_j}\phi_j] \quad (7)$$

Hence the k^{th} equation, where $k : 1, \ldots ,2n$, is the following:

$$\begin{aligned} 0 = \sum_{i=1}^{n} [& -\frac{\propto_i}{p_i - \beta_k}\theta_i - W^*(\beta_k) M(\beta_k) \frac{\propto_i}{\beta_k + p_i}\phi_i \\ & + M(p_i)\frac{\propto_i}{p_i - \beta_k}\sum_{j=1}^{n} \frac{\propto_j}{p_i + p_j}\phi_j] \end{aligned} \quad (8)$$

Here, 2n unknowns $\phi_1, \ldots, \phi_n$, $\theta_1, \ldots, \theta_n$ and finite number of equations k^{th} equation being (8) are combined in compact form as follows:

$$R_\gamma \delta = 0$$

where R_γ is obtained from this notation according to the $2n$ equations defined by (8) above and,

$$\delta = [\phi_1, \ldots, \phi_n, \theta_1, \ldots, \theta_n]^T$$

We get the desired $R_\gamma \delta = 0$ form by using (8) and construction of R_γ is given in the Matlab code, which can be found in the Appendix.

After obtaining (8) as in the form of $R_\gamma \delta = 0$, an upper ($\gamma_{upper} = ||\mathrm{W}||_\infty$) and lower ($\gamma_{lower} = max|\mathrm{W}(\text{zeros of M})|$) bound for γ should be determined. Then, singular value decomposition is applied on $R_\gamma$by changing γ in $[\gamma_{lower}, \gamma_{upper}]$. As a result, γ_{opt} is the maximum γ value that makes R_γ singular and δ vector is the corresponding vector.

Thereby, non – zero δ is discovered. In this manner, calculation of $G(s)$ and Q_{opt} is straightforward.

Hence, $G(s)$ can be obtained by simply performing $G(s) = \frac{N_G}{D_G}$, using the values of $\phi_1, \ldots, \phi_n$, $\theta_1, \ldots, \theta_n$ in (5).

Recall that, we are interested in obtaining a general form and expression for Q_{opt}. In order to achieve this purpose, we define Q_{opt} and Q_{opt}^{-1}, firstly:

$$F_{opt} = \frac{W(T)G}{G} = \frac{WG - M\Pi_+ M^* WG}{G} = W - \frac{M\Pi_+ M^* WG}{G}$$
$$=> \quad Q_{opt} = \frac{\Pi_+ M^* WG}{G} = \frac{W\phi_i}{G}$$
$$and$$
$$Q_{opt}^{-1} = \left(\frac{G}{\sum_{i=1}^{n} \frac{\propto_i}{s+p_i} \phi_i} \right)$$

The reason that we compute and use Q_{opt}^{-1} is to switch delay terms of Q_{opt} from denominator to numerator. In this way, a simple FIR expression can be obtained. This operation provides us convenience in computation using Matlab's function `getDelayModel()` function.

In the rest of the chapter we examine the structure of Q_{opt} for $M(s) = e^{-hs}$, $h = 0.5$, and show that:

$$Q_{opt} = \frac{Q_1}{1 + H_{opt}} = \frac{Q_1}{1 + (Q_2 + H_{opt-FIR})} = \frac{Q_1 Q_F}{1 + Q_F H_{opt-FIR}}$$

where Q_1, Q_2 are finite dimensional, $Q_2 \epsilon H_\infty$ and $Q_F = \frac{1}{1+Q_2}$

And also,

$$H_{opt-FIR}(s) = \mathcal{L}\{h(t)\}$$
$$h(t) \; is \; FIR \;\; i.e. \;\; h(t) \begin{Bmatrix} = 0 \; for \; t > h \\ \neq 0 \; for \; 0 \leq t \leq h \end{Bmatrix} \tag{9}$$

At this point, our main goal is to reveal the general form of Q_1, Q_2 and $H_{opt-FIR}$.

As first step, $H_{opt-FIR}$ can be also defined in the following general form since $h(t)$ is FIR:

$$H_{opt-FIR}(s) = \quad C(sI - A)^{-1}B - Ce^{Ah}e^{-hs}(sI - A)^{-1}B$$

where A, B, C are the matrices of the state – space model to construct $H_{opt-FIR}$

Remark 1: In order to see the fact defined above, we can define $h(t)$ from (9) as following:

$$\begin{aligned} h(t) &= Ce^{At}Bu(t) - Ce^{At}Bu(t-h) \\ &= Ce^{At}Bu(t) - Ce^{Ah}e^{-At}e^{At}Bu(t-h) \\ &= Ce^{At}Bu(t) - Ce^{Ah}e^{A(t-h)}Bu(t-h) \end{aligned}$$

Then,

$$H_{opt-FIR}(s) = \mathcal{L}\{h(t)\} = C(sI-A)^{-1}B - Ce^{Ah}e^{-hs}(sI-A)^{-1}B$$

where

$$\begin{aligned} H_{opt-Filter}(s) &= C(sI-A)^{-1}B \\ H_{opt-delay}(s) &= Ce^{Ah}e^{-hs}(sI-A)^{-1}B \end{aligned}$$ ■

From the Remark 1, the non – zero part of $h(t)$ comes from:

$$h_{non-zero}(t) = \mathcal{L}^{-1}\{H_{opt-Filter}(s)\}$$

Thus, the inverse Laplace transform of an $H_{FIR}(s)$ is the impulse response of the filter, which can be obtained from the delay – free part of $H_{FIR}(s)$, on the time interval identified by time delay term.

From here, by using D_G and (6) form of N_G and definition of Q_{opt}^{-1}, following computation procedure is applied in order to reach our goal:

$$\begin{aligned} G(s) &= \frac{N_G}{D_G} \\ &= \frac{-W^*M\sum_{i=1}^{n}\frac{\propto_i}{s+p_i}\phi_i - \sum_{i=1}^{n}\frac{\propto_i}{p_i-s}\theta_i + \sum_{i=1}^{n}\frac{\propto_i}{p_i-s}\left[M(p_i)\sum_{j=1}^{n}\frac{\propto_j}{p_i+p_j}\phi_j\right]}{\gamma^2 - W^*W} \end{aligned}$$

Here we define,

$$\mu_i = \left(M(p_i) \sum_{j=1}^{n} \frac{\propto_j}{p_i + p_j} \phi_j - \theta_i \right)$$

and,

$$\frac{1}{\gamma^2 - W^*W} = \frac{\frac{1}{\gamma^2}}{1 - \frac{W^*W}{\gamma^2}} = \left(\frac{1}{\gamma^2}\right) \left[1 + \frac{\frac{W^*W}{\gamma^2}}{1 - \frac{W^*W}{\gamma^2}} \right]$$

Thus, Q_{opt}^{-1} takes form as following:

$$Q_{opt}^{-1} = \left(\frac{1}{\gamma^2 - W^*W}\right) \left[\frac{\sum_{i=1}^{n} \frac{\propto_i}{p_i - s} \mu_i}{\sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i} - W^*M \right]$$

$$= \left(\frac{1}{\gamma}\right) \left[1 + \frac{\frac{W^*W}{\gamma^2}}{1 - \frac{W^*W}{\gamma^2}} \right] \left[\frac{\frac{1}{\gamma} \sum_{i=1}^{n} \frac{\propto_i}{p_i - s} \mu_i}{\sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i} - \frac{1}{\gamma} W^*M \right]$$

$$= \left(\frac{1}{\gamma}\right) \left[1 + \frac{\frac{W^*W}{\gamma^2}}{1 - \frac{W^*W}{\gamma^2}} \right] \left[1 + \frac{\sum_{i=1}^{n} \frac{\propto_i}{p_i - s} \frac{\mu_i}{\gamma} - \sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i}{\sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i} - \frac{1}{\gamma} W^*M \right]$$

$$= \left(\frac{1}{\gamma}\right) \left[1 + \frac{\frac{W^*W}{\gamma^2}}{1 - \frac{W^*W}{\gamma^2}} + \left(\frac{1}{1 - \frac{W^*W}{\gamma^2}}\right) \left(\frac{\sum_{i=1}^{n} \frac{\propto_i}{p_i - s} \frac{\mu_i}{\gamma} - \sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i}{\sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i} \right) - \left(\frac{1}{1 - \frac{W^*W}{\gamma^2}}\right) \left(\frac{W^*M}{\gamma}\right) \right]$$

$$= \left(\frac{1}{\gamma}\right) \left[1 + \left(\frac{1}{1 - \frac{W^*W}{\gamma^2}}\right) \left(\frac{W^*W}{\gamma^2} + \left(\frac{\sum_{i=1}^{n} \frac{\propto_i}{p_i - s} \frac{\mu_i}{\gamma} - \sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i}{\sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i} \right) - \left(\frac{W^*M}{\gamma}\right) \right) \right]$$

In this manner,

$$Q_{opt} = \frac{\gamma}{1 + \left[\left(\frac{1}{1 - \frac{W^*W}{\gamma^2}}\right) \left(\frac{W^*W}{\gamma^2} + \left(\frac{\sum_{i=1}^{n} \frac{\propto_i}{p_i - s} \frac{\mu_i}{\gamma} - \sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i}{\sum_{i=1}^{n} \frac{\propto_i}{s + p_i} \phi_i} \right) - \left(\frac{W^*M}{\gamma}\right) \right) \right]}$$

where $K_{opt}(s) \coloneqq \left(\frac{\sum_{i=1}^{n} \frac{\propto_i \mu_i}{p_i - s\ \gamma} - \sum_{i=1}^{n} \frac{\propto_i}{s+p_i} \phi_i}{\sum_{i=1}^{n} \frac{\propto_i}{s+p_i} \phi_i} \right)$ is strictly proper

At this point, $H_{opt}(s)$ come in sight as at below:

$$\begin{aligned} H_{opt}(s) &:= \left(\frac{1}{1 - \frac{W^*W}{\gamma_{opt}^2}} \right) \left(\frac{W^*W}{\gamma_{opt}^2} + K_{opt} - \left(\frac{W^*M}{\gamma_{opt}} \right) \right) \\ &= H_{opt-delayfree}(s) + H_{opt-delay}(s) \end{aligned}$$

Herein, if we examine $H_{opt}(s)$ partially, it is easy to comprehend that $H_{opt}(s)$ consists two parts as $H_{opt-delayfree}(s)$ and $H_{opt-delay}(s)$. Hence, the $H_{opt-delay}(s)$ part of $H_{opt}(s)$ is takes form as following:

$$H_{opt-delay}(s) := \left(\frac{1}{1 - \frac{W^*W}{\gamma_{opt}^2}} \right) \left(\frac{W^*M}{\gamma_{opt}} \right) = \left(\frac{\gamma_{opt} W^*M}{\gamma_{opt}^2 - W^*W} \right)$$

As can be seen, the denominator of $H_{opt-delay}(s)$ is equal to D_G which is already defined before. In this manner, poles of $H_{opt-delay}(s)$ are equal to β_k values exactly since the zeros of D_G are equal to β_k values.

Now, we examine the following:

$$H_{opt-delayfree}(s) = H_{opt}(s) - H_{opt-delay}(s)$$

Moreover, from $H_{opt-delay}(s)$ we can construct a compatible $H_{opt-Filter}(s)$ such that $\left(H_{opt-Filter}(s) + H_{opt-delay}(s) \right)$ is FIR by using Remark 1. Then we define:

$$Q_2 = H_{opt-delayfree}(s) - H_{opt-Filter}(s)$$

In this respect, $H_{opt-delayfree}(s)$ can be decomposed to its residues. By comparing poles of these residues with β_k values, separate Q_2 and $H_{opt-Filter}(s)$ will be obtained. It is straightforward that $H_{opt-Filter}(s)$ has

the same order with $H_{opt-delay}(s)$ and Q_2 must be a proper, stable and rational transfer function.

Overall, $H_{opt-FIR}$ and Q_2 come in sight and we have and Q_{opt} as in the following form:

$$\begin{aligned} Q_{opt} &= \frac{\gamma_{opt}}{1 + H_{opt}} \\ &= \frac{\gamma_{opt}}{1 + (H_{opt-delayfree} + H_{opt-delay})} \\ &= \frac{\gamma_{opt}}{1 + (Q_2 + H_{opt-Filter} - H_{opt-Delay})} \\ &= \frac{\gamma_{opt}}{1 + (Q_2 + H_{opt-FIR})} \end{aligned}$$

Lastly, by using $Q_F = \frac{1}{1+Q_2}$, we have:

$$Q_{opt} = \frac{\gamma_{opt} Q_F}{1 + Q_F H_{opt-FIR}} \qquad \blacksquare$$

This final result also yields the structures of Q_1 and Q_2 as at below:

$$Q_1 = \gamma_{opt}$$
$$Q_2 = a\ proper, stable\ and\ rational\ transfer\ function$$

In most cases, obtaining G(s), Q_{opt} and Q_{opt}^{-1} are difficult by hand calculation because of complex sum operations, matrix computations, applying the bound of γ for singular value decomposition and so forth. At the same time, we are interested in a tool which can compute Q_{opt} and give us the structural result.

With the computation power of Matlab, G(s), Q_{opt} and Q_{opt}^{-1} can be found by the procedure indicated in Sarason's Theorem explicitly. Obviously, G(s) and Q_{opt} have time delay term or terms since we deal with infinite dimensional systems. In other respects, Q_{opt} has H_{opt} at denominator so that delay

term or terms exist at the denominator of Q_{opt}. Thus, results for G(s), Q_{opt} and Q_{opt}^{-1} are all in state-space model in Matlab.

Up to here, we have already determined the main result as $Q_{opt} = \frac{\gamma_{opt} Q_F}{1+Q_F H_{opt-FIR}}$. Obtaining the structures of $H_{opt-FIR}$ and Q_F are sufficient to reveal the structural result for Q_{opt}. In order to achieve this purpose, we need to use Q_{opt}^{-1} to have delay part at the numerator, firstly. When Q_{opt}^{-1} is obtained, finding H_{opt} in state – space model is trivial since the process in here is simply multiplying with γ_{opt} and subtracting 1.

Hereby, H_{opt} is obtained in state – space form. As indicated before, $H_{opt}(s)$ has two main parts as $H_{opt-delayfree}(s)$ and $H_{opt-delay}(s)$.

Therefore, $H_{opt}(s)$ can be decomposed as delay – free state space model and internal delay vector, which is the time interval indicator, by using Matlab. In order to do this process, `getDelayModel()` function of Matlab is used. This function gives us a delay – free state – space model and internal delay vector. In this way, we can easily construct $H_{opt-delayfree}(s)$ and $H_{opt-delay}(s)$ and have $H_{opt}(s)$ as it is. Illustration and working principle of this decomposition can be seen at following diagram [30]:

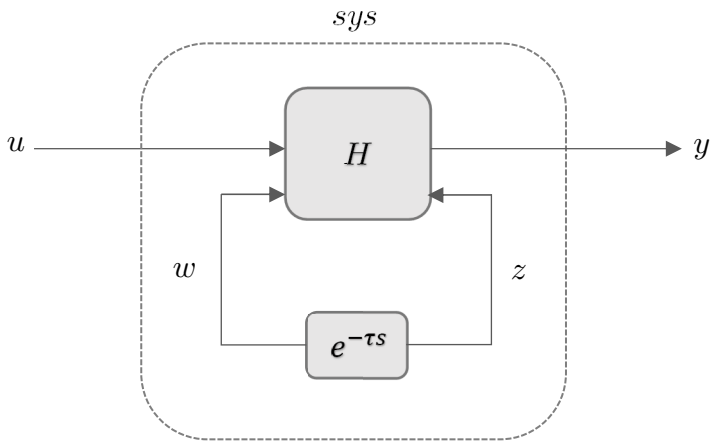

Figure 3.1 MATLAB's `getDelayModel()` function structure

After $H_{opt-delayfree}(s)$ and the time interval indicator is obtained, the residue decomposition on $H_{opt-delayfree}(s)$ is applied and Q_2 is found. Thus, Q_F is also found by implementing $Q_F = \frac{1}{1+Q_2}$. On the other hand, $H_{opt-Filter}(s)$ is automatically revealed since $Q_2 = H_{opt-delayfree}(s) - H_{opt-Filter}(s)$.

Thereafter, providing the structural result for $H_{opt-FIR}(s)$ is just simple by using $H_{opt-Filter}(s) = C(sI - A)^{-1}B$ and then $H_{opt-FIR}(s) = C(sI - A)^{-1}B - Ce^{Ah}e^{-hs}(sI - A)^{-1}B$ since we have A, B and C as state – space matrices from $H_{opt-Filter}(s)$.

At this point, it is important to implement $C(sI - A)^{-1}B - Ce^{Ah}e^{-hs}(sI - A)^{-1}B$ as two decomposed parts which are $C(sI - A)^{-1}B$ and $Ce^{Ah}e^{-hs}(sI - A)^{-1}B$. Otherwise, the result will end up with state – space model in Matlab, again.

As a final result, the structural result for Q_{opt} is discovered by simply substituting the revealed $H_{opt-FIR}(s)$ and Q_F structures, γ_{opt} and 1.

Chapter 4

Examples

In this part of the thesis, some examples will be given regarding the infinite dimensional one-block H_∞ control problem:

$$\gamma_{opt} = \inf_{Q \epsilon H_\infty} ||W - MQ||_\infty = ||F_{opt}||_\infty$$

where $W(s)$ is finite dimensional, and $M(s) = e^{-hs}$ a time delay so that it makes our problem infinite dimensional one-block H_∞ control problem.

Our main goal is to obtain general structure for Q_{opt}. In all examples, calculation process represented in the previous chapter will be applied. Thus, γ_{opt} and $G(s)$ will be found firstly. As obvious, Q_{opt} can be calculated after obtaining γ_{opt} and $G(s)$ since $Q_{opt} = \frac{\Pi_+ M^* W G}{G}$. Lastly, structural result for Q_{opt} will be provided as $Q_{opt} = \frac{\gamma_{opt}}{1+(Q_2+H_{opt-FIR})} = \frac{\gamma_{opt} Q_F}{1+Q_F H_{opt-FIR}}$ by applying the computation process indicated at previous part.

Therefore, our main objective is to find γ_{opt}, Q_F and $H_{opt-FIR}$ in each example.

4.1 First Order Weight W

Let $W(s) = \frac{1}{s+0.1}\ \epsilon\ H_\infty$ and $M(s) = e^{-0.5s}\ \epsilon\ H_\infty$.

In order to produce different examples and structures, Bode diagram of $W(s)$ will be examined for all examples. For this example, $W(s)$ has Bode diagram as following:

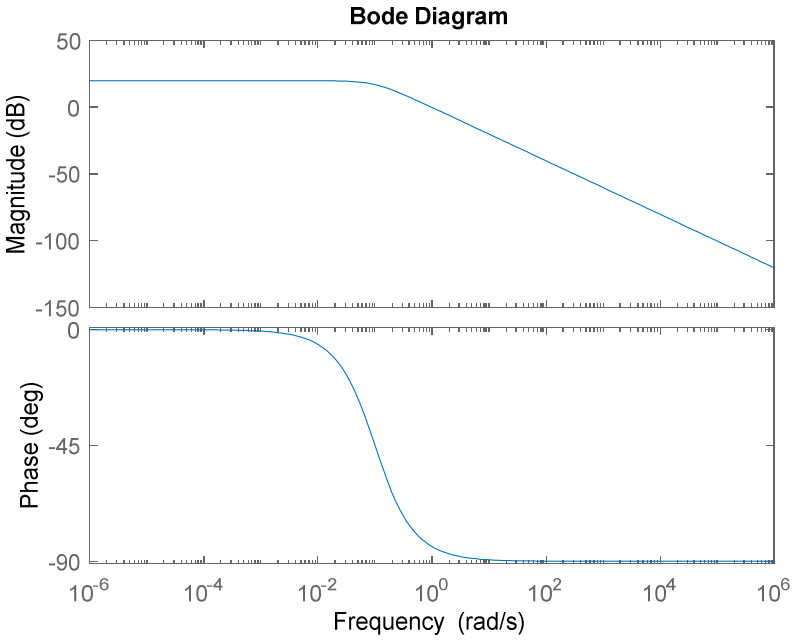

Figure 4.1 Bode Diagram of $W(s)$ (1st Order Case)

After that, bound for γ is calculated as [0, 10] from

$$[\ \gamma_{lower} = max|\mathrm{W}(\text{zeros of M(s)})|\ , \gamma_{upper} = ||\mathrm{W}||_\infty\ ]$$

On the other hand, by performing very small increase rate on the bound of γ, $D_G = \gamma^2 - W^*W = 0$ is implemented and $\beta_k{}'s$ are obtained every time. Then, β_k values are substituted to N_G one by one in order to obtain R_γ.

$N_G(\beta_k) = 0$ is the k^{th} equation, which is defined as (8), in $R_\gamma \delta = 0$. So, R_γ is computed at every step and singular value decomposition is applied on it to compute singular vector δ, γ_{opt} is the largest γ for which R_γ is singular. Following figure illustrates this process:

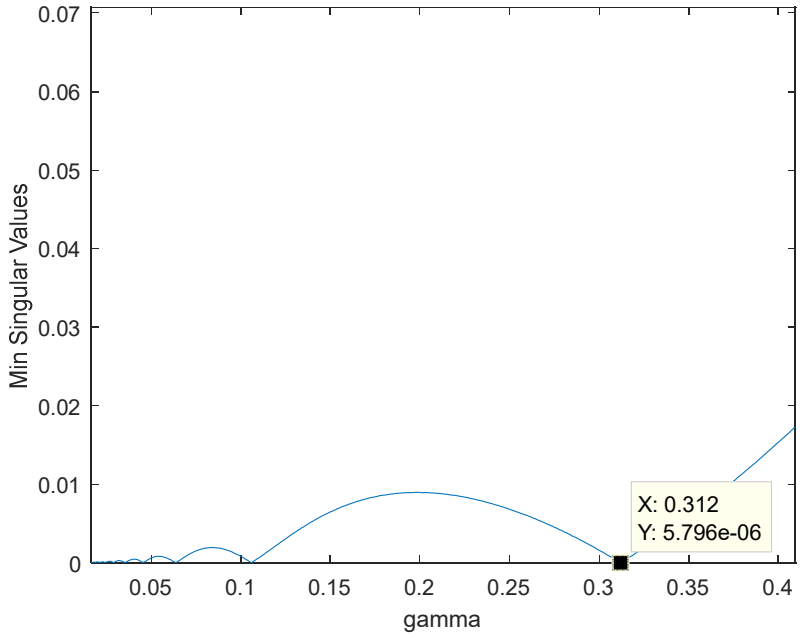

Figure 4.2 Minimum Singular Values of R_γ versus γ (1^{st} Order Case)

Therefore, γ_{opt} equal to 0.312 and corresponding β_k values are $\{3.2036\text{j}, -3.2036\text{j}\}$. After that, the singular vector δ is obtained as following:

$$\delta = \begin{bmatrix} \phi_1 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} -0.1936 \\ -0.9811 \end{bmatrix}$$

With this result, $G(s)$ can be easily found since δ and γ_{opt} help us to find N_G and D_G respectively. After finding $G(s)$, $Q_{opt}(s)$ can be found since $Q_{opt} = \frac{\Pi_+ M^* W G}{G}$. Results for $G(s)$ and Q_{opt} are obtained in state – space form and Bode plots for both of them provided:

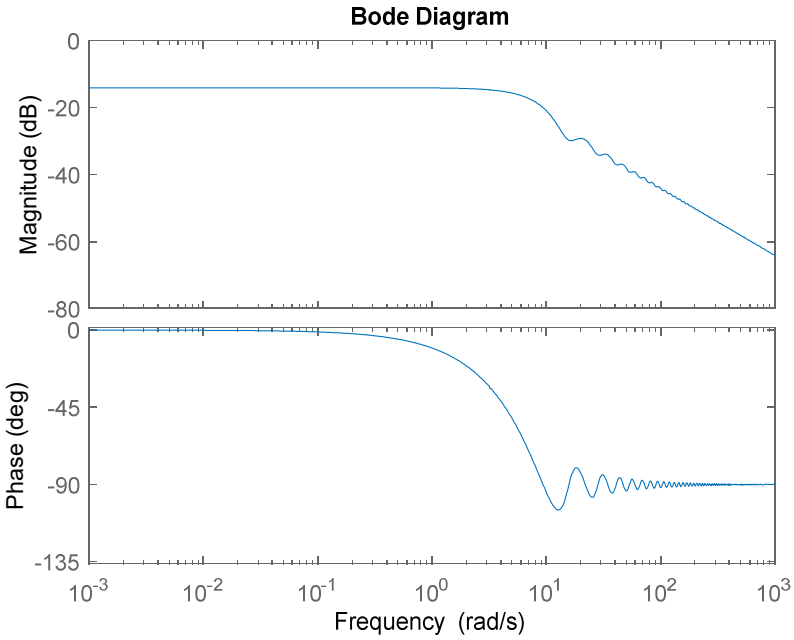

Figure 4.3 Bode plot for $G(s)$ (1^{st} Order Case)

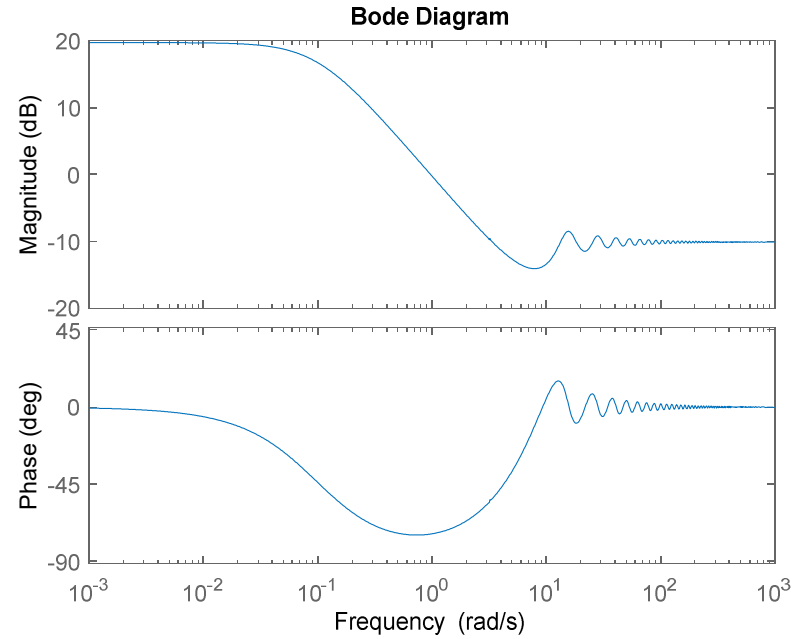

Figure 4.4 Bode plot for Q_{opt} (1^{st} Order Case)

We have no clue about the structure of Q_{opt} up to now, since we do not perform hand calculation and we obtained the results in state – space form.

Thereafter, we will try to find the structure of Q_{opt}. We know that the general form of Q_{opt} is $Q_{opt} = \frac{\gamma_{opt} Q_F}{1+Q_F H_{opt-FIR}}$. It is straightforward to obtain $H_{opt}(s)$ by taking inverse of Q_{opt}. Then we have $H_{opt}(s)$ in the state – space form. The Bode plot of it is provided as at below:

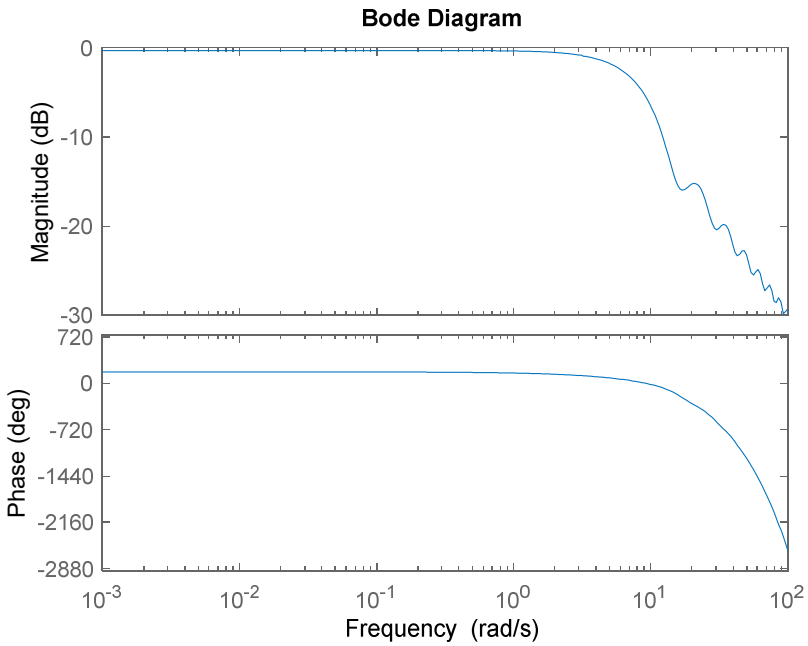

Figure 4.5 Bode plot for H_{opt} (1^{st} Order Case)

At this point, what should be done is obtaining the delay – free part of $H_{opt}(s)$. From the procedure indicated and mentioned in Chapter 3, delay – free part of $H_{opt}(s)$ is computed as:

$$H_{opt-delayfree}(s) = \frac{-2.0846 * 10^{-8} \ (\mathrm{s} - 9.594 * 10^{6}) \ (\mathrm{s} - 51.26)}{(s^2 \ + \ 10.26)}$$
$$\cong \frac{0.2 \ (\mathrm{s} - 51.26)}{(s^2 \ + \ 10.26)}$$

As can be seen from above result, $H_{opt-delayfree}(s)$ has poles which are exactly equal to β_k values. In this manner, $Q_2 = 0$ which gives us following equalities for this example:

$$H_{opt-Filter}(s) = H_{opt-delayfree}(s)$$
$$H_{opt}(s) = H_{opt-FIR}(s)$$
$$Q_F = 1$$

By simply converting $H_{opt-Filter}(s)$ to state – state form we obtain A, B and C as state – space matrices of $H_{opt\_FIR}(s)$. The rest is implementing the following:

$$H_{opt-FIR}(s) = C(sI - A)^{-1}B - Ce^{Ah}e^{-hs}(sI - A)^{-1}B$$

As stated in previous part, $H_{opt-FIR}(s)$ is put into process as two separate parts, which are $C(sI - A)^{-1}B$ and $Ce^{Ah}e^{-hs}(sI - A)^{-1}B$.

Obviously, $C(sI - A)^{-1}B$ part indicates the delay – free part of $H_{opt-FIR}(s)$, which is the filter $H_{opt-Filter}(s)$ itself. From $Ce^{Ah}e^{-hs}(sI - A)^{-1}B$ part we obtain the delay part of $H_{opt-FIR}(s)$ which is the $H_{opt-delay}(s)$. Therefore, we separately get all parts of $H_{opt-FIR}(s)$. This result is sufficient to reveal $H_{opt-FIR}(s)$'s and also $H_{opt}(s)$'s structure in this example:

$$H_{opt-delay}(s) = \frac{-3.2051\ (\mathrm{s} + 0.1007)e^{-0.5s}}{(s^2\ +\ 10.26)} \cong \frac{-3.2051\ (\mathrm{s} + 0.1)e^{-0.5s}}{(s^2\ +\ 10.26)}$$

$$H_{opt}(s) = H_{opt-delayfree}(s) - H_{opt-delay}(s)$$
$$H_{opt}(s) = Q_2 + H_{opt-Filter}(s) - H_{opt-delay}(s)$$
$$H_{opt}(s) = H_{opt-FIR}(s) = H_{opt-Filter}(s) - H_{opt-delay}(s)$$
$$=> H_{opt}(s) = H_{opt-FIR}(s) = \frac{0.2\ (\mathrm{s} - 51.26) + 3.2051\ (\mathrm{s} + 0.1)e^{-0.5s}}{(s^2\ +\ 10.26)}$$

Although the denominator expression of $H_{opt}(s)$ has roots on the imaginary axis, these get cancelled by the zeros of the numerator, which are the β_k values, exactly at the same points. Thereby, $H_{opt}(s)$ is stable which ensures the stability of $Q_{opt}(s)$. The stability of $Q_{opt}(s)$ can be observed from the Nyquist plot of $H_{opt}(s)$. As can be seen at the figure below, there is no

encirclement around s = -1 since there is no pole at the right half plane as at below:

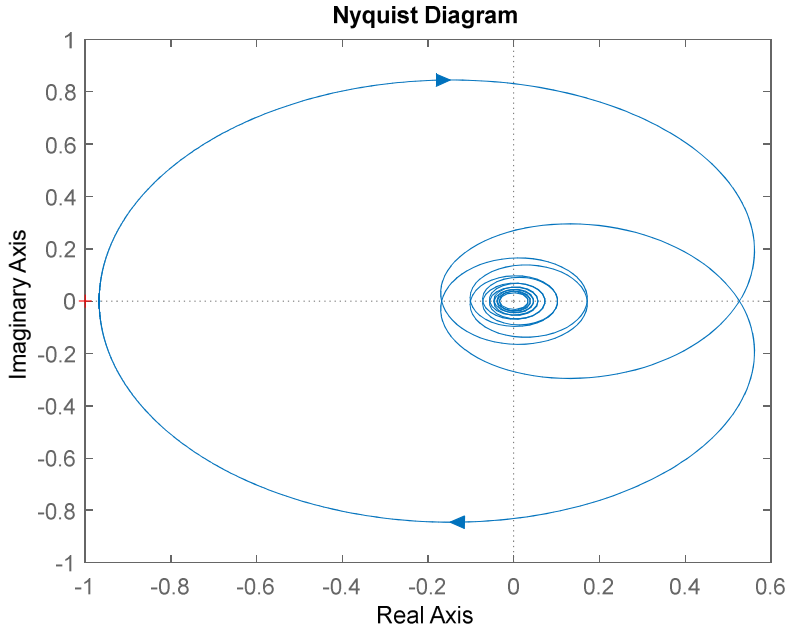

Figure 4.6 Nyquist plot of $H_{opt}(s)$ (1^{st} Order Case)

On the other hand, impulse response of $h(t)$ is observed as expected since $h(t)$ is FIR. It can be seen at the following figure:

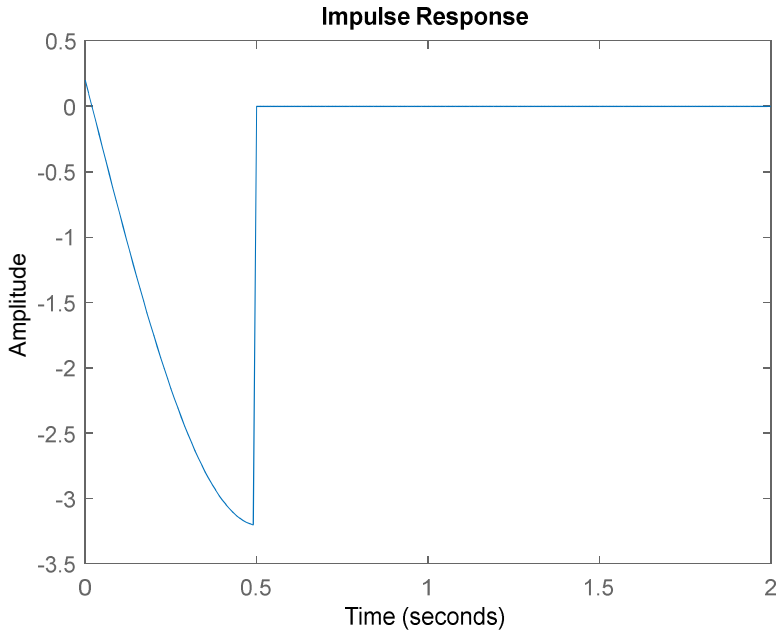

Figure 4.7 Impulse Response for $h(t)$ (1^{st} Order Case)

This result completes the general structure searching for $Q_{opt}(s)$ since we have already determined the general form of $Q_{opt}(s)$ as $Q_{opt} = \frac{\gamma_{opt}}{1+H_{opt}} = \frac{\gamma_{opt}Q_F}{1+Q_F H_{opt-FIR}}$. In this manner, the structural result for Q_{opt} in this example comes forward as following:

$$Q_{opt} = \frac{\gamma_{opt}Q_F}{1+Q_F H_{opt-FIR}}$$

$$= \frac{\gamma_{opt}}{1+H_{opt-FIR}}$$

$$= \frac{0.312}{1+\left(\frac{0.2\ (\mathrm{s}-51.26)+3.2051\ (\mathrm{s}+0.1)e^{-0.5s}}{(s^2\ +\ 10.26)}\right)} \quad \blacksquare$$

This example can be considered main example with detailed explanations. Following examples will be given as only resultative and also used as check-sum of our computation method for different types of $W(s)$.

4.2 Second Order Weight W

Let $W(s) = \frac{1}{(s+0.1)(s+0.05)} \ \epsilon \ H_\infty$ and $M(s) = e^{-0.5s} \ \epsilon \ H_\infty$.

Bode plot for $W(s)$ is the following:

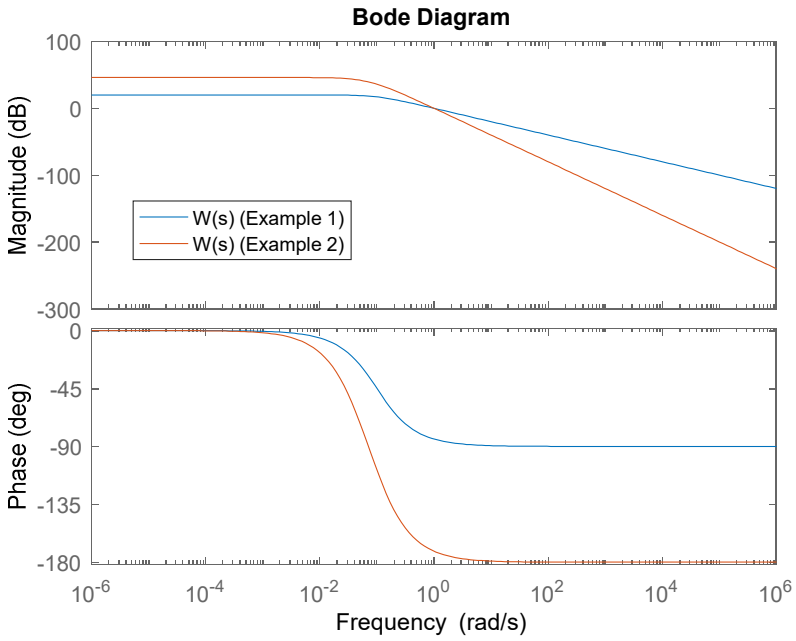

Figure 4.8 Bode Diagram of $W(s)$ (2^{nd} Order Case)

Then, from $[\ \gamma_{lower} = max|\mathrm{W}(\text{zeros of M(s)})|\ , \gamma_{upper} = ||\mathrm{W}||_\infty\ ]$ bound for γ is calculated as [0, 200]

Again, by performing very small increase rate on the bound of γ, R_γ is computed at every step and singular value decomposition is applied on it, γ_{opt} is the largest γ for which R_γ is singular. This procedure is illustrated at the following figure:

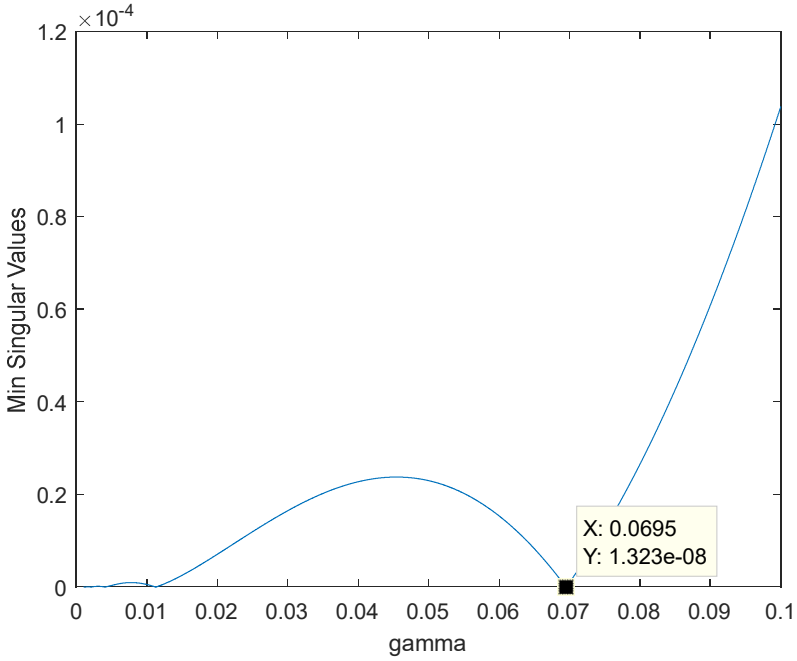

Figure 4.9 Minimum Singular Values versus γ (2^{nd} Order Case)

Thus, γ_{opt} is equal to 0.0695 and corresponding β_k values are $\{3.7943, 3.7926\text{j}, -3.7943, -3.7926\text{j}\}$. Then, the singular vector δ are obtained as following:

$$\delta = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -0.0130 \\ -0.0133 \\ -0.4447 \\ -0.8955 \end{bmatrix}$$

With γ_{opt} and δ vector, $G(s)$ and $Q_{opt}(s)$ are obtained in state – space form, again. The Bode plots for both of them are at below:

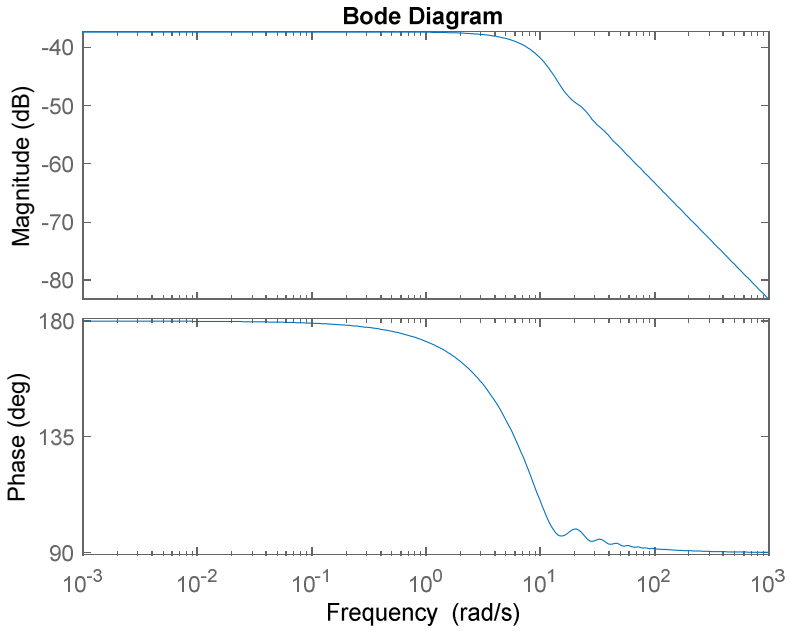

Figure 4.10 Bode plot for $G(s)$ (2^{nd} Order Case)

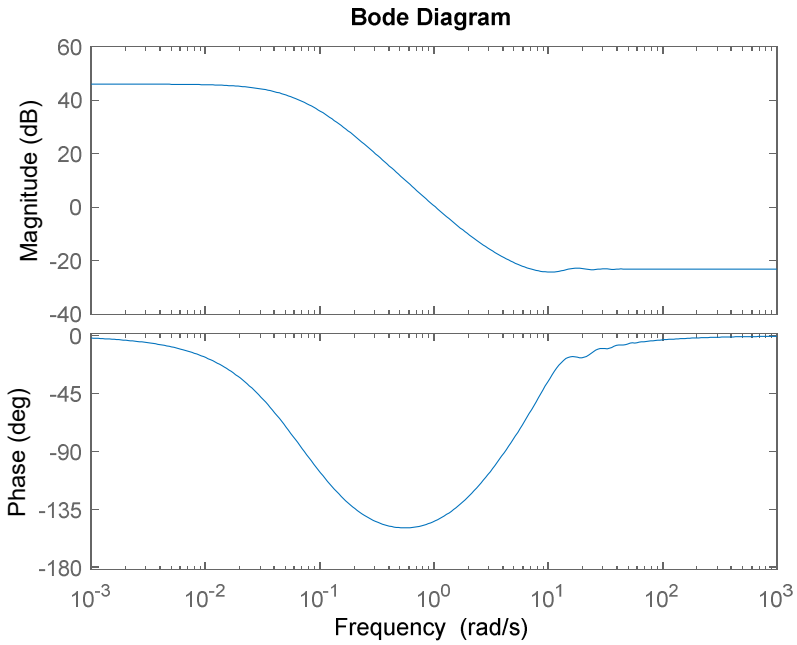

Figure 4.11 Bode plot for $Q_{opt}(s)$ (2^{nd} Order Case)

From here, we can easily obtain $H_{opt}(s)$ by taking inverse of Q_{opt} and have $H_{opt}(s)$ in the state – space form. The Bode plot of $H_{opt}(s)$ is given at following figure:

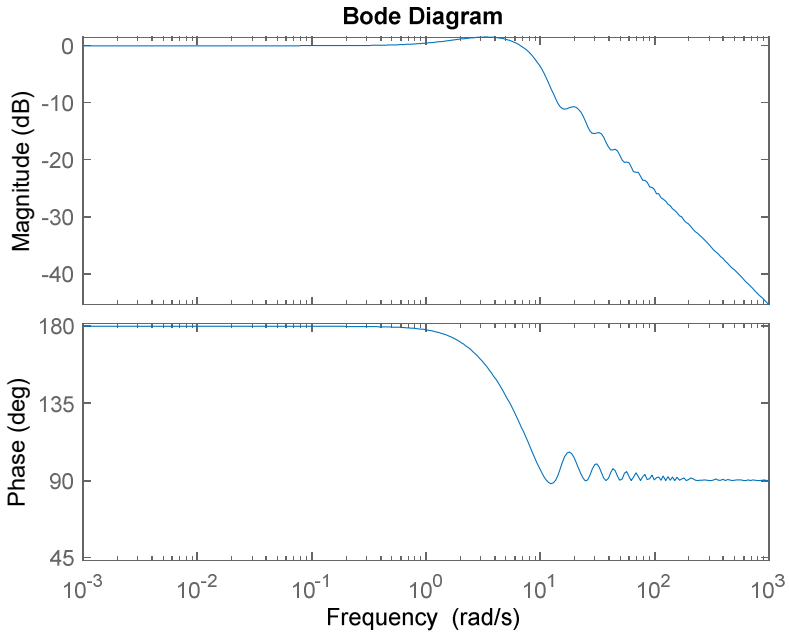

Figure 4.12 Bode plot for H_{opt} (2^{nd} Order Case)

As in the first example, the delay – free part of $H_{opt}(s)$ is computed as:

$$H_{opt-delayfree}(s) \cong \frac{-5.376\ (\mathrm{s}-3.989)\ (\mathrm{s}+2.234)\ (s^2\ +\ 1.905\mathrm{s}\ +\ 12.26)}{(\mathrm{s}-3.794)\ (\mathrm{s}+3.794)\ (\mathrm{s}+2.838)\ (s^2+\ 14.38)}$$

As obvious, the expression of $H_{opt-delayfree}(s)$ has poles exactly at β_k values. There is also an additional negative pole which comes from Q_2. In order to obtain $H_{opt-Filter}(s)$, we should find $Q_2(s)$ from the residues of $H_{opt-delayfree}(s)$ and substract it from $H_{opt-delayfree}(s)$ equation.

Hence,

$$Q_2(s) = \frac{2.3242}{\mathrm{s}+2.838}$$

From $Q_F = \frac{1}{1+Q_2}$ we have:

$$Q_F = \frac{\mathrm{s} + 2.838}{\mathrm{s} + 5.162}$$

Then, by executing $H_{opt-Filter}(s) = H_{opt-delayfree}(s) - Q_2(s)$, we get:

$$H_{opt-Filter}(s) = \frac{-7.003\ (\mathrm{s} - 3.928)(s^2\ +\ 1.195\mathrm{s}\ +\ 12.45)}{(\mathrm{s} - 3.794)\ (\mathrm{s} + 3.794)(s^2 +\ 14.38)}$$

After converting $H_{opt-Filter}(s)$ to state – space form, we get A, B and C state – space matrices to calculate $H_{opt-delay}(s)$ and $H_{opt-FIR}(s)$. With the same implementation on first example, $H_{opt-delay}(s)$ is computed as:

$$\begin{aligned} H_{opt-delay}(s) &= \frac{3.1581 * 10^{-11}\ (\mathrm{s} + 4.557 * 10^{11})(\mathrm{s} + 0.1)(\mathrm{s} + 0.05)\ e^{-0.5s}}{(\mathrm{s} - 3.794)(\mathrm{s} + 3.794)(s^2\ +\ 4.565 * 10^{-13} +\ 14.38)} \\ &\cong \frac{14.39\ (\mathrm{s} + 0.1)(\mathrm{s} + 0.05)\ e^{-0.5s}}{(\mathrm{s} - 3.794)(\mathrm{s} + 3.794)(s^2 +\ 14.38)} \end{aligned}$$

Then,

$$\begin{aligned} &H_{opt-FIR}(s) = H_{opt-Filter}(s) - H_{opt-delay}(s) \\ &= \frac{-7.003\ (\mathrm{s} - 3.928)(s^2\ +\ 1.195\mathrm{s}\ +\ 12.45) - 14.39\ (\mathrm{s} + 0.1)(\mathrm{s} + 0.05)\ e^{-0.5s}}{(\mathrm{s} - 3.794)(\mathrm{s} + 3.794)(s^2 +\ 14.38)} \end{aligned}$$

Thus,

$$\begin{aligned} &H_{opt}(s) = Q_2(s) +\ H_{opt-FIR}(s) \\ &H_{opt}(s) = \\ &\frac{2.3242}{(\mathrm{s} + 2.838)} + \left(\frac{-7.003\ (\mathrm{s} - 3.928)(s^2\ +\ 1.195\mathrm{s}\ +\ 12.45) - 14.39\ (\mathrm{s} + 0.1)(\mathrm{s} + 0.05)\ e^{-0.5s}}{(\mathrm{s} - 3.794)(\mathrm{s} + 3.794)(s^2 +\ 14.38)} \right) \end{aligned}$$

Here, the denominator expression of $H_{opt}(s)$ has roots on the imaginary axis and the right-half plane. These get cancelled by the zeros of the numerator, which are the β_k values, exactly at the same points. On the other hand, $Q_2(s)$ is both proper and stable transfer function. Thereby, $H_{opt}(s)$ is stable

which ensures the stability of $Q_{opt}(s)$. The stability of $Q_{opt}(s)$ can be observed from the Nyquist plot of $H_{opt}(s)$:

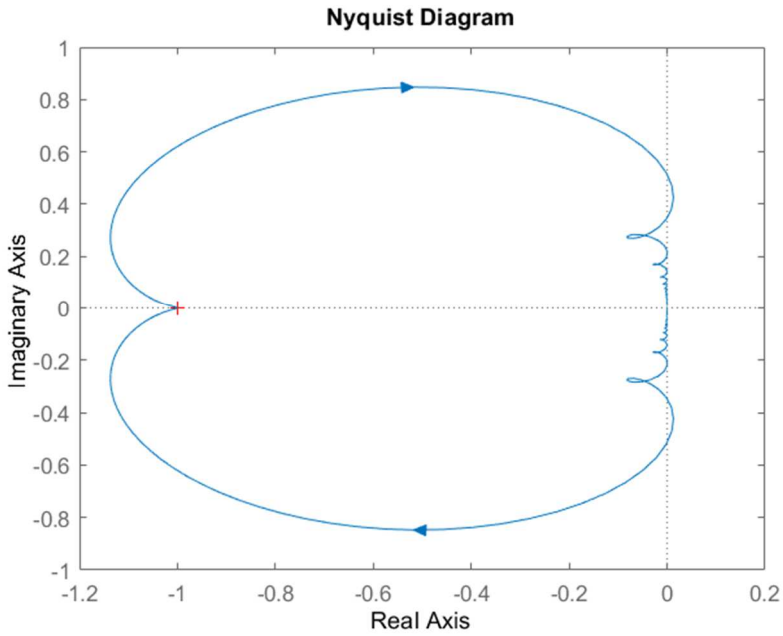

Figure 4.13 Nyquist plot of $H_{opt}(s)$ (2^{nd} Order Case)

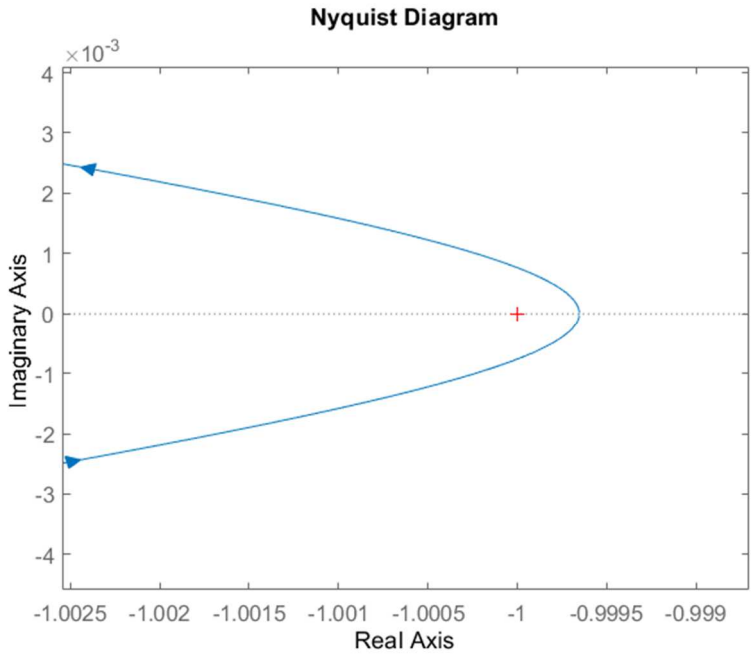

Figure 4.14 Zoomed Nyquist plot of $H_{opt}(s)$ (2^{nd} Order Case)

As can be seen at the figures above, there is no encirclement around s = -1, since there is no unstable pole.

Also, the impulse response of $h(t)$ is observed as FIR which can be seen at figure below:

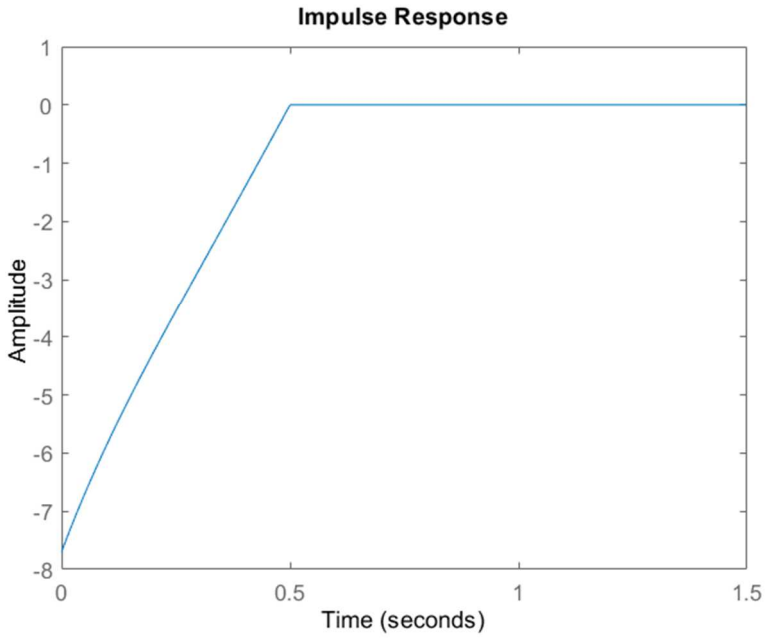

Figure 4.15 Impulse Response for $h(t)$ (2^{nd} Order Case)

This result completes the general structure searching for $Q_{opt}(s)$ since we know the form of $Q_{opt}(s)$ as $Q_{opt} = \frac{\gamma_{opt} Q_F}{1+Q_F H_{opt-FIR}}$. Hence, the structural result of Q_{opt} for this example is computed as at below:

$$Q_{opt} = \frac{\gamma_{opt} Q_F}{1 + Q_F H_{opt-FIR}}$$

$$= \frac{0.0695\left(\frac{\mathrm{s}+2.838}{\mathrm{s}+5.162}\right)}{1+\left(\frac{\mathrm{s}+2.838}{\mathrm{s}+5.162}\right)\left(\frac{-7.003\ (\mathrm{s}-3.928)(s^2\ +\ 1.195\mathrm{s}\ +\ 12.45)-14.39\ (\mathrm{s}+0.1)(\mathrm{s}+0.05)\ e^{-0.5s}}{(\mathrm{s}-3.794)(\mathrm{s}+3.794)(s^2+\ 14.38)}\right)}$$ ■

4.3 Third Order Weight W

Let $W(s) = \frac{1}{(s+0.1)(s+0.05)(s+10)} \ \epsilon \ H_\infty$ and $M(s) = e^{-0.5s} \ \epsilon \ H_\infty$.

Relative Bode plots for $W(s)$ are given in the following figure:

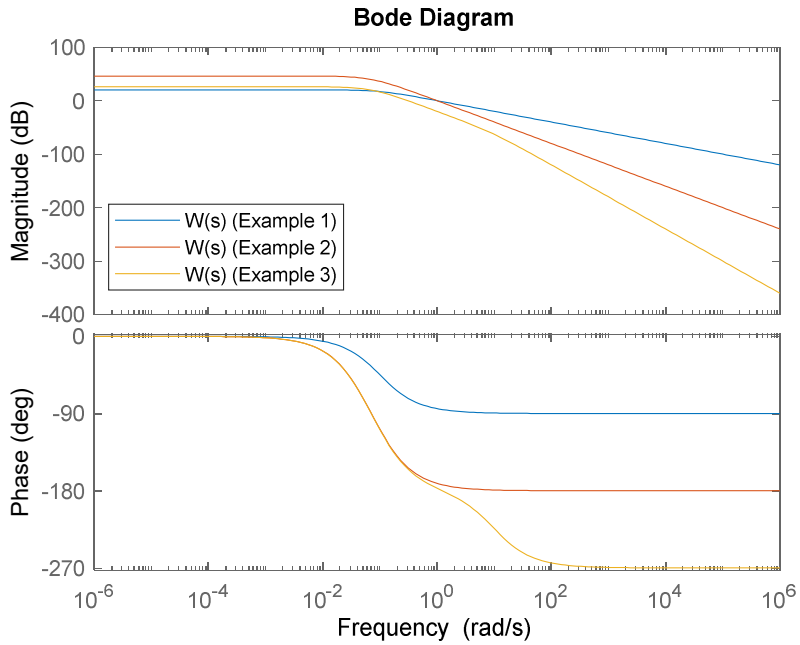

Figure 4.16 Bode Diagram of $W(s)$ (3^{rd} Order Case)

From [$\gamma_{lower} = max|\mathrm{W}(\text{zeros of M(s)})|$, $\gamma_{upper} = ||\mathrm{W}||_\infty$] bound for γ is calculated as [0, 20]

As before, by performing very small increase rate on the bound of γ, R_γ is computed at every step and singular value decomposition is applied on it. This procedure is illustrated at the following figure:

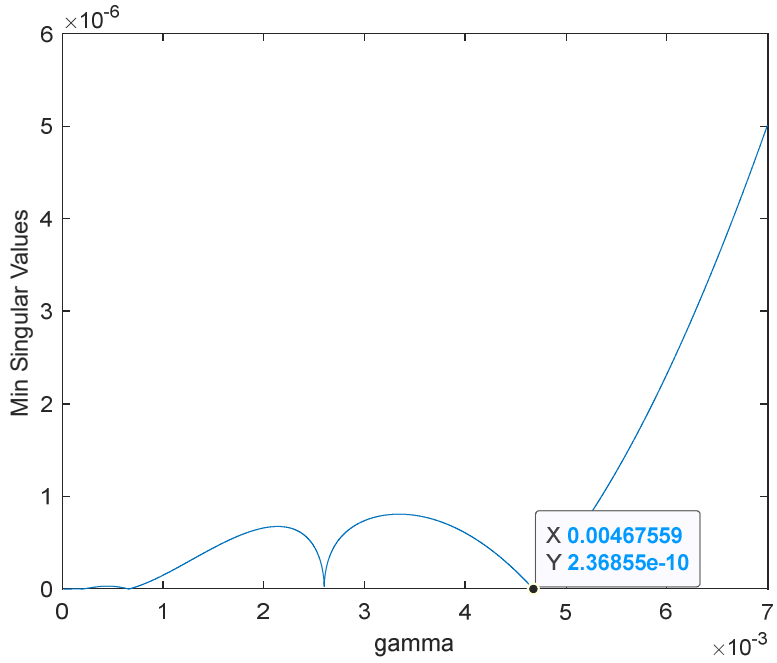

Figure 4.17 Minimum Singular Values versus γ (3^{rd} Order Case)

Thus, γ_{opt} equal to 0.0046756 and corresponding β_k values are $\{9.7428, 4.9644, 4.422\text{j}, -9.7428, -\ 4.9644, -4.422\text{j}\}$. Then, the singular vector δ are obtained as following:

$$\delta = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} 0.0051 \\ 0.1287 \\ 0.1311 \\ 0 \\ 0.4357 \\ 0.8811 \end{bmatrix}$$

With γ_{opt} and δ vector, $G(s)$ and $Q_{opt}(s)$ are obtained in state – space form. The Bode plots for both of them are at below:

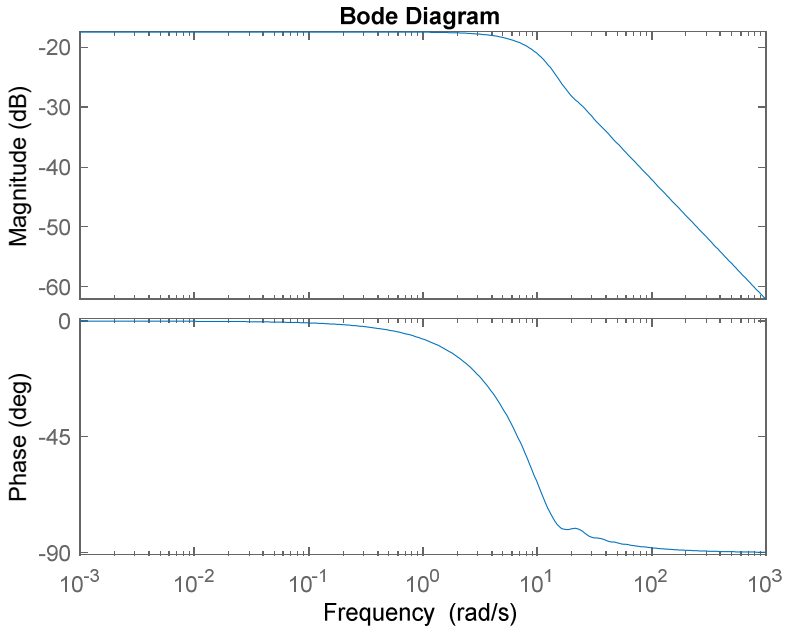

Figure 4.18 Bode plot for $G(s)$ (3^{rd} Order Case)

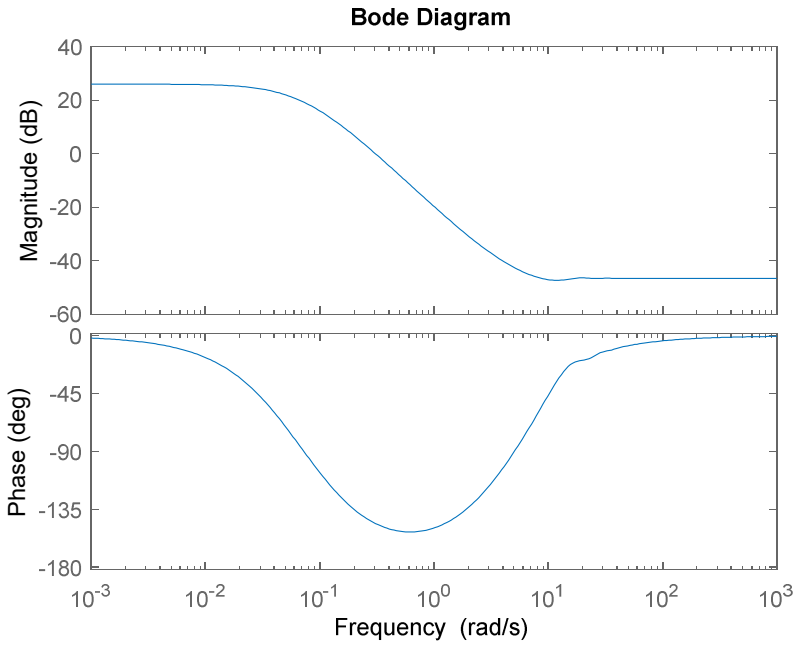

Figure 4.19 Bode plot for $Q_{opt}(s)$ (3^{rd} Order Case)

In here, we can obtain $H_{opt}(s)$ by taking inverse of Q_{opt} and have $H_{opt}(s)$ in the state – space form. The Bode plot of $H_{opt}(s)$ is given at following figure:

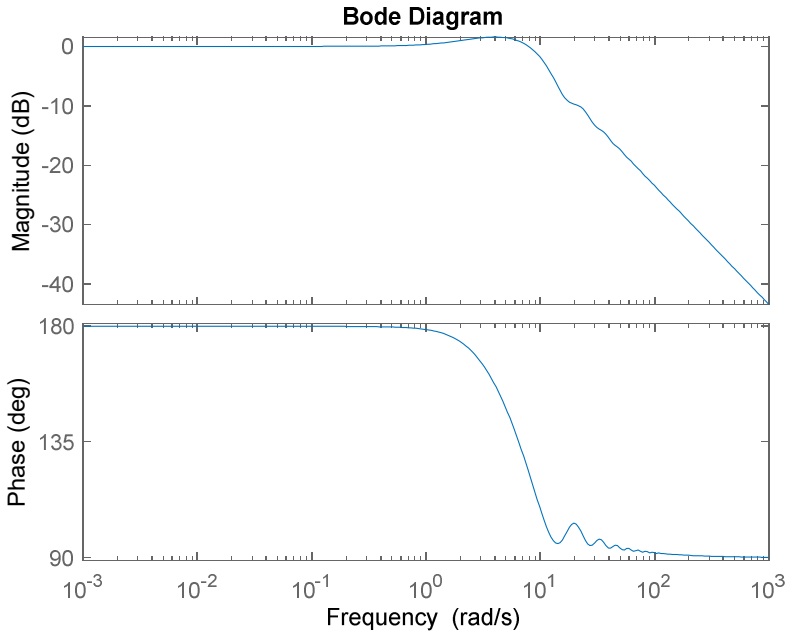

Figure 4.20 Bode plot for H_{opt} (3^{rd} Order Case)

As in the first example, the delay – free part of $H_{opt}(s)$ is computed as:

$$H_{opt-delayfree}(s) = \frac{-6.6902\ (s+2.831)(s-5.267)(s-9.701)(s+9.907)(s+10.48)(s^2\ +\ 1.899s\ +\ 16.54)}{(s+9.78)(s+9.743)(s+4.964)(s+3.715)(s-4.964)(s-9.743)(s^2\ +\ 2.259*10^{-13}s\ +\ 19.56)}$$

$$H_{opt-delayfree}(s) \cong \frac{-6.6902\ (s+2.831)\ (s-5.267)\ (s-9.701)\ (s+9.907)\ (s+10.48)\ (s^2\ +\ 1.899s\ +\ 16.54)}{(s+9.78)(s+3.715)\ (s+9.743)(s-9.743)(s+4.964)\ (s-4.964)(s^2+19.56)}$$

Here, the expression of $H_{opt-delayfree}(s)$ has poles exactly at β_k values. There are also additional negative poles which come from $Q_2(s)$. In order to obtain $H_{opt-Filter}(s)$, we should find $Q_2(s)$ from the residues of $H_{opt-delayfree}(s)$ and substract it from $H_{opt-delayfree}(s)$ equation.

Hence,

$$Q_2(s) = \frac{3.9046}{(\mathrm{s} + 3.715)} - \frac{3.1076}{(\mathrm{s} + 9.78)} = \frac{0.79699(\mathrm{s} + 33.43)}{(\mathrm{s} + 9.78)(\mathrm{s} + 3.715)}$$

From $Q_F = \frac{1}{1+Q_2}$ we have:

$$Q_F = \frac{(\mathrm{s} + 9.78)\ (\mathrm{s} + 3.715)}{(s^2\ +\ 14.29s\ +\ 62.98)}$$

Remark 2: Note that in the first example, we had a first order weight which gave $Q_F = 1$. In the second example, we had second order weight which gave first order Q_F. This example has third order weight giving second order Q_F. This is the general trend.

Then, by executing $H_{opt-Filter}(s) = H_{opt-delayfree}(s) - Q_2(s)$, we get following:

$$H_{opt-Filter}(s) = \frac{-7.4871\ (\mathrm{s} - 5.193)\ (\mathrm{s} - 9.71)\ (\mathrm{s} + 12.84)\ (s^2 +\ 1.197\mathrm{s}\ +\ 16.36)}{(\mathrm{s} + 9.743)(\mathrm{s} - 9.743)(\mathrm{s} + 4.964)\ (\mathrm{s} - 4.964)(s^2 + 19.56)}$$

As next step, we convert $H_{opt-Filter}(s)$ to state – space form and get A, B and C state – space matrices to calculate $H_{opt-delay}(s)$ and $H_{opt-FIR}(s)$. With the same implementation on first example, $H_{opt-delay}(s)$ is computed as following:

$$H_{opt-delay}(s) = \frac{-3.2022 * 10^{-7}\ (s^2 -\ 0.4356s\ +\ 6.679 * 10^8)\ (s + 10)(s + 0.1)(s + 0.04999)\ e^{-0.5s}}{(\mathrm{s} + 9.743)(\mathrm{s} - 9.743)(\mathrm{s} + 4.964)(\mathrm{s} - 4.964)(s^2 + 2.79 * 10^{-13} + 19.56)}$$

$$H_{opt-delay}(s) \cong \frac{-213.8749\ (s + 10)(\mathrm{s} + 0.1)(\mathrm{s} + 0.05)\ e^{-0.5s}}{(\mathrm{s} + 9.743)(\mathrm{s} - 9.743)(\mathrm{s} + 4.964)(\mathrm{s} - 4.964)(s^2 + 19.56)}$$

After that we get:

$$
\begin{aligned}
H_{opt-FIR}(s) &= H_{opt-Filter}(s) - H_{opt-delay}(s) \\
&= \frac{-7.4871\ (\mathrm{s}-5.193)(\mathrm{s}-9.71)(\mathrm{s}+12.84)(s^2+\ 1.197\mathrm{s}\ +\ 16.36)}{(s+9.743)(s-9.743)(s+4.964)(s-4.964)(s^2+19.56)} \\
&\quad + \frac{213.8749\ \big(s+10\big)\big(s+0.1\big)\big(s+0.05\big)\ e^{-0.5s}}{\big(s+9.743\big)\big(s-9.743\big)\big(s+4.964\big)\big(s-4.964\big)\big(s^2+19.56\big)}
\end{aligned}
$$

Thus,

$$
\begin{aligned}
H_{opt}(s) &= Q_2(s) + \ H_{opt-FIR}(s) \\
H_{opt}(s) &= \frac{0.79699\big(s+33.43\big)}{\big(s+9.78\big)\big(s+3.715\big)} \\
&\quad + \frac{-7.4871\ (\mathrm{s}-5.193)(\mathrm{s}-9.71)(\mathrm{s}+12.84)(s^2+\ 1.197\mathrm{s}\ +\ 16.36)}{(s+9.743)(s-9.743)(s+4.964)(s-4.964)(s^2+19.56)} \\
&\quad + \frac{213.8749\ (s+10)(s+0.1)(s+0.05)\ e^{-0.5s}}{(s+9.743)(s-9.743)(s+4.964)(s-4.964)(s^2+19.56)}
\end{aligned}
$$

Once again, the denominator expression of $H_{opt}(s)$ has roots on the imaginary axis and the right-half plane. These get cancelled by the zeros of the numerator, which are the β_k values, exactly at the same points. Also, $Q_2(s)$ is both proper and stable transfer function again. Thereby, $H_{opt}(s)$ is stable which ensures the stability of $Q_{opt}(s)$. The stability of $Q_{opt}(s)$ can be observed from the Nyquist plot of $H_{opt}(s)$:

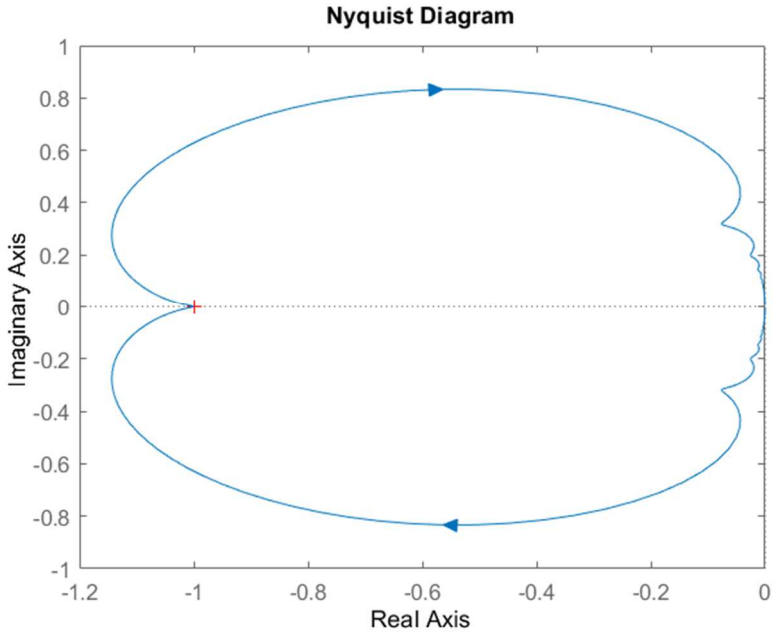

Figure 4.21 Nyquist plot of $H_{opt}(s)$ (3^{rd} Order Case)

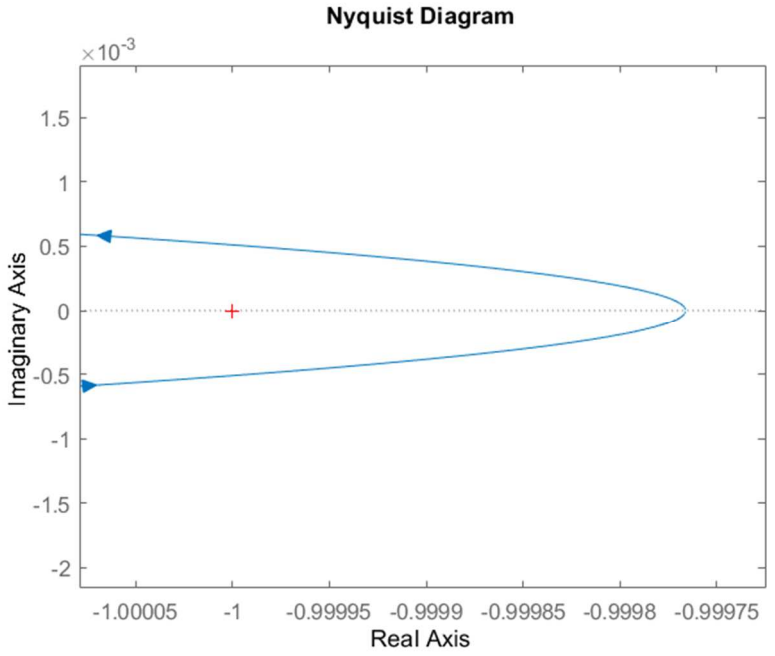

Figure 4.22 Zoomed Nyquist plot of $H_{opt}(s)$ (3^{rd} Order Case)

As can be seen at the figures above, there is no encirclement around s = -1, since there is no unstable pole. Once again, the impulse response of $h(t)$ is observed as FIR again. It can be seen at following figure:

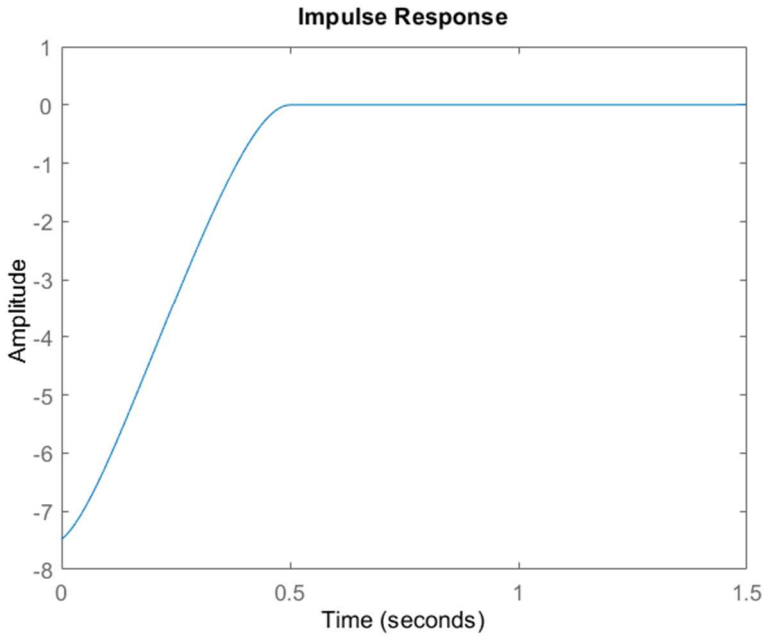

Figure 4.23 Impulse Response for $h(t)$ (3^{rd} Order Case)

This result completes the general structure searching for $Q_{opt}(s)$ since we know the form of $Q_{opt}(s)$ as $Q_{opt} = \frac{\gamma_{opt} Q_F}{1+Q_F H_{opt-FIR}}$. Hence, the structural result of Q_{opt} for this example is computed as at below:

$$Q_{opt} = \frac{\gamma_{opt} Q_F}{1 + Q_F H_{opt-FIR}}$$

$$= \frac{0.0046756 \left(\left(\frac{(s+9.78)\ (s+3.715)}{(s^2\ +\ 14.29s\ +\ 62.98)} \right) \right)}{1 + \left[\begin{array}{c} \left(\frac{(s+9.78)(s+3.715)}{(s^2\ +\ 14.29s\ +\ 62.98)} \right) \\ * [\frac{-7.4871\ (s-5.193)(s-9.71)(s+12.84)(s^2+\ 1.197s\ +\ 16.36)}{(s+9.743)(s-9.743)(s+4.964)(s-4.964)(s^2+19.56)} \\ + \frac{+213.8749\ (s+10)(s+0.1)(s+0.05)\ e^{-0.5s}}{(s+9.743)(s-9.743)(s+4.964)(s-4.964)(s^2+19.56)}] \end{array} \right]} \quad \blacksquare$$

4.4 Second Order Weight W with Larger Magnitude

Let $W(s) = \frac{10(s+1)}{(s+0.1)\ (s+0.05)}\ \epsilon\ H_\infty$ and $M(s) = e^{-0.5s}\ \epsilon\ H_\infty$.

Relative Bode plots for $W(s)$ are given in the following figure:

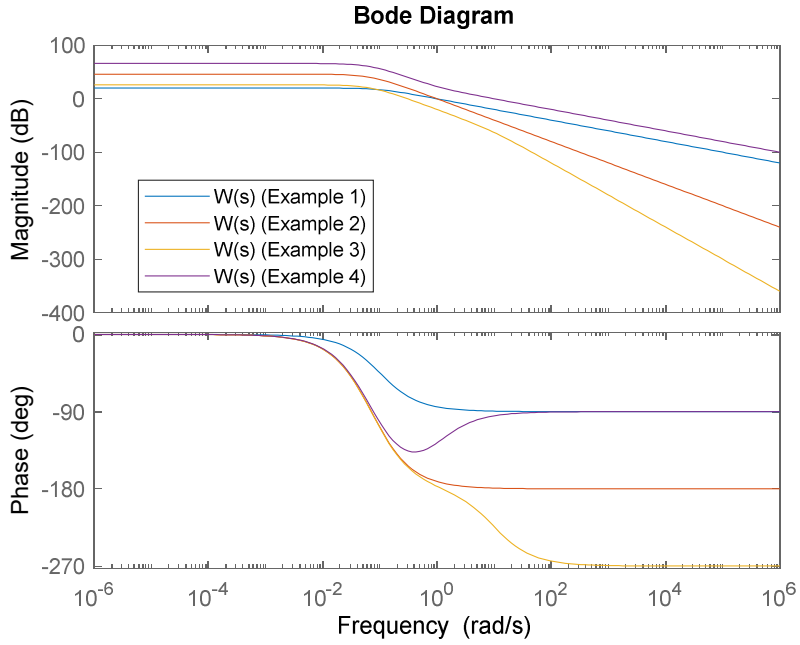

Figure 4.24 Bode Diagram of $W(s)$ (2^{nd} Order Larger Magnitude Case)

After that, from [$\gamma_{lower} = max|\mathrm{W}(\text{zeros of M(s)})|$, $\gamma_{upper} = ||\mathrm{W}||_\infty$] bound for γ is calculated as [0, 2000]

As before, by performing very small increase rate on the bound of γ, R_γ is computed at every step and singular value decomposition is applied on it. This is illustrated at the following figure:

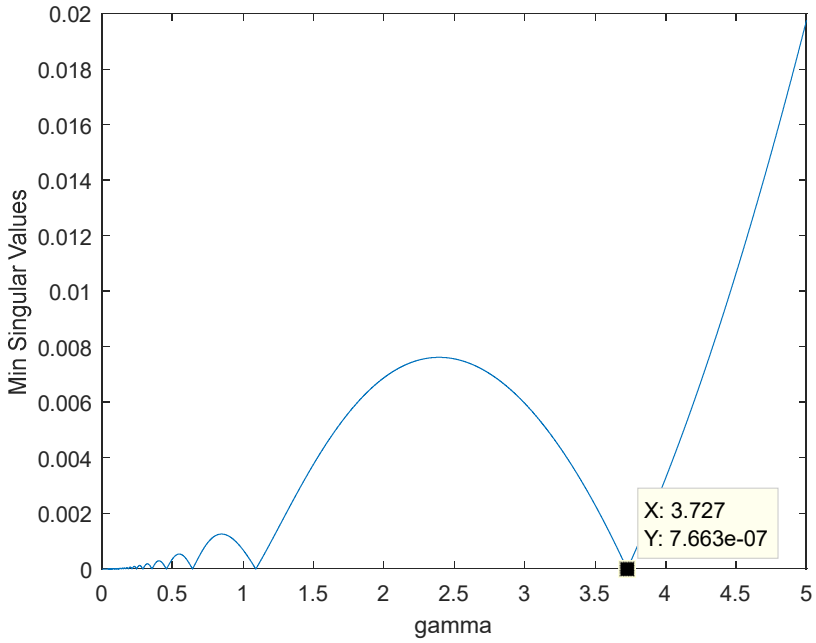

Figure 4.25 Minimum Singular Values versus γ (2^{nd} Order Larger Magnitude Case)

Thus, γ_{opt} is equal to 3.7269 and corresponding β_k values are $\{0.9440, 2.8422\text{j}, -0.9440, -2.8422\text{j}\}$. Then, the singular vector δ are obtained as following:

$$\delta = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 0.0012 \\ 0.0013 \\ 0.4608 \\ 0.8875 \end{bmatrix}$$

With γ_{opt} and δ vector, $G(s)$ and $Q_{opt}(s)$ are obtained in state – space form. The Bode plots for both of them are at below:

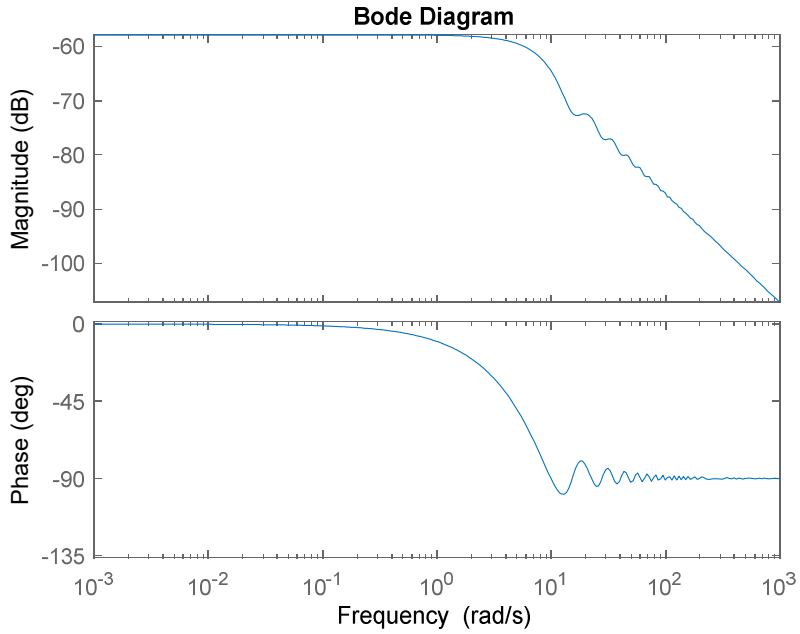

Figure 4.26 Bode plot for $G(s)$ (2^{nd} Order Larger Magnitude Case)

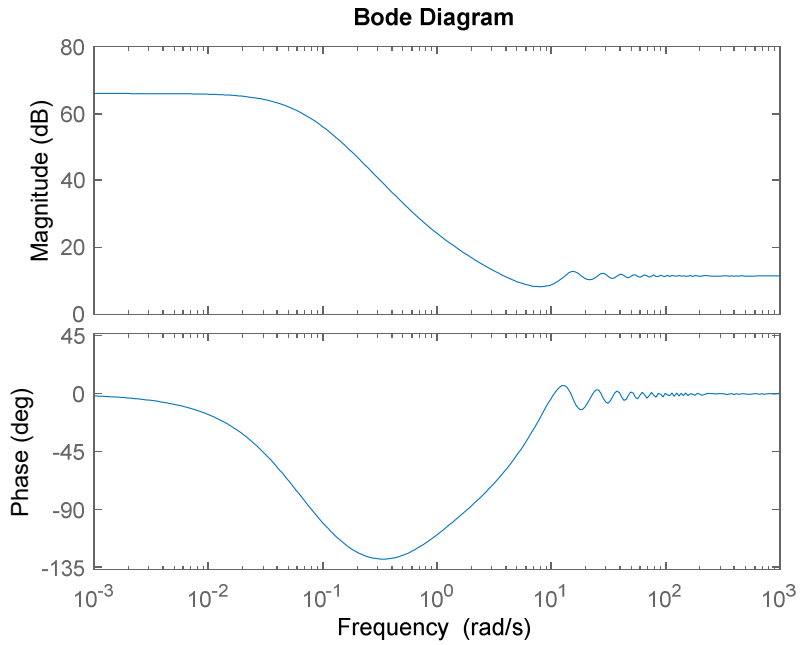

Figure 4.27 Bode plot for $Q_{opt}(s)$ (2^{nd} Order Larger Magnitude Case)

Again, we can easily obtain $H_{opt}(s)$ by taking inverse of Q_{opt} and have $H_{opt}(s)$ in the state – space form. The Bode plot of $H_{opt}(s)$ is given at following figure:

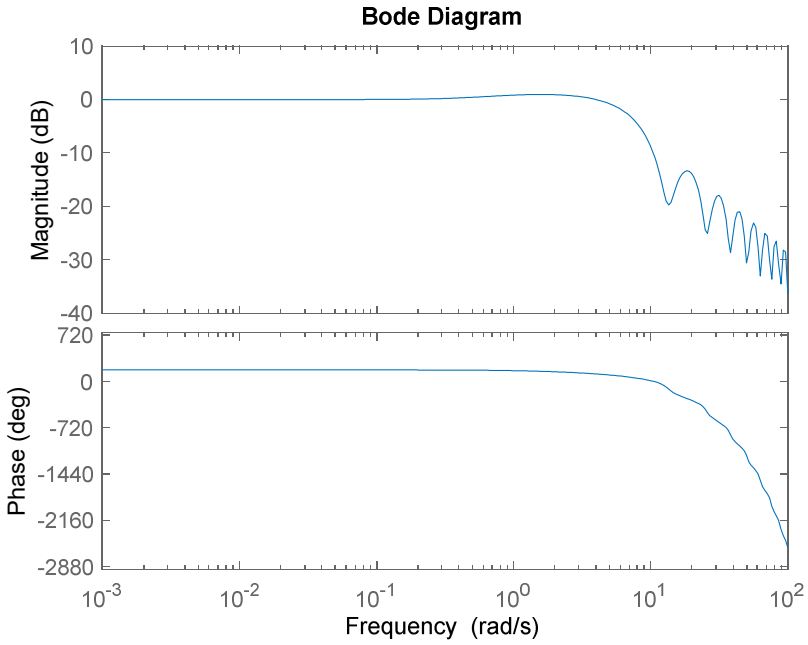

Figure 4.28 Bode plot for H_{opt} (2^{nd} Order Larger Magnitude Case)

As before, the delay – free part of $H_{opt}(s)$ is computed as:

$$H_{opt-delayfree}(s)$$
$$= \frac{-1.0247e*10^{-7}\ (s+1.245*10^{7})(s+4.606)(s+1.426)(s+0.7126)(s-0.9501)}{(s-0.944)(s+0.944)(s+0.7878)(s^2\ +\ 8.078)}$$
$$H_{opt-delayfree}(s)$$
$$\cong \frac{-\ 1.2755\ (s+4.606)\ (s+1.426)\ (s+0.7126)\ (s-0.9501)}{(s-0.944)(s+0.944)(s+0.7878)\ (s^2\ +\ 8.078)}$$

The expression of $H_{opt-delayfree}(s)$ has poles exactly at β_k values, again. There is also an additional negative pole which comes from $Q_2(s)$. In order to obtain $H_{opt-Filter}(s)$, we should find $Q_2(s)$ from the residues of $H_{opt-delayfree}(s)$ and substract it from $H_{opt-delayfree}(s)$ equation.

Hence,

$$Q_2(s) = \frac{0.17232}{\mathrm{s} + 0.7878}$$

From $Q_F = \frac{1}{1+Q_2}$ we have:

$$Q_F = \frac{\mathrm{s} + 0.7878}{s + 0.9601}$$

As expected, second order W led to first order Q_F. Then, by executing $H_{opt-Filter}(s) = H_{opt-delayfree}(s) - Q_2(s)$, we get following:

$$H_{opt-Filter}(s) = \frac{-1.4479\ (\mathrm{s} + 3.376)\ (\mathrm{s} + 1.891)\ (\mathrm{s} - 0.9495)}{(\mathrm{s} + 0.944)\ (\mathrm{s} - 0.944)\ (s^2\ +\ 8.078)}$$

After that, we convert $H_{opt-Filter}(s)$ to state – space form and get A, B and C state – space matrices to calculate $H_{opt-delay}(s)$ and $H_{opt-FIR}(s)$. With the same implementation on first example, $H_{opt-delay}(s)$ is computed as following:

$$H_{opt-delay}(s) \cong \frac{-2.6832\ (s + 0.05)\ (s + 0.1)\ (s - 1)\ e^{-0.5s}}{(\mathrm{s} + 0.944)\ (\mathrm{s} - 0.944)\ (s^2\ +\ 8.078)}$$

Then,

$$H_{opt-FIR}(s) = H_{opt-Filter}(s) - H_{opt-delay}(s)$$
$$= \frac{-1.4479\ (\mathrm{s} + 3.376)\ (\mathrm{s} + 1.891)\ (\mathrm{s} - 0.9495) + 2.6832\ (s + 0.05)\ (s + 0.1)\ (s - 1)\ e^{-0.5s}}{(\mathrm{s} + 0.944)\ (\mathrm{s} - 0.944)\ (s^2\ +\ 8.078)}$$

Therefore,

$$H_{opt}(s) = Q_2(s) + \ H_{opt-FIR}(s)$$
$$H_{opt}(s) = \frac{0.17232}{(\mathrm{s} + 0.7878)}$$
$$+ \left(\frac{-1.4479\ (\mathrm{s} + 3.376)\ (\mathrm{s} + 1.891)\ (\mathrm{s} - 0.9495) + 2.6832\ (s + 0.05)\ (s + 0.1)\ (s - 1)\ e^{-0.5s}}{(\mathrm{s} + 0.944)\ (\mathrm{s} - 0.944)\ (s^2\ +\ 8.078)}\right)$$

As in the previous examples, the denominator expression of $H_{opt}(s)$ has roots on the imaginary axis and the right-half plane. These get cancelled by the zeros of the numerator, which are the β_k values, exactly at the same

points. On the other hand, $Q_2(s)$ is both proper and stable transfer function again. Thereby, $H_{opt}(s)$ is stable which ensures the stability of $Q_{opt}(s)$. The stability of $Q_{opt}(s)$ can be observed from the Nyquist plot of $H_{opt}(s)$:

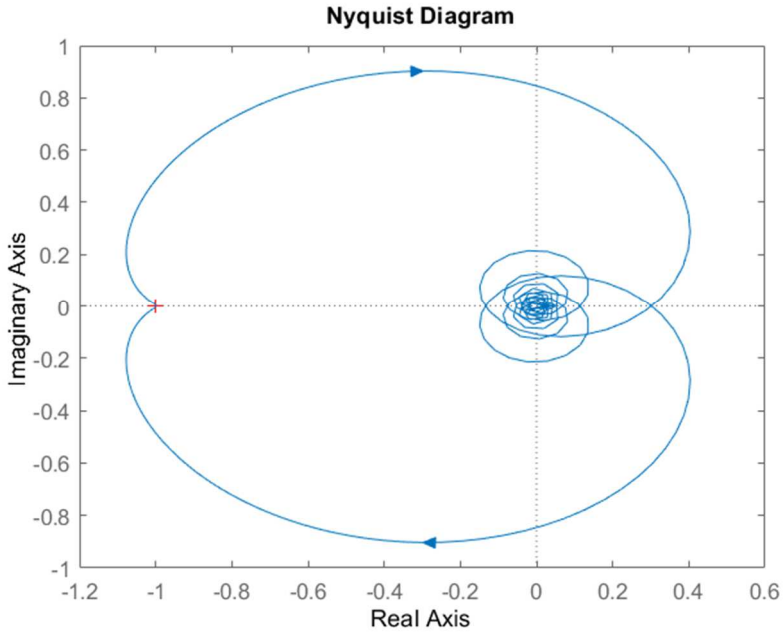

Figure 4.29 Nyquist plot of $H_{opt}(s)$ (2^{nd} Order Larger Magnitude Case)

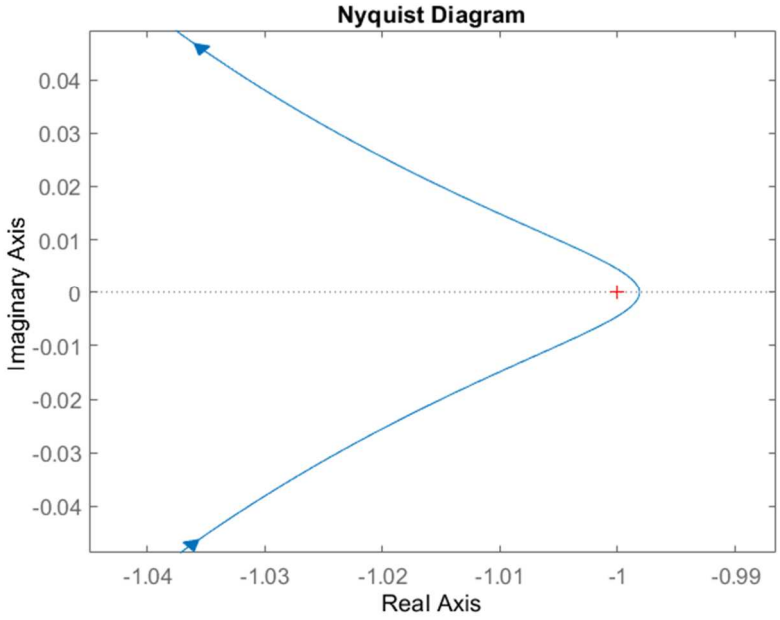

Figure 4.30 Zoomed Nyquist plot of $H_{opt}(s)$ (2^{nd} Order Larger Magnitude Case)

The impulse response of $h(t)$ is observed as FIR again. It can be seen at the figure below:

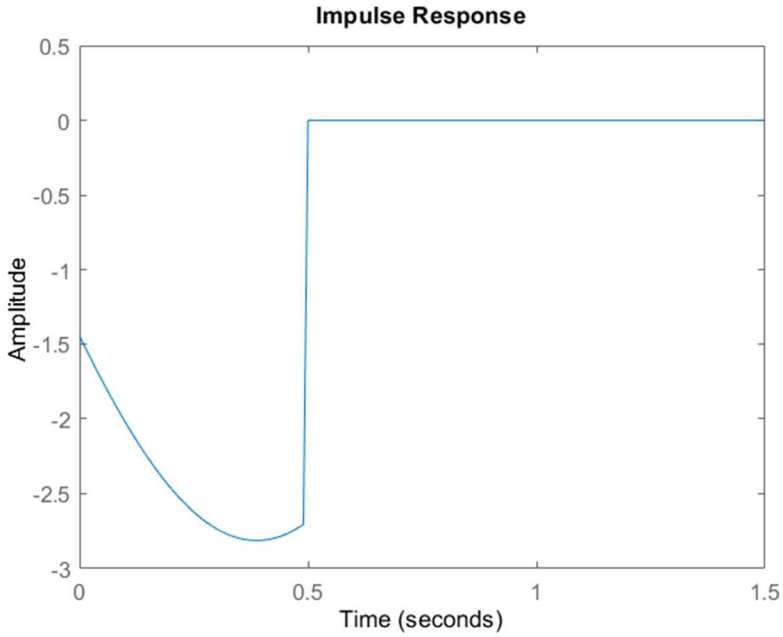

Figure 4.31 Impulse Response for $h(t)$ (2^{nd} Order Larger Magnitude Case)

Once again, this result completes the general structure searching for $Q_{opt}(s)$ since we know the form of $Q_{opt}(s)$ as $Q_{opt} = \frac{\gamma_{opt} Q_F}{1+Q_F H_{opt-FIR}}$. Hence, the structural result of Q_{opt} for this example is computed as at below:

$$
\begin{aligned}
Q_{opt} &= \frac{\gamma_{opt} Q_F}{1 + Q_F H_{opt-FIR}} \\
&= \frac{3.7269 \left(\frac{\mathrm{s}+0.7878}{s+0.9601}\right)}{1 + \left[\begin{array}{c} \left(\frac{s+0.7878}{s+0.9601}\right) \\ * \left[\frac{-1.4479\ (\mathrm{s}+3.376)(\mathrm{s}+1.891)(\mathrm{s}-0.9495)}{(\mathrm{s}+0.944)(\mathrm{s}-0.944)(s^2\ +\ 8.078)}\right. \\ \left. + \frac{2.6832\ (s+0.05)(s+0.1)(s-1)\ e^{-0.5s}}{(\mathrm{s}+0.944)(\mathrm{s}-0.944)(s^2\ +\ 8.078)}\right] \end{array}\right]}
\end{aligned}
$$

■

Chapter 5

Conclusions

All in all, different types of robust control problems and the general structure of one – block H_∞ control problems are examined and represented in the first chapter. Different methods and approaches, such as the Sarason's Theorem, the Nevanlinna – Pick Interpolation and the Nehari's Theorem, for one – block control problems are mentioned briefly in this chapter.

Among these methods and approaches, the Sarason's Theorem, which we interested in this thesis, is analyzed and studied at the second and third chapter in detail so that it is used for the solution of the infinite dimensional one – block H_∞ control problems. Thus, we provided a general structure as $Q_{opt} = \frac{\gamma_{opt}}{1+(Q_2+H_{opt-FIR})} = \frac{\gamma_{opt}Q_F}{1+Q_F H_{opt-FIR}}$ for optimal Q_{opt} in the infinite dimensional one – block H_∞ control problem.

Lastly, different type one – block control problem examples are given in the fourth chapter to give an illustration about how to obtain and implement $Q_{opt} = \frac{\gamma_{opt}Q_F}{1+Q_F H_{opt-FIR}}$ structure.

Therefore, the chapters in this thesis concludes our main interest and effort to find the structure of Q_{opt} for infinite dimensional systems, as a compact representation.

When we identify the Q_{opt}, it is obviously infinite dimensional since $H_{opt-FIR}$ equation in the H_{opt} is infinite dimensional because of the delay term. Nevertheless, $H_{opt-FIR}$ can be approximated for finite dimensional implementation very easily since it is a stable FIR filter. There are many different approximation techniques for FIR filter, including Matlab's "`tfest(data,np,nz,iodelay)`"[31] and "`fitfrd(A,N)`" [31], and also see [6]. In this respect, real life implementation of Q_{opt} is easy and practical for infinite dimensional systems.

If we mention about what we did not consider in this thesis, we should indicate the mixed sensitivity problem for unstable plants. In this particular problem, the structure of $F(s)$ is $(W - MQ)$, where $W(s)$ has a peculiar structure and $M(s)$ consists two parts as $M = M_n M_d$. As future work, this situation should be also discussed and considered.

On the other hand, Matlab provide us state – space representation for all infinite dimensional systems which consists delay terms. This brings us difficulty in order to understand and exhibit structural results. Also, taking inverse of a transfer function which consists delay terms is restrained as error by Matlab in respect of the causality of the system. However, computational processes such as taking inverse of an infinite dimensional transfer function are needed especially for structural results as indicated before

As it is explained in the previous chapters, γ_{opt} is computed and found iteratively by searching it with singular value decomposition in every step. When we apply all values in $[\gamma_{min}, \gamma_{max}]$ by small increase rate, the resolution is increased whereas computational process takes too much time. Since we did not focus on finding γ_{opt} in a very efficient computation process, it

takes lots of time to find it with Matlab. As future work, computation of γ_{opt} can be enhanced with different approach such as bi – section search algorithm. In this way, the tool that we provided in this thesis become very efficient and fast to use and implement.

The examples illustrate that the Nyquist graphs of $H_{opt}(s)$ are very close to -1 at $s = 0$. This situation implies Small Gain Margin for $Q_{opt}(s)$; i.e small increase in the gain of $H_{opt}(s)$ may lead to unstable $Q_{opt}(s)$. This means that $H_{opt}(s)$ should be computed as precisely as possible, at least in the low frequency region.

Bibliography

[1] H. Özbay, "Introduction to Feedback Control Theory," *CRC Press*, Boca Raton, 1999.

[2] H. Özbay, S. Gümüşsoy, K. Kashima, and Y. Yamamoto, "Frequency domain techniques for H_∞ control of distributed parameter systems," *Society for Industrial and Applied Mathematics*, Philadelphia, 2018

[3] J. C. Doyle, B. A. Francis, and A. Tannenbaum, "Feedback Control Theory," *Macmillan*, New York, 1992. (Cited on pp. 71-73, 149).

[4] C. Foias, H. Özbay, and A. Tannenbaum, "Robust Control of Infinite Dimensional Systems: Frequency Domain Methods," *Springer*, London, 1996.

[5] D. Sarason, "Generalized Interpolation in H_∞," *Transactions of the American Mathematical Society*, vol. 127, no 2, pp. 179-203, 1967.

[6] M. Nagahara and Y. Yamamoto, "*FIR Digital Filter Design by Sampled-Data H^∞ Discretization*," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3110-3115, 2014.

[7] A.D. Kara and S. Yüksel, "Robustness to Incorrect System Models in Stochastic Control and Application to Data-Driven Learning," *2018 IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL, pp. 2753 – 2758, 2018.

[8] T. Başar and P. Bernhard, "H_∞ Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach," *Birkhäuser*, Boston, 1995.

[9] H. Kwakernaak, "Robust Control and H_∞ - Optimization - Tutorial Paper," *Automatica*, vol. 29, no. 2, pp. 255-273, 1993.

[10] H. Özbay, "A Simpler Formula for the Singular Values of a Certain Hankel Operator," *Systems & Control Letters*, vol. 15, no .5, pp. 381-390, 1990.

[11] I. Tolaimate and N. Elalami, "Robust Control Problem as H_2 and H_∞ Control Problem Applied to the Robust Controller Design of Active Queue Management Routers for Internet Protocol," *International Journal of Systems Applications*, vol. 5, no. 6, pp. 683-691, 2011.

[12] G. Zames, "Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms and Approximate Inverses," *IEEE Transactions on Automatic Control*, vol. 26, pp. 301-320, 1981.

[13] C. Foias, A. Tannenbaum, and G. Zames, "Weighted sensitivity minimization for delay systems," *IEEE Transactions on Automatic Control*, vol. 31, pp. 763–766, 1986.

[14] J. R. Partington and K. Glover, "Robust stabilization of delay systems by approximation of coprime factors," *Systems & Control Letters*, vol. 14, pp. 325–331, 1990.

[15] H. Özbay, "Robust control of infinite dimensional systems," *in Encyclopedia of Systems and Control, Springer-Verlag*, London, 2014.

[16] G. E. Dullerud and F. Paganini, "A Course in Robust Control Theory: A Convex Approach," *Springer-Verlag*, New York, 2000.

[17] P. P. Khargonekar and K. Poolla, "Robust stabilization of distributed systems," *Automatica*, vol. 22, pp. 77–84, 1986.

[18] V. L. Kharitonov, "Robust stability analysis of time delay systems: A survey," *Annual Reviews in Control*, vol. 23, pp. 185–196, 2011.

[19] Van der Sande, T. P. J., et al. "Robust control of an electromagnetic active suspension system: Simulations and measurements." *Mechatronics,* vol. 23, no. 2, pp. 204-212, 2013.

[20] P.F. Quet And H. Özbay, "On the design of AQM supporting TCP flows using robust control theory," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1031–1036, 2004.

[21] P. Yan and H. Özbay, "On the robust controller design for Hard Disk Drive servo systems with time delays." *In: 2013 European Control Conference (ECC)*, IEEE, pp. 1286-1291, 2013.

[22] V. Milić, Ž. Šitum, and M. Essert. "Robust H_∞ position control synthesis of an electro-hydraulic servo system." *ISA transactions,* vol. 49, no. 4, pp. 535-542, 2010.

[23] G. Gu and P. P. Khargonekar, "Linear and nonlinear algorithms for identification in H∞ with error bounds," *IEEE Transactions on Automatic Control*, vol. 37, pp. 953–963, 1992.

[24] O. Toker and H. Özbay, "H_∞ Optimal and suboptimal controllers for infinite dimensional SISO plants," *IEEE Transactions on Automatic Control*, vol. 40, pp. 751–755, 1995.

[25] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their L_∞-error bounds," *International Journal of Control*, vol. 39, pp. 1115–1193, 1984.

[26] Z. Nehari, "On bounded bilinear forms," *Annals of Mathematics*, vol. 65, pp. 153–162, 1957.

[27] The Mathworks, Inc., MATLAB version 9.9.0.1467703 (R2020b), Natick, Massachusetts, 2020.

[28] The Mathworks, Inc., *Robust Control Toolbox*, https://www.mathworks.com/products/robust.html, November 2020

[29] The Mathworks, Inc., *Control System Toolbox*, https://www.mathworks.com/products/control.html, November 2020

[30] The Mathworks, Inc., "*getDelayModel*", https://www.mathworks.com/help/control/ref/getdelaymodel.html, November 2020

[31] The Mathworks, Inc., "*tfest*", https://www.mathworks.com/help/ident/ref/tfest.html, November 2020

[32] The Mathworks, Inc., "*fitfrd*", https://www.mathworks.com/help/robust/ref/fitfrd.html, November 2020

Appendix A

Matlab Codes

```
% % FIRST COMPILE Sarason.m FILE TO OBTAIN FUNCTION

% % COMMANDS FOR CHOOSING W AND M TO OBTAIN EXAMPLES AT BELOW
% % CAN BE RUN BY SELECTING AND PRESSING F9
% % OR JUST BY COPY-PASTE INTO COMMAND WINDOW

% % PLEASE CAREFULLY READ THE INSTRUCTION AND WARNINGS
% % AT BELOW BEFORE USE THE FUNCTION
clc;
close all;
clear all;

s = zpk('s');

% % General Example
W = 1/(s + 0.1);
% % 2nd Order Inf Dim. Ex (Be careful about gamma and singular_val)
% W = 1/((s + 0.1)*(s + 0.05));
% % 3rd Order Inf Dim. Ex (Be careful about gamma and singular_val)
% W = 1/((s + 0.1)*(s + 0.05)*(s+10));
% % 2nd Order W Always Over Magnitude of Ex3's W (Be careful about gamma and singular_val)
% W = (10*(s + 1))/((s + 0.1)*(s + 0.05));
M = exp((-0.5)*s);


% In the Sarason.m function, gamma_opt calculation needs high precision
% For this purpose, gamma range and singular values' lower limit
% may wanted to be ADJUSTED MANUALLY
% Otherwise, gamma range will be calculated in respect of
% gamma_min = max(abs(W_M_zeros)) and gamma_max = ||W||inf , but it
% may take so much time to calculate singular values since range
% is too large and precision is kept high.
[gamma_opt,QF,Hopt_Filter, Hopt_delay] = Sarason(M,W)
```

Sarason.m file:

```
function [gamma_opt,QF,Hopt_Filter_FINAL, Hopt_delay_FINAL]=Sarason(M,W)

s = zpk('s');

freq = logspace(-3,3,300);
t = 0 : 0.01 : 1.5;

% Bode Diagram of W(s)
figure()
Bode(W,{0.000001,1000000})

%Boundaries of gamma
%gamma_min calculation
M_zeros = zero(M); %zeros of M(s) except s=inf
if evalfr(M,inf) == 0 %M(s) at s=inf is zero
    if length(M_zeros) ~= 0
        for l = 1:length(M_zeros)
            W_M_zeros(l) = evalfr(W,M_zeros(l));
        end
            W_M_zeros(l+1) = evalfr(W,inf);
    else
        W_M_zeros = evalfr(W,inf);
    end
else
    for l = 1:length(M_zeros)
        W_M_zeros(l) = evalfr(W,M_zeros(l));
    end
end
gamma_min = max(abs(W_M_zeros));

% gamma_max calculation
% gamma_max = ||W||inf
gamma_max = getPeakGain(W);


W_C = W';

[W_Num, W_Den]  = tfdata(W,'v');

% Partial fraction expansion (partial fraction decomposition)
% the residues r = [rn ... r2 r1],
% the poles p = [pn ... p2 p1],
% (Here p of [r,p,k] represents poles of W )
% (which are -Pi's) and the polynomial k
[r,p,k] = residue(W_Num,W_Den);

%Constrating W matrix which has elements of partial fraction
for j=1:length(p) %or length(r)
   W_1xn(1,j) = r(j) / zpk(p(j),[],1) ;

   M_1xn_Pi(j,:) = evalfr(M,-p(j));
end

%Defining W* matrix
```

```
W_1xn_C = (W_1xn').';

for jj = 1:length(p)
    W_Pi(jj,:) = evalfr(W_1xn,-p(jj));
end


%Defining parts of G_Num
M_pi_W_C_W_pi = zpk(zeros(1,length(p)));
for o = 1 : length(p)
    for oo = 1 : length(p)
        M_pi_W_C_W_pi(o) = M_pi_W_C_W_pi(o) + W_1xn_C(oo)*M_1xn_Pi(oo)*W_Pi(oo,o);
    end
end

W_C_M_W_1xn = W_C * M * W_1xn ;


G_Num = [(M_pi_W_C_W_pi - W_C_M_W_1xn) (-W_1xn_C)];


% G(s) = ( -W*Mpsi_1 + psi_2 + psi_3 ) / gamma^2 - W*W
% G(s) is in H(M) and is not zero.
% So, when G_Den = 0 then G_Num have to be 0
% Then, roots of (gamma-W*W = 0) are beta_k values
% Lets try gamma values to obtain beta_k values to continue calculation.

% Remark: largest gamma value makes R_gamma singular is optimum.

gamma = (gamma_min+0.001): 0.001 :(gamma_max-0.001);
% FOLLOWING GAMMA RANGES CAN BE CHOOSEN MANUALLY
% IN RESPECT OF THE EXAMPLES IN THE THESIS

% For 2nd Order Inf Dim. Ex
% gamma = linspace(0.001,0.1,6000); %just for proper sing. val. graph
% gamma = linspace(0.0694913763968,0.0694913763972,30000); % for calc.

% For 3rd Order Inf Dim. Ex
% gamma = linspace(0.00001,0.007,10000); %just for proper sing. val. graph
% gamma = linspace(0.00467539343,0.00467539345,10000); % for calc.
% gamma = linspace(0.004675393438208,0.004675393438210,10000); % for calc.

% For 2nd Order W Always Over Magnitude of Ex1's W
% gamma = linspace(0.001,5,5000); %just for proper sing. val. graph
% gamma = linspace(3.726930244144,3.726930244145,10000); % for calc.


small_singular_vals=[];
small_gammas=[];
for i = 1 : length(gamma)
%  CONSTRUCTION OF R_GAMMA
% corresponding beta_k values
    Bk = zero((gamma(i)^2) - (W_C)*W);
    for ii = 1 : length(Bk) %or length(r)
        R_gamma(ii,:) = evalfr(G_Num,Bk(ii));
    end
```

```
    [U,S,V] = svds(R_gamma,1,'smallest');

    singular_val(i) = S;
% FOLLOWING SINGULAR VALUE LOWER LIMITS CAN BE CHOOSEN MANUALLY

%     if singular_val(i) < 0.00000000000000015  %for 2nd Order Inf Dim. Ex

%     if singular_val(i) < 0.0000000000000015  %for 3rd Order OTHER(s_plus_10) Inf Dim. Ex
%     if singular_val(i) < 0.00000000000000000015 %for 3rd Order OTHER(s_plus_10) Inf Dim. Ex

%     if singular_val(i) < 0.0000000000000012 %for 2nd Order W Always Over Magnitude of Ex3's W


    if singular_val(i) < 0.0001  % For General Ex. and sing. val. graphs

        small_singular_vals = [small_singular_vals singular_val(i)];
        small_gammas = [small_gammas gamma(i)];

        R_gamma_opt = R_gamma;
        gamma_opt = gamma(i);

        gamma_index = i;
        corr_Bk = Bk;
% Here => DELTA VECTOR := corr_sing_vec
        corr_sing_vec = V;
        corr_sing_vec_other = U;
    end
end


figure()
plot(gamma,singular_val)
xlabel('gamma')
ylabel('Min Singular Values')

% % Just for high precision gamma_opt calculations
% figure()
% plot(small_gammas,small_singular_vals)
% xlabel('small gammas')
% ylabel('small Min Singular Values')

% Reset too small unexpected im part of delta vector
for u = 1 : length(corr_sing_vec)
    if abs(imag(corr_sing_vec(u))) < 0.00001
        corr_sing_vec(u) = real(corr_sing_vec(u));
    end
end

% phi vector
for iii = 1 : length(p)
    phi(iii) = corr_sing_vec(iii);
end
% theta vector
for iii = 1 : length(p)
    theta(iii) = corr_sing_vec(length(p)+ iii);
end
```

```
% Finding G(s) in ss format
G = (G_Num * corr_sing_vec) / ((gamma_opt^2) - (W_C)*W);

figure()
Bode(G,freq)

% Finding Qopt(s) in ss format
Qopt = (W_1xn * transpose(phi)) / G;

figure()
Bode(Qopt,freq)



% Finding Hopt(s) in ss format
Hopt = (inv(Qopt) * gamma_opt) - 1;

figure()
Bode(Hopt,freq)
% figure()
% nyquist(H_opt_PAPER,freq)


% % %  AT BELOW, Hopt(where h(t) is FIR ) IS OBTAINED IN SS FORMAT
% % %  BECAUSE OF DELAY TERM(ACTUALLY IT IS BECAUSE OF MATLAB).
% % %  SO, A METHOD IS DEVELOPED TO SEE IT(its parts) AS TF.
% % %  FIRST, NONDELAY PART IS OBTAINED WITH getDelayModel().
% % %  THEN, ITS TF PARTS ARE OBSERVABLE AS MULTIINPUTS(see getDelayModel()
% % %  function structure)
% % %  AS A RESULT OF PROCEDURE, Hopt IS RE-OBTAINED AND CHECK WITH
% % %  ORIGINAL RESULT. THEY ARE ENDED UP AS SAME RESULT.
[H_Hopt,tau_Hopt] = getDelayModel(Hopt);

% In some instances getDelayModel() gives annoying models which cannot
% be converted to zpk. Following part solves this issue
if length(H_Hopt.OutputName) == 1
    if isempty(H_Hopt.E)
        H_Hopt = ss(H_Hopt.A, H_Hopt.B,...
        H_Hopt.C, H_Hopt.D);
    else
        H_Hopt = dss(H_Hopt.A, H_Hopt.B,...
        H_Hopt.C, H_Hopt.D,H_Hopt.E);
    end
end

% For unexpected too small parts of getDelayModel() results
H_Hopt.A(abs(H_Hopt.A)<0.00001)=0;
H_Hopt.B(abs(H_Hopt.B)<0.00001)=0;
H_Hopt.C(abs(H_Hopt.C)<0.00001)=0;
H_Hopt.D(abs(H_Hopt.D)<0.00001)=0;


H_Hopt = zpk(H_Hopt);
H_Hopt = minreal(H_Hopt,0.0001);

% % THIS PART IS FOR MAKING VERY SMALL MAG. PART OF zpk(H_H_opt_PAPER)
% for iiii = 1 : size(H_H_opt_PAPER,1) % row number of H
```

```
%     for jjjj = 1 : size(H_H_opt_PAPER,2) % column number of H
%         Mag = getPeakGain(H_H_opt_PAPER(iiii,jjjj));
%         if Mag < 0.05
%             H_H_opt_PAPER(iiii,jjjj) = 0
%         end
%     end
% end

% Applying the model of getDelayModel()
Hopt_u_to_y = H_Hopt(1,1);
Hopt_u_to_z = H_Hopt(2,1);
Hopt_w_to_y = H_Hopt(1,2);
Hopt_w_to_z = H_Hopt(2,2);
Hopt_z_to_w = exp(-tau_Hopt*s);

% % Delay part of Hopt
% Hopt_delay = Hopt_u_to_z * Hopt_z_to_w * Hopt_w_to_y;
% Hopt_delay = minreal(Hopt_delay,0.001);

% Delay-free part of Hopt
Hopt_delayfree = Hopt_u_to_y;
% H_opt_PAPER_NonDelay = minreal(H_opt_PAPER_NonDelay,0.001);

% H_opt_PAPER_Final = H_opt_PAPER_NonDelay + H_opt_PAPER_Delay;
% impulse(H_opt_PAPER_Final,t)

[Hopt_delayfree_Num, Hopt_delayfree_Den] = tfdata(Hopt_delayfree,'v');
[r_H,p_H,k_H] = residue(Hopt_delayfree_Num,Hopt_delayfree_Den);

% In case of incompatibility, tol can be adjusted
Marix_Comparison = ismembertol(abs(p_H),abs(corr_Bk),0.00001);
ProperStable_Index = find(Marix_Comparison==false);


for jjjjj=1:length(p_H) %or length(r)
   Hopt_delayfree_1xn(1,jjjjj) = r_H(jjjjj) / zpk(p_H(jjjjj),[],1);
end

Hopt_ProperStables = Hopt_delayfree_1xn(1,ProperStable_Index);
Q1 = gamma_opt;
Q2 = 0;
if length(Hopt_ProperStables) > 0
    for ooo = 1 : length(Hopt_ProperStables)
        Q2 = Q2 + Hopt_ProperStables(ooo);
    end
end

QF = 1 / (1 + Q2);

% Hopt_Filter
Hopt_Filter = Hopt_delayfree - Q2;
Hopt_Filter = minreal(Hopt_Filter,0.001);



Hopt_Filter_SS = ss(Hopt_Filter);
```

```
A = Hopt_Filter_SS.A;
% % Following minimization may be needed sometimes
% A(abs(A)<0.00001)=0
% for n = 1 : size(A,1)
%     for m = 1 : size(A,2)
%         if abs(imag(A(n,m))) < 0.000001
%             A(n,m) = real(A(n,m))
%         end
%         if abs(real(A(n,m))) < 0.000001
%             A(n,m) = imag(A(n,m))
%         end
%     end
% end

B = Hopt_Filter_SS.B;
% % Following minimization may be needed sometimes
% B(abs(B)<0.00001)=0
% for n = 1 : size(B,1)
%     for m = 1 : size(B,2)
%         if abs(imag(B(n,m))) < 0.000001
%             B(n,m) = real(B(n,m))
%         end
%         if abs(real(B(n,m))) < 0.000001
%             B(n,m) = imag(B(n,m))
%         end
%     end
% end

C = Hopt_Filter_SS.C;
% % Following minimization may be needed sometimes
% C(abs(C)<0.00001)=0
% for n = 1 : size(C,1)
%     for m = 1 : size(C,2)
%         if abs(imag(C(n,m))) < 0.000001
%             C(n,m) = real(C(n,m))
%         end
%         if abs(real(C(n,m))) < 0.000001
%             C(n,m) = imag(C(n,m))
%         end
%     end
% end

D = Hopt_Filter_SS.D;
% % Following minimization may be needed sometimes
% D(abs(D)<0.00001)=0
% for n = 1 : size(D,1)
%     for m = 1 : size(D,2)
%         if abs(imag(D(n,m))) < 0.000001
%             D(n,m) = real(D(n,m))
%         end
%         if abs(real(D(n,m))) < 0.000001
%             D(n,m) = imag(D(n,m))
%         end
%     end
% end

I_Hopt_Filter = eye(size(A));
```

```
% Hopt_Filter itself is obtained correctly
Hopt_Filter_FINAL = zpk(C*(inv(s*I_Hopt_Filter - A))*B);
Hopt_Filter_FINAL = minreal(Hopt_Filter_FINAL,0.001);


C_eAh = C*expm(A*tau_Hopt);

% Delay part of Hopt
Hopt_delay_FINAL = exp(-tau_Hopt*s)*minreal(zpk(C_eAh*inv(s*I_Hopt_Filter- A)*B),0.001);
% Hopt_Filter_2 = exp(-tau_H_opt_PAPER*s)*minreal(zpk(C_eAh*(inv(s*I_Hopt_Filter-
A)*B+D)),0.001)
% minreal(H_opt_PAPER_Delay/Hopt_Filter_2,0.001)

% Hopt_FIR = Hopt_Filter - Hopt_delay
Hopt_FIR = (Hopt_Filter_FINAL - Hopt_delay_FINAL);

% HOPT IS FOUND
Hopt_FINAL = Hopt_FIR + Q2;

% % Checksum
% figure()
% Bode(Hopt-Hopt_FINAL)

figure()
impulse(Hopt_FIR,t)

figure()
nyquist(Hopt_FINAL,freq)


% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %


% % Checksum
% Qopt_FINAL = (Q1*QF)/(1+(QF*Hopt_FIR));
% figure()
% Bode(Qopt-Qopt_FINAL)


% % Checksum
% CHECK THAT MAGNITUDE OF F_OPT IS CONSTANT AND EQUAL TO GAMMA_OPT
F_opt = ((W*G) - (M*(W_1xn * transpose(phi)))) / G;
figure()
Bode(F_opt)

end
```