Creating Two Stability Intervals along the Delay Axis via Multi-Parameter Controller Design: Linear SISO Case

A Thesis Presented
by
Dongzuo Tian
to
The Department of Mechanical and Industrial Engineering
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Mechanical Engineering

Northeastern University
Boston, Massachusetts
June 2016
To my family and friends, my greatest support.
Contents

List of Figures iv
List of Tables v
List of Acronyms vi
Acknowledgments vii
Abstract of the Thesis viii

1 Introduction 1
  1.1 Background ......................................................... 1
  1.2 Motivation ......................................................... 2

2 Preliminaries 3
  2.1 Problem Statement .................................................. 3
  2.2 Imaginary Crossings ................................................. 5
  2.3 Critical Delay Values ............................................... 6
  2.4 Root Tendency ....................................................... 6
  2.5 Conclusion ......................................................... 8

3 Delay Margin Design 9
  3.1 Control Design Approach ......................................... 9
  3.2 Example Case Study ............................................... 11
  3.3 Conclusion ......................................................... 15

4 Design of Two Stable Intervals: The Case with a Stable Plant 16
  4.1 Control Design Approach ......................................... 16
  4.2 Example Cases Study ............................................... 23
    4.2.1 Example 1: A Computational Study .......................... 23
    4.2.2 Example 2: An Experimental Study .......................... 26
    4.2.3 Example 3: Higher Order System Example .................. 28
  4.3 Conclusion ......................................................... 31
5 Design of Two Stable Intervals: The Case with an Unstable Plant

5.1 Motivation .......................................................... 33
5.2 Control Design Approach .............................................. 34
5.3 Examples ................................................................. 36
  5.3.1 Example A: 1st order open-loop plant ....................... 36
  5.3.2 Example B: 2nd order open-loop plant ....................... 36
5.4 Conclusion .............................................................. 37

6 Conclusion ............................................................... 40

Bibliography ............................................................... 42

A Initial Condition ........................................................ 46
List of Figures

2.1 Model of the SISO LTI system with single delay $\tau$ in the feedback .................................................. 4

3.1 Desired stability decomposition of the closed loop system, given $\bar{\tau}$ .................................................... 10

3.2 Example: Unit step response of the closed loop system simulations at $\tau = 0, 0.95, 1, 1.05$, with the controller $C(s)$ designed using the optimization method described in Section 3.1 and the delay margin $\bar{\tau} = 1$ .................................................. 14

3.3 PID controller gains under different $\zeta$ values with $\omega_n = 5$ .................................................. 14

4.1 Desired two disjoint stability intervals decomposition of the closed loop system .................................................. 17

4.2 Example 1: Unit step response of the closed loop system simulations at $\tau = 0, 0.095, 0.1, 0.105, 0.495, 0.5, 0.505, 0.695, 0.7, 0.705$, with the controller $C(s)$ designed using the optimization method described in Section 4.1 and the pre-determined stability decomposition $\tau \in [0, 0.1) \cup (0.5, 0.7]$ .................................................. 25

4.3 Experimental setup of Rotary Servo Base Unit with amplifier and data acquisition board .................................................. 26

4.4 Model of the speed control setting with a filter .................................................. 27

4.5 Example 2: Step response of the motor-shaft angular velocity with controller $C_b(s)$ in both experiments and simulations at delay values $\tau = 0, 0.25, 0.3, 0.32, 0.99, 1.0, 1.05, 1.4, 1.8, 1.85$ (Experimental data: solid line; Simulation data: dashed line). Experimental results and model predictions are superimposed. Control design procedure utilizes the pre-determined stability decomposition of $\tau \in [0, 0.3) \cup [1.0, 1.8]$ .................................................. 29

4.6 Model of Example 3 .................................................. 30

5.1 Example A: Unit step response of the closed loop system simulations at $\tau = 0, 0.29, 0.3, 0.31, 2.01, 2.02, 2.03, 2.05, 2.06, 2.07$ with the controller $C(s)$ designed for $G_1(s)$ and the pre-determined stability decomposition $\tau \in [0, 0.3) \cup (2.02, 2.06)$ .................................................. 38

5.2 Example B: Unit step response of the closed loop system simulations at $\tau = 0.39, 0.40, 0.41, 0.69, 0.7, 0.71, 0.89, 0.9, 0.91$ with the controller $C(s)$ designed for $G_2(s)$ and the pre-determined stability decomposition $\tau \in [0, 0.4) \cup (0.7, 0.9)$ .................................................. 39

A.1 Venn diagram of $Q_0$, $Q_1$ and $Q_2$ in the Appendix .................................................. 47
List of Tables

5.1 Obtained PID gains for different $\zeta$ values, when $\omega_n = \sigma$ and $\tau = 1$ . . . . . . . . . . 15
List of Acronyms

LHP/RHP  Left Half Plane/Right Half Plane.
LMI    Linear Matrix Inequality.
LTI    Linear Time-Invariant.
PID    Proportional-Integral-Derivative.
RT     Root Tendency.
TDS    Time Delayed System.
Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Prof. Rifat Sipahi for the continuous support of my master’s study and research, for his patience, motivation, enthusiasm, and immense knowledge. He helps me in all the time of research and writing of this thesis. Only with his advices and guidance that I can go far in my field of study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Bahram Shafai, insightful comments, and hard questions.

My sincere thanks also goes to my roommates, Chengxiang Ma and Wei Wang. I will always remember the time we spent together no matter happiness or sorrowfulness. Moreover, I shall thank for Jing, without your help I cannot finish my presentation in such good manners.

Last but not the least, I would like to thank my family: my parents Xia Wang and Qiang Tian, for giving birth to me at the first place and supporting me spiritually throughout my life.
Abstract of the Thesis

Creating Two Stability Intervals along the Delay Axis via Multi-Parameter Controller Design: Linear SISO Case

by

Dongzuo Tian
Master of Science in Mechanical Engineering
Northeastern University, June 2016
Prof. Rifat Sipahi, Advisor

In many applications of control systems, such as remote control, network control systems, population dynamics, vibration control, and traffic flow, time delays arise naturally. As a general point of view, the presence of delays may cause poor performance, import instability, and loss of functionality in all these applications. Careful control design can help address this issue however the presence of delay makes this design effort very difficult. Intriguingly, it is known that sometimes larger delays are not detrimental at all. It was shown in the literature that in some cases system performance can be higher with larger delays than with some smaller delays. On the other hand current state of the art has not addressed how to design controllers for systems to exhibit such stability properties.

Even for the class of linear time-invariant single-input single-output systems (LTI SISO) affected by delays, control design is challenging. Published work so far has been limited to the design of a limited number of parameters and could render the system stable only for delays from zero up to a certain upper bound. A tool to design controllers achieving stability in multiple disjoint intervals of delays currently does not exist. In this thesis, we address this problem by formulating the control design problem to not only create stability in multiple pre-set delay intervals but also achieve this in the presence of multiple design parameters. The feasibility of the approach is presented for the single stable delay interval for comparison with the existing work, as well as for creating two disjoint stability intervals along the delay axis.

The novel approach in this thesis is based on taking advantage of several inherent stability characteristics of LTI SISO systems affected by delays, such as critical eigenvalues and their sensitivity, and connecting these carefully with readily available nonlinear optimization schemes to solve the multi-parameter design problem effectively and efficiently. Considering different scenarios, we
present the results for controlling both stable and unstable plant transfer functions, and validate the
approach and its efficacy by comparing simulation and experimental results obtained from a speed
control of a DC motor.
Chapter 1

Introduction

1.1 Background

Time delays are commonly seen in many real world problems, involving sensing, actuation and decision making. These delays arise from a various number of sources, such as signal transmission times, computation delay, and physical transport delay. For many control processes, the study of time delay systems is a vital topic, involving network control systems, vibration control, and remote control. The presence of delays in a control system may cause poor performance, degraded robustness or pose major limitation on the ability to control. On this account, there is a large body of literature concerning the effects of time delays in control systems [1, 2, 3, 4].

For the study of LTI systems, the effect of delay can be understood by studying the system eigenvalues. Since delays can significantly influence the eigenvalues, and if the system is not designed properly, then these eigenvalues with the presence of delay will cause poor performance or instability. To remedy this, one must design the controller with caution and consider delay explicitly in the design process [5, 6, 7].

Control design for a LTI system with time delay is quite challenging, which is due to the infinite-dimensional nature of the arising design problem [8]. Based on Lyapunov-based LMI optimization tools, some solutions to this problem are proposed to design multiple controller gains [4]. In general however, LMI approaches may bring with conservatism on stability conditions and are challenged in designing structured controllers. On the other hand, in frequency domain the control design is still of much limitation, mainly due to the difficulty of designing multiple controller parameters with respect to system eigenvalues. Therefore, many studies in this aspect only focus on designing few parameters for single-input single-output (SISO) systems [5, 6, 9, 10, 11, 12, 13].
CHAPTER 1. INTRODUCTION

1.2 Motivation

For LTI systems, stability can be analyzed without any conservatism, hence frequency domain techniques are still attractive and practical. These techniques can also help formulate structured controllers. In this direction, performance of the closed-loop system was achieved using rightmost root calculation to tune the controller gains \([13, 15, 16]\). In these studies, the delay in the loop is given, and a controller optimizing the real part of the rightmost root needs to be computed. Moreover, other contributions in the literature include PID controller designs for SISO systems with a loop delay \([5, 6, 12, 13]\), and tuning of the controller parameters so that stability can be achieved within a delay margin \(\tau\), that is, the system is stable for all delays \(\tau\) satisfying \(\tau \in [0, \tau]\) \([17]\). Furthermore, numerous techniques were published for the “analysis” of the delay parameter space. In these work, the objective is to find out, given system characteristic equation, what the stability/instability decomposition of the system is in any parameter space, including delay \(\tau\) \([11]\). This analysis may reveal that the system can be stable in more than one delay interval \([2, 18, 19]\), an intriguing and useful property of LTI systems. Indeed, it was shown that in some situations the system disturbance rejection capabilities can be equally desirable and be even better when the delay falls in the larger stable delay interval \([20]\).

While recovering system stability in multiple stable delay intervals is attractive from control point of view, most of the existing work remains to focus on the forward problem where system characteristic equation is given, and stability intervals along \(\tau\) are to be computed. A control design approach based on these techniques cannot certify stability in pre-set delay intervals, but requires brut force re-checking, plotting, and parameter scanning, which also limits the design to one or two parameters. Nevertheless, in many design problems, the reverse problem is more relevant, where one reveals the controller gains by which certain pre-determined stability intervals along \(\tau\) can be crafted strategically. Such an effort is extremely complicated mathematically, and if properly constructed, would have practicable impacts on the control of LTI SISO systems.

To the best of our knowledge, the work presented in this thesis is the first attempt in the above described design problem with stability “certificates” in pre-set delay intervals. To lay out the main principles of the approach, here we focus on two aspects: creating one interval to attain certain delay margin along \(\tau\), for the purpose of comparison with the published work as discussed in Chapter 3; creating two disjoint stability intervals along \(\tau\), as discussed in Chapter 4 for stable open-loop plants, and in Chapter 5 for unstable open-loop plants. These goals are achieved by designing multiple coefficients of the system characteristic equation using nonlinear optimization tools.
Chapter 2

Preliminaries

Stability is one of the most important properties used to build or analyze a control system. In this chapter, we mainly discuss the frequency domain techniques for single-input single-output (SISO) LTI systems, and summarize the powerful stability analysis tools as well as the knowledge attained on the topic in the past few decades.

2.1 Problem Statement

A SISO system with a single delay $\tau$ in the control loop is studied in this article, as shown in Fig. 2.1. In general, characteristic equation of this closed-loop system can be expressed as,

$$F(s, e^{-s\tau}) = A(s) + B(s)e^{-s\tau} = 0,$$

where $s \in \mathbb{C}$ is the Laplace variable, the delay is non-negative $\tau \geq 0$, and polynomials $A(s)$ and $B(s)$ are nothing but respectively the numerator and denominator of $C(s) \cdot G(s)$. Here, these polynomials are in the general form,

$$A(s) = s^m + a_{m-1}s^{m-1} + \ldots + a_1s + a_0,$$
$$B(s) = b_n s^n + b_{n-1}s^{n-1} + \ldots + b_1 s + b_0.$$

where $a_k, b_k$ are real, and $\deg(A(s)) = m \geq \deg(B(s)) = n$ holds for causality reasons.

When $m > n$, the system is of retard type [2]. This indicates that the highest order derivative in the system dynamics is not affected by any delay terms. The case of $m = n$ corresponds to neutral class systems [21], which present a number of unique properties related to their spectrum.
CHAPTER 2. PRELIMINARIES

To keep the presentation compact, in the sequel, we present all the developments first for retarded class systems. Adapting these results for neutral class systems will then easily follow.

The main problem at hand is that, given \( G(s) \) in Fig. 2.1, we would like to design \( C(s) \) such that the closed-loop system in Fig. 2.1 with the corresponding characteristic equation (2.1) is asymptotically stable for all delay values \( \tau \) in the feedback falling within pre-determined intervals.

In this study, we consider two scenarios of the intervals along the delay axis:

**I. One stability interval:** The system is asymptotically stable in \( \tau \in [0, \bar{\tau}) \), and marginally stable at \( \bar{\tau} \), where \( \bar{\tau} \) is termed the delay margin [22].

**II. Two stability intervals:** The system is asymptotically stable in \( \tau \in [0, h_1) \cup (h_2, h_3) \), and marginally stable at \( h_1, h_2 \) and \( h_3 \).

To design a retarded type system to be in stability/instability transition at certain delay values, we need to study the roots of the characteristic equation (2.1) that are on the imaginary axis, \( s = j\omega \) [2]. We also need to make sure no other crossings occur for delay values interfering the pre-set stability intervals, and also the proper direction of \( s = j\omega \) roots across the imaginary axis with respect to delay \( \tau \) is attained. Moreover, since for retarded systems, stability of (2.1) at \( \tau = 0 \) is preserved as \( \tau \to 0^+ \), it suffices to guarantee the stability of \( A(s) + B(s)e^{-s\tau} \) is also a stable polynomial for \( \tau = 0^+ \) [2]. These knowledge will be the starting point in the following developments.

Note that designing the polynomials \( A(s) \) and \( B(s) \) to achieve a desired stability decomposition as described in I and II is quite complicated, mainly because solving the transcendental equation (2.1) is impossible. On the other hand, many studies can be followed in order to develop a systematic approach to achieve this nontrivial task (see Chapters 3, 4 and 5). Before we present these new results, in the sequel, we first revisit the work of [23], to carry out the mathematics.
2.2 Imaginary Crossings

Since from the stability point of view, the most critical roots of (2.1) are those that are on the imaginary axis, \( s = j\omega \), following [23], we can calculate these roots by writing down a new equation by replacing \( s \) with \(-s\) in (2.1). That is, \( F(s, e^{-s\tau}) = 0 \) and \( F(-s, e^{s\tau}) = 0 \) have the same roots on the imaginary axis. This is as per the symmetricity of system characteristic roots with respect to the real axis of the complex plane. It then follows:

\[
\begin{align*}
A(s) + B(s)e^{-s\tau} &= 0, \\
A(-s) + B(-s)e^{s\tau} &= 0.
\end{align*}
\]

(2.3)

After elimination of \( e^{-s\tau} \) from (2.3), one obtains

\[
A(s)A(-s) - B(s)B(-s) = 0,
\]

(2.4)

which is an even polynomial in \( s \). Then, with \( s = j\omega \) and hence \( \omega^2 = -s^2 \), equation (2.4) can be rearranged into

\[
W(\omega^2) = |A(j\omega)|^2 - |B(j\omega)|^2 = 0,
\]

(2.5)

whose \( \omega^2 > 0 \) roots are related to \( s = j\omega \) roots of (2.1) for some \( \tau \geq 0 \).

Remark 1 For the imaginary axis roots \( s = j\omega^* \) of (2.1), we have that \( W(\omega^{*2}) = 0 \) shares the same roots \( \omega^* \) [24]. In the case of commensurate delays, i.e., with additional exponential functions in (2.1) such as \( e^{-k\tau s} \), \( k \) being an integer larger than one, imaginary roots of the characteristic equation constitute only a subset of the roots of the transformed equation \( W(\omega^{*2}) = 0 \) [25]. One should therefore pay attention in handling such cases when using this elimination technique. Since we do not have commensurate delays in (2.1), this does not pose any issues in what follows.

Once \( W(\omega^2) = 0 \) in (2.5) is solved for \( \omega^2 > 0 \), the set of solutions, namely, the crossing set, is formed as \( \omega = \{\omega_{\tau 1}, \omega_{\tau 2}, ..., \omega_{\tau M}\} \), where \( M \) is the maximum number of crossings. Since the highest power of \( A(s) \) is \( m \), we have that \( M \leq m \). Next, the delay values creating an imaginary crossing at \( s = j\omega_{\tau \nu} \) are calculated. For this, one obtains \( e^{j\omega^*} = -\frac{B(j\omega)}{A(j\omega)} \) from (2.4), which then yields, for each \( \omega = \omega_{\tau \nu} \),

\[
\tau_{k,\nu} = \frac{1}{\omega_{\tau \nu}} \left( \angle B(j\omega_{\tau \nu}) - \angle A(j\omega_{\tau \nu}) + \pi + 2k\pi \right)
\]

(2.6)

where \( k = 0, 1, ..., \infty \), \( \nu = 1, ..., M \). That is, for each crossing \( \omega_{\tau \nu} \), \( \nu = 1, ..., M \), we have infinitely many delays \( \tau_{k,\nu} \) and the union of these delays \( \bigcup_{k,\nu} \tau_{k,\nu} \) gives rise to all delays that create imaginary axis crossings of the closed-loop system represented by (2.1) [18, 19, 24].
CHAPTER 2. PRELIMINARIES

Remark 2 With the assumption that $\omega_{cv} > 0$ without loss of generality, we then have in general,

$$\omega_{c1} \geq \omega_{c2} \geq \ldots \geq \omega_{cM} > 0,$$

(2.7)

where stability of the delay-free system guarantees that $\omega_{cM} \neq 0$ for any finite $\tau$.

2.3 Critical Delay Values

Denote now by $\tau_{\nu}$, the minimum positive delay value $\tau$ in (2.6) with respect to each $\omega_{cv}$.

This delay can be expressed as:

$$\tau_{\nu} = \frac{1}{\omega_{cv}} \begin{cases} 
\arctan 2(y, x), & \text{if } \arctan 2(y, x) \geq 0; \\
\arctan 2(y, x) + 2\pi, & \text{if } \arctan 2(y, x) < 0.
\end{cases}$$

(2.8)

where

$$x = \Re \left( -\frac{B(j\omega_{cv})}{A(j\omega_{cv})} \right), \quad y = \Im \left( -\frac{B(j\omega_{cv})}{A(j\omega_{cv})} \right).$$

Since as per (2.6), for each $\omega_{cv}$, there exist infinitely many periodically spaced $\tau_{k,\nu}$ values with the period $\frac{2\pi}{\omega_{cv}}$, then using (2.8) each $\tau_{k,\nu}$ in (2.6) can be compactly formulated as

$$\tau_{k,\nu} = \tau_{\nu} + \frac{2\pi}{\omega_{cv}} k, \quad k = 0, 1, \ldots, \infty, \quad \nu = 1, \ldots, M.$$  

(2.9)

2.4 Root Tendency

For each imaginary crossing at $\omega_{cv}, \nu = 1, \ldots, M$, there exist infinitely many delay values $\tau_{k,\nu}$ as per (2.9). As a critical delay value $\tau_{k,\nu}$ is slightly increased, the characteristic root on the imaginary axis $s = j\omega_{cv}$ will be perturbed, either into $\mathbb{C}_-$ or $\mathbb{C}_+$. Depending on which direction the root moves, the system will have two more, or less, unstable roots. Root tendency ($RT$) is aimed at calculating this movement via a sensitivity formula at $j\omega_{cv}$ with respect to $\tau_{k,\nu}$ [18, 19, 24, 26]. It is defined as

$$RT_{k,\nu} = \text{sgn} \left[ \Re \left( \frac{ds}{d\tau} \bigg|_{\tau = \tau_{k,\nu}, s = j\omega_{cv}} \right) \right], \quad k = 0, 1, \ldots, \infty, \quad \nu = 1, \ldots, M.$$  

(2.10)

It is known that (2.10) is invariant for all delays corresponding to a specific crossing, see e.g. [18, 19]. That is, (2.10) is invariant with respect to the counter $k$. This implies that given $\nu$, $RT_{k,\nu}$ is fixed for all $k$. For this reason, we can write $RT_{\nu}$ instead of $RT_{k,\nu}$.
CHAPTER 2. PRELIMINARIES

Now, with the aid of (2.1), we have the following
\[
\frac{ds}{d\tau} = -s \left[ \frac{A'(s)}{A(s)} - \frac{B'(s)}{B(s)} + \tau \right]^{-1}.
\] (2.11)

Based on this, the \( RT_\nu \) is expressed as
\[
RT_\nu = -\text{sgn} \left[ \text{Re} \frac{1}{j\omega_{c\nu}} \left( \frac{A'(j\omega_{c\nu})}{A(j\omega_{c\nu})} - \frac{B'(j\omega_{c\nu})}{B(j\omega_{c\nu})} + \tau \right) \right],
\]
\[
= \text{sgn} \left[ \text{Im} \frac{1}{\omega_{c\nu}} \left( \frac{B'(j\omega_{c\nu})}{B(j\omega_{c\nu})} - \frac{A'(j\omega_{c\nu})}{A(j\omega_{c\nu})} \right) \right].
\] (2.12)

We have that when \( \tau \) increases slightly past \( \tau_{k,\nu} \), if \( RT_\nu = +1 \), then the root at \( s = j\omega_{c\nu} \) crosses the imaginary axis always from LHP to RHP (stable to unstable) for all \( \tau_{k,\nu} \); else if \( RT_\nu = -1 \), then the root always crosses from RHP to LHP (unstable to stable) for all the corresponding delays \( \tau_{k,\nu} (k = 0, 1, 2, ...) \), see [18, 19] for root clustering concepts and [23, 24, 26] discussions on root tendency.

**Remark 3** [23] If the characteristic root only touches the imaginary axis then so \( W(\omega^2) \) touches \( \omega^2 \)-axis. If the characteristic roots cross from LHP(RHP) to RHP(LHP), then \( W(\omega^2) \) crosses from below(above) to above(below) the positive \( \omega^2 \)-axis.

Based on Remark 3, the following relationships hold:
\[
RT_\nu = \text{sgn} \left[ \frac{dW(\sigma)}{d\sigma} \right]_{\sigma = \omega_{c\nu}^2} = \text{sgn}[W'(\omega_{c\nu}^2)].
\] (2.13)

**Remark 4** In general, for SISO systems, when crossing values \( \omega_{c\nu} \) are unique, and sorted in order \( \omega_{c1} > \ldots > \omega_{c\nu} > \ldots > \omega_{cM} \), then \( RT_\nu \) corresponding to each one of these crossings have alternating signs. Moreover since (2.1) represents a retarded class system, \( W(\omega^2) \) goes to \(+\infty\) with \( \omega \) increasing. Owing to this, we have \( RT_1 = +1 \) at the largest crossing \( \omega_{c1} \), \( RT_2 = -1 \) at \( \omega_{c2} \) and so on [24]. Special cases must be carefully considered, for example, when \( \omega_{c\nu} \) is a tangent point of the curve \( W(\omega^2) \) [23] or when (2.1) has multiple roots [24, 28]. For the sake of considering the general cases, here we shall assume that crossings \( \omega_{c\nu} > 0 \) are distinct.

Since \( RT \) is determined in a concise form by the property of Walton’s polynomial \( W(\omega^2) \) as per (2.13) and Remark 3, we do not need to calculate the \( RT_\nu \) explicitly using (2.12) in most cases.
CHAPTER 2. PRELIMINARIES

2.5 Conclusion

In this chapter, we present the key concepts related to stability of LTI systems and specifically for SISO systems with a single loop delay. Starting with the retarded type system, we utilized frequency domain techniques to summarize the conditions determining stability switches in the delay parameter space. While frequency domain techniques have been studied since 1960s for the stability stable of different types of TDS, existing work so far did not address the control design problem at hand.

Based on the above discussions, there are three items of key importance: Imaginary crossing stands for a pair of pure imaginary roots \( s = \pm j\omega \), where these roots can cross from LHP to RHP (or from RHP to LHP); critical delay values are the delays related to these imaginary crossings, lying periodically on the delay axis for a given crossing \( \omega \); root tendency is associated with the sensitivities at \( s = \pm j\omega \) with respect to delay \( \tau \). With the knowledge of the above three items, the picture of stability intervals along the delay axis can be depicted via counting the number of unstable roots of the system, see \([1], [2], [8], [13], [12], [23], [22], [26]\). Through this knowledge, we can formulate the necessary and sufficient conditions for a system to have the desired stability intervals along delay axis, laying out the basis of the following Chapters 3, 4 and 5.

Needless to say, there are two difficulties in the problem at hand. Firstly, the existing literature mainly focuses on the stability “analysis” for the given TDS, or the design of only one or two parameters. The state of the art reached a level that now drawing a stability plot or chart with one or two parameters can be considered quite simple, however, multiple parameter design is still a challenging task. Also, due to the computational capability of any software, it is not possible to run a loop of even more than three nested-loops. Moreover, we cannot perform plotting in the case of multi-parameter design. Secondly, the previous control design studies mostly concentrate on creating one stability interval (delay margin design), which do not extend to the design of multiple stability intervals. With these difficulties in mind, we will build up a new approach based on nonlinear programming optimization in the following chapters.
Chapter 3

Delay Margin Design

In many control problems affected by delays, the system maintains its stability for $\tau \in [0, \bar{\tau})$, and is in transition from stability to instability at $\tau = \bar{\tau}$. This $\bar{\tau}$ is known as the “delay margin” [22]. Several frequency-domain stability analysis techniques adapted for delay margin design via system parameters analysis, especially with respect to one or two-dimensional parameters are investigated in [11, 17]. In this chapter, we solve the same delay margin design problem for retard class LTI SISO systems using our newly developed method explained below, using nonlinear optimization tools.

3.1 Control Design Approach

As shown in Fig. 2.1, given $\bar{\tau}$, we focus on designing a controller $C(s)$ for an open-loop plant with transfer function $G(s)$ subject to delay, achieving closed-loop stability for $\tau \in [0, \bar{\tau})$ (Fig. 4.1). Notice that we do not impose any conditions on stability for $\tau > \bar{\tau}$. That is, the system can always be unstable $\forall \tau, \tau > \bar{\tau}$, or it may recover stability for $\exists \tau, \tau > \bar{\tau}$. Regardless of these scenarios, the aim here is, given $\bar{\tau}$, to achieve stability for $\tau \in [0, \bar{\tau})$ and marginal stability at $\bar{\tau}$, by designing $C(s)$.

Based on the knowledge presented in Chapter 4, we propose to design the controller $C(s)$ based on the following procedures:

A. Structure of $C(s)$

The plant transfer function is written as $G(s) = N(s)/D(s)$, which is given. Moreover, the controller transfer function is written as $C(s) = P(s)/Q(s)$, which contains all the unknown
CHAPTER 3. DELAY MARGIN DESIGN

Figure 3.1: Desired stability decomposition of the closed loop system, given \( \bar{\tau} \) parameters \( k_i \) to be designed. In this case, we have the following system characteristic function

\[ F(s, e^{-s\tau}) = D(s)Q(s) + N(s)P(s)e^{-s\tau}, \quad (3.1) \]

where \( A(s) \) reads as \( D(s)Q(s) \), and \( B(s) \) reads as \( N(s)P(s) \) in (2.1). This characteristic equation (3.1) has infinitely many zeros \( \bar{s} \), which determine the stability/instability characteristics of the system as a parameter of \( \tau \). Based on the assumptions made, the degree \( n \) of \( D(s)Q(s) \) is larger than that of \( N(s)P(s) \), qualifying the system to be of retarded type. Hence, stability of this system is guaranteed if and only if all \( \bar{s} \) have negative real parts. Moreover, system stability can be lost if at least one of \( \bar{s} \) touches the imaginary axis \( \bar{s} = j\omega \).

B. Number of the crossings

From (2.5), we can rearrange Walton’s equation into

\[ W(\omega^2) = |D(j\omega)Q(j\omega)|^2 - |N(j\omega)P(j\omega)|^2 = 0, \quad (3.2) \]

which is an even polynomial in \( \omega \). Then the number of \( \omega > 0 \) solutions, \( M \) is of key consideration as was emphasized in [17]. Since the delay margin \( \tau = \bar{\tau} \) is related to the stability/instability transition, there must be at least one crossing, denoted by \( \omega_{c1} \). However, by the nature of the problem setup, the closed-loop system may have more than one crossing. At this point, the designer has two choices:

- **Prevent**: Impose additional conditions such that the remaining candidate \( \omega \) solutions are infeasible. This can be achieved by forcing those candidate solutions to be complex numbers in (3.2).

- **Allow**: Accept the occurrence of additional crossing frequencies, but reject \( C(s) \) solutions for which the critical delays of the closed-loop system corresponding to these unaccounted crossings interfere with the intended stability intervals.

C. Two special points of \( \tau \)
CHAPTER 3. DELAY MARGIN DESIGN

We have two additional items to be considered: \( \tau = 0 \) and \( \tau = \bar{\tau} \) points of the closed-loop system.

At \( \tau = 0 \), Routh’s array of \( F(s, 1) \) in (3.1) is of concern, where the Number of Sign (NS) changes in the first column of the array elements determine the number of unstable roots of the system \([29]\). Thus, delay-free \( (\tau = 0) \) system is stable if and only if \( NS = 0 \).

At \( \tau = \bar{\tau} \), stability/instability transition arises with respect to the crossing \( s = j\omega_{c1} \). Then we must set the minimum positive of the critical delay value of \( \omega_{c1} \) equal to \( \bar{\tau} \) based on (2.8), and its root tendency \( RT_1 = +1 \) using (2.13). For example, in this delay margin design problem, if the designer chooses the option \( allow \), we also have to enforce the minimum positive of the critical delay values of the additional crossings to be larger than the delay margin \( \bar{\tau} \), guaranteeing that they will not interfere in the desired delay decomposition shown in Fig. 3.1, see discussions on this in \([17]\).

D. \textit{fmincon} optimization

When we list all the constraints from the above procedures A-C, and solve for the unknown parameters \( k_i \), we can consider this design problem as a nonlinear programming problem based on optimization theory. In this case, an optimization toolbox, namely \textit{fmincon} tool of MATLAB \([30]\), can be utilized to solve this problem.

3.2 Example Case Study

Take the delay margin \( \bar{\tau} = 1 \), \( G(s) \) as a second-order plant \( \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \), and \( C(s) \) as a PID controller \( k_p + \frac{k_i}{s} + k_ds \) with \( k_p, k_i \) and \( k_d \) being the design parameters. In this case, we can rewrite (5.1) as

\[
(s^2 + 2\zeta\omega_n s + \omega_n^2)s + (k_ds^2 + k_ps + k_i)\omega_n^2 e^{-\tau s}.
\]  

where \( A(s) \) reads \( (s^2 + 2\zeta\omega_n s + \omega_n^2)s \) and \( B(s) \) reads \( (k_ds^2 + k_ps + k_i)\omega_n^2 \). Then, Walton’s polynomial reads

\[
W(\omega^2) = \omega^6 + (4\zeta^2\omega_n^2 - k_d^2\omega_n^4 - 2\omega_n^2)\omega^4 + (-k_p\omega_n^4 + 2k_i k_d\omega_n^4 + \omega_n^4)\omega^2 - k_i^2\omega_n^4 = 0. \]  

Following our design approach, we choose the option \textit{prevent}, and guarantee the number of crossings \( M \) to be 1. As per Remark 8, equation (3.4) has only one positive real root. With \( \sigma = \omega^2 \), this equation can be rewritten as

\[
W(\sigma) = \sigma^3 + l_2\sigma^2 + l_1\sigma + l_0 = 0,
\]  

11
where

\[ l_2 = 4\zeta^2\omega_n^2 - k_d^2\omega_n^4 - 2\omega_n^2, \quad l_1 = -k_p^2\omega_n^4 + 2k_i k_d\omega_n^4 + \omega_n^4, \quad l_0 = -k_i^2\omega_n^4. \]

Since \( W(\sigma) \) is of order-3 with respect to \( \sigma \), and negative of the discriminant of a 3\textsuperscript{rd} order polynomial indicates one real pole and two complex poles [31], we can make sure only one real root exists using this condition, thereby rendering a crossing only at \( \omega_{c1} \).

**Remark 5** As per Remark 4, since in this case there is only one crossing, the crossing direction of \( j\omega_{c1} \) is automatically guaranteed to be +1.

Next, setting the unknowns to be \( \omega_{c1}, k_p, k_i \) and \( k_d \), all the constraints are listed as follows:

- **I. Imaginary crossing value**

\[ \omega_{c1}^6 + (4\zeta^2\omega_n^2 - k_d^2\omega_n^4 - 2\omega_n^2)\omega_{c1}^4 + (-k_p^2\omega_n^4 + 2k_i k_d\omega_n^4 + \omega_n^4)\omega_{c1}^2 - k_i^2\omega_n^4 = 0. \] (3.6)

- **II. Negative discriminant**

\[ \Delta = l_1^2 l_2^2 - 4l_1^3 l_2 - 27l_0^2 + 18l_1 l_2 < 0, \] (3.7)

where

\[ l_2 = 4\zeta^2\omega_n^2 - k_d^2\omega_n^4 - 2\omega_n^2, \quad l_1 = -k_p^2\omega_n^4 + 2k_i k_d\omega_n^4 + \omega_n^4, \quad l_0 = -k_i^2\omega_n^4. \]

- **III. Delay free system stability**

\[ k_i\omega_n^2 > 0, \quad k_d\omega_n^2 + 2\zeta\omega_n > 0, \quad k_p\omega_n^2 + \omega_n^2 - \frac{k_i\omega_n^2}{k_d\omega_n^2 + 2\zeta\omega_n} > 0. \] (3.8)

- **IV. Minimum critical delay value**

\[ \bar{\tau} = \begin{cases} \arctan \frac{2(y, x)}{\omega_{c1}}, & \text{if } \arctan 2(y_1, x_1) \geq 0; \\ \arctan \frac{2(y, x) + 2\pi}{\omega_{c1}}, & \text{if } \arctan 2(y_1, x_1) < 0. \end{cases} \] (3.9)

where

\[ x = \Re \left( \frac{\omega_n^2(k_d s^2 + k_p s + k_i)}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} \right) \bigg|_{s=j\omega_{c1}}, \]

\[ y = \Im \left( \frac{\omega_n^2(k_d s^2 + k_p s + k_i)}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} \right) \bigg|_{s=j\omega_{c1}}. \]
CHAPTER 3. DELAY MARGIN DESIGN

The above constraints I-IV formulate the sufficient conditions to design a controller to achieve the pre-determined delay margin for the system. To solve for the unknown vector \( \chi = [\omega_c, k_p, k_i, k_d] \) satisfying the above constraints, an optimization problem is built up using \texttt{fmincon} in MATLAB, with the objective function favoring the largest “speed of the crossing”, which means the absolute value of the sensitivity of \( s \) to \( \tau \). If it is large it means the eigenvalue will enter more deep into LHP. That might be a way to optimize the gains where the objective function is to maximize. In this case, we select the objective function to minimize as,

\[
f = - \left[ \frac{1}{\omega_{cr}} \left( \frac{B'(j\omega_{cr})}{B(j\omega_{cr})} - \frac{A'(j\omega_{cr})}{A(j\omega_{cr})} \right) \right].
\]

Among the feasible solution sets, we calculate the quadratic sum of PID controller gains \( k_p^2 + k_i^2 + k_d^2 \), and pick the smallest value in some sense selecting a controller with minimal effort.\(^1\)

Set \( \zeta = 0.7 \) and \( \omega_n = 5 \) for a stable plant. First, we pick \([0,0,0,0]\) as the initial condition, and obtain the following feasible solution with the minimum quadratic sum of gains, \( [\omega_c, k_p, k_i, k_d] = [2.4947, 1.0203, 0.5220, 0.1258] \). We next use these parameters in a PID controller \( C(s) \) in SIMULINK and run unit step response simulations of the closed-loop system with delay values, \( \tau = 0, 0.95, 1, 1.05 \) (seconds). As simulations show in Fig. 3.2, the stability of the delay-free system and the delay margin \( \bar{\tau} = 1 \) match perfectly with the results for the control design approach.

Since the whole control design approach has been formulated in a self-contained MATLAB m.file, we then investigate effectively how the controller gains \( k_p, k_i \) and \( k_d \) change when different \( \zeta \) values are picked in the second order plant. Setting \( \omega_n = 5 \) and \( \zeta =0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 \), the feasible solutions\(^2\) with the minimum quadratic sum of controller gains are obtained, and the details of each solution vector \( [\omega_c, k_p, k_i, k_d] \) are given in Table 3.1. We next plot the corresponding \( k_p, k_i \) and \( k_d \) gains in Fig 3.3 under different damping ratios \( \zeta \) of the open loop plant. From this figure, one notices that with the increase of \( \zeta, k_p \) must be increased to maintain system stability for \( \tau \in [0, 1) \). However, when \( \zeta \) is close to 1, \( k_p \) tends to maintain the same level. On the other hand, \( k_i \) decreases rapidly and \( k_d \) increases slowly, both monotonically with the increase of \( \zeta \). At \( \zeta = 0.4, k_d \) is approximately zero, and hence we can have a PI at this point. We also perform SIMULINK simulations of the closed loop for each case, which matches the designed delay margin \( \bar{\tau} = 1 \) perfectly (simulations suppressed).

\(^1\)Details about how to formulate a \texttt{fmincon} solver are provided in Section 4.1

\(^2\)While the optimization routine converges successfully for \( \zeta < 0.4 \), the arising controller gains correspond to a different minimum of the objective function, not presenting continuous changes in controller gains. This part of the computation is suppressed to prevent confusion.
CHAPTER 3. DELAY MARGIN DESIGN

Figure 3.2: Example: Unit step response of the closed loop system simulations at $\tau = 0, 0.95, 1, 1.05$, with the controller $C(s)$ designed using the optimization method described in Section 3.1 and the delay margin $\bar{\tau} = 1$

Figure 3.3: PID controller gains under different $\zeta$ values with $\omega_n = 5$
CHAPTER 3. DELAY MARGIN DESIGN

<table>
<thead>
<tr>
<th>ζ</th>
<th>ωc1</th>
<th>kp</th>
<th>ki</th>
<th>kd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>1.9509</td>
<td>0.6041</td>
<td>1.3107</td>
<td>0.0001</td>
</tr>
<tr>
<td>0.5</td>
<td>2.2396</td>
<td>0.8470</td>
<td>0.9148</td>
<td>0.0264</td>
</tr>
<tr>
<td>0.6</td>
<td>2.3872</td>
<td>0.9549</td>
<td>0.7803</td>
<td>0.0903</td>
</tr>
<tr>
<td>0.7</td>
<td>2.4947</td>
<td>1.0203</td>
<td>0.5220</td>
<td>0.1258</td>
</tr>
<tr>
<td>0.8</td>
<td>2.6188</td>
<td>1.0472</td>
<td>0.2509</td>
<td>0.1755</td>
</tr>
<tr>
<td>0.9</td>
<td>2.7702</td>
<td>1.0077</td>
<td>0.0466</td>
<td>0.2507</td>
</tr>
<tr>
<td>1.0</td>
<td>2.8095</td>
<td>1.0133</td>
<td>0.0392</td>
<td>0.3037</td>
</tr>
</tbody>
</table>

Table 3.1: Obtained PID gains for different ζ values, when ωn = 5 and τ = 1

3.3 Conclusion

For the controller design to render a certain delay margin in the closed-loop system, we formulated the problem as an optimization problem. With the consideration of imaginary crossings, critical delay values, and root tendency, we could design a controller with multiple parameters, considering the conditions with respect to the number of crossings, exact value of the crossings, minimum positive delay value of the crossings corresponding to the delay margin value, and stability of the delay-free case.

An example of PID controller design for a second order plant is studied through the developed procedure. Besides PID controller gains kp, ki and kd, we also design the crossing value ωc1, in light of the above listed constraints. Using the MATLAB toolbox fmincon, the solution satisfying the listed constraints are easily solved. This optimization-based approach can also be adapted into the following Chapters 4 and 5.
Chapter 4

Design of Two Stable Intervals: The Case with a Stable Plant

As discussed in Chapter 3, we design the delay margin to let the system become stable for \( \tau \in [0, \bar{\tau}) \). In most cases, we desire to set \( \bar{\tau} \) as large as possible, in order to cover the working delay in the loop. However, in some real-world applications, there can be some limitations on system parameters, which will also cause a limitation for the delay margin. In such situations, we propose another strategy, which is to design two different stability intervals instead of one, and to have the second stability interval cover the working delay point. This new approach is well stated in the rest of this chapter.

4.1 Control Design Approach

In this chapter, the main interest is to design a controller \( C(s) \) for an open-loop plant with transfer function \( G(s) \) subject to delay, achieving closed-loop stability in two pre-determined delay intervals (Fig. 4.1). While the examples in this section will focus on stable \( G(s) \), the methodology presented in the sequel applies equally for both stable and unstable plant transfer functions.

To keep the presentation compact, we first take \( G(s) \) as a second-order stable plant\(^1\),

\[
G(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}.
\]

(4.1)

Following the standard definitions, here \( \zeta \) and \( \omega_n \) are defined respectively as damping ratio and natural frequency of the open-loop plant.

\(^1\)See discussion in Remark on how to extend the approach to higher-order systems.
For the design of $C(s)$, the following conditions must be met:

**Condition A.** The SISO LTI closed-loop system is stable for the delay-free case ($\tau = 0$);

**Condition B.** Given $h_1$, $h_2$ and $h_3$ with $0 < h_1 < h_2 < h_3$, the SISO LTI closed-loop system is stable for delays satisfying $\tau \in [0, h_1) \cup (h_2, h_3)$.

**Condition C.** The controller $C(s)$ itself is stable.

We propose to design the controller $C(s)$ based on the following steps:

**Step 1:** Assignment of $\tau = h_1$, $h_2$, $h_3$ to crossings $\omega_{cl}$.

The system has two separate stable intervals in the parameter space of $\tau$. Specifically, the first three critical delay values, which cause stability-instability switches, are denoted by $h_1$, $h_2$ and $h_3$.

A strategic decision here is that the number of crossings $M$ will be set to 2 (see also Step 3), where we have two crossing values $\omega_{c1}$ and $\omega_{c2}$. Due to Remark 4, note that we have $\omega_{c1} > \omega_{c2}$, with $s = j\omega_{c1}$ causing destabilizing crossing and $s = j\omega_{c2}$ creating a stabilizing crossing. Accordingly $h_1$ and $h_3$ are assigned to the same crossing $\omega_{c1}$, while being the successive delays values causing this crossing. That is, $\tau_{0,1} = \tilde{\tau}_1 = h_1$, and $\tau_{1,1} = h_3$ in (2.9), hence $h_3 = h_1 + \frac{2\pi}{\omega_{c1}}$. Next, $\omega_{c2}$ is assigned to $h_2$, which corresponds to $\tau_{0,2} = \tilde{\tau}_2 = h_2$ in (2.5).

Moreover, since $h_1$ is the minimum positive $\tilde{\tau}_1$ value with respect to $\omega_{c1}$ and $h_3$ is the successive crossing delay $\tau_{1,1}$, we have the condition $2h_1 < h_3$. In other words, although $h_1$, $h_2$, $h_3$ can be arbitrarily selected by the designer, $2h_1 < h_3$ must still be respected. In general, the initial selection of $h_1$ and $h_3$ follows $h_3 = h_3^*$ where

$$h_3^* = \begin{cases} h_3, & \text{if } 2h_1 < h_3; \\ 2h_1 + \lambda, & \text{if } 2h_1 \geq h_3, \end{cases}$$

and $\lambda$ is a positive number.

Now that $h_1$ and $h_3$ are given, $\omega_{c1}$ can be calculated from (2.9)

$$\omega_{c1} = \frac{2\pi}{h_3 - h_1}.$$
CHAPTER 4. DESIGN OF TWO STABLE INTERVALS: THE CASE WITH A STABLE PLANT

Step 2: Structure of $C(s)$

Let $C(s)$ be in the form of $\frac{N(s)}{D(s)}$, where $\deg[D(s)] \geq \deg[N(s)]$. Since the design of stability intervals in Fig. 4.1 relies on the infinite dimensional system characteristic equation, one must make sure there are sufficiently many design parameters to achieve this goal. For example, the polynomials $D(s)$ and $N(s)$ can be expressed as follows:

$$D(s) = s^2 + \alpha_1 s + \alpha_0, \quad N(s) = \beta_2 s^2 + \beta_1 s + \beta_0,$$

where $\alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2 \in \mathbb{R}$ are design parameters.

The characteristics equation in this case becomes

$$F(s, e^{-s\tau}, \alpha_k, \beta_k) = (s^2 + \alpha_1 s + \alpha_0)(s^2 + 2\zeta \omega_n s + \omega_n^2) + (\beta_2 s^2 + \beta_1 s + \beta_0)e^{-s\tau},$$

where the general form in (4.1) reads

$$\begin{align*}
A(s) &= (s^2 + \alpha_1 s + \alpha_0)(s^2 + 2\zeta \omega_n s + \omega_n^2), \\
B(s) &= \beta_2 s^2 + \beta_1 s + \beta_0.
\end{align*}$$

At this point, we have 6 unknowns $\omega_c, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2$ in total, denoted by the vector $\chi = [\omega_c, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2]$. Next we formulate the constraints that this vector should comply with.

Step 3: Constraints $\mathcal{C}$ for all specifications

From (4.4), Walton’s equation can be written as

$$W(\omega^2) = \omega^8 - (4\omega_0^2 \zeta^2 - \alpha_1^2 + 2\omega_n^2 + 2\alpha_0)\omega^6
+ (4\alpha_1^2 \omega_n^2 \zeta^2 - 8\alpha_0 \omega_n^2 \zeta^2 - 2\alpha_1^2 \omega_n^2 + \omega_n^4 + 4\alpha_0 \omega_n^2 + \alpha_0^2 - \beta_2^2)\omega^4
- (-4\alpha_0 \omega_n^2 \zeta^2 - \alpha_1^2 \omega_n^4 + 2\alpha_0 \omega_n^4 + 2\alpha_0 \omega_n^2 - 2\beta_0 \beta_2 + \beta_1^2)\omega^2 + (\alpha_0^2 \omega_n^4 - \beta_0^2).$$

Through (4.5), we have 2 constraints related to the imaginary axis crossing values:

$$\begin{align*}
\mathcal{C}_1 : W(\omega_{c1}^2) &= 0, \\
\mathcal{C}_2 : W(\omega_{c2}^2) &= 0.
\end{align*}$$

As for the minimum positive critical delay value, refering to (4.8), we revise $x$ and $y$ variables as

$$\begin{align*}
x_k &= \Re \left( -\frac{\beta_2 s^2 + \beta_1 s + \beta_0}{(s^2 + \alpha_1 s + \alpha_0)(s^2 + 2\zeta \omega_n s + \omega_n^2)} \right)_{s = j\omega_{c2}}, \\
y_k &= \Im \left( -\frac{\beta_2 s^2 + \beta_1 s + \beta_0}{(s^2 + \alpha_1 s + \alpha_0)(s^2 + 2\zeta \omega_n s + \omega_n^2)} \right)_{s = j\omega_{c2}}.
\end{align*}$$
4. In this case, the set of solutions satisfying the three inequalities is denoted by \( \mathcal{G}_3 \). We have to check the discriminant of \( \delta < 0 \).

5. In order to guarantee the number of crossings \( M \) to be 2, we need to make sure \((4.8)\) has only 2 distinct positive real roots as per Remark 3. Firstly, \((4.8)\) with \( \sigma = \omega^2 \) can be rewritten as:

\[
\sigma^4 + r_3 \sigma^3 + r_2 \sigma^2 + r_1 \sigma + r_0 = 0,
\]

where

\[
\begin{align*}
  r_3 &= -(-4\omega_{c1}^2 \xi^2 - \alpha_1^2 + 2\omega_n^2 + 2\alpha_0),
  r_2 &= 4\alpha_1^2 \omega_{n1}^2 \xi^2 - 8\alpha_0 \omega_{n1}^2 \xi^2 - 2\alpha_1^2 \omega_n^2 + \omega_1^4 + 4\alpha_0 \omega_n^2 + \alpha_0^2 - \beta_2^2,
  r_1 &= -(-4\alpha_0^2 \omega_{n1}^2 \xi^2 - \alpha_1^2 \omega_n^4 + 2\alpha_0 \omega_n^4 + 2\alpha_0^2 \omega_n^2 - 2\beta_0 \beta_2 + \beta_1^2),
  r_0 &= \alpha_0^2 \omega_n^4 - \beta_0^2.
\end{align*}
\]

Since \( \omega_{c1}^2 \) and \( \omega_{c2}^2 \) must be the two roots of \((4.6)\), one factor of \((4.6)\) is \((\omega_{c1}^2 + \omega_{c2}^2) \sigma + \omega_{c1} \omega_{c2}^2 \). Then, using factorization in \((4.6)\), the other factor must be

\[
\sigma^2 + (r_3 + \omega_{c1}^2 + \omega_{c2}^2) \sigma + \frac{r_0}{\omega_{c1} \omega_{c2}} = 0.
\]

To make sure \( M = 2 \) with the option prevent, the zeros of \((4.7)\) in \( \sigma \) should not be positive. Owing to this, we have to check the discriminant of \((4.7)\):

\[
\delta = (r_3 + \omega_{c1}^2 + \omega_{c2}^2)^2 - \frac{4r_0}{\omega_{c1} \omega_{c2}}.
\]

The following two scenarios are to be considered:

I. Eq.\((4.7)\) has two complex zeros, that is \( \delta < 0 \). Denote by \( I_1 \) the set of all solutions \( \chi \) satisfying \( \delta < 0 \).

II. Eq.\((4.7)\) has two negative real zeros, that is, \( \delta > 0 \) & \( r_3 + \omega_{c1}^2 + \omega_{c2}^2 > 0 \) & \( \frac{r_0}{\omega_{c1} \omega_{c2}} > 0 \).

In this case, the set of solutions satisfying the three inequalities is denoted by \( I_2 \). Consequently, the following constraint guarantees that \( M = 2 \) holds,

\[ \mathcal{G}_5 : \chi \in I_1 \cup I_2. \]
Remark 6 As per Remark 4 and $M = 2$ based on $C_5$, and $\omega_{c1} > \omega_{c2}$ holds as per Step 1, the crossing directions of $j\omega_{c1}$ and $j\omega_{c2}$ are automatically guaranteed and therefore do not need to be explicitly considered.

Next, from Condition A that the delay-free system is stable, Routh’s array on $F(s, 1)$ in (4.4) can be implemented. Thus, for the delay-free ($\tau = 0$) closed-loop system stability, this constraint becomes:

$$C_6 : NS = 0.$$ 

which is formulated specifically for the problem at hand, as:

$$2\alpha_0\omega_n\zeta + \alpha_1\omega_n^2 + \beta_1 - \frac{(2\omega_n\zeta + \alpha_1)(\alpha_0\omega_n^2 + \beta_0)}{2\alpha_1\omega_n\zeta + \omega_n^2 + \alpha_0 + \beta_2 - \frac{2\alpha_0\omega_n\zeta + \alpha_1\omega_n^2 + \beta_1}{2\omega_n\zeta + \alpha_1}} > 0,$$

$$\& 2\alpha_0\omega_n\zeta + \omega_n^2 + \alpha_0 + \beta_2 - \frac{2\alpha_0\omega_n\zeta + \alpha_1\omega_n^2 + \beta_1}{2\omega_n\zeta + \alpha_1} > 0,$$

$$\& \alpha_0\omega_n^2 + \beta_0 > 0,$$

$$\& 2\omega_n\zeta + \alpha_1 > 0.$$ 

For Condition C, where the controller $C(s)$ must be a stable one, the constraint that the roots of the denominator $s^2 + \alpha_1 s + \alpha_0$ of $C(s)$ to be in LHP is given by:

$$C_7 : \alpha_0 > 0 \ \& \ \alpha_1 > 0.$$ 

Finally, from Step 1, we can write down another constraint:

$$C_8 : \omega_{c2} < \omega_{c1}.$$ 

It should be noted that constraints $C_1-8$ are sufficient conditions for the system to satisfy Conditions A-C.

Step 4: fmincon optimization

To solve for the vector $\chi = [\omega_{c2}, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2]$ satisfying the above listed constraints, we can consider this design problem as an optimization problem. A solution to $\chi$ may not exist, and if it does, it may not be unique. Since the aim is to find a solution, multiple solutions are however of no concern.

It can be observed that the type of constraints $C_1-8$ we have matches that of the nonlinear optimization toolboxes, namely fmincon tool of MATLAB. On the other hand, we so far did not formulate an optimization problem. For this, an objective function $f$ is needed. Among many different
CHAPTER 4. DESIGN OF TWO STABLE INTERVALS: THE CASE WITH A STABLE PLANT

ways one can formulate this function, here we propose to minimize the sum of the magnitudes of the coefficients in the controller, that is,

\[ f = \alpha_0^2 + \alpha_1^2 + \beta_0^2 + \beta_1^2 + \beta_2^2. \]  

(4.9)

Notice that choice of \( f \) is ad-hoc and is introduced so that the nonlinear solver can be utilized. This way the constraint equations can be enforced while designing the controller coefficients \( \alpha_m, \beta_v \).

With the objective function (4.9), it now becomes possible to utilize \textit{fmincon} tool in MATLAB. Since we have a small-medium scale optimization problem with the number of variables \(< 100\), Sequential Quadratic Programming method can be used. In this method, \textit{fmincon} solves a Quadratic Programming subproblem at each iteration. An estimate of the Hessian of the Lagrangian is updated at each iteration and a line search is performed using a merit function. An active set strategy is used to solve the programming subproblem [30].

For the implementation of \( \mathcal{C}_{1-8} \) in the standard form for the \textit{fmincon} solver, we have the following adaptations,

\[
\begin{align*}
\mathcal{C}_1 & \rightarrow c_{eq1}(\alpha, \beta) = 0, \\
\mathcal{C}_2 & \rightarrow c_{eq2}(\chi) = 0, \\
\mathcal{C}_3 & \rightarrow c_{eq3}(\alpha, \beta) = 0, \\
\mathcal{C}_4 & \rightarrow c_{eq4}(\chi) = 0, \\
\mathcal{C}_5 & \rightarrow c_1(\chi) \leq 0, \\
\mathcal{C}_6 & \rightarrow c_2(\alpha, \beta) \leq 0 \text{ & } A \cdot \chi \leq b, \\
\mathcal{C}_{7-8} & \rightarrow lb \leq \chi \leq ub.
\end{align*}
\]

\textbf{Remark 7} In the implementation of \textit{fmincon} active-set algorithm, tolerances for inequality and upper/lower bound constraints must be set. With \( \Delta_k \) being small positive values (or vectors with all
the elements being small positive values), constraints $C_{1-8}$ are revised as

$$
C_1 \rightarrow c_{eq}(\alpha, \beta) = 0,
C_2 \rightarrow c_{eq}(\chi) = 0,
C_3 \rightarrow c_{eq}(\alpha, \beta) = 0,
C_4 \rightarrow c_{eq}(\chi) = 0,
C_5 \rightarrow c_1(\chi) \leq -\Delta_1,
C_6 \rightarrow c_2(\alpha, \beta) \leq -\Delta_2 & A \cdot \chi \leq b - \Delta_3,
C_7-8 \rightarrow lb + \Delta_4 \leq \chi \leq ub - \Delta_5.
$$

**Remark 8** Fmincon numerical tool is implemented under several considerations: (i) firstly in the domain of optimization constraints are assumed to be continuous functions in terms of their arguments; (ii) we have that the obtained result, if any, will be very likely a local minimum; (iii) in general there are no set rules as to how one selects a feasible initial point.

Consideration (i) is neglected here partly because all the constraints, possibly except at some special singularity points, will remain as continuous functions of their arguments. For consideration (ii), since we only care about whether the constraints are satisfied in this problem, the local minimum $\chi$ point is equally admissible. Finally, we have to pay attention to consideration (iii), in which initial condition could affect whether this optimization method leads to an admissible solution. A rule for selecting an appropriate initial condition is well stated in the Appendix.

Based on the Appendix A, a rigorous way of formulating an initial condition follows

$$
\begin{aligned}
\omega_{c2} &\rightarrow \text{ slightly less than } \omega_{c1}; \quad \omega_{c2} = \omega_{c1} - \epsilon_3, \\
\alpha_0 &\rightarrow \text{ slightly less than } \omega_{c1}^2; \quad \alpha_0 = \omega_{c1}^2 - \epsilon_0, \\
\alpha_1 &\rightarrow \epsilon_1, \\
\beta_0, \beta_1, \beta_2 &\rightarrow \text{ a small positive value } = \epsilon_2.
\end{aligned}
$$

where $\epsilon_l$ are small positive values. In other words, with the above proposed initial condition, the characteristic equation in (4.4) becomes

$$
F(s, e^{-st}) \cong (s^2 + \epsilon_1 s + (\omega_{c1} - \epsilon_0)^2)(s^2 + 2\zeta \omega_n s + \omega_n^2) + \epsilon_2(s^2 + s + 1)e^{-st}.
$$

(4.10)
CHAPTER 4. DESIGN OF TWO STABLE INTERVALS: THE CASE WITH A STABLE PLANT

The small gains of $\beta_k = \epsilon_2$ will not let the time delayed term $e^{-\tau s}$ render a crossing, and the system is almost marginally stable for a large interval of $\tau$ due to the factor $s^2 + \epsilon_1 s + (\omega_c - \epsilon_0)^2$.

As we tested, this rule for the initial condition works well for a large interval of $\omega_n$, and for $\zeta$ from 0 (undamped) to some value above 1 (over-damped); see details in the next subsection. For some cases, this rule may not work well for undamped systems as this creates potentially an additional crossing at $\omega_n$, violating our design constraints at the very first step of our problem setup. In such cases we may need to propose higher order controllers for $C(s)$, which can be done once again by following the above steps.

Remark 9 Designing a controller for higher-order open-loop systems that leads to a closed-loop system with two separate stability intervals is even more challenging. Considering this, a practicable approach is to design a low order controller $C_1(s)$ first following the above Steps 1-4, then to set another controller $C_2(s)$ in cascade to $C_1(s)$, where $C_2(s)$ is carefully designed based on the feasible solution points revealed when designing $C_1(s)$. By each iteration of designing $C_k(s)$ in cascade, a higher order system can be built. An example of this is presented in the next section.

Remark 10 In the case when $m = n$ in (2.2), we have a neutral class system. For such systems, stability may not be preserved as $\tau : 0 \rightarrow 0^+$ and could be lost for all $\tau > 0$, even if the delay-free ($\tau = 0$) system is stable. To prevent this on the LTI SISO system at hand, an additional condition is needed, which is $|b_n| < 1$ in (2.2) [12].

4.2 Example Cases Study

4.2.1 Example 1: A Computational Study

Take $\zeta = 0.7$ and $\omega_n = 10$ rad/sec in $G(s)$ and pick $h_1 = 0.1, h_2 = 0.5, h_3 = 0.7$ in Condition C (see also Fig. 4.1 and Fig. 2.1). Following Steps 1-4, $\mathscr{C}_{1-8}$ can be written down as constraints within the fmincon solver in MATLAB.

Furthermore, in $\mathscr{C}_5$, we reconstruct the feasible interval of $I_1 \cup I_2$ to simplify the way of expressing the constraints. For the two roots of (4.7) to become complex conjugate, the following constraint denoted by set $I_1$ must hold,

$$I_1 : \ (4\omega_n^2\zeta^2 + \alpha_1^2 - 2\omega_n^2 - 2\alpha_0 + \omega_c^2) - \frac{4(\alpha_0^2\omega_n^4 - \beta_0^2)}{\omega_c^2} < 0.$$
CHAPTER 4. DESIGN OF TWO STABLE INTERVALS: THE CASE WITH A STABLE PLANT

Then for the two roots of (4.7) to lie in LHP, two constraints denoted by set $I_3$ must hold as per Routh’s Array,

$$I_3 : 4\omega_n^2\zeta^2 + \alpha_1^2 - 2\omega_n^2 - 2\alpha_0 + \omega_c^2 + \omega_c^2 > 0 \quad \& \quad \frac{\omega_0^2 - \beta_0^2}{\omega_c^2 \omega_c^2} > 0.$$ 

Obviously, $I_1 \cup I_3 \equiv I_1 \cup I_2$, hence the above two constraints are inserted into the fmincon solver.

Next, we test whether initial condition works well. Given $h_1$ and $h_3$, we have $\omega_c^1 = \frac{2\pi}{h_3 - h_1} = 10.4720$, then the initial condition becomes

$$\omega_c \rightarrow \text{slightly less than } 10.4720,$$

$$\alpha_0 \rightarrow \text{slightly less than } 109.6628,$$

$$\alpha_1 \rightarrow 0,$$

$$\beta_0, \beta_1, \beta_2 \rightarrow 0.01.$$ 

The initial points for $\omega_c$ and $\alpha_0$ are then put on a grid $(p \times q)$, and slightly varied,

$$\chi_0 = [10.4720 - 0.05 \times p, 109.6628 - 0.1 \times q, 0, 0.01, 0.01, 0.01].$$

The optimization is re-run for $p \times q$ times, where $p = 1, 2, \ldots, 20, q = 1, 2, \ldots, 20$. For these 400 initial points, we find 98.5% of all the solutions of $\chi$ converge to the same point $[\omega_c, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2] = [8.7378, 91.7096, 0.0010, -1.7621, 111.457, 21.5547]$, while 1.5% of them converged to different solutions. Moreover, there were no issues with convergence. That is, all 400 runs lead to a feasible solution for $\chi$.

The solution $\chi$ that converged at the highest percentage of 98.5% is next taken for further study. In this solution, we have $\omega_c = 8.7378$ rad/sec, which is smaller than $\omega_c^1$ as expected.

Furthermore, these two crossing frequencies signify the oscillation frequency of the closed-loop system output at the critical delays; $\omega_c$ for $h_1$ and $h_3$, and $\omega_c$ for $h_2$. We next use the respective controller $C(s)$ in Simulink and run unit step response simulations of the closed-loop system with delay values, $\tau = 0, 0.095, 0.1, 0.105, 0.495, 0.5, 0.505, 0.695, 0.7, 0.705$ (seconds). As simulations indicate in Fig. 4.2, the stability of the delay-free system and the stability boundary of critical delay values match perfectly with the results obtained from the controller design method.

Furthermore, in the time window $15 - 20$ sec, at the critical values $\tau = h_1 = 0.1$ and $\tau = h_3 = 0.7$, we observe about 8.25 cycles of oscillations, corresponding to approximately about 10.36 rad/sec, which matches with $\omega_c^1 = 10.4720$, and similarly at $\tau = h_2 = 0.5$, we observe about 6.75 cycles $\approx 8.48$ rad/sec, which is very close to the exact solution $\omega_c = 8.7378$. 

24
Figure 4.2: Example 1: Unit step response of the closed loop system simulations at \( \tau = 0, 0.095, 0.1, 0.105, 0.495, 0.5, 0.505, 0.695, 0.7, 0.705 \), with the controller \( C(s) \) designed using the optimization method described in Section 4.1 and the pre-determined stability decomposition \( \tau \in [0,0.1) \cup (0.5,0.7) \)
CHAPTER 4. DESIGN OF TWO STABLE INTERVALS: THE CASE WITH A STABLE PLANT

Remark 11 Notice that the controller designed above has stable poles but close to the imaginary axis due to small $\alpha_1$ value. This controller works well in the ideal closed-loop model and simulations. However, in the presence of sensor noise that may appear in an experimental system and with possible uncertainties in $\alpha_1$, e.g., due to digital implementation, this weakly-damped controller may cause internal stability problems \[32\]. To prevent this, a lower positive bound on $\alpha_1$ can be set as another constraint (see Example 2).

4.2.2 Example 2: An Experimental Study

We next implement the controller design approach on a Rotary Servo Base Unit speed-control experiment comprising of a DC-motor with gearbox and a shaft mounted encoder, as shown in Fig. 4.3. In our experimental setup, the Rotary Servo Base Unit is linked to a computer through a National Instruments data acquisition board supported by QUARC embedded system software using MATLAB/SIMULINK toolbox. A fixed sample period of 0.001 sec and relative tolerance of $10^{-8}$ are utilized with ODE45 time integration scheme in all the experiments.

Figure 4.3: Experimental setup of Rotary Servo Base Unit with amplifier and data acquisition board

For speed control, the model of the DC-motor is captured by a first order TF from voltage
to motor shaft angular velocity, as

$$G_{\text{motor}}(s) = \frac{8.4792}{0.028s + 1}. \quad (4.11)$$

A first-order low-pass filter with cut-off frequency of 50 rad/sec is designed to block out the high frequency noise, which is inserted into SIMULINK at the encoder reading channel,

$$G_{\text{filter}}(s) = \frac{1}{0.02s + 1}. \quad (4.12)$$

The open loop TF then becomes a second-order system given by that of DC-motor in series with the filter

$$G(s) = G_{\text{motor}}(s)G_{\text{filter}}(s) = \frac{15141.375}{s^2 + 85.7143s + 1785.715}, \quad (4.13)$$

where $\zeta = 1.01419$ and $\omega_n = 42.25772$. Consistent with Fig. 2.1, we define the speed as the output of the filter, and a unity negative feedback with an artificial transport time delay is introduced in the SIMULINK interface, as shown in Fig. 4.4.

![Figure 4.4: Model of the speed control setting with a filter](image)

Here, we pick $h_1 = 0.3, h_2 = 1.0, h_3 = 1.8$ in Condition C, and follow Steps 1-4 as was done in Example 1 to bring the constraints into the fmincon solver. Given $h_1$ and $h_3$, we have

$$\omega_c = \frac{2\pi}{h_3 - h_1} = 4.1888,$$

hence the initial condition becomes

$$\begin{align*}
\omega_c &\rightarrow \text{slightly less than } 4.1888, \\
\alpha_0 &\rightarrow \text{slightly less than } 17.5460, \\
\alpha_1 &\rightarrow 0, \\
\beta_0, \beta_1, \beta_2 &\rightarrow 0.001.
\end{align*}$$
Solution set using this initial condition is computed as \([\omega_c, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2] = [3.4992, 15.5508, 0.0010, 16057.60, 853.6112, 891.4406]\). This correspond to the following numerical expression of the controller

\[
C_a(s) = \frac{891.4406s^2 + 853.6112s + 16057.60}{s^2 + 0.001s + 15.5508}.
\] (4.14)

Notice that this controller has very low damping \(\alpha_1 \approx 0\), and therefore may not be appropriate from internal stability point of view. Therefore, we change the constraint on \(\alpha_1\), now setting \(\alpha_1 \geq 0.2\) in our optimization. This yields \([\omega_c, \alpha_0, \alpha_1, \beta_0, \beta_1, \beta_2] = [3.2884, 15.0340, 0.2000, 14697.23, 1033.999, 722.7142]\), revising the numerical expression of the controller to

\[
C_b(s) = \frac{722.7142s^2 + 1033.999s + 14697.23}{s^2 + 0.2s + 15.0340}.
\] (4.15)

We next test the validity of the controller using a step input with a gain of 57.52, at \(\tau = 0, 0.25, 0.3, 0.32, 0.99, 1.0, 1.05, 1.4, 1.8, 1.85\) in both simulations and experiments, as shown in Fig. 4.5. In the figure, we see that except some minimal discrepancies, which may be due to slight variations in DC-motor dynamics and nonlinearities, the tendency of curves and stability properties can be clearly observed. Interestingly, at \(\tau = 1.4\), when the operating point is inside the second stable interval, the output of the closed-loop system has fairly good settling time and transient characteristics, which is worthy of future study.

4.2.3 Example 3: Higher Order System Example

The developed controller design approach can be applied to a higher order system as noted in Remark 9. Consider the DC motor system in Example 2, where we designed \(C(s) = C_a(s)\) in (4.14). We now propose a PID controller \(C_{\text{pid}}(s) = \frac{k_ds^2 + k_ps + k_i}{s}\) to be added in cascade to \(C_a(s)\), as depicted in Fig. 4.6. By tuning the PID gains \(k_p, k_i\) and \(k_d\), we aim to create two disjoint stability intervals following the proposed control design procedure. We find out however that \(h_1, h_2, h_3\) values in Example 2 cannot be attained with the addition of the PID controller, possibly because we have only three parameters to design, which are not sufficiently many for the over constrained optimization problem at hand. We therefore set free \(\alpha_0\) and \(\alpha_1\) in \(C_a\), to relax the parameter space.

The system shown in Fig. 4.6 has the following characteristic equation:

\[
(s^2 + \alpha_1s + \alpha_0)(s^2 + 85.7147s + 1785.7149)s + (891.4406s^2 + 853.6112s + 16058)(k_ds^2 + k_ps + k_i)e^{-s\tau} = 0,
\] (4.16)
Figure 4.5: Example 2: Step response of the motor-shaft angular velocity with controller $C_b(s)$ in both experiments and simulations at delay values $\tau = 0, 0.25, 0.3, 0.32, 0.99, 1.0, 1.05, 1.4, 1.8, 1.85$ (Experimental data: solid line; Simulation data: dashed line). Experimental results and model predictions are superimposed. Control design procedure utilizes the pre-determined stability decomposition of $\tau \in [0, 0.3) \cup (1.0, 1.8)$. 

29
which can be easily converted to the Walton’s equation from (4.5). Since the constant term in this $W(\omega^2)$ function is negative, odd number of positive real roots of $\sigma = \omega^2$ exist as per Descartes rule of signs [33]. Thus, at least three crossings will arise in this closed-loop system.

Moreover, we have $h_1$ and $h_3$ assigned to $\omega_{c1}$, while $h_2$ is associated with $\omega_{c2}$. Since the function $W(\omega^2)$ is of order-5 with respect to $\sigma = \omega^2$, in this case, using factorization in $W(\omega^2)$, we have one factor to be $\sigma^2 - (\omega_{c1}^2 + \omega_{c2}^2)\sigma + \omega_{c1}^2\omega_{c2}^2$, and the other factor is of order-3. Since negative of the discriminant of a $3^{rd}$ order polynomial indicates one real pole and two complex poles [31], using this condition, we can make sure only one positive real root exists in the remaining factor, rendering another crossing at $\omega_{c3}$.

**Remark 12** In this design procedure, we aim to maintain $\omega_{c1}$ and $\omega_{c2}$ as valid crossings while eliminating the possibility of any other crossings. In some cases, such as this one considered here however it may not be possible to completely rule out the emergence of unaccounted and undesirable crossings ($\omega_{c3}$ in this example). We therefore attempt to allow as few such crossings as possible to minimize the possibility of the arising undesirable delay values interfering our intended stability in the intervals $[0, h_1)$ and $(h_2, h_3)$.

Next, we constraint $\omega_{c3}$ to be as small as possible such that it does not cause frequent solutions of delay values in (2.9). For this, we have $\omega_{c3} < \omega_{c2}$ in the optimization environment. This is done as a necessary condition to prevent those delays values to undesirably interfere with the pre-selected delay intervals $[0, h_1) \cup (h_2, h_3)$, consistent with Remark [12]. Consequently, we have the order $\omega_{c1} > \omega_{c2} > \omega_{c3}$ with root crossing directions implied (see Remarks [4][4]) as destabilizing at $\omega_{c1}$ and $\omega_{c3}$, and stabilizing at $\omega_{c2}$.
For **sufficiency** such that one can guarantee that the two disjoint stability intervals \([0, h_1)\) and \((h_2, h_3)\) are guaranteed to exist in \(\tau\) domain, the minimum positive delay value \(\tilde{\tau}_3\) corresponding to \(\omega_{c3}\) must appear in \(\tau\) domain but at a value greater than \(h_3\). Hence, a constraint is set for this as \(\tilde{\tau}_3 > h_3\). The minimum positive delay values for \(\omega_{c1}\) and \(\omega_{c2}\) are carried over from Example 2. At last, the NS is checked for delay-free closed-loop system stability.

Considering all the above constraints, another *fmincon* solver is set with a new unknown vector \(\chi = [k_p, k_i, k_d, \alpha_0, \alpha_1, \omega_{c2}, \omega_{c3}]\). The initial condition is easily extracted from the feasible solution point in Example 2, which is \(k_p \rightarrow 1, k_i \rightarrow 0, k_d \rightarrow 0, \alpha_0 \rightarrow 15,5508, \alpha_1 \rightarrow 0.0001\) and \(\omega_{c3} \rightarrow 0\). The optimization yields the following solution point

\[
\chi = [1.1422, 0.7142, 0.0321, 15.2672, 0.0185, 3.2825, 0.3182],
\]
rendering \(\omega_{c3} = 0.3182\) relatively small as desired. Moreover, it returns \(\tilde{\tau}_3 = 6.425904\), which is the smallest delay value for which the system characteristic root will be on the imaginary axis at \(s = j\omega_{c3}\). The delay value \(\tilde{\tau}_3 = 6.425904\), as designed, is greater than \(h_3 = 1.8\) and hence does not interfere with the pre-determined stability intervals.

Next, with the PID gains found as \(k_p = 1.1422, k_i = 0.7142, k_d = 0.0321\), and the controller \(C(s)\) as

\[
C(s) = \frac{891.4406s^2 + 853.6112s + 16057.60}{s^2 + 0.0185s + 15.2672},
\]
we perform Simulink simulations of the closed loop shown in Fig. 4.6, which present agreement of system stability with the predetermined stable delay intervals \(\tau \in [0, 0.3) \cup (1, 1.8)\) (simulations suppressed).

### 4.3 Conclusion

As outlined in Chapter 3, we present here in details the optimization based approach to design two disjoint stability intervals along delay axis, with examples on stable open-loop plants. Some of these results are currently in review [43].

Starting with a general second order plant and a retard type system, **Steps 1-4** are formulated to solve this reverse problem. In general, the approach has a lot in common with the delay margin design, while there are still some distinctions that need to be addressed. Firstly, since there are two separate stability intervals with the first three delay values causing stability/instability changes being \(h_1, h_2\) and \(h_3\), at least two imaginary crossings must arise. Here, we assign \(h_1\) and
CHAPTER 4. DESIGN OF TWO STABLE INTERVALS: THE CASE WITH A STABLE PLANT

$h_3$ with the same $\omega_{c2}$, and $h_2$ with $\omega_{c1}$ for simplicity. Secondly, since the design of two stability intervals depends on the infinite dimensional system eigenvalues, there must be sufficiently many design parameters, which leads to a more complex controller design, causing the possibility of more crossings. Thus, we guarantee the number of crossings via checking the discriminant of the polynomial responsible for these crossings. Thirdly, for this nonlinear programming optimization problem, a good initial condition is required. Since there is no general rules for initial condition, we propose the method by reducing the number of the constraints and trying to start with an initial condition satisfying some special cases, as stated in Appendix A.

Moreover, three case studies are discussed in this Chapter; two on a second order plant with computational studies and an experimental study. In each case, step responses of simulations and experiments are displayed, matching perfectly with the design method. In addition, a higher order example is developed by setting a PID controller in cascade to the original controller obtained from the experimental case. The same steps are followed with straightforward changes only due to the presence of the third crossing $\omega_{c3}$. This higher order case also sheds light on the controller design approach for a plant of higher order, via designing the controller step by step in cascade, which deserves further attention in future studies.

In summary, the proposed optimization-based control design approach for a stable plant is well developed, and did address the difficulty of designing multiple parameters while achieving two stability intervals along the delay axis. We acknowledge that success of this approach depends on a number of factors as discussed above, some convergence issues were also observed in some case studies, which need to be carefully taken into account in future versions of the developed tools.
Chapter 5

Design of Two Stable Intervals: The Case with an Unstable Plant

In a SISO LTI control system setting with a delayed feedback, there is a potential limitation on the delay margin \( \bar{\tau} \) when the plant has unstable poles. A study on the upper bound \( \tau^* \) of this delay margin for certain unstable plant is proposed in the literature [35]. Due to the limitation caused by unstable poles of the plant, it is not possible to stabilize the loop for a delay \( \tau' > \tau^* \) using the delay margin design approach in Chapter 3. A remedy to this could be to design two separate stability intervals as shown in Fig. 4.1, and to let the second stability interval cover the point \( \tau = \tau' \). We summarize this approach and then provide two numerical examples in this chapter.

5.1 Motivation

As discussed in Chapter 3, computation of the delay margin \( \bar{\tau} \) for certain systems is one of the main contribution in time-delay system studies over the past few decades. Delay margin stands for the largest delay the system can withstand without losing stability, that is, the system remains stable for \( \tau \in [0, \bar{\tau}] \) and is marginally stable for \( \tau = \bar{\tau} \).

Considering an open-loop unstable plant, the controller design achieving a certain delay margin can meet inherent limitations. For instance, a recent paper [35] shows that the delay margin \( \bar{\tau} \) is upper-bounded by \( \tau^* \) for some classes of open-loop unstable plants in the SISO LTI case. This upper bound \( \tau^* \) is determined by the locations of the unstable poles of the plant with an explicit expression, no matter what kind of controller is used. The cited study reveals important messages about the control design for an open-loop unstable plant. On the one hand, given only the
information about the open-loop unstable poles of the plant \( G(s) \), we can calculate the exact upper bound \( \tau^* \) of the delay margin, independent of \( C(s) \). On the other hand, this upper bound is tight for some special cases.

In our study, we mainly focus on two cases discussed in [35]:

- **Single unstable pole**: If plant \( G(s) = G_1(s) \) has a real unstable pole at \( p \), then the upper bound of the delay margin \( \sup(\bar{\tau}) \triangleq \tau^* = \frac{2}{p} \).

- **A pair of complex unstable poles**: If plant \( G(s) = G_2(s) \), without any zeros, has unstable poles at \( re^{\pm j\phi} \) with \( \phi \in (0, \pi/2) \), then

\[
\sup(\bar{\tau}) \triangleq \tau^* = \frac{\pi}{r} \sin \phi + \max \left[ \frac{2}{r} \cos \phi, \frac{2}{r} \phi \sin \phi \right].
\]

We consider \( G_2(s) \) to be

\[
\frac{1}{(s-re^{j\phi})(s-re^{-j\phi})}
\]

without loss of generality.

Then, the question that arises is: Can we design a controller that can stabilize the loop at \( \tau' > \tau^* \)?

The answer to this question could lie in the control design approach presented in Chapter 4, which is to design two disjoint stability intervals along the delay axis, \([0, h_1) \) and \((h_2, h_3) \), with \( 0 < h_1 < h_2 < h_3 \), where \( h_2 \) and \( h_3 \) are chosen such that \( \tau' \in (h_2, h_3) \). Thus, based on a similar approach in Section 4.1, we formulate an optimization problem to solve this challenging control design problem.

### 5.2 Control Design Approach

Following Section 4.1, we present a summary of the control design approach:

**A. Assignment of \( \tau = h_1, h_2, h_3 \) to crossings \( \omega_c \).**

Propose the simplifying assumption that the crossing for \( h_1 \) and \( h_3 \) occurs at the same point on the imaginary axis, \( j\omega_{c1} = j\omega_{c3} \), while the crossing \( j\omega_{c2} \) at \( h_2 \) satisfies \( \omega_{c2} \neq \omega_{c1} \). In this case, \( h_1 \) and \( h_3 \) must satisfy the condition \( 2h_1 > h_3 \). This then automatically guarantees that the crossing direction is the same at \( h_1 \) and \( h_3 \) [18]. We also have that the stabilizing crossing at \( j\omega_{c2} \) must satisfy \( \omega_{c2} < \omega_{c1} \) [23]. In connection with this, as per [23], we know that \( j\omega_{c1} \) and \( j\omega_{c2} \) crossings are guaranteed to be respectively destabilizing and stabilizing by the nature of the problem. For this reason, one does not need to explicitly consider the crossing directions in the sequel.
CHAPTER 5. DESIGN OF TWO STABLE INTERVALS: THE CASE WITH AN UNSTABLE PLANT

B. Structure of $C(s)$

In general, $C(s)$ has the form

$$C(s) = \sum_{v=0}^{V} \beta_v s^v,$$

with $\alpha_u = 1$, $V \leq U$, $\alpha_u \in \mathbb{R}$, $\beta_v \in \mathbb{R}$. When designing $C(s)$ for $G_1(s)$, we have $V = 2$ and $U = 3$, and for $G_2(s)$, the choice of $V = 2$ and $U = 2$ suffices to reveal feasible solutions for $C(s)$.

C. Number of the crossings, $M$

The crossing frequencies $\omega_{c1}$ and $\omega_{c2}$ must satisfy (2.1) as these are the crossings corresponding to the critical delays $h_\ell$. Since the systems can have more than two crossings, we have two options with respect to $M$: prevent and allow (as stated in Chapter 3.1). For example, with the option prevent, we guarantee that the closed loop system has a crossing if and only if $W(\omega^2) = 0$ holds [23]. In the case of $G_2(s)$, $W(\omega^2)$ is a fourth order polynomial in $\omega^2$. That is, the system will have a maximum of four crossings. Since $\omega_{c1}$ and $\omega_{c2}$ are enforced to satisfy (2.1), the polynomial $W$ can be factored $W(\omega^2) = \mathcal{F}_1(\omega^2)\mathcal{F}_2(\omega^2)$ with $\omega_{c1}$ and $\omega_{c2}$ satisfying $\mathcal{F}_1$. Then, the discriminant of the quadratic factor $\mathcal{F}_2$ should be negative to guarantee the remaining two candidate crossings to be complex hence infeasible.

D. Special points of $\tau$

There remain two additional items to be considered: (a) delay-free ($\tau = 0$) system stability, and (b) the minimum positive of the critical delay value for each crossing $\omega_\ell$ is indeed $h_\ell$. Both of these conditions are trivial to establish, first one requiring a Routh’s array implementation with first column elements $r_n, \ldots, r_0$ all being the same sign, and the second one requiring that $h_\ell - 2\pi/\omega_\ell < 0$, $\ell = 1, 2$. This last inequality is due to the fact that delay values for a crossing are periodically spaced by $2\pi/\omega$.

E. fmincon optimization

Finally, we propose an objective function $f = \sum_{u=0}^{U} \alpha_u^2 + \sum_{v=0}^{V} \beta_v^2$, and utilize MATLAB fmincon nonlinear optimization toolbox [30] to minimize $F$ under the above laid out constraints, and opting for either prevent or allow. Notice that choice of $f$ is ad-hoc and is introduced so that the nonlinear solver can be utilized. This way the constraint equations can be enforced while designing the controller coefficients $\alpha_u, \beta_v$. 

35
5.3 Examples

5.3.1 Example A: 1st order open-loop plant

Consider

\[ G(s) = \frac{1}{s - 1}. \]

In this case, the upper bound of the delay margin, following [35], is calculated as \( \tau^* = 2 \). That is, for \( \tau' > \tau^* \) the closed-loop system is guaranteed to be unstable if we were to assign a single delay interval \([0, \tau^*]\) to it.

The question is could we still stabilize this closed-loop system for some delay \( \tau' \) larger than \( \tau^* = 2 \)? Following Section 5.2 with option allow, we propose to design a controller \( C(s) \) to stabilize the closed-loop for two pre-determined delay intervals \( \tau \in [0, 0.3) \cup (2.02, 2.06) \). The optimization routine returns \( \alpha_m, \beta_v \) such that the proposed controller becomes

\[ C(s) = \frac{1387.2s^2 + 2760.7s + 2130.6}{s^3 + 534.5926s^2 + 474.3106s + 2129.1}. \]

Time-domain integration of the solution in MATLAB/SIMULINK reveals that indeed the closed-loop system is stable as intended for delay values in the pre-determined stability intervals, see Fig. 5.1.

Note in the above study that the second stability interval is quite narrow partly because we did not repeatedly preform the design to expand it, but also because convergence was not always successful when a larger interval was imposed. Nevertheless, given the above feasible solution, one can implement a continuation approach to tune controller parameters to expand this interval, if needed. Since the primary aim here was to demonstrate the possibility that the inherent limitation on \( \tau^* \) can actually be removed by considering two stability intervals, we did not pursue additional steps to expand the second stability interval.

5.3.2 Example B: 2nd order open-loop plant

Consider

\[ G(s) = \frac{1}{s^2 - 0.2s + 100}. \]

In this case, upper bound of the delay margin, following [35], is calculated as \( \tau^* = 0.626287 \). For \( \tau' > \tau^* \) the closed-loop system is guaranteed to be unstable if we were to assign a single delay interval \([0, \tau^*]\) to it.

We pose the same question for this design problem: could we still stabilize this closed-loop system for some delay \( \tau' \) larger than \( \tau^* = 0.626287 \)? Selecting the desired stability intervals
as \( \tau \in [0, 0.4) \cup (0.7, 0.9) \), the optimization routine with option prevent returns \( \alpha_m, \beta_v \) such that the proposed controller becomes

\[
C(s) = -\frac{216.1090s^2 + 62.3303s + 21701}{s^2 + 15.6025s + 245.5855}.
\]

Stability of the system for the intended stable delay intervals can once again be validated in MATLAB/SIMULINK, see Fig. 5.2.

5.4 Conclusion

The inherent limitation posed on the upper bound of the delay margin by unstable open-loop poles of the plant is a well known issue. However, this limitation can be in part alleviated by considering instead multiple disjoint stability intervals along the delay axis, which may allow stabilization for delays larger than the theoretical upper-bound of the delay margin. This promise is demonstrated here via our control design technique using nonlinear optimization tools stated in the previous chapters. Two examples, including the plant with a single unstable pole and another with a pair of complex unstable poles, are studied to demonstrate this opportunity.

While this technique is not yet completely generalized, it unveils new features and points out opportunities regarding our ability to stabilize LTI SISO systems affected by a loop delay. Future research should be focused on furthering the know-how on the design of controllers that can achieve stabilization in two stability intervals in the delay parameter space with satisfactory width, and investigating why and how a single stable delay interval poses limitations on how large we can make this interval. Moreover, there is a chance for us to design more than two stability intervals to achieve stability for delays even farther along the delay axis, which can be developed building on the results of this thesis.
Figure 5.1: Example A: Unit step response of the closed loop system simulations at $\tau = 0, 0.29, 0.3, 0.31, 2.01, 2.02, 2.03, 2.05, 2.06, 2.07$ with the controller $C(s)$ designed for $G_1(s)$ and the pre-determined stability decomposition $\tau \in [0, 0.3) \cup (2.02, 2.06)$
Figure 5.2: Example B: Unit step response of the closed loop system simulations at \( \tau = 0, 0.39, 0.40, 0.41, 0.69, 0.7, 0.71, 0.89, 0.9, 0.91 \) with the controller \( C(s) \) designed for \( G_2(s) \) and the pre-determined stability decomposition \( \tau \in [0, 0.4) \cup (0.7, 0.9) \)
Chapter 6

Conclusion

An eigenvalue-based optimization scheme is developed in this thesis to be able to design controllers for LTI SISO systems subject to delay, with the aim to let the closed-loop system match certain given structure of stability intervals along the delay axis. This novel method is motivated by the following challenges:

- **Difficulty to design multiple parameters**: The presence of delay in system characteristic equations will cause the control design to be an infinite-dimensional nonlinear eigenvalue problem. Even though some literature focus on tuning one or two parameters for SISO systems, it is a mathematical challenge to design multiple parameters simultaneously. Some solutions to this problem using Lyapunov-based LMI optimization tools is possible. However, limitations include the difficulties to design structured controller and providing only sufficient but not necessary conditions of stability.

- **Challenge of multiple stability intervals design**: In many studies, it has been observed and analyzed that the system can be stabilized in more than one delay interval. While most of the existing work remains to solve the forward problem in which the characteristic equation is given, little efforts are made on the control design with “stability certificates” to recover stability in multiple delay intervals. Since more than one crossing must be considered with respect to multiple stable delay intervals, this problem is much more complicated than the delay margin design mainly concerned with only one crossing.

To build up a strategy to solve the control design problem, we first start with a LTI SISO system with a single loop delay. Next, some well-known knowledge of such systems in frequency
CHAPTER 6. CONCLUSION

domain are summarized from the literature, to lay out the mathematical expressions of three crucial factors – imaginary crossing, critical delay value, and root tendency. Based on this, we can understand the whole picture of stability intervals along the delay axis via listing the constraints in frequency domain. As these constraints match well with the structure of nonlinear programming, we take advantage of the MATLAB toolbox fmincon, and formulate an optimization problem to solve the unknown controller parameters. For different types of problems, this optimization-based approach is discussed in separate chapters with a focus respectively on:

- **Delay margin design**: Control design for certain delay margin has been studied for long, however, its design with respect to multiple controller gains remains a nontrivial task. Using our approach, this problem is solved effectively and with no conservatism. A PID controller design example is demonstrated with step by step analysis.

- **Design of two stable intervals – The case with a stable plant**: Starting with control design for a stable second order plant, we propose Steps 1-4 for two disjoint stable delay intervals. Considering the limitation of fmincon tool, a rigorous rule for initial condition is developed by creating a special case. A computational and an experimental case are studied on a second order plant. Moreover, this design approach can be adapted to a higher order system using cascade controller design by iteration, shown by another example.

- **Design of two stable intervals – The case with an unstable plant**: Unstable plant will cause a limitation on the upper bound of the design margin, and this upper bound in some cases is determined by the locations of the unstable poles of the plant, with an explicit expression. With multiple stable delay interval design, this limitation is overcome via placing the second stability region larger than its theoretical delay-margin upper bound. Examples of control design for both first and second order unstable plants are presented.

To conclude, the challenging reverse design problem is successfully solved and presented, and demonstrated on several case studies, offering a strong foundation toward solving more complex problems involving multiple stability intervals. In future study, we will focus on finding a good initial condition for unstable-plant control design, since it is still not fully generalized. In addition, control design for higher order systems is of concern, and we will further study the cascaded controller design by iteration for general high-order plants.
Bibliography


BIBLIOGRAPHY


Appendix A

Initial Condition

To find out the best way to select the initial condition, we first reduce the number of constraints and attempt to search for an initial condition point $P$ satisfying some special cases. These cases are stated as follows:

I. At $\tau = 0$, the delay-free system $F(s, 1)$ is stable. In this case, the set of solutions to $\chi$ is denoted by $Q_0$.

II. Denoted by $Q_1$ is the set of possible solutions $\chi$ rendering a crossing $\omega_{c1}$ and a critical delay at $h_1$.

III. Denoted by $Q_2$ is the set of possible solutions $\chi$ rendering a crossing $\omega_{c2}$ and a critical delay at $h_2$.

Note that the three conditions I-III above do not guarantee the stability decomposition in Fig. 4.1, but only guarantee a crossing at $h_1$ and $h_2$, and the stability of the delay-free system. In Fig. 4.1, the intersection area of sets $Q_0$, $Q_1$ and $Q_2$ is schematically pictured by $\overline{ABC}$. Since the constraints corresponding to $Q_0$, $Q_1$ and $Q_2$ are necessary conditions of $\mathcal{C}_{1-8}$, the feasible solution set, if it exists, satisfying $\mathcal{C}_{1-8}$, denoted by $Q_f$, must lie inside the region $\overline{ABC}$ region. Thus, a good initial condition would be to pick an initial point $P$ in side of $\overline{ABC}$.

When there are multiple design parameters, such as the case here, it may not be easy to select a point inside the parametric region $\overline{ABC}$. Nevertheless, it is quite practical to select a point nearby the boundaries of $\overline{ABC}$. Fortunately, there exist several ways to calculate the parametric settings, e.g., at points $P_0$, $P_1$, $P_2$. The points $P_0$ satisfying Condition I is trivial to obtain from the stability analysis of the delay-free system. The points $P_1$ and $P_2$ follow the same type of mathematical calculations, and guarantee a single crossing either at $h_1$ and $h_2$. Conditions on system parameters guaranteeing a crossing can easily be obtained using frequency sweeping methods, see,
e.g., [7]. Simply, set $s = j\omega$, $\tau = h_1$ or $\tau = h_2$ in (4.4), and as $\omega$ increases from zero to an upper bound, one can solve the real and imaginary part of the system characteristic equation from which an implicit formula can be obtained on system parameters satisfying the crossing $s = j\omega$ at $\tau$.

Application of the above rationale for the problem at hand leads to the following

\begin{equation}
F^\Re(\omega, \alpha_k, \beta_k, \tau) = \omega^4 + (-2\alpha_1 \omega_n \zeta - \cos(\tau \omega) \beta_2 - \omega_n^2 - \alpha_0)\omega^2 \\
+ \sin(\tau \omega)\beta_1 \omega + \alpha_0 \omega_n^2 + \cos(\tau \omega)\beta_0, \tag{A.1}
\end{equation}

\begin{equation}
F^\Im(\omega, \alpha_k, \beta_k, \tau) = (-2\omega_n \zeta - \alpha_1)\omega^3 + \sin(\tau \omega)\beta_2 \omega^2 \\
+ (2\alpha_0 \omega_n \zeta + \alpha_1 \omega_n^3 + \cos(\tau \omega)\beta_1)\omega - \sin(\tau \omega)\beta_0. \tag{A.2}
\end{equation}

Since it is not easy to solve (A.1)-(A.2) simultaneously, we propose to pick $\alpha_0 = \omega c_{12}$ and $\alpha_1 = 0$ to simplify (A.1)-(A.2) as

\begin{equation}
F^\Re(\omega, \beta_0, \beta_1, \beta_2, \tau) = \omega^4 + (\cos(\tau \omega) \beta_2 - \omega_n^2 - \omega c_{12}^2)\omega^2 + \sin(\tau \omega)\beta_1 \omega + \omega c_{12}^2 \omega_n^2 + \cos(\tau \omega)\beta_0, \tag{A.3}
\end{equation}

\begin{equation}
F^\Im(\omega, \beta_0, \beta_1, \beta_2, \tau) = -2\omega_n \zeta \omega^3 + \sin(\tau \omega)\beta_2 \omega^2 + (2\omega c_{12}^2 \omega_n \zeta + \cos(\tau \omega)\beta_1)\omega - \sin(\tau \omega)\beta_0. \tag{A.4}
\end{equation}
APPENDIX A. INITIAL CONDITION

For the set $Q_1$ and $Q_2$, at $\tau = h_1$, $\omega = \omega_{c1}$ and $\tau = h_2$, $\omega = \omega_{c2}$ respectively, we have the following four equations from (A.3) and (A.4).

\[
\begin{align*}
F_{\Re}(\omega_{c1}, \beta_0, \beta_1, \beta_2, h_1) &= 0, \\
F_{\Im}(\omega_{c1}, \beta_0, \beta_1, \beta_2, h_1) &= 0, \\
F_{\Re}(\omega_{c2}, \beta_0, \beta_1, \beta_2, h_2) &= 0, \\
F_{\Im}(\omega_{c2}, \beta_0, \beta_1, \beta_2, h_2) &= 0.
\end{align*}
\tag{A.5}
\]

Since $\omega_{c1}$, $h_1$ and $h_2$ are known in Step 1, a particular solution of $\omega_{c2}$, $\beta_0$, $\beta_1$, $\beta_2$ can be extracted from the above four equations, that is, $\omega_{c2} \to \omega_{c1}$, $\beta_0 = \beta_2\omega_{c1}^2$, $\beta_1 \to 0$. In this case, we can pick $\beta_2 \to 0^+$, then $\beta_0 \to 0^+$. With the choice of small $\beta_k = \epsilon$, the coefficient of the exponential term $e^{-st}$ will be vanishing, hence this point is at the boundary of the set $Q_1$, denoted by $P_1$, and also at the boundary of the set $Q_2$, denoted by $P_2$.

Considering the set $Q_0$, with the same rules as above in picking $\alpha_k$ and $\beta_k$ values, when $\alpha_0 = \omega_{c1}$, $\alpha_1 = 0$, $\beta_0, \beta_1, \beta_2 \to 0^+$ and $\tau = 0$, the characteristic equation in (4.4) approaches to

\[
F(s, 1) \cong (s^2 + \omega_{c1}^2)(s^2 + 2\zeta\omega_n s + \omega_n^2).
\tag{A.6}
\]

which comprises of an oscillating term due to $s^2 + \omega_{c1}^2$, and a stable term due to $s^2 + 2\zeta\omega_n s + \omega_n^2$. Hence the system is marginally stable. This solution point is at the boundary of the the set $Q_0$, denoted by $P_0$.

In Fig. A.1, since our parametric space is actually 6-dimensional, the three points $P_0$, $P_1$ and $P_2$ are indeed the same point, located at the boundary of the intersection area bounded by $ABC$. This point is therefore a good initial condition point to be used in the optimization scheme.

Furthermore, since $C_8$ gives a constraint for $\omega_{c2}$, it therefore makes sense $\omega_{c2}$ is slightly less than $\omega_{c1}$ for initial condition. In some sense, this lets the optimization go only in one direction (reduce $\omega_{c2}$ further) as iterations take place along the $\omega_{c2}$ direction, consistent with Remark 3.

For parameters $\beta_0$, $\beta_1$ and $\beta_2$ in the numerator, they are assigned with a small positive value $\epsilon$ as explained above. This corresponds to a weak controller, which initially guarantees that $\alpha_0, \alpha_1$ assignment above initially achieves its goal of satisfying $Q_0$ as well as $Q_1$ and $Q_2$. 

48