

AN AUTOMATED VIRTUAL TOOL TO COMPUTE THE ENTIRE
SET OF PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLERS FOR A
CONTINUOUS LINEAR TIME INVARIANT SYSTEM

A Thesis

by

BHARAT NARASIMHAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

December 2007

Major Subject: Electrical Engineering

AN AUTOMATED VIRTUAL TOOL TO COMPUTE THE ENTIRE
SET OF PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLERS FOR A
CONTINUOUS LINEAR TIME INVARIANT SYSTEM

A Thesis

by

BHARAT NARASIMHAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Shankar P. Bhattacharyya
Committee Members,	Aniruddha Datta
	Jim Ji
	Darbha Swaroop
Head of Department,	Costas N. Georghiades

December 2007

Major Subject: Electrical Engineering

ABSTRACT

An Automated Virtual Tool to Compute the Entire
Set of Proportional Integral Derivative Controllers for a Continuous Linear Time
Invariant System. (December 2007)

Bharat Narasimhan, B.E (Hons), Birla Institute of Technology and Science, Pilani -
Dubai Campus

Chair of Advisory Committee: Dr. Shankar P. Bhattacharyya

This thesis presents the very practical and novel approach of using the Graphical User Interface (GUI) to compute the entire set of Proportional Integral Derivative (PID) controllers given the transfer function or the frequency response of the system under consideration.

Though there is a wide spread usage of PID controllers in the industry, until recently no formal algorithm existed on determining a set of PID values that will stabilize the given system. The industry still relies on algorithms like the Ziegler-Nicholas or ad-hoc approaches in determining the value of PID controllers. Also when it comes to model free approaches, the use of Fuzzy logic and Neural network do not guarantee stability of the system.

For a continuous Linear Time Invariant system Bhattacharyya and others have developed an algorithm that determines the entire set of PID controllers given the transfer function or just the frequency response of the system. The GUI has been developed based on this theory. The GUI also evaluates the user input performance specifications and generates a subset of stable controllers given the performance criteria for the system.

This thesis presents an approach of automating the computation of entire set

of stabilizing Proportional Integral Derivative (PID) controllers given the system transfer function or the frequency response data of the system. The Graphical User Interface (GUI) developed bridges the gap between the developed theory and the industry.

To K.R Vaidyanathan - my ever smiling grandad

ACKNOWLEDGMENTS

I would like to thank Dr. Shankar Bhattacharyya, my committee chair, for his guidance and support throughout this research work. I sincerely thank him for giving me this opportunity to work under him. It has been a great experience to be working on a subject of such importance and to learn from one of the leading researchers in this area. He is responsible for helping me gain a better understanding of the subject and also helped me be more professional with my work. His timely input and creative ideas have always been an essential part of this research work.

I would also like to thank the other committee members, Dr. Aniruddha Datta, Dr. Jim Ji and Dr. Darbha Swaroop, for all their cooperation and help toward a timely completion of this research work.

I also would like to thank Dr. Bryan Rasmussen for taking his time in being part of my thesis defense committee and for contributing constructive criticism of my work.

I also wish to thank Sandipan Mitra and Indu Ramamurthi for always being around when I needed help with my work and Tammy Carda for all the invaluable help with the administrative aspect of this research work. Her very efficient manner of handling the most complex of administrative issues has resulted in a timely completion of my thesis.

Finally, I would like to thank my parents and brother for all the motivation, support and patience they have showered upon me over my entire life. I would also like to thank my other family members for their constant support.

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION	1
	A. Objective	4
	B. Organization of Thesis	4
II	BACKGROUND AND TERMINOLOGIES	6
	A. Background	6
	B. Mikhailov's Criterion	7
	C. Hermite-Biehler Theorem	8
	D. Generalization of the Hermite-Biehler Theorem	9
III	STABILIZING A LINEAR TIME INVARIANT SYSTEM USING A PROPORTIONAL CONTROLLER	17
	A. Theory and Algorithm	17
	B. The GUI for Calculating the Entire Set of Proportional Controllers for a LTI System	23
	1. GUI Based Calculation of Proportional Gain	24
	2. GUI Based Performance Evaluation of Proportional Controllers	25
	3. Illustrative Example	26
IV	STABILIZING A LINEAR TIME INVARIANT SYSTEM USING A PI CONTROLLER	32
	A. Theory and Algorithm	32
	B. GUI for Calculating the Entire Set of Proportional In- tegral Controllers for a LTI System	38
	1. GUI Based Calculation of Proportional Integral Values	38
	2. GUI Based Performance Evaluation of Proportional Integral Controllers	40
	3. Illustrative Example	41
V	STABILIZING A LINEAR TIME INVARIANT SYSTEM USING A PID CONTROLLER	49
	A. Theory and Algorithm	49

CHAPTER	Page
B. GUI for Calculating the Entire Set of Proportional Integral Derivative Controllers for a LTI System	55
1. GUI Based Calculation of Proportional Integral Derivative Values	56
2. GUI Based Performance Evaluation of Proportional Integral Derivative Controllers	58
3. Illustrative Example	59
VI A MODEL FREE APPROACH IN STABILIZING A LTI SYSTEM USING A PID CONTROLLER	66
A. Theory and Algorithm	66
B. GUI for Model Free Approach for Calculating the Entire Set of Proportional Integral Derivative Controllers for a LTI System	74
1. GUI Based Calculation of Proportional Integral Derivative Values	75
2. Illustrative Example	78
VII DISCUSSION AND CONCLUSION	83
A. Case Study	83
B. Future Work	84
REFERENCES	86
VITA	90

LIST OF TABLES

TABLE	Page
I Routh Table	3

LIST OF FIGURES

FIGURE	Page
1	Normalized Mikhailov's plot for a Hurwitz stable polynomial 8
2	Interlacing property for a Hurwitz's stable polynomial 10
3	Interlacing property for a non-Hurwitz stable polynomial 12
4	Normalized Mikhailov's plot for a non-Hurwitz stable polynomial . . . 13
5	GUI to determine the entire set of Proportional Controllers for a given transfer function 26
6	GUI with the entire set of k_p displayed for the given transfer function 29
7	GUI with all the performance displayed for the given transfer function 30
8	GUI with a sub-set of stabilizing values of k_p satisfying the condition of PM $> 45dB$ and Overshoot $< 35\%$ 31
9	The stabilizing set of (k_p, k_i) values for example (IV.1) 39
10	GUI for determining the entire set of Proportional Integral controllers for a given transfer function 42
11	GUI with the entire set of k_i for a fixed value of k_p 45
12	GUI with the entire set of k_p displayed for the given transfer function 46
13	GUI with all the performance displayed for the given transfer function 47
14	GUI with a subset of stabilizing values in $\{k_p, k_i\}$ space satisfying the condition of PM $> 7dB$ and Rise-time $< 2sec$ 48
15	Stabilizing values in $\{k_i, k_d\}$ space for a fixed value of k_p 56
16	The stabilizing set of $\{k_p, k_i, k_d\}$ values for example (V.1) 57

FIGURE	Page
17	GUI for determining the entire set of Proportional Integral Derivative controllers for a given transfer function 60
18	GUI with the entire set of $\{k_i, k_d\}$ for a fixed value of k_p 62
19	GUI with the entire set of $\{k_p, k_i, k_d\}$ displayed for the given transfer function 63
20	GUI with the performance displayed for a selected value of PID controller 64
21	GUI with a subset of stabilizing values in $\{k_i, k_d\}$ space satisfying the condition of GM $> 1.2dB$ and Overshoot $< 30\%$ 65
22	Bode plot of the plant $p(s)$ 72
23	Plot of $g(\omega)$ against varying values of frequency ω 74
24	Stabilizing values in $\{k_i, k_d\}$ space for a fixed value of k_p 75
25	The stabilizing set of $\{k_p, k_i, k_d\}$ values for VI.1 76
26	GUI for determining the entire set of Proportional Integral Derivative controllers given the frequency response of the system 78
27	Plot of $g(\omega)$ against varying frequency 79
28	The GUI with the range of feasible k_p values displayed for the given bode plot 80
29	The GUI with the set in $\{k_i, k_d\}$ space for a value of $k_p = -5$ 81
30	Complete set of PID values for the given frequency response 82

CHAPTER I

INTRODUCTION

In the different systems existing around us like trains, automobiles, airplanes controllers are used. The basic function of the controller is to track the set point and negate external disturbance in the plant. There are two main approaches to Control Theory, Classical Control and Modern Control. The Classical Control theory is a frequency domain approach and until recently relies on Mathematical Models of the system. It is widely used for Linear Time Invariant (LTI) systems. On the other hand the Modern Control theory involves time domain analysis and based on Linear Algebra.

Proportional Integral Derivative (PID) controllers belong to the Classical Control Theory approach. They have been widely used in almost all industries ranging from simple systems like Temperature control to complex Distillation plants. The PID controller acts on the error value that is determined by the difference in the set point and the output of the process. It is implemented as follows.

$$C(s) = k_p + \frac{k_i}{s} + k_d s \quad (1.1)$$

where k_p, k_i, k_d are proportional, integral and derivative gain respectively and $C(s)$ is the transfer function of the controller. The derivative term may sometimes be implemented as $\frac{k_d s}{(1+Ts)}$, where T is a very small constant. For low frequency signals the derivative term contributes low outputs, but as the frequency of the signal increases the derivative contributes to higher outputs, thus giving erroneous outputs in case of a noisy signal. To prevent this issue observed due to a pure differentiator when the

The journal model is *IEEE Transactions on Automatic Control*.

error signal has large noise a modified derivative term may be used.

The PID controller as seen above has three modes, the Proportional, Integral and the Derivative mode. In the Proportional mode a constant acts on the present error. In the Integral mode, the integral of the errors are calculated. It is equivalent of looking at all the past errors and taking a corrective action accordingly. While in the Derivative mode action is taken on rate of change of errors. It is equivalent of taking a corrective action by looking at the trend in change of errors. The final output of the PID controller is the sum of the three modes.

The major issue with PID controllers is until recently no formal algorithm existed to calculate the entire set of PID controllers to stabilize the given system. The existing algorithms like Ziegler-Nichols (ZN) [1], Internal Model Controller (IMC) [2] etc. have their own limitations. Algorithms like ZN method generates only one stabilizing value for the given system. Though a complete set of PID controllers can be determined by using the classical Routh-Hurwitz criterion, we will see from the following discussion that such an approach involves solving non-linear inequalities and is computationally very intense.

The Ziegler-Nichols criterion had been developed by extensive simulations of stable, simple plants. In the ZN approach either the step response or the frequency response of the system can be used to generate a set of PID values stabilizing the given system. Only one set of PID value is obtained using this approach. Moreover this criterion is applicable only to open loop stable plants. Though the single PID value generated by the ZN criterion guarantees stability it does not account for the user's performance specification.

The Routh Criterion on the other hand can theoretically generate the entire set of PID values, but this involves solving non-linear inequalities and is computationally very intensive as seen from the following example where the transfer function $P(s)$ is

Table I. Routh Table

3	1	$k_d - 2$	k_i
2	3	$k_p + 1$	
1	$\frac{3k_d - k_p - 3}{3}$	k_i	
0	$\frac{-k_p^2 - 4k_p + 3k_d k_p - 9k_i + 3k_d}{3k_d - k_p - 3}$		

to be stabilized using a PID controller $C(s)$.

$$P(s) = \frac{1}{s^3 + 3s^2 - 2s + 1}$$

$$C(s) = k_p + \frac{k_i}{s} + k_d s$$

The resulting characteristic equation is $s^4 + 3s^3 + (k_d - 2)s^2 + (k_p + 1)s + k_i$. According to the Routh criterion the entire first column should be greater than zero for the closed loop to be stable. From table I non linear inequalities are generated as under

$$3k_d - k_p - 3 > 0$$

$$\frac{-k_p^2 - 4k_p + 3k_d k_p - 9k_i + 3k_d}{3k_d - k_p - 3} > 0$$

As observed for a relatively low order plant the corresponding inequalities are computationally intense and hence the Routh criterion is not a practical solution to generate the entire stabilizing set in k_p, k_i, k_d space.

Recent research has enabled the prediction of the entire stabilizing set for PID controllers by formulating linear inequalities [3],[4]. The motivation for the current work is to simplify this algorithm by having interactive Graphical User Interface (GUI) and bring this powerful algorithm closer to the industry which demands simple but yet powerful tools to solve complex stabilizing problems.

A. Objective

The objective of this study was to provide the industry and control systems engineer in general a simple but powerful Graphical User Interface to

- To determine the entire set of stabilizing PID controllers given the mathematical model of a continuous LTI system.
- To determine the entire set of stabilizing PID controllers given the frequency response of a continuous LTI system.
- To determine the subset of the stable set of controllers depending on the performance specifications given by the user.

The above were achieved using the algorithms proposed in [3],[4]. The GUI was implemented on MATLAB[®] due to the software being widely used in both the industry and the academia alike. The GUI was then used in an application for designing PID controllers of High Speed Flywheel for Traction Applications [5].

B. Organization of Thesis

In the following part of the thesis many topics will be covered. In Chapter II a brief overview of concepts like Mikhailov's plot, Hermite-Biehler theorem will be covered. Also the concept of signature will be introduced. These concepts and terminologies will be frequently referred to in the rest of the discussion. Chapters III, IV and V introduces the algorithm to determine the entire set of Proportional (P), Proportional Integral (PI) and Proportional Integral Derivative (PID) controllers for a given transfer function. Each algorithm is followed by an insight into the Graphical User Interface (GUI) developed in MATLAB[®] followed by an example in each section.

Chapter VI introduces the concept of a model free approach for determining the entire set of PID controllers [4]. Again the discussion follows a similar pattern as in the previous chapters. In Chapter VII a case study on use of GUI in designing PID controllers for High Speed flywheels for traction application is discussed [5]. In the concluding Chapter VII the research work is summarized and possible future work is discussed.

CHAPTER II

BACKGROUND AND TERMINOLOGIES

In this chapter a brief background on control systems, certain terminologies and concepts like the Mikahailov's Criterion, Hermite-Biehler Theorem etc are introduced. These concepts will be referred to frequently in the later chapters and will help in understanding the algorithm developed by Bhattacharyya and other in [3], [4] and the GUI developed in a complete manner.

A. Background

In the past four decades elegant control system techniques like H_2 , H_∞ [6], L_1 Optimal [7] control have been developed. YJBK parametrization [8] was one of the major breakthrough in modern control theory. The major drawback is the order of the stabilizing controller obtained by YJBK parametrization is always quite high and sometimes comparable to the order of the plant under consideration. Also recently it has been pointed out in [9] that the controllers obtained through the above mentioned techniques are susceptible to even a small uncertainty resulting in the entire system unstable.

The industry on the other hand is wary of such high order controllers and these theoretically optimal and high order controllers are not widely used by the industry. Thus there has been a recent shift in focus in the academia towards fixed low order controllers. Hara, Shiokata and Iwasaki in [10] developed a Fixed order controller design via generalized KYP lemma, while in [11] Henrion, Hansson and Wallin put forth reduced Linear Matrix Inequalities for fixed order controllers. Gryazina and Polyak in [12] have introduced a graphical technique using D-Decomposition [13], [14] to obtain a set of stabilizing fixed order controllers.

More than 90% of the controllers used in the industry are Proportional Integral Derivative controllers.[15], [16], [17]. PID controllers are simple in structure and can handle both steady-state and transient response [18]. Entire set of stabilizing PID controllers can be obtained using the algorithm introduced in [3], [4]. The focus of this thesis is to develop a Graphical User Interface for the theory developed in [3], [4], making it more practical and useful.

B. Mikhailov's Criterion

A Hurwitz stable polynomial is a polynomial that has all the zeros in the left half of the complex plane. For a Hurwitz stable polynomial $p(s)$ the Mikhailov's criterion states that as the frequency (ω) increases from 0 to ∞ the plot of $p(j\omega)$ turns in an anti-clockwise direction and goes through n quadrants in the complex plane where n is the degree of the Hurwitz stable polynomial. A detailed proof of the Mikhailov's criterion can be found in [19].

Consider the following example on Mikhailov's plot for the Hurwitz stable polynomial:

Example II.1

$$p(s) = s^5 + 7s^4 + 24s^3 + 48s^2 + 55s + 25 \quad (2.1)$$

As seen in figure (1), as the $p(s)$ is a Hurwitz stable polynomial with degree 5 the Mikhailov's plot essentially $p(j\omega)$ as ω goes from 0 to ∞ turns in an anti-clockwise direction and goes through 5 quadrants. In figure (1) it is to be noted that $p(j\omega)$ has been normalized by $(1 + \omega^2)^{\frac{n}{2}}$, where n is the degree of the polynomial. Also it is observed that the net change in phase as ω progress from 0 to ∞ is $\frac{5\pi}{2}$. In general for a Hurwitz stable polynomial, this net change is $\frac{n\pi}{2}$, where n is the degree of the polynomial. For a detailed proof refer [19].

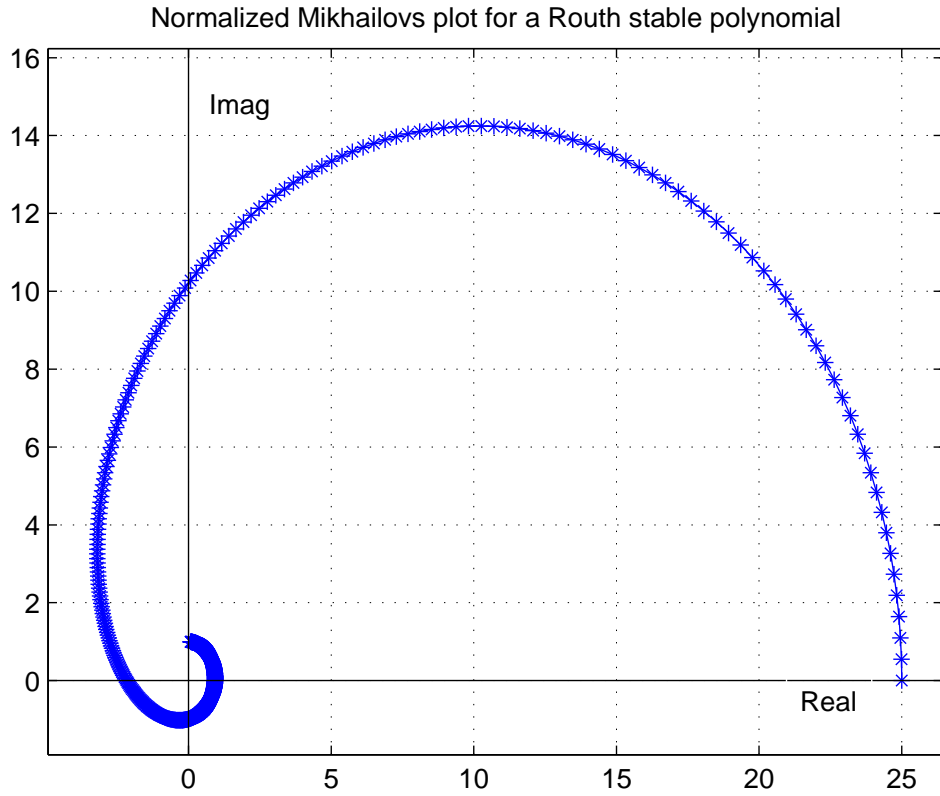


Fig. 1. Normalized Mikhailov's plot for a Hurwitz stable polynomial

C. Hermite-Biehler Theorem

The Hermite-Biehler theorem gives a necessary and sufficient condition for a given polynomial to be Hurwitz stable. Consider a real polynomial $p(s)$ with degree n . The hodograph $p(j\omega)$ of the polynomial $p(s)$ can be written as follows,

$$p(j\omega) = p_e(\omega) + j\omega p_o(\omega) \quad (2.2)$$

Then the polynomial $p(s)$ is Hurwitz stable if and only if all the real, distinct, non-negative zeros with odd multiplicity of $p_e(\omega)$ and $p_o(\omega)$ satisfy the following interlacing

criterion

$$0 < \omega_{e1} < \omega_{o1} < \omega_{e2} < \omega_{o2} < \omega_{e3} \cdots \quad (2.3)$$

A detailed proof of the Hermite-Biehler theorem can be found in [20]. Consider the following example,

Example II.2 Referring to the Hurwitz stable polynomial in equation (2.1)

$$p(s) = s^5 + 7s^4 + 24s^3 + 48s^2 + 55s + 25$$

$$p(s) = p_e(\omega) + jp_o(\omega)$$

where

$$p_e(\omega) = 7\omega^4 - 48\omega^2 + 25$$

$$p_o(\omega) = \omega^5 - 24\omega^3 + 55\omega$$

From the figure (2) it is observed that the polynomial $p_e(\omega)$ and $p_o(\omega)$ have the interlacing property and hence Hurwitz stable polynomial.

D. Generalization of the Hermite-Biehler Theorem

The Hermite-Biehler theorem dealt only with Hurwitz stable polynomials. The Generalization of the Hermite-Biehler theorem is an extension to handle polynomials not necessarily Hurwitz and was introduced by Bhattacharyya and other in [21].

For a given polynomial $p(s)$, let $l(p)$ and $r(p)$ denoted the number of left and right half zeros in the complex plane. Let $\angle\Delta_0^\infty\theta$ denote the total change in phase of $p(j\omega)$ as ω goes from 0 to ∞ . In general it is noted that the roots in the left half of the complex plane ($l(p)$) contribute $\frac{-\pi}{2}$ while the roots on the right half of the complex plane ($r(p)$) contribute $\frac{\pi}{2}$ to the net change in phase ($\angle\Delta_0^\infty\theta$).

$$\angle\Delta_0^\infty\theta = \frac{\pi}{2}[l(p) - r(p)] \quad (2.4)$$

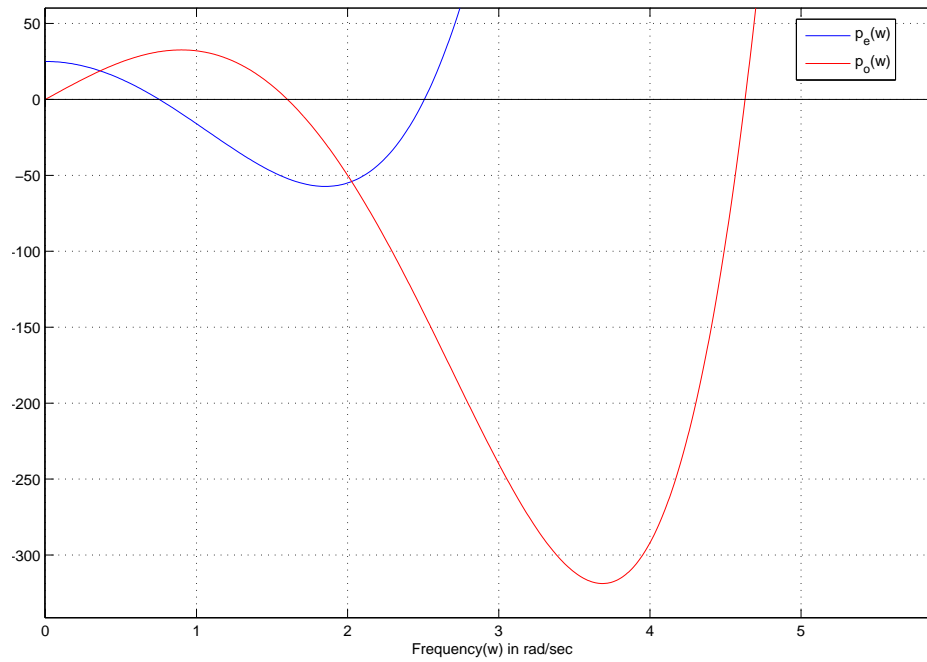


Fig. 2. Interlacing property for a Hurwitz's stable polynomial

Consider a example of a non-Hurwitz polynomial as follows

Example II.3

$$p(s) = s^5 - 7s^4 + 24s^3 + 48s^2 + 55s + 25 \quad (2.5)$$

The roots for the non-Hurwitz polynomial are

$$\begin{aligned} \text{roots} = & 4.3364 + 4.2487i \\ & 4.3364 - 4.2487i \\ & -0.4873 + 0.8568i \\ & -0.4873 - 0.8568i \\ & -0.6982 \end{aligned} \quad (2.6)$$

From equation (2.6) it is observed that there are 2 roots to the right half of the complex plane and 3 to the left half of the complex plane. Also from figure (3) it is observed that the interlacing property is not met and hence it is confirmed that the polynomial in equation (2.5) is non-Hurwitz by the Hermite-Biehler theorem. From the Mikahilov's plot (figure (4)) it is observed that though the encirclement is in the anti-clockwise direction, it passes through only 3 quadrants. Had the polynomial been Hurwitz stable the Mikhailov's plot would pass through 5 quadrants as 5 is the degree of the polynomial. For the given polynomial in equation (2.5), it is observed

$$l(p) = 3$$

$$r(p) = 2$$

From equation (2.4)

$$\Delta_0^\infty \angle \theta = \frac{\pi}{2}(3 - 2) = \frac{\pi}{2}$$

The value of $\angle \Delta_0^\infty \theta = \frac{\pi}{2}$ and is also verified from the Mikhailov's plot in figure (4)

For the Generalization of the Hermite-Biehler theorem some additional terminology is introduced.

Let,

$$\text{sgn}(x) = \begin{cases} +1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \end{cases}$$

Also, for a given polynomial $p(s)$, we define signature $\sigma(p(s))$ as the difference in the number of zeros lying in the left and the right half of the complex plane.

$$\sigma(p(s)) = l(p(s)) - r(p(s)) \tag{2.7}$$

Where $l(p(s))$ and $r(p(s))$ are the number of left and the right half zeros of the

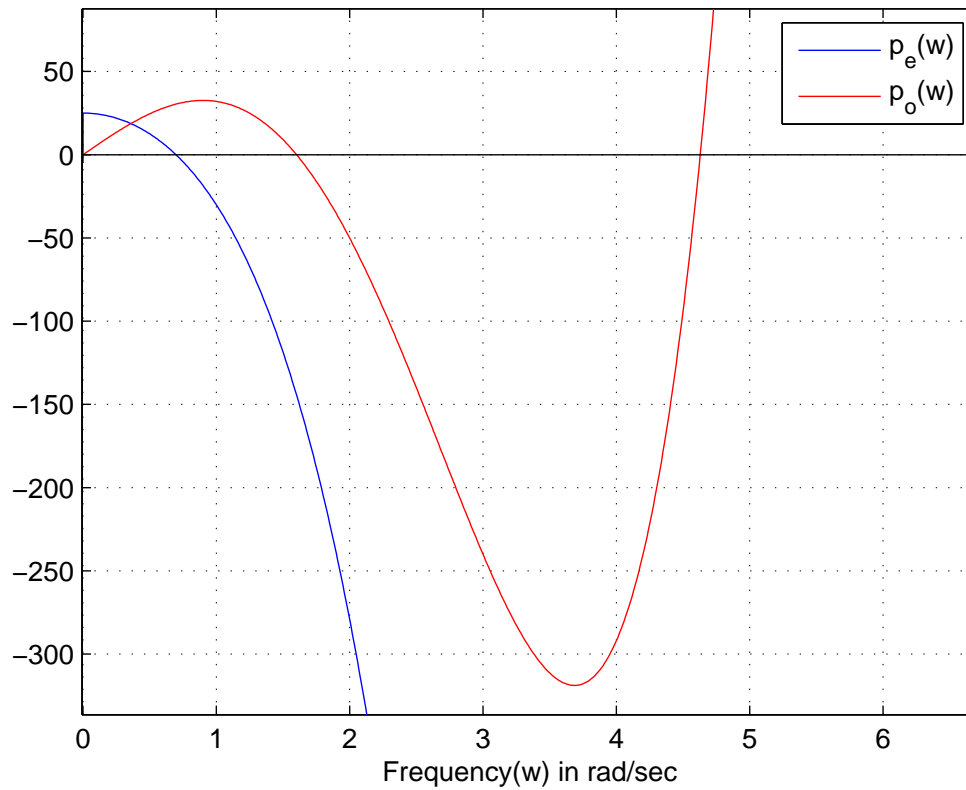


Fig. 3. Interlacing property for a non-Hurwitz stable polynomial

polynomial $p(s)$ in the complex plane respectively. Also the polynomial $p(s)$ can be decomposed into even and odd parts as see in equation (2.8)

$$p(s)|_{s=j\omega} = p_e(\omega) + j\omega p_o(\omega) \quad (2.8)$$

The calculation of $\Delta_0^\infty p(j\omega)$ introduced below was derived by Bhattacharyya and others and a detailed proof is available in [21].

Assuming $p(s)$ has no roots on the imaginary axis that is all the roots are either on the left or the right half of the complex plane.

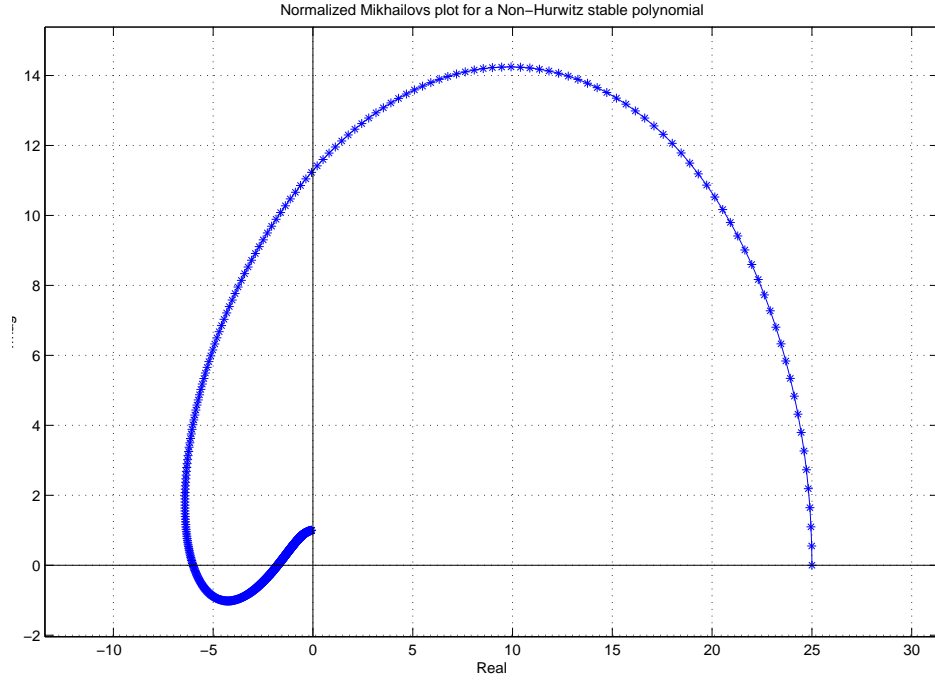


Fig. 4. Normalized Mikhailov's plot for a non-Hurwitz stable polynomial

Let,

$$0 < \omega_1 < \omega_2 < \omega_3 \cdots < \omega_{l-1} < \infty \quad (2.9)$$

be the distinct zeros of odd multiplicity of the $p_o(\omega) = 0$

Then if the degree of $p(s)$ is even,

$$\begin{aligned} \Delta_0^\infty p(j\omega) &= \frac{\pi}{2} \{ \text{sgn}[\dot{p}_o(\omega)|_{\omega=\omega_0}] [\text{sgn}(p_e(\omega_0)) - \text{sgn}(p_e(\omega_1))] \\ &\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_1}] [\text{sgn}(p_e(\omega_1)) - \text{sgn}(p_e(\omega_2))] \\ &\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_2}] [\text{sgn}(p_e(\omega_2)) - \text{sgn}(p_e(\omega_3))] \\ &\quad + \cdots \\ &\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_{l-1}}] [\text{sgn}(p_e(\omega_{l-1})) - \text{sgn}(p_e(\infty))] \} \quad (2.10) \end{aligned}$$

If the degree of the $p(s)$ is odd,

$$\begin{aligned}
\Delta_0^\infty p(j\omega) &= \frac{\pi}{2} \{ \text{sgn}[\dot{p}_o(\omega)|_{\omega=\omega_0}] [\text{sgn}(p_e(\omega_0)) - \text{sgn}(p_e(\omega_1))] \\
&\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_1}] [\text{sgn}(p_e(\omega_1)) - \text{sgn}(p_e(\omega_2))] \\
&\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_2}] [\text{sgn}(p_e(\omega_2)) - \text{sgn}(p_e(\omega_3))] \\
&\quad + \dots \\
&\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_{l-1}}] [\text{sgn}(p_e(\omega_{l-1}))] \} \tag{2.11}
\end{aligned}$$

From equation (2.4), (2.7), (2.10) and (2.11) it can be inferred that for even values of n where n is the degree of $p(s)$

$$\begin{aligned}
\sigma(p) = l(p) - r(p) &= \text{sgn}[\dot{p}_o(\omega)|_{\omega=\omega_0}] [\text{sgn}(p_e(\omega_0)) - \text{sgn}(p_e(\omega_1))] \\
&\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_1}] [\text{sgn}(p_e(\omega_1)) - \text{sgn}(p_e(\omega_2))] \\
&\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_2}] [\text{sgn}(p_e(\omega_2)) - \text{sgn}(p_e(\omega_3))] \\
&\quad + \dots \\
&\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_{l-1}}] \\
&\quad [\text{sgn}(p_e(\omega_{l-1})) - \text{sgn}(p_e(\infty))] \tag{2.12}
\end{aligned}$$

And for odd values of n ,

$$\begin{aligned}
\sigma(p) = l(p) - r(p) &= \text{sgn}[\dot{p}_o(\omega)|_{\omega=\omega_0}] [\text{sgn}(p_e(\omega_0)) - \text{sgn}(p_e(\omega_1))] \\
&\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_1}] [\text{sgn}(p_e(\omega_1)) - \text{sgn}(p_e(\omega_2))] \\
&\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_2}] [\text{sgn}(p_e(\omega_2)) - \text{sgn}(p_e(\omega_3))] \\
&\quad + \dots \\
&\quad + \text{sgn}[\omega \dot{p}_o(\omega)|_{\omega=\omega_{l-1}}] [\text{sgn}(p_e(\omega_{l-1}))] \tag{2.13}
\end{aligned}$$

Consider the following example

Example II.4

$$p(s) = s^7 + 10s^6 + 27s^5 - 14s^4 - 148s^3 - 136s^2 + 60s + 200 \quad (2.14)$$

Then

$$p(j\omega) = p_o(j\omega) + jp_e(j\omega)$$

where,

$$p_o(j\omega) = -10\omega^6 - 14\omega^4 + 136\omega^2 + 200$$

$$p_e(j\omega) = -\omega^7 + 27\omega^5 + 148\omega^3 + 60\omega$$

The real positive roots of $p_e(j\omega)$ are (0, 5.6325).

As the degree of the polynomial $p(s)$ is odd, from equation(2.13)

$$\begin{aligned} l - r &= \operatorname{sgn}[\dot{p}_o(0)][\operatorname{sgn}(p_e(0)) - \operatorname{sgn}(p_e(5.6325))] \\ &\quad + \operatorname{sgn}[\dot{p}_o(5.6325)][\operatorname{sgn}(p_e(5.6325))] \\ &= 1(1 + 1) - 1(-1) = 3 \end{aligned} \quad (2.15)$$

To verify that the calculated $l - r$ is indeed the right value, the roots of $p(s)$ are determined:

$$\begin{aligned} \text{roots} &= -3 - j \\ &\quad -3 - j \\ &\quad -1 - j \\ &\quad -1 + j \\ &\quad -5 \\ &\quad 1 \\ &\quad 2 \end{aligned}$$

It is observed from the roots of $p(s)$ that $l - r$ is 3 which is equivalent to the value calculated using equation (2.13)

CHAPTER III

STABILIZING A LINEAR TIME INVARIANT SYSTEM USING A
PROPORTIONAL CONTROLLER

In this chapter the algorithm to determine the entire set of Proportional (P) Controllers for a give transfer function using the generalized version of Hermite-Biehler theorem is introduced. The Graphical User Interface (GUI) developed in MATLAB[®] is also studied with an example. The advantage of the powerful GUI developed is, it lays a layer over the algorithm and the end user does not have to deal with the algorithm directly, but is guided through the algorithm using the GUI.

A. Theory and Algorithm

The theory and the algorithm to compute the entire set of stabilizing Proportional Controllers for a given transfer function is derived by Bhattacharyya and others is introduced in [19]. An overview of the same is now presented here.

Consider a plant $p(s)$ with the transfer function

$$p(s) = \frac{N(s)}{D(s)} \quad (3.1)$$

The controller in consideration $c(s)$ is a pure Proportional Controller given by,

$$c(s) = k_p \quad (3.2)$$

The closed loop characteristic polynomial is then give by,

$$\delta(s, k_p) = D(s) + k_p N(s) \quad (3.3)$$

Let n be the degree of the characteristic polynomial $\delta(s, k_p)$. The objective is to calculate all the values of k_p such that the characteristic polynomial is Hurwitz stable.

That is all the n roots of the equation ($\delta(s, k_p) = 0$) are on the left half of the complex plane. Thus if the polynomial is Hurwitz stable, the signature ($\sigma(p)$) should be equal to n

Writing the characteristic polynomial in equation(3.3) as even and odd part

$$\begin{aligned} N(s) &= N_e(s) + sN_o(s) \\ D(s) &= D_e(s) + sD_o(s) \\ \delta(s, k_p) &= [D_e(s) + k_p N_e(s)] + s[D_o(s) + k_p N_o(s)] \end{aligned} \quad (3.4)$$

As observed in equation (3.4) k_p appears both in the real and imaginary part of $\delta(s, k_p)$. To use the generalized Hermite-Biehler theorem it is desirable to have the gain k_p isolated either in the odd or even part of the polynomial. This can be achieved by multiplying the characteristic polynomial by $N(-s)$.

Defining,

$$\begin{aligned} N^*(s) &= N(-s) = N_e(s) - sN_o(s) \\ \nu(s, k_p) &= \delta(s, k_p)N^*(s) \end{aligned}$$

Thus to ensure that the polynomial $\delta(p(s, k_p))$ is Hurwitz stable, the signature of $\nu(s, k_p)$ is as in equation (3.5) and can be easily realized

$$\sigma(\nu(s, k_p)) = n - (l(N(s)) - r(N(s))) \quad (3.5)$$

In equation (3.5), $n, l(N(s)), r(N(s))$ are the degree of the characteristic polynomial, number of roots of the numerator in the left and right half of the complex plane. For a detailed derivation of equation (3.5) refer to [22]

Thus the gain stabilization problem has been reduced to a root counting problem for the equation $\nu(s, k_p) = 0$ where it is desired to have $n + r(N(s))$ roots on the left

half of the complex plane and $l(N(s))$ roots on the right half of the complex plane.

Decomposing $\nu(s, k_p)$ in terms of even and odd parts of $N(s)$ and $D(s)$ as follows,

$$\nu(s, k_p) = h_1(s) + k_p h_2(s) + s g_1(s)$$

where,

$$h_1(s) = D_e(s)N_e(s) - s^2 D_o(s)N_o(s)$$

$$h_2(s) = N_e(s)N_e(s) - s^2 N_o(s)N_o(s)$$

$$g_1(s) = D_o(s)N_e(s) - D_e(s)N_o(s)$$

Substituting $s = j\omega$ as follows

$$\nu(\omega, k_p) = p_1(\omega) + k_p p_2(\omega) + j\omega q_1(\omega)$$

where,

$$p_1(\omega) = D_e(\omega)N_e(\omega) + \omega^2 D_o(\omega)N_o(\omega)$$

$$p_2(\omega) = N_e(\omega)N_e(\omega) + \omega^2 N_o(\omega)N_o(\omega)$$

$$q_1(\omega) = D_o(\omega)N_e(\omega) - D_e(\omega)N_o(\omega)$$

The generalized Hermite-Biehler theorem derived in [21] can be applied to $\nu(\omega, k_p)$.

Let the real, non-negative, distinct roots with odd multiplicity of q_1 be For even degree of $\nu(\omega, k_p)$

$$0 < \omega_1 < \omega_2 < \omega_3 \cdots < \omega_{l-1} < \infty \quad (3.6)$$

Based on equation (2.13) and (2.12), imaginary signature for even degree of $\nu(\omega, k_p)$ is defined as follows

$$\gamma(I) = [i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1} 2i_{l-1} + (-1)^l i_l] \cdot (-1)^{l-1} \text{sgn}[q(\infty)]$$

where,

$$i_n = \text{sgn}(q_1(\omega_n)) \quad n \text{ goes from } 0 \text{ to } l$$

Also in the above equations ω_l is defined as ∞ .

And for odd degree of $\nu(\omega, k_p)$

$$0 < \omega_1 < \omega_2 < \omega_3 \cdots < \omega_{l-1} \quad (3.7)$$

Based on equation (2.13) and (2.12), imaginary signature for odd degree of $\nu(\omega, k_p)$ is defined as follows

$$\gamma(I) = [i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1}2i_{l-1}] \cdot (-1)^{l-1} \text{sgn}[q(\infty)]$$

where,

$$i_n = \text{sgn}(q_1(\omega_n)) \quad n \text{ goes from } 0 \text{ to } l-1$$

The signature for the polynomial $\nu(s, k_p)$ is obtained from equation (3.5). The signature can be calculated given the transfer function. Then the set of feasible strings are calculated that satisfy the signature condition.

Let F^* denote the set of feasible strings and A the set of all possible strings then

$$F^* = \{I \in A \mid \gamma(I) = n - (l(N(s)) - r(N(s)))\} \quad (3.8)$$

The constant gain k_p is determined if and only if the following conditions hold:

- F^* is not empty that is at least one feasible string exist.
- There exist a string $I = \{i_0, i_1, i_2, \cdots\} \in F^*$ such that

$$\max_{t:i_t>0}(L_t) < \min_{t:i_t<0}(U_t)$$

where,

$$L_t = -\frac{p_1(\omega_t)}{p_2(\omega_t)} \text{ for } i_t \in I, i_t > 0$$

$$U_t = -\frac{p_1(\omega_t)}{p_2(\omega_t)} \text{ for } i_t \in I, i_t < 0$$

If the above conditions are satisfied by the feasible strings $I_1, I_2, I_3, \dots, I_s \in F^*$, then the set of all stabilizing gains is given by

$$k_p = \cup_{r=1}^s k_r \quad (3.9)$$

where

$$k_r = (\max_{t:i_t>0, i_t \in I_r}(L_t), \min_{t:i_t<0, i_t \in I_r}(U_t)) \quad r = 1, 2, \dots, s$$

A detailed proof for the above theorem is available in [23].

Consider an example as follows:

Example III.1 Consider the open-loop transfer function to be stabilized by a Proportional Controller. The objective is to determine the entire set of Proportional Controllers k_p that will stabilize the given transfer function.

$$p(s) = \frac{N(s)}{D(s)} \quad (3.10)$$

where,

$$N(s) = s^3 - 4s^2 + 1s + 2$$

$$D(s) = s^5 + 8s^4 + 32s^3 + 46s^2 + 46s + 17$$

$N_e(\omega), N_o(\omega), D_e(\omega), D_o(\omega)$ are calculated as follows

$$N_e(\omega) = 4\omega^2 + 2$$

$$N_o(\omega) = -\omega^2 + 1$$

$$D_e(\omega) = 8\omega^4 - 46\omega^2 + 17$$

$$D_o(\omega) = \omega^4 - 32\omega^2 + 46$$

Calculating $\nu(\omega, k_p) = \delta(\omega, k_p)N^*(s)$, where $\delta(\omega, k_p)$ is the characteristic polynomial,

$$\begin{aligned}\nu(\omega, k_p) &= -\omega^8 + 65\omega^6 - 246\omega^4 + 22\omega^2 + 34 \\ &\quad + k_p(\omega^6 + 14\omega^4 + 17\omega^2 + 4) \\ &\quad + j\omega(12\omega^6 - 180\omega^4 + 183\omega^2 + 75)\end{aligned}$$

where we define $p_1(\omega), p_2(\omega), q(\omega)$ as follows,

$$\begin{aligned}p_1(\omega) &= -\omega^8 + 65\omega^6 - 246\omega^4 + 22\omega^2 + 34 \\ p_2(\omega) &= \omega^6 + 14\omega^4 + 17\omega^2 + 4 \\ q(\omega) &= 12\omega^6 - 180\omega^4 + 183\omega^2 + 75\end{aligned}$$

Determining roots of q that are real, non negative and odd multiplicity

$$\omega_0 = 0 < \omega_1 = 1.2018 < \omega_2 = 3.7240 < \omega_3 = \infty \quad (3.11)$$

Since the degree of $\nu(\omega, k_p)$ given by $n + m$ is even and the signature is 6 the string that satisfy the signature condition is $1, -1, 1, -1$. In this particular example only one set of valid string was found to exist. There always exists a possibility for many valid strings.

Thus evaluating for the entire set of k_p by imposing the above sign conditions on the real part of $\nu(\omega, k_p)$

$$\begin{aligned}p_1(\omega_0) + k_p p_2(\omega_0) &> 0 \\ p_1(\omega_1) + k_p p_2(\omega_1) &< 0 \\ p_1(\omega_2) + k_p p_2(\omega_2) &> 0 \\ p_1(\omega_3) + k_p p_2(\omega_3) &< 0\end{aligned}$$

Solving the above linear inequalities, the values of k_p that stabilize the system is

obtained. The range of k_p thus obtained is

$$-8.5000 \leq k_p \leq 4.2109 \quad (3.12)$$

B. The GUI for Calculating the Entire Set of Proportional Controllers for a LTI System

This section describes the Graphical User Interface (GUI) developed and an example of generating the entire set of k_p values for a given Linear Time Invariant system using the GUI. As seen in the previous section the algorithm used to calculate the set of k_p values requires a complete understanding of concepts like Mikhailov's criterion, Hermite-Biehler theorem and the generalized version of the Hermite-Biehler theorem. This process may be time consuming and many a times there may be gaps in understanding of the algorithm. To solve this issue, proposed is a Graphical User Interface. This GUI has been completely developed in MATLAB[®].

MATLAB[®] was selected as it can handle computationally intense algorithms and is extremely efficient in handling large Matrices. Moreover MATLAB[®] is available both in the industry and academia alike. Also the software has a control system toolbox, which handles a lot of computation related to control systems. The GUI has been developed as an add-on package for this control systems toolbox. It must be noted that such an attempt has been made in [24] using LabVIEW[®], a software by National Instruments. The GUI developed in LabVIEW[®] was accepted and is being commercialized by National Instruments, due to be released in the next version of the control systems toolbox by National Instruments. A similar attempt is being made with MATLAB[®].

1. GUI Based Calculation of Proportional Gain

In case of the model based approach, the inputs to the GUI are the numerator and the denominator of the transfer function. The entire program is function based and if desired all the steps followed in the algorithm can be viewed in the MATLAB[®] command window.

The input Numerator and Denominator accepted from the user is converted in terms of ' $j\omega$ '. It is then decomposed into even and odd parts. All this computation is carried out in two functions 'D_jw_e_o.m' and 'N_jw_e_o.m'.

The decomposed Numerator and Denominator $D_e(j\omega)$, $D_o(j\omega)$, $N_e(j\omega)$, $N_o(j\omega)$ is used to calculate p_1 , p_2 and q using the function 'p_q.m'.

The function 'real_non_negative_odd_roost.m' then determines the roots of the imaginary part of $\nu(\omega, k_p) = 0$. The imaginary part of $\nu(\omega, k_p)$ is q which was determined in the previous function. The 'satis_roots.m' function further evaluates the roots to determine the real, non negative roots with odd multiplicity.

The 'string_gen.m' function evaluates all possible strings based on the signature equation and the degree of the polynomial $\nu(\omega, k_p)$. It generates an output of all the valid strings satisfying the signature condition.

The function 'determine_A_b.m' and 'determine_Kp' together generate the final range of k_p values that will stabilize the given system. This is done by solving the linear inequalities generated by the real part of $\nu(\omega, k_p)$ at the various values of ω with the inequality being determined by the valid strings generated. The rest of the program deals with plotting the k_p and scaling it to handle values like $+\infty$, $-\infty$.

2. GUI Based Performance Evaluation of Proportional Controllers

The GUI based design also helps in generating subsets of the stabilizing set depending on user inputs like Gain-margin, Phase-margin, Rise-time, Settling-time and Overshoot. This feature helps give the user sets of controller that not only stabilize the system, but also satisfy the user performance criterion. This is achieved by calculating the entire set of possible controllers as discussed in the previous section and then reducing this generated set to a finite set of points. The five specified performances are calculated at each of the points. Though this process may seem computationally intense, with respect to the GUI the computation does not take long.

Alternatively a feature has also been provided for the user to manually explore the set. The three different modules essentially, module 1 which generates the set of stable controllers for a given transfer function, module 2 which handles the manual exploration of the stable set and module 3 which generates the subset of stable controllers given the performance specification are in most aspects independent to each other. This has been done on purpose so as to cross check the validity of the output against each other.

The inputs to the module that generates the subset based on specification from the user are the Numerator, Denominator and the various performance specifications. The entire set of k_p that stabilizes the system is obtained from module 1 which was described earlier in this section.

For a given value of controller the closed loop transfer functions are generated the step response obtained. The function 'setpspecs.m' generates data like the rise-time, settling-time, overshoot. The entire process is run in a 'for' loop for all the set of finite controllers. Once the data is available for the entire set, they are compared against the user defined specifications and the subset is generated.

3. Illustrative Example

Consider the following example where for a given plant $p(s)$ the entire set of k_p is to be determined.

$$p(s) = \frac{8.14s^3 + 9.68s^2 + 5.32s + 29.84}{s^7 + 16s^6 + 134s^5 + 716s^4 + 2000s^3 + 4500s^2 + 6000s - 100} \quad (3.13)$$

As seen in the figure (5), the coefficient of the Numerator and the Denominator are the inputs to the GUI. The entire set of k_p values that stabilizes the system is determined by the algorithm discussed earlier in this section.

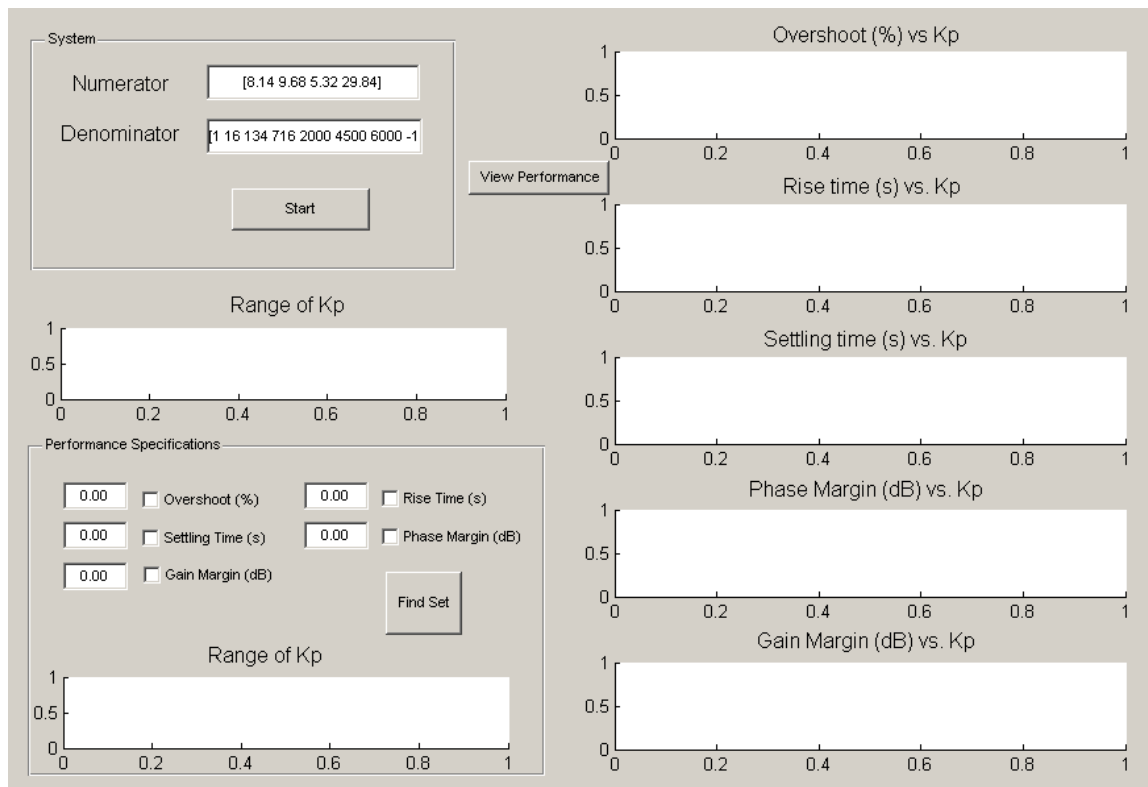


Fig. 5. GUI to determine the entire set of Proportional Controllers for a given transfer function

The steps involved in achieving this can be seen in the MATLAB[®] command

window if desired. $N_e(\omega), N_o(\omega), D_e(\omega), D_o(\omega)$ are calculated and displayed as follows

$$\begin{aligned} N_e(\omega) &= -9.68\omega^2 + 29.8400 \\ N_o(\omega) &= -8.14\omega^2 + 5.32 \\ D_e(\omega) &= -16\omega^6 - 716\omega^4 - 4500\omega^2 - 100 \\ D_o(\omega) &= -\omega^6 + 134\omega^4 - 2000\omega^2 + 6000 \end{aligned}$$

Calculating $\nu(\omega, k_p) = \delta(\omega, k_p)N^*(s)$, where $\delta(\omega, k_p)$ is the characteristic polynomial

$$\begin{aligned} \nu(\omega, k_p) &= 8.14\omega^{10} - 941.2\omega^8 - 9584.56\omega^6 + 5445.44\omega^4 - 101392\omega^2 - 2984 \\ &\quad + k_p(66.2596\omega^6 + 7.0928\omega^4 - 549.4\omega^2 + 890.4256) \\ &\quad + j\omega(-120.56\omega^8 + 4586.4\omega^6 - 17080.56\omega^4 - 94634\omega^2 + 179572) \end{aligned}$$

where we define $p1(\omega), p2(\omega), q(\omega)$

$$\begin{aligned} p_1(\omega) &= 8.14\omega^{10} - 941.2\omega^8 - 9584.56\omega^6 + 5445.44\omega^4 - 101392\omega^2 - 2984 \\ p_2(\omega) &= 66.2596\omega^6 + 7.0928\omega^4 - 549.4\omega^2 + 890.4256 \\ q(\omega) &= -120.56\omega^8 + 4586.4\omega^6 - 17080.56\omega^4 - 94634\omega^2 + 179572 \end{aligned}$$

Calculating the roots of q that are real, non negative with odd multiplicity.

$$\omega_0 = 0 < \omega_1 = 1.27315 < \omega_2 = 2.6827 < \omega_3 = 5.7519 \quad (3.14)$$

Since the degree of $\nu(\omega, k_p)$ given by $n + m$ is odd and the signature is 8 the string that satisfy the signature condition is $1, -1, 1, -1, 1$. In this particular example only one set of valid string was found to exist. There always exists a possibility that many valid strings exist.

Thus evaluating for the entire set of k_p by imposing the above sign conditions on

the real part of $\nu(\omega, k_p)$.

$$p_1(\omega_0) + k_p p_2(\omega_0) > 0$$

$$p_1(\omega_1) + k_p p_2(\omega_1) < 0$$

$$p_1(\omega_2) + k_p p_2(\omega_2) > 0$$

$$p_1(\omega_3) + k_p p_2(\omega_3) < 0$$

$$p_1(\omega_4) + k_p p_2(\omega_4) < 0$$

Solving the above linear inequalities, the values of k_p that stabilize the system is obtained. The range of k_p thus obtained is,

$$3.35 \leq k_p \leq 190.50 \quad (3.15)$$

The output is displayed in the plot inbuilt in the GUI as seen in the figure (6).

Further for all the values of k_p determined, the performances like Overshoot, Rise-time, Settling-time, Gain and Phase margin are found as discussed in the previous section and is displayed in the GUI as seen in figure (7). Also for a given specification a sub-set of k_p values can be determines. As seen in the figure (8), for the performance specification of Phase Margin greater than $45dB$ and the Overshoot less than 35% , the subset in red and the entire set in blue are displayed.

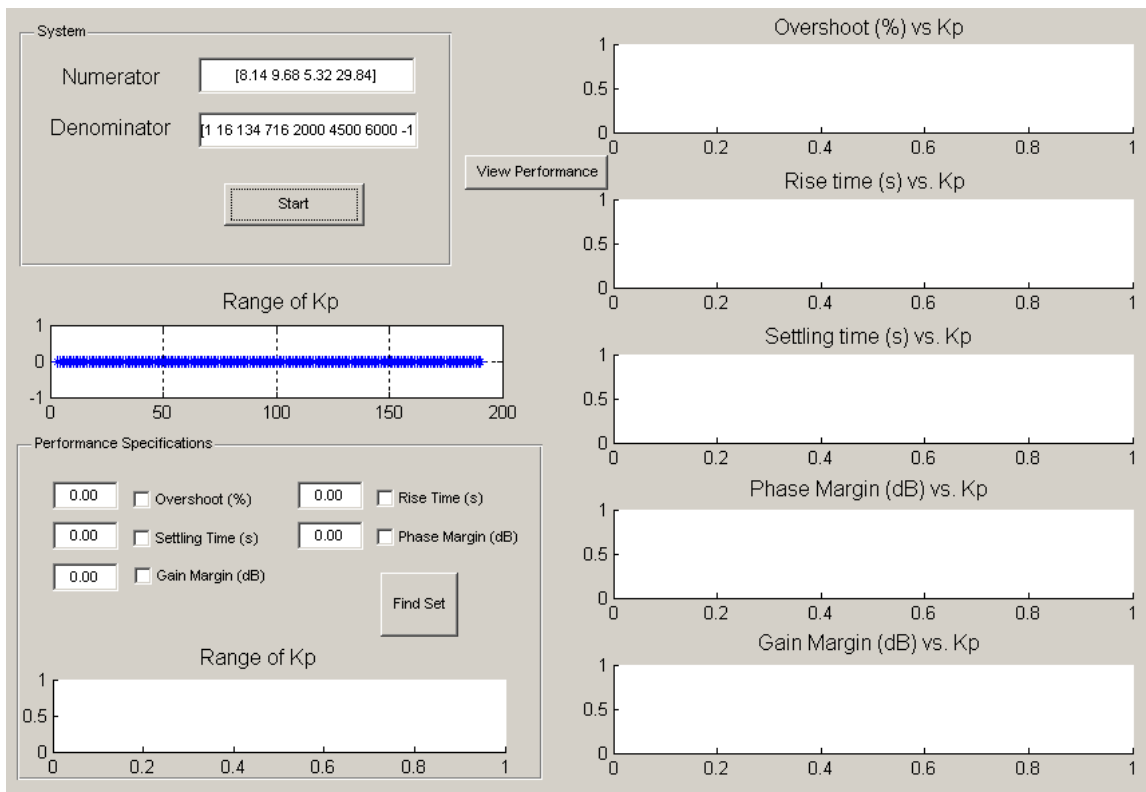


Fig. 6. GUI with the entire set of k_p displayed for the given transfer function

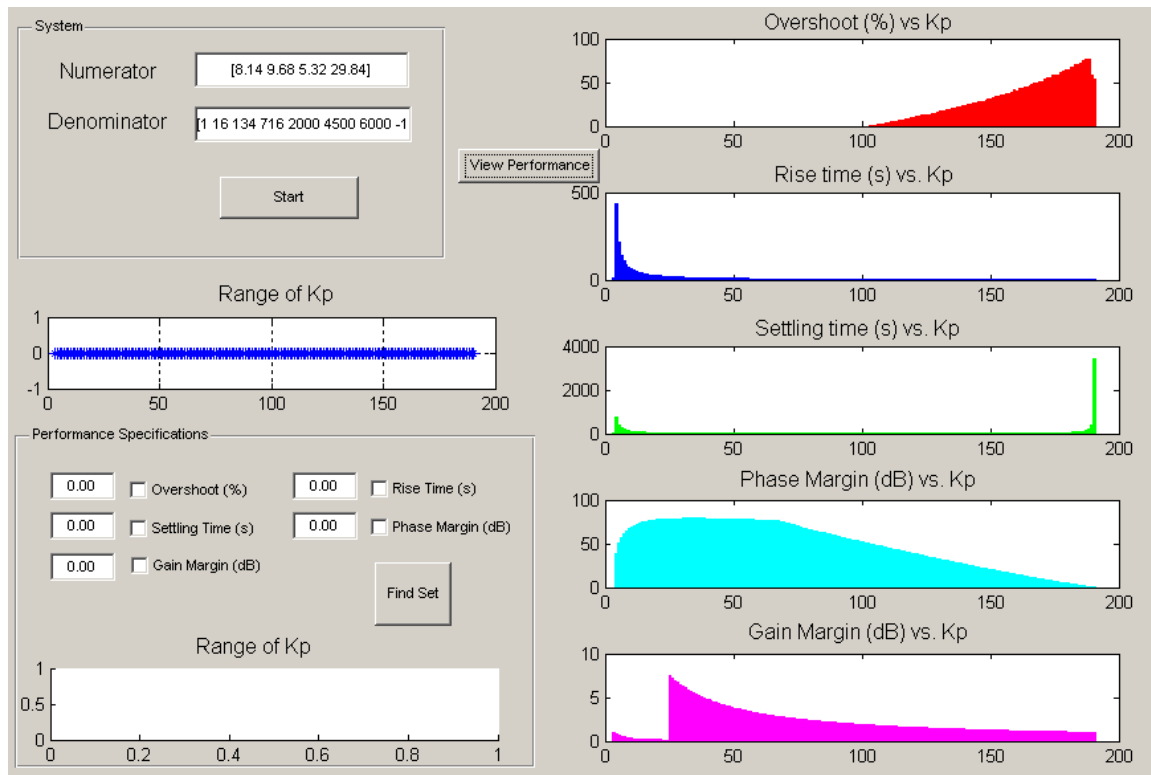


Fig. 7. GUI with all the performance displayed for the given transfer function

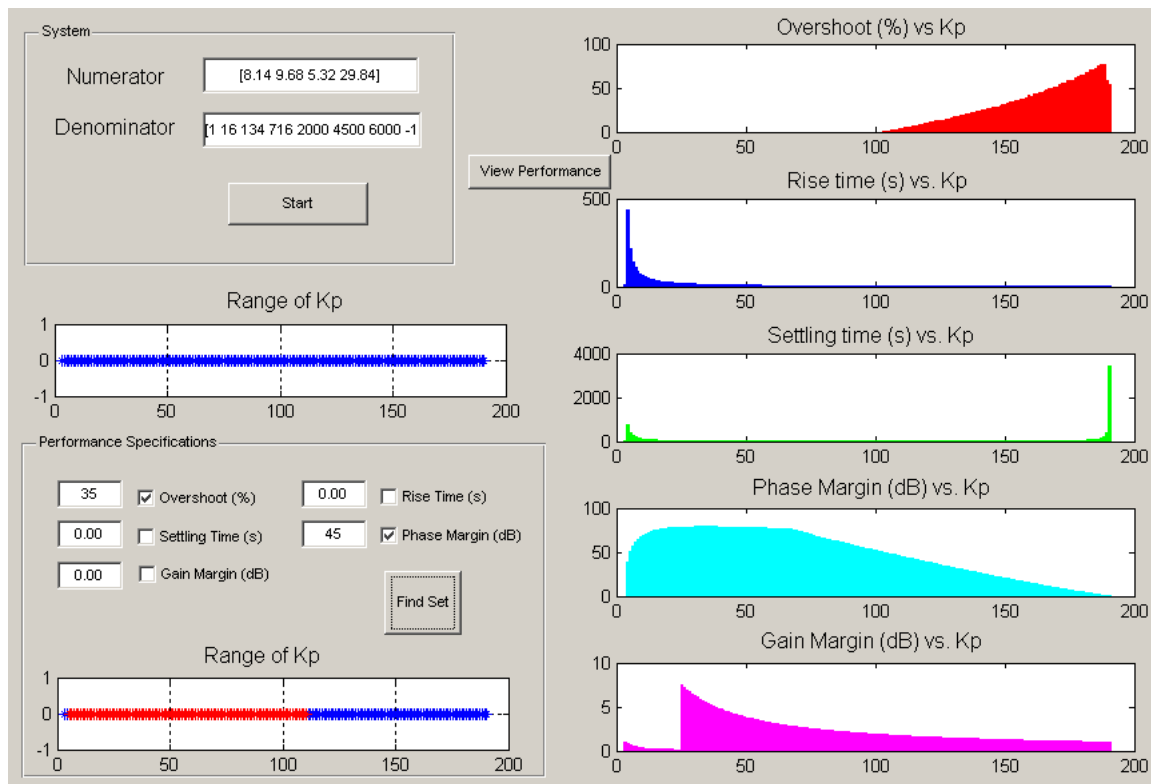


Fig. 8. GUI with a sub-set of stabilizing values of k_p satisfying the condition of PM $> 45dB$ and Overshoot $< 35\%$

CHAPTER IV

STABILIZING A LINEAR TIME INVARIANT SYSTEM USING A PI
CONTROLLER

In this chapter using the results obtained in Chapter II, the complete set of Proportional Integral (PI) controllers are determined for a given transfer function. The flow of text in this chapter is similar to Chapter III which dealt with pure Proportional Controllers. The algorithm to determine the entire set of PI controllers is introduced first followed by an example for better clarity. The GUI developed in MATLAB[®] with its features is then introduced which is followed by an illustrative example.

A. Theory and Algorithm

The theory and the algorithm to compute the entire set of stabilizing Proportional Integral controllers for a given transfer function is derived by Bhattacharyya and others and is introduced in [25]. An overview of the same is now presented.

Consider a plant $p(s)$ with the transfer function

$$p(s) = \frac{N(s)}{D(s)} \quad (4.1)$$

The controller in consideration $c(s)$ is a Proportional Integral controller given by

$$c(s) = k_p + \frac{k_i}{s} = \frac{k_p s + k_i}{s} \quad (4.2)$$

The closed loop characteristic polynomial is then give by

$$\delta(s, k_p, k_i) = D(s)s + (k_p s + k_i)N(s) \quad (4.3)$$

Let n be the degree of the characteristic polynomial $\delta(s, k_p, k_i)$. The objective is to calculate the entire set of k_p, k_i values such that the characteristic polynomial is

Hurwitz stable. That is all the n roots of the equation $(\delta(s, k_p, k_i) = 0)$ should be on the left half of the complex plane. Thus as seen in Chapter II, if the polynomial is Hurwitz stable, the signature should be equal to n

Writing the characteristic polynomial in equation(4.3) as even and odd part

$$\begin{aligned}
 N(s) &= N_e(s) + sN_o(s) \\
 D(s) &= D_e(s) + sD_o(s) \\
 &\text{in,} \\
 \delta(s, k_p, k_i) &= [(D_o(s) + k_p N_o(s))s^2 + k_i N_e] \\
 &\quad + s[D_e(s) + k_i N_o(s) + k_p N_e(s)] \tag{4.4}
 \end{aligned}$$

As observed in equation (4.4) both k_p and k_i appears both in the real and imaginary part of $\delta(s, k_p, k_i)$. Similar to the approach in Chapter II, it is desirable to have the k_i in the even part and k_p in the odd part of the polynomial. This can be achieved by multiplying the characteristic polynomial by $N(-s)$. Defining

$$N^*(s) = N(-s) = N_e(s) - sN_o(s) \tag{4.5}$$

$$\nu(s, k_p, k_i) = \delta(s, k_p, k_i)N^*(s) \tag{4.6}$$

Thus to ensure that the polynomial $\delta(s, k_p, k_i)$ is Hurwitz stable, the signature of $\nu(s, k_p, k_i)$ is as in equation (4.7) and can be easily realized

$$\sigma(\nu(s, k_p, k_i)) = n - (l(N(s)) - r(N(s))) \tag{4.7}$$

In equation (4.7) n is the degree of the characteristic polynomial $\delta(s, k_p, k_i)$, $l(N(s))$ and $r(N(s))$ are the number of roots of the numerator of the transfer function on the left and the right half of the complex plane respectively. The derivation of equation (4.7) is similar to [22]

Thus the gain stabilization problem has been reduced to a root counting problem for the equation $\nu(s, k_p, k_i) = 0$ where it is desired to have $n + r(N(s))$ roots on the left half of the complex plane and $l(N(s))$ roots on the right half of the complex plane.

Decomposing $\nu(s, k_p, k_i)$ in terms of even and odd parts of $N(s)$ and $D(s)$ as follows

$$\nu(s, k_p, k_i) = h_1(s) + k_i h_2(s) + g_1(s) + k_p g_2(s)$$

where,

$$h_1(s) = s^2(D_o(s)N_e(s) - D_e(s)N_o(s))$$

$$h_2(s) = N_e(s)N_e(s) - s^2N_o(s)N_o(s)$$

$$g_1(s) = s[D_e(s)N_e(s) - s^2D_o(s)N_o(s)]$$

$$g_2(s) = s[N_e(s)N_e(s) - s^2N_o(s)N_o(s)]$$

Substituting $s = j\omega$ as follows

$$\nu(\omega, k_p, k_i) = p_1(\omega) + k_i p_2(\omega) + j[q_1(\omega) + k_p q_2(\omega)]$$

where,

$$p_1(\omega) = -\omega^2[D_o(\omega)N_e(\omega) - D_e(\omega)N_o(\omega)]$$

$$p_2(\omega) = N_e(\omega)N_e(\omega) + \omega^2N_o(\omega)N_o(\omega)$$

$$q_1(\omega) = \omega[D_e(\omega)N_e(\omega) + \omega^2D_o(\omega)N_o(\omega)]$$

$$q_2(\omega) = \omega[N_e(\omega)N_e(\omega) + \omega^2N_o(\omega)N_o(\omega)]$$

The generalized Hermite-Biehler theorem derived in [21] can be applied to $\nu(\omega, k_p, k_i)$.

For even degree of $\nu(\omega, k_p, k_i)$ let the real, non-negative, distinct roots with odd multiplicity of $q_1 + k_p q_2$ be

$$\omega_0 = 0 < \omega_1 < \omega_2 < \omega_3 \cdots < \omega_{l-1} < \infty \quad (4.8)$$

Based on equation (2.13) and (2.12), imaginary signature for even degree $\nu(\omega, k_p, k_i)$ is defined as follows

$$\gamma(I) = [i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1}2i_{l-1} + (-1)^l i^l] \cdot (-1)^{l-1} \text{sgn}[q(\infty)]$$

where

$$i_n = \text{sgn}(q_1(\omega_n) + k_p q_2(\omega_n)) \quad n \text{ goes from } 0 \text{ to } l$$

In the above equations ω_l is defined as ∞ .

In case of odd degree of $\nu(\omega, k_p, k_i)$, let the real, non-negative, distinct roots with odd multiplicity of $q_1 + k_p q_2$ be

$$\omega_0 = 0 < \omega_1 < \omega_2 < \omega_3 \cdots < \omega_{l-1} \quad (4.9)$$

Based on equation (2.13) and (2.12), imaginary signature for odd degree $\nu(\omega, k_p, k_i)$ is defined as follows

$$\gamma(I) = [i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1}2i_{l-1}] \cdot (-1)^{l-1} \text{sgn}[q(\infty)]$$

where

$$i_n = \text{sgn}(q_1(\omega_n) + k_p q_2(\omega_n)) \quad n \text{ goes from } 0 \text{ to } l - 1$$

The signature for the polynomial $\nu(s, k_p, k_i)$ is known from equation (4.7). The signature can be calculated given the transfer function. Then the set of feasible strings are calculated that satisfy the signature condition. Let F^* denote the set of feasible strings, then

$$F^* = I \in A | \gamma(I) = n - (l(N(s)) - r(N(s))) \quad (4.10)$$

In equation (4.10) A is the set of all possible strings.

The set of controllers in $\{k_p, k_i\}$ space is determined for a given plant with rational

transfer function if and only if the following conditions hold

- F^* is not empty that is at least one feasible string exist.
- There exist a string $I = \{i_0, i_1, i_2, \dots\} \in F^*$ and values of k_i such that for all $t = 0, 1, 2 \dots$ for which

$$N^*(j\omega_t) \neq 0 \quad [p_1(\omega) + k_i p_2(\omega)]i_t > 0 \quad (4.11)$$

Also if there exist a set of values in k_i such that the above condition is satisfied for feasible strings $I_1, I_2, \dots \in F^*$

Then the set of stabilizing k_i values for a fixed k_p is the unions of all k_i values satisfying

$$[p_1(\omega) + k_i p_2(\omega)]i_t > 0 \text{ for } I_1, I_2, \dots \quad (4.12)$$

Consider an example as follows:

Example IV.1 Consider the open-loop transfer function to be stabilized by a Proportional Integral (PI) controller. The objective is to determine the entire set of PI controllers $\{k_p, k_i\}$ that will stabilize the given transfer function.

$$p(s) = \frac{N(s)}{D(s)} \quad (4.13)$$

where

$$\begin{aligned} N(s) &= s^3 + 4s^2 + 2s + 9 \\ D(s) &= s^4 + 4s^3 + 5s^2 + 8s + 16 \end{aligned}$$

$N_e(\omega), N_o(\omega), D_e(\omega), D_o(\omega)$ are calculated

$$N_e(\omega) = -4\omega^2 + 9$$

$$\begin{aligned}
N_o(\omega) &= -\omega^2 + 2 \\
D_e(\omega) &= \omega^4 - 5\omega^2 + 16 \\
D_o(\omega) &= -4\omega^2 + 8
\end{aligned}$$

Calculating $\nu(\omega, k_p) = \delta(\omega, k_p)N^*(s)$, where $\delta(\omega, k_p)$ is the characteristic polynomial

$$\begin{aligned}
\nu(\omega, k_p) &= -\omega^8 - 9\omega^6 - 42\omega^4 - 40\omega^2 \\
&\quad + k_i(\omega^6 + 12\omega^4 - 68\omega^2 + 81) \\
&\quad + j[(13\omega^5 - 93\omega^3 + 144\omega) \\
&\quad + k_p(\omega^7 + 12\omega^5 - 68\omega^3 + 81\omega)]
\end{aligned} \tag{4.14}$$

where we define $p_1(\omega), p_2(\omega), q_1(\omega), q_2(\omega)$

$$\begin{aligned}
p_1(\omega) &= -\omega^8 - 9\omega^6 - 42\omega^4 - 40\omega^2 \\
p_2(\omega) &= \omega^6 + 12\omega^4 - 68\omega^2 + 81 \\
q_1(\omega) &= 13\omega^5 - 93\omega^3 + 144\omega \\
q_2(\omega) &= \omega^7 + 12\omega^5 - 68\omega^3 + 81\omega
\end{aligned}$$

For $k_p = 2.1196$ determining roots of the imaginary part of $\nu(\omega, k_p)$ that is $q_1 + k_p q_2$ that are real, non negative and have odd multiplicity

$$\omega_0 = 0 < \omega_1 = 1.5094 < \omega_2 = 1.6777 < \omega_3 = \infty \tag{4.15}$$

Since the degree of $\nu(\omega, k_p, k_i)$ is even and the signature is 6 the string that satisfy the signature condition is $1, -1, 1, -1$. In this particular example only one set of valid string was found to exist. There always exists a possibility for many valid strings.

Thus evaluating for the entire set of k_i by imposing the above sign conditions on

the real part of $\nu(\omega, k_p, k_i)$

$$p_1(\omega_0) + k_i p_2(\omega_0) > 0$$

$$p_1(\omega_1) + k_i p_2(\omega_1) < 0$$

$$p_1(\omega_2) + k_i p_2(\omega_2) > 0$$

$$p_1(\omega_3) + k_i p_2(\omega_3) < 0$$

Solving the above linear inequalities for different values of k_i , the entire set in the $\{k_p, k_i\}$ space that stabilize the system is obtained and is as displayed in

figure (9)

B. GUI for Calculating the Entire Set of Proportional Integral Controllers for a LTI System

This section deals with the development of the GUI for determining the entire set of Proportional Integral controllers for a given transfer function based on the discussion in the previous sections of this chapter. Like in the case of Proportional Controller, the GUI has been developed in MATLAB[®].

1. GUI Based Calculation of Proportional Integral Values

The development of the GUI for Proportional Integral controller is on the same lines as that developed for Proportional Controller. The input to the GUI are the Numerator and the Denominator of the transfer function in the 's' domain. For a range of k_p values given by the user linear inequalities are solved to obtain the corresponding values of k_i . Scanning a range of k_p values and solving the corresponding linear inequalities, the entire set in $\{k_p, k_i\}$ space can be generated. The values of k_p that need to be scanned can be changed manually.

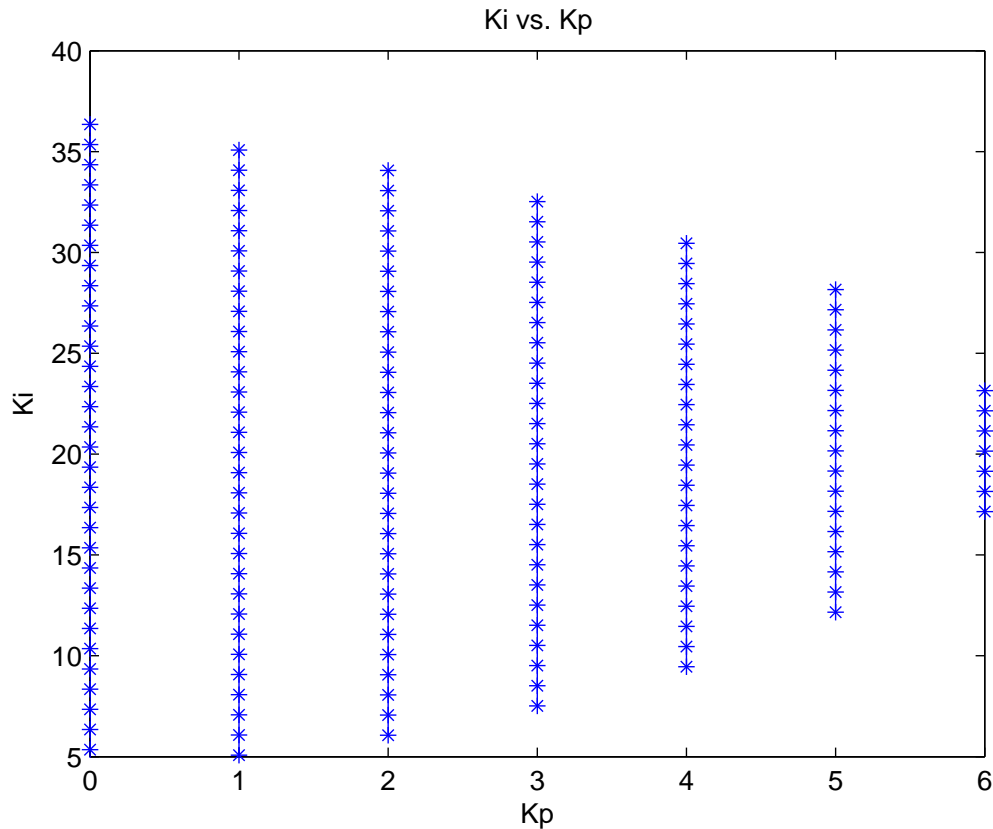


Fig. 9. The stabilizing set of (k_p, k_i) values for example (IV.1)

The ‘main_PI.m’ handles the overall GUI aspect of the program. This program call functions depending on the user interaction with the GUI. The computation of the entire set of k_p, k_i controllers is handled by the function ‘main_PI_computation.m’. This function replaces the ‘ s ’ with ‘ $j\omega$ ’ for the given transfer function. It then decomposes the Numerator and the Denominator into even and odd polynomials.

The function ‘p.q.m’ generates p_1, p_2, q_1, q_2 from the decomposed numerator and denominator $N_o(j\omega), N_e(j\omega), D_o(j\omega), D_e(j\omega)$

A feasible range of k_p values within -100 to 100 is scanned for the possible values of k_p that may stabilize the system. This is performed by comparing the number of

real non-negative roots with odd multiplicity obtained from the equation $q_1 + k_p q_2 = 0$ with the actual number of roots required. This entire computation is completed in the function ‘determine_required_roots.m’ and ‘determine_Kp.m’

The control is then transferred to the user where the user is required to select one value of k_p against which the entire set of k_i values are obtained. For a particular value of k_p selected the function ‘string_gen.m’ generates a set of valid strings that satisfy the signature equation.

‘determine_ki_main.m’ and ‘determine_A_b.m’ together solve the inequalities generated by evaluating the real part of the $\nu(\omega, k_p, k_i)$ at different values of ω at which the imaginary part is zero and the sign of inequalities being determined by the output of ‘string_gen.m’. Based on the above computation the output of ‘determine_ki_main.m’ is the entire set of k_i values for the selected k_p .

2. GUI Based Performance Evaluation of Proportional Integral Controllers

For a given set of performance condition like overshoot, rise-time, settling-time, gain margin and phase margin the GUI can determine the subset of stable controllers in the $\{k_p, k_i\}$ space that satisfy these conditions. This subset is displayed in addition to the original set of stable controllers, enabling the user to visually see the reduction in set of controllers.

The performance computation for the PI controller is similar to that of the P controller which was discussed in the previous chapter. The entire two dimensional set in the $\{k_p, k_i\}$ space is generated initially. This set is then converted into a finite set of points. It must be noted that by using the algorithm introduced in [25], the set of $\{k_p, k_i\}$ values over which the performance has to be evaluated is limited to the finite stabilizing set obtained rather than the infinite 2d space. For each point in the set the performance is evaluated. For the calculation of the settling time, rise

time and overshoot the step response of the closed loop system is generated for each $\{k_p, k_i\}$ value. From the step response the three performance criteria are obtained. For the calculation of the gain and phase margin the open loop transfer function is calculated for each $\{k_p, k_i\}$ value. This though being computationally intense, it is handled well by MATLAB[®].

Also if the user desires, for a selected value of k_p the user can view all the performance specifications for all the corresponding value of k_i . In this manner a relative comparison can be made between different available values of k_i based on the five performance criteria. As in case of the Proportional Controllers, the above described modules, namely ‘generating the entire stable set of controllers’, ‘subset generation based on the user inputs’ and ‘exploring the set of stable k_i values manually’ are relatively independent of each other for the purpose of cross checking the output of the GUI.

3. Illustrative Example

Consider the following example where for a given plant $p(s)$ the entire set of PI controllers is to be determined.

$$p(s) = \frac{200s^3 + 200s^2 + 508.8}{s^8 + 33s^7 + 459s^6 + 3477s^5 + 15544s^4 + 3182s^3 + 56856s^2 + 60568s} \quad (4.16)$$

As seen in the figure (10), the coefficient of the Numerator and the Denominator are the inputs to the GUI. The entire set of $\{k_p, k_i\}$ values that stabilizes the system is determined by the algorithm discussed earlier in this section.

The steps involved in achieving this can be seen in the MARLAB[®] command window if desired. $N_e(\omega)$, $N_o(\omega)$, $D_e(\omega)$, $D_o(\omega)$ are calculated and displayed as follows

$$N_e(\omega) = -200\omega^2 + 508.8$$

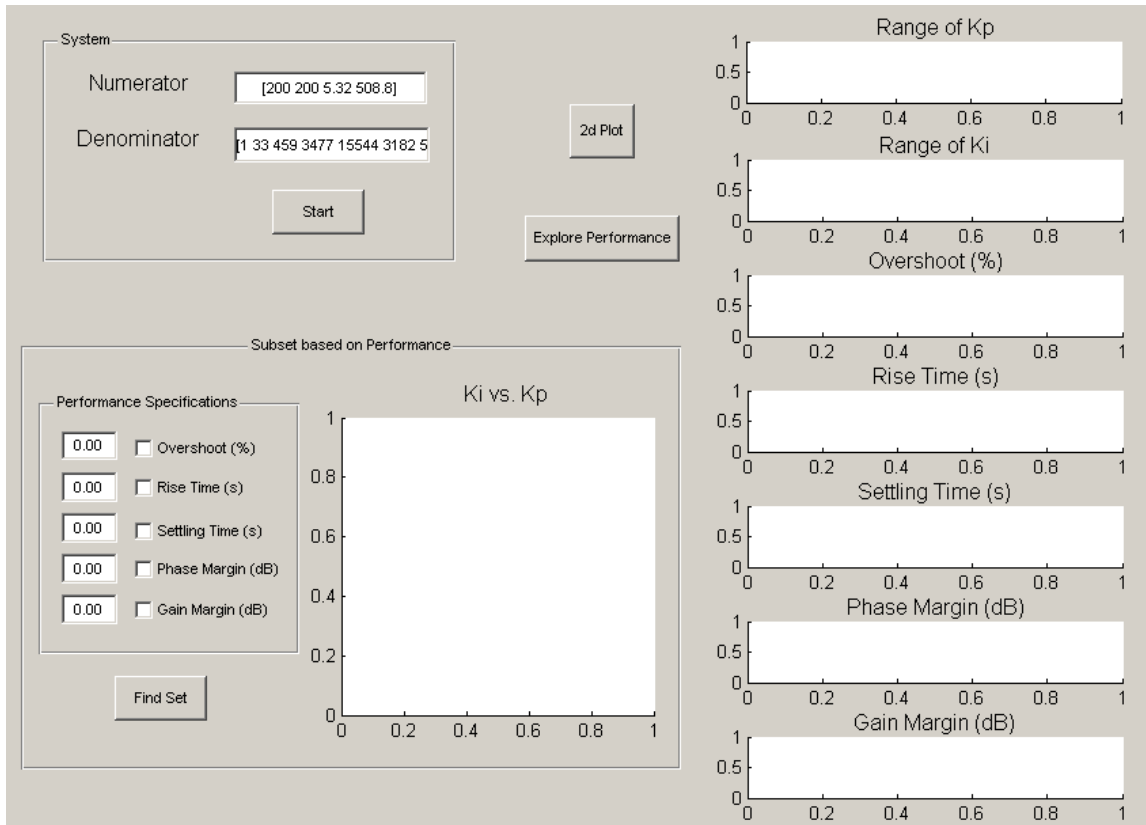


Fig. 10. GUI for determining the entire set of Proportional Integral controllers for a given transfer function

$$N_o(\omega) = -200\omega^2 + 5.32$$

$$D_e(\omega) = \omega^8 - 459\omega^6 + 15544\omega^4 - 56856\omega^2$$

$$D_o(\omega) = -33\omega^8 + 3477\omega^6 - 3182\omega^2 + 60568$$

Calculating $\nu(\omega, k_p, k_i) = \delta(\omega, k_p, k_i)N^*(s)$, where $\delta(\omega, k_p, k_i)$ is the characteristic polynomial

$$\begin{aligned} \nu(\omega, k_p, k_i) = & -200\omega^{12} + 85200\omega^{10} - 2396609.6\omega^8 \\ & + 8965702.4\omega^6 + 13732601.6\omega^4 - 30816998.4\omega^2 \\ & + k_i(40000\omega^6 + 40000\omega^4 - 203520\omega^2) \end{aligned}$$

$$\begin{aligned}
&+258877.44) \\
&+j\omega[(6400\omega^{10} - 603091.2\omega^8 - 2705939\omega^6 \\
&+7166387.2\omega^4 - 28928332.8\omega^2) \\
&+k_p(40000\omega^6 + 40000\omega^4 - 203520\omega^2 \\
&+258877.44)]
\end{aligned}$$

where we define $p_1(\omega), p_2(\omega), q_1(\omega), q_2(\omega)$

$$\begin{aligned}
p_1(\omega) &= -200\omega^{12} + 85200\omega^{10} - 2396609.6\omega^8 \\
&\quad + 8965702.4\omega^6 + 13732601.6\omega^4 - 30816998.4\omega^2 \\
p_2(\omega) &= 40000\omega^6 + 40000\omega^4 - 203520\omega^2 \\
&\quad + 258877.44 \\
q_1(\omega) &= \omega(6400\omega^{10} - 603091.2\omega^8 - 2705939\omega^6 \\
&\quad + 7166387.2\omega^4 - 28928332.8\omega^2) \\
q_2(\omega) &= \omega(40000\omega^6 + 40000\omega^4 - 203520\omega^2 \\
&\quad + 258877.44)
\end{aligned}$$

Determining roots of $q_1 + k_p q_2 = 0$ that are real, non negative and odd multiplicity for k_p value 216.7391

$$\omega_0 = 0 < \omega_1 = 1.1077 < \omega_2 = 1.2548 < \omega_3 = 3.6104 < \omega_4 = 9.0891 < \omega_5 = \infty \tag{4.17}$$

Since the degree of $\nu(\omega, k_p)$ given by $n + m$ is even and the signature is 10 the string that satisfy the signature condition is $1, -1, 1, -1, 1, -1$. In this particular example only one set of valid string was found to exist. There always exists a possibility for

many valid strings.

Thus evaluating for the entire set of k_i by imposing the above sign conditions on the real part of $\nu(\omega, k_p, k_i)$ for a given value of k_p

$$p_1(\omega_0) + k_p p_2(\omega_0) > 0$$

$$p_1(\omega_1) + k_p p_2(\omega_1) < 0$$

$$p_1(\omega_2) + k_i p_2(\omega_2) > 0$$

$$p_1(\omega_3) + k_i p_2(\omega_3) < 0$$

$$p_1(\omega_4) + k_i p_2(\omega_4) < 0$$

$$p_1(\omega_5) + k_i p_2(\omega_5) < 0$$

Solving the above linear inequalities, the values of k_i , for a given k_p that stabilize the system is obtained. For a fixed value of $k_p = 216.7391$, the range of k_i thus obtained is

$$0 \leq k_i \leq 40.3053 \tag{4.18}$$

The output is displayed in the plot inbuilt in the GUI as seen in the figure (11). Scanning the entire range of k_p and solving linear inequalities to obtain the corresponding values of k_i the entire stable set PI controllers are obtained as seen in figure (12). This is obtained in a separate pop up window on clicking the ‘2d plot’ button available on the GUI. Further in a 1d set of fixed k_p and varying k_i , the GUI can determine, the performances like Overshoot, Rise-time, Settling-time, Gain and Phase margin are found as discussed in the previous section and is displayed in the GUI as seen in figure (13). Also for a given specification a sub-set of stabilizing $\{k_p, k_i\}$ values can be determined. As seen in the figure (14), for the performance specification of Phase Margin greater than $7dB$ and the Rise-time less than $2sec$, the subset satisfying the performance criteria in red and the entire stabilizing set in blue are

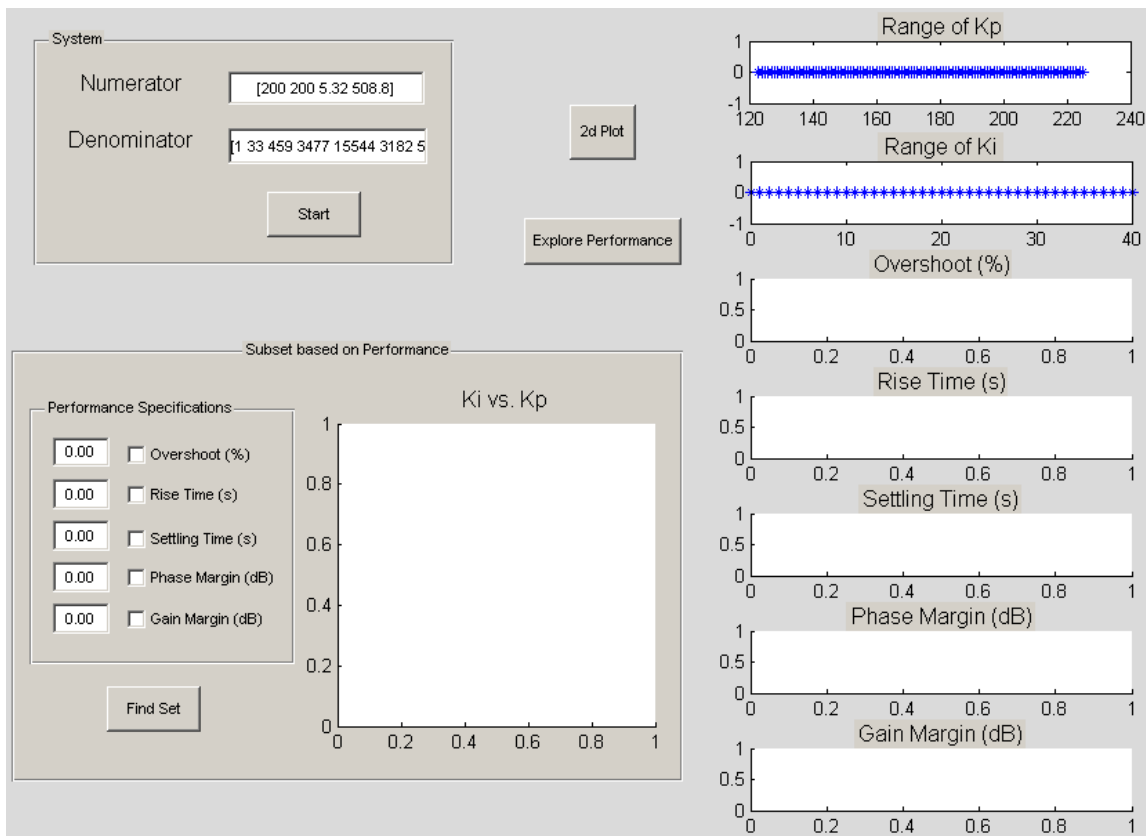


Fig. 11. GUI with the entire set of k_i for a fixed value of k_p

displayed.

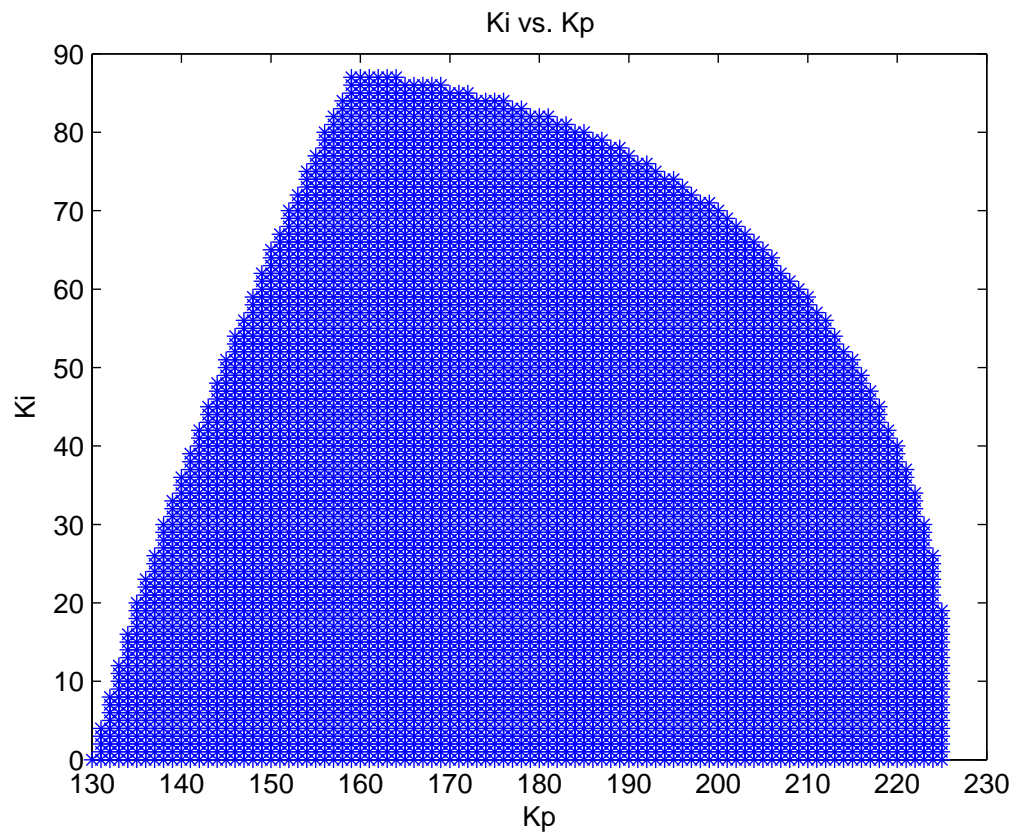


Fig. 12. GUI with the entire set of k_p displayed for the given transfer function

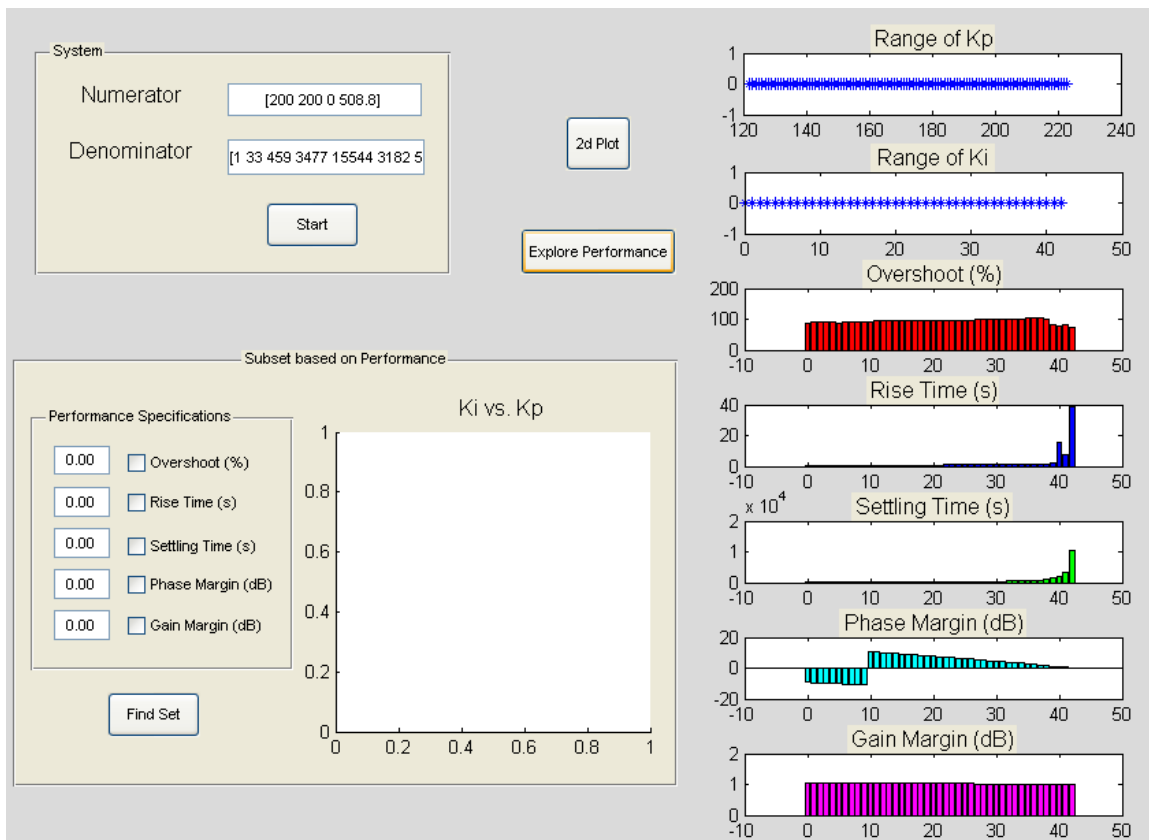


Fig. 13. GUI with all the performance displayed for the given transfer function

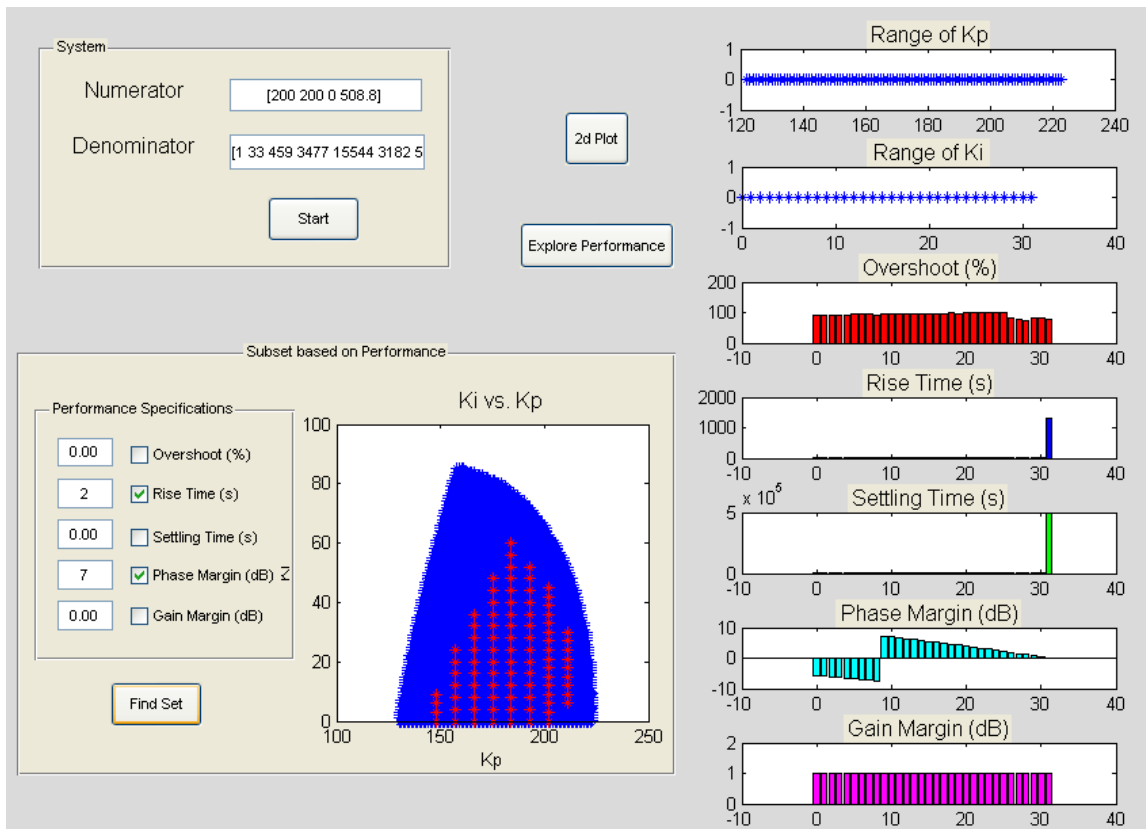


Fig. 14. GUI with a subset of stabilizing values in $\{k_p, k_i\}$ space satisfying the condition of $PM > 7dB$ and Rise-time $< 2sec$

CHAPTER V

STABILIZING A LINEAR TIME INVARIANT SYSTEM USING A PID
CONTROLLER

Using the Generalized Hermite-Biehler theorem introduced in [21], in Chapter III and IV the algorithm and the GUI to calculate the entire set of P and PI controllers were presented. In this chapter the algorithm to calculate the entire set of PID controllers is introduced. As it will be seen from the following discussion the logic in calculation of the entire set of PID controller is similar to calculation of entire set of P and PI controllers for a given transfer function.

A. Theory and Algorithm

The theory and the algorithm to compute the entire set of stabilizing Proportional Integral Derivative controllers for a given transfer function is derived by Bhattacharyya and others and is introduced in [3]. An overview of the same is now presented.

Consider a plant $p(s)$ with the transfer function

$$p(s) = \frac{N(s)}{D(s)} \quad (5.1)$$

The controller in consideration $c(s)$ is a Proportional Integral Derivative controller given by

$$c(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_d s^2 + k_p s + k_i}{s} \quad (5.2)$$

The closed loop characteristic polynomial is then given by

$$\delta(s, k_p, k_i, k_d) = D(s)s + (k_d s^2 + k_p s + k_i)N(s) \quad (5.3)$$

Let n be the degree of the characteristic polynomial $\delta(s, k_p, k_i, k_d)$. The objective is

to calculate the entire set of $\{k_p, k_i, k_d\}$ values such that the characteristic polynomial is Hurwitz stable. That is all the n roots of the equation ($\delta(s, k_p, k_i, k_d) = 0$) are on the left half of the complex plane. Thus as seen in Chapter II, if the polynomial is Hurwitz stable, the signature ($\sigma(p)$) should be equal to n .

Writing the characteristic polynomial in equation (5.3) as even and odd part,

$$N(s) = N_e(s) + sN_o(s)$$

$$D(s) = D_e(s) + sD_o(s)$$

in

$$\begin{aligned} \delta(s, k_p, k_i, k_d) = & [(D_o(s) + k_d N_e(s) + k_p N_o(s))s^2 + k_i N_e(s)] \\ & + s[D_e(s) + k_p N_e(s) + k_p N_e(s)s^2 + k_i N_o(s)] \end{aligned} \quad (5.4)$$

As observed in equation (5.4) k_p, k_i and k_d appears both in the real and imaginary part of $\delta(s, k_p, k_i, k_d)$. Similar to the approach in case of pure Proportional Controllers or Proportional Integral controllers in Chapter III, it is desirable to have the k_i and k_d only in the even part and k_p only in the odd part of the polynomial. This can be achieved by multiplying the characteristic polynomial by $N(-s)$. Defining

$$N^*(s) = N(-s) = N_e(s) - sN_o(s) \quad (5.5)$$

$$\nu(s, k_p, k_i, k_d) = \delta(s, k_p, k_i, k_d)N^*(s) \quad (5.6)$$

Thus to ensure that the polynomial $\delta(s, k_p, k_i, k_d)$ is Hurwitz stable, the signature of $\nu(s, k_p, k_i, k_d)$ is as in equation (5.7) and can be easily realized

$$\sigma(\nu(s, k_p, k_i, k_d)) = n - (l(N(s)) - r(N(s))) \quad (5.7)$$

The derivation of equation (5.7) is similar to [22]

Thus the gain stabilization problem has been reduced to a root counting problem for the equation $\nu(s, k_p, k_i, k_d) = 0$ where it is desired to have $n + r(N(s))$ roots on the left half of the complex plane and $l(N(s))$ roots on the right half of the complex plane.

Decomposing $\nu(s, k_p, k_i, k_d)$ in terms of even and odd parts of $N(s)$ and $D(s)$ as under

$$\nu(s, k_p, k_i, k_d) = h_1(s) + (k_i + k_d s^2)h_2(s) + g_1(s) + k_p g_2(s)$$

where,

$$h_1(s) = s^2(D_o(s)N_e(s) - D_e(s)N_o(s))$$

$$h_2(s) = N_e(s)N_e(s) - s^2N_o(s)N_o(s)$$

$$g_1(s) = s[D_e(s)N_e(s) - s^2D_o(s)N_o(s)]$$

$$g_2(s) = s[N_e(s)N_e(s) - s^2N_o(s)N_o(s)]$$

Substituting $s = j\omega$ as follows

$$\nu(\omega, k_p, k_i, k_d) = p_1(\omega) + (k_i - k_d \omega^2)p_2(\omega) + j[q_1(\omega) + k_p q_2(\omega)]$$

where,

$$p_1(\omega) = -\omega^2[D_o(\omega)N_e(\omega) - D_e(\omega)N_o(\omega)]$$

$$p_2(\omega) = N_e(\omega)N_e(\omega) + \omega^2N_o(\omega)N_o(\omega)$$

$$q_1(\omega) = \omega[D_e(\omega)N_e(\omega) + \omega^2D_o(\omega)N_o(\omega)]$$

$$q_2(\omega) = \omega[N_e(\omega)N_e(\omega) + \omega^2N_o(\omega)N_o(\omega)]$$

The generalized Hermite-Biehler theorem derived in [21] can be applied to $\nu(\omega, k_p, k_i, k_d)$.

For even degree of $\nu(\omega, k_p, k_i, k_d)$ let the real, non-negative, distinct roots with

odd multiplicity of the imaginary part of $\nu(\omega, k_p, k_i)$ which is $q_1 + k_p q_2$ be

$$\omega_0 = 0 < \omega_1 < \omega_2 < \omega_3 \cdots < \omega_{l-1} \omega = \infty \quad (5.8)$$

Based on equation (2.13) and (2.12), imaginary signature for even degree of $\nu(\omega, k_p, k_i, k_d)$ is defined as follows

$$\gamma(I) = [i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1} 2i_{l-1} + -1^l i_l] \cdot (-1)^{l-1} \text{sgn}[q(\infty)] \quad (5.9)$$

where

$$i_n = \text{sgn}(q_1(\omega_n) + k_p q_2(\omega_n)) \quad n \text{ goes from } 0 \text{ to } l$$

In the above equations ω_l is defined as ∞ .

For odd degree of $\nu(\omega, k_p, k_i, k_d)$ let the real, non-negative, distinct roots with odd multiplicity of the imaginary part of $\nu(\omega, k_p, k_i)$ which is $q_1 + k_p q_2$ be

$$\omega_0 = 0 < \omega_1 < \omega_2 < \omega_3 \cdots < \omega_{l-1} \quad (5.10)$$

Based on equation (2.13) and (2.12), imaginary signature for odd degree of $\nu(\omega, k_p, k_i, k_d)$ is defined as follows

$$\gamma(I) = [i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1} 2i_{l-1}] \cdot (-1)^{l-1} \text{sgn}[q(\infty)] \quad (5.11)$$

where

$$i_n = \text{sgn}(q_1(\omega_n) + k_p q_2(\omega_n)) \quad n \text{ goes from } 0 \text{ to } l - 1$$

The signature for the polynomial $\nu(s, k_p, k_i, k_d)$ is known from equation (5.7). The signature can be calculated given the transfer function. Then the set of feasible strings are calculated that satisfy the signature condition.

Let F^* denote the set of feasible strings, then

$$F^* = I \in A_{k_p} | \gamma(I) = n - (l(N(s)) - r(N(s))) \quad (5.12)$$

In equation (5.12) A is the entire set of strings.

The set of controllers in $\{k_p, k_i, k_d\}$ space is determined for a given plant with rational transfer function if and only if the following conditions hold

- F^* is not empty that is at least one feasible string exist.
- There exist a string $I = \{i_0, i_1, i_2, \dots\} \in F^*$ and values of k_i and k_d such that for all $t = 0, 1, 2 \dots$ for which

$$N^*(j\omega_t) \neq 0 \quad [p_1(\omega) + (k_i - k_d\omega^2)p_2(\omega)]i_t > 0 \quad (5.13)$$

Also if there exist a set of values in k_i and k_d such that the above condition is satisfied for feasible strings $I_1, I_2, \dots \in F^*$

Then the set of stabilizing $\{k_i, k_d\}$ values for a fixed k_p is the unions of all $\{k_i, k_d\}$ sets satisfying

$$[p_1(\omega) + (k_i - k_d\omega^2)p_2(\omega)]i_t > 0 \text{ for } I_1, I_2, \dots \quad (5.14)$$

Consider an example as follows:

Example V.1 Consider the open-loop transfer function to be stabilized by a Proportional Integral Derivative (*PID*) controller. The objective is to determine the entire set of PID controllers $\{k_p, k_i, k_d\}$ that will stabilize the given transfer function.

$$p(s) = \frac{N(s)}{D(s)} \quad (5.15)$$

where

$$\begin{aligned} N(s) &= s^3 + 6s^2 - 2s + 1 \\ D(s) &= s^5 + 3s^4 + 29s^3 + 15s^2 - 3s + 60 \end{aligned}$$

$N_e(\omega), N_o(\omega), D_e(\omega), D_o(\omega)$ are calculated

$$\begin{aligned} N_e(\omega) &= -6\omega^2 + 1 \\ N_o(\omega) &= -\omega^2 - 2 \\ D_e(\omega) &= 3\omega^4 - 15\omega^2 + 60 \\ D_o(\omega) &= \omega^4 - 29\omega^2 - 3 \end{aligned}$$

Calculating $\nu(\omega, k_p, k_i, k_d) = \delta(\omega, k_p, k_i, k_d)N^*(s)$, where $\delta(\omega, k_p, k_i, k_d)$ is the characteristic polynomial

$$\begin{aligned} \nu(\omega, k_p) &= 3\omega^8 - 166\omega^6 - 19\omega^4 - 117\omega^2 \\ &\quad + (k_i - k_d\omega^2)(\omega^6 + 40\omega^4 - 8\omega^2 + 1) \\ &\quad + j[(-\omega^9 + 9\omega^7 + 154\omega^5 - 369\omega^3 + 60\omega) \\ &\quad + k_p(\omega^7 + 40\omega^5 - 8\omega^3 + \omega)] \end{aligned}$$

where we define $p_1(\omega), p_2(\omega), q_1(\omega), q_2(\omega)$

$$\begin{aligned} p_1(\omega) &= 3\omega^8 - 166\omega^6 - 19\omega^4 - 117\omega^2 \\ p_2(\omega) &= \omega^6 + 40\omega^4 - 8\omega^2 + 1 \\ q_1(\omega) &= -\omega^9 + 9\omega^7 + 154\omega^5 - 369\omega^3 + 60\omega \\ q_2(\omega) &= \omega^7 + 40\omega^5 - 8\omega^3 + \omega \end{aligned}$$

For $k_p = 8.0128$ determining roots of $q_1 + k_p q_2$ that are real, non negative and have odd multiplicity

$$\omega_0 = 0 < \omega_1 = 0.4496 < \omega_2 = 0.8296 < \omega_3 = 5.6212 \quad (5.16)$$

Since the degree of $\nu(\omega, k_p, k_i, k_d)$ given by is odd and the signature is 7 the string

that satisfy the signature condition is $1, -1, 1, -1$. In this particular example only one set of valid string was found to exist. There always exists a possibility for many valid strings.

Thus evaluating for the entire set of $\{k_i, k_d\}$ by imposing the above sign conditions on the real part of $\nu(\omega, k_p, k_i, k_d)$

$$p_1(\omega_0) + (k_i - \omega^2)p_2(\omega_0) > 0$$

$$p_1(\omega_1) + (k_i - \omega^2)p_2(\omega_1) < 0$$

$$p_1(\omega_2) + (k_i - \omega^2)p_2(\omega_2) > 0$$

$$p_1(\omega_3) + (k_i - \omega^2)p_2(\omega_3) < 0$$

Solving the above linear inequalities, the entire set of controllers in the $\{k_i, k_d\}$ space for a fixed value of $k_p = 8.0128$ is obtained and is as seen in the figure (15). Also by varying k_p similar linear inequalities in the $\{k_i, k_d\}$ space can be generated. The plot of the entire set of PID controllers that stabilize the given system is as seen in the figure (16).

B. GUI for Calculating the Entire Set of Proportional Integral Derivative Controllers for a LTI System

As in the case with P, PI controllers the GUI for determining the entire set of PID controllers that will stabilize the system has been developed in MATLAB[®]. Like the other GUIs, this too has three modules namely, determining the entire stabilizing set in $\{k_p, k_i, k_d\}$ space for the given transfer function, determining the subset of controllers that satisfy the given performance specification and for a given set of controllers determining the performance specification like gain margin, phase margin, rise-time, settling-time and overshoot. For a chosen PID value, the GUI also displays

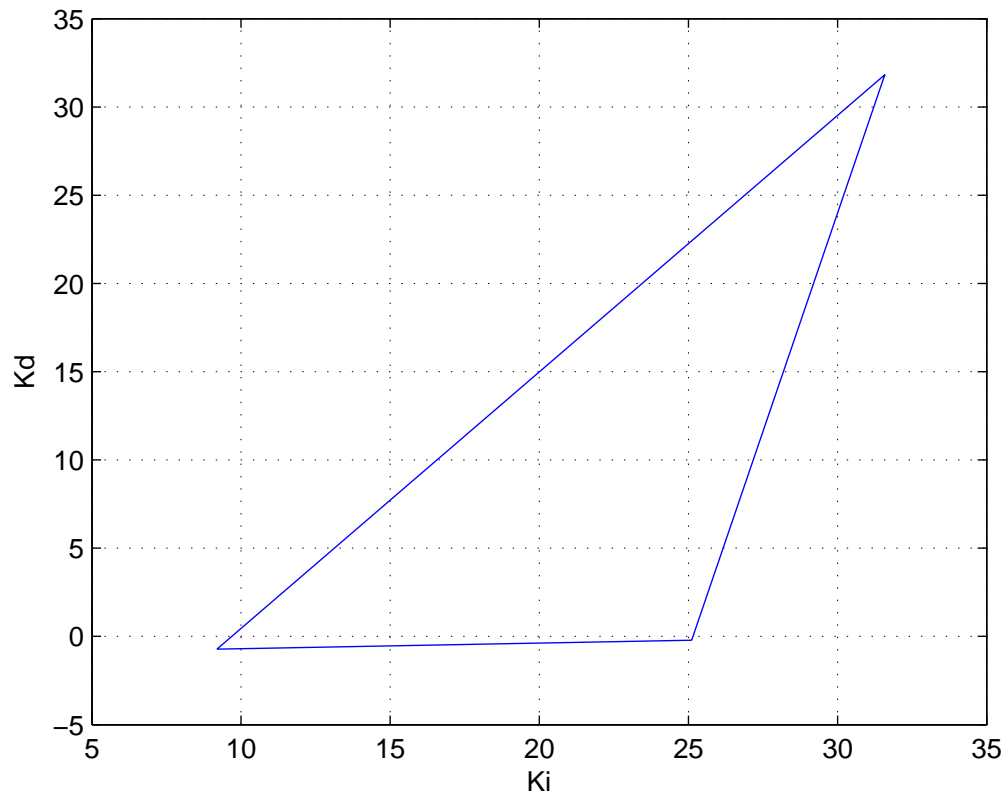


Fig. 15. Stabilizing values in $\{k_i, k_d\}$ space for a fixed value of k_p

the unit step response and the output of the error signal.

1. GUI Based Calculation of Proportional Integral Derivative Values

The GUI determines the entire set of PID values given the transfer function. A 3d set in $\{k_p, k_i, k_d\}$ space can be viewed or for a more detailed analysis for a selected value of k_p a 2d set in $\{k_i, k_d\}$ space can be viewed also. This feature has been developed on the same lines as the previous two GUI's discussed for the Proportional and Proportional Integral controller.

The main program that initializes the GUI is 'main_PID.m'. This program calls

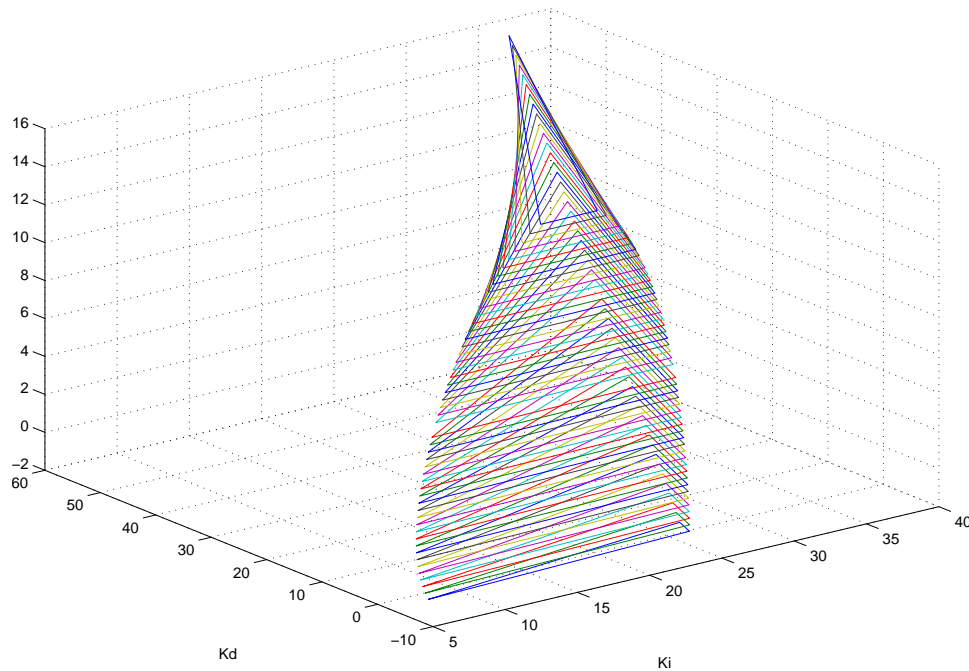


Fig. 16. The stabilizing set of $\{k_p, k_i, k_d\}$ values for example (V.1)

other programs depending on the user input. The program that is responsible for the basic computation is ‘main_PID_computation.m’. The entire code is functional based enabling each code to be independently tested and for purposed of recycling the code for other programs. The two inputs to the GUI are numerator and the denominator of the transfer function whose PID values are to be determined. These values are transferred to ‘main_PID_computation.m’.

‘main_PID_computation.m’ first calls the function ‘s2jw.m’ that converts the numerator and the denominator from the ‘s’ domain in terms of ‘j ω ’. The output of this function namely the numerator and the denominator in ‘j ω ’ terms is decomposed into even and odd parts by the function ‘N_jw_e.o.m’ and ‘D_jw_e.o.m’ respectively.

The function ‘p-q.m’ converts the decomposed even and odd parts of the nu-

erator and denominator into p_1, p_2, q_1, q_2 . This involves the calculation of the $\nu(\omega, k_p, k_i, k_d)$ which is the product of the characteristic polynomial and conjugate of the numerator and then generating the above polynomials p_1, p_2, q_1, q_2 .

The function ‘determine_required_roots.m’ and ‘determine_Kp.m’ determine the set of k_p values that satisfies the necessary condition for stability. This is done by calculating the minimum number of real non negative roots with odd multiplicity and comparing them with actual roots of the imaginary part of $\nu(\omega, k_p, k_i, k_d)$ for various values of k_p ranging from -500 to 500 .

The control is then transferred to the user who selects a value of k_p from the available values. The real, non negative, roots with odd multiplicity of the imaginary part of $\nu(\omega, k_p, k_i, k_d) = 0$ is determined. Note the imaginary part for a fixed value of k_p is a function of only ω . The function ‘string_gen.m’, determine_A_b.m and ‘determine_Ki_Kd_ineq.m’ together solve linear inequalities in $\{k_i, k_d\}$ space generated by evaluating the real part of $\nu(\omega, k_p, k_i, k_d)$ at the various values of ω . Again note that the real part of $\nu(\omega, k_p, k_i, k_d)$ for a fixed values of ω are functions of k_i and k_d .

2. GUI Based Performance Evaluation of Proportional Integral Derivative Controllers

Like the previous GUI’s developed for Proportional and Proportional Integral controllers the GUI developed for Proportional Integral Derivative controller can also determine the subset of stabilizing controllers that satisfy the given performance criteria like overshoot, rise-time, settling-time, gain margin and phase margin. For a selected value of k_p a 2d stabilizing set in $\{k_i, k_d\}$ space is generated. This set is then divided into finite points and is analyzed against each specified performance criteria. This function is performed by ‘main_determ_subset.m’. This function calls other sub functions like ‘stepspecs.m’ which determines the overshoot, rise-time, settling-time

of the closed loop transfer function.

Also the GUI can also display the performance for selected values of k_p, k_i, k_d . This is handled by another sub function 'view_perfm.m'. This enables the user to manually explore the set and in general determine the trend in increase or decrease of performance criteria. Moreover during this manual exploration the plot of the step response and the error signal are also displayed.

3. Illustrative Example

Consider the following example where for a given plant $p(s)$ the entire set of PID controllers is to be determined.

$$p(s) = \frac{4s^3 + 4s + 1}{s^8 + 13s^7 + 75s^6 + 249s^5 + 517s^4 + 583s^3 + 557s^2 + 155s + 25} \quad (5.17)$$

As seen in the figure (17), the coefficient of the Numerator and the Denominator are the inputs to the GUI. The entire set of $\{k_p, k_i, k_d\}$ values that stabilizes the system is determined by the algorithm discussed earlier in this section.

The steps involved in achieving this can be seen in the MATLAB[®] command window if desired. $N_e(\omega), N_o(\omega), D_e(\omega), D_o(\omega)$ are calculated and displayed as follows

$$N_e(\omega) = 1$$

$$N_o(\omega) = -4\omega^2 + 4$$

$$D_e(\omega) = \omega^8 - 75\omega^6 + 517\omega^4 - 557\omega^2 + 25$$

$$D_o(\omega) = -13\omega^6 + 249\omega^4 - 583\omega^2 + 155$$

Calculating $\nu(\omega, k_p, k_i, k_d) = \delta(\omega, k_p, k_i, k_d)N^*(s)$, where $\delta(\omega, k_p, k_i, k_d)$ is the characteristic polynomial

$$\nu(\omega, k_p, k_i, k_d) = -4\omega^12 + 304\omega^10 - 2355\omega^8$$

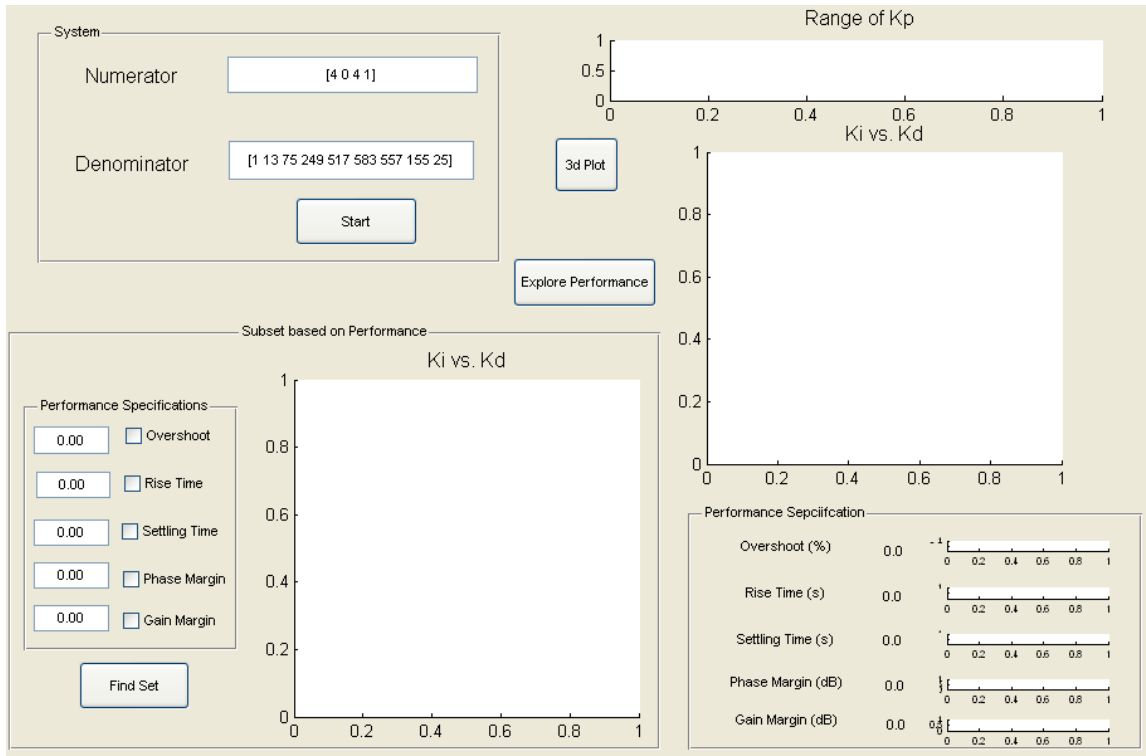


Fig. 17. GUI for determining the entire set of Proportional Integral Derivative controllers for a given transfer function

$$\begin{aligned}
 &+4047\omega^6 - 1745\omega^4 - 55\omega^2 \\
 &+(k_i - k_d\omega^2)(16\omega^6 - 32\omega^4 + 16\omega^2 + 1 \\
 &+j\omega[52\omega^{10} - 1047\omega^8 + 3253\omega^6 \\
 &-2435\omega^4 + 63\omega^2 + 25 \\
 &+k_p(16\omega^6 - 32\omega^4 + 16\omega^2 + 1)]
 \end{aligned}$$

where we define $p_1(\omega)$, $p_2(\omega)$, $q_1(\omega)$, $q_2(\omega)$

$$\begin{aligned}
 p_1(\omega) &= -4\omega^{12} + 304\omega^{10} - 2355\omega^8 \\
 &\quad +4047\omega^6 - 1745\omega^4 - 55\omega^2 \\
 p_2(\omega) &= (k_i - k_d\omega^2)(16\omega^6 - 32\omega^4 + 16\omega^2 + 1
 \end{aligned}$$

$$\begin{aligned}
q_1(\omega) &= \omega(52\omega^{10} - 1047\omega^8 + 3253\omega^6 \\
&\quad - 2435\omega^4 + 63\omega^2 + 25) \\
q_2(\omega) &= \omega(16\omega^6 - 32\omega^4 + 16\omega^2 + 1)
\end{aligned}$$

Determining roots of $q_1 + k_p q_2 = 0$ that are real, non negative and odd multiplicity for k_p value 99.9361

$$\omega_0 = 0 < \omega_1 = 0.7842 < \omega_2 = 0.9945 < \omega_3 = 2.1388 < \omega_4 = 3.7441 < \omega_5 = \infty \quad (5.18)$$

Since the degree of $\nu(\omega, k_p)$ given by $n + m$ is even and the signature is 10 the string that satisfy the signature condition is $1, -1, 1, -1, 1, -1$. In this particular example only one set of valid string was found to exist. There always exists a possibility for many valid strings.

Thus evaluating for the entire set of $\{k_i, k_d\}$ values by imposing the above sign conditions on the real part of $\nu(\omega, k_p, k_i, k_d)$ for a given value of k_p .

$$\begin{aligned}
p_1(\omega_0) + (k_i - k_d\omega_0^2)p_2(\omega_0) &> 0 \\
p_1(\omega_1) + (k_i - k_d\omega_1^2)p_2(\omega_1) &< 0 \\
p_1(\omega_2) + (k_i - k_d\omega_2^2)p_2(\omega_2) &> 0 \\
p_1(\omega_3) + (k_i - k_d\omega_3^2)p_2(\omega_3) &< 0 \\
p_1(\omega_4) + (k_i - k_d\omega_4^2)p_2(\omega_4) &< 0 \\
p_1(\omega_5) + (k_i - k_d\omega_5^2)p_2(\omega_5) &< 0
\end{aligned}$$

Solving the above linear inequalities, the 2d set in the $\{k_i, k_d\}$ space, for a given k_p that stabilize the system is obtained. For a fixed value of $k_p = 99.9361$, the 2d set in the $\{k_i, k_d\}$ space is as seen in the figure (18). Scanning the entire range of k_p and solving linear inequalities to obtain the corresponding values in the $\{k_i, k_d\}$

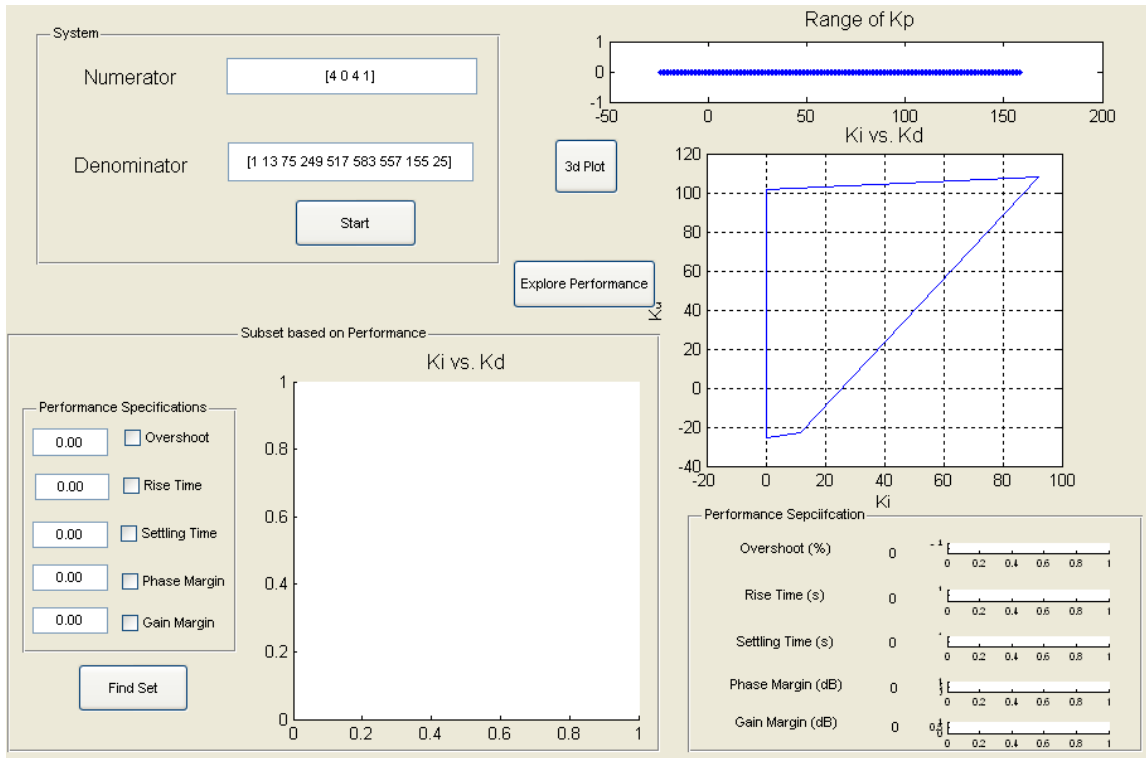


Fig. 18. GUI with the entire set of $\{k_i, k_d\}$ for a fixed value of k_p

space the entire stable set PID controllers are obtained as seen in figure (19). This is obtained in a separate pop up window when the '3d plot' button available on the GUI is enabled. Further in a 2d set in $\{k_i, k_d\}$ space for a fixed k_p , the GUI can determine the performances like Overshoot, Rise-time, Settling-time, Gain and Phase margin as discussed in the previous section and is displayed in the GUI as seen in figure (20). Also for a given specification a sub-set of stabilizing $\{k_p, k_i, k_d\}$ values can be determined. As seen in the figure (21), the performance specification of Gain Margin greater than $1.2dB$ and the Overshoot less than 30% , the subset satisfying the performance criteria in red and the entire stabilizing set in blue are displayed.

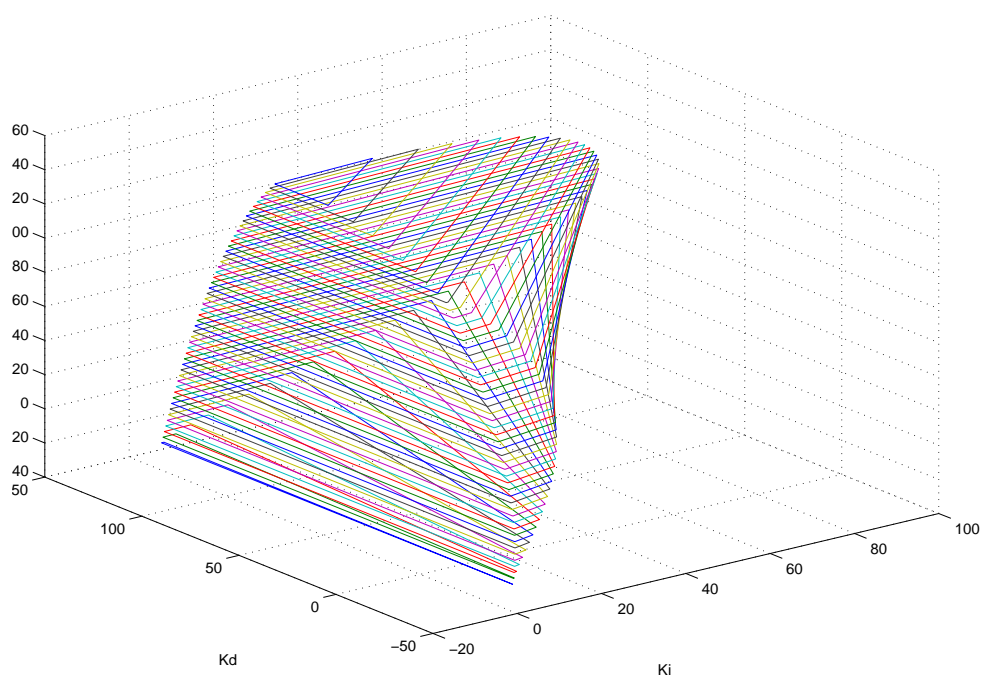


Fig. 19. GUI with the entire set of $\{k_p, k_i, k_d\}$ displayed for the given transfer function

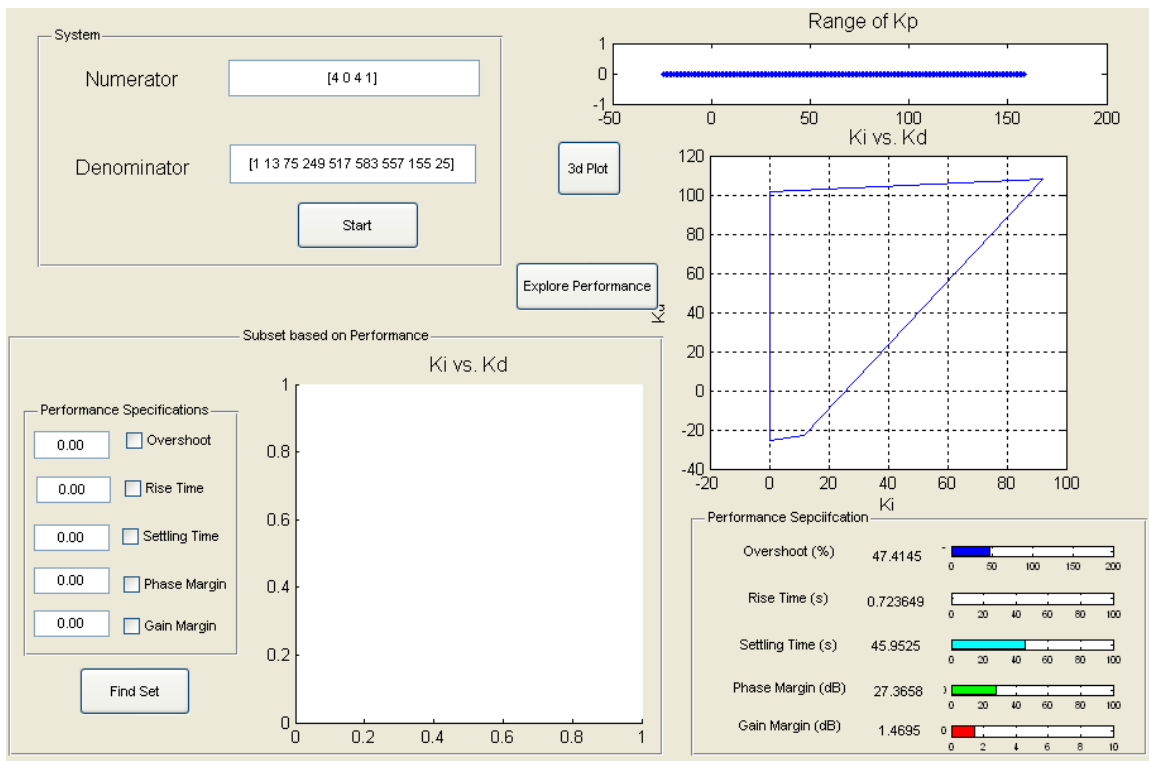


Fig. 20. GUI with the performance displayed for a selected value of PID controller

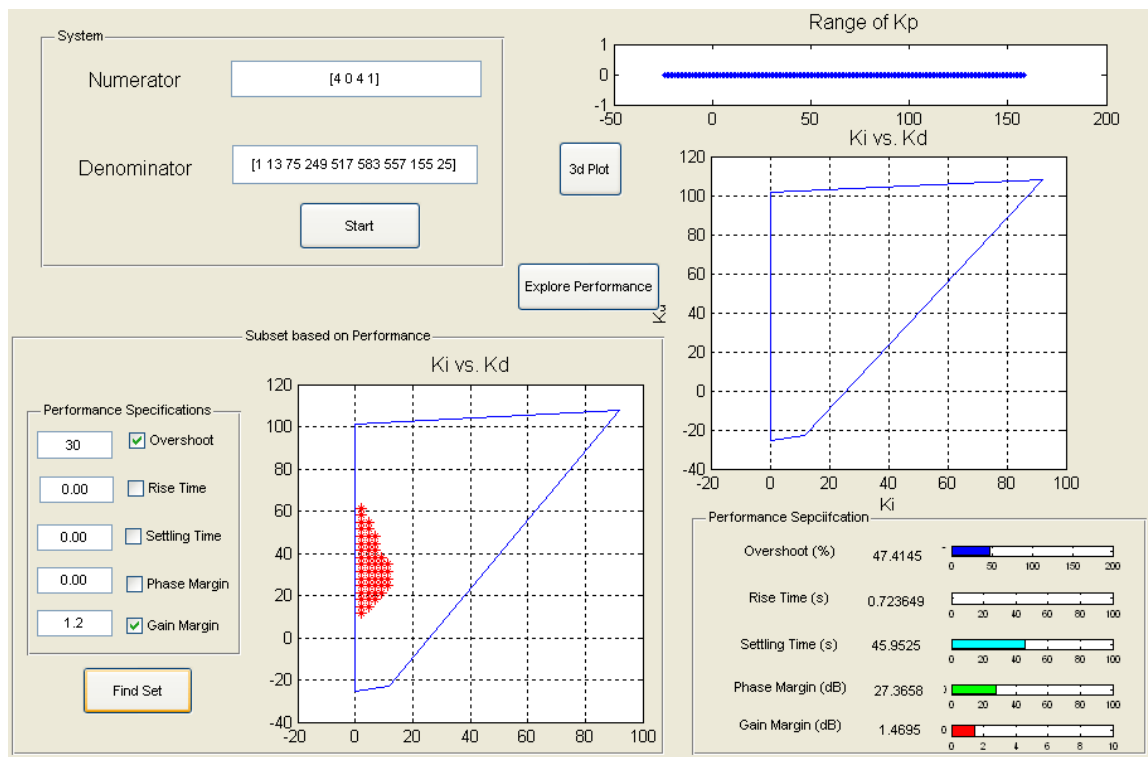


Fig. 21. GUI with a subset of stabilizing values in $\{k_i, k_d\}$ space satisfying the condition of $GM > 1.2dB$ and $Overshoot < 30\%$

CHAPTER VI

A MODEL FREE APPROACH IN STABILIZING A LTI SYSTEM USING A PID
CONTROLLER

In the previous chapters the algorithms discussed for determining the set of Proportional, Proportional Integral, Proportional Integral Derivative controllers required a Mathematical model of the system. In this chapter introduced is the model free approach where only the frequency response of the system to be stabilized is required. The frequency response can be easily obtained by performing experiments and does not require any modeling of the plant in terms of transfer function or state space. In the case when the plant is unstable, it could be stabilized by a known feedback compensator, and a frequency response then obtained. From this frequency response for the stable setup the frequency response for the unstable plant can be determined by dividing out the known compensator. Further in this chapter is introduced the GUI built in MATLAB[®], which takes the frequency response of the system as the input and constructs a 3d set in $\{k_p, k_i, k_d\}$ space that will stabilize the system. Also is introduced the algorithm to determine the subset of controllers that satisfy the specification on the gain and phase margin. The GUI also incorporates the performance based evaluation where the gain and the phase margin along with the frequency response of the system are the inputs and the output being the subset of stable controllers that meet the performance requirements.

A. Theory and Algorithm

The theory and the algorithm to compute the entire set of stabilizing Proportional Integral Derivative controllers given just the frequency response of the system is derived by Bhattacharyya and others and is introduced in [4]. An overview of the

same is now presented.

Consider the plant $p(s) = \frac{N(s)}{D(s)}$ that is to be stabilized by a PID controller $c(s)$. It is to be noted that the actual transfer function $p(s)$ is not known, but the frequency response of the same is available. Also let the PID controller $c(s)$ be of the form

$$c(s) = \frac{k_d s^2 + k_p s + k_i}{s(1 + sT)} \quad (6.1)$$

Equation (6.1) is a modified PID controller where T is a very small value. $(1 + sT)$ is introduced to avoid a pure differentiator.

The characteristic polynomial $\delta(s)$ is given as follows

$$\delta(s, k_p, k_i, k_d) = s(1 + sT)D(s) + (k_d s^2 + k_p s + k_i)N(s) \quad (6.2)$$

Let n be the degree of the denominator of the plant under consideration. The objective is to calculate the entire set of $\{k_p, k_i, k_d\}$ values such that the characteristic polynomial is Hurwitz stable. That is all the $(n + 2)$ roots of the equation $(\delta(s, k_p, k_i, k_d) = 0)$ are on the left half of the complex plane. Thus as seen in Chapter II, if the polynomial is Hurwitz stable, the signature $(\sigma(p))$ should be equal to $(n + 2)$

Similarly as seen with a polynomial, the concept of signature is now extended to a rational function

$$R(s) = \frac{A(s)}{B(s)} \quad (6.3)$$

Then if $z_R^+, p_R^+, z_R^-, p_R^-$ denote the number of open right half zeros and poles and left half zeros and poles respectively, the signature of $R(s)$ is given by

$$\sigma(R) = z_R^- - z_R^+ - (p_R^- - p_R^+) \quad (6.4)$$

Defining $\pi(s)$ as follows

$$\pi(s, k_p, k_i, k_d) = \frac{\delta(s, k_p, k_i, k_d)}{N(s)}$$

$$= s(1 + sT) + (k_d s^2 + k_p s + k_i)p(s) \quad (6.5)$$

As seen in the equation (6.5) if the equation is decomposed into real and imaginary parts, both the parts would be still dependent on k_p, k_i, k_d . As seen in the previous chapters when dealing with the Proportional, Proportional Integral and Proportional Integral Derivative controllers for a model based controller in this case too the stabilization problem is converted to a root counting problem by multiplying $p(-s)$.

Defining $\nu(s)$ as follows

$$\begin{aligned} \nu(s, k_p, k_i, k_d) &= \pi(s, k_p, k_i, k_d)p(-s) \\ &= s(1 + sT)p(-s) + (k_d s^2 + k_p s + k_i)p(s)p(-s) \end{aligned} \quad (6.6)$$

For the rational function $\nu(s, k_p, k_i, k_d)$ to be stable, the signature condition required is as follows

$$\sigma(\nu(s, k_p, k_i, k_d)) = n - m + 2z^+ + 2 \quad (6.7)$$

In equation (6.7) n, m, z^+ are degree of the numerator, degree of the denominator and the total number of right half zeros of the system under consideration respectively. For a detailed proof on evaluation of the signature for $\nu(s)$ refer [4].

Further in the calculation of the signature of $\nu(s, k_p, k_i, k_d)$, $n - m$ is calculated by evaluating the high frequency slope of the magnitude of the plot of $p(j\omega)$. Also z^+ can be evaluated from the phase plot as follows for a stable system

$$\Delta_0^\infty[\phi(\omega)] = -[(n - m) + 2z^+] \frac{\pi}{2} \quad (6.8)$$

In case of an unstable system, required is a stabilizing controller $c(s)$ and the frequency response of the corresponding stable closed-loop system $g(j\omega)$. From the closed loop frequency response, the frequency response of the original system $p(j\omega)$ is calculated

as follows

$$p(j\omega) = \frac{g(j\omega)}{c(j\omega)(1 - p(j\omega))} \quad (6.9)$$

The high frequency slope r_p of the magnitude of $p(j\omega)$ is calculated. Also the controller's relative degree r_c and the total number of zeros in the right half of the complex plane z_c^+ is calculated. z^+ is then calculated as follows

$$z^+ = \frac{1}{2}[-r_p - r_c - 2z_c^+ - \sigma(g)]$$

where

$$\sigma(g) = \frac{2}{\pi} \Delta_0^\infty \angle p(j\omega)$$

Thus the $\sigma(\nu(s, k_p, k_i, k_d))$ for both an unstable and a stable system can be determined.

In equation (6.6) replacing s by $j\omega$

$$\begin{aligned} \nu(\omega, k_p, k_i, k_d) &= j\omega(1 + j\omega T)p(-j\omega) + (k_i + j\omega k_p - \omega^2 k_d)p(j\omega)p(-j\omega) \\ &= (k_i - k_d \omega^2)|p(j\omega)|^2 - \omega^2 T p_r(\omega) + \omega p_i(\omega) \\ &\quad + j\omega(k_p |p(j\omega)|^2 + p_r(\omega) + \omega T p_i(\omega)) \end{aligned} \quad (6.10)$$

In equation (6.10) it is now observed that the real part of $\nu(j\omega)$ is dependent on on k_i, k_d while the imaginary part is dependent only on k_p . Thus defining

$$\nu_r(\omega, k_i, k_d) = (k_i - k_d \omega^2)|p(j\omega)|^2 - \omega^2 T p_r(\omega) + \omega p_i(\omega) \quad (6.11)$$

$$\nu_i(\omega, k_p) = k_p |p(j\omega)|^2 + p_r(\omega) + \omega T p_i(\omega) \quad (6.12)$$

$\nu(\omega, k_p, k_i, k_d)$ is now defined as follows

$$\nu(\omega, k_p, k_i, k_d) = \nu_r(\omega, k_i, k_d) + j\omega \nu_i(\omega, k_p) \quad (6.13)$$

Applying the Generalized Hermite-Biehler theorem derived in [21], evaluating $\nu_i(j\omega, k_p) = 0$. For a fixed value of k_p let the real, non negative roots with odd multiplicity be as follows.

$$\omega_0 = 0 < \omega_1 < \omega_2 < \omega_3 \cdots < \omega_{l-1} < \infty \quad (6.14)$$

These roots can be evaluated using the only information available, the frequency response of the system as follows.

$$\begin{aligned} k_p &= -\frac{p_r(\omega) + \omega T p_i(\omega)}{|p(j\omega)|^2} \\ &= -\frac{\cos\phi(\omega) + \omega T \sin\phi(\omega)}{|p(j\omega)|} \end{aligned} \quad (6.15)$$

Calculating valid strings $I = [i_0, i_1, i_2, \dots, i_l]$ depending on $n - m$ being even or odd.

In the case when $n - m$ is even the strings are evaluated as follows

$$\sigma(\nu(s)) = [i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1}2i_{l-1} + (-1)^l i_l] \cdot (-1)^{l-1} \text{sgn}[q(\infty)]$$

where

$$i_n = \text{sgn}(q_1(\omega_n)) \quad n \text{ goes from } 0 \text{ to } l$$

Also in the above equation ω_0 is defined as 0 and ω_l is defined as ∞ . In the case when $n - m$ is odd ω_l and i_l does not exist and hence $\sigma(\nu(s))$ reduces to

$$\sigma(\nu(s)) = [i_0 - 2i_1 + 2i_2 + \cdots + (-1)^{l-1}2i_{l-1}] \cdot (-1)^{l-1} \text{sgn}[q(\infty)]$$

where

$$i_n = \text{sgn}(q_1(\omega_n)) \quad n \text{ goes from } 0 \text{ to } l - 1$$

For a fixed value of k_p , the entire set of $\{k_i, k_d\}$ controllers stabilizing the system can be calculated by solving inequalities generated by evaluating the real part of $\nu(\omega, k_i, k_d)$ at the values of ω determined. The sign of the inequalities are decided by

the valid strings I .

$$\begin{aligned} \nu_r(j\omega_t, k_i, k_d)i_t &> 0 \\ k_i - k_d\omega_t^2 + \frac{\omega_t \sin\phi(\omega_t) - \omega_t^2 T \cos\phi(\omega_t)}{|p(j\omega_t)|} i_t &> 0 \end{aligned}$$

By sweeping over a range of k_p and solving the corresponding linear inequalities to obtain the set of $\{k_i, k_d\}$ values, the entire set of stabilizing controllers in $\{k_p, k_i, k_d\}$ space can be generated.

Moreover the values over which k_p has to be swept can also be determined. The k_p values to be selected have to be such that on solving $\nu_i(\omega, k_p)$ for values of ω they generate a minimum of k real, non-negative roots with odd multiplicity where k is given by

$$\begin{aligned} k &= \frac{n - m + 2z^+ + 2}{2} - 1 && \text{if } n - m \text{ is even} \\ k &= \frac{n - m + 2z^+ + 3}{2} - 1 && \text{if } n - m \text{ is odd} \end{aligned}$$

Consider an example as follows:

Example VI.1 Consider a plant $p(s)$ whose frequency response $p(j\omega)$ is obtained and is as seen in the figure (22) The objective is to determine the entire set of PID controllers $\{k_p, k_i, k_d\}$ that will stabilize the given system. Calculating the entire set of PID controller for the system using the algorithm discussed. The signature for the rational function $\nu(\omega, k_p, k_i, k_d)$ is as follows

$$\nu(\omega, k_p, k_i, k_d) = n - m + 2z^+ + 2 \quad (6.16)$$

Where $n - m$ is calculated from the high frequency slope of the magnitude of $p(j\omega)$. From figure (22), the high frequency slope is calculated to $-60dB/decade$ and hence $n - m$ is 3.

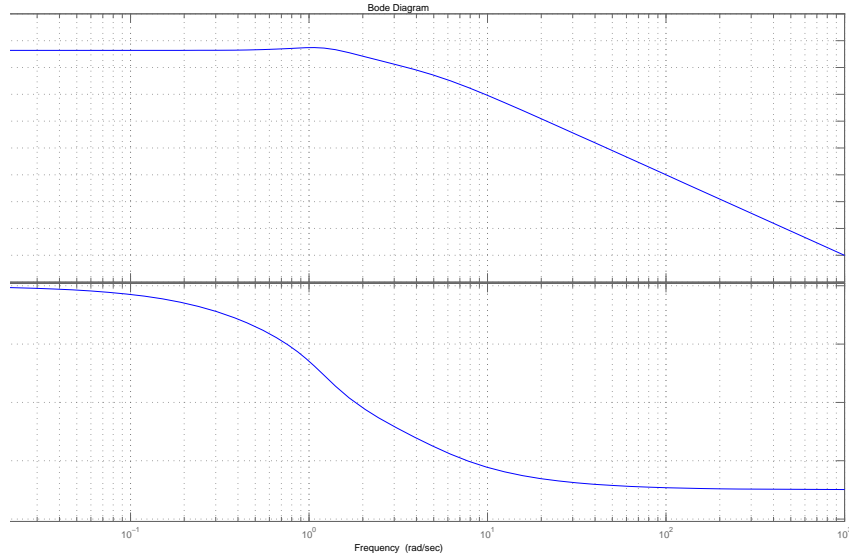


Fig. 22. Bode plot of the plant $p(s)$

As the system under consideration is stable, p^+ is zero and hence z^+ can be calculated as follows

$$\Delta_0^\infty[\phi(\omega)] = -[(n - m) + 2z^+] \frac{\pi}{2} \quad (6.17)$$

The total change of phase as seen from the phase plot in figure (22) is $-\frac{6\pi}{2}$ and hence z^+ is 2.

The signature of the $\nu(j\omega, k_p, k_i, k_d)$ can now be determined and is given by

$$\sigma(\nu(j\omega, k_p, k_i, k_d)) = 3 + 2(2) + 2 = 9 \quad (6.18)$$

Also as $n - m$ is odd, the valid strings can be generated as follows

$$[i_0 - 2i_1 + 2i_2 + \dots + (-1)^{l-1} 2i_{l-1}] \cdot (-1)^{l-1} j = 9 \quad (6.19)$$

where,

$$j = \text{sgn}[\nu_i(0, k_p)] \quad \text{where } k_p \text{ is fixed} \quad (6.20)$$

The equation (6.19) can be satisfied only if $l \geq 4$. Figure (23) the plot of $g(\omega)$ where $g(\omega)$ is defined as follows

$$k_p = -\frac{\cos\phi(\omega) + \omega T \sin\phi(\omega)}{|p(j\omega)|} = g(\omega) \quad (6.21)$$

As seen in the figure (23) the maximum number of real, non negative roots are 4 and including ω_0 defined earlier is 5, which is also the minimum number of required roots $(l + 1)$ to satisfy the signature condition. The range of k_p that satisfy this condition is from -23 to 19 . From the feasible values of k_p selecting $k_p = 10$. The values of ω for which $g(\omega)$ is 10 are as follows

$$\omega_1 = 0.4536 < \omega_2 = 1.0163 < \omega_3 = 3.5282 < \omega_4 = 109.4055 \quad (6.22)$$

The corresponding value of valid strings is then given by $I = 1, -1, 1, -1, 1$.

Generating linear inequalities in the $\{k_i, k_d\}$ space by evaluating $\nu_r(\omega, k_i, k_d)$ at the values of ω the entire set of stabilizing controllers for a particular value of k_p can be obtained.

The linear inequalities generated for $k_p = 10$

$$\begin{aligned} k_i &> 0 \\ k_i - 0.2057k_d &< 9.1321 \\ k_i - 1.0328k_d &> -15.3594 \\ k_i - 12.4479k_d &< 350.3452 \\ k_i - 11969.5569k_d &> -144209721.1202 \end{aligned}$$

Solving the above linear inequalities, the entire set of controllers in the $\{k_i, k_d\}$ space for a fixed value of $k_p = 10$ is obtained and is as seen in the figure (24). Also by varying k_p similar linear inequalities in the $\{k_i, k_d\}$ space can be generated. The plot

of the entire set of PID controllers that stabilize the given system is as seen in the figure (25)

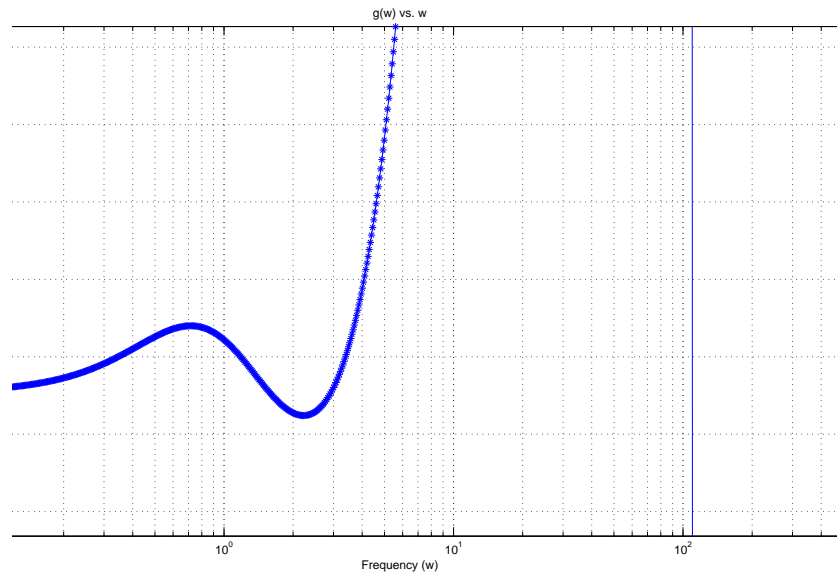


Fig. 23. Plot of $g(\omega)$ against varying values of frequency ω

B. GUI for Model Free Approach for Calculating the Entire Set of Proportional Integral Derivative Controllers for a LTI System

As seen in the previous section the entire set of PID values that can stabilize the system can be calculated just using the frequency response, which in turn can be obtained by experimental evaluation. A GUI has been developed in MATLAB[®] to bring this theory closer to the user. The only inputs to the GUI is the Bode plot of the system and the GUI will generate a 3d set in the $\{k_p, k_i, k_d\}$ space. The GUI developed for model free approach though has got relatively less number of features than the Model based approach. The GUI does the basic calculation of the entire

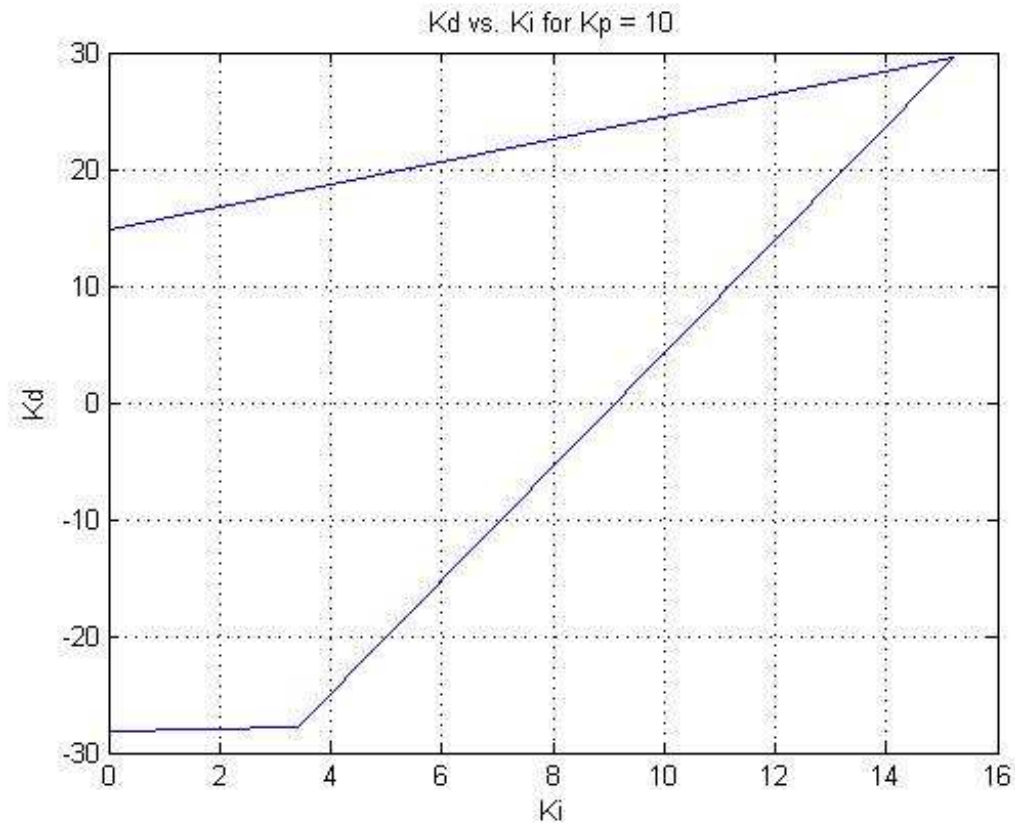


Fig. 24. Stabilizing values in $\{k_i, k_d\}$ space for a fixed value of k_p

stabilizing set for a given system. Also with respect to determining the subset of stabilizing controllers for the given performance specification, the GUI can handle only Gain and Phase margin. The speed of calculating the subset is much higher than the model based approach though.

1. GUI Based Calculation of Proportional Integral Derivative Values

The input to the GUI is the bode plot of the system under consideration. The user is required to store the input as a '.txt' file in the same directory in which the GUI is present. Also the file has to be named as 'bode_data.txt'. The requirement on

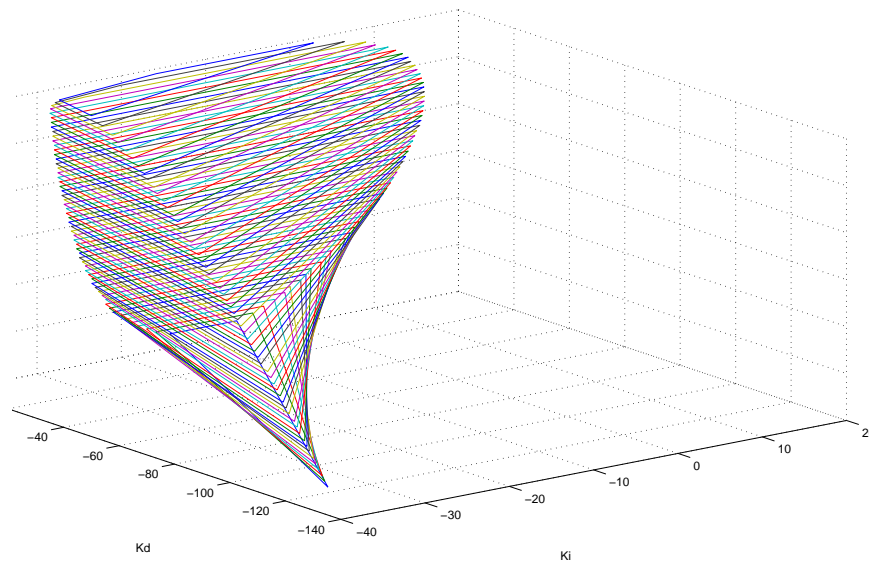


Fig. 25. The stabilizing set of $\{k_p, k_i, k_d\}$ values for VI.1

the text file is that it must contain three columns of data having the magnitude, the phase in degrees and the corresponding frequency at which it was observed. This is the only input required for a stable system. For an unstable system the number of poles in the right half of the complex plane is also required. An alternative to this approach is to stabilize the system with one known controller (not necessarily PID) and to follow the algorithm as discussed in the previous section. For unstable system, the GUI handles only the first method where the number of poles in the right half of the complex plane must be known.

The input and the outputs to the GUI is controlled by the program ‘main_data_PID.m’. To initiate the GUI it is required to run this program. This program calls other programs depending upon the interaction between the user and the GUI. The base program that finds the stabilizing set is ‘main_data_PID_computation.m’. The GUI does not display the steps in achieving the final set of PID controllers that will stabilize

the system. The user if requires can see all the steps in the MATLAB[®] command window.

The program ‘determine_high_f_slope.m’ calculates the high frequency slope which as seen in the previous section is $n-m$. ‘determine_net_change_phase.m’ calculates the total net change in phase of the bode plot, which is used in calculating z^+ . Both $n-m$ and z^+ is required to calculate the signature. Also ‘determine_min_roots_odd_mult.m’ determines the total number of minimum roots required, which helps in calculating the range of feasible k_p values.

The function ‘determine_all_kp.m’ is a relatively complex function that calculates all the values of k_p and the corresponding values of ω at which the $\nu_i(\omega, k_i, k_d)$ reduce to zero. The range of k_p values scanned is from -5000 to 5000. Due to limitations of programming some times erroneous results may occur. This occurs when there is a sudden change in k_p values as the function $g(\omega)$ progress. During the experimental evaluation of the program the maximum change observed was from 1000 to -1000 and hence the value of k_p to be scanned ranges from -5000 to 5000 . Though there can be no guarantee of k_p exceeding this range, it can be mitigated by increasing the range of k_p values scanned. A plot of $g(\omega)$ is also readily available and the user can intervene the running of the program and increase the k_p range if desired.

The GUI then transfers control to the user who selects a desired value of k_p corresponding to which the entire set of controllers in the $\{k_i, k_d\}$ space is generated. This is done by ‘string_gen_MF.m’, ‘determine_A_B_c.m’ and ‘determine_Ki_Kd_ineq.m’, which together generate the set of linear inequalities in the $\{k_i, k_d\}$ space where the inequality sign is decided by the function ‘string_gen_MF.m’

2. Illustrative Example

Consider the following example where for a frequency response of the plant the entire set of PID controllers is to be determined. The frequency response obtained for plant $p(s)$ is given by $p(j\omega)$. As seen from the figure (26). The obtained frequency response and the number of roots of the numerator of the transfer function which are on the right half of the complex plane z^+ are the only inputs to the GUI.

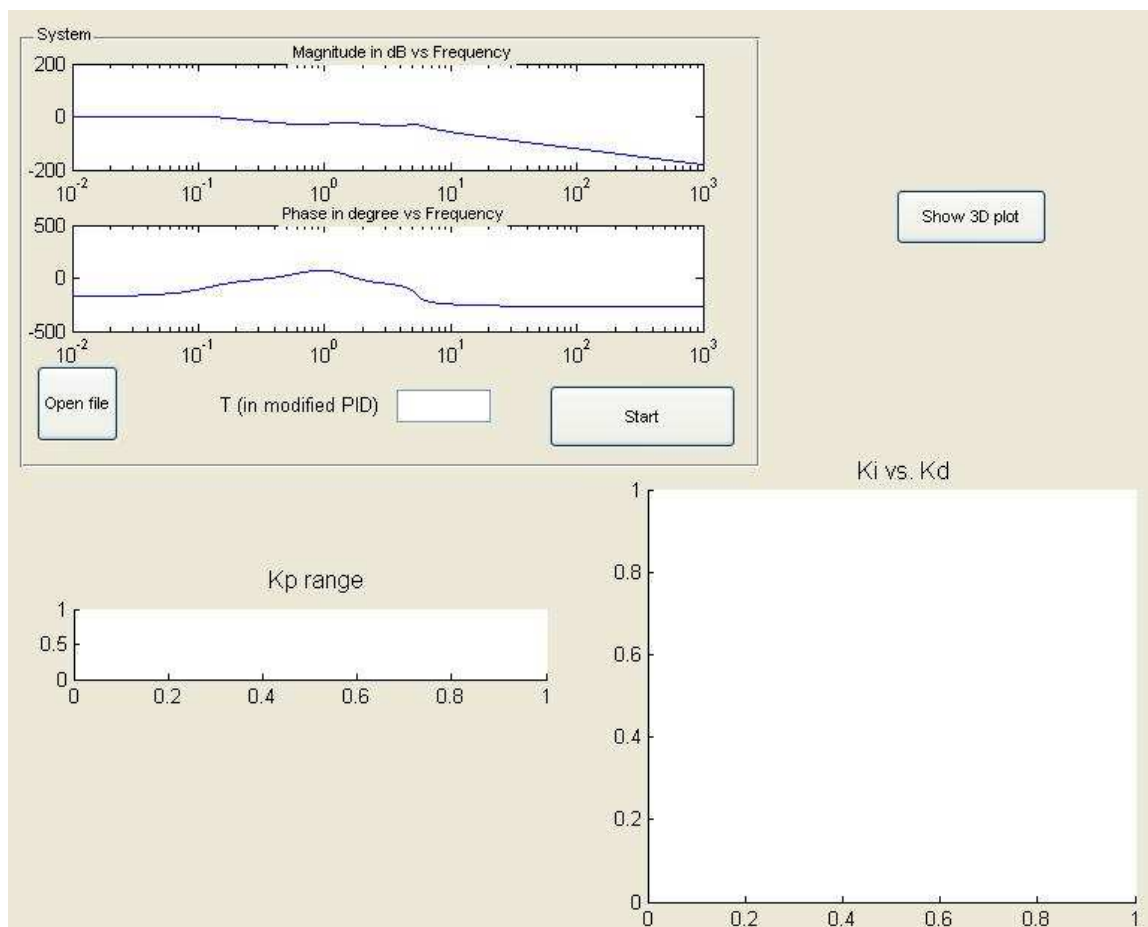


Fig. 26. GUI for determining the entire set of Proportional Integral Derivative controllers given the frequency response of the system

The steps involved in obtaining the entire set of PID controllers can be viewed in

the MATLAB[®] command window if desired. The signature for the rational function $\nu(\omega, k_p, k_i, k_d)$ for this example is computed using the high frequency slope $n-m$ of the magnitude of the bode plot. The displayed output is $n-m = 3$ and $\nu(\omega, k_p, k_i, k_d) = 7$.

The plot of $g(\omega)$ which is required in the computation of set of k_p values is also an output during the calculations of the GUI. This plot of $g(\omega)$ can be used to verify the calculations done by the GUI. The plot of $g(\omega)$ is displayed in figure (27). The

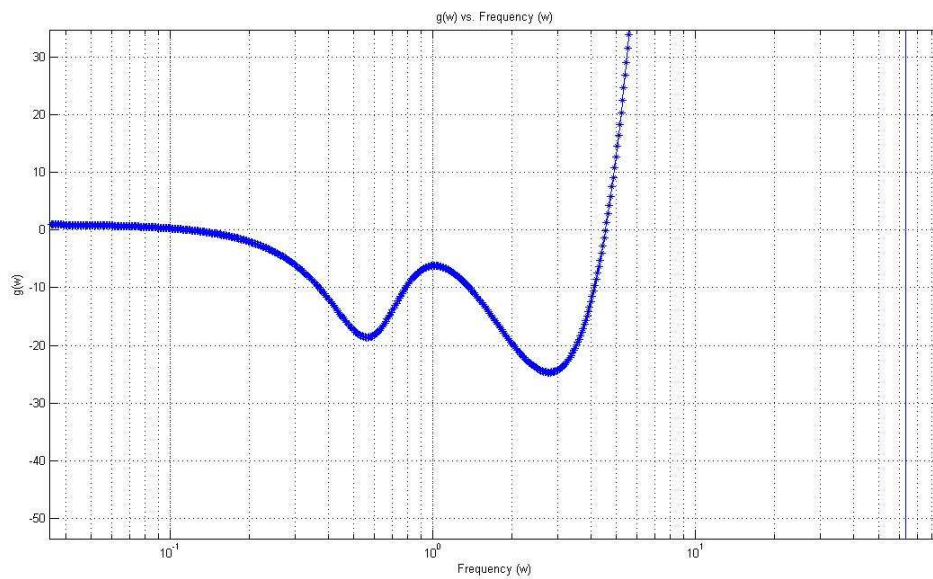


Fig. 27. Plot of $g(\omega)$ against varying frequency

range of k_p generated is displayed in the figure inbuilt in the GUI as seen in figure(28) At this point the control of the program is transferred to the user, who is required to select the desired k_p value against which the set in $\{k_i, k_d\}$ space is displayed. For k_p value of -5 the output of the GUI is as seen in the figure(29) This was generated by generating linear inequalities (displayed below) in the $\{k_i, k_d\}$ space

$$k_i < 0$$

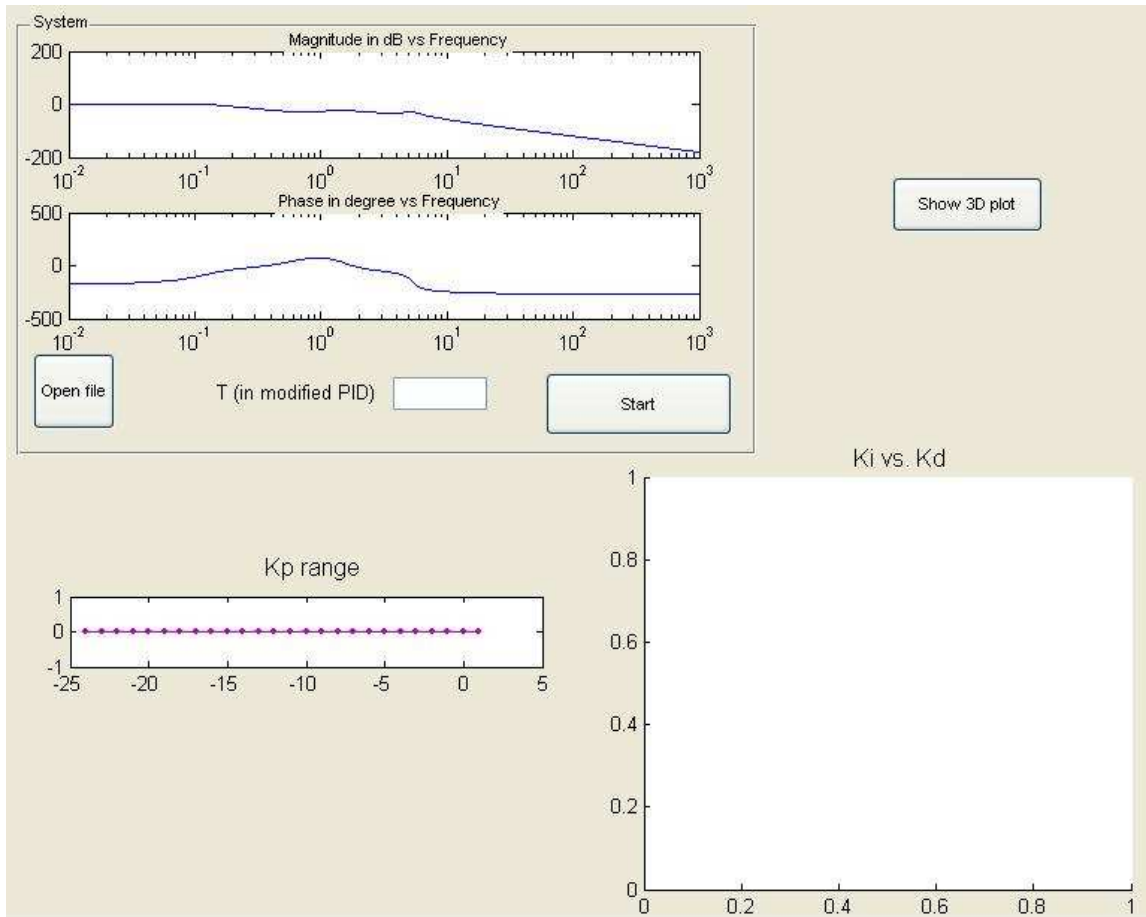


Fig. 28. The GUI with the range of feasible k_p values displayed for the given bode plot

$$\begin{aligned}
 k_i - 0.0713k_d &> 0.41223 \\
 k_i - 18.8484k_d &< 163.9076 \\
 k_i - 3959.1103k_d &> -15574817.7108
 \end{aligned}$$

(6.23)

Also the GUI can generate the entire 3d PID set. This set is generated by computing the set of k_i, k_d values by varying k_p over the generated feasible set. The generated 3d PID set is displayed in a separate pop-up window and is seen in figure (30)

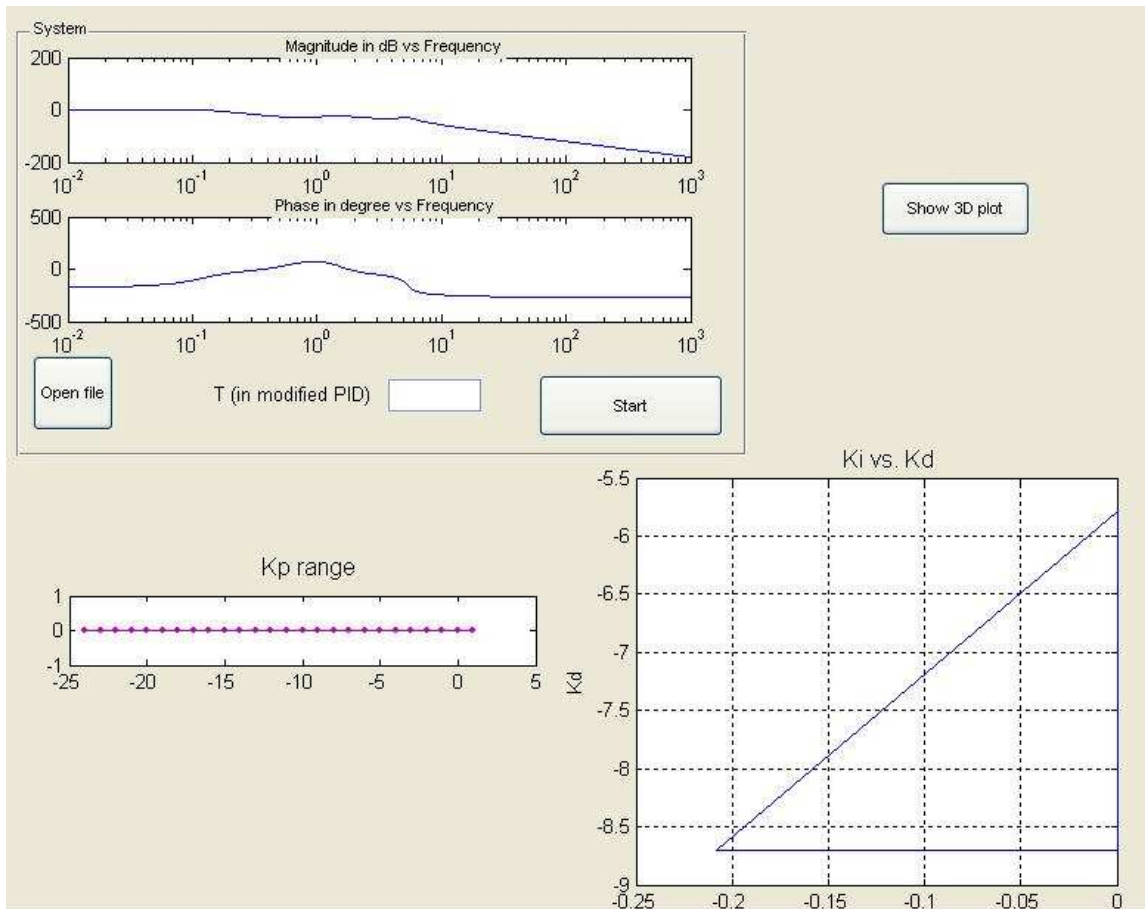


Fig. 29. The GUI with the set in $\{k_i, k_d\}$ space for a value of $k_p = -5$

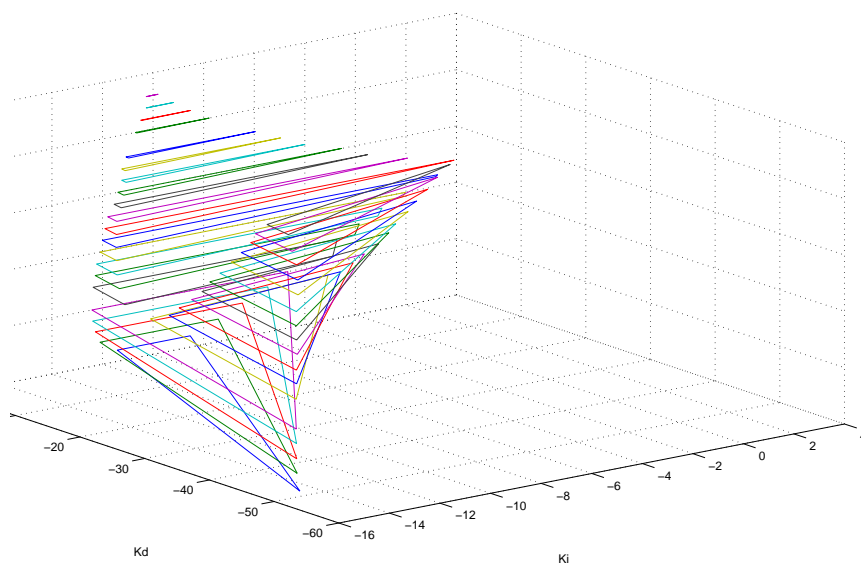


Fig. 30. Complete set of PID values for the given frequency response

CHAPTER VII

DISCUSSION AND CONCLUSION

In the previous chapters an overview of the Theory and Algorithm developed by Bhattacharyya and others [3], [4], [25] and the Graphical User Interface for a Proportional, Proportional Integral, and Proportional Integral Derivative controller were introduced. The theory developed to determine the entire set of controllers with or without the model, was a major leap in the control systems field and the development of the GUI is an important step in making this theory practicable and simplifying it to an extent of entering only the transfer function or the bode plot to the GUI.

The GUI also incorporates the concept of scanning through the entire set of stable controllers to compute the subset of controllers satisfying the performance specification.

The GUI helps in carrying out complex calculation and minimized the user involvement. Also the GUI enables the user to have no or minimal knowledge of the theory behind the computation of the entire set of PID controllers making this powerful theory more widely available.

In the following section a case study is reviewed where the GUI could have been possibly used and possible future work in this direction.

A. Case Study

The High speed Flywheel Energy Storage System (FESS) is controlled by a DC bus voltage regulator. The regulation of this DC bus voltage and also the current and the speed of High speed Flywheel Energy Storage System is done by a PID controller. In [5] a detailed derivation of the Mathematical model of the speed controlled FESS is introduced. Also the paper derives the entire set of PID controllers for the system.

The Authors also mentions of extending the work in [5] by computing subsets of stabilizing sets of PID controllers based on parameters like Gain and Phase margin, Overshoot, Rise Time and Settling Time.

All of the above can now be done by the GUI developed in MATLAB[®] with a click of a button. The only inputs to the GUI in case for the above example is the Mathematical Model of the speed controlled FESS. Alternatively, with the GUI for Model free approach available, the frequency response of the system under consideration can be used. Both the GUIs help in calculating the entire set of PID controllers. Moreover the GUI can even compute the subset of the stabilizing PID controllers given the performance specification required. With respect to the work done on High speed Flywheel Energy Storage System in [5], the use of GUI not only simplifies the calculation of the stabilizing set of PID controllers, but also helps in calculating a subset of controllers based on the performance specification which in the paper is left for future work.

B. Future Work

GUI for computing the entire set of PID controllers for a discrete time system has also been developed and is introduced in [26]. The algorithm for the GUI is discussed in [27], [28]. The integration of all the available GUI like GUIs computing P, PI, PID for Continuous time system and the GUIs computing P, PD and PID for Discrete time system should be the next step. All the above mentioned GUIs do not take into account time delay. The theory for calculating the set of P, PI, PID controllers in case of delay is now available [29]. The GUI handling time delays should also be developed and integrated with the other GUI. Moreover the theory discussed in this thesis can be extended to any three term controller. Given the structure of the

controller a GUI can be developed to find the entire stabilizing set in the 3d space. The Final integrated GUI will thus give the user the ability to work on any three term controller, a continuous or discrete time system and a delay or delay free case. The GUI then should be used in case studies like the High speed Flywheel Energy Storage System to show the users the apparent advantage and also to increase the reliability of the GUI.

REFERENCES

- [1] J.G. Ziegler and N.B. Nichols, “Optimum settings for automatic controllers,” *Transactions of the American Society of Mechanical Engineers*, vol. 64, pp. 759–768, 1942.
- [2] M. Morari and E. Zafiriou, *Robust Process Control*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [3] Ming-Tzu Ho, A. Datta, and S.P. Bhattacharyya, “A linear programming characterization of all stabilizing PID controllers,” in *Proc. of the American Control Conference*, Albuquerque, NM, June 1997, pp. 3922–3928.
- [4] L.H. Keel and S.P. Bhattacharyya, “PID controller synthesis: Free of analytical models,” in *Proc. of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.
- [5] S. Talebi and H.A. Toliyat, “A novel algorithm for designing PID controllers for high-speed flywheels,” Unpublished paper, Texas A&M University, College Station, TX, 2007.
- [6] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, “State space solutions to standard h_2 and h_∞ control problems,” *IEEE Transactions on Automatic Control*, vol. AC-34, pp. 831–847, August 1989.
- [7] M.A. Dahleh and I.J. Diaz-Bobillo, *Control of Uncertain Systems: A Linear Programming Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [8] D.C. Youla, H.A. Jabr, and J.J. Bongiorno, “Modern wiener-hopf design of optimal controllers - part 2: The multivariable case,” *IEEE Transactions on Automatic Control*, vol. AC-21, pp. 319–338, 1976.

- [9] L.H. Keel and S.P. Bhattacharyya, “Robust, fragile or optimal ?,” Technical report, Department of Electrical Engineering, Texas A&M University, College Station, TX, September 1995.
- [10] S. Hara, D. Shiokata, and T. Iwasaki, “Fixed order controller design via generalized KYP lemma,” in *Proc. of IEEE Conference on Control Applications*, Taipei, Taiwan, September 2004.
- [11] D. Henrion, A. Hansson, and R. Wallin, “Reduced LMIs for fixed order polynomial controller design,” in *Proc. of the Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, July 2004.
- [12] E.N Gryazina and B.T Polyak, “On the root invariant regions structure for linear systems,” in *Proc. of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.
- [13] Yu. I. Neimark, *Stability of Linearized Systems*, Leningrad Aeronautical Engineering Academy, Leningrad, Russia, 1949.
- [14] D.D Siljak, *Nonlinear Systems: The Parameter Analysis and Design*, Wiley, New York, 1969.
- [15] K. Astrom and T. Hagglund, *PID Controllers: Theory, Design, and Tuning*, Instrument Society of America, Research Triangle Park, NC, 1995.
- [16] J. Quevedo and T. Escobet, “Digital control: Past, present and future of PID control,” in *Proc. IFAC Workshop*, Terrassa, Spain, April 2000.
- [17] Yun Li, Kiam Heong Ang, and Gregory C.Y Chong, “Patents software and hardware for PID control,” *IEEE Control Systems Magazine*, pp. 42–56, February 2006.

- [18] Yun Li, Kiam Heong Ang, and Gregory C.Y Chong, “PID control systems analysis and design,” *IEEE Control Systems Magazine*, pp. 32–41, February 2006.
- [19] L.H. Keel and S.P. Bhattacharyya, “A generalization of mikhailov’s criterion with applications,” in *Proc. of the American Control Conference*, Chicago, IL, June 2000, vol. 6, pp. 4311–4315.
- [20] G.J. Silva, A. Datta, and S.P. Bhattacharyya, *PID Controllers for Time-Delay Systems*, chapter 2, pp. 24–26, Birkhäuser, Boston, MA, 2005.
- [21] G.J. Silva, A. Datta, and S.P. Bhattacharyya, *PID Controllers for Time-Delay Systems*, chapter 2, pp. 29–36, Birkhäuser, Boston, MA, 2005.
- [22] G.J. Silva, A. Datta, and S.P. Bhattacharyya, *PID Controllers for Time-Delay Systems*, chapter 3, p. 41, Birkhäuser, Boston, MA, 2005.
- [23] G.J. Silva, A. Datta, and S.P. Bhattacharyya, *PID Controllers for Time-Delay Systems*, chapter 3, p. 45, Birkhäuser, Boston, MA, 2005.
- [24] Indu Ramamurthi, “A versatile simulation tool for virtual implementation of Proportional Integral and Derivative (PID) controllers,” M.S. thesis, Texas A&M University, College Station, TX, May 2007.
- [25] M.T. Ho, A. Datta, and S.P. Bhattacharyya, “Control systems design using low order controllers: Constant gain, PI and PID,” in *Proc. of the American Control Conference*, Albuquerque, NM, June 1997, pp. 571–578.
- [26] Sandipan Mitra, “Computer aided synthesis and design of PID controllers,” M.S. thesis, Texas A&M University, College Station, TX, December 2007.

- [27] Xu Hao, A. Datta, and S.P. Bhattacharyya, “Computation of all stabilizing PID gains for digital control systems,” *Automatic Control, IEEE Transactions on*, vol. 46, pp. 647–652, April 2001.
- [28] L.H. Keel and S.P. Bhattacharyya, “Data driven synthesis of three term digital controllers,” in *Proc. of the American Control Conference*, Minneapolis, MN, June 2006, pp. 262–267.
- [29] G.J. Silva, A. Datta, and S.P. Bhattacharyya, *PID Controllers for Time-Delay Systems*, chapter 5–8, pp. 77–189, Birkhäuser, Boston, MA, 2005.

VITA

Name: Bharat Narasimhan

Address: 9/82 Chaitanya, Garodia Nagar, Ghatkopar (East),
Mumbai - 400077, India

Education: **Master of Science**
Electrical Engineering, Dec 2007
Texas A&M University
College Station, Texas, USA

Bachelor of Engineering (Hons)
Electronics and Electrical Engineering, June 2005
Birla Institute of Technology and Science - Pilani
Dubai Campus, UAE

Experience: **Electrical Engineering (Control Systems) Intern**
Schlumberger Oilfield Services. Sugar Land, TX
(Jan 2007 - June 2007)
Design and Implementation of Artificial Neural Network
and PID controller for Gain regulation loop in density tool

Electrical Engineering (Control Systems) Intern
Schlumberger Oilfield Services. Sugar Land, TX
(June 2006 - August 2006)
Examine the current Gain regulation loop in density tool
and complete a feasibility study on improving the regulation
loop using Artificial Neural Network and PID