

**A VERSATILE SIMULATION TOOL FOR VIRTUAL  
IMPLEMENTATION OF PROPORTIONAL INTEGRAL AND  
DERIVATIVE (PID) CONTROLLERS**

A Thesis

by

INDU RAMAMURTHI

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

May 2007

Major Subject: Electrical Engineering

**A VERSATILE SIMULATION TOOL FOR VIRTUAL  
IMPLEMENTATION OF PROPORTIONAL INTEGRAL AND  
DERIVATIVE (PID) CONTROLLERS**

A Thesis

by

INDU RAMAMURTHI

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,  
Committee Members,

Head of Department

Shankar P. Bhattacharyya  
Aniruddha Datta  
Prasad Enjeti  
Charles Lessard  
Costas N. Georghiades

May 2007

Major Subject: Electrical Engineering

## ABSTRACT

A Versatile Simulation Tool for  
Virtual Implementation of Proportional Integral and Derivative (PID) Controllers.

(May 2007)

Indu Ramamurthi, B.E., National Institute of Technology, Nagpur, India

Chair of Advisory Committee: Dr. Shankar P. Bhattacharyya

This thesis proposes an interactive software tool that can be used to compute complete sets of Proportional Integral Derivative (PID) Controllers from knowledge of the plant transfer function/frequency response data. This is based on research results and algorithms developed by Bhattacharyya and others. Until these research results were published, it was not known if a nominal system could be stabilized using PID Controllers, and current PID Controller designs are carried out using ad hoc tuning rules. By using simulations, engineers can best plan for and observe the stabilizing effect each of the variables has on system performance in a realistic environment.

The software application developed calculates and optimizes complete stabilizing sets of PID Controllers for a rational Linear Time Invariant (LTI) system, and has been developed for analytical models of plants with and without time delay. Further, these PID Controller sets are optimized to project subsets simultaneously satisfying multiple performance index specifications. Sets of PID Controllers that stabilize a system are also

calculated automatically from knowledge of the frequency response of the plant. It allows the user ease of design and the ability to customize the final solution while permitting full control over source parameters.

This thesis includes an introduction to the algorithms that have been developed for plant stabilization, a complete description of the graphical user interface, the simulation of the algorithms performed using LabVIEW, and a summary of future work.

**To Chinmoy**

*Thank You for always being there...*

## ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Dr. Shankar Bhattacharyya for his dynamic and innovative views, constant encouragement and patience throughout this project. It has been a great privilege and an exceptional learning experience to work on this project under his guidance. I would like to thank Ranadip Pal for his initial work on Linear Time Invariant Systems, Sandipan Mitra for his help and interest in my work, and Tammy Carda for helping me immensely and encouraging me throughout the time I have been at Texas A&M University.

I also wish to thank all the members of my advisory committee – Dr. Aniruddha Datta, Dr. Prasad Enjeti and Dr. Charles Lessard for their invaluable comments and interest in my research work.

Finally, I would like to thank my family and friends for the support they have always extended and in particular, I must acknowledge Chinmoy whose encouragement and editing assistance was invaluable.

## TABLE OF CONTENTS

CHAPTER	Page
ABSTRACT .....	iii
DEDICATION .....	v
ACKNOWLEDGEMENTS .....	vi
TABLE OF CONTENTS .....	vii
LIST OF FIGURES.....	ix
 CHAPTER	
I INTRODUCTION.....	1
A Background .....	1
B Research Brief .....	2
C Results .....	2
II LabVIEW – THE SIMULATION TOOL.....	4
A Introduction .....	4
B LabVIEW as the User Interface .....	4
C Inputs in LabVIEW Simulation.....	5
D Outputs in LabVIEW Simulation.....	7
III DELAY FREE RATIONAL TIME INVARIANT PLANT .....	9
A Introduction .....	9
B Theory .....	9
C Block Diagram / Algorithm.....	14
D Final Program generated in LabVIEW.....	30
IV PID CONTROLLER SYNTHESIS: FREE OF ANALYTICAL MODELS .....	39
A Introduction .....	39
B Theory .....	39

CHAPTER	Page
C Algorithm .....	40
D Block Diagram .....	42
E Example.....	45
V PID STABILIZATION OF SYSTEMS WITH TIME DELAY .....	53
A First Order Stable Plant .....	53
1. Theory .....	53
2. Algorithm / Block Diagram.....	56
3. Example.....	58
B Open Loop Stable Plant.....	62
1. Theory .....	62
2. Algorithm / Block Diagram.....	63
C Stabilization of Arbitrary LTI Systems with Time Delay .....	66
1. Theory .....	66
2. Algorithm / Block Diagram.....	68
VI OPTIMIZATION OF STABILIZING SETS OF PID CONTROLLERS.....	79
A Introduction .....	79
B Performance Index Specifications.....	79
C Algorithm .....	80
D Final Program – Front Panel of VI.....	80
VII SUMMARY AND FUTURE RESEARCH DIRECTIONS .....	87
A Summary .....	87
B Future Research Directions .....	87
REFERENCES .....	89
VITA .....	90



## LIST OF FIGURES

	Page
Fig. 2.1.1: Inputs to a Rational Linear Time Invariant System .....	6
Fig. 2.1.2: Inputs to the Model Free Case.....	6
Fig. 2.1.3: Inputs to a Rational Linear Time Invariant System with Time Delay .....	6
Fig. 2.2.1: $k_p$ Range for the Linear Time Invariant Plant with the Transfer Function Shown in Fig.2.1.1 .....	8
Fig. 2.2.2: Stabilizing $k_i - k_d$ Range For $k_p = 1$ .....	8
Fig. 3.1: Closed Loop System with a PID Controller.....	10
Fig. 3.2: Overall Block Diagram to Compute Stabilizing Ranges of $k_p, k_i$ & $k_d$ .....	17
Fig. 3.3: $kp$ Co-Ordinates Sub VI, Computes $D_e(-\omega^2), D_o(-\omega^2)$ $N_e(-\omega^2), N_o(-\omega^2)$ .....	18
Fig. 3.4: Positive Real Roots Are Found for Specific $kp$ .....	19
Fig. 3.5: $Ki - Kd$ Range for Fixed $kp$ .....	20
Fig. 3.6: Front Panel of Linear Time Invariant System– Page 1 .....	21
Fig. 3.7: Final 3-D Plot Showing Stabilizing Range for $k_p - k_i - k_d$ .....	22
Fig. 3.8: LabVIEW Generated Results for $D_e(j\omega), D_o(j\omega), N_e(j\omega), N_o(j\omega)$ .	23
Fig. 3.9: $p_1(\omega), p_2(\omega)$ Generated from Fig. 3.8.....	24
Fig. 3.10: $q(\omega)$ & Final Roots When $k_p = -1.7$ .....	25
Fig. 3.11: $q(\omega)$ & Final Roots When $k_p = -1$ .....	26

	Page
Fig. 3.12: Conditions to be Met: Minimum Number of Roots & Signature.....	26
Fig. 3.13: Strings of Integers Satisfying Inequalities .....	27
Fig. 3.14: Final Coefficient Array .....	28
Fig. 3.15: Flowchart for a Linear Time Invariant System .....	29
Fig. 3.16: Front Panel of the LabVIEW Program for $kp$ Generation .....	31
Fig. 3.17: Page 2 of the LabVIEW Program for $k_i$ & $k_d$ Generation .....	32
Fig. 3.18: Generation of Stabilizing $k_i$ & $k_d$ Subsets for Specific Gain Margin .....	33
Fig. 3.19: Generation of Stabilizing $k_i$ & $k_d$ Subsets for Specific Phase Margin .....	34
Fig. 3.20: Generation of Stabilizing $k_i$ & $k_d$ Subsets for Specific Rise Time ...	35
Fig. 3.21: Generation of Stabilizing $k_i$ & $k_d$ Subsets for Specific Peak Time ..	36
Fig. 3.22: Generation of Stabilizing $k_i$ & $k_d$ Subsets for Specific Settling Time .....	37
Fig. 3.23: Generation of Stabilizing $k_i$ & $k_d$ Subsets for Specific Overshoot ...	38
Fig. 4.1: Block Diagram Part 1- Frequency Response Data.....	43
Fig. 4.2: Block Diagram Part 2 - Frequency Response Data.....	44
Fig. 4.3: Input – The Nyquist and Bode Plots for Frequency Response Data.	45
Fig. 4.4: Strings of Integers Generated by LabVIEW .....	46
Fig. 4.5: Front Panel of VI – Graph Used to Compute Sets of Stabilizing $kp$ Values.....	48

	Page
Fig. 4.6: Graph Showing Stabilizing $k_i$ & $k_d$ for Specific $k_p$ Generated in Fig. 4.5 .....	49
Fig. 4.7: Graph of $k_p$ vs. $\omega$ Generated by LabVIEW .....	50
Fig. 4.8: $k_i - k_d$ Stabilizing Range for $k_p = 0.4$ .....	51
Fig. 4.9: 3-D Stabilizing Set of $k_p$ , $k_i$ and $k_d$ .....	52
Fig. 5.1.1: Block Diagram of a First Order Stable Plant with Time Delay .....	57
Fig. 5.1.2: Front Panel of the LabVIEW Program for Stable First Order System, for $k_p$ Generation .....	59
Fig. 5.1.3: Front Panel of the LabVIEW Program for Stable First Order System, for $k_i$ & $k_d$ Generation .....	60
Fig. 5.1.4: 3D Representation of Stabilizing $k_p$ , $k_i$ and $k_d$ .....	61
Fig. 5.2.1: Front Panel of the LabVIEW Program for Unstable First Order System, for $k_p$ Generation .....	64
Fig. 5.2.2: Front Panel of the LabVIEW Program for Unstable First Order System, for $k_i$ & $k_d$ Generation .....	65
Fig. 5.3.1: Block Diagram Part 1 for Arbitrary Order Time Delay System .....	72
Fig. 5.3.2: Block Diagram Showing $\sqrt{M(\omega)}$ .....	73
Fig. 5.3.3: Columns Calculation .....	74
Fig. 5.3.4: Block Diagram Part 2 for Arbitrary Order Time Delay System .....	75
Fig. 5.3.5: Front Panel of Time Delay System of Arbitrary Order - Page 1.....	76
Fig. 5.3.6: Front Panel of Time Delay System of Arbitrary Order - Page 2.....	77
Fig. 5.3.7: Complete Set with $(k_i, k_d)$ Lines Causing Instability Removed .....	78

	Page
Fig. 6.1: Front Panel of VI – Page 1 .....	81
Fig. 6.2: Result Showing Subset with Gain Margin = 1dB .....	82
Fig. 6.3: Front Panel of VI – Page 2 .....	83
Fig. 6.4: Combination of Gain Margin = 1dB, Phase Margin = 30 deg .....	84
Fig. 6.5: Front Panel of VI – Page 3 .....	85
Fig. 6.6: Combination of Gain Margin = 1dB, Phase Margin = 30 deg & Peak Time = 10secs.....	86

# CHAPTER I

## INTRODUCTION

The thesis proposes a novel approach to the problem of stabilizing any given plant using PID Controllers. Recent research results indicate that complete sets of PID Controllers that stabilize a system can be estimated from the Transfer Function / Frequency Response Data of a Linear Time Invariant System. For the first time an interactive software has been developed that automates research results and makes it accessible to all, by minimizing complex parameters that engineers have to comprehend to solve the problem of stabilization. The solution to the algorithms requires a good insight into control systems, and having a simulation interface allows the engineer to arrive at the solution with minimal knowledge of how the algorithm actually works.

### A. Background

Several control strategies have been developed over the last several decades, but most control systems in the world are operated by PID (Proportional – Integral - Derivative) Controllers. More than 95% of the Controllers used in process control applications are of the PID type. Some of the reasons that PID Controllers are so widely used in the industry are its simple structure (fixed, low order), robustness to modeling errors, relatively good tracking and disturbance rejection.

---

This thesis follows the style of *IEEE Transactions on Industrial Electronics*.

## **B. Research Brief**

The primary objective of the thesis was to develop a versatile design tool for analyzing and verifying the stability of Linear Time Invariant systems. The result was the simulation tool. The software has been developed in LabVIEW because of its elaborate GUIs, and its data acquisition capabilities, which will prove very useful if the system is turned into an online one. The thesis introduces the control system algorithms, and how they have been automated, to allow the user full control over system parameters. Chapter II describes the necessity of having a Graphical user interface, and the advantages of simulation and interface integration. A brief about the algorithms are presented in the first section of Chapters III, IV and V, followed by an explanation of the “Block Diagram” – where the code and calculations are outlined, the “Front Panel” - the User Interface of the software tool, and an example to show the research results. Chapter VI summarizes the research results, and proposes recommendations for future work, and outlines a scheme for the hardware implementation of the above.

## **C. Results**

Bhattacharyya and others developed Control System algorithms for the following:

- (i) Complete sets of PID Controllers for Continuous time rational plants of arbitrary order
- (ii) Compute complete sets of PID Controllers for model free systems

- (iii) Compute complete sets of PID Controllers in first and arbitrary order Continuous LTI systems with time delay

A brief overview of the above mentioned research results and an illustrative example are provided. Further, optimization of stabilizing sets of PID Controllers to project subsets simultaneously satisfying multiple performance index specifications has also been developed.

## CHAPTER II

### LabVIEW – THE SIMULATION TOOL

#### A. Introduction

A Graphical User interface allows the user control over important input and output and simultaneously ensures that there is no over configuring. It provides the user ease of design and control. The manipulation of a simulation model might be straightforward for the engineer who devises it, but sharing of the functionality with other engineers might prove difficult. Though it is natural for the designer to distinguish between the necessary inputs and the default parameters, the same is not true for a secondary user, and both would benefit from an interface that shows all the relevant inputs and outputs, hiding unnecessary parameters, especially when extensive coding is essential. In the simulations accomplished as part of the thesis, extensive codes were written with several default parameters to ensure smooth automated flow of the program.

#### B. LabVIEW as the User Interface

The solution proposed for the problem of stabilization of a plant using a PID Controller is to determine complete sets of  $k_p$ ,  $k_i$  and  $k_d$  in the closed loop which stabilize the system. On solving the algorithms, the stabilizing sets are the solution of a linear programming problem, which has been implemented in LabVIEW.



The inputs to the system are the transfer function (Numerator and Denominator) or the frequency response data (model – free systems). The program computes the complete set of  $k_p$  values; as the user scans over the range of  $k_p$ , the program dynamically computes ranges of  $k_i$  and  $k_d$  for each value of  $k_p$ . Any specific set of  $k_p$ ,  $k_i$  and  $k_d$  values form a stabilizing PID controller.

### **C. Inputs in LabVIEW Simulation**

The input of the LabVIEW programs is the transfer function or the frequency response data. In the latter case, the input is a file with the frequency response of the unstable plant, and the number of right hand poles / zeros that the plant contains. Once the ranges of  $k_p$ ,  $k_i$  and  $k_d$  are finalized, the user can scan the range of  $k_i$  and  $k_d$  for a fixed value of  $k_p$  to get subsets of  $k_i$  and  $k_d$  which meets the user's specs. In the time delay case, the user also inputs the Time Delay of the system along with the transfer function. The program deletes those ranges of  $k_i$  and  $k_d$  values in the stabilizing range that would otherwise cause instability to the closed loop plant.

The program also gives the user a 3D graphical representation of the stabilizing sets of  $k_p$ ,  $k_i$  and  $k_d$ , which is very useful. Fig 2.1.1, Fig 2.1.2 and 2.1.3 show the Graphical Input Interfaces for the Rational Time Invariant System, the Frequency Response

System, and the Linear Time Invariant System with Time Delay. The ranges that the stabilizing set can extend to are also defined.

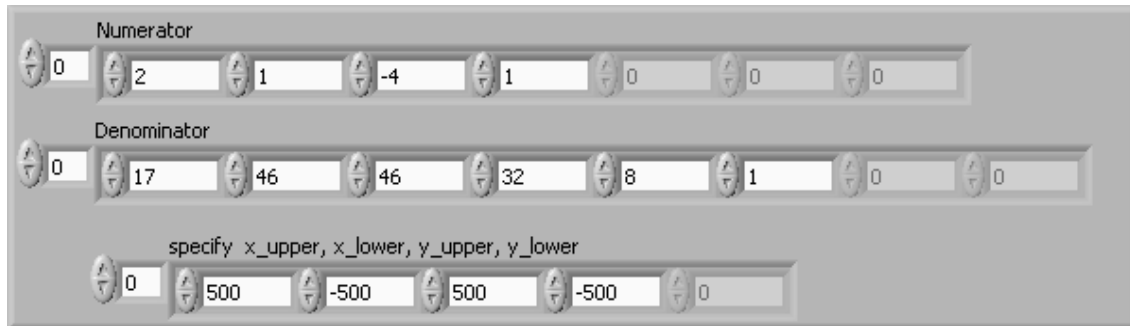


Fig. 2.1.1: Inputs to a Rational Linear Time Invariant System

(Numerator and Denominator Transfer Functions and Upper and Lower Limits)

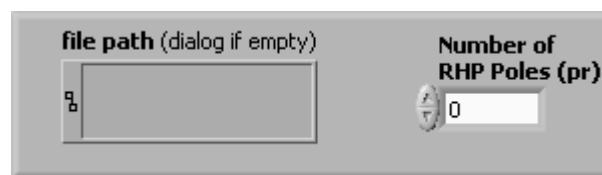


Fig. 2.1.2: Inputs to the Model Free Case

(Frequency Response and Number of Right Hand Plant Poles)

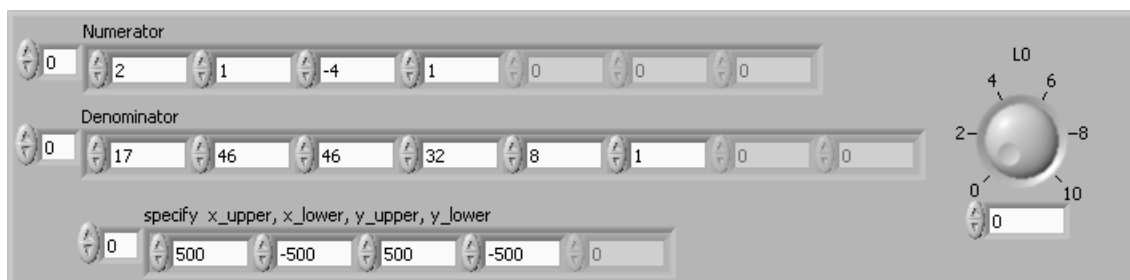


Fig. 2.1.3 Inputs to a Rational Linear Time Invariant System with Time Delay

(Numerator and Denominator, Time Delay  $L_0$ , and Upper and Lower Limits)

The input slides and knobs show the user the variable parameters, and the parameters he can control, eliminating the confusion of sorting through the unnecessary constants and algorithms. In this capacity, a user interface makes inputs easy to manipulate, and keeps the user confined to the boundaries of simulation.

#### **D. Outputs in LabVIEW Simulation**

The program computes the range of  $k_p$  values that stabilize the system. For a particular value of  $k_p$ , the program computes the entire range  $k_i - k_d$  values that stabilize the plant. It also gives the user a 3D plot showing the entire range of values on the  $k_p - k_i - k_d$  plot.

Fig.2.2.1 shows the stabilizing  $k_p$  range that is computed by software. Fig. 2.2.2 shows the Stabilizing  $k_i - k_d$  range that is calculated by solving sets of Linear Inequalities for  $k_p = 1$ .

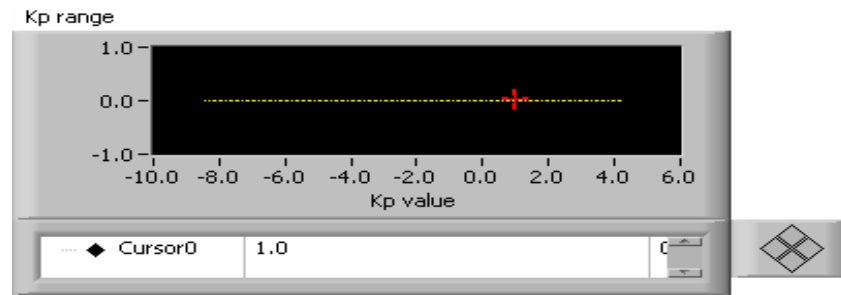


Fig. 2.2.1:  $k_p$  Range for the Linear Time Invariant Plant with the Transfer Function Shown in Fig. 2.1.1

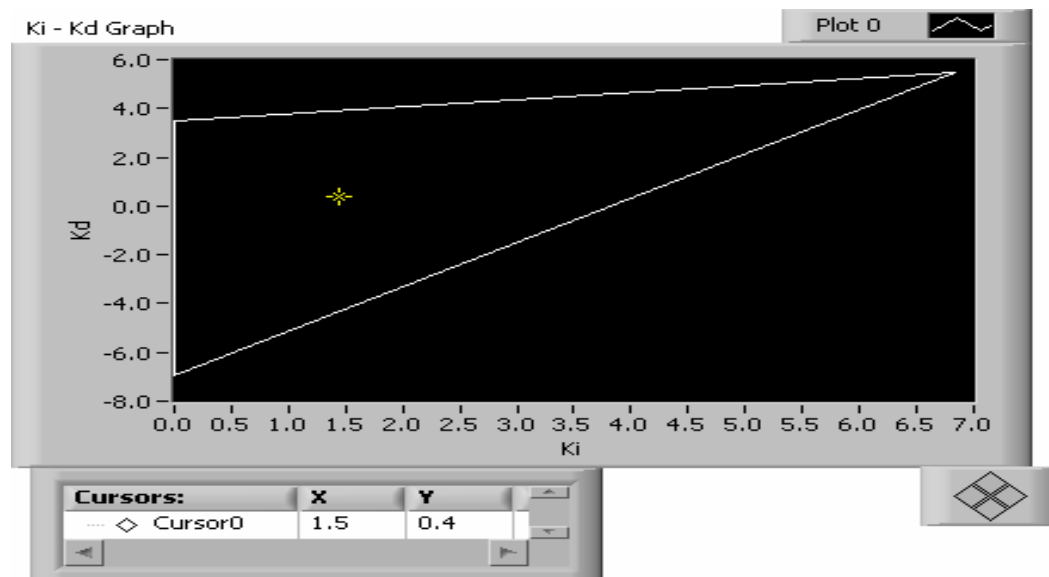


Fig. 2.2.2: Stabilizing  $k_i - k_d$  Range For  $k_p = 1$ .

## CHAPTER III

### DELAY FREE RATIONAL TIME INVARIANT PLANT

#### A. Introduction

This result is based on an extension of the classical Hermite Biehler theorem for root counting. The problem of stabilization is to determine the  $k_p$ ,  $k_i$  and  $k_d$  values which would stabilize the Closed Loop system.

#### B. Theory

The system considered is  $G(s)$  where:

$$G(s) = \frac{N(s)}{D(s)} \tag{3.1}$$

$N(s)$  – Numerator,  $D(s)$  – Denominator, in the Laplace variable 's'.

$N(s)$  and  $D(s)$  are coprime polynomials

$C(s)$  - PID controller where:

$$C(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_i + k_p s + k_d s^2}{s} \tag{3.2}$$

$k_p$  – Proportional Gain;

$k_i$  – Integral Gain

$k_d$  – Derivative Gain.

The closed loop polynomial is shown in Fig. 3.1.

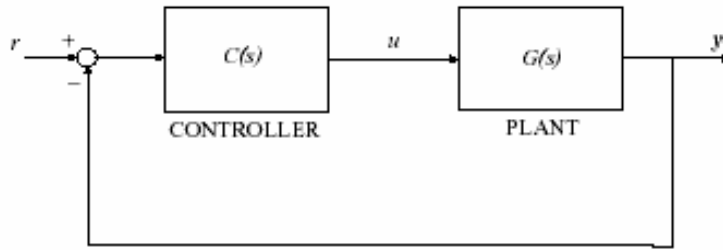


Fig 3.1: Closed Loop System with a PID Controller

The problem of stabilization is to determine the  $k_p$ ,  $k_i$  and  $k_d$  values which will stabilize the Closed Loop system. The closed loop polynomial given by:

$$\delta(s, k_p, k_i, k_d) = sD(s) + (k_i + k_d s^2)N(s) + k_p sN(s) \quad (3.3)$$

The PID Controller stabilizes the system if for the calculated values of  $k_p, k_i$  and  $k_d$ , the closed-loop characteristic polynomial  $\delta(s, k_p, k_i, k_d)$  is Hurwitz. Since all the three parameters affect the even and even and odd parts of  $\delta(s, k_p, k_i, k_d)$ , a new polynomial is constructed where the even part depends on  $(k_i, k_d)$  and the odd part depends on  $k_p$ .

Defining:

$$N(s) = N_e(s^2) + sN_o(s^2)$$

$$D(s) = D_e(s^2) + sD_o(s^2)$$

$$N^*(s) = N(-s) = N_e(s^2) - sN_o(s^2) \quad (3.4)$$

$n, m$  are the degrees of  $\delta(s, k_p, k_i, k_d)$  and  $N(s)$  respectively. Multiplying  $\delta(s, k_p, k_i, k_d)$  with  $N^*(s)$  & substituting  $s = j\omega$  and separating even and odd parts:

$$\delta(j\omega, k_p, k_i, k_d)N^*(j\omega) = p(\omega, k_i, k_d) + jq(\omega, k_p)$$

$$p(\omega, k_i, k_d) = p_1(\omega) + (k_i - k_d\omega^2)p_2(\omega)$$

$$q(\omega, k_p) = q_1(\omega) + k_pq_2(\omega) \quad (3.5)$$

Hence:

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

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

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

$$q_2(\omega) = \omega(N_e(-\omega^2)N_e(-\omega^2) + \omega^2N_o(-\omega^2)N_o(-\omega^2)) \quad (3.6)$$

Here  $k_i, k_d$  appear in  $p(\omega, k_i, k_d)$  while  $k_p$  appears in  $q(\omega, k_p)$ , and does not depend on

$k_i$  or  $k_d$ . For a given  $k_p$ :

$$0 = \omega_0 < \omega_1 < \omega_2 \dots < \omega_{l-1} \quad (3.7)$$

are the real, non-negative, distinct finite zeros of  $q(\omega, k_p)$  with odd multiplicities. A sequence of numbers,  $i_0, i_1, \dots, i_l$  are defined such that:

$$(i) \quad i_0 = \begin{cases} \text{sgn}[p_{1_f}^{(k_n)}(0)] & \text{If } N^*(s) \text{ has a zero of multiplicity } k_n \text{ at the origin} \\ \alpha & \text{Otherwise} \end{cases} \quad (3.8)$$

Where  $\alpha \in \{-1, 1\}$  and

$$p_{1_f}(\omega) := \frac{p_1(\omega)}{(1 + \omega^2)^{\frac{(m+n)}{2}}} \quad (3.9)$$

(ii) For  $t = 1, 2, \dots, l-1$ :

$$i_t = \begin{cases} 0 & \text{If } N^*(j\omega_t) = 0 \\ \alpha & \text{Otherwise} \end{cases} \quad (3.10)$$

$$(iii) \quad i_l = \begin{cases} \alpha & \text{If } n+m \text{ is even} \\ 0 & \text{If } n+m \text{ is odd} \end{cases} \quad (3.11)$$

Defining the string  $I : N \rightarrow R$  as the following sequence of numbers:

$$I := \{i_0, i_1, \dots, i_l\} \quad (3.12)$$

$A_{k_p}$  is the set of all possible strings  $I$  that can be generated to satisfy the requirements,

and  $\gamma(I)$  is associated with any element  $I \in A_{k_p}$ ,



For a given fixed  $k_p$  let  $0 = \omega_0 < \omega_1 < \omega_2 < \dots < \omega_{l-1}$  be the real, non-negative distinct finite zeros of  $q_f(\omega, k_p)$  with odd multiplicities:  $\omega_l = \infty$  and for each string  $I = \{i_0, i_1, \dots\}$  in  $A_{k_p}$ .

Let  $\gamma(I)$  is the imaginary associated with the string  $I$  defined by:

$$\gamma(I) := [i_0 - 2i_1 + 2i_2 + \dots + (-1)^{l-1} 2i_{l-1} + (-1)^l i_l] \cdot (-1)^{l-1} \text{sgn}[q(\infty, k_p)] \quad (3.13)$$

The set  $F_{k_p}^*$  of feasible strings for the PID stabilization problem is defined as:

$$F_{k_p}^* = \{I \in A_{k_p} \mid \gamma(I) = n - (l(N(s)) - r(N(s)))\} \quad (3.14)$$

The PID stabilization problem with a fixed  $k_p$  is solvable for a given plant with rational transfer function,  $G(s)$  if and only if the following conditions hold:

(i)  $F_{k_p}^*$  is not empty, and there is at least one feasible string

(ii) There exists a string

$$I = \{i_0, i_1, \dots\} \in F_{k_p}^* \quad (3.15)$$

and values of  $k_i$  and  $k_d$  such that for all  $t = 0, 1, 2, \dots$  for which

$$N^*(j\omega_t) \neq 0 \cdot p(\omega_t, k_i, k_d) i_t > 0, \quad (3.16)$$

Where  $p(\omega_t, k_i, k_d)$  is already defined.

Also, if there exist values of  $k_i$  and  $k_d$  such that the above condition is satisfied for the feasible strings

$$I_1, I_2, \dots, I_s \in F_{k_p}^* \quad (3.17)$$

Then the set of stabilizing  $(k_i, k_d)$  values corresponding to the fixed  $k_p$  is the union of  $(k_i, k_d)$  values satisfying

$$p(\omega, k_i, k_d) i_t > 0 \text{ for } I_1, I_2, \dots, I_s. \quad (3.18)$$

### C. Block Diagram / Algorithm

The Numerator and Denominator of the Transfer Function are the inputs to the Block “*Initial Calculations*,” where the signature, Numerator and Denominator order are calculated.

The signature is given by:

$$n - (l(N(s) - rN(s))) \quad (3.19)$$

The range of  $k_p$  is varied from -100 to +100 in steps of 0.1, and at each point, the value of  $q_f(\omega, k_p)$  is estimated. If the real, non negative finite zeros of  $q_f(\omega, k_p)$  with odd multiplicities is greater than or equals the minimum number of roots required according to the signature, then that value of  $k_p$  is a part of the stabilizing range for the plant. From

here the complete stabilizing range of  $k_p$  in a window of values is estimated. For higher precision the range of values  $k_p$  is scanned over can be controlled.

The user can also scan through the stabilizing range of  $k_p$  and instantaneously observe stabilizing ranges of  $k_i - k_d$ . The Block Diagram that computes this is shown in Fig.3.5. Here, the stabilizing range of  $k_i - k_d$  at a specific value of  $k_p$  is determined. The user can determine the limits for the  $k_i - k_d$  range scan.

Hence, the inputs to the *ki-kd range* SubVI are the Numerator, Denominator, Numerator Order, Denominator Order, signature, and the specific value of  $k_p$  that the user selects.

Using Eqn.3.18 inequalities are constructed to determine the stabilizing range of  $k_i$  and  $k_d$  for the value of  $k_p$  that the user selects.

$k_i$  and  $k_d$  are the unknowns in the inequality, and the sign of the inequalities are determined by the value in the sign array, as given by Eqn.3.17. Hence, the equation has been divided into the constant term and  $k_i k_d$  coefficient arrays. This is fed into a loop that solves inequalities to estimate the  $k_i - k_d$  ranges that simultaneously satisfy the above inequalities for a particular value of  $k_p$ .

The Block diagram which is used to calculate  $k_p$ ,  $k_i$  &  $k_d$  ranges is shown in Fig.3.2. The SubVI “ $k_p$  Co-Ordinates”, where the range of  $k_p$  is shown in detail in Fig.3.3. In Fig.3.4 Positive Real Roots are found for specific  $k_p$  is calculated, which is essential in computing the stabilizing ranges of  $k_p$ ,  $k_i$  and  $k_d$ . Fig. 3.6 denotes the front panel that the user gets on opening the software tool, and Fig. 3.7 shows the 3-D graphical representation of the complete stabilizing  $k_p$ ,  $k_i$  and  $k_d$  set for the Transfer function denoted in Fig. 3.6.

The advantage of using LabVIEW is that these complex mathematical calculations are carried out instantaneously, and the output is a graphical representation of the calculated ranges. So, as the user slides through the feasible  $k_p$  range, the corresponding  $k_i$  -  $k_d$  stabilizing ranges are graphically monitored.

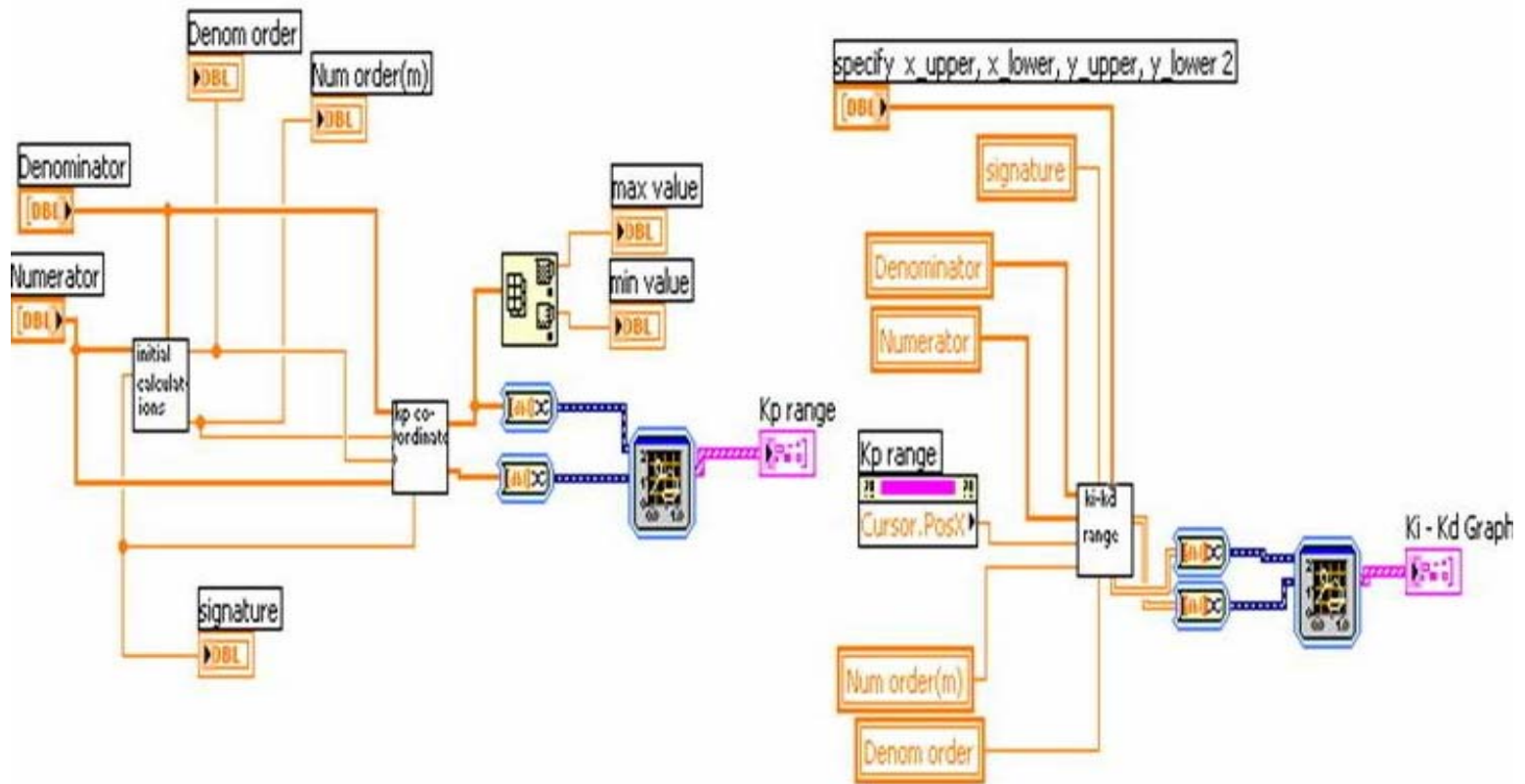


Fig. 3.2: Overall Block Diagram to Compute Stabilizing Ranges of  $k_p$ ,  $k_i$  &  $k_d$

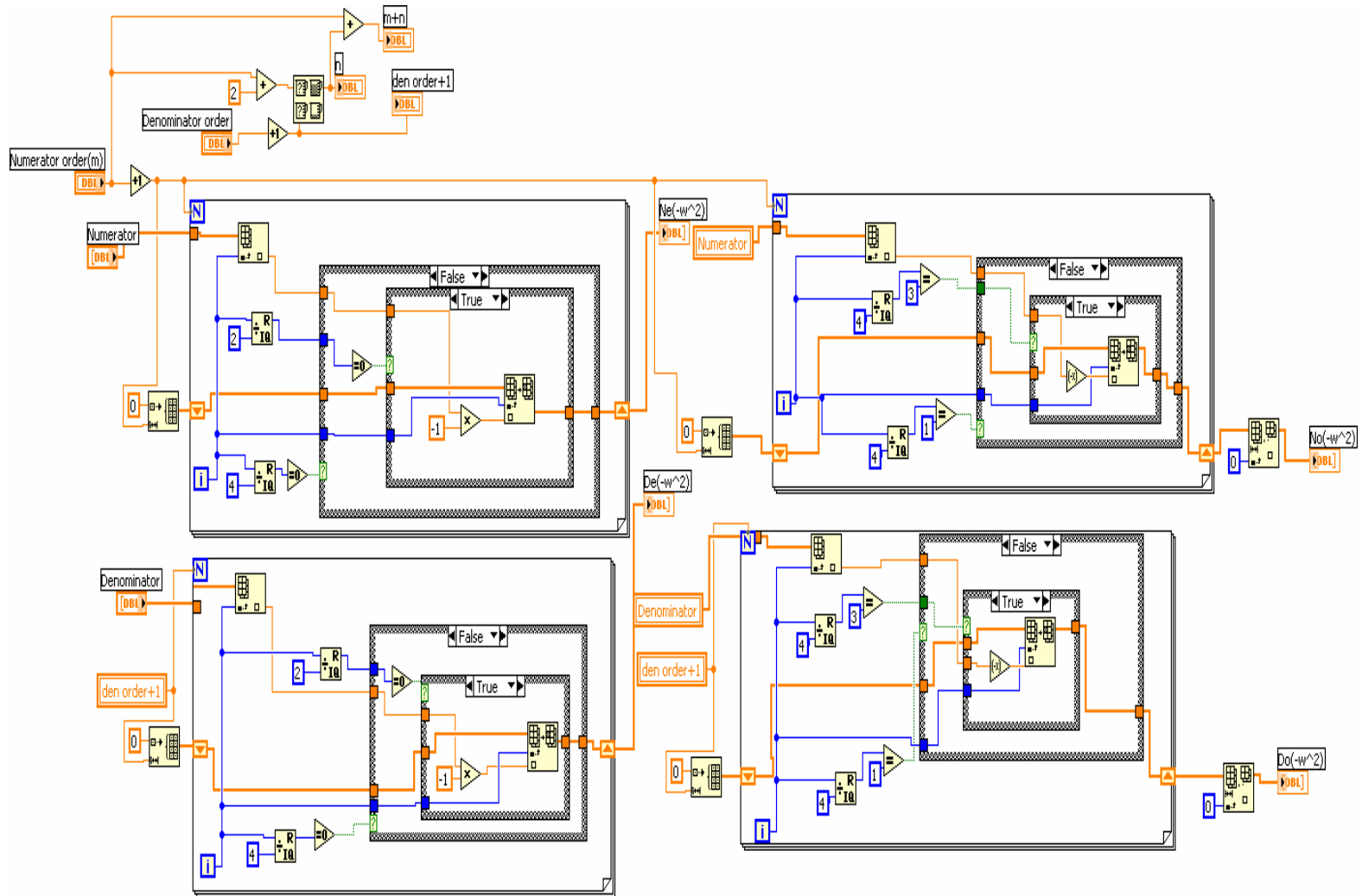


Fig. 3.3:  $k_p$  Co-Ordinates SubVI, Computes  $D_e(-\omega^2)$ ,  $D_o(-\omega^2)$ ,  $N_e(-\omega^2)$ ,  $N_o(-\omega^2)$

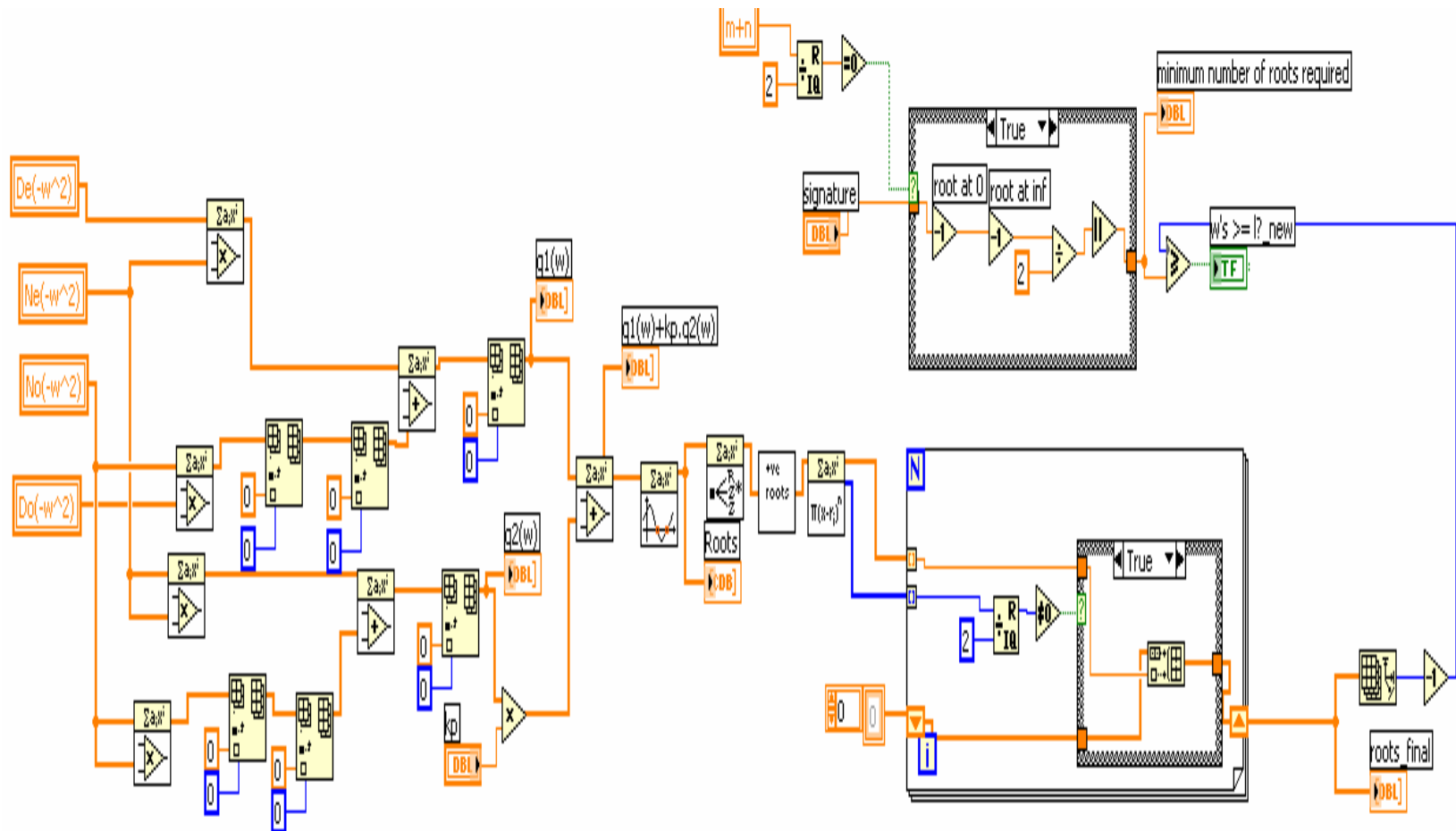


Fig. 3.4: Positive Real Roots Are Found for Specific  $k_p$

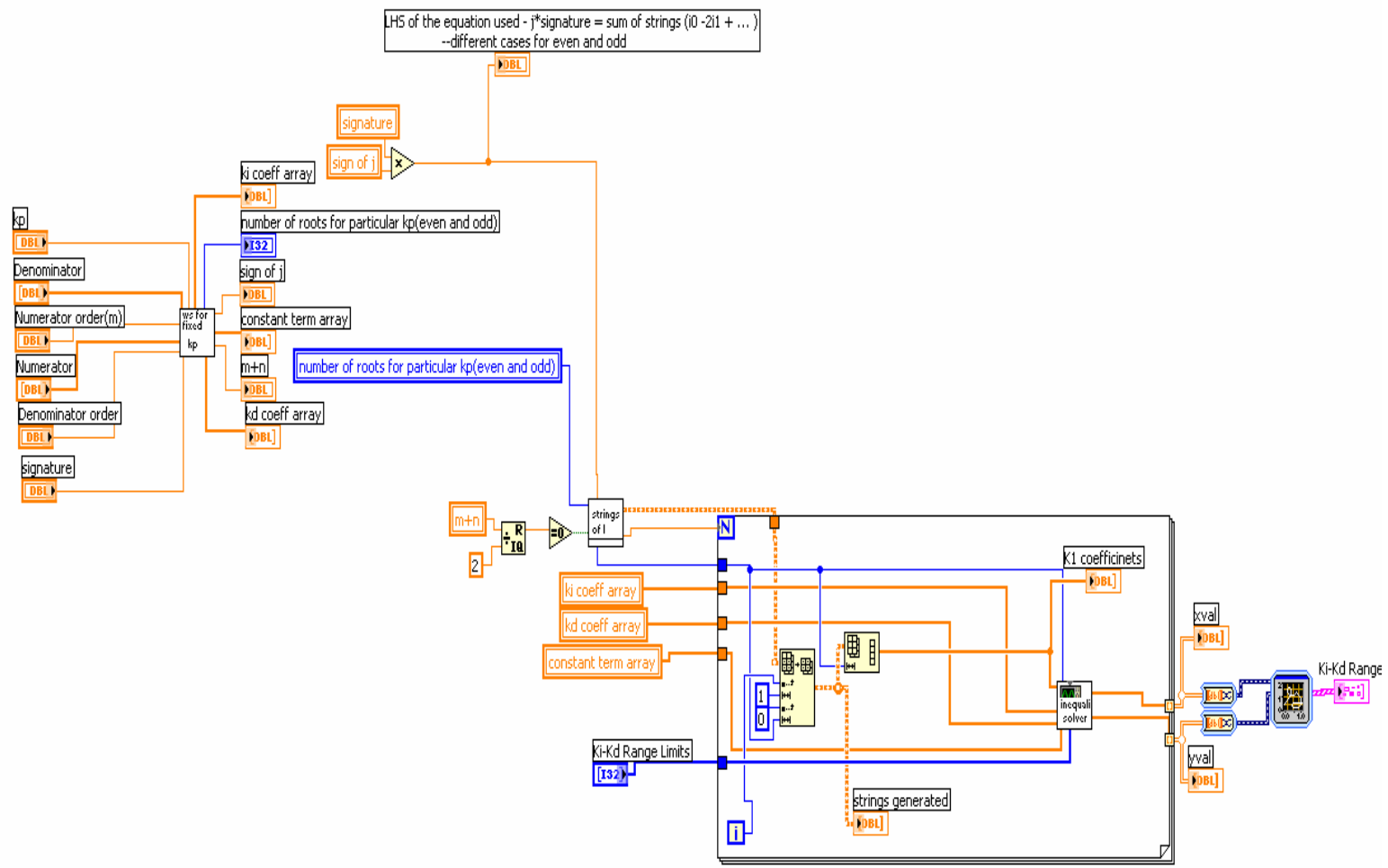


Fig. 3.5:  $k_i - k_d$  Range for Fixed  $k_p$



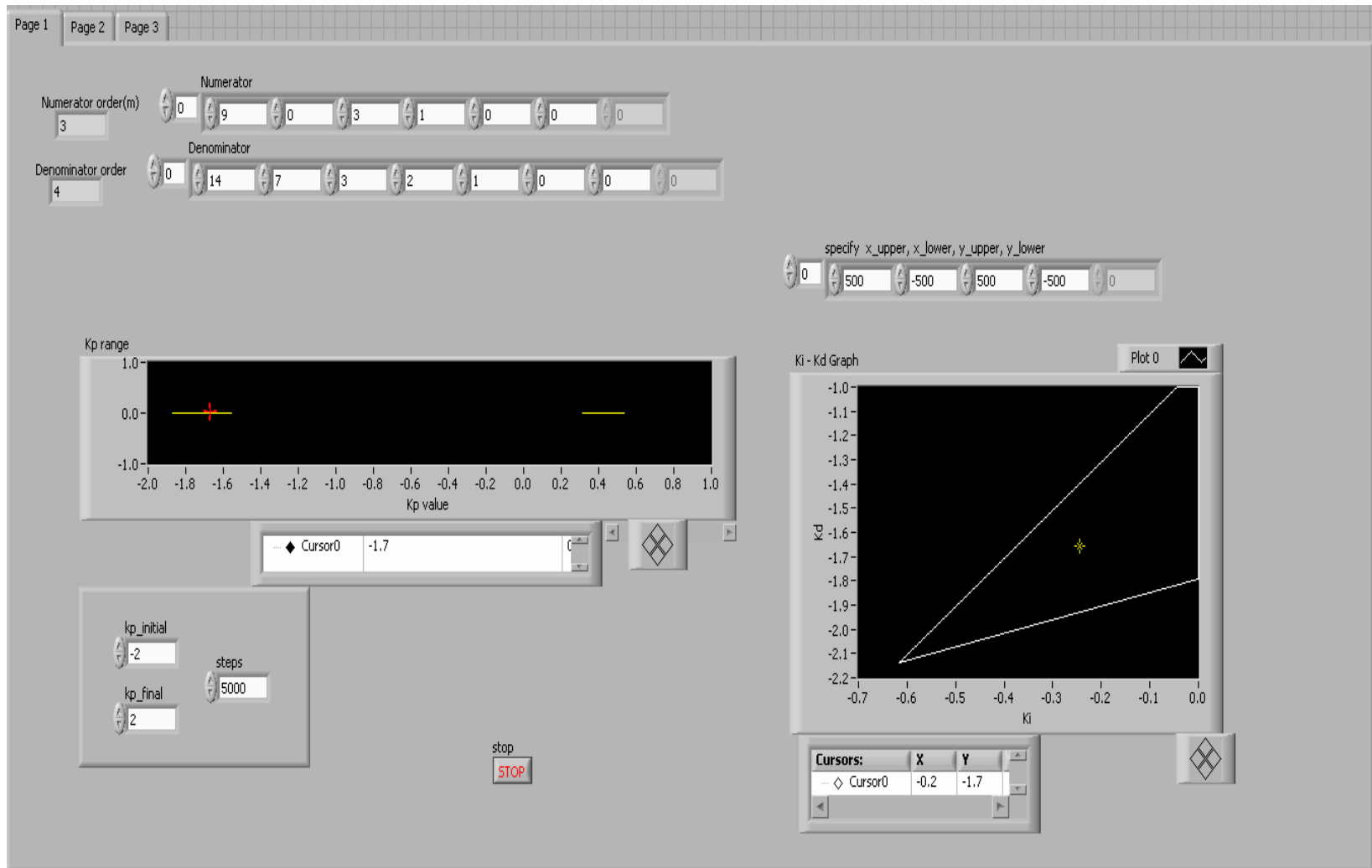


Fig. 3.6: Front Panel of Linear Time Invariant System– Page 1

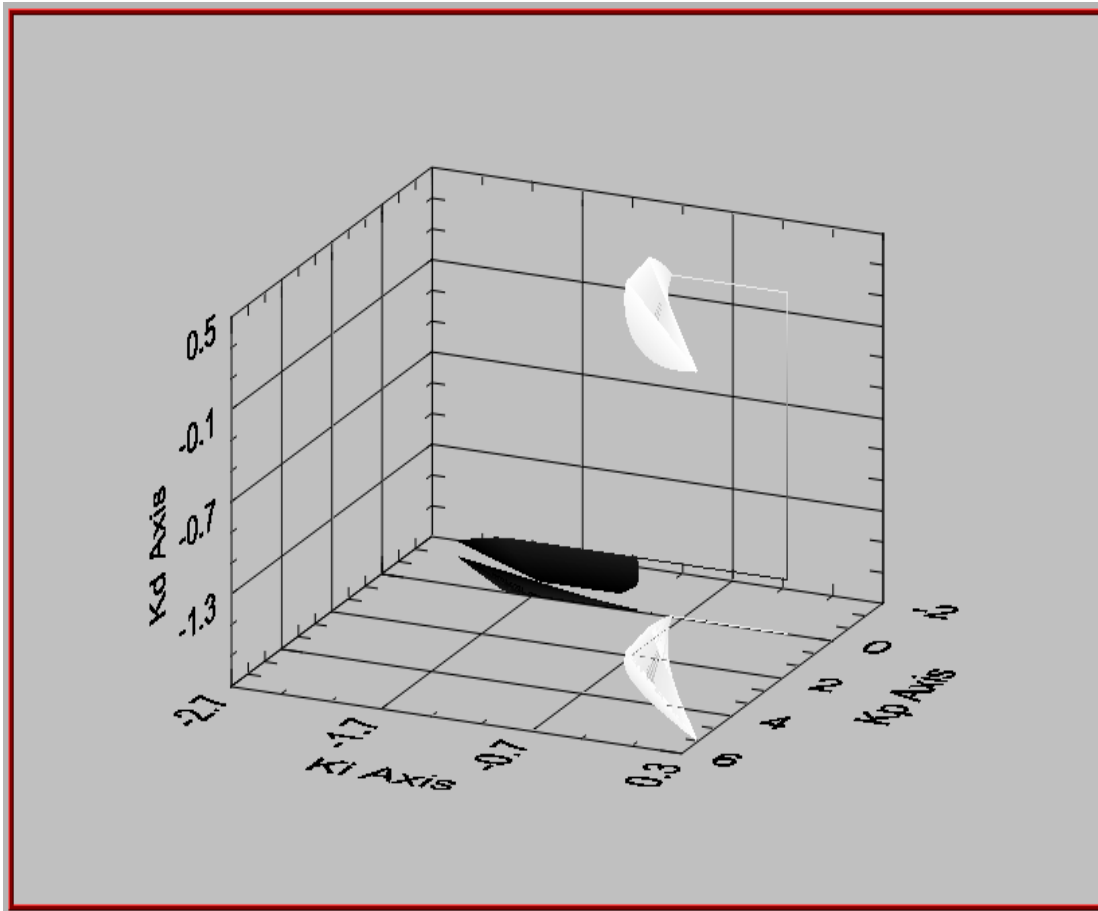


Fig. 3.7: Final 3-D Plot Showing Stabilizing Range for  $k_p - k_i - k_d$

Example:

For the plant defined by:

$$N(s) = s^3 + 3s^2 + 9$$

$$D(s) = s^4 + 2s^3 + 3s^2 + 7s + 14, \text{ and}$$

$$C(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_i + k_p s + k_d s^2}{s}$$

The closed loop polynomial is:

$$\delta(s, k_p, k_i, k_d) = sD(s) + (k_i + k_d s^2)N(s) + k_p sN(s), \text{ where:}$$

$$D(s) = D_e(s^2) + sD_o(s^2)$$

$$N(s) = N_e(s^2) + sN_o(s^2)$$

In the above case,

$$D_e(j\omega) = \omega^4 - 3\omega^2 + 14$$

$$D_o(j\omega) = -2\omega^2 + 7$$

$$N_e(j\omega) = -3\omega^2 + 9$$

$$N_o(j\omega) = -\omega^2$$

Fig. 3.8 denotes the front panel of the software same results as generated by LabVIEW, and Fig.3.9 shows the subsequent  $p_1(\omega), p_2(\omega)$  generated



Fig. 3.8: LabVIEW Generated Results for  $D_e(j\omega), D_o(j\omega), N_e(j\omega), N_o(j\omega)$

And

$$\begin{aligned} N^*(s) &= N(-s) = N_e(s^2) - sN_o(s^2) \\ &= (-4s^2 + 2) - s(s^2 + 1) \end{aligned}$$

$$\begin{aligned} \delta(s, k_p, k_i, k_d)N^*(s) &= [s^2(-12s^6 - 180s^4 - 183s^2 + 75) + (k_i + k_d s^2)(-s^6 + 14s^4 - 17s^2 + 4)] \\ &\quad + s[(-s^8 - 65s^6 - 246s^4 - 22s^2 + 34) + k_p(-s^6 + 14s^4 - 17s^2 + 4)] \end{aligned}$$

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

$$p_1(\omega) = -12\omega^8 + 180\omega^6 - 183\omega^4 - 75\omega^2$$

$$p_2(\omega) = \omega^6 + 14\omega^4 + 17\omega^2 + 4$$

$$q_1(\omega) = -\omega^9 + 65\omega^7 - 246\omega^5 + 22\omega^3 + 34\omega$$

$$q_2(\omega) = \omega^7 + 14\omega^5 + 17\omega^3 + 4\omega$$

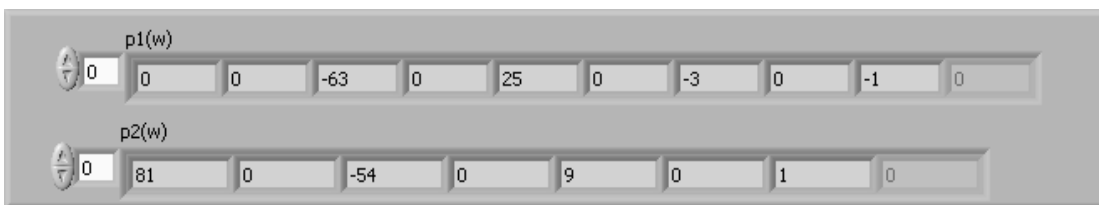


Fig. 3.9:  $p_1(\omega)$ ,  $p_2(\omega)$  Generated from Fig. 3.8

When  $k_p$  is fixed at -1.7:

$$q(\omega, -1.7) = q_1(\omega) + k_p q_2(\omega) = -2.7\omega^7 - 4.3\omega^5 + 22.8\omega^3 - 11.7\omega.$$

Fig. 3.10 shows how LabVIEW computes  $q(\omega)$  and corresponding roots when  $k_p = -1.7$

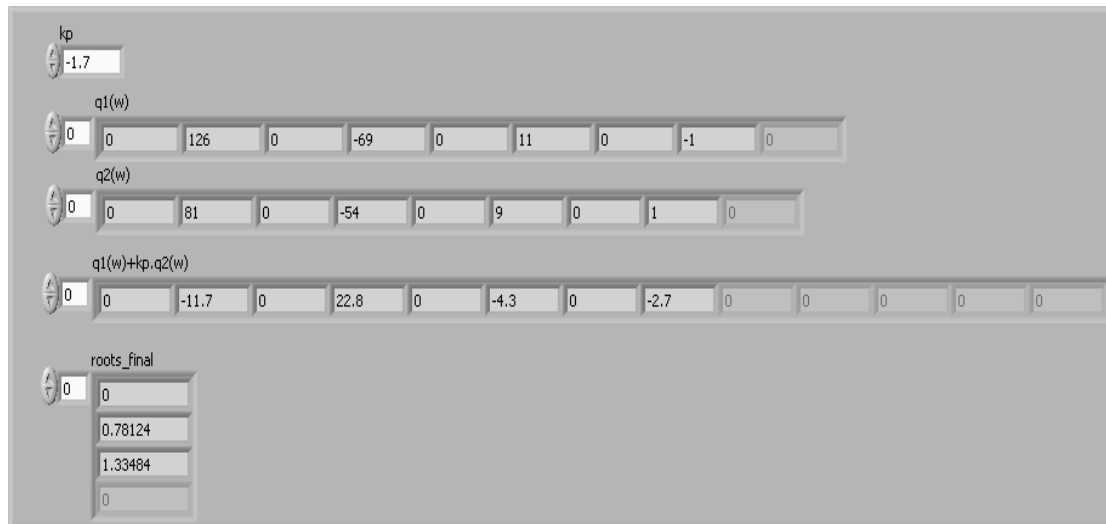


Fig. 3.10:  $q(\omega)$  & Final Roots When  $k_p = -1.7$

Hence, the real, non-negative distinct finite zeros of  $q_f(\omega, -1.7)$  with odd multiplicities are  $\omega_0 = 0, \omega_1 = 0.781, \omega_2 = 1.335$

When  $k_p$  is fixed at -1:

$$q(\omega, -1) = q_1(\omega) + k_p q_2(\omega) = -45\omega^7 - 15\omega^5 + 2\omega^3 - 2\omega$$

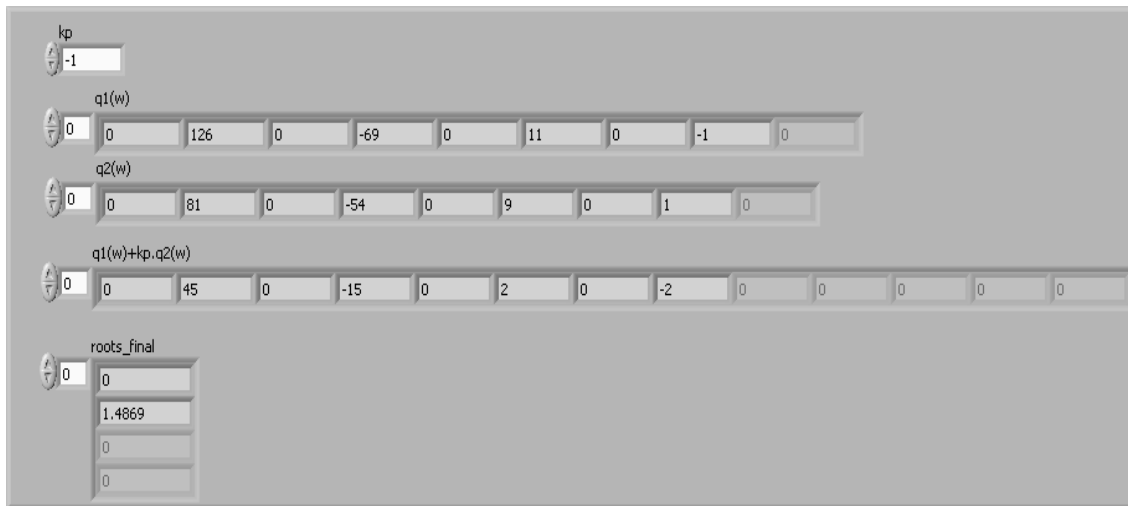


Fig. 3.11:  $q(\omega)$  & Final Roots When  $k_p = -1$

And the real, non-negative distinct finite zeros of  $q_f(\omega, -1)$  with odd multiplicities are

$\omega_0 = 0, \omega_1 = 1.487$  as shown in Fig. 3.11

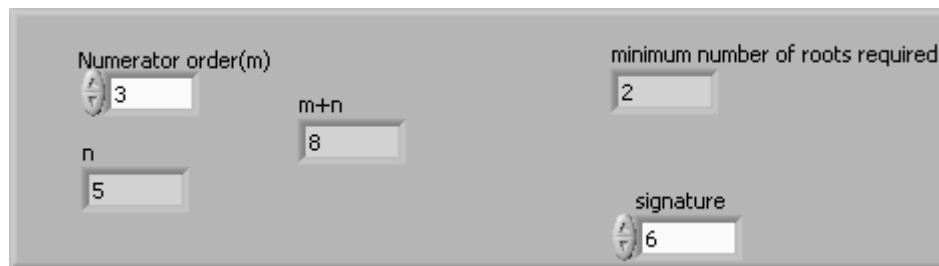


Fig. 3.12: Conditions to be Met: Minimum Number of Roots & Signature

Since  $n + m = 8$  as computed in Fig. 3.12, which is even hence the minimum positive real roots of odd multiplicity other than zero that should be present = 2.

From the integer string generated in Fig. 3.13, the set  $A_1 = \{-1, 1, -1, 1\}$  satisfies the signature = 6.

strings generated	
0	-1 1 -1 1 0 0
0	0 0 0 0 0 0
0	0 0 0 0 0 0
0	0 0 0 0 0 0

Fig. 3.13: Strings of Integers Satisfying Inequalities

$$A_1 = \left\{ \begin{array}{ll} \{-1, -1, -1, -1\} & \{1, -1, -1, -1\} \\ \{1, -1, -1, 1\} & \{1, -1, -1, 1\} \\ \{-1, -1, 1, -1\} & \{1, -1, 1, -1\} \\ \{-1, -1, 1, 1\} & \{1, -1, 1, 1\} \\ \{-1, 1, -1, -1\} & \{1, 1, -1, -1\} \\ \{-1, 1, -1, 1\} & \{1, 1, -1, 1\} \\ \{-1, 1, 1, -1\} & \{1, 1, 1, -1\} \\ \{-1, 1, 1, 1\} & \{1, 1, 1, 1\} \end{array} \right.$$

Since  $l(N(s)) - r(N(s)) = -6$  and

$$(-1)^{l-1} \operatorname{sgn}[q(\infty, 1)] = 1,$$

It follows from 4.3 that every string

$$I = \{i_0, i_1, i_2, i_3, i_4\} \in F_1^*$$

Needs to satisfy

$$i_0 - 2i_1 + 2i_2 - 2i_3 + i_4 = -6$$

Hence, the stabilizing  $(k_i, k_d)$  values corresponding to  $k_p = -1.7$  must satisfy the string of inequalities:

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

Substituting for  $\omega_0, \omega_1, \omega_2$  and  $\omega_3$  in the above expressions, we obtain from Fig.3.14, the Final Coefficient Array, from which stabilizing ranges of  $(k_i, k_d)$  are computed



Fig. 3.14: Final Coefficient Array

$$\begin{cases} k_i < 0 \\ 81k_i - 31.5066k_d < 29.96 \\ 19.01k_i - 33.8774k_d < 59.933 \\ -k_d < 1 \end{cases}$$

These linear inequalities are solved to find the stabilizing ranges of  $k_p, k_i$  &  $k_d$



The stabilizing range of  $k_p$  values:

$$\delta(s, k_p, k_i, k_d)k_p \in (-1.87, 1.556) \cup (0.316, 0.533).$$

By sweeping over different  $k_p$  values in this interval and following the procedure, the complete set of  $(k_p, k_i, k_d)$  is Hurwitz. A simplified flowchart of the algorithm is shown in Fig. 3.15

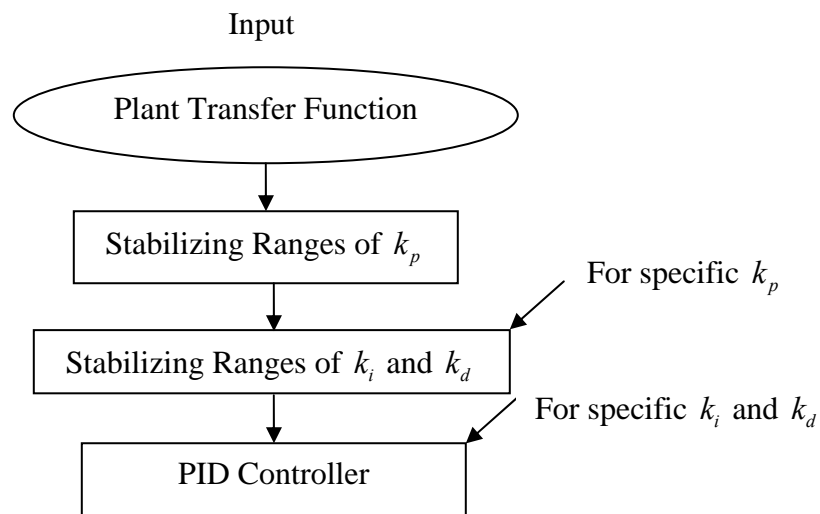


Fig 3.15: Flowchart for a Linear Time Invariant System

#### D. Final Program generated in LabVIEW

The front panel of the VI is shown in Fig.3.16. In this interactive program, the user is able to navigate to the page of his choice. Once ranges of  $k_p$ ,  $k_i$  and  $k_d$  are determined, the user can optimize these ranges with respect to performance index specifications. After specifying  $k_p = 1$ , the  $k_i$ - $k_d$  range is shown on the right hand side. Page two shown in Fig. 3.17 allows the user to scan through the  $k_i$ - $k_d$  range for a specific  $k_p$  value displaying instantaneously corresponding values of Gain Margin, Phase Margin, Rise time, Peak time, Settling time and Overshoot. It also displays the Step Response and Bode plots of the closed loop transfer function which gives the user an idea of the stability of the system.

In the example shown in Fig.3.16, the transfer function used is:

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

$$D(s) = s^5 + 8s^4 + 32s^3 + 46s^2 + 46s + 17, \text{ and}$$

Front panel of the VI where the  $k_i - k_d$  range is calculated for the value of  $k_p$  set to 1

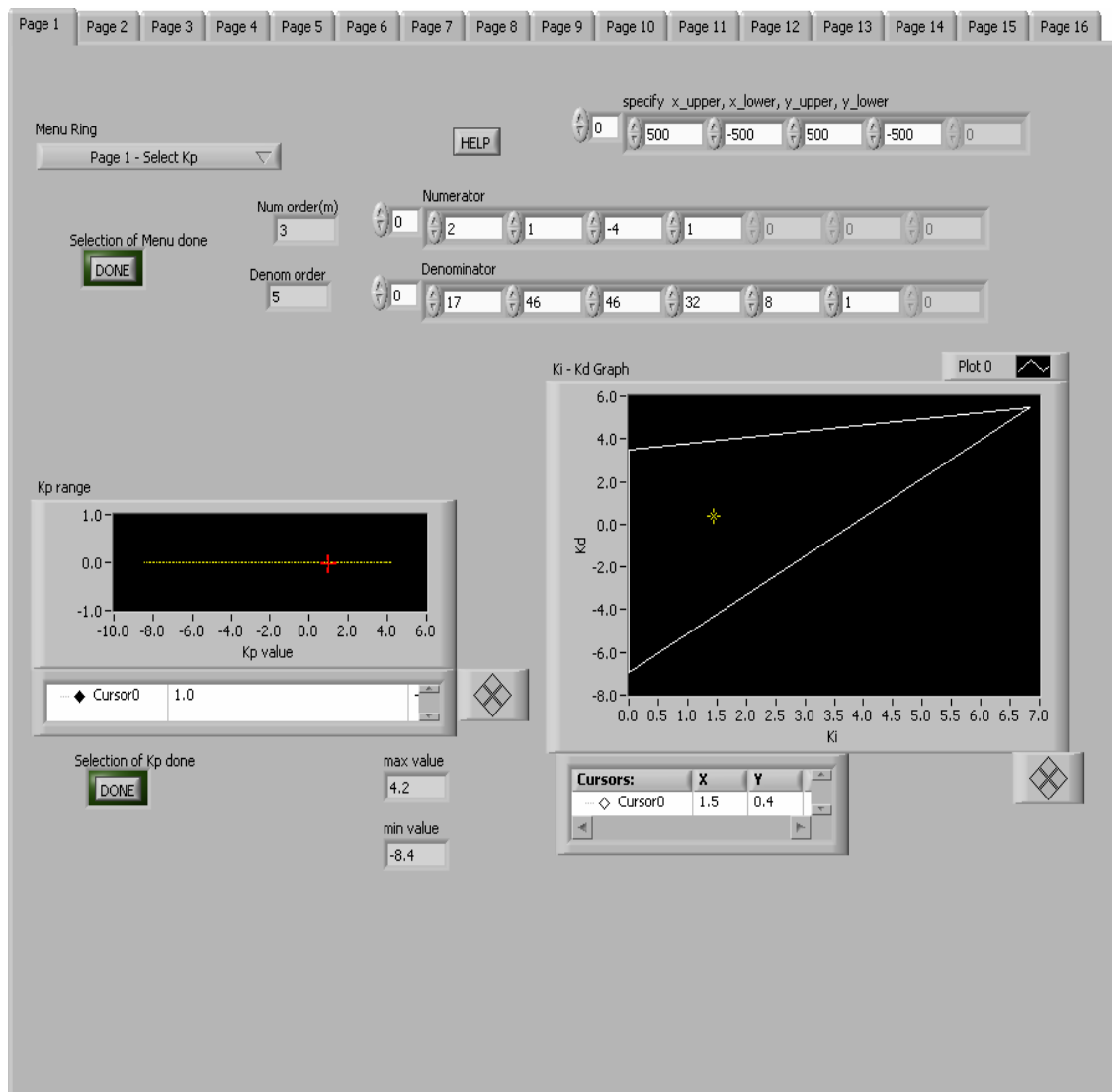


Fig. 3.16: Front Panel of the LabVIEW Program for  $k_p$  Generation

Fig. 3.17 shows Page2 of the VI where for a specific value inside the Ki-Kd plane for a specific value of  $K_p$ , corresponding Gain Margin, Phase Margin, Rise Time, Peak Time, Settling Time and %Overshoot values are computed

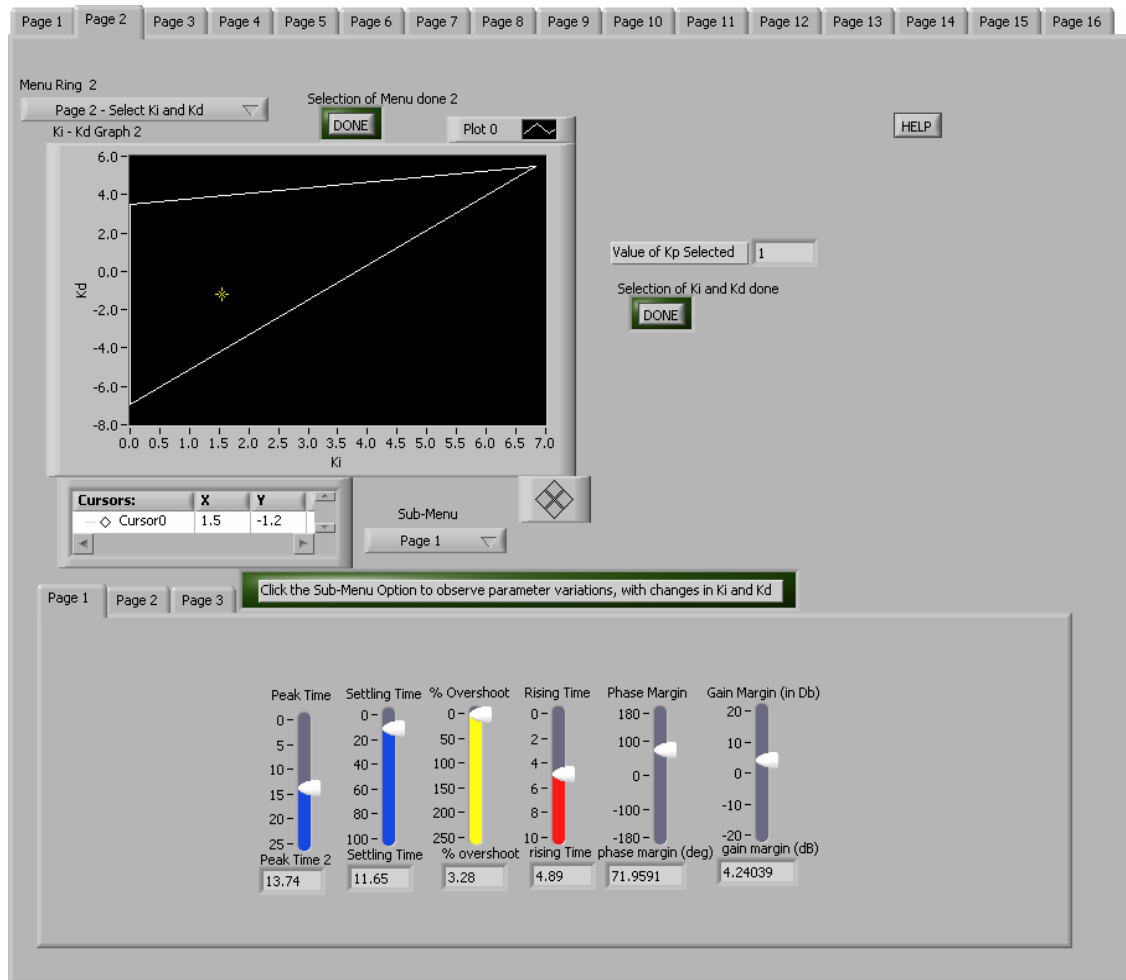


Fig. 3.17: Page 2 of the LabVIEW Program for  $k_i$  &  $k_d$  Generation

Fig. 3.18 shows Page 3 of the VI that allows the user to comb through the range of acceptable Ki-Kd space for a specific value of  $K_p$  and displays the range of Ki-Kd values that would stabilize the system till the User defined Gain Margin value

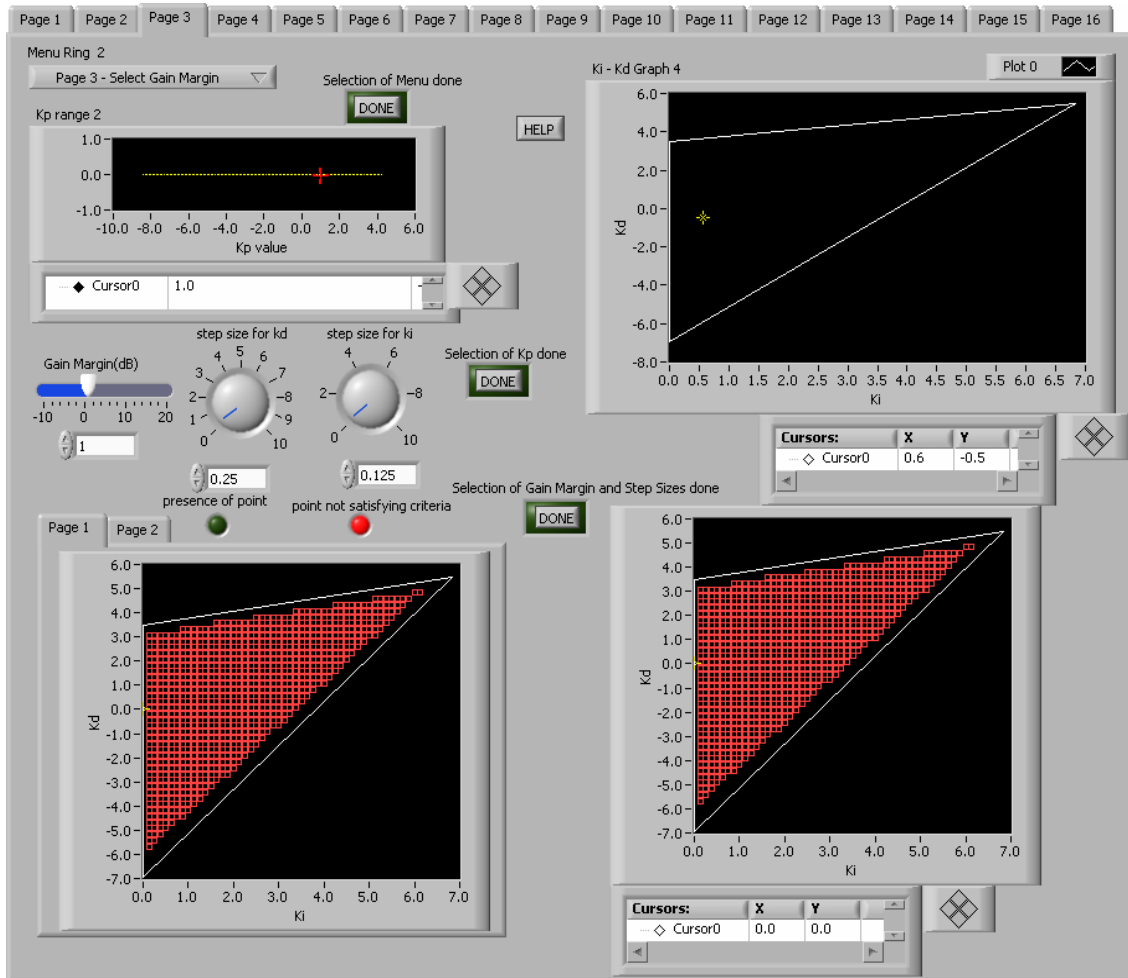


Fig. 3.18: Generation of Stabilizing  $k_i$  &  $k_d$  Subsets for Specific Gain Margin

Fig. 3.19 shows Page 4 of the VI that allows the user to comb through the range of acceptable  $K_i$ - $K_d$  space for a specific value of  $K_p$  and displays the range of  $K_i$ - $K_d$  values that would stabilize the system till the User defined Phase Margin value

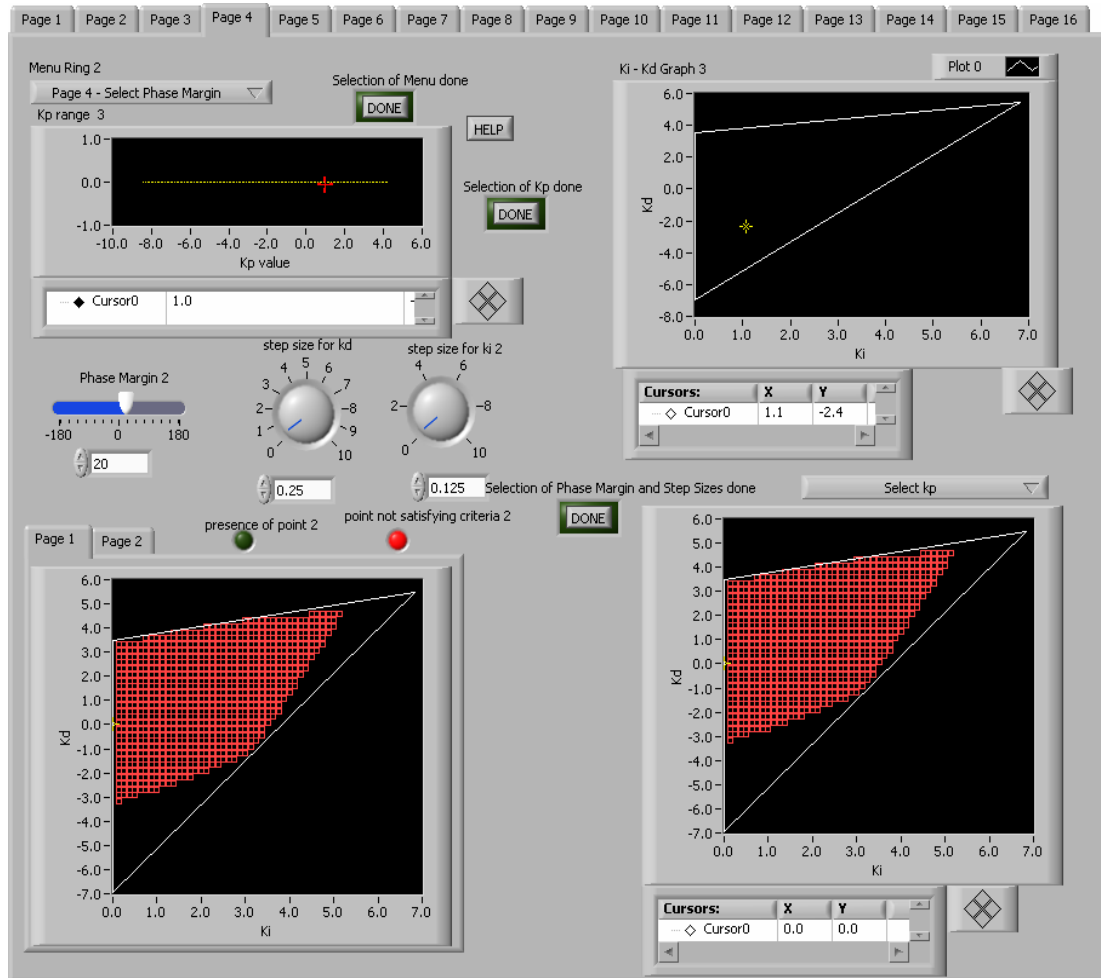


Fig. 3.19: Generation of Stabilizing  $k_i$  &  $k_d$  Subsets for Specific Phase Margin

Fig. 3.20 shows Page-5 of VI that indicates all the points in the  $k_i$ - $k_d$  space that is stable for a specific  $K_p$  value when the system has the user-defined Rise Time

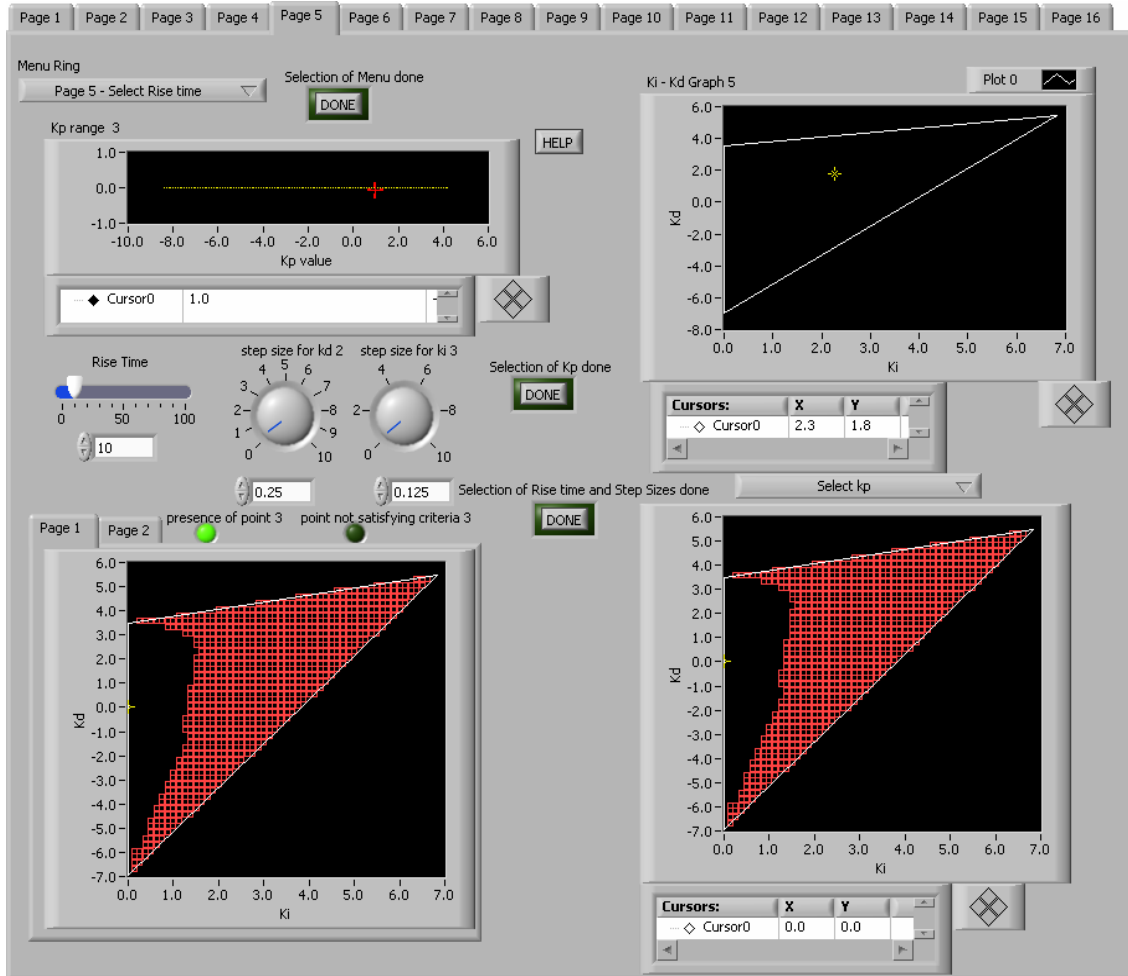


Fig. 3.20: Generation of Stabilizing  $k_i$  &  $k_d$  Subsets for Specific Rise Time

Fig. 3.21 shows Page-6 of the VI that indicates all the points in the  $K_i$ - $K_d$  space that is stable for a specific  $K_p$  value when the system has the user-defined Peak Time

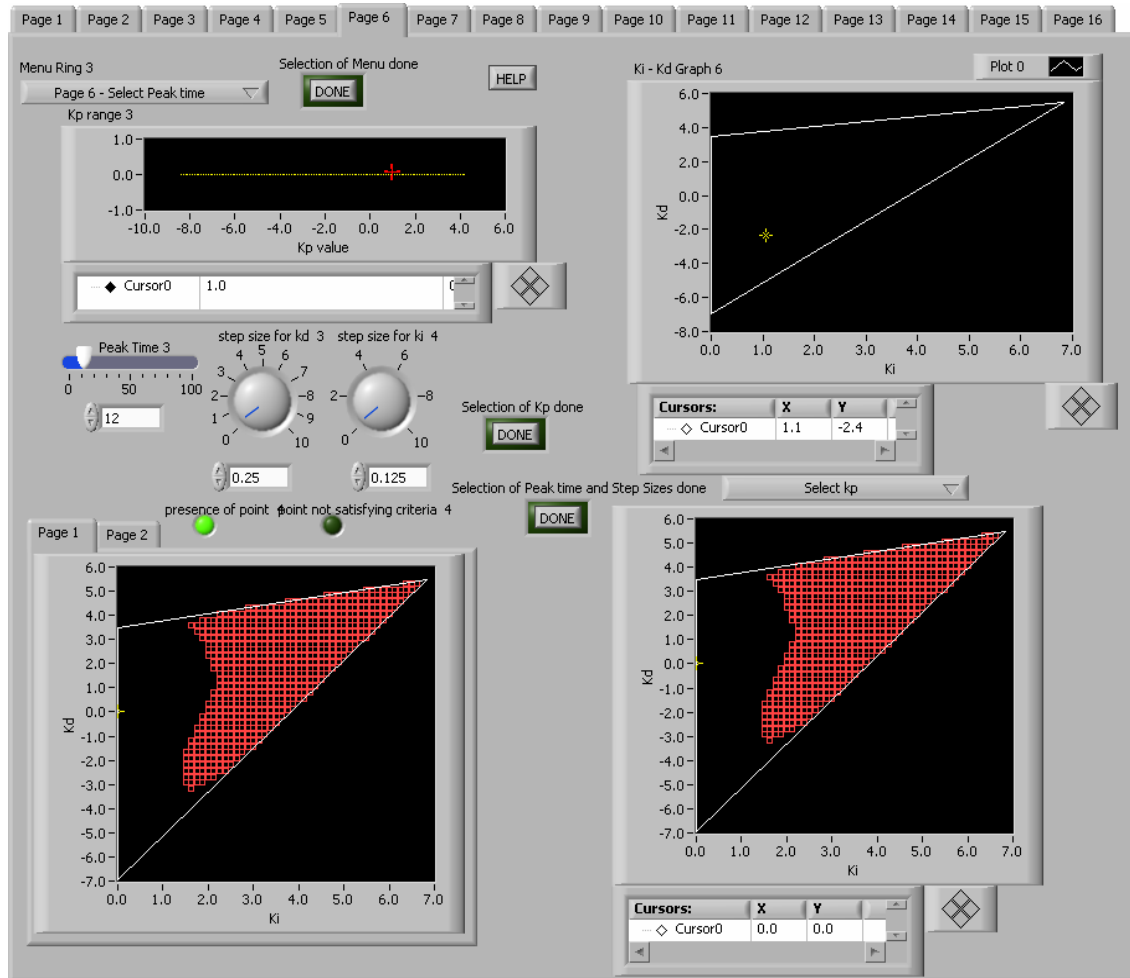


Fig. 3.21: Generation of Stabilizing  $k_i$  &  $k_d$  Subsets for Specific Peak Time

Fig. 3.22 shows Page-7 of the VI that indicates all the points in the Ki-Kd space that is stable for a specific Kp value when the system has the user-defined Settling Time



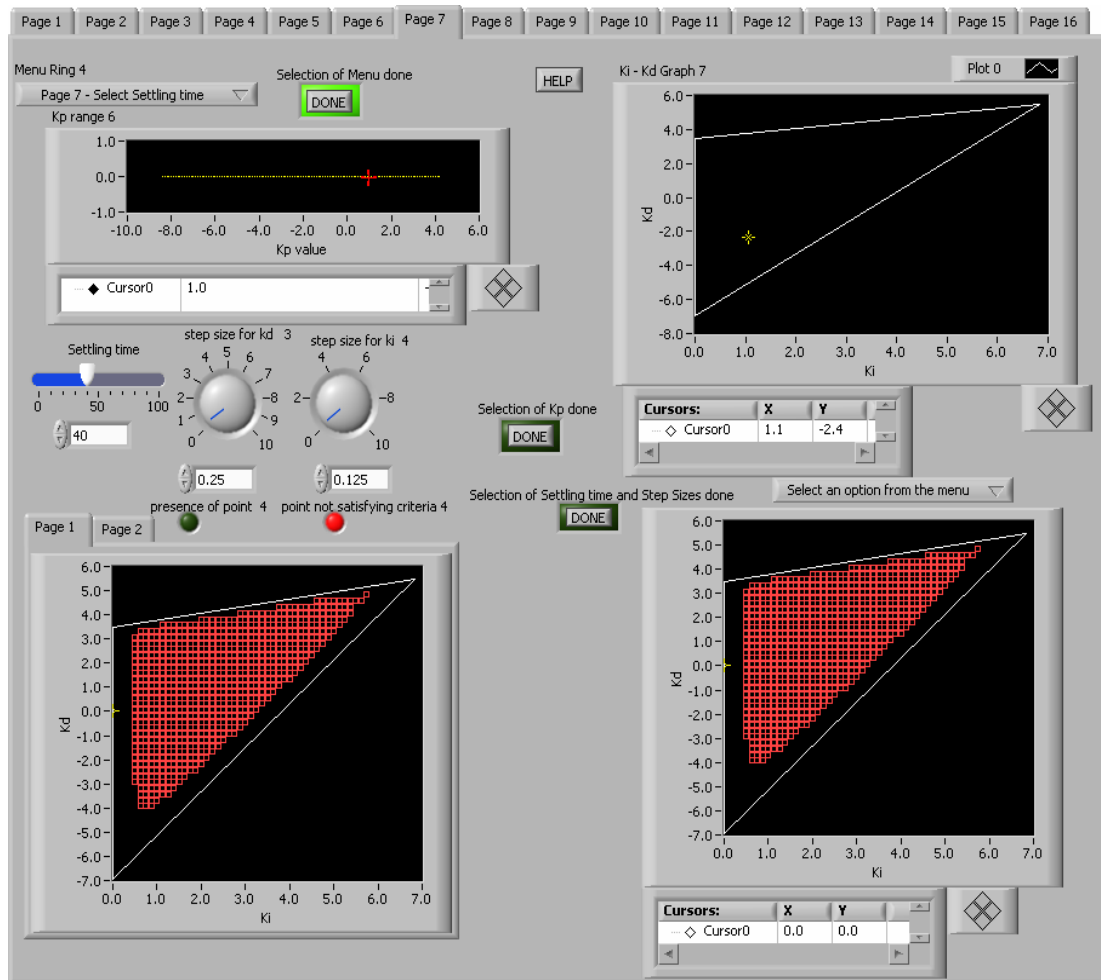


Fig. 3.22: Generation of Stabilizing  $k_i$  &  $k_d$  Subsets for Specific Settling Time

Fig. 3.23 shows Page-8 of the VI that indicates all the points in the  $K_i$ - $K_d$  space that is stable for a specific  $K_p$  value when the system has the user-defined % Overshoot

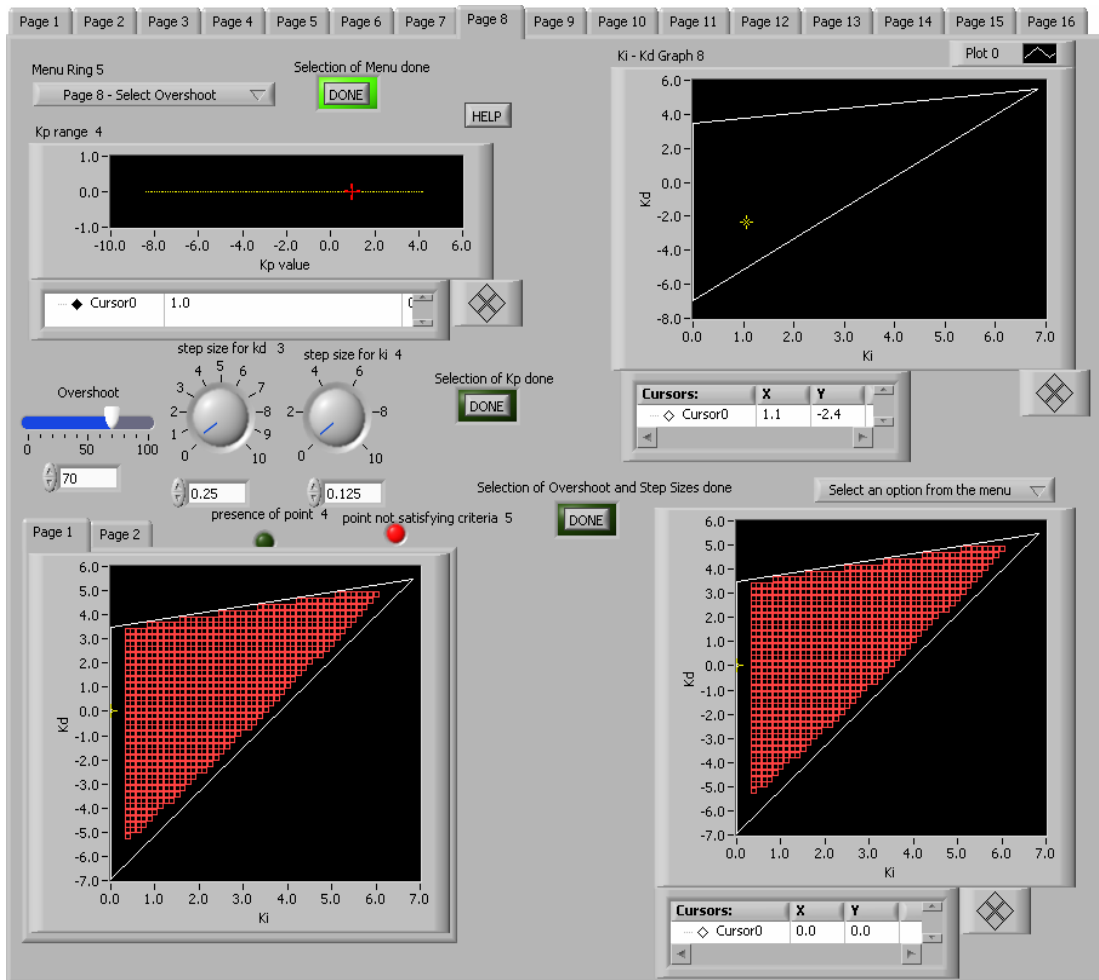


Fig. 3.23: Generation of Stabilizing  $k_i$  &  $k_d$  Subsets for Specific Overshoot

Page three to page seven allow the user to scan over the desired  $k_i$ - $k_d$  range and find subsets within stabilizing ranges that satisfy performance index specifications. Pages 8 to 14 are to allow the user to scan over more precise areas on the  $k_i$ - $k_d$  plane for a specific  $k_p$  value. Fig. 1.6 shows the front panel of the VI where the  $k_i$ - $k_d$  plane is scanned with respect to the Gain Margin. The precision of the scan in all the pages is user-defined.

## CHAPTER IV

### PID CONTROLLER SYNTHESIS: FREE OF ANALYTICAL MODELS

#### A. Introduction

Calculation of stabilizing sets of PID Controllers can be estimated from Raw Plant data, i.e. knowledge of only the frequency response (Nyquist or Bode plot) of the plant and the number of RHP poles. Hence, the complete set of PID Controllers that stabilizes a LTI plant can be computed from the frequency response data. This data can be obtained experimentally, and the number of unstable poles can be determined from physical considerations. The algorithms give a good idea about the frequency range over which accurate information of plant frequency response is needed for PID control, and the complete set of stabilizing gains. The result is really valuable when an accurate analytical / state space model of the system cannot be constructed.

#### B. Theory

In a Linear Time Invariant Systems with an underlying transfer function  $P(s)$  with  $n$  poles and  $m$  zeros, if the only information available to the user is:

- (i) Knowledge of the frequency response magnitude and phase –

$$P(j\omega), \omega \in [0, \infty).$$

- (ii) Knowledge of the number of RHP poles -  $p_r$  : If the plant has no  $j\omega$  poles or zeros then the magnitude and its phase are well-defined for all  $\omega \geq 0$ .

$$P(j\omega) = |P(j\omega)|e^{j\phi(\omega)} = P_r(\omega) + jP_i(\omega) \quad (4.1)$$

Where

$|P(j\omega)|$  -magnitude of the plant

$\phi(\omega)$  - Phase of the plant, at the frequency  $\omega$

The PID Controller is of the form:

$$C(s) = \frac{k_i + k_p s + k_d s^2}{s(1 + sT)}, T > 0 \quad (4.2)$$

where T is assumed to be fixed and small.

### C. Algorithm

- (i) Find the relative degree  $n - m$  from the high frequency slope of the Bode Magnitude plot, and  $z_r$  from the net phase change from the Bode phase plot as shown in the example in Fig. 4.3
- (ii) Fix  $k_p = k_p^*$  and solve for

$$k_p^* = -\frac{P_r(\omega) + \omega T P_i(\omega)}{|P(j\omega)|^2} = -\frac{\cos \phi(\omega) + \omega T \sin \phi(\omega)}{|P(j\omega)|} \quad (4.3)$$

And let  $\omega_1 < \omega_2 < \dots < \omega_{l-1}$  denote the distinct frequencies which are the solutions for Eqn.4.1

- (iii) Set  $\omega_0 = 0, \omega_l = \infty$  and determine all strings of integers  $i_t \in \{+1, 0, -1\}$  and  $j \in \{+1, -1\}$  such that:

For  $n - m$  even:

$$\{i_0 - 2i_1 + 2i_2 + \dots + (-1)^{l-1} 2i_{l-1} + (-1)^l i_l\} \cdot (-1)^{l-1} j = n - m + 2z_r + 2 \quad (4.4)$$

For  $n - m$  odd:

$$\{i_0 - 2i_1 + 2i_2 + \dots + (-1)^{l-1} 2i_{l-1} + (-1)^l 2i_{l-1}\} \cdot (-1)^{l-1} j = n - m + 2z_r + 2 \quad (4.5)$$

- (iv) For the fixed  $k_p = k_p^*$  chosen in Step (ii) solve for the stabilizing  $(k_i, k_d)$  from:

$$\left[ k_i - k_d \omega_t^2 + \frac{\omega_t \sin \phi(\omega_t) - \omega_t^2 T \cos \phi(\omega_t)}{|P(j\omega)|} \right] i_t > 0 \quad (4.6)$$

For  $t = 0, 1, \dots$

## D. Block Diagram

As shown in Fig.4.1, the data file is taken in through the data interface defined in LabVIEW. There are three columns with the Frequency, Magnitude and Phase information of the plant.

The blocks “*delete duplicates*” and “*Mag & Phase Conv*” converts the data into the format used in the rest of the program. The VI “*slope & n-m*” calculates the high frequency slope and hence  $n-m$  that is used in the block “*Min Intersections*” to calculate the Minimum number of intersections required for a stabilizing range of  $k_p$  to exist. Fig. 4.2 shows the block diagram used to generate stabilizing  $(k_i, k_d)$  values for a stabilizing value of  $k_p$  generated in Fig. 4.1. The SubVI “*Gen Eqn*” computes the equation defined in Eqn 4.6 vs.  $\omega$  values.

In “*Strings of Integers*” the values of  $\omega$  that correspond to the stabilizing range of  $k_p$  is calculated, and based on the number of positive, real non-zero roots, the complete sets of strings of integers are defined. The sets of strings that satisfy the signature condition as described in Eqn 4.4 and Eqn 4.5 are identified in the SubVI “*Linear Inequalities*”. These are then translated to the  $(k_i, k_d)$  co-ordinates in the SubVI “*Ki-Kd All Intersections*”, to give the final range of  $(k_i, k_d)$  values for a specific value of  $k_p$ .

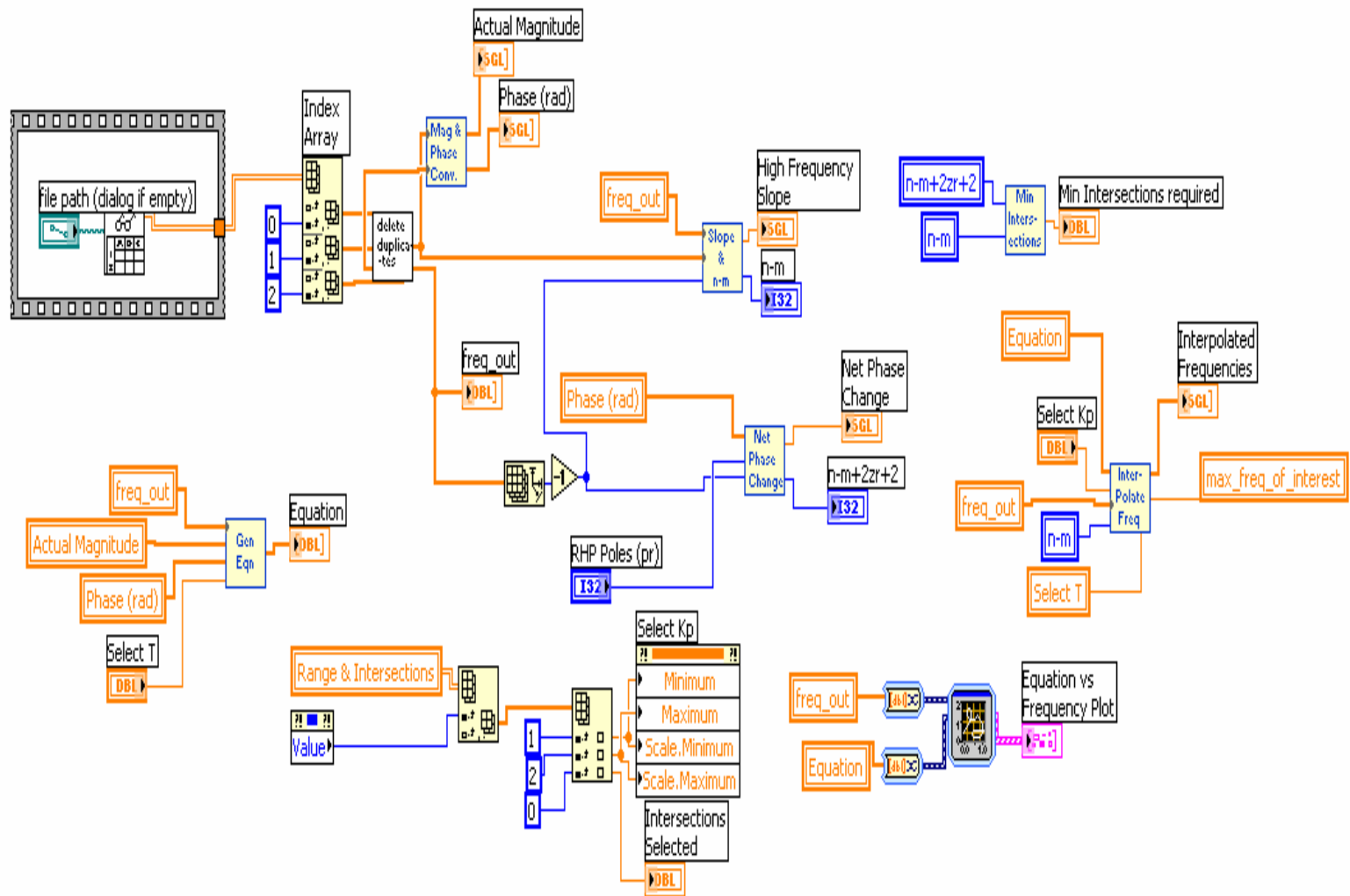


Fig.4.1: Block Diagram Part 1- Frequency Response Data





### E. Example

Taking a set of frequency data points for the stable plant

$$P(j\omega) := \{P(j\omega), \omega \in (0, 60) \text{ Sampled every } 0.01\}.$$

The Nyquist and Bode plots are shown in Fig. 4.3

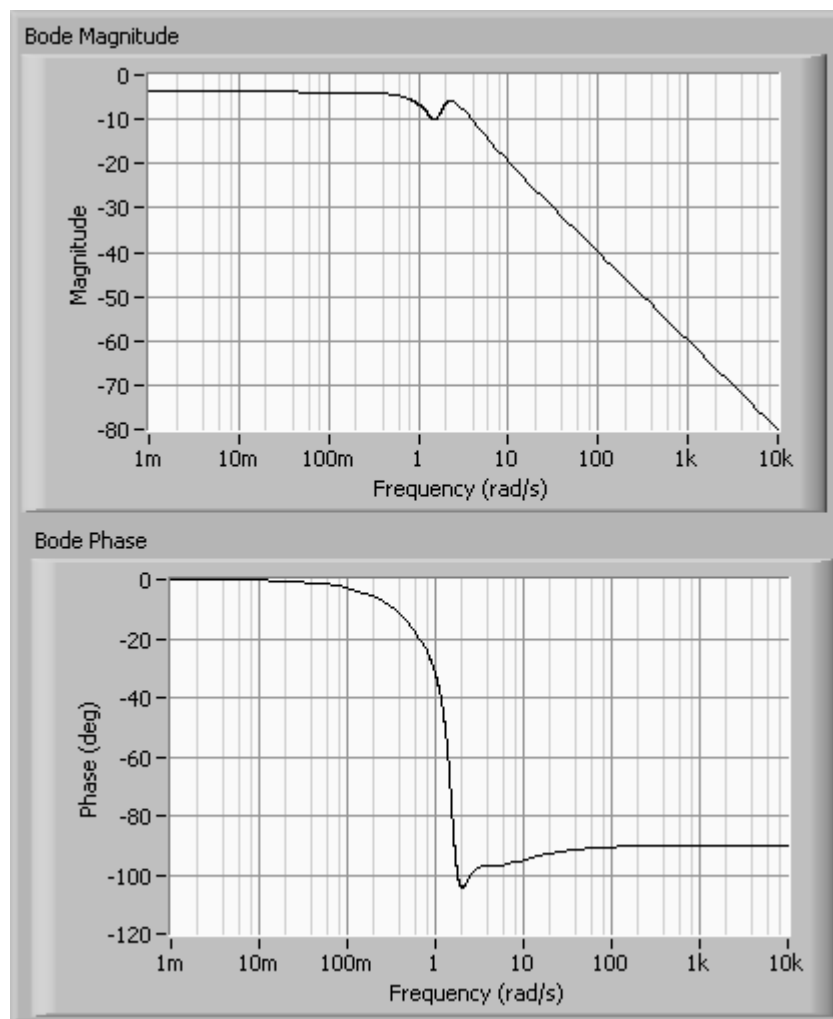


Fig. 4.3 Input – The Nyquist and Bode Plots for Frequency Response Data

The high frequency slope of the Bode magnitude plot is

$$-20db/decade$$

Thus  $n - m = 1$ .

The number of right hand plane zeros,

$$z_r = 2.$$

The required signature for stability is given in Eqn 4.4 and Eqn 4.5 is:

$$\sigma(\Pi) = (n - m) + 2z_r + 2 = 1 + 2 \cdot 2 + 2 = 7 \quad (4.7)$$

Since  $n - m$  is odd,

$$\{i_0 - 2i_1 + 2i_2 + \dots + (-1)^{l-1}2i_{l-1} + (-1)^l 2i_{l-1}\} \cdot (-1)^{l-1} j = n - m + 2z_r + 2 = 7 \quad (4.8)$$

At least 4 terms are required to satisfy the above, hence  $l \geq 4$ .

In Fig. 4.4 all the possible Strings of Integers generated are shown

		strings				
0		1	-1	1	-1	0
0		-1	1	-1	1	0
		0	0	0	0	0

Fig. 4.4: Strings of Integers Generated by LabVIEW

As shown in Fig. 4.5 the  $\text{Im}(\omega, k_p^*) = 0$  graph used to compute sets of stabilizing  $kp$  values is shown. From here values in Fig.4.6 can be computed and we observe that the frequency range over which plant data needs to be accurately known for estimating  $k_p$ ,

$k_i$  and  $k_d$  is  $[0.38, 0.57]$ . Also there are at the most three positive frequencies as solutions, which satisfies:

$$i_0 - 2i_1 + 2i_2 - 2i_3 = 7 \quad (4.9)$$

Hence,  $k_p$  must be chosen so that  $\text{Im}(\omega, k_p^*) = 0$  has exactly three positive real zeros.

The range of  $k_p$  that stabilizes the plant can be graphically determined by estimating the range over which the minimum number of intersections  $k_p$  makes is greater than or equal to the minimum number of intersections as shown in Fig. 4.7.

Fixing  $k_p = 0.4$  and computing the set of  $\omega$ 's that satisfies:

$$-\frac{\cos \phi(\omega) + \omega T \sin \phi(\omega)}{|P(j\omega)|} = 0.4 \quad (4.10)$$

The solution for  $\omega$  is:

$$\{\omega_1, \omega_2, \omega_3\} = \{1.74, 2.586, 3.102\} \quad (4.11)$$

Since the requirement  $i_0 - 2i_1 + 2i_2 - 2i_3 = 7$ , the strings satisfying this requirement is computed in 4.8

The solution of these inequalities gives the  $k_i - k_d$  co-ordinates for a specific value of  $k_p$ .

The stabilizing set of  $k_i - k_d$  values for  $k_p = 0.4$  are shown in Fig 4.8 and the 3-D

Representation of the entire stabilizing set is depicted in Fig. 4.9.

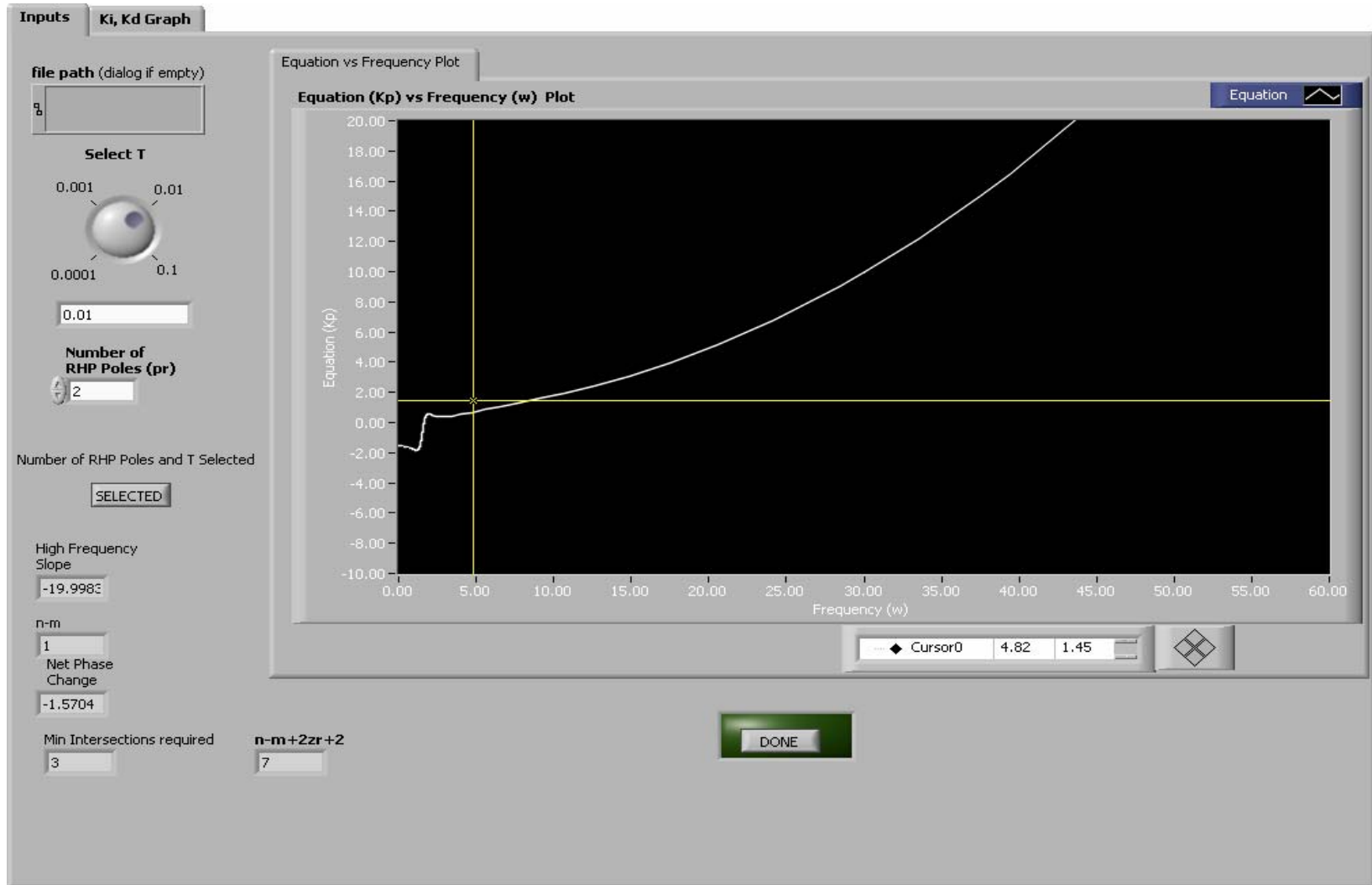


Fig. 4.5: Front Panel of VI – Graph Used to Compute Sets of Stabilizing  $k_p$  Values

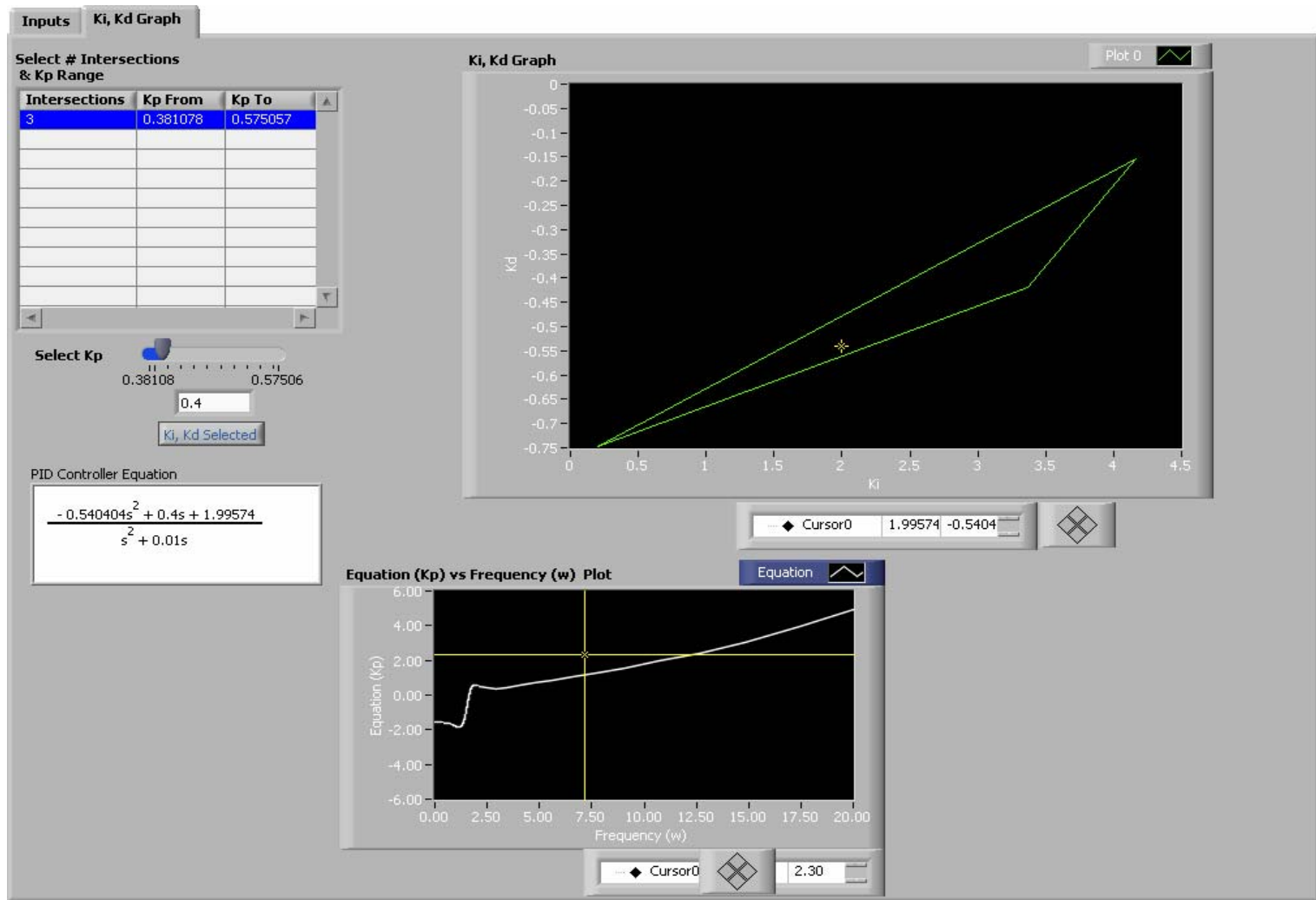


Fig. 4.6: Graph Showing Stabilizing  $k_i$  &  $k_d$  for Specific  $k_p$  Generated in Fig. 4.5

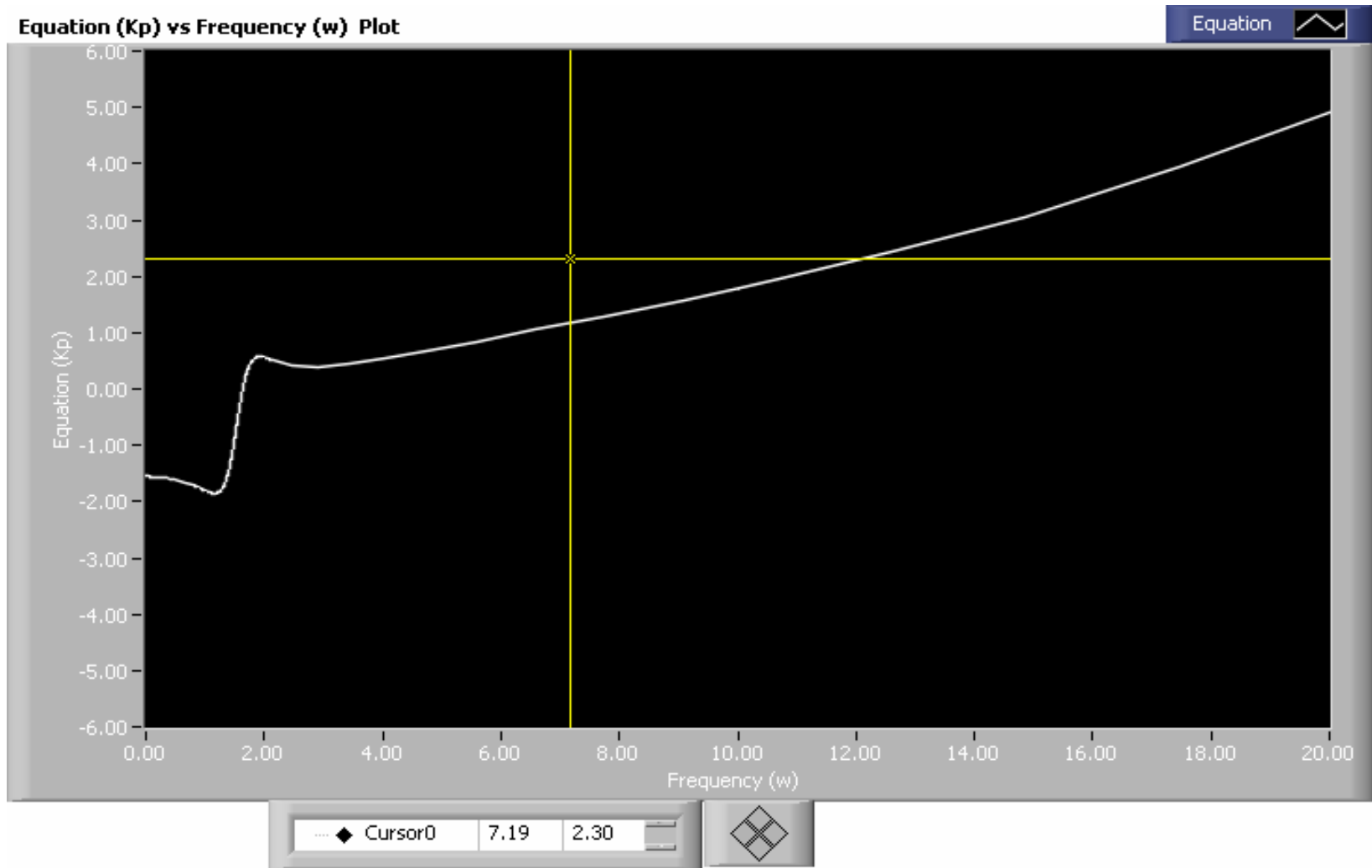


Fig. 4.7: Graph of  $k_p$  vs.  $\omega$  Generated by LabVIEW

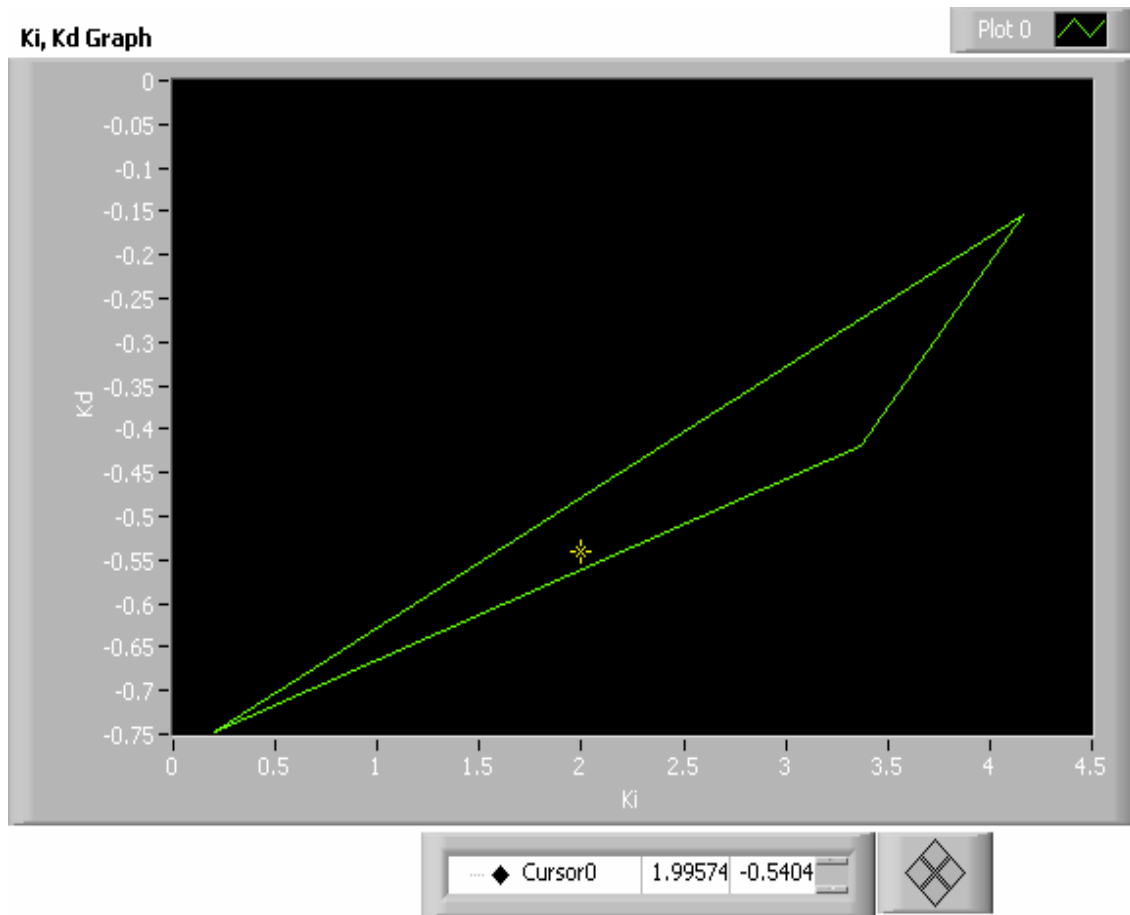


Fig. 4.8:  $k_i - k_d$  Stabilizing Range for  $k_p = 0.4$

The complete set of stabilizing PID gains for  $k_p = 0.4$  is shown in Fig. 4.8. And by sweeping over  $k_p$  we have the entire set of stabilizing PID gains as shown in Fig. 4.9.

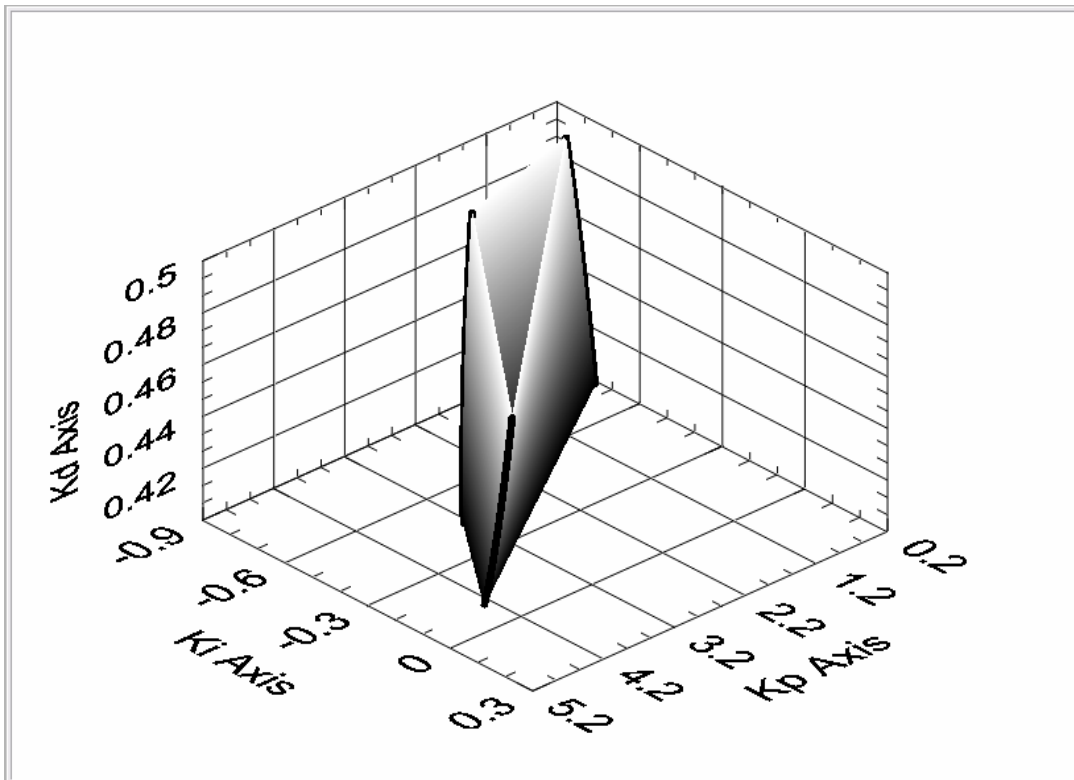


Fig. 4.9: 3-D Stabilizing Set of  $k_p$ ,  $k_i$  and  $k_d$



## CHAPTER V

### PID STABILIZATION OF SYSTEMS WITH TIME DELAY

Owing to the complex nature of systems with time delay, a single delay in the feedback loop representing delay in control actions or delayed measurements has been accounted for. This is modeled as  $e^{-Ls}$  where  $L$  represents the time in seconds. The following sections deal with First Order System and Arbitrary Order Systems with time delay. In case of First Order Plants with time delay, the complete set of stabilizing PID Controllers can be estimated. For Plants of higher order, sets of PID Controllers that cause instability in the system when time delay is introduced are computed and deleted from the stabilizing range that has also been computed in Chapter III.

#### A. First Order Stable Plant

##### 1. Theory

The system considered is a simple feedback control system where  $G(s)$  is given by:

$$G(s) = \frac{k}{1+Ts} e^{-Ls} \quad (5.1.1)$$

$C(s)$  is the PID controller where:

$$C(s) = k_p + \frac{k_i}{s} + k_d s \quad (5.1.2)$$

$k_p$  - Proportional Gain

$k_i$  - Integral Gain

$k_d$  - Derivative Gain.

When the time-delay of the plant,  $L=0$ , then the closed loop characteristic equation of the system is:

$$\delta(s) = (T + k k_d) s^2 + (1 + k k_p) s + k k_i \quad (5.1.3)$$

Closed loop stability in a second order polynomial is equivalent to all the coefficients having the same sign. Assuming the steady State gain  $k$  of the plant is positive these conditions are:

$$k_p > -\frac{1}{k}, k_i > 0 \text{ And } k_d > -\frac{T}{k} \quad (5.1.4)$$

OR

$$k_p < -\frac{1}{k}, k_i < 0 \text{ And } k_d < -\frac{T}{k} \quad (5.1.5)$$

When the time delay of the system is not zero, the Closed loop characteristic polynomial is defined as:

$$\delta(s) = (kk_i + kk_p s + kk_d s^2)e^{-Ls} + (1+Ts)s \quad (5.1.6)$$

The quasi polynomial is:

$$\delta^*(s) = e^{Ls} \delta(s) = kk_i + kk_p s + kk_d s^2 + (1+Ts)se^{Ls} \quad (5.1.7)$$

Substituting  $s = j\omega$  and separating the real and imaginary parts:

$$\delta_r(\omega) = kk_i - kk_d \omega^2 - \omega \sin(L\omega) - T\omega^2 \cos(L\omega) \quad (5.1.8)$$

$$\delta_i(\omega) = \omega[kk_p + \cos(L\omega) - T\omega \sin(L\omega)] \quad (5.1.9)$$

In an Open-Loop Stable Plant where  $T > 0$ , and assuming  $k > 0$  and  $L > 0$ ,

The imaginary part of  $\delta^*(j\omega)$  has only simple real roots if and only if:

$$-\frac{1}{k} < k_p < \frac{1}{k} \left[ \frac{T}{L} \alpha_1 \sin(\alpha_1) - \cos(\alpha_1) \right] \quad (5.1.10)$$

Where  $\alpha_1$  is a solution of the equation:

$$\tan(\alpha) = -\frac{T}{T+L} \alpha \quad (5.1.11)$$

in the interval  $(0, \pi)$ .

Defining  $z = L\omega$

For each value of  $k_p$  in the range, the necessary and sufficient conditions on  $k_i$  and

$k_d$  for the roots of  $\delta_r(L\omega)$  and  $\delta_i(L\omega)$  to interlace are the following set of inequalities:

$$\begin{aligned}
k_i &> 0 \\
k_d &> m_1 k_i + b_1 \\
k_d &< m_2 k_i + b_2 \\
k_d &> m_3 k_i + b_3 \\
k_d &< m_4 k_i + b_4
\end{aligned} \tag{5.1.12}$$

Where the parameters  $m_j$  and  $b_j$  for  $j=1,2,3,\dots$  are given by:

$$m_j \equiv m(z_j)$$

$$b_j \equiv b(z_j)$$

$$\text{And } m_j \equiv \frac{L^2}{z^2}$$

$$b(z) \equiv -\frac{L}{kz} \left[ \sin(z) + \frac{T}{L} z \cos(z) \right] \tag{5.1.13}$$

## 2. Algorithm / Block Diagram

Fig. 5.1.1 shows the algorithm that is used to calculate ranges of  $k_p$ ,  $k_i$  and  $k_d$ . The SubVI “*kp-SubVI*” calculates the stabilizing range of  $k_p$  as per Eqn 5.1.10. “*Feasible z's*” is the SubVI that is used to determine precise  $\omega$  values that are used to calculate the range of  $k_i$  and  $k_d$ . Within the “*k<sub>i</sub>-k<sub>d</sub> coeffs*” block, values of  $m(z_j)$  and  $b(z_j)$  are determined. These values are routed to the “*Inequality Solver*” SubVI where the final  $k_i$ - $k_d$  co-ordinates are determined.

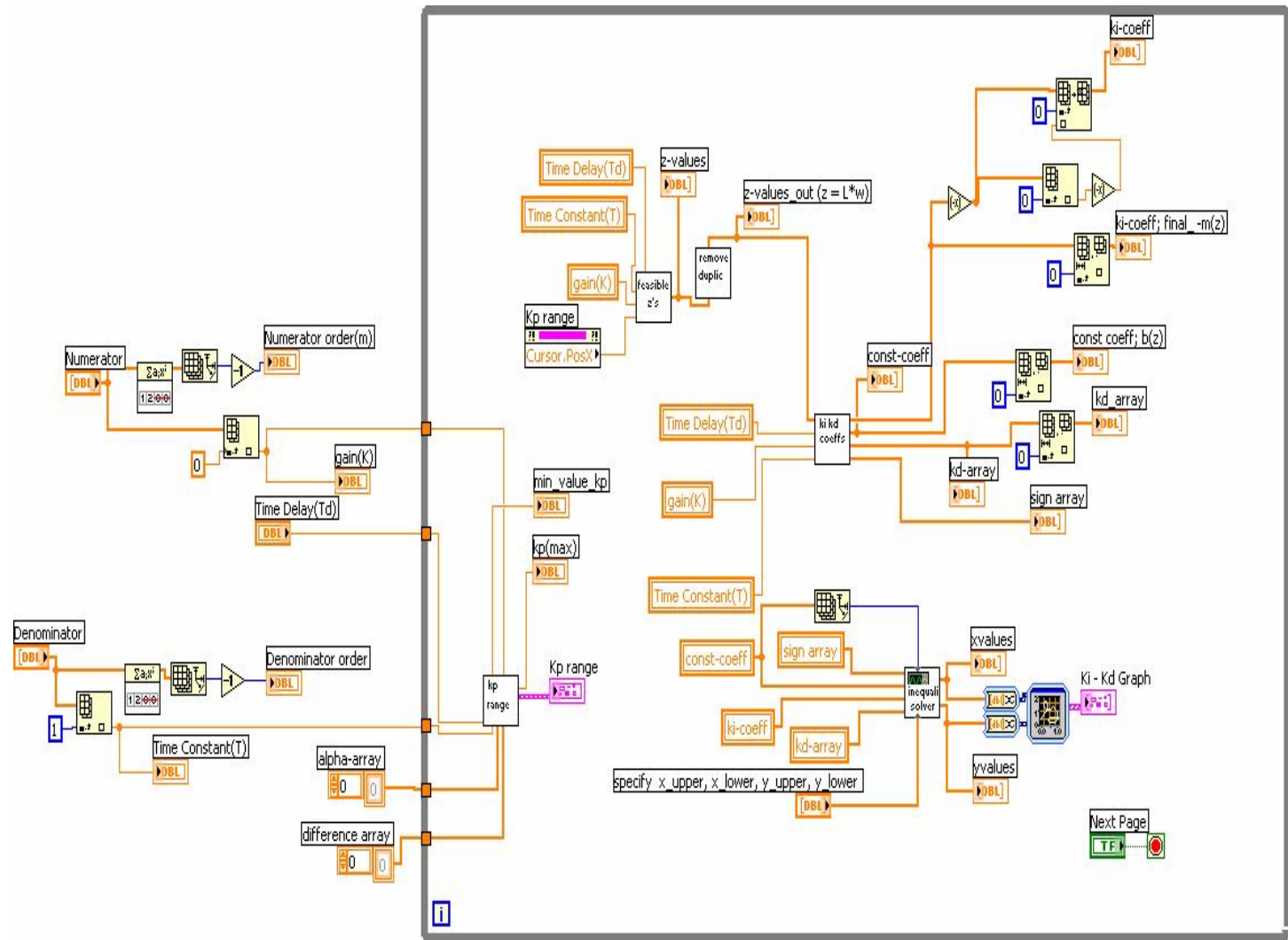


Fig. 5.1.1: Block Diagram of a First Order Stable Plant with Time Delay

Also, for each set of values in the stabilizing range the corresponding Gain and Phase Margin values are estimated.

### 3. Example

In the transfer function  $G(s) = \frac{k}{1+Ts} e^{-Ls}$ , the plant parameters are defined as :

$k = 1$ ,  $T = 2$  seconds and  $L = 4$  seconds

The quasi-polynomial is computed to be:

$$\delta^*(s) = k_i + k_p s + k_d s^2 + (2s^2 + s)e^{4s}$$

The solution to the general case has been implemented in LabVIEW, and the following results are obtained:

The solution to the equation  $\tan(\alpha) = -\frac{1}{3}\alpha$  is obtained in the interval  $(0, \pi)$ .

Here,  $\alpha_1 = 2.4556$ . Hence, the imaginary part of  $\delta^*(s)$  has only simple roots if and only if  $-1 < k_p < 1.5515$ . This example is shown in Fig 5.1.2 and Fig. 5.1.3. The 3-D representation of the stabilizing  $k_p$ ,  $k_i$  and  $k_d$  set is shown in Fig.5.1.4

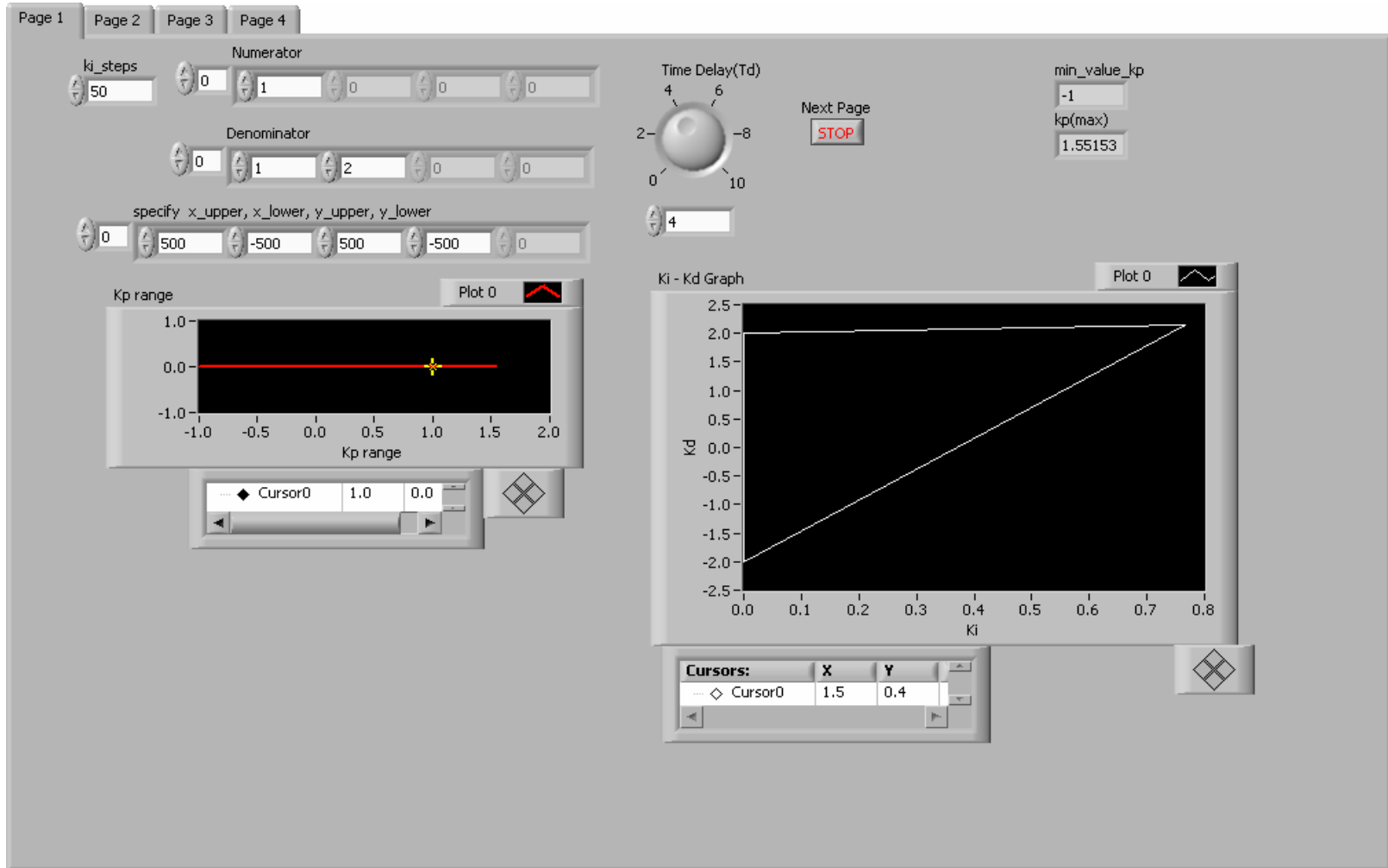


Fig. 5.1.2: Front Panel of the LabVIEW Program for Stable First Order System, for  $k_p$  Generation

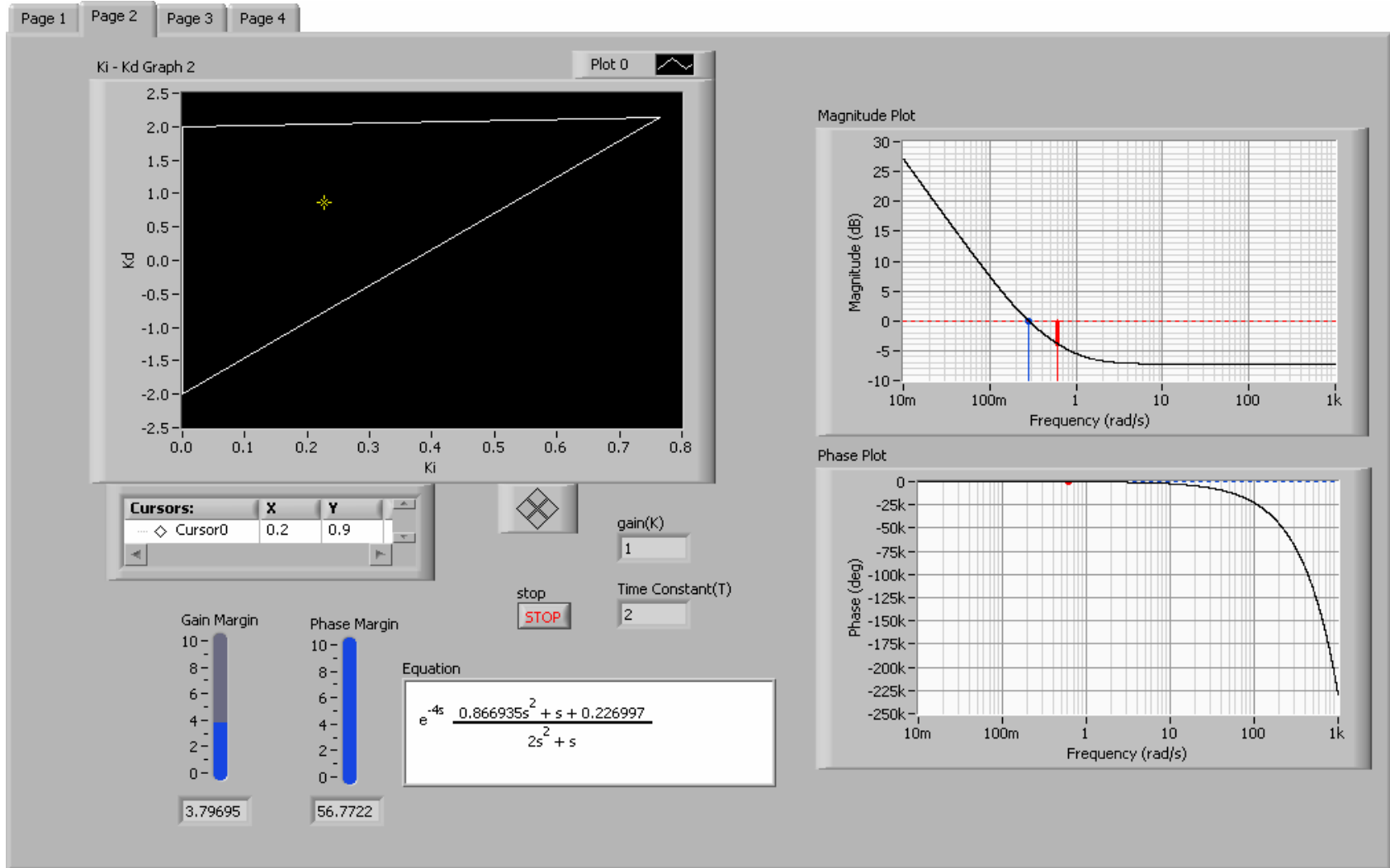


Fig. 5.1.3: Front Panel of the LabVIEW Program for Stable First Order System, for  $k_i$  &  $k_d$  Generation



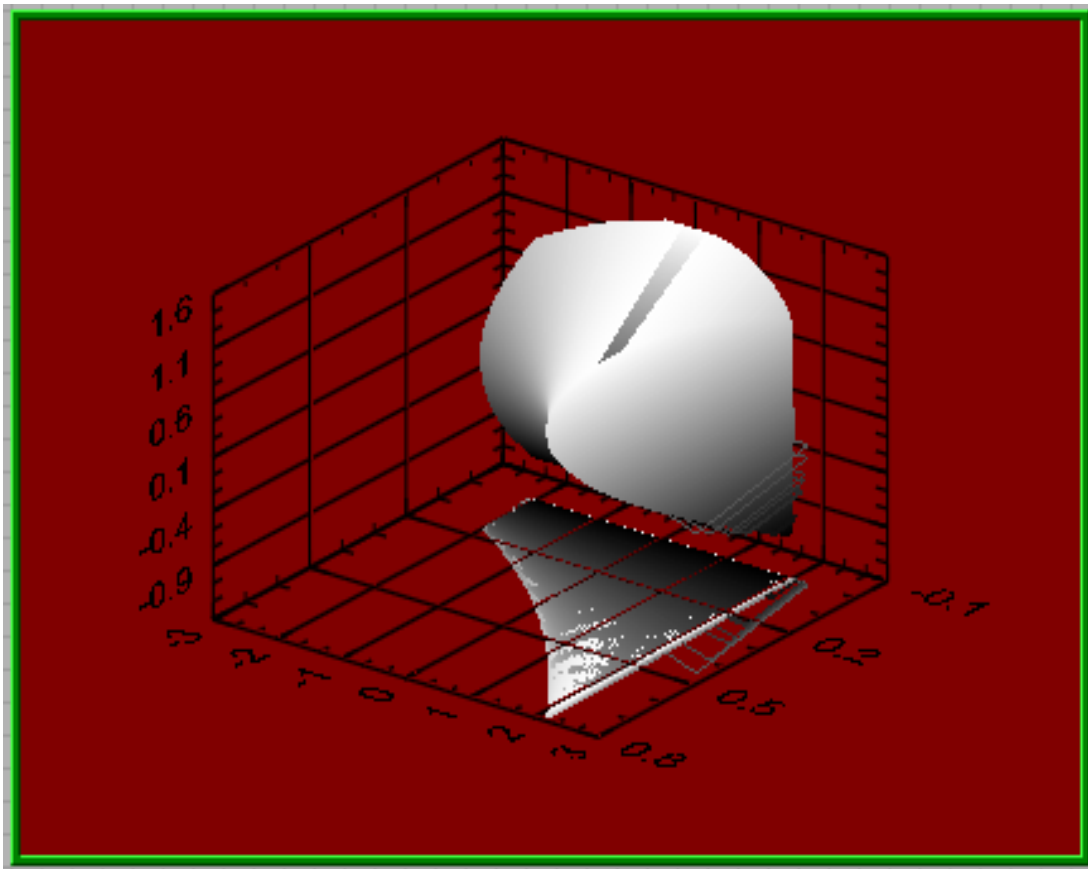


Fig. 5.1.4: 3D Representation of Stabilizing  $k_p$ ,  $k_i$  and  $k_d$

## B. Open Loop Unstable Plant

### 1. Theory

For  $\left|\frac{T}{L}\right| > 0.5$  the imaginary part of  $\delta^*(j\omega)$  have simple real roots if and only if:

$$\frac{1}{k} \left[ \frac{T}{L} \alpha_1 \sin(\alpha_1) - \cos(\alpha_1) \right] < k_p < -\frac{1}{k} \quad (5.2.1)$$

Where  $\alpha_1$  is the solution of the equation:

$$\tan(\alpha) = -\frac{T}{T+L} \alpha \quad (5.2.2)$$

in the interval  $(0, \pi)$ .

In the special case of

$$\left|\frac{T}{L}\right| = 1, \alpha_1 = \frac{\pi}{2}. \quad (5.2.3)$$

For  $\left|\frac{T}{L}\right| \leq 0.5$ , the roots of the imaginary part of  $\delta^*(j\omega)$  are not all real.

Similar to Section 5.1

Defining  $z = L\omega$

For each value of  $k_p$  in the range, the necessary and sufficient conditions on  $k_i$  and  $k_d$  for the roots of  $\delta_r(L\omega)$  and  $\delta_i(L\omega)$  to interlace are the following set of inequalities:

$$\begin{aligned}
 k_i &< 0 \\
 k_d &< m_1 k_i + b_1 \\
 k_d &> m_2 k_i + b_2 \\
 k_d &< m_3 k_i + b_3 \\
 k_d &> m_4 k_i + b_4
 \end{aligned} \tag{5.2.4}$$

Where the parameters  $m_j$  and  $b_j$  for  $j=1,2,3,..$  are given by:

$$m_j \equiv m(z_j)$$

$$b_j \equiv b(z_j)$$

$$\text{And } m_j \equiv \frac{L^2}{z^2}$$

$$b(z) \equiv -\frac{L}{kz} \left[ \sin(z) + \frac{T}{L} z \cos(z) \right] \tag{5.2.5}$$

## 2. Algorithm / Block Diagram

The block diagram and the SubVI's are similar to those built in Section 5.1. The only difference is in the “*Inequality Solver*” SubVI, where the inequality signs are reversed as given by Eqn 5.2.2.

The final Front Panel of the VI is given in Fig 5.2.1 and Fig. 5.2.2

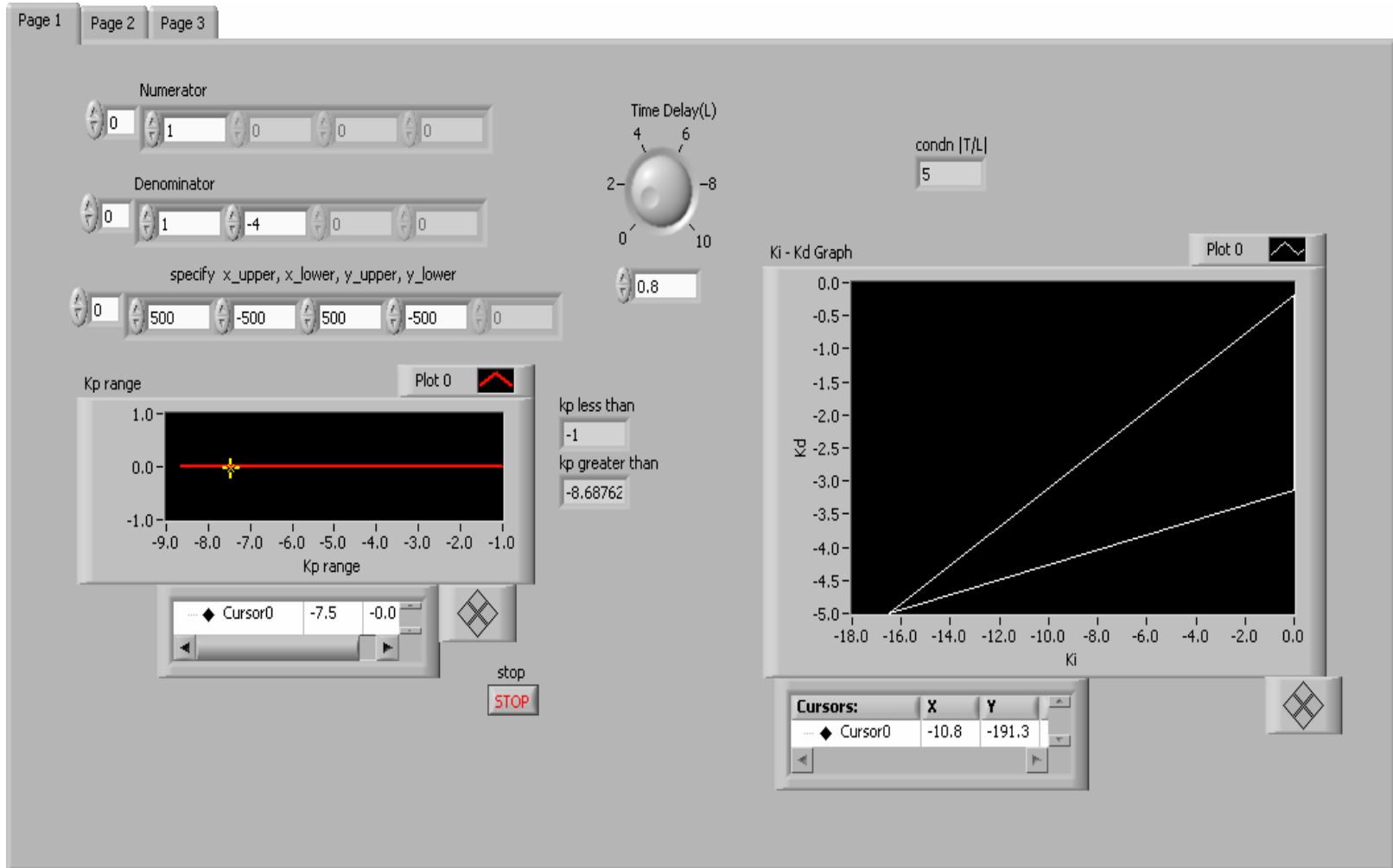


Fig. 5.2.1: Front Panel of the LabVIEW Program for Unstable First Order System, for  $k_p$  Generation

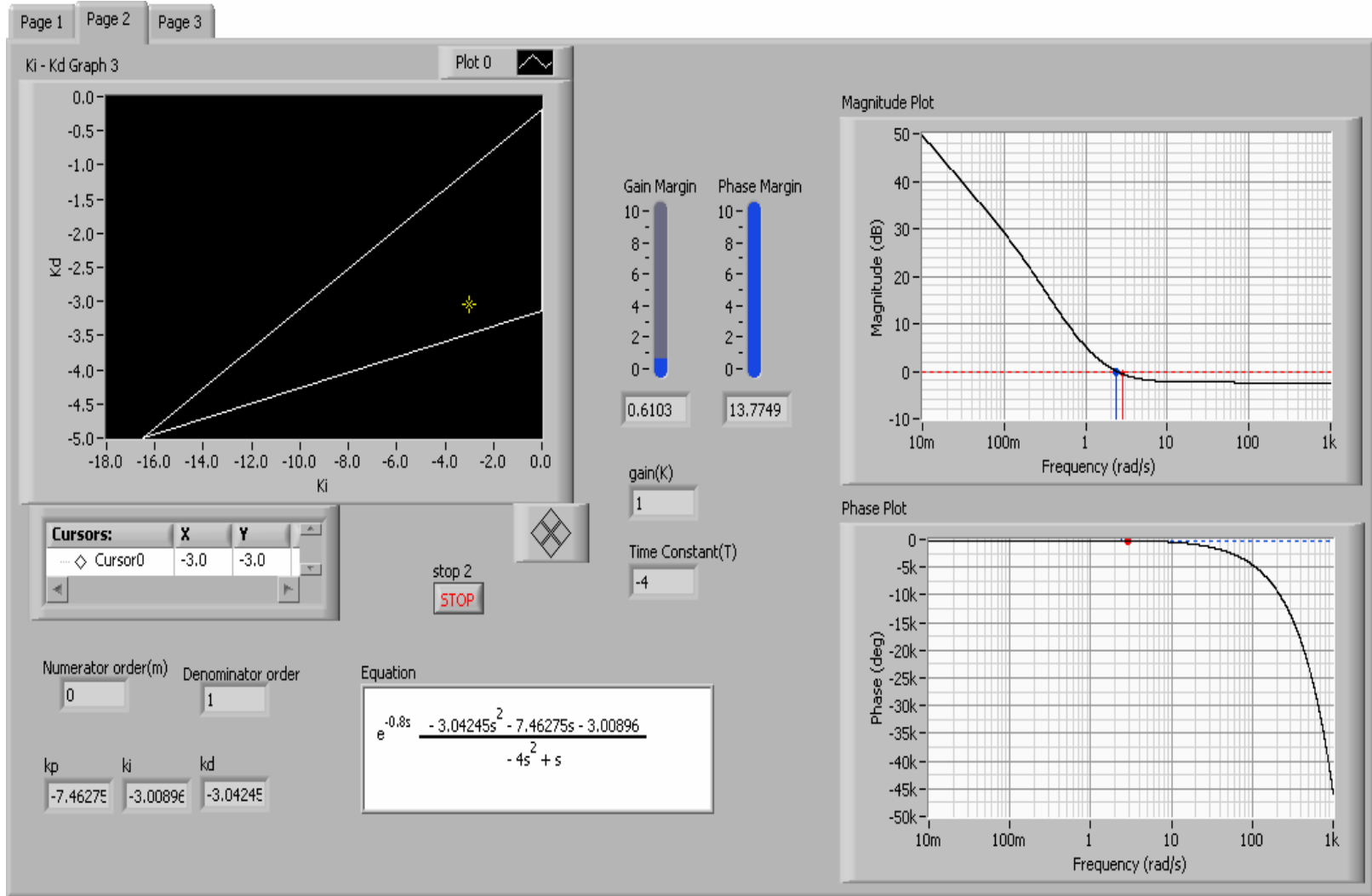


Fig. 5.2.2: Front Panel of the LabVIEW Program for Unstable First Order System, for  $k_i$  &  $k_d$  Generation

## C. Stabilization of arbitrary LTI Systems with Time Delay

### 1. Theory

The feedback system,  $G(s)$  with a time delay of  $L$  seconds is given by:

$$G(s) = G_0(s)e^{-Ls} = \frac{N(s)}{D(s)}e^{-Ls} \quad (5.3.1)$$

Where  $N(s)$  and  $D(s)$  are polynomials with real coefficients. The PID controller is given by the following transfer function:

$$C(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_d s^2 + k_p s + k_i}{s} \quad (5.3.2)$$

To determine the set of  $(k_p, k_i, k_d)$  that stabilize the closed loop system:

$$H(s) = C(s)G(s) = C(s)G_0(s)e^{-Ls} = H_0(s)e^{-Ls} \quad (5.3.3)$$

Where:

$$\begin{aligned} H_0 &= C(s)G_0(s) \\ &= \frac{k_d s^2 + k_p s + k_i}{s} \frac{N(s)}{D(s)} \end{aligned}$$

$$\begin{aligned}
&= (k_d s^2 + k_p s + k_i) \frac{N(s)}{sD(s)} \\
&= (k_d s^2 + k_p s + k_i) R_0(s)
\end{aligned} \tag{5.3.4}$$

$$\text{And } R_0(s) = \frac{N(s)}{sD(s)}.$$

The Phase and Magnitude conditions are given by:

$$\arg \left[ (k_i - k_d \omega^2 + j k_p \omega) R_0(j\omega) \right] - L\omega = -\pi$$

And

$$\left| (k_i - k_d \omega^2 + j k_p \omega) R_0(j\omega) \right| = 1 \tag{5.3.5}$$

These conditions are reduced to:

$$L(\omega, k_p, k_i, k_d) = \frac{\pi + \arg \left\{ \left[ (k_i - k_d \omega^2) + j k_p \omega \right] R_0(j\omega) \right\}}{\omega} \tag{5.3.6}$$

$$k_i - k_d \omega^2 = \pm \sqrt{\frac{1}{|R_0(j\omega)|^2} - (k_p \omega)^2} \tag{5.3.7}$$

$$M(\omega) = \frac{1}{|R_0(j\omega)|^2} - (k_p \omega)^2 \tag{5.3.8}$$

It can also be rewritten as:

$$k_i - k_d \omega^2 = \pm \sqrt{M(\omega)} \quad (5.3.9)$$

$\omega$  values having  $M(\omega) \geq 0$  only need to be considered when computing  $S_L$

After Substitutions,

$$L(\omega, k_p, k_i, k_d) = L(\omega) = \frac{\pi + \arg \left\{ \left[ \pm \sqrt{M(\omega)} + j k_p \omega \right] R_0(j\omega) \right\}}{\omega} \quad (5.3.10)$$

## 2. Algorithm / Block diagram

The following steps are followed for ease of developing the final software:

1. Compute  $S_0$  - The complete set of  $k_p$ ,  $k_i$  and  $k_d$  in the delay free system.
2. Find  $S_N$  defined as:

$$S_N = \left\{ (k_p, k_i, k_d) \in R^3 \mid \lim_{s \rightarrow \infty} \left| \frac{(k_d s^2 + k_p s + k_i) N(s)}{s D(s)} \right| \geq 1 \right\} \quad (5.3.11)$$

If  $\deg[N(s)] > \deg[D(s)] - 1$ ,  $S_N = R^3$ , which means  $S_R = \phi$

If  $\deg[N(s)] < \deg[D(s)] - 1$ ,  $S_N = \phi$

If  $\deg[N(s)] = \deg[D(s)] - 1$ , then

$$S_N = \left\{ (k_p, k_i, k_d) \in R^3 \mid |k_d| \geq \left| \frac{a_q}{b_{q-1}} \right| \right\}$$

Where  $a_q, b_{q-1}$  are the leading coefficients of  $D(s)$  and  $N(s)$  respectively.



3. Compute  $S_1 = S_0 \setminus S_N$
4. For a fixed  $k_p$ , determine the set  $S_{R,k_p}$  as follows:

(a) Determine the sets  $\Omega^+$  and  $S_{L,k_p}^+$  :

$$\Omega^+ = \left\{ \omega \mid \omega > 0 \ \& \ M(\omega) \geq 0 \ \& \ L(\omega) = \frac{\pi + \arg \left\{ \left[ \sqrt{M(\omega)} + jk_p \omega \right] R_0(j\omega) \right\}}{\omega} \leq L_0 \right\}$$

$$S_{L,k_p}^+ = \left\{ (k_i, k_d) \mid (k_i, k_d) \notin S_{N,k_p} \ \& \ \exists \omega \in \Omega^+ \text{ such that } k_i - k_d \omega^2 = \sqrt{M(\omega)} \right\}$$

(5.3.12)

Where  $S_{L,k_p}^+$  is a set of straight lines in the  $(k_i, k_d)$  space.

(b) Determine the sets  $\Omega^-$  and  $S_{L,k_p}^-$  :

$$\Omega^- = \left\{ \omega \mid \omega > 0 \ \& \ M(\omega) \geq 0 \ \& \ L(\omega) = \frac{\pi + \arg \left\{ \left[ -\sqrt{M(\omega)} + jk_p \omega \right] R_0(j\omega) \right\}}{\omega} \leq L_0 \right\}$$

$$S_{L,k_p}^- = \left\{ (k_i, k_d) \mid (k_i, k_d) \notin S_{N,k_p} \ \& \ \exists \omega \in \Omega^- \text{ such that } k_i - k_d \omega^2 = -\sqrt{M(\omega)} \right\}$$

(5.3.13)

5. Compute

$$S_{L,k_p} = S_{L,k_p}^+ \cup S_{L,k_p}^-$$

And

$$S_{R,k_p} = S_{1,k_p} \setminus S_{L,k_p} \tag{5.3.14}$$

6. By sweeping over  $k_p$ , we will have the complete set of PID Controllers that stabilizes all the plants with delay up to  $L_0$  :

$$S_R = \bigcup_{k_p} S_{R,k_p} \quad (5.3.15)$$

The complete set of stabilizing  $k_i$  and  $k_d$  for a specific value  $k_p$ ,  $S_0$  is computed as shown in Chapter III. The sets that are deleted from this set is calculated from the steps outlined above. The first step is to find  $\sqrt{M(\omega)}$ ,  $\Omega^-$  and  $\Omega^+$ . This is shown in Fig. 5.3.1. From the complete set of x and y Co-Ordinates ( $k_i$  and  $k_d$  Co-Ordinates), the Plant Transfer Function, Time Delay ( $L_0$ ) and the  $k_p$  cursor position are fed into the SubVI  $\sqrt{M(\omega)}$ .

Fig. 5.3.2 shows the Block Diagram of the SubVI in detail.  $R_0(s)$  is estimated as shown in Eqn 5.3.4, in the SubVI “array value”.  $M(\omega)$  is then calculated for a fixed value of  $k_p$  per Eqn.5.3.8. For each value of  $\omega$ , these values are fed into the SubVI “ $w_s$ ,  $L_w$  &  $M_w$ ’s. Within this block the values of  $L(\omega)$  Corresponding to  $\pm\sqrt{M(\omega)}$  is calculated, and tested to satisfy the condition:  $L(\omega) \leq L_0$ . If this condition is false, then the roots cause instability, hence only those values of  $\omega$  are finally put into the array that satisfies

the above condition, and corresponding to each selected value of  $\omega$  are the values of  $L(\omega)$  and  $M(\omega)$  associated with it, as are the outputs of the SubVI.

Fig. 5.3.3 computes final columns of probable  $k_i$  and  $k_d$  Co-Ordinates and provides values to the Block diagram shown in Fig. 5.3.4 where the second part of the calculations for estimating the stable set for an arbitrary order system with time delay is computed.

The inputs to this block are the previously calculated values -  $\sqrt{M(\omega)}$ ,  $\Omega^-$  and  $\Omega^+$ ,  $k_i$  and  $k_d$  Co-Ordinates from the delay free plant. The sets  $S_{L,k_p}^+$  and  $S_{L,k_p}^-$  are estimated from the Eqn 5.3.12 and Eqn 5.3.13. Each of the estimated values are fed in one at a time, and the SubVI “*coln caln*” determines if that value of  $(k_i, k_d)$  causes instability or not.

The output is an array with the Co-Ordinates of  $(k_i, k_d)$  that causes instability, and these sets of lines are deleted from the original set of  $(k_i, k_d)$  without time delay.

The Front Panel of the VI is shown in Fig. 5.3.5 and Fig. 5.3.6. In Fig. 5.3.5, the Transfer Function of the plant and the time delay of the plant are entered as the inputs.

The  $(k_i, k_d)$  range is computed for the Delay Free System, and the corresponding sets that cause instability are deleted from this range. The final result is shown in Fig.5.3.7

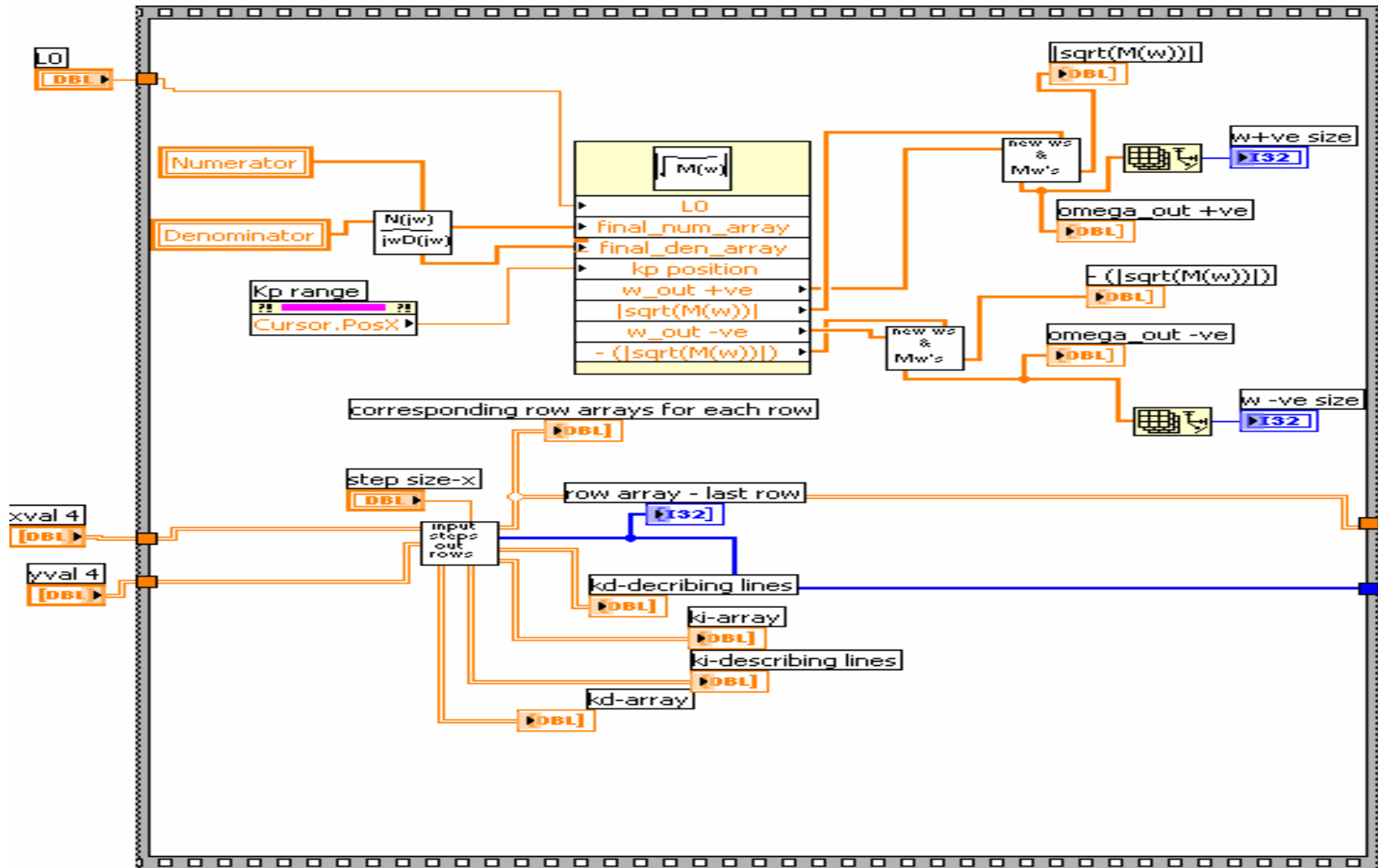


Fig. 5.3.1: Block Diagram Part 1 for Arbitrary Order Time Delay System

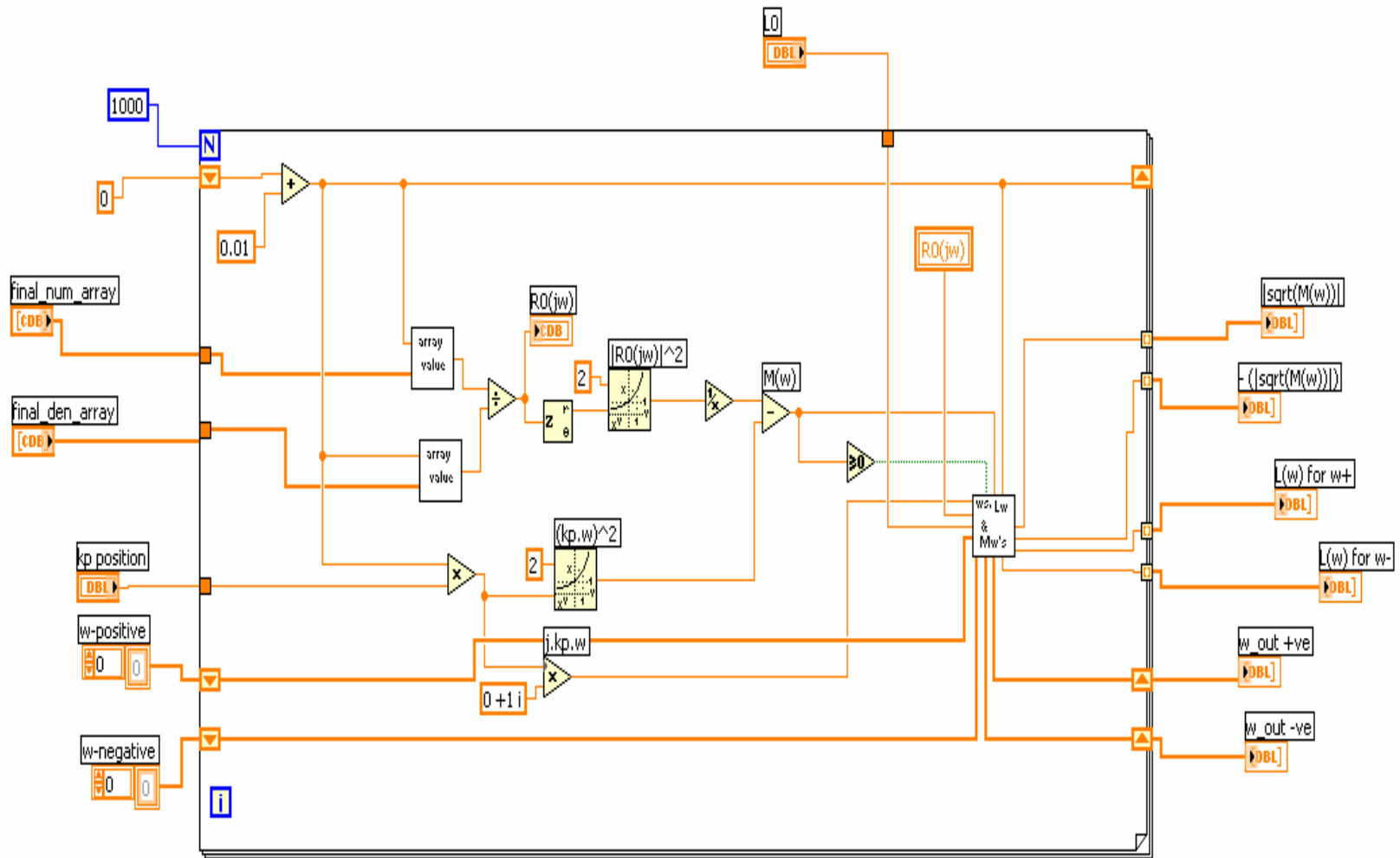


Fig.5.3.2: Block Diagram Showing  $\sqrt{M(\omega)}$

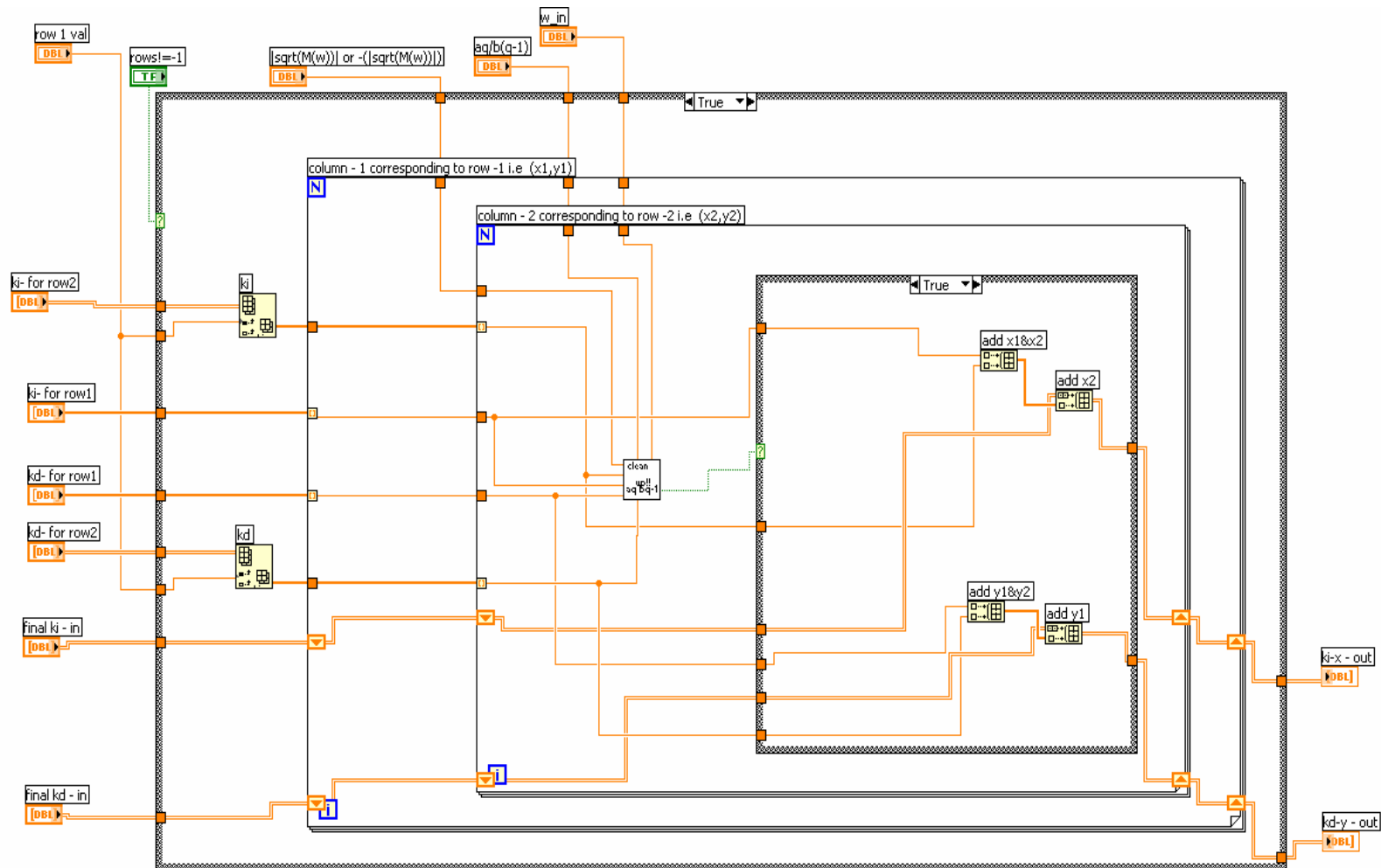


Fig.5.3.3 Columns Calculation

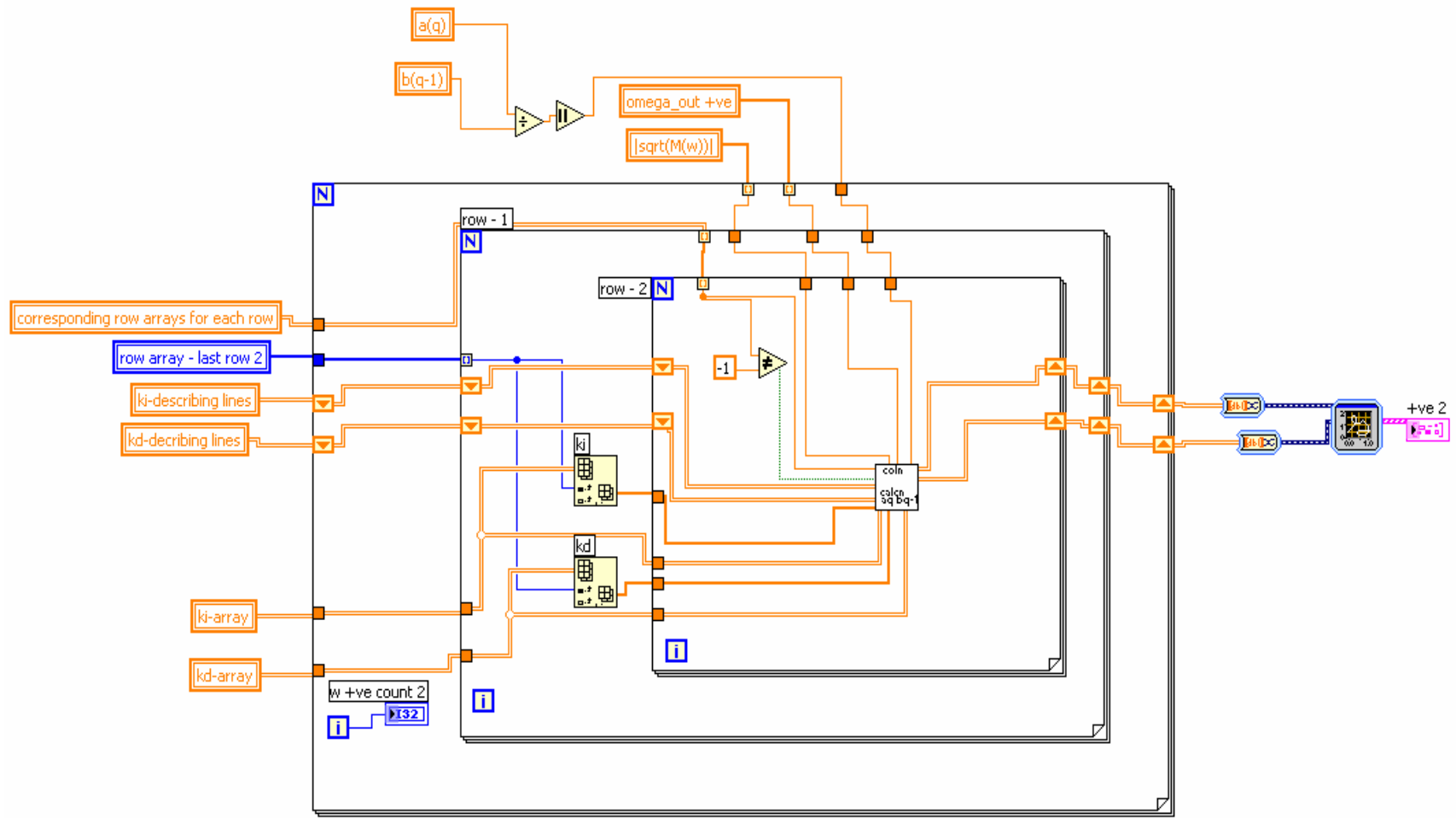


Fig. 5.3.4: Block Diagram Part 2 for Arbitrary Order Time Delay System

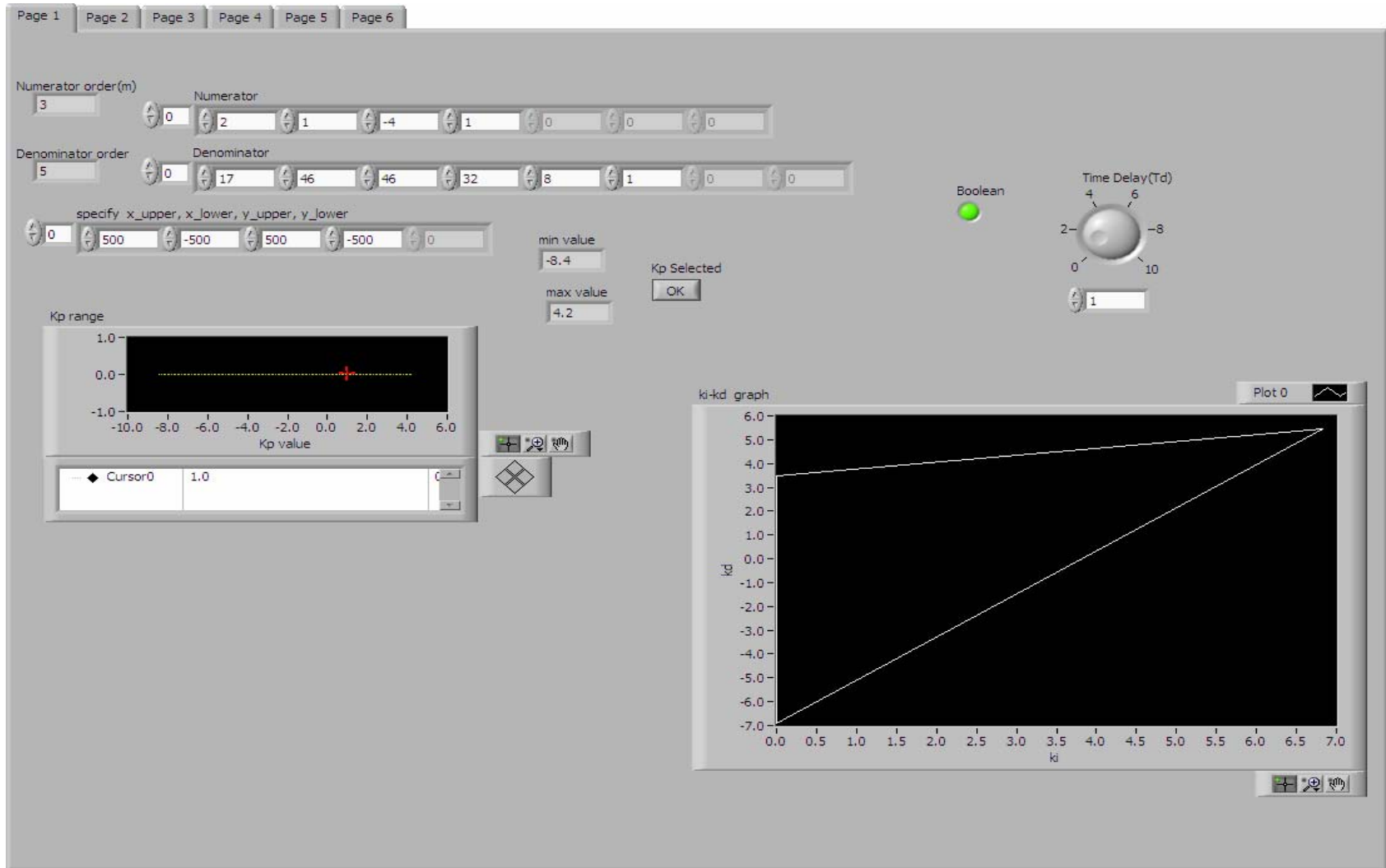


Fig. 5.3.5: Front Panel of Time Delay System of Arbitrary Order - Page 1



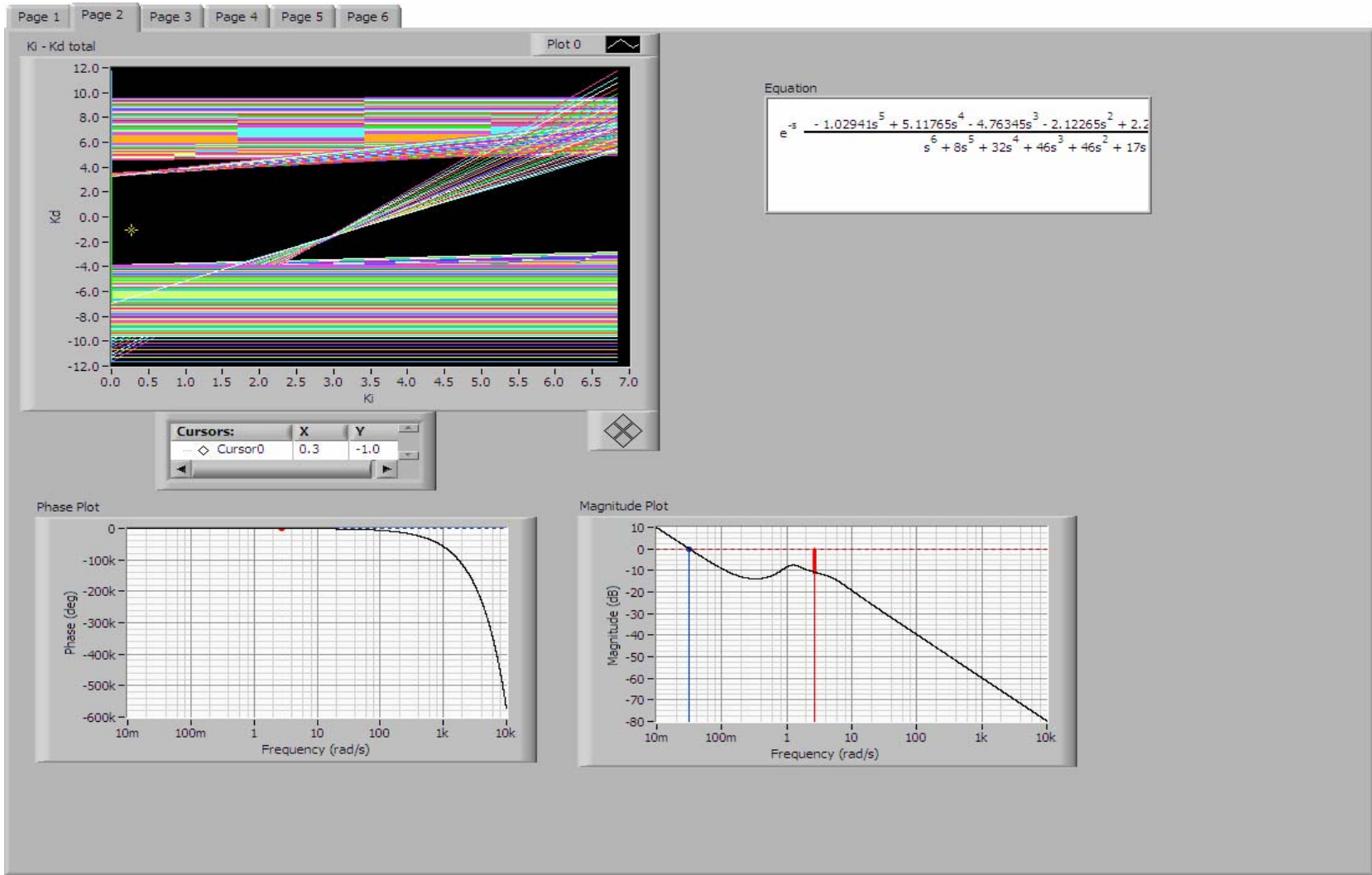


Fig. 5.3.6: Front Panel of Time Delay System of Arbitrary Order - Page 2

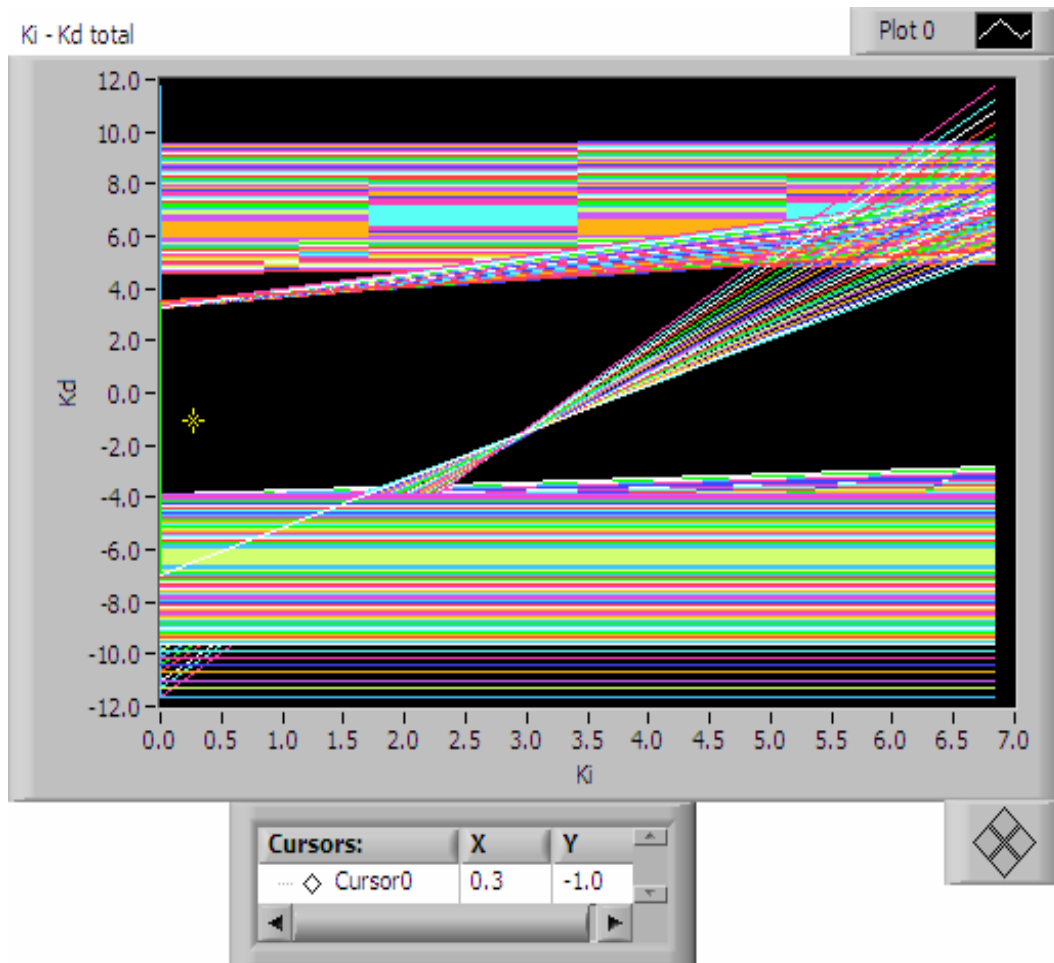


Fig. 5.3.7: Complete Set with  $(k_i, k_d)$  Lines Causing Instability Removed

## CHAPTER VI

### OPTIMIZATION OF STABILIZING SETS OF PID CONTROLLERS

#### A. Introduction

PID Controller sets are optimized to project subsets that are based on user requirements. The program has been developed to allow the user to streamline the controller sets based on the performance index specifications of Gain Margin, Phase Margin, Rise Time, Peak Time, Settling Time and Overshoot. This a very useful result as the user is now able pick points in the PID Controller sub-space that fulfils his requirements.

#### B. Performance Index Specifications

Gain Margin – The change in open loop gain required to make the system unstable

Phase Margin - The change in open loop phase shift required to make a closed loop system unstable.

Rise Time – The time taken by the system to rise from 10% to 90% of its output value

Peak Time – Time taken by the system to reach its maximum overshoot point

Settling Time – Time taken for system transients to decay to a small value. The measure used in the program is 1%

Overshoot – The maximum amount the system overshoots its final value divided by its final value (expressed as a percentage)

### C. Algorithm

The program scans over a user-defined grid of  $k_i - k_d$  values for a specific value of  $k_p$ , and captures final subsets that meet the user performance index specifications.

The user can also comb the  $k_i - k_d$  plane at a specific  $k_p$  value to find subsets that simultaneously satisfy performance index specifications.

### D. Final Program - Front Panel of the VI

Fig. 6.1 shows the tool that is used to scan the  $k_i - k_d$  plane for a subset satisfying the performance index at a value chosen by the user. In Fig. 6.1 Gain Margin is chosen and the value is set to 1dB. The magnified result is shown in Fig. 6.2. Page 2 of the VI is shown in Fig. 6.3. Here also, the user can chose among all the performance indices available to him. In this case, Phase Margin is set to 30 deg. The upper Left Hand Side shows the result of the scan completed. In Fig. 6.4 the combined result of pages 1 & 2, i.e. all the points that satisfy both the criterion encountered so far is shown. Similarly Page 3 offers users the complete set of Performance Indices to choose to scan, and displays the final comprehensive result on the Lower Right Hand Corner as shown in Fig. 6.5. This result is magnified in Fig. 6.6.

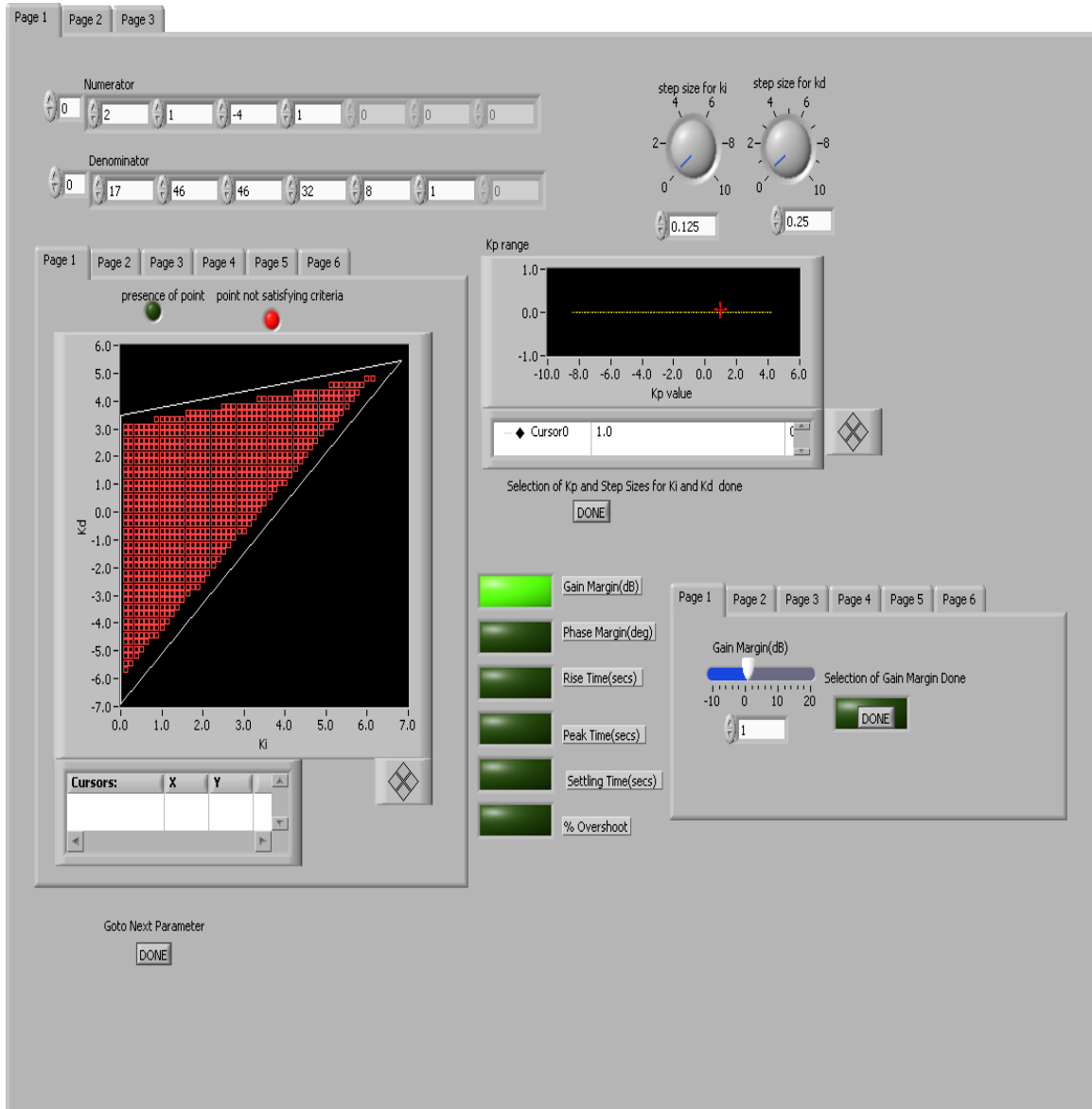


Fig 6.1: Front Panel of VI – Page 1

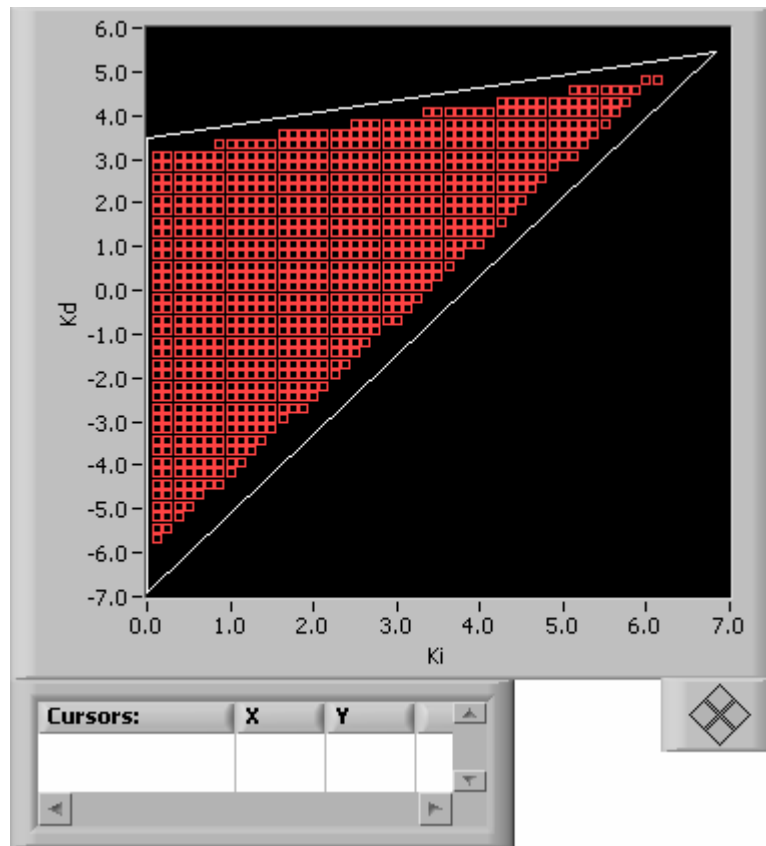


Fig. 6.2: Result Showing Subset with Gain Margin = 1dB

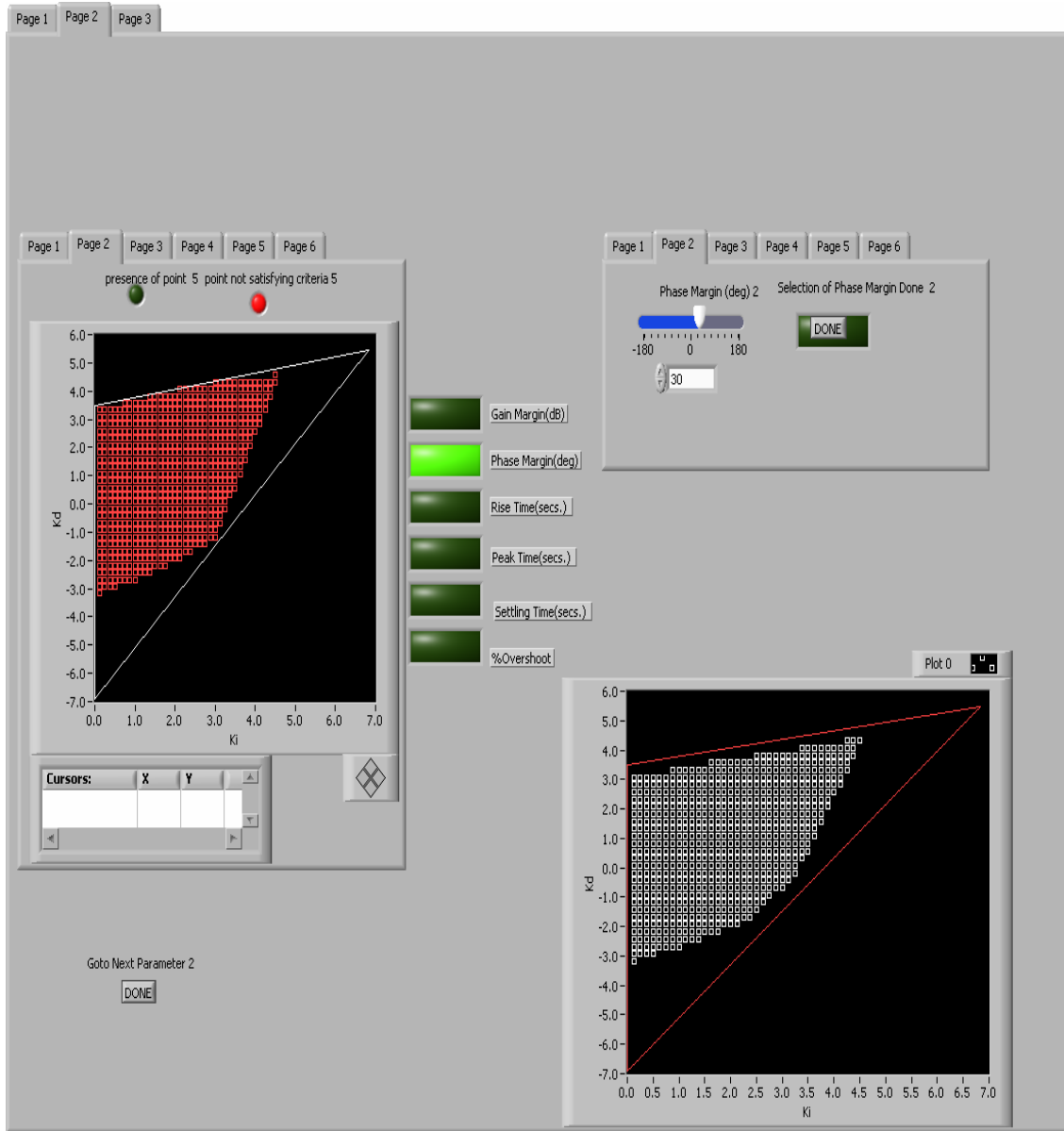


Fig. 6.3: Front Panel of VI – Page 2

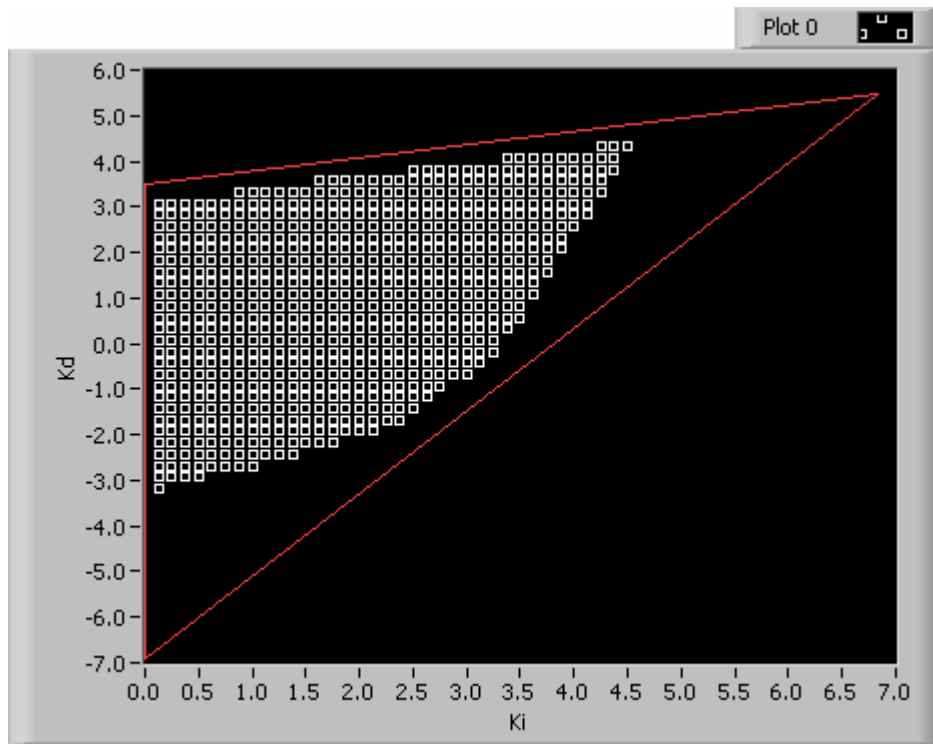


Fig. 6.4: Combination of Gain Margin = 1dB, Phase Margin = 30 deg



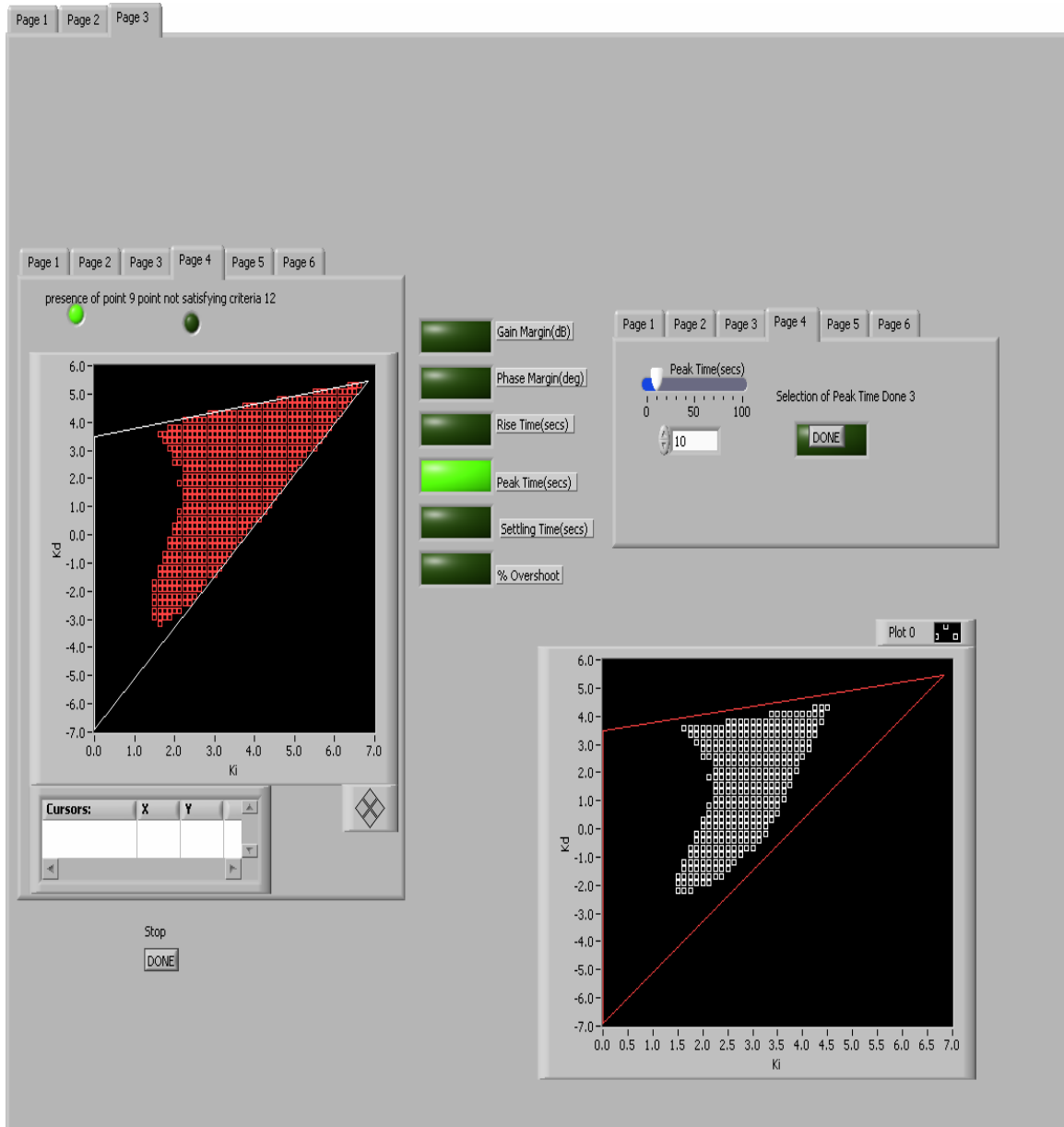


Fig. 6.5: Front Panel of VI – Page 3

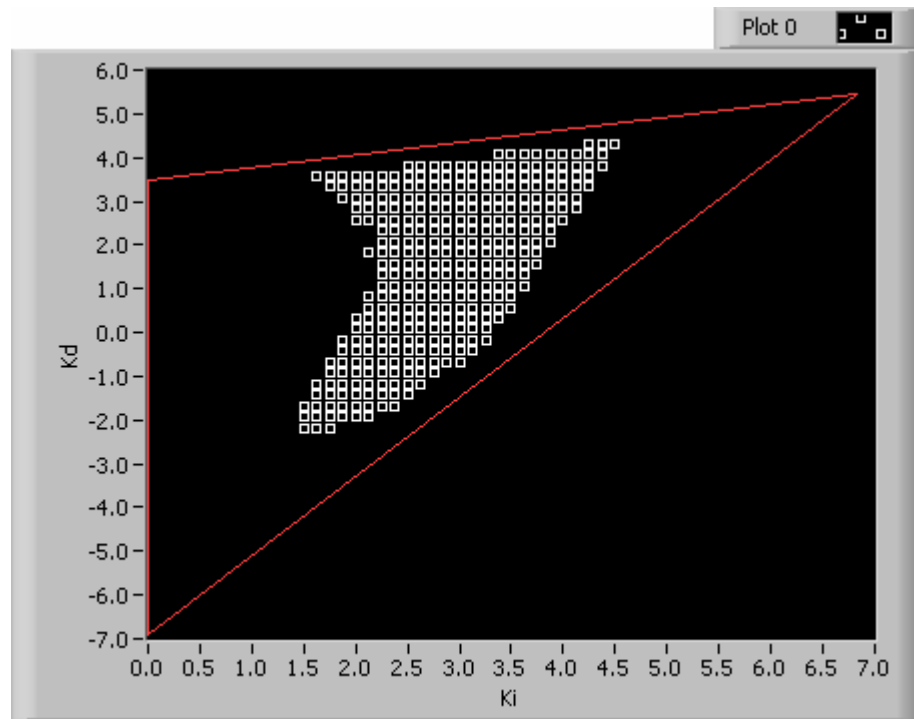


Fig. 6.6: Combination of Gain Margin = 1dB, Phase Margin = 30 deg & Peak Time = 10secs

## **CHAPTER VII**

### **CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS**

#### **A. Summary**

The primary objective of the thesis was to present an overview of a software tool that has been developed based on of Control System algorithms that are very useful in gauging the stability of systems. The thesis presented a very practical and novel approach to the computation of complete sets of stabilizing Proportional, Integral and Derivative Controllers when information available is of different kinds.

Further, the use of an automated software eliminates the necessity of an in – depth understanding of the subject, and is made accessible to engineers with a basic understanding of the Control System problem they need to solve.

#### **B. Future Work**

A very useful and practical application of the developed software is in the industry, where the frequency response of the system can be used in calculating complete ranges of Proportional, Integral and Derivative gains that stabilize the system. Compact RIO is an instrument that can be programmed to get the frequency response from a motor drive.

It can also function as a PID Controller. Compact RIO can be remotely monitored by the LabVIEW software.

Hence, the following can be used to stabilize the system in closed loop after getting the open loop frequency response of the system, eg: A DC Motor Drive

- Get Open Loop frequency response from the DC Motor Drive
- Compact RIO will generate complete stabilizing sets of Proportional Gains in real time
- User selects values of Proportional gains from the entire range generated by the LabVIEW program
- Embed the values in the PID Controller (Compact RIO) to stabilize the System

Compact Rio allows collection of data, calculations and implementation to be carried out real time. Once the values of the PID Controller are established, values are set in the PID Controller and the user can see the system stabilized in real time using the values he selects.

Once the setup is complete, control can be established remotely though internet. Hence, the user can remotely run the utility containing the program get the frequency response data set online, select the values of Proportional Gains he wants, and view the stabilizing effect.

## REFERENCES

- 1) H. Xu, A. Datta and S. P. Bhattacharyya, "Computation of All Stabilizing PID Gains for Digital Control Systems," *IEEE Transactions on Automatic Control*, vol. AC-46, pp. 647-652, April 2001.
- 2) A. Datta, M. T. Ho and S. P. Bhattacharyya, *Structure and Synthesis of PID Controllers*. New York: Springer-Verlag Inc., 2000.
- 3) G.F. Franklin, J. David Powell, A. Emami-Naeini, *Feedback Control of Dynamic Systems*. 5th edition, Eaglewood Cliffs, NJ: Prentice Hall. October, 2005.
- 4) G.J. Silva, A. Datta, S. P. Bhattacharyya, *PID Controllers for Time Delay Systems*. Boston: Birkhäuser. December, 2004
- 5) L.H.Keel, S.P.Bhattacharyya, "PID Controller Synthesis: Free of Analytical Models" in *16th IFAC World Congress*. Prague, Czech Republic. July 2005.

## VITA

Name : Indu Ramamurthi

Address : 2G, Saraswathi Nilayam,  
Jawahar Nagar, Trivandrum – 695041, India

Email : indu.ramamurthi@gmail.com

Education : B.E., Electrical Engineering, National Institute of Technology, Nagpur  
M.S., Electrical Engineering Texas A&M University