A UNIFIED PROCEDURE FOR CONTINUOUS-TIME AND DISCRETE-TIME
ROOT LOCUS AND BODE DESIGN

A Thesis by

Tooran Emami

Bachelor of Science, University of Esfahan, 1994

Submitted to the Department Electrical Engineering
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

May 2006
A UNIFIED PROCEDURE FOR CONTINUOUS-TIME AND DISCRETE-TIME ROOT LOCUS AND BODE DESIGN

I have examined the final copy of this Thesis for form and content and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master with a major in Electrical Engineering.

_________________________________
John M. Watkins, Committee Chair

We have read this Thesis and recommend its acceptance:

_________________________________
Edwin M. Sawan, Committee Member

_________________________________
Brain Driessen, Committee Member
DEDICATION

To my parents and my family:
Jeff, Reza, Mona
ACKNOWLEDGMENTS

I would like to extend gratitude and appreciation to my advisor Dr. Watkins from the Department of Electrical and Computer Engineering at Wichita State University for his many years of thoughtful patient guidance, wisdom, and unceasing support. Working with Dr. Watkins was a learning and joyful experience.

Dr. Sawan was one of my committee members for this thesis. As the Chair of the department of Electrical and Computer Engineering at Wichita State University, his schedule was very full, and I greatly thank him for agreeing to be a committee member and helping me with other important academic items during my graduate academic career.

Dr. Driessen from the Department of Mechanical Engineering at Wichita State University served as a member of my thesis committee. I would like to convey my gratitude to him.

I would like to give special thanks to Kristie Bixby (editorial assistant) for her expertise and professional input in editing my written thesis report.

My parents and my brother Morteza were an integral part of my continuing with graduate work in electrical engineering. I found that their words were always encouraging. I truly thank them for their love and continued support.

My children are my life and their being is vital. My husband Jeff did a superb job in caring for our children and supporting me during graduate school. I truly thank them.

Finally, I would like to extend my gratitude to all the graduate students at Wichita State University for their friendship and help.

Thank you one and all.
ABSTRACT

As an alternative to the numerous distinct controller design algorithms in continuous-time and discrete-time classical control textbooks, a simple, unified design approach is presented for all standard continuous-time and discrete-time, classical compensators independent of the form of the system information. This approach is based on a simple root locus design procedure for a proportional-derivative (PD) compensator. From this procedure, design procedures for unified notation lead, proportional-integral (PI), proportional-integral –derivative (PID), and PI-lead compensator are developed. The delta operator, which serves as a link between the continuous-time and discrete-time procedures, offers improved numerical properties to the traditional discrete–time shift operator. With this proposed approach, designers can concentrate on the larger control system design issues, such as compensator selection and closed-loop performance, rather than the intricacies of a particular design procedure.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.2 Overview</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>TRANSFORM TECHNIQUES</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.1 Introduction</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.2 Continuous-Time Transforms</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.2.1 Laplace Transforms</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.2.2 State Space Equations and Transfer Function</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>2.3 Discrete-time Transforms</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>2.3.1 Z-Transforms</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>2.3.2 State Space Equations and Transfer Function</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>2.3.3 Delta Transform</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2.3.3 State Space Equations and Transfer Function</td>
<td>11</td>
</tr>
<tr>
<td>3.</td>
<td>FEEDBACK CONTROL SYSTEM</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>3.1 Introduction</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>3.2 System Type and Steady-State Error</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>3.3 Compensator Design</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>3.3.1 Time-Domain (root locus) Specifications</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>3.3.2 Frequency-Domain (Bode) Specifications</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>3.3.4 Compensator Design Fundamentals</td>
<td>19</td>
</tr>
<tr>
<td>4.</td>
<td>TYPES OF COMPENSATOR</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>4.1 Introduction</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>4.2 Basic Computation</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>4.3 PD Compensator</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>4.4 Lead Compensator</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>4.5 PI Compensator</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>4.6 PID Compensator</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>4.7 PI-Lead Compensator</td>
<td>29</td>
</tr>
</tbody>
</table>
5. EXAMPLES

5.1 Proportional-Derivative (PD) Examples

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.1 Continuous Time Root Locus</td>
<td>31</td>
</tr>
<tr>
<td>5.1.2 Delta Form Root Locus</td>
<td>33</td>
</tr>
<tr>
<td>5.1.3 Continuous Time Frequency Response</td>
<td>35</td>
</tr>
<tr>
<td>5.1.4 Delta Form Frequency Response</td>
<td>38</td>
</tr>
</tbody>
</table>

5.2 Lead Examples

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.1 Continuous Time Root Locus</td>
<td>40</td>
</tr>
<tr>
<td>5.2.2 Delta Operator Root Locus</td>
<td>42</td>
</tr>
<tr>
<td>5.2.3 Continuous Time Frequency Response</td>
<td>44</td>
</tr>
<tr>
<td>5.2.4 Delta Form Frequency Response</td>
<td>46</td>
</tr>
</tbody>
</table>

5.3 PI Examples

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.1 Continuous Time Root Locus</td>
<td>48</td>
</tr>
<tr>
<td>5.3.2 Delta Operator Root Locus</td>
<td>50</td>
</tr>
<tr>
<td>5.3.3 Continuous Time Frequency Response</td>
<td>52</td>
</tr>
<tr>
<td>5.3.4 Delta Form Frequency Response</td>
<td>54</td>
</tr>
</tbody>
</table>

5.4 PID Examples with Co-located Zeros

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.1 Continuous Time Root Locus</td>
<td>56</td>
</tr>
<tr>
<td>5.4.2 Delta Operator Root Locus</td>
<td>58</td>
</tr>
<tr>
<td>5.4.3 Continuous Time Frequency Response</td>
<td>60</td>
</tr>
<tr>
<td>5.4.4 Delta Form Frequency Response</td>
<td>62</td>
</tr>
</tbody>
</table>

5.5 PID Examples with Non-located Zeros

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5.1 Continuous Time Root Locus</td>
<td>64</td>
</tr>
<tr>
<td>5.5.2 Delta Operator Root Locus</td>
<td>66</td>
</tr>
<tr>
<td>5.5.3 Continuous Time Frequency Response</td>
<td>68</td>
</tr>
<tr>
<td>5.5.4 Delta Form Frequency Response</td>
<td>70</td>
</tr>
</tbody>
</table>

5.6 PI-Lead Examples with Co-located Zeros

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6.1 Continuous Time Root Locus</td>
<td>72</td>
</tr>
<tr>
<td>5.6.2 Delta Operator Root Locus</td>
<td>74</td>
</tr>
<tr>
<td>5.6.3 Continuous Time Frequency Response</td>
<td>76</td>
</tr>
<tr>
<td>5.6.4 Delta Form Frequency Response</td>
<td>78</td>
</tr>
</tbody>
</table>

5.7 PI-Lead Examples with Non-located Zeros

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.7.1 Continuous Time Root Locus</td>
<td>80</td>
</tr>
<tr>
<td>5.7.2 Delta Operator Root Locus</td>
<td>82</td>
</tr>
<tr>
<td>5.7.3 Continuous Time Frequency Response</td>
<td>84</td>
</tr>
<tr>
<td>5.7.4 Delta Form Frequency Response</td>
<td>86</td>
</tr>
</tbody>
</table>

6. CONCLUSION

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>88</td>
</tr>
</tbody>
</table>

7. REFERENCES

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF REFERENCES</td>
<td>90</td>
</tr>
</tbody>
</table>
8. APPENDIXES........................................................................................................... 92

Appendix A......................................................................................................................... 93
Matlab M-files for PD Design.......................................................................................... 93
Appendix B.......................................................................................................................... 98
Matlab M-files for Lead design......................................................................................... 98
Appendix C.......................................................................................................................... 103
Matlab M-files for PI Design.......................................................................................... 103
Appendix D.......................................................................................................................... 108
Matlab M-files for PID Design, with Co-Located Zeros ............................................. 108
Appendix E.......................................................................................................................... 113
Matlab M-files for PID design, with Non-located Zeros............................................. 113
Appendix F.......................................................................................................................... 118
Matlab M-files for PI-lead Design, with Co-Located Zeros ......................................... 118
Appendix G.......................................................................................................................... 123
Matlab M-files for PI-lead design, with Non-located Zeros ......................................... 123
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Relationship between Steady-State Error and Type Number</td>
<td>16</td>
</tr>
<tr>
<td>2. Unified Design Point $\gamma_o$</td>
<td>19</td>
</tr>
<tr>
<td>3. Feasibility of Compensator Designs</td>
<td>24</td>
</tr>
<tr>
<td>4. Final PD Compensator in Continuous Time</td>
<td>33</td>
</tr>
<tr>
<td>5. Final PD Results in Delta Form</td>
<td>35</td>
</tr>
<tr>
<td>6. Design Specifications</td>
<td>3</td>
</tr>
<tr>
<td>7. Final Continuous PD Compensator from Frequency Domain Design</td>
<td>37</td>
</tr>
<tr>
<td>8. $\delta$-Operator PD Compensator from Frequency Domain Design</td>
<td>49</td>
</tr>
<tr>
<td>9. Final Lead Compensator in Continuous Time</td>
<td>41</td>
</tr>
<tr>
<td>10. Final Lead Compensator in Delta Form</td>
<td>43</td>
</tr>
<tr>
<td>11. Continuous Lead compensator from Frequency Domain Design</td>
<td>45</td>
</tr>
<tr>
<td>12. $\delta$-Operator Lead Compensator from Frequency Domain Design</td>
<td>47</td>
</tr>
<tr>
<td>13. Final PI Compensator in Continuous Time</td>
<td>49</td>
</tr>
<tr>
<td>14. Final PI Compensator in Delta Form</td>
<td>51</td>
</tr>
<tr>
<td>15. Continuous PI compensator from Frequency Domain Design</td>
<td>53</td>
</tr>
<tr>
<td>16. $\delta$-Operator PI Compensator from Frequency Domain Design</td>
<td>55</td>
</tr>
<tr>
<td>17. Final PID Compensator in Continuous Time</td>
<td>57</td>
</tr>
<tr>
<td>18. Final PID Compensator in the Delta Form</td>
<td>59</td>
</tr>
<tr>
<td>19. Continuous PID compensator from Frequency Domain Design</td>
<td>61</td>
</tr>
<tr>
<td>20. $\delta$-Operator PID Compensator from Frequency Domain Design</td>
<td>63</td>
</tr>
<tr>
<td>21. Final PID Compensator in Continuous Time</td>
<td>65</td>
</tr>
<tr>
<td>No.</td>
<td>Title</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>22.</td>
<td>Final PID Compensator in the Delta Form</td>
</tr>
<tr>
<td>23.</td>
<td>Continuous PID compensator from Frequency Domain Design</td>
</tr>
<tr>
<td>24.</td>
<td>$\delta$-Operator PID Compensator from Frequency Domain Design</td>
</tr>
<tr>
<td>25.</td>
<td>Final PI-Lead Compensator in Continuous Time</td>
</tr>
<tr>
<td>26.</td>
<td>Final PI-Lead Compensator in the Delta Form</td>
</tr>
<tr>
<td>27.</td>
<td>Continuous PI-Lead compensator from Frequency Domain Design</td>
</tr>
<tr>
<td>28.</td>
<td>$\delta$-Operator PI-Lead Compensator from Frequency Domain Design</td>
</tr>
<tr>
<td>29.</td>
<td>Final PI-Lead Compensator in Continuous Time</td>
</tr>
<tr>
<td>30.</td>
<td>Final PI-Lead Compensator in the Delta Form</td>
</tr>
<tr>
<td>31.</td>
<td>Continuous PI-Lead compensator from Frequency Domain Design</td>
</tr>
<tr>
<td>32.</td>
<td>$\delta$-Operator PI-Lead Compensator from Frequency Domain Design</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Input-output relationships</td>
<td>6</td>
</tr>
<tr>
<td>2.</td>
<td>Standard closed-loop block diagram</td>
<td>13</td>
</tr>
<tr>
<td>3.</td>
<td>Closed-loop block diagram</td>
<td>16</td>
</tr>
<tr>
<td>4.</td>
<td>Relationship between the angle of a pole or a zero and design point</td>
<td>22</td>
</tr>
<tr>
<td>5.</td>
<td>Relationship between $\theta_c$ and $\theta_{c, \text{max}}$</td>
<td>23</td>
</tr>
<tr>
<td>6.</td>
<td>Stability region of delta Form</td>
<td>25</td>
</tr>
<tr>
<td>7.</td>
<td>Relationship between poles and zero in $\gamma$ domain</td>
<td>26</td>
</tr>
<tr>
<td>8.</td>
<td>The specification region for the closed-loop system</td>
<td>32</td>
</tr>
<tr>
<td>9.</td>
<td>Step-response with PD compensator designed using continuous time root locus</td>
<td>32</td>
</tr>
<tr>
<td>10.</td>
<td>Step response and root locus plot with PD compensator designed using $\delta$-operator root locus</td>
<td>34</td>
</tr>
<tr>
<td>11.</td>
<td>Step-response and Bode plot with PD compensator designed using continuous time Bode plot</td>
<td>37</td>
</tr>
<tr>
<td>12.</td>
<td>Step-response and Bode plot with PD compensator designed using $\delta$-operator Bode plot</td>
<td>39</td>
</tr>
<tr>
<td>13.</td>
<td>Step response and root locus plot with lead compensator designed using continuous time root locus</td>
<td>41</td>
</tr>
<tr>
<td>14.</td>
<td>Step response and root locus plot with lead compensator designed using $\delta$-operator root locus</td>
<td>43</td>
</tr>
<tr>
<td>15.</td>
<td>Step-response and Bode plot with lead compensator designed using continuous time Bode plot</td>
<td>45</td>
</tr>
<tr>
<td>16.</td>
<td>Step-response and Bode plot with lead compensator designed using $\delta$-operator Bode plot</td>
<td>47</td>
</tr>
</tbody>
</table>
17. Step response and root locus plot with PI compensator designed using continuous time root locus……………………………………………………….. 49
18. Step response and root locus plot with PI compensator designed using δ-operator root locus …………………………………………………………………….. 51
19. Step-response and Bode plot with PI compensator designed using continuous time Bode plot……………………………………………………….. 53
20. Step-response and Bode plot with PI compensator designed using δ-operator Bode plot …………………………………………………………………….. 55
21. Step response and root locus plot with PID compensator designed using continuous time root locus……………………………………………………….. 57
22. Step response and root locus plot with PID compensator designed using δ-operator root locus …………………………………………………………………….. 59
23. Step-response and Bode plot with PID compensator designed using continuous time Bode plot………………………………………………………………….. 61
24. Step-response and Bode plot with PID compensator designed using δ-operator Bode plot …………………………………………………………………….. 63
25. Step response and root locus plot with PID compensator designed using continuous time root locus………………………………………………………………….. 65
26. Step response and root locus plot with PID compensator designed using δ-operator root locus …………………………………………………………………….. 67
27. Step-response and Bode plot with PID compensator designed using continuous time Bode plot………………………………………………………………….. 69
28. Step-response and Bode plot with PID compensator designed using δ-operator Bode plot …………………………………………………………………….. 71
29. Step response and root locus plot with PI-lead compensator designed using continuous time root locus………………………………………………………………….. 73
30. Step response and root locus plot with PI-lead compensator designed using δ-operator root locus …………………………………………………………………….. 75
31. Step-response and Bode plot with PI-lead compensator designed using continuous time Bode plot………………………………………………………………….. 77
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>Step-response and Bode plot with PI-lead compensator designed using ( \delta )-operator Bode plot.</td>
<td>79</td>
</tr>
<tr>
<td>33</td>
<td>Step response and root locus plot with PI-lead compensator designed using continuous time root locus</td>
<td>81</td>
</tr>
<tr>
<td>34</td>
<td>Step response and root locus plot with PI-lead compensator designed using ( \delta )-operator root locus</td>
<td>83</td>
</tr>
<tr>
<td>35</td>
<td>Step-response and Bode plot with PI-lead compensator designed using continuous time Bode plot</td>
<td>85</td>
</tr>
<tr>
<td>36</td>
<td>Step-response and Bode plot with PI-lead compensator designed using ( \delta )-operator Bode plot.</td>
<td>87</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

1.1 Motivation

In control education today, there seems to be a gap between the theory taught in the typical undergraduate classroom and what students are able to apply to practical systems. One obvious reason for this is the lack of undergraduate control system laboratories. The control systems community has recognized this need [1, 2]. A less obvious reason for this gap is the “cookbook” approach to compensator design found in typical classical control textbooks [5, 6, 7]. For example, a quick comparison reveals significant differences in the procedures for root locus lead design and root locus PI design. Even more importantly, there are significant differences in the procedure for lead compensator design using root locus techniques and Bode techniques. Furthermore, for even fairly simple systems, these design procedures may yield poor results [8]. Students can become even more confused as new methods are introduced for continuous-time and discrete-time control system design [9, 10, 11, 12]. Consequently, students concentrate on the different “recipes,” which may or may not yield satisfactory results and, thus, tend to miss the larger picture.

Discrete-time time methods are classified as indirect or direct. In the indirect methods, a continuous-time compensator is designed from a continuous-time model of the system and discretized for a discrete-time implementation. Indirect methods require only limited knowledge of discrete-time control. However, as Ogata [9] points out, “discretizing the continuous-time control system creates new phenomenon not present in the original continuous-time control system.” To overcome this issue, direct methods
employ root locus or Bode techniques to compute the discrete-time compensator from a discrete-time model of the system. In discrete-time control textbooks [9, 10, 11, 12], direct design of classical, discrete-time compensators receives far less attention than analogous design methods in textbooks on continuous-time control design. Furthermore, root locus techniques are not developed for each compensator, and Bode techniques rely on a transformation of the pulse transfer function.

In this research, design methods were developed to permit students to apply a simple, unified design approach for all compensators, independent of the type of the system information. That is, in root locus design, computational procedures were based on the open-loop transfer function, whereas in Bode design, computational procedures were based on the magnitude and phase of the open-loop frequency response. With this proposed approach, students would be able to concentrate on larger control system design issues, such as compensator selection and closed-loop performance, rather than the intricacies of a particular design procedure. Variations of the proposed design methods have been applied successfully in classical continuous-time [13, 14] and discrete-time [15] control classes at the United State Naval Academy and Wichita State University. In this research, compensator design results for the continuous-time and discrete time systems are unified by using the $\delta-$operator [19].

Typically, discrete time system analysis is performed using the $q$ forward shift operator and the Z-transform. Unfortunately, the resulting discrete domains are only loosely connected with the continuous domain that spawns them. Specifically, the underlying continuous domain descriptions cannot be obtained by setting the sampling period to zero in the discrete domain approximations to them. Furthermore, it is widely
known that serious numerical issues arise when using shift operator formulations of algorithms at high sampling rates relative to the natural frequencies of the system being estimated [12, 16, 17, 19].

1.2 Overview

The objective of this research was to provide a common framework for continuous-time and discrete-time controller design using the delta operator. There is a close connection between continuous time results and formulations of the delta operator in discrete time because the delta operator acts like an approximate derivative and approaches a true derivative as the sampling time approaches zero. Furthermore, there is a simple linear transformation between the delta operator parameters and the shift operator parameters.

In addition to this introduction this document contains five additional chapters. Chapter 2 presents the transform techniques and discusses why they are needed. The Laplace transformation and the relationship between the state space equation and the transfer function in continuous time systems are discussed in Section 2.2. The Z-transformation, the delta operator, and the corresponding space state equation are introduced in Section 2.3.

Chapter 3 lays the groundwork for feedback control system design. The need for feedback is discussed in Section 3.1. The steady-state error and the effect of the type of system on the error are discussed in Section 3.2. Root locus design specifications, Bode design specifications, and compensator design fundamentals are introduced in Section 3.3.
Chapter 4 presents the design methods for five compensators. Procedures for lead, proportional-integral (PI), lag, proportional-integral-derivative (PID), and PI-lead compensators are developed based on a simple root locus design proportional-derivative (PD) procedure. While the procedures presented in this document are analogous to those presented in [13, 14], the presentation is self-contained and only assumes knowledge of standard classical control concepts.

Chapter 5 incorporates examples, which are the numerical simulation results using MATLAB for seven different methods of compensator design. Designs were created for the unified procedures for continuous-time and discrete-time using root-locus and Bode design. The results illustrate the similarity between the continuous-time and the $\delta$-domain discrete time compensators. Chapter 6 concludes the thesis by summarizing the advantages of this method for continuous and discrete–time compensator design.
CHAPTER 2
TRANSFORM TECHNIQUES

2.1 Introduction

Most classical control system design methods involve the transformation of the
time domain differential or difference equation into a frequency domain linear algebraic
equation [11]. In digital control system design, this is usually done using the $q$ forward
shift operator and the associated discrete frequency variable $z$. The problem is that the
resulting discrete-time domain is significantly different from the original continuous-time
domain, because the underlying continuous domain description cannot be obtained by
setting the sampling period to zero in the discrete domain approximation. The delta
operator, which serves as a link between continuous-time and discrete-time procedures,
offers improved numerical properties to the traditional discrete-time shift operator [12,
16, 17, 19].

Linear dynamic systems can be described by differential or difference equations. Control systems use transform techniques to study the behavior of linear systems. The
first step is to transform the time domain differential or difference equations into
frequency domain linear algebraic equations. This transformation into linear algebraic
equations yields many important concepts, such as poles, zeros, and transfer functions.

The transfer function defines the input-output relationship of a system for zero initial
condition by

$$G(\alpha) = \frac{Y(\alpha)}{U(\alpha)} \quad (2.1)$$
where $G(\alpha)$ is the transfer function, $Y(\alpha)$ is the system output, $R(\alpha)$ is the system input and $\alpha$ is frequency variable and depends on the domain of the system. The system transfer function is shown on Figure 1.

![Input-output relationships](image)

Figure 1. Input-output relationships.

2.2 Continuous-Time Transforms

2.2.1 Laplace Transforms

The analysis of continuous linear dynamic systems allows the use of the Laplace transformation. Consider a function $f(t)$, which is zero for $t < 0$. The Laplace transform, as the linear algebraic transformation of this function, is given by $F(s)$ as

$$F(s) = \int_{0^{-}}^{\infty} f(t) e^{-st} dt = L\{f(t)\}$$

(2.2)

where $0^{-}$ denotes the limit of zero approached from the negative direction. If there exist constants $m$ and $\sigma$, such as $|f(t)| \leq m e^{\sigma t}$, then the integral in equation (2.2) transforms to all the complex numbers $s$ that satisfy $Re\{s\} > \sigma$ [12].

One of the most important properties of Laplace transformation is the transform of the derivative of a signal as

$$L\left\{\frac{d}{dt} f(t)\right\} = sF(s) - f(0)$$

(2.3)
where $f(0)$ is the initial condition [11].

### 2.2.2 State Space Equations and Transfer Function

A general state space model of a linear continuous time model is expressed by

$$\frac{d}{dt} x(t) = A x(t) + B u(t)$$

$$y(t) = C x(t)$$

where $x(t)$ is the state of the system and contains $n$ elements for an $n$th-order system, $u(t)$ is the $m\times1$ input vector to the system, $y(t)$ is the $p\times1$ output vector, $A$ is an $n\times n$ system matrix, $B$ is an $n\times m$ input matrix, and $C$ is $p\times n$ output matrix [11].

The Laplace transform of the state space equation with the initial condition of $x(0) = x_0$ is given by [12]

$$X(s) = (sI - A)^{-1} (B U(s) + x_0)$$

and the output is given by

$$Y(s) = C(sI - A)^{-1} B U(s) + C(sI - A)^{-1} x_0.$$  \hfill (2.6)

It can be seen the linear differential equation (2.4) has been translated to linear algebraic equations (2.5) and (2.6). The term $C(sI - A)^{-1} B$ is often called the transfer function, and for zero initial conditions, the continuous-time transfer function is defined by [12]

$$G(s) = C(sI - A)^{-1} B.$$
2.3 Discrete-Time Transforms

2.3.1 Z-Transforms

In a discrete time system, the Z-transform of a sequence is equivalent to the Laplace transform in continuous time system. For a given sequence \( \{ f_k = f(k\Delta) \} \), where \( \Delta \) is the sampling period. The Z-transform is defined by \( F(z) \) as

\[
F(z) = \mathcal{Z}\{ f(k\Delta) \} = \sum_{k=0}^{\infty} f(k\Delta)z^{-k\Delta}.
\]  

(2.8)

In the Laplace transform, if \( |f(t)| \leq me^{\sigma t} \) for all \( t \), then the Z-transform converges absolutely for all complexes \( z \) that belong to the region \( |z| > e^\sigma \) [12].

2.3.2 State Space Equations and Transfer Function

One of the important properties of the Z-transform is the transform of the shift operator of a signal and is defined as

\[
Z\{ q f(k\Delta) \} = z[Z\{ f(k\Delta) \} - f_0]
\]

(2.9)

where \( q \) is the shift operator \( \{ q f(k\Delta) \} = \{ f(k\Delta + 1) \} \), \( k \) is an integer number, and \( f_0 \) is the initial condition. Assuming a sampling period of \( \Delta \) and a zero-order-hold, the discrete equivalent of (2.4) is given by

\[
qx_k = Ax_k + Bu_k
\]
\[
y_k = Cx_k
\]

(2.10)

where

\[
x_k \triangleq x(k\Delta)
\]

(2.11)
\[ u_k \Delta u (k, \Lambda) \]  \hspace{1cm} (2.12)\\
\[ y_k \Delta y (k, \Lambda) \]  \hspace{1cm} (2.13)\\
\[ A_k \Delta e^{A \lambda} \]  \hspace{1cm} (2.14)\\
\[ B_z = \int_0^d e^{A \lambda - t} B \, dt \]  \hspace{1cm} (2.15)\\

and

\[ C_z = C. \]  \hspace{1cm} (2.16)\\

The Z-transform of the discrete state space equation with the initial condition \( x(0) = x_0 \) is given by

\[ X(z) = (zI - A_z)^{-1} (B_z U(z) + z x_0) \]  \hspace{1cm} (2.17)\\

and

\[ Y(z) = C_z (zI - A_z)^{-1} (B_z U(z) + z x_0) \]  \hspace{1cm} (2.18)\\

where \( X(z) \), \( U(z) \) and \( Y(z) \) are the Z-transform of \( x_k \), \( u_k \), and \( y_k \) from equation (2.10). The term, \( G'(z) = C_z (zI - A_z)^{-1} B_z \), is called the Z-transfer function. The Z-transfer function can be evaluated directly from the continuous time transfer function by sampling the output with a zero order hold input as [12, 16].
\[ G'(z) = \frac{z^{-1}}{z} Z[L^{-1}\left\{\frac{1}{s} G(s)\right\}] . \] (2.19)

The shift operator \( q \) is often used to describe discrete time systems. Unfortunately, the forward shift operator is not at all like the continuous time operator \( d / dt \). Because of the discontinuous time in discrete time, the sampling period cannot be zero. Therefore, the results in discrete time are different from the continuous time domain [12, 16, 19].

### 2.3.3 Delta Transform

In 1990, Middleton and Goodwin [12] developed a new operation for the discrete time system. They demonstrated that if the shift operator is replaced by a difference operator, it behaves more like a derivative. They defined the new operator as the delta operator

\[ \delta \Delta = \frac{q - 1}{\Delta} \] (2.20)

and equivalently

\[ \delta x_k = \frac{x_{k+1} - x_k}{\Delta} = \frac{x(k \Delta + \Delta) - x(k \Delta)}{\Delta} . \] (2.21)

The relationship between \( \delta \) and \( q \) \( \{x_{k+1} = qx_k\} \) is a simple linear function, and \( \delta \) has the same flexibility in modeling discrete time system as \( q \) does. On the other hand, the \( \delta \) operator leads to models that are more like their continuous-time system counter parts [12, 16, 19].
2.3.4 State Space Equations and Transfer Function

Using the delta operator [12, 16, 19], state space equations are given by

\[ \delta x = A_{\delta} x + B_{\delta} u \]
\[ y = C_{\delta} x \]  \hspace{1cm} (2.22)

where

\[ A_{\delta} = \frac{A_z - I}{\Delta} \]  \hspace{1cm} (2.23)

\[ B_{\delta} = \frac{B_z}{\Delta} \]  \hspace{1cm} (2.24)

and

\[ C_{\delta} = C_z \]  \hspace{1cm} (2.25)

where \( A_z, B_z, \) and \( C_z\) are matrices from the shift operator state space equation in equation (2.10).

From Section 2.2.1 it can be seen that there is a transformation between the forward shift operator \( q \) and the Z-transform variable \( z \). By analogy, another variable \( \gamma \) in the delta operator discrete time system can be defined as \( \gamma = \frac{z - 1}{\Delta} \).

In the delta operator discrete time systems, the \( \gamma \)-transform of a sequence is equivalent to the Laplace transform of a continuous time system. The \( \delta \)-operator state-space matrices can be evaluated directly from the continuous-time state matrices as [12, 19]:

\[ A_{\delta} = \Omega A \]  \hspace{1cm} (2.26)

and,
\[ B_\delta = \Omega B \]  \hspace{1cm} (2.27)

where

\[ \Omega = \frac{1}{\Delta} \int_0^\Delta e^{\Delta \tau} d \tau = I + A \Delta + A^2 \Delta^2 + \cdots. \]  \hspace{1cm} (2.28)

As \( \Delta \to 0 \), \( \Omega \to I \) and the close connection between the delta domain and the underlying continuous time domain is apparent. This document uses this relationship between the delta domain and the continuous time domain to develop a unified procedure for compensator design.

The transformation of equation (2.22) with initial condition \( x(0) = x_0 \) to the \( \gamma \) domain is given by

\[ X(\gamma) = (\gamma I - A_\delta)^{-1} \{ (1 + \Delta \gamma) x_0 + B_\delta U(\gamma) \} \]  \hspace{1cm} (2.29)

\[ Y(\gamma) = C_\delta X(\gamma) \]  \hspace{1cm} (2.30)

where \( X(\gamma), Y(\gamma), \) and \( U(\gamma) \) are \( \gamma \)-transform of \( x, y, \) and \( u \) from equation (2.22). For zero initial conditions, the \( \gamma \)-transfer function is given by \( G^\gamma(\gamma) = C_\delta (\gamma I - A_\delta)^{-1} B_\delta \). The \( \gamma \)-transfer function can be evaluated directly from the continuous time transfer function by sampling the output with a zero-order-hold input as

\[ G^\gamma(\gamma) = \frac{\gamma}{1 + \gamma \Delta} T[L^{-1}\{\frac{1}{s} G(s)\}] \]  \hspace{1cm} (2.31)

where \( T \) is the \( \gamma \)-transform of the continuous time system[12, 16].
3.1 Introduction

The purpose of feedback control is to ensure that the response of the plant be controlled and satisfies required specifications. These specifications are given in terms of the closed-loop system. The feedback control system generates an input to the plant system based on a comparison of the desired and actual closed-loop response. In the automated feedback control system design process, the task for the control system is to use information about the plant system to design a feedback control system to achieve the specifications [18]. A standard continuous-time closed-loop block diagram is shown in Figure 2, where $R(s)$ is the desired input, $Y(s)$ is the actual output, $E(s)$ is the controller input, and $U(s)$ is the controller output.

![Standard closed-loop block diagram](image)

Figure 2. Standard closed-loop block diagram.

The design of a feedback control system requires a relationship between the open-loop information and the closed-loop performance. The behavior of a closed-loop system depends on the design of the controller, and in this document the design of compensator.
Compensator design is often based on the assumption that the closed-loop transfer function can be approximated by a standard 2\textsuperscript{nd} order system as

\[
T(s) = \frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}.
\] (3.1)

The desired 2\textsuperscript{nd} order closed-loop transfer function is determined from the specifications by the desired values for the damping ratio \(\zeta\) and the natural frequency \(\omega_n\). The relationships between percent overshoot (P.O.) and \(\zeta\), \(\text{P.O.} = 100e^{-\pi\sqrt{1-\zeta^2}}\), and settling time \(T_s\) and \(\zeta\) and \(\omega_n\), \(T_s \approx \frac{4}{\zeta\omega_n}\) are often used, [18]. If the closed-loop system is not a 2\textsuperscript{nd} order system, then the closed-loop response can be evaluated using the following results:

- Any extra closed-loop zeros in the left-hand plane will cause the closed-loop step response to have a higher percent overshoot and a smaller time to peak than the response expected from the desired values of \(\zeta\) and \(\omega_n\). This effect increases as the zeros move toward the imaginary axis [18].
- Additional closed-loop poles in the left-hand plane will cause the closed-loop step response to have a lower percent overshoot and a longer time to peak than the response expected from the desired values of \(\zeta\) and \(\omega_n\). This effect increases as the poles move toward the imaginary axis [18].

The design of a feedback control system is often done in three steps. In the first step, the designer determines if a compensator for the system is required, by setting \(G_c(s) = 1\) and determining if the control gain \(K\) can be chosen to achieve the
performance specifications. This kind of feedback control is called “uncompensated”. If the specifications cannot be achieved using proportional control, then the second step is to choose the type of compensator required based on the closed-loop transient and steady-state error specifications. Finally, in the third step the parameters associated with the chosen compensator and control gain $K$ are computed to meet the performance specifications [18].

### 3.2 System Type and Steady-State Error

In the unity feedback system shown in Figure 1, the steady-state error is defined by

$$ e_{ss} = \lim_{t \to \infty} e(t) = \lim_{s \to 0} sE(s) $$

where

$$ E(s) = R(s) - Y(s). $$

The steady-state error associated with the closed-loop system uses the system type number and the error constants associated with the open-loop transfer function. The system type number $N$ is the number of integrators (open-loop pole at the origin). In the open-loop transfer function, the error constant is defined by

$$ K_n = \lim_{s \to 0} s^n K G_c(s) G_p(s) $$

where $K$ is the control gain, $G_c(s)$ is the compensator, $G_p(s)$ is the plant transfer function, and $n \geq 0$ is a integer. Typically the step, ramp, and parabolic error constant are used in compensator design. The three-error constants are defined $K_p = K_0$ (position error constant) for the step input, $K_v = K_1$ (velocity error constant) for ramp the input
and $K_a = K_3$ (acceleration error constant) for parabola input [18]. The relationships between system type and the steady-state error are given in Table 1.

### Table 1

#### RELATIONSHIPS BETWEEN STEADY-STATE ERROR AND TYPE NUMBER

<table>
<thead>
<tr>
<th>Type Number</th>
<th>Step Input</th>
<th>Ramp Input</th>
<th>Parabola Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\frac{1}{1+K_p}$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$\frac{1}{K_v}$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>$\frac{1}{K_s}$</td>
</tr>
</tbody>
</table>

### 3.3 Compensator Design

The integrated design procedure using time or frequency domain plant data requires a generalization of the angle criterion from root locus design. The standard closed-loop system is shown in Figure 3, where $K$ is the control gain, $G_c(\gamma)$ is the compensator, and $G_p(\gamma)$ represents the plant dynamics.

![Figure 3. Closed-loop block diagram.](image)
As discussed earlier, an alternative discrete time operator is defined by the delta operator

\[ \delta = \frac{q - 1}{\Delta} \]  

(3.6)

where \( \Delta \) is the sampling period. In view of the connection between \( q \) and \( z \), the transform associated with \( \delta \) is given as \( \gamma = (z - 1)/\Delta \) [12, 16]. The variable \( \gamma \) has the following relation with the continuous time domain:

\[
\gamma = \begin{cases} 
  s & \Delta = 0, \\
  e^{\Delta s} - 1 & \Delta \neq 0,
\end{cases}
\]  

(3.7)

where \( s \) is the Laplace transforming variable.

### 3.3.1 Time-Domain (Root Locus) Specifications

In root locus design, the compensator must satisfy the well-known angle and magnitude criteria:

\[
\angle G_c(\gamma_0) + \angle G_p(\gamma_0) = \pm 180^\circ \\
K \left| G_c(\gamma_0)G_p(\gamma_0) \right| = 1.
\]

(3.8)

The unified notation design point \( \gamma_0 = \begin{cases} 
  s_0 & \Delta = 0 \\
  e^{s_0 \Delta} - 1 & \Delta \neq 0
\end{cases} \), is determined from the continuous-time design point \( s_0 = -\zeta \omega_n + j\omega_n \sqrt{1 - \zeta^2} \) where \( \zeta \) is the damping ratio and \( \omega_n \) is the natural frequency.
3.3.2 Frequency-Domains (Bode) Specifications

In Bode design methods, the specifications are incorporated through the desired phase margin \((PM)\) and gain crossover frequency \((\omega_{gc})\), which result in another set of angle and magnitude constraints:

\[
\angle G_c(\gamma_0) + \angle G_p(\gamma_0) = \pm 180^\circ + PM
\]

\[
K \left| \frac{G_c(\gamma_0)G_p(\gamma_0)}{K} \right| = 1
\]

where \(\gamma_0\) is the unified notation design point given by \(\gamma_0 = \begin{cases} 
\frac{j \omega_{gc}}{\Delta} & \Delta = 0 \\
\frac{e^{j \omega_{gc} \Delta} - 1}{\Delta} & \Delta \neq 0
\end{cases} \).

Using standard 2\textsuperscript{nd} order assumptions, such as those found in [5], the \(PM\) and \(\omega_{gc}\) can also be determined from the continuous-time design point as

\[
PM = \tan^{-1} \left( \frac{\frac{2 \zeta}{\sqrt{-2 \zeta + \sqrt{1 + 4 \zeta^2}}}}{} \right)
\]

and

\[
\omega_{gc} = \frac{2 \zeta \omega_n}{\tan(PM)}.
\]
### 3.3.3 Compensator Design Fundamentals

Equations (3.8) and (3.9) can be combined to obtain the generalized angle and magnitude constraints:

\[ \angle G_c(\gamma_0) + \angle G_p(\gamma_0) = \phi \]
\[ K \left| G_c(\gamma_0)G_p(\gamma_0) \right| = 1 \]  

where the desired angle in the angle constraint is

\[ \phi = \begin{cases} 
\pm 180^\circ, & \text{root locus} \\
\pm 180^\circ + PM, & \text{Bode} 
\end{cases} \]  

The design point \( \gamma_0 \) is given by Table 2 for the different design procedures.

**TABLE 2**

<table>
<thead>
<tr>
<th>Unified Design Point ( \gamma_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta = 0 )</td>
</tr>
<tr>
<td>Root Locus ( s_0 )</td>
</tr>
<tr>
<td>Bode ( j\omega_{\Delta} )</td>
</tr>
</tbody>
</table>

Using the angle constraint in equations (3.12) and (3.13), the desired compensator angle \( \theta_c \) can be computed from the plant information and the design point without knowledge of the compensator type [15].

\[ \angle G_c(\gamma_0) = \phi - \angle G_p(\gamma_0) = \theta_c. \]  

\[ (3.14) \]
In root locus methods, $\theta_c$ determines a geometric relationship between the design point and the compensator poles and zeros. In Bode methods, $\theta_c$ is the phase that must be added at the gain crossover frequency [16]. Specifically, the control gain $K$ is determinate by the magnitude criterion from equation (3.12)

$$K = \frac{1}{\left| G_c(\gamma_0) G_p(\gamma_0) \right|}.$$  

(3.15)
CHAPTER 4
TYPE OF COMPENSATORS

4.1 Introduction

In this chapter, the design procedures for five compensators—lead, proportional integral (PI), proportional-integral-derivative (PID) and PI-lead (practical PID)—are developed from the design procedure for proportional-derivative (PD) compensator. The design procedures were generalized in Section 3.3.

4.2 Basic Computation

The unified design approach provides design procedures for all standards compensator based on a basic procedure. In all of the design methods, the angle of a pole or zero $\gamma = -\alpha$ must be computed from a design point specification $\gamma_0 = -\sigma_0 + j\omega_0$, and this computation is defined by

$$\theta := \angle(\gamma_0 + \alpha) = \tan^{-1}\left(\frac{\omega_0}{\alpha - \sigma_0}\right). \quad (4.1)$$

The basic computation for a unified design approach is the solution to calculate $\theta$, where $\theta$ is the angle of compensator pole or zero. When $\theta$ is computed, the unknown pole or zero compensator can be obtained by

$$\alpha = \sigma_0 + \frac{\omega_0}{\tan(\theta)}. \quad (4.2)$$

The relationship between a pole or zero, and design point is shown in Figure 4 [18].
Figure 4. Relationship between the angle of a pole or a zero and design point.

4.1 PD Compensator

As in the continuous-time case, the design procedures for all compensators are based on the PD design procedure [13, 14]. The PD compensator in the unified notation has a transfer function

\[
G_c(\gamma) = \frac{\gamma + \alpha}{\Delta \gamma + 1}. \tag{4.3}
\]

The angle of the PD compensator at the design point \( \gamma_0 \) is

\[
\theta_c = \angle G_c(\gamma_0) = \angle(\gamma_0 + \alpha) - \angle(\Delta \gamma_0 + 1). \tag{4.4}
\]

Therefore

\[
\theta_z = \angle(\gamma_0 + \alpha) = \theta_c + \angle(\Delta \gamma_0 + 1) \tag{4.5}
\]

must hold and the compensator zero is given by

\[
\alpha = \sigma_0 + \frac{\omega_0}{\tan(\theta_z)}. \tag{4.6}
\]
where $\gamma_0 = -\sigma_0 + j\omega_0$ is the design point in the unified notation. For this compensator, and each compensator to follow, the gain $K$ is computed using the magnitude constraint in equation (3.15).

The improvement that the PD compensator can achieve is limited. In general, the compensator zero should be minimum phase because a non-minimum phase compensator can lead to poor performance and/or instability in the closed-loop system, as shown in Figure 5.

![Diagram](image)

**Figure 5.** Relationship between $\theta_z$ and $\theta_{c, \text{max}}$.

Under this assumption, the maximum value for $\theta_z$ is $\angle (\gamma_0)$ and achieves by the delta operator derivative compensator $G_c(\gamma) = \frac{\gamma}{\Delta \gamma + 1}$. The PD compensator reduces to a proportional controller if $\Delta \alpha = 1$, and therefore, the minimum value for $\theta_z$ is $\angle (\Delta \gamma_0 + 1)$ . It follows that the design point can be achieved or, equivalently, that the PD compensator design problem is feasible if and only if $\angle (\Delta \gamma_0 + 1) \leq \theta_z \leq \angle (\gamma_0)$ or, equivalently, $0 \leq \theta_z \leq \theta_{c, \text{max}}$.
where

\[ \theta_{c,\text{max}} = \angle \gamma_0 - \angle (\Delta \gamma_0 + 1). \]  

(4.7)

Feasibility relationships are shown in Table 3 for other compensators—lead, PI, PID, and PI-lead—using relationships developed in the sequel [15, 18] between these compensators and the PD compensator.

**TABLE 3**

**FEASIBILITY OF COMPENSATOR DESIGNS**

<table>
<thead>
<tr>
<th>Compensator</th>
<th>Feasibility Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD, lead</td>
<td>(0 \leq \theta_c \leq \theta_{c,\text{max}})</td>
</tr>
<tr>
<td>PI, lag</td>
<td>(-\theta_{c,\text{max}} \leq \theta_c \leq 0)</td>
</tr>
<tr>
<td>PID, PI-lead</td>
<td>(-\theta_{c,\text{max}} \leq \theta_c \leq \theta_{c,\text{max}})</td>
</tr>
</tbody>
</table>

To illustrate the computations introduced in this Section, a simple root locus example is provided. For the transfer function \(G_p(\gamma) = \frac{0.005 \gamma + 0.995}{\gamma^2 + 0.995\gamma}\), suppose a design point \(\gamma_0 = -0.9963 + j0.4950\) is given. The desired compensator angle is computed from equation (3.14) and \(\theta_c = 62.85^\circ\). For the given data, the design for the basic compensator is feasible because \(0 \leq \theta_c \leq \theta_{c,\text{max}} = 153.294^\circ\), where \(\theta_{c,\text{max}}\) is computed from equation (4.7). From equation (4.6), the compensator zero is \(\alpha = 1.2391\), and from the magnitude criteria in equation (3.15), the control gain \(K\) is 0.9987.
4.3 Lead Compensator

The lead compensator in the unified notation has a transfer function

\[
G(\gamma) = \frac{\gamma + \alpha}{\gamma + \beta}
\]

(4.8)

where \(\alpha < \beta\). As in the continuous-time case [13, 14], the PD compensator design provides the limits on the lead compensator as discussed below. The angle of lead compensator at the design point \(\gamma_0\) is \(\angle G_c(\gamma_0) = \angle(\gamma_0 + \alpha) - \angle(\gamma_0 + \beta) = \theta_z - \theta_p\), and the lead pole and zero must be selected to satisfy the angle constraint equation (3.14), or equivalently,

\[
\theta_z = \theta_z - \theta_p.
\]

(4.9)

In general, it is not desirable to place the compensator pole to the left of \(\frac{-1}{\Delta}\) (even in the stability region of the delta operator, as show in Figure 6), because this pole location can lead to an oscillatory control signal.

![Figure 6. Stability region of delta operation [12.](image)](image)


As a result, \( \frac{-1}{\Delta} \leq \beta \leq 0 \), and it follows that \( \theta_p = \theta_z - \theta_c \), as illustrated in Figure 7. Note that this is analogous to the constraint obtained in the continuous-time case [13, 14]. It follows from the relationship between the lead and PD compensators that the lead compensator is feasible if and only if the PD compensator is feasible.

![Figure 7. Relationship between poles and zeros in \( \gamma \) domain.](image)

The lead compensator design has three unknowns and only two constraints. As in the continuous-time case, the lead compensator zero is chosen using the lead zero to the right side of PD compensator zero [18]

\[
0 \leq \alpha_{lead} \leq \alpha_{pd}.
\]  

(4.10)

After the lead zero is chosen the lead pole is computed from

\[
\beta = \sigma_0 + \frac{\alpha_0}{\tan(\theta_p)}
\]

(4.11)

where, \( \theta_p = \theta_z - \theta_c \).
4.4 PI Compensator

The PI compensator in the unified notation has a transfer function

\[ G_c(\gamma) = \frac{\gamma + \alpha}{\gamma} \]  \hspace{1cm} (4.12)

where \( \alpha > 0 \). If \( N \) is the type of the uncompensated system, then the PI compensator increase the type of system to \( N + 1 \). PI compensators improve the steady-state error by increasing the system type. The PI compensator integrator also causes degradation in the transient response and results in higher percent overshoot and/or larger settling time as compared to the uncompensated system [18].

The angle of the PI compensator at the design point \( \gamma_0 \) is \( \angle G_c(\gamma_0) = \angle(\gamma_0 + \alpha) - \angle(\gamma_0) \), and the PI zero is computed from \( \theta_z = \angle(\gamma_0 + \alpha) - \angle(\gamma_0) \). This design expression can be rewritten to collect the known terms on one side of the equation \( \theta_z = \angle(\gamma_0 + \alpha) = \theta_c + \angle(\gamma_0) \) where \( \theta_z \) can be computed from the plant information and the design point assuming a PI compensator is desired. The compensator zero, \( \alpha \), is computed using equation (4.6).

Note that the PI compensator is a special case of a lag compensator. The lag compensator has the same form as the lead compensator, but \( \alpha > \beta \) for the lag compensator. The design procedure for the lag compensator is identical to that of the lead compensator, except that instead of \( \alpha \) satisfying equation (4.10), it must be chosen such that \( \alpha_{lag} > \alpha_{pi} \) [15, 18].
4.6 PID Compensator

The transfer function of the PID compensator in the unified notation is given by

\[ G_c(\gamma) = \frac{(\gamma + \alpha_1)(\gamma + \alpha_2)}{\gamma(\Delta \gamma + 1)}. \]  \hspace{1cm} (4.13)

The angle of the PID compensator at the design point \( \gamma_0 \) is

\[ \angle G_c(\gamma_0) = \angle(\gamma_0 + \alpha_1) + \angle(\gamma_0 + \alpha_2) - \angle(\gamma_0) - \angle(\Delta \gamma + 1), \]

and the PID zeros are computed from

\[ \theta_{z_1} + \theta_{z_2} = \angle(\gamma_0 + \alpha_1) + \angle(\gamma_0 + \alpha_2) = \theta_c + \angle(\Delta \gamma + 1) + \angle(\gamma_0) = \gamma_{z,PID}. \]  \hspace{1cm} (4.14)

Since there are three unknown parameters and only two constraints, there is a degree of freedom in selecting the PID parameters [15]. Two design methods are considered. In the first method, one of the PID zeros is chosen, most likely to cancel a plant pole or shape the loop response. The remaining PID zero is computed from the angle constraint in equation (4.14). That is, given the PID zero \( \alpha_1 \), the remaining PID zero is computed using equations (4.5) and (4.6), with \( \theta_{z_2} = \theta_{z,PID} - \theta_{z_1} \). In the second method, the two PID zeros are assumed to be co-located and computed directly from the angle constraint. If \( \alpha_1 = \alpha_2 = \alpha \), then the angle constraint in equation (4.14) becomes \( 2\theta_z = \gamma_{z,PID} \), and the PID zeros are computed using equations (4.5) and (4.6), with \( \theta_z = \frac{\theta_{z,PID}}{2} \) [15, 18].
4.5 PI-lead Compensator

The PI-lead compensator has a transfer function in the unified notation

\[ G_c(\gamma) = \frac{(\gamma + \alpha_1)(\gamma + \alpha_2)}{\gamma(\gamma + \beta)}. \]  

(4.15)

The angle of PI-lead compensator at the design point \( \gamma_0 \) is

\[ \angle G_c(\gamma_0) = \angle(\gamma_0 + \alpha_1) + \angle(\gamma_0 + \alpha_2) - \angle(\gamma_0) - \angle(\gamma_0 + \beta) \]  

(4.16)

and the PI-lead pole and zeros are computed from

\[ \theta_{z_1} + \theta_{z_2} - \theta_{p_1} = \angle(\gamma_0 + \alpha_1) + \angle(\gamma_0 + \alpha_2) - \angle(\gamma_0) - \angle(\gamma_0 + \beta) \]

\[ = \theta_c + \angle(\gamma_0) = \theta_{c, PI-lead} \].  

(4.17)

Since there are four unknown parameters and only two constraints, there are two degrees of freedom in selecting the PI-lead parameters. As with the PID design, two design methods are considered.

In the first method, one of the PI-lead zeros is chosen, most likely to cancel a plant pole or shape the loop response. Given the PI-lead zero \( \alpha_1 \), the remaining PI-lead zero and pole must satisfy

\[ \theta_{z_2} - \theta_{p_1} = \theta_{c, PI-lead} - \theta_{z_1} \]  

(4.18)

where the quantities on the right side of equation (4.18) are known, and the quantities on the left side of equation (4.18) are unknown. Comparing equations (4.18) with the angle constraint in equation (4.11) for the lead compensator design the selection of \( \alpha_2 \) and \( \beta \) are equivalent to a lead compensator design for the desired compensator angle.
\[ \hat{\theta}_c := \theta_{c,\text{PI-lead}} - \theta_z. \] Using this analogy, the selection of \( \alpha \) must satisfy a constraint similar to equation (4.10). After \( \alpha \) is selected, \( \beta \) is computed from

\[ \beta = \sigma_0 + \frac{\omega_0}{\tan(\theta_z - \hat{\theta}_c)}. \] (4.19)

In the second method, the two PI-lead zeros are assumed to be co-located, \( \alpha_1 = \alpha_2 = \alpha \), and the angle constraint in equation (4.17) becomes \( 2\theta_z - \theta_{p_i} = \theta_{c,\text{PI-lead}} \). This angle constraint is analogous to the angle constraint in equation (4.9) for the lead compensator. Mimicking the lead design procedure, the PI-lead zeros must be to the right of the PID zeros for the co-located case, i.e., \( \alpha_{\text{PI-lead}} < \alpha_{\text{PID}} \). After the PI-lead zeros are chosen, the PI-lead pole is computed from \( \beta = \sigma_0 + \frac{\omega_0}{\tan(\theta_{p_i})} \), where \( \theta_{p_i} = 2\theta_z - \theta_{c,\text{PI-lead}} \). [15, 18].
CHAPTER 5
EXAMPLES

5.1 Proportional-Derivative (PD) Examples

5.1.1 Continuous Time Root Locus

In this example, a marginally stable open-loop continuous time transfer function

\[ G_p(s) = \frac{1}{s(s + 1)} \]

is given. The goal is to use root locus to design a PD compensator where the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds. As shown in Figure 8, from the closed-loop specification, the design point is selected to be at \( s_o = -1 + j. \)

The root locus response of the uncompensated system cannot achieve the desired design point, so a compensator is needed. The closed-loop step-response with the PD compensator designed for the desired design point is shown in Figure 9. The response has an overshoot of 6.7% and a settling time of 3.73 seconds, which does not satisfy the percent overshoot specification. One solution for meeting the design specification is to choose a new design point in the specification region shown in Figure 8. One of the many design points in the specification region is \( s_o = -1 + j0.5. \) After designing a PD compensator at this design point, the closed-loop step-response has an overshoot of 0.764% and a settling time of 2.6 seconds. The response meets both specifications. The results in Table 4 show that the design point is achieved and the design specifications are met. The desired compensator angle \( \theta_c \) is computed from (3.14), the compensator zero is computed from the (4.5) and (4.6), and the gain \( K \) is computed from the magnitude criterion in (3.15).
Figure 8. The specification region for the closed-loop system.

Figure 9. Step-response with PD compensator designed using continuous time root locus.
### Table 4

**Final PD Compensator in Continuous Time**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>1.07 Radian</td>
</tr>
<tr>
<td>PD Zero ($\alpha_{pp}$)</td>
<td>1.25</td>
</tr>
<tr>
<td>K (gain)</td>
<td>1</td>
</tr>
<tr>
<td>Desired Design Point ($s_0$)</td>
<td>$-1 + j 0.5$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-1 \pm j 0.5$</td>
</tr>
<tr>
<td>Settling Time ($T_s$) measured</td>
<td>2.60 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$) measured</td>
<td>0.764%</td>
</tr>
</tbody>
</table>

### 5.1.2 Delta Operation Root Locus

The continuous transfer functions in Section 5.1.1 can be converted to the $\gamma$-domain by (2.31) and results in

$$G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$$

when the sampled period, $\Delta$, is 0.01 seconds. In this section, root locus is used to design a PD compensator where the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds.

Using the second-order standard and assumptions in the continuous domain, these specifications should be met by a design point of $s_0 = -1 + j 0.5$. From Table 2, this corresponds to $\gamma_0 = -0.9963 + j 0.4950$ for the delta domain root locus. At the desired design point, the closed-loop step-response has an overshoot of 0.753% and a settling time of 2.62 seconds. The plots of closed-loop step-response and the compensated root locus are shown in Figure 10. As can be seen, the design point is achieved, and the design
specifications are met. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero from (4.5) and (4.6), and the gain $K$ from the magnitude criterion in (3.15). The results are shown in Table 5. The Matlab M-files used for design can be found in Appendix A.

![Step Response of PD Compensator](image)

![Root Locus Response of PD Compensator](image)

Figure 10. Step response and root locus plot with PD compensator designed using $\delta$-operator root locus.
### TABLE 5

**FINAL PD RESULTS IN DELTA FORM**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>1.097 rad</td>
</tr>
<tr>
<td>PD Zero ($\alpha_{PD}$)</td>
<td>1.2391</td>
</tr>
<tr>
<td>$K$ (gain)</td>
<td>0.9987</td>
</tr>
<tr>
<td>Desired Design Point ($\gamma_0$)</td>
<td>$-0.9963 + j\ 0.4950$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-0.9963 \pm j\ 0.4950$</td>
</tr>
<tr>
<td>Settling Time ($T_s_{measured}$)</td>
<td>2.62 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)</td>
<td>0.753%</td>
</tr>
</tbody>
</table>

### 5.1.3 Continuous Time Frequency Response

Given the frequency response of the open-loop transfer function $G_s(s) = \frac{1}{s(s+1)}$, the goal is to design a PD compensator where the closed-loop step-response has an overshoot of less than 4.32 percent and a settling time of less than 4 seconds in frequency domain. The specifications are summarized in Table 6.

### TABLE 6

**DESIGN SPECIFICATIONS**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Overshoot ($P.O._{measured}$)</td>
<td>4.32%</td>
</tr>
<tr>
<td>Settling Time ($T_s_{measured}$)</td>
<td>4 sec</td>
</tr>
</tbody>
</table>

The $s$-domain design point, $s_0 = -1 + j1$, is used to specify a phase margin of 65.5° from (3.10) and a gain crossover frequency of 0.9102 rad/sec from (3.11). The
design of a PD compensator at the desired design point, $\gamma_0 = j0.9102$ with a phase margin of $65.5^\circ$ results in an overshoot of $7.34\%$ and a settling time of $5.17$ seconds, which does not meet the closed-loop design specifications.

To meet the transient specifications, we move the design point to $\gamma_0 = j1.2441$ with a phase margin of $76^\circ$. The open-loop Bode plot with the resulting PD compensator is shown in Figure 11. It has a phase margin of $76^\circ$ and a gain crossover frequency of $1.24$ rad/sec, which achieves the desired design. The closed-loop step-response is shown in Figure 11 and has a settling time of $3.98$ seconds and an overshoot of $3.64\%$. The response meets both design specifications. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero from (4.5) and (4.6), and the gain $K$ from the magnitude criterion in (3.15). Results are summarized in Table 7. The Matlab M-files used for design can be found in Appendix A.
Figure 11. Step-response and Bode plot with PD compensator designed using continuous time Bode plot.

### TABLE 7

**FINAL CONTINUOUS PD COMPENSATOR FROM FREQUENCY DOMAIN DESIGN**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>0.6499 rad</td>
</tr>
<tr>
<td>PD Zero ($\alpha_{PD}$)</td>
<td>1.6369</td>
</tr>
<tr>
<td>K (gain)</td>
<td>0.9658</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>3.98 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>3.64%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>76°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>1.2441 rad/sec</td>
</tr>
</tbody>
</table>
5.1.4 Delta Operation Frequency Response

If the continuous time transfer function in Section 5.1.3 is sampled with a sampling period of $\Delta = 0.01$ seconds, the equivalent $\delta$-domain transfer function can be found from (2.31) and is given by $G_\delta(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$. The goal is to design a PD compensator in the frequency domain, where the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds.

Using the standard second-order assumptions, these specifications correspond to a gain crossover frequency of 1.2441 rad/sec and a phase margin of 76°. From Table 2, this corresponds to $\gamma_o = -0.0077 + j 1.2440$ with a phase margin of 76° for the $\delta$-domain frequency response design. As shown in Figure 12, the open loop Bode plot with the resulting PD compensator, $\gamma_o$, has a phase margin of 76.5° and the closed-loop step-response has a settling time of 3.95 seconds and an overshoot of 3.39%, meeting both design specifications. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero from (4.5) and (4.6), the gain $K$ from the magnitude criterion in (3.15). Results are summarized in Table 8. The Matlab M-files used for design can be found in Appendix A.
Figure 12. Step-response and Bode plot with PD compensator designed using \( \delta \)-operator Bode plot.

<table>
<thead>
<tr>
<th>( \delta )-OPERATOR PD COMPENSATOR FROM FREQUENCY DOMAIN DESIGN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ( (\theta_c) )</td>
</tr>
<tr>
<td>PD Zero ( (\alpha_{PD}) )</td>
</tr>
<tr>
<td>K (gain)</td>
</tr>
<tr>
<td>Settling Time ( (T_s)_{measured} )</td>
</tr>
<tr>
<td>Percent Overshoot ( (P.O.)_{measured} )</td>
</tr>
<tr>
<td>Phase Margin ( (PM)_{estimate} )</td>
</tr>
<tr>
<td>Gain Crossover Frequency ( (\omega_{gc})_{estimate} )</td>
</tr>
</tbody>
</table>
5.2 Lead Examples

5.2.1 Continuous Time Root Locus

In this example, a marginally stable open-loop continuous time transfer function

\[ G_p(s) = \frac{1}{s(s + 1)} \]

is given. The goal is to use root locus to design a lead compensator where the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds. As shown in Figure 8, the design point is selected to be \( s_0 = -1 + j1 \), from the closed-loop specifications.

The root locus response of the uncompensated system cannot achieve the desired design point, so a compensator is needed. The closed-loop step-response with the lead compensator designed for the desired design point is shown in Figure 13. The response has an overshoot of 1.29 % and a settling time of 2.64 seconds, which meets the percent overshoot and the settling time specification. The results in Table 9 show that the design point is achieved and the design specifications are met. The desired compensator angle \( \theta_c \) is computed from (3.14), the compensator zero is computed from the (4.6) and (4.10), the compensator pole from (4.11), and the gain \( K \) is computed from the magnitude criterion in (3.15). The Matlab M-files used for design can be found in Appendix B.
Step Response of Lead Compensator

Time (sec)

Amplitude

-1.8
-1.6
-1.4
-1.2
-1.0
-0.8
-0.6
-0.4
-0.2
0

2
4

Root Locus Response of Lead Compensator

Real Axis

Imaginary Axis

0
1
2
3
4
5
6
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

System: sys
Settling Time (sec): 2.64
System: sys
Peak amplitude: 1.01
Overshoot (%): 1.29
At time (sec): 3.51

Figure 13. Step response and root locus plot with lead compensator designed using continuous time root locus.

TABLE 9

FINAL LEAD COMPENSATOR IN CONTINUOUS TIME

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>-5.4978 radian</td>
</tr>
<tr>
<td>Lead Zero ($\alpha_{lead}$)</td>
<td>0.8500</td>
</tr>
<tr>
<td>Lead Pole ($\beta_{lead}$)</td>
<td>1.7391</td>
</tr>
<tr>
<td>K (gain)</td>
<td>1.7391</td>
</tr>
<tr>
<td>Desired Design Point ($s_0$)</td>
<td>$-1 + j1$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-1 \pm j1$</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>2.64 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>1.29%</td>
</tr>
</tbody>
</table>
5.2.2 Delta Operator Root Locus

The continuous transfer functions in Section 5.2.1 can be converted to the $\gamma$-domain by (2.31) and results in

$$G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$$

when the sampled period $\Delta$ is 0.01 seconds. In this section, root locus is used to design a lead compensator where the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds.

Using the standard second-order assumptions in the continuous domain, these specifications should be met by a design point of $s_0 = -1 + j1$. From Table 2, this corresponds to $\gamma_0 = -1 + j0.9900$ for the delta domain root locus. At the desired design point, the closed-loop step-response has an overshoot of 1.27% and a settling time of 2.65 seconds. The plots of closed-loop step-response and the compensated root locus are shown in Figure 14. As can be seen, the design point is achieved, and the design specifications are met. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero from (4.6) and (4.10), the compensator pole from (4.11), and the gain $K$ from the magnitude criterion in (3.15). The results are shown in Table 10. The Matlab M-files used for design can be found in Appendix B.
Figure 14. Step response and root locus plot with lead compensator designed using $\delta$-operator root locus.

TABLE 10

FINAL LEAD COMPENSATOR IN DELTA FORM

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>-5.4928 radian</td>
</tr>
<tr>
<td>Lead Zero ($\alpha_{lead}$)</td>
<td>0.8500</td>
</tr>
<tr>
<td>Lead Pole ($\beta_{lead}$)</td>
<td>1.7372</td>
</tr>
<tr>
<td>K (gain)</td>
<td>1.7372</td>
</tr>
<tr>
<td>Desired Design Point ($\gamma_0$)</td>
<td>$-1 + j \cdot 0.9900$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-1 \pm j \cdot 0.9900$</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>2.65 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>1.27%</td>
</tr>
</tbody>
</table>
Continuous Time Frequency Response

Given the frequency response of the open-loop transfer function \( G_p(s) = \frac{1}{s(s + 1)} \),

the goal is to design a lead compensator where the closed-loop step-response has an
overshoot of less than 4.32 percent and a settling time of less than 4 seconds in frequency
domain. The \( s \)-domain design point, \( s_0 = -1 + j1 \), is used to specify a phase margin of
65.5° from (3.10) and a gain crossover frequency of 0.9102 rad/sec from (3.11). From
Table 2, this corresponds to \( \gamma_0 = j0.9102 \) with a phase margin of 65.5°.

The open-loop Bode plot with the resulting lead compensator is shown in Figure
15. It has a phase margin of 65.5° and a gain crossover frequency of 0.91 rad/sec, which
achieves the desired design. The closed-loop step-response is shown in Figure 15 and has
a settling time of 3.93 seconds and an overshoot of 3.94%. The response meets both
design specifications. The desired compensator angle \( \theta_c \) is computed from (3.14), the
compensator zero from (4.6) and (4.10), the compensator pole from (4.11), and the gain
\( K \) from the magnitude criterion in (3.15). Results are summarized in Table 11. The
Matlab M-files used for design can be found in Appendix B.
Figure 15. Step-response and Bode plot with lead compensator designed using continuous time Bode plot.

**TABLE 11**
CONTINUOUS LEAD COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>0.3113 rad</td>
</tr>
<tr>
<td>Lead Zero ($\alpha_{lead}$)</td>
<td>0.8500</td>
</tr>
<tr>
<td>Lead Pole ($\beta_{lead}$)</td>
<td>1.6339</td>
</tr>
<tr>
<td>K (gain)</td>
<td>1.8484</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>3.93 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>3.94%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>65.5°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{ge}$)$_{estimate}$</td>
<td>0.9102 rad/sec</td>
</tr>
</tbody>
</table>
5.2.4 Delta Operation Frequency Response

If the continuous time transfer function in Section 5.2.3 is sampled with a sampling period of $\Delta = 0.01$ seconds, the equivalent $\delta$-domain transfer function can be found from (2.31), and is given by $G_\delta(p) = \frac{0.005p + 0.995}{p^2 + 0.995p}$. The goal is to design a lead compensator in frequency domain, where the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds.

Using the standard second-order assumptions, these specifications correspond to a gain crossover frequency of $0.9102$ rad/sec and a phase margin of $65.5^\circ$. From Table 2, this corresponds to $\gamma_0 = -0.0041 + j0.9102$ with a phase margin of $65.5^\circ$ for the $\delta$-domain frequency response design. As shown in Figure 16, the open loop Bode plot with the resulting lead compensator has a phase margin of $65.9^\circ$ and the closed-loop step-response has a settling time of 3.91 seconds and an overshoot of 3.74%, meeting both design specifications. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero from (4.6) and (4.10), the compensator pole from (4.11), the gain $K$ from the magnitude criterion in (3.15). Results summarized are given in Table 12. The Matlab M-files used for design can be found in Appendix B.
Figure 16. Step-response and Bode plot with lead compensator designed using $\delta$-operator Bode plot.

TABLE 12

$\delta$-OPERATOR LEAD COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>0.3159 rad</td>
</tr>
<tr>
<td>Lead Zero ($\alpha_{lead}$)</td>
<td>0.8500</td>
</tr>
<tr>
<td>Lead Pole ($\beta_{lead}$)</td>
<td>1.6462</td>
</tr>
<tr>
<td>K (gain)</td>
<td>1.8596</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>3.91 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>3.74%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>65.5°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>0.9102 rad/sec</td>
</tr>
</tbody>
</table>
5.3 PI Examples

5.3.1 Continuous Time Root Locus

In this example, a marginally stable open-loop continuous time transfer function
\[ G_p(s) = \frac{1}{s(s + 1)} \]
is given. The goal is to use root locus to design a PI compensator where the closed-loop step-response is specified to have an overshoot of less than 40 percent and a settling time of less than 10 seconds and the steady-state for a ramp input must be zero. From the closed-loop specification, the design point is selected to at \( s_0 = -0.4 + j1.37 \).

The uncompensated system cannot meet the steady state error requirement, so a compensator is needed to increase the system type. The closed-loop step-response with the PI compensator designed for the desired design point has an overshoot of 48.5% and a settling time of 8.16 seconds, which does not satisfy the percent overshoot specification. One solution for meet in the design specification is to choose a new design point to the right side of uncompensated system. One of the many design points in the specification region is \( s_0 = -0.36 + j0.66 \). After designing a PI compensator at this point, the closed-loop step-response has an overshoot of 39.4% and a settling time of 8.22 seconds, as it shown in Figure 17. The response meets both specifications. The results in Table 13 show that the design point is achieved and the design specifications are met. The desired compensator angle \( \theta_c \) is computed from (3.14), the compensator zero is computed from the (4.5) and (4.6), and the gain \( K \) is computed from the magnitude criterion in (3.15). The Matlab M-files used for design can be found in Appendix C.
Figure 17. Step response and root locus plot with PI compensator designed using continuous time root locus.

TABLE 13

FINAL PI COMPENSATOR IN CONTINUOUS TIME

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>-0.2707 radian</td>
</tr>
<tr>
<td>PI Zero ($\alpha_{pi}$)</td>
<td>0.2064</td>
</tr>
<tr>
<td>K (gain)</td>
<td>0.7668</td>
</tr>
<tr>
<td>Desired Design Point ($s_0$)</td>
<td>-0.36 + j 0.66</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>-0.36 ± j 0.66</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>8.22 sec</td>
</tr>
<tr>
<td>Percent Overshoot (P.O.)$_{measured}$</td>
<td>39.4%</td>
</tr>
</tbody>
</table>
5.3.2 Delta Operator Root Locus

The continuous transfer functions in Section 5.3.1 can be converted to the $\gamma$-domain by (2.31) and results in the $G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$ when the sampled period $\Delta$ is 0.01 seconds. In this section, root locus is used to design a PI compensator where the closed-loop step-response is specified to have an overshoot of less than 40 percent and a settling time of less than 10 seconds and the steady-state for a ramp input must be zero.

Using the standard second-order assumptions in the continuous domain, these specifications should be met by a design point of $s_0 = -0.36 + j 0.66$. From Table 2, this corresponds to $\gamma_0 = -0.3615 + j 0.6576$ for the delta domain root locus. At the desired design point, the closed-loop step-response has an overshoot of 39% and a settling time of 8.23 seconds. The plots of closed-loop step-response and the compensated root locus are shown in Figure 20. As can be seen, the design point is achieved, and the design specifications are met. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero from (4.5) and (4.6), and the gain $K$ from the magnitude criterion in (3.15). The results are shown in Table 14. The Matlab M-files used for design can be found in Appendix C.
Figure 18. Step response and root locus plot with PI compensator designed using $\delta$-operator root locus.

### TABLE 14

**FINAL PI COMPENSATOR IN DELTA FORM**

<table>
<thead>
<tr>
<th>Compensator Angle ($\theta_c$)</th>
<th>-0.2674 radian</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI Zero ($\alpha_{pp}$)</td>
<td>0.2039</td>
</tr>
<tr>
<td>K (gain)</td>
<td>0.7656</td>
</tr>
<tr>
<td>Desired Design Point ($\gamma_0$)</td>
<td>-0.3615 + j 0.6576</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>-0.3615 ± j 0.6576</td>
</tr>
<tr>
<td>Settling Time ($T_s^{\text{measured}}$)</td>
<td>8.28 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O._{\text{measured}}$)</td>
<td>39%</td>
</tr>
</tbody>
</table>
5.3.3 Continuous Time Frequency Response

Given the frequency response of the open-loop transfer function \( G_p(s) = \frac{1}{s(s + 1)} \),
the goal is to design a PI compensator where the closed-loop step-response has an overshoot of less than 40 percent and a settling time of less than 10 seconds in frequency domain and the steady-state for a ramp input must be zero.

The \( s \)-domain design point, \( s_0 = -0.4 + j1.37 \), is used to specify a phase margin of 31.22° from (3.10) and a gain crossover frequency of 1.3198 rad/sec from (3.11). The design of a PI compensator at the desired design point, \( \gamma_0 = j1.3198 \) with a phase margin of 31.22° results in an overshoot of 44.3% and a settling time of 8.25 seconds, which does not meet the closed-loop overshoot specifications.

To meet the transient specifications, we move the design point to \( \gamma_0 = j1.1343 \) with a phase margin of 38.43°. The open-loop Bode plot with the resulting PI compensator is shown in Figure 19. It has a phase margin of 38.4° and a gain crossover frequency of 1.1343 rad/sec, which achieves the desired design. The closed-loop step-response is shown in Figure 19, and has a settling time of 9.5 seconds and an overshoot of 33.2%. The response meets both design specifications. The desired compensator angle \( \theta_c \) is computed from (3.14), the compensator zero from (4.5) and (4.6), and the gain \( K \) from the magnitude criterion in (3.15). Results are summarized in Table 15. The Matlab M-files used for design can be found in Appendix C.
Figure 19. Step-response and Bode plot with PI compensator designed using continuous time Bode plot.

**TABLE 15**

FINAL CONTINUOUS PI COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ( \theta_c )</td>
<td>-0.0588 rad</td>
</tr>
<tr>
<td>PI Zero ( \alpha_{PZ} )</td>
<td>0.0588</td>
</tr>
<tr>
<td>K (gain)</td>
<td>1.713</td>
</tr>
<tr>
<td>Settling Time ( T_s )_{measured}</td>
<td>9.5 sec</td>
</tr>
<tr>
<td>Percent Overshoot ( P.O. )_{measured}</td>
<td>33.2%</td>
</tr>
<tr>
<td>Phase Margin ( PM )_{estimate}</td>
<td>38.43°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ( \omega_{gc} )_{estimate}</td>
<td>1.1343 rad/sec</td>
</tr>
</tbody>
</table>
5.3.4 Delta Operation Frequency Response

If the continuous time transfer function in Section 5.3.3 is sampled with a sampling period of $\Delta = 0.01$ seconds, the equivalent $\delta$-domain transfer function can be found from (2.31) and is given by $G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$. The goal is to design a PI compensator in frequency domain, where the closed-loop step-response is specified to have an overshoot of less than 40 percent and a settling time of less than 10 seconds and the steady-state for a ramp input must be zero.

Using the standard second-order assumptions, these specifications correspond to a gain crossover frequency of 1.1343 rad/sec and a phase margin of 38.43°. From Table 2, this corresponds to $\gamma_0 = -0.0064 + j1.1343$ with a phase margin of 38.43° for $\delta$-domain frequency response design. The open loop Bode plot with the resulting PI compensator, as shown in Figure 20, has a phase margin of 39° and the closed-loop step-response has a settling time of 9.44 seconds and an overshoot of 32.2%, meeting both design specifications. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero from (4.5) and (4.6), the gain $K$ from the magnitude criterion in (3.15). Results with the PI compensator in the delta operation frequency response are given in Table 16. The Matlab M-files used for design can be found in Appendix C.
Figure 20. Step-response and Bode plot with PI compensator designed using $\delta$-operator Bode plot.

### TABLE 16

$\delta$-OPERATOR PI COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>-0.0462 rad</td>
</tr>
<tr>
<td>PI Zero ($\alpha_{py}$)</td>
<td>0.0524</td>
</tr>
<tr>
<td>K (gain)</td>
<td>1.7139</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>9.44 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>33.2%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>38.43°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>1.1343 rad/sec</td>
</tr>
</tbody>
</table>
5.4 PID Examples with Co-located Zeros

5.4.1 Continuous Time Root Locus

In this example, a marginally stable open-loop continuous time transfer function
\[ G_p(s) = \frac{1}{s(s+1)} \] is given. The goal is to use root locus to design a PID compensator where the two PID zeros are assumed to be co-located, \( \alpha_1 = \alpha_2 = \alpha \), and the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds and the steady-state for a ramp input must be zero. From the closed-loop specification, the design point is selected to be at \( s_0 = -1 + j1 \).

The root locus response of uncompensated system cannot achieve the desired design point, so a compensator is needed. The closed-loop step-response with the PID compensator designed for the desired design point has an overshoot of 20.8% and a settling time of 3.46 seconds, which does not satisfy the percent overshoot specification. One of the many design points in the specification region is \( s_0 = -0.37 + j0.1 \). After designing a PID compensator at this design point, the closed-loop step-response as is shown in Figure 21 has an overshoot of 1.74% and a settling time of 0.549 seconds. The response meets both specifications. The results in Table 17 show that the design point is achieved and the design specifications are met. The desired compensator angle \( \theta_c \) is computed from (3.14), the compensator zero is computed from the (4.5) and (4.6), and the gain \( K \) is computed from the magnitude criterion in (3.15). The Matlab M-files used for design can be found in Appendix D.
Figure 21. Step response and root locus plot with PID compensator designed using continuous time root locus.

TABLE 17

FINAL PID COMPENSATOR IN CONTINUOUS TIME

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>-0.1065 radian</td>
</tr>
<tr>
<td>PID Zeros ($\alpha_{PID}$)</td>
<td>0.3887</td>
</tr>
<tr>
<td>K (gain)</td>
<td>9.0526</td>
</tr>
<tr>
<td>Desired Design Point ($s_0$)</td>
<td>$-0.37 + j 0.1$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-0.37 \pm j 0.1$</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>0.549 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>1.74%</td>
</tr>
</tbody>
</table>
5.4.2 Delta Operator Root Locus

The continuous transfer functions in Section 5.4.1 can be converted to the $\gamma$-domain by (2.31) and results in the $G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$ when the sampled period $\Delta$ is 0.01 seconds. In this section, root locus is used to design a PID compensator where the two PID zeros are assumed to be co-located, $\alpha_1 = \alpha_2 =: \alpha$, and the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds and the steady-state for a ramp input must be zero and the steady-state for a ramp input must be zero.

Using the standard second-order assumptions in the continuous domain, these specifications should be met by a design point of $s_0 = -0.37 + j 0.1$. From Table 2, this corresponds to $\gamma_0 = -0.3694 + j 0.0996$ for the delta domain root locus. At the desired design point, the closed-loop step-response has an overshoot of 1.74% and a settling time of 0.532 seconds. The plots of closed-loop step-response and the compensated root locus are shown in Figure 22. As can be seen, the design point is achieved, and the design specifications are met. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero from (4.5) and (4.6), and the gain $K$ from the magnitude criterion in (3.15). The results are shown in Table 18. The Matlab M-files used for design can be found in Appendix D.
Figure 22. Step response and root locus plot with PID compensator designed using $\delta$-operator root locus.

### TABLE 18
**FINAL PID COMPENSATOR IN DELTA FORM**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>-0.1060 radian</td>
</tr>
<tr>
<td>PID Zeros ($\alpha_{PID}$)</td>
<td>0.3879</td>
</tr>
<tr>
<td>K (gain)</td>
<td>9.0560</td>
</tr>
<tr>
<td>Desired Design Point ($\gamma_0$)</td>
<td>$-0.3694 + j0.0996$</td>
</tr>
<tr>
<td>Closed-loop Poles</td>
<td>$-0.3694 \pm j0.0996$</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>0.532 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>1.74%</td>
</tr>
</tbody>
</table>
### 5.4.3 Continuous Time Frequency Response

Given the frequency response of the open-loop transfer function \( G_p(s) = \frac{1}{s(s + 1)} \), the goal is to design a PID compensator where the two PID zeros are assumed to be co-located, \( \alpha_1 = \alpha_2 = \alpha \), and the closed-loop step-response has an overshoot of less than 14 percent and a settling time of less than 4 seconds in frequency domain and the steady-state for a ramp input must be zero.

The \( s \)-domain design point, \( s_0 = -1 + j1 \), is used to specify a phase margin of 65.5° from (3.10) and a gain crossover frequency of \( \omega_{gc} = 0.9102 \text{ rad/sec} \) from (3.11). The design of a PID compensator at the desired design point, \( \gamma_0 = j0.9102 \), with a phase margin 65.5° results in an overshoot of 21.8% and a settling time of 7.98 seconds, which does not meet the closed-loop overshoot specifications.

To meet the transient specifications, we move the design point to \( \gamma_0 = j3.9013 \) with a phase margin of 38.43°. The open-loop Bode plot with the resulting PID compensator is shown in Figure 23. It has a phase margin of 76.3° and a gain crossover frequency of 3.9013 rad/sec, which achieves the desired design. The closed-loop step-response is shown in Figure 23 and has a settling time of 2.84 seconds and an overshoot of 13.6%. The response meets both design specifications. The desired compensator angle \( \theta_c \) is computed from (3.14), the compensator zero from (4.5) and (4.6), and the gain \( K \) from the magnitude criterion in (3.15). Results are summarized in Table 19. The Matlab M-files used for design can be found in Appendix D.
Figure 23. Step-response and Bode plot with PID compensator designed using continuous time Bode plot.

TABLE 19

CONTINUOUS PID COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

| Compensator Angle ($\theta_c$) | 1.0807 rad |
| PID Zeros ($\alpha_{PID}$) | 0.9756 |
| K (gain) | 3.7904 |
| Settling Time ($T_s$)$_{measured}$ | 2.84 sec |
| Percent Overshoot ($P.O.$)$_{measured}$ | 13.6% |
| Phase Margin ($PM$)$_{estimate}$ | 76.3° |
| Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$ | 3.9013 rad/sec |
5.4.4 Delta Operation Frequency Response

If the continuous time transfer function in Section 5.4.3 is sampled with a sampling period of $\Delta = 0.01$ seconds, the equivalent $\delta$-domain transfer function can be found from (2.31) and is given by $G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$. The goal is to design a PID compensator where the two PID zeros are assumed to be co-located, $\alpha_1 = \alpha_2 = \alpha$ and the closed-loop step-response is specified to have an overshoot of less than 14 percent and a settling time of less than 4 seconds and the steady-state for a ramp input must be zero.

Using standard second-order assumptions, these specifications correspond to a gain crossover frequency of 3.9013 rad/sec and a phase margin of 76.3°. From Table 2, this corresponds to $\gamma_0 = -0.0761 + j 3.9003$ with a phase margin of 76.3° for $\delta$-domain frequency response design. As shown in Figure 24, the open loop Bode plot with the resulting PID compensator has a phase margin of 77.5° and the closed-loop step-response has a settling time of 3.15 seconds and an overshoot of 12%, meeting both design specifications. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero from (4.5) and (4.6), and the gain $K$ from the magnitude criterion in (3.15). Results with the PID compensator in the delta operation frequency response are given in Table 20. The Matlab M-files used for design can be found in Appendix D.
Figure 24. Step-response and Bode plot with PID compensator designed using $\delta$-operator Bode plot.

### TABLE 20

$\delta$-OPERATOR PID COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>1.1002 rad</td>
</tr>
<tr>
<td>PID Zeros ($\alpha_{PID}$)</td>
<td>0.8912</td>
</tr>
<tr>
<td>K (gain)</td>
<td>3.8609</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>3.15 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>12%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>76.3°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>3.9013 rad/sec</td>
</tr>
</tbody>
</table>
5. 5  PID Examples with Non-located Zeros

5.5.1 Continuous Time Root Locus

In this example, a marginally stable open-loop continuous time transfer function
\[ G_p(s) = \frac{1}{s(s+1)} \] is given. The goal is to use root locus to design a PID compensator
where the one of the PID zeros is chosen to cancel a plant pole and the remaining PID
zero is computed from (4.6) and (4.14), the closed-loop step-response is specified to have
an overshoot of less than 15 percent and a settling time of less than 4 seconds and the
steady-state error for a ramp input must be zero. From the closed-loop specification, the
design point is selected to be at \( s_o = -1 + j1.65 \).

The root locus response of uncompensated system cannot achieve the desired
design point, so a compensator is needed. The closed-loop step-response with the PID
compensator designed for the desired design point has an overshoot of 28.8% and a
settling time of 3.86 seconds, which does not satisfy the percent overshoot specification.
One of the many design points in the specification region is \( s_o = -2 + j0.1 \). After
designing a PID compensator at this design point, the closed-loop step-response as is
shown in Figure 25, has an overshoot of 13.6% and a settling time of 2.69 seconds. The
response meets both specifications. The results in Table 21 show that the design point is
achieved and the design specifications are met. The desired compensator angle \( \theta_c \) is
computed from (3.14), the compensator zero \( \alpha_2 \), is computed from (4.6) and (4.14), and
the gain \( K \) is computed from the magnitude criterion in (3.15). The Matlab M-files used
for design can be found in Appendix E.
Figure 25. Step response and root locus plot with PID compensator designed using continuous time root locus.

**TABLE 21**

**FINAL PID COMPENSATOR IN CONTINUOUS TIME**

<table>
<thead>
<tr>
<th>Compensator Angle ($\theta_c$)</th>
<th>-3.2912 radian</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID Zeros ($\alpha_1$), ($\alpha_2$)</td>
<td>1 and 1.0025</td>
</tr>
<tr>
<td>K (gain)</td>
<td>4.000</td>
</tr>
<tr>
<td>Desired Design Point ($s_0$)</td>
<td>$-2 + j \ 0.1$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-2 \pm j \ 0.1$</td>
</tr>
<tr>
<td>Settling Time ($T_s$)_measured</td>
<td>2.69 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)_measured</td>
<td>13.6%</td>
</tr>
</tbody>
</table>
5.5.2 Delta Operator Root Locus

The continuous transfer functions in Section 5.5.1 can be converted to the $\gamma$-domain by (2.31) and results in the

$$G_p(\gamma) = \frac{0.005 \gamma + 0.995}{\gamma^2 + 0.995 \gamma}$$

when the sampled period $\Delta$ is 0.01 seconds. In this section, root locus is used to design a PID compensator where the one of the PID zeros is chosen to cancel a plant pole and the remaining PID zero is computed from (4.6) and (4.14), the closed-loop step-response is specified to have an overshoot of less than 15 percent and a settling time of less than 4 seconds and the steady-state error for a ramp input must be zero.

Using standard second-order assumptions in the continuous domain, these specifications should be met by a design point of $s_0 = -2 + j \ 0.1$. From Table 2, this corresponds to $\gamma_0 = -1.9802 + j \ 0.0980$ for the delta domain root locus. At the desired design point, the closed-loop step-response has an overshoot of 13.8% and a settling time of 2.73 seconds. The plots of closed-loop step-response and the compensated root locus are shown in Figure 26. As can be seen, the design point is achieved, and the design specifications are met. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero $\alpha_2$ is computed from (4.6) and (4.14), and the gain $K$ is computed from the magnitude criterion in (3.15). The results are shown in Table 22. The Matlab M-files used for design can be found in Appendix E.
Figure 26. Step response and root locus plot with PID compensator designed using δ-operator root locus.

TABLE 22

FINAL PID COMPENSATOR IN DELTA FORM

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle (θ₁)</td>
<td>-3.2907 radian</td>
</tr>
<tr>
<td>PID Zeros (α₁), (α₂)</td>
<td>0.9950 and 0.9874</td>
</tr>
<tr>
<td>K (gain)</td>
<td>3.9204</td>
</tr>
<tr>
<td>Desired Design Point (γ₀)</td>
<td>-1.9802 + j 0.0980</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>-1.9802 ± j 0.0980</td>
</tr>
<tr>
<td>Settling Time (Tₛ)measured</td>
<td>2.73 sec</td>
</tr>
<tr>
<td>Percent Overshoot (P.O.)measured</td>
<td>13.8%</td>
</tr>
</tbody>
</table>
5.5.3 Continuous Time Frequency Response

Given the frequency response of the open-loop transfer function \(G_p(s) = \frac{1}{s(s + 1)}\), the goal is to design a PID compensator where the one of the PID zeros is chosen to cancel a plant pole and the remaining PID zero is computed from (4.6) and (4.14), and the closed-loop step-response has an overshoot of less than 15 percent and a settling time of less than 4 seconds in frequency domain.

The \(s\)-domain design point, \(s_0 = -1 + j1\), is used to specify a phase margin of 65.5° from (3.10) and a gain crossover frequency of \(\omega_{gc} = 0.9102\) rad/sec from (3.11). The design of a PID compensator at the desired design point, \(\gamma_0 = j0.9102\) with a phase margin 65.5° results has an overshoot of 20.8% and a settling time of 8.35 seconds, which does not meet the closed-loop overshoot specifications.

To meet the transient specifications, we move the design point to \(\gamma_0 = j3.8875\) with a phase margin of 76.3°. The open-loop Bode plot with the resulting PID compensator is shown in Figure 27. It has a phase margin of 76.3° and a gain crossover frequency of 3.89 rad/sec, which achieves the desired design. The closed-loop step-response is shown in Figure 23, and has a settling time of 2.84 seconds and an overshoot of 13.6%. The response meets both design specifications. The desired compensator angle \(\theta_c\) is computed from (3.14), the compensator zero \(\alpha_2\) from (4.6) and (4.14), and the gain \(K\) is computed from the magnitude criterion in (3.15). Results are summarized in Table 23. The Matlab M-files used for design can be found in Appendix E.
Figure 27. Step-response and Bode plot with PID compensator designed using continuous time Bode plot.

TABLE 23

CONTINUOUS PID COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>1.0807rad</td>
</tr>
<tr>
<td>PID Zeros ($\alpha_1$, $\alpha_2$)</td>
<td>1 and 0.9446</td>
</tr>
<tr>
<td>K (gain)</td>
<td>3.7776</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>2.85 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>13.5%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>76.3°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>3.8875 rad/sec</td>
</tr>
</tbody>
</table>
**5.5.4 Delta Operation Frequency Response**

If the continuous time transfer function in Section 5.5.3 is sampled with a sampling period of $\Delta = 0.01$ seconds, the equivalent $\delta$-domain transfer function can be found from (2.31), and is given by $G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$. The goal is to design a PID compensator where the one of the PID zeros is chosen, at the location to cancel a plant pole and the remaining PID zero is computed from (4.6) and (4.14), and the closed-loop step-response is specified to have an overshoot of less than 15 percent and a settling time of less than 4 seconds and the steady-state error for a ramp input must be zero.

Using the standard second-order assumptions, these specifications correspond to a gain crossover frequency of 3.8875 rad/sec and a phase margin of 76.3°. From Table 2, this corresponds to $\gamma_0 = -0.0756 + j\ 3.8865$ for the delta domain with a phase margin of 76.3° for $\delta$-domain frequency response design. The open loop Bode plot with the resulting PID compensator, as shown in Figure 28, has a phase margin of 77.5° and the closed-loop step-response has a settling time of 3.18 seconds and an overshoot of 12%, meeting both design specifications. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zeros from (4.6) and (4.14), and the gain $K$ from the magnitude criterion in (3.15). Results are summarized in Table 24. The Matlab M-files used for design can be found in Appendix E.
Figure 28. Step-response and Bode plot with PID compensator designed using $\delta$-operator Bode plot.

### TABLE 24

$\delta$-OPERATOR PID COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>1.1001 rad</td>
</tr>
<tr>
<td>PID Zeros ($\alpha_1$, $\alpha_2$)</td>
<td>0.9950 and 0.7831</td>
</tr>
<tr>
<td>K (gain)</td>
<td>3.8450</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>3.18 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>12%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>76.3°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>3.8875 rad/sec</td>
</tr>
</tbody>
</table>
5.6 PI-Lead Examples with Co-located Zeros

5.6.1 Continuous Time Root Locus

In this example, a marginally stable open-loop continuous time transfer function

\[ G_p(s) = \frac{1}{s(s + 1)} \]

is given. The goal is to use root locus to design a PI-lead compensator where the two PI-lead zeros are assumed to be co-located, \( \alpha_1 = \alpha_2 = \alpha \), the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds and the steady-state error for a ramp input must be zero. From the closed-loop specifications, the design point is selected to be at \( s_0 = -1 + j1 \).

The root locus response of uncompensated system cannot achieve the desired design point, so a compensator is needed. The closed-loop step-response with the PI-lead compensator designed for the desired design point has an overshoot of 15.3 % and a settling time of 16.3 seconds, which does not satisfy the design specification. One of the many design points in the specification region is \( s_0 = -8 + j8 \). After designing a PI-lead compensator at this design point, the closed-loop step-response as is shown in Figure 29 has an overshoot of 1.82% and a settling time of 1.67 seconds. The response meets both specifications. The results in Table 25 show that the design point is achieved and the design specifications are met. The desired compensator angle \( \theta_c \) is computed from (3.14), the compensator zeros are computed from (4.6) and (4.10), the compensator poles from (4.11), and the gain \( K \) is computed from the magnitude criterion in (3.15). The Matlab M-files used for design can be found in Appendix F.
Figure 29. Step response and root locus plot with PI-Lead compensator designed using continuous time root locus.

**TABLE 25**

FINAL PI-LEAD COMPENSATOR IN CONTINUOUS TIME

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>-4.7790 radian</td>
</tr>
<tr>
<td>PI-Lead Zeros ($\alpha_{\text{PI-lead}}$)</td>
<td>0.25</td>
</tr>
<tr>
<td>PI-Lead Poles ($\beta_{\text{PI-lead}}$)</td>
<td>15.4613</td>
</tr>
<tr>
<td>K (gain)</td>
<td>119.9787</td>
</tr>
<tr>
<td>Desired Design Point ($s_0$)</td>
<td>$-8 + j 8$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-8 \pm j 8$</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{\text{measured}}$</td>
<td>1.67 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{\text{measured}}$</td>
<td>1.82%</td>
</tr>
</tbody>
</table>
5.6.2 Delta Operator Root Locus

The continuous transfer functions in Section 5.4.1 can be converted to the $\gamma$-domain by (2.31) and result in
\[ G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma} \]
when the sampled period $\Delta$ is 0.01 seconds. In this section, root locus is used to design a PI-lead compensator where the two PI-lead zeros are assumed to be co-located, $\alpha_1=\alpha_2=\alpha$, and the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds and the steady-state error for a ramp input must be zero.

Using standard second-order assumptions in the continuous domain, these specifications should be met by a design point of
\[ s_0 = -8 + j 8. \]
From Table 2, this corresponds to $\gamma_0 = -7.9836 + j 7.3771$ for the delta domain root locus. At the desired design point, the closed-loop step-response has an overshoot of 1.89% and a settling time of 1.72 seconds. The plots of closed-loop step-response and the compensated root locus are shown in Figure 30. As can be seen, the design point is achieved, and the design specifications are met. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zeros from (4.6) and (4.10), the compensator poles from (4.11), and the gain $K$ from the magnitude criterion in (3.15). The results are shown in Table 26. The Matlab M-files used for design can be found in Appendix F.
Figure 30. Step response and root locus plot with PI-Lead compensator designed using $\delta$-operator root locus.

TABLE 26

FINAL PI-LEAD COMPENSATOR IN DELTA FORM

<table>
<thead>
<tr>
<th>Compensator Angle ($\theta_c$)</th>
<th>-4.7384 radian</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI-Lead Zeros ($\alpha_{PI-lead}$)</td>
<td>0.25</td>
</tr>
<tr>
<td>PI-lead Pole ($\beta_{pi-lead}$)</td>
<td>14.8788</td>
</tr>
<tr>
<td>K (gain)</td>
<td>111.0327</td>
</tr>
<tr>
<td>Desired Design Point ($\gamma_0$)</td>
<td>$-7.9836 + j 7.3771$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-7.9836 \pm j 7.3771$</td>
</tr>
<tr>
<td>Settling Time ($T_s_{measured}$)</td>
<td>1.72 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O._{measured}$)</td>
<td>1.89%</td>
</tr>
</tbody>
</table>
5.6.3 Continuous Time Frequency Response

Given the frequency response of the open-loop transfer function \( G_p(s) = \frac{1}{s(s + 1)} \), the goal is to design a PI-lead compensator where the two PI-lead zeros are assumed to be co-located, \( \alpha_1 = \alpha_2 = \alpha \), the closed-loop step-response has an overshoot of less than 4.32 percent and a settling time of less than 4 seconds in frequency domain and the steady-state error for a ramp input must be zero.

The \( s \)-domain design point, \( s_0 = -1 + j1 \), is used to specify a phase margin of 65.5° from (3.10) and a gain crossover frequency of \( \omega_{gc} = 0.9102 \) rad/sec from (3.11). The design of a PI-lead compensator at the desired design point, \( \gamma_0 = j0.9102 \) with a phase margin 65.5°, results has an overshoot of 9.4% and a settling time of 18.4 seconds, which does not meet the closed-loop specifications.

To meet the transient specifications, we move the design point to \( \gamma_0 = j7.2814 \) with a phase margin of 65.5°. The open-loop Bode plot with the resulting PI-lead compensator is shown in Figure 31. It has a phase margin of 65.5° and a gain crossover frequency of 7.28 rad/sec, which achieves the desired design. The closed-loop step-response is shown in Figure 31 and has a settling time of 1.57 seconds and an overshoot of 1.74 %. The response meets both design specifications. The desired compensator angle \( \theta_c \) is computed from (3.14), the compensator zeros from (4.6) and (4.10), the compensator pole from (4.11), and the gain \( K \) from the magnitude criterion in (3.15). Results are summarized in Table 27. The Matlab M-files used for design can be found in Appendix F.
Figure 31. Step-response and Bode plot with PI-Lead compensator designed using continuous time Bode plot.

**TABLE 27**

**CONTINUOUS PI-LEAD COMPENSATOR FROM FREQUENCY DOMAIN DESIGN**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>1.0072 rad</td>
</tr>
<tr>
<td>PI-Lead Zeros ($\alpha_{pl-lead}$)</td>
<td>0.25</td>
</tr>
<tr>
<td>PI-Lead Poles ($\beta_{pl-lead}$)</td>
<td>13.4910</td>
</tr>
<tr>
<td>K (gain)</td>
<td>112.544</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>1.57 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>1.74%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>65.5°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>7.2814 rad/sec</td>
</tr>
</tbody>
</table>
5.6.4 Delta Operation Frequency Response

If the continuous time transfer function in Section 5.6.3 is sampled with a sampling period of $\Delta = 0.01$ seconds, the equivalent $\delta$-domain transfer function can be found from (2.31) and is given by

$$G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}.$$  

The goal is to design a PI-lead compensator where the two PI-lead zeros are assumed to be co-located, $\alpha_1 = \alpha_2 = \alpha$, the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds and the steady-state error for a ramp input must be zero and the steady-state error for a ramp input must be zero.

Using standard second-order assumptions, these specifications correspond to a gain crossover frequency of 7.2814 rad/sec and a phase margin of 65.5°. From Table 2, this corresponds to $\gamma_0 = -0.2650 + j 7.2750$ with a phase margin of 65.5° for $\delta$-domain frequency response design. As shown in Figure 32, the open loop Bode plot with the resulting PID compensator has a phase margin of 68.4° and the closed-loop step-response has a settling time of 1.59 seconds and an overshoot of 1.72%, meeting both design specifications. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zeros from (4.6) and (4.10), the compensator pole from (4.11), and the gain $K$ from the magnitude criterion in (3.15). Results are summarized in Table 28. The Matlab M-files used for design can be found in Appendix F.
Figure 32. Step-response and Bode plot with PI-Lead compensator designed using $\delta$-operator Bode plot.

### TABLE 28

$\delta$-OPERATOR PI-LEAD COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>1.0436 rad</td>
</tr>
<tr>
<td>PI-Lead Zeros ($\alpha_{PI-lead}$)</td>
<td>0.25</td>
</tr>
<tr>
<td>PI-Lead Pole ($\beta_{PI-lead}$)</td>
<td>13.7459</td>
</tr>
<tr>
<td>K (gain)</td>
<td>112.7874</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>1.59 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>1.72%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>65.5°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>7.2814 rad/sec</td>
</tr>
</tbody>
</table>
5.7 PI-Lead Examples with Non-located Zeros

5.7.1 Continuous Time Root Locus

In this example, a marginally stable open-loop continuous time transfer function \( G_p(s) = \frac{1}{s(s+1)} \) is given. The goal is to use root locus to design a PI-lead compensator where one of the PI-lead zero is chosen to cancel a plant pole and the remaining PI-lead zeros is computed from (4.18), the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds and the steady-state error for a ramp input must be zero and the steady-state error for a ramp input must be zero. From the closed-loop specification, the design point is selected to be at \( s_0 = -1 + j1 \).

The root locus of the uncompensated system cannot achieve the desired design point, so a compensator is needed. The closed-loop step-response with the PI compensator designed for the desired design point has an overshoot of 17.9% and a settling time of 13.1 seconds, which does not satisfy the design specification. One of the many design points in the specification region is \( s_0 = -8 + j5 \). After designing a PI-lead compensator at this design point, the closed-loop step-response as is shown in Figure 33 has an overshoot of 3.32% and a settling time of 2.44 seconds. The response meets both specifications. The results in Table 29 show that the design point is achieved and the design specifications are met. The desired compensator angle \( \theta_c \) is computed from (3.14), the compensator zero \( \alpha_2 \), from (4.18), the compensator pole \( \beta \) from (4.19), and the gain \( K \) computed from the magnitude criterion in (3.15). The Matlab M-files used for design can be found in Appendix G.
Figure 33. Step response and root locus plot with PI-Lead compensator designed using continuous time root locus.

TABLE 29

FINAL PI-LEAD COMPENSATOR IN CONTINUOUS TIME

<table>
<thead>
<tr>
<th>Compensator Angle ($\theta_c$)</th>
<th>-4.3204 radian</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID zeros ($\alpha_1$), ($\alpha_2$)</td>
<td>1 and 0.1550</td>
</tr>
<tr>
<td>PI-Lead Poles ($\beta$)</td>
<td>16.1594</td>
</tr>
<tr>
<td>K (gain)</td>
<td>91.5511</td>
</tr>
<tr>
<td>Desired Design Point ($s_0$)</td>
<td>$-8 + j5$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-8 \pm j5$</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>2.44 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>3.32%</td>
</tr>
</tbody>
</table>
5.7.2 Delta Operator Root Locus

The continuous transfer functions in Section 5.7.1 can be converted to the $\gamma$-domain by (2.31) and results in $G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$ when the sampled period $\Delta$ is 0.01 seconds. In this section, root locus is used to design a PI-lead compensator where one of the PI-lead zeros is chosen to cancel a plant pole and the remaining PI-lead zero is computed from (4.18), the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds and the steady-state error for a ramp input must be zero and the steady-state error for a ramp input must be zero.

Using the standard second-order assumptions in the continuous domain, these specifications should be met by a design point of $s_0 = -8 + j 5$. From Table 2, this corresponds to $\gamma_0 = -7.8037 + j 4.6137$ for the delta domain root locus. At the desired design point, the closed-loop step-response has an overshoot of 3.24% and a settling time of 2.62 seconds. The plots of closed-loop step-response and the compensated root locus are shown in Figure 34. As can be seen, the design point is achieved, and the design specifications are met. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero $\alpha_2$ from (4.18), the compensator pole from (4.19), and the gain $K$ from the magnitude criterion in (3.15). The results are shown in Table 30. The Matlab M-files used for design can be found in Appendix G.
Figure 34. Step response and root locus plot with PI-Lead compensator designed using δ-operator root locus.

**TABLE 30**

**FINAL PI-LEAD COMPENSATOR IN DELTA FORM**

<table>
<thead>
<tr>
<th>Compensator Angle ($\theta_c$)</th>
<th>-4.2951 radian</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI-Lead Zeros ($\alpha_1$, $\alpha_2$)</td>
<td>0.9950 and 0.1550</td>
</tr>
<tr>
<td>PI-Lead Pole ($\beta$)</td>
<td>15.3433</td>
</tr>
<tr>
<td>K (gain)</td>
<td>85.0324</td>
</tr>
<tr>
<td>Desired Design Point ($\gamma_0$)</td>
<td>$-7.8037 + j\ 4.6137$</td>
</tr>
<tr>
<td>Closed-Loop Poles</td>
<td>$-7.8037 \pm j\ 4.6137$</td>
</tr>
<tr>
<td>Settling Time ($T_s$) measured</td>
<td>2.62 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$) measured</td>
<td>3.24%</td>
</tr>
</tbody>
</table>
5.7.3 Continuous Time Frequency Response

Given the frequency response of the open-loop transfer function $G_p(s) = \frac{1}{s(s + 1)}$, the goal is to design a PI-lead compensator where one of the PI-lead zeros is chosen to cancel a plant pole and the remaining PI-lead zero is computed from (4.18), the closed-loop step-response has an overshoot of less than 4.32 percent and a settling time of less than 4 seconds and the steady-state error for a ramp input must be zero and the steady-state error for a ramp input must be zero.

The $s$-domain design point, $s_0 = -1 + j1$, is used to specify a phase margin of 65.5° from (3.10) and a gain crossover frequency of $\omega_{gc} = 0.9102 \text{ rad/sec}$ from (3.11). The design of a PI-lead compensator at the desired design point, $\gamma_0 = j0.9102$ with a phase margin 65.5°, results in an overshoot of 12.8% and a settling time of 14.3 seconds, which does not meet the closed-loop specifications.

To meet the transient specifications, we move the design point to $\gamma_0 = j5.2821$ with a phase margin of 71.7°. The open-loop Bode plot with the resulting PI-lead compensator is shown in Figure 35. It has a phase margin of 71.7° and a gain crossover frequency of 5.28 rad/sec, which achieves the desired design. The closed-loop step-response is shown in Figure 35 and has a settling time of 2.63 seconds and an overshoot of 2.85%. The response meets both design specifications. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero $\alpha_2$ from (4.18), the compensator pole from (4.19), and the gain $K$ from the magnitude criterion in (3.15). Results are summarized in Table 31. The Matlab M-files used for design can be found in Appendix G.
Figure 35. Step-response and Bode plot with PI-Lead compensator designed using continuous time Bode plot.

**TABLE 31**

CONTINUOUS PI-LEAD COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>1.0648 rad</td>
</tr>
<tr>
<td>PI-lead Zeros ($\alpha_1$, $\alpha_2$)</td>
<td>1 and 0.1550</td>
</tr>
<tr>
<td>PI-Lead Pole ($\beta$)</td>
<td>17.7311</td>
</tr>
<tr>
<td>K (gain)</td>
<td>97.6827</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>2.63 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>2.85%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>71.7°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>5.2821 rad/sec</td>
</tr>
</tbody>
</table>
5.7.4 Delta Operation Frequency Response

If the continuous time transfer function in Section 5.6.3 is sampled with a sampling period of $\Delta = 0.01$ seconds, the equivalent $\delta$-domain transfer function can be found from (2.31), and is given by $G_p(\gamma) = \frac{0.005\gamma + 0.995}{\gamma^2 + 0.995\gamma}$. The goal is to design a PI-lead compensator where one of the PI-lead zeros is chosen to cancel a plant pole and the remaining PI-lead zero is computed from (4.18), and the closed-loop step-response is specified to have an overshoot of less than 4.32 percent and a settling time of less than 4 seconds and the steady-state error for a ramp input must be zero and the steady-state error for a ramp input must be zero.

Using the standard second-order assumptions, these specifications correspond to a gain crossover frequency of 5.2821 rad/sec and a phase margin of 71.7°. From Table 2, this corresponds to $\gamma_0 = -0.1395 + j5.2796$ with a phase margin of 71.7° for $\delta$-domain frequency response design. The open loop Bode plot with the resulting PI-lead compensator, as shown in Figure 32, has a phase margin of 73.4° and the closed-loop step-response has a settling time of 2.69 seconds and an overshoot of 2.71%, meeting both design specifications. The desired compensator angle $\theta_c$ is computed from (3.14), the compensator zero $\alpha_2$ from (4.18), the compensator pole from (4.19), and the gain $K$ from the magnitude criterion in (3.15). Results are summarized Table 32. The Matlab M-files used for design can be found in Appendix G.
**Step Response of PI-Lead Compensator**

![Step Response Plot]

- System: sys
- Peak amplitude: 1.03
- Overshoot (%): 2.71
- At time (sec): 0.84
- Settling Time (sec): 2.69

**Bode Response of PI-Lead Compensator**

![Bode Plot]

- System: sys
- Phase Margin (deg): 73.4
- Delay Margin (sec): 0.244
- At frequency (rad/sec): 5.25
- Closed Loop Stable? Yes

Figure 36. Step-response and Bode plot with PI-Lead compensator designed using $\delta$-operator Bode plot.

**TABLE 28**

$\delta$-OPERATOR PI-LEAD COMPENSATOR FROM FREQUENCY DOMAIN DESIGN

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compensator Angle ($\theta_c$)</td>
<td>1.0912 rad</td>
</tr>
<tr>
<td>PI-Lead Zeros ($\alpha_1, \alpha_2$)</td>
<td>0.9950 and 0.1550</td>
</tr>
<tr>
<td>PI-Lead Pole ($\beta$)</td>
<td>17.8603</td>
</tr>
<tr>
<td>K (gain)</td>
<td>98.2150</td>
</tr>
<tr>
<td>Settling Time ($T_s$)$_{measured}$</td>
<td>2.69 sec</td>
</tr>
<tr>
<td>Percent Overshoot ($P.O.$)$_{measured}$</td>
<td>2.71%</td>
</tr>
<tr>
<td>Phase Margin ($PM$)$_{estimate}$</td>
<td>71.7°</td>
</tr>
<tr>
<td>Gain Crossover Frequency ($\omega_{gc}$)$_{estimate}$</td>
<td>5.2821 rad/sec</td>
</tr>
</tbody>
</table>
CHAPTER 6

CONCLUSION

Overview

The advantage of using the $\delta$ operator to parameterize discrete time systems is that the $\delta$ operator description provides a common framework for continuous-time and discrete-time controller design. As a result of designing compensators using the $\delta$ operator, the discrete time model for a sampled continuous time system is very close to the continuous time system model. Thus, continuous time insights can be used in discrete design [12, 16, 17, 19].

Continuous-time and discrete-time design methods were streamlined with the objective of moving the students’ focus from the computational procedures of the algorithms to the more important issues of control system design such as compensator selection and closed-loop performance. Established continuous-time and discrete-time control concepts were presented in a logical progression that facilitates comprehension for students in first courses in continuous-time and discrete-time control. The design procedures for five compensators—lead, PI, lag, PID, and PI-lead—were developed from a PD design procedure. These procedures are analogous to the continuous-time design procedures presented in [13, 14]. This common design approach helps to bridge the gap between the more intuitive continuous-time design and the more practical direct discrete-time design.
REFERENCES
LIST OF REFERENCES


APPENDIX A

Matlab M-files for PD Design

• PD Compensator Design Function

function [k,numc,denc,numcl,dencl]=pd(num,den,T,sd,PM)
% PD (proportional-derivative) compensator design in unified
% notation for continuous or discrete time systems using root locus
% or Bode procedures. Continuous time systems are represented in
% the s-domain and discrete time systems are represented in the delta
% domain.
%
[k,numc,denc,numcl,dencl]=pd(num,den,T,sd,PM) calculates
the control gain K, the PD compensator Gc(s), and the closed-loop
response such that:

1. sd, is the design point at the transient response specification
   in root locus design and in Bode design.
% unified design point
---------------------------------------------------------------
| T=0  |  T=0 |
---------------------------------------------------------------
| root locus | s0  | (exp(sd*T)-1)/T |
---------------------------------------------------------------
| Bode   | jwgc | (exp(j*wgc*T)-1)/T |
---------------------------------------------------------------

2. K= 1/(abs(Gc(sd))*abs(Gp(sd)))
   where,
   sd, is the design point
   Gc(sd), is the pd compensator for the unified system evaluated
   at the design point
   Gp(sd), is the open-loop plant system for the unified system
   evaluated at the design point

3. the compensator transfer function for the unified system is
   Gc(gama)= tf(numc,denc)

4. the closed-loop transfer function for the unified system is
   Gcl(gama)=tf(numcl,dencl)

5. the open-loop transfer function for the unified system is
   Gp(gama)=tf(num,den)

6. T, is the sampling time for discrete-time systems and 0
   for continuous-time systems

7. PM, is the phase margin specification for Bode designs and 0
   for root locus designs

% Authors: T. Emami and J. M. Watkins
% Date: 02-14-06

93
%open-loop transfer function
[num,den]=tfchk(num,den);

% Evaluation of open-loop transfer function at the design point in unified transforms theory
X=(polyval(num,sd,T))/(polyval(den,sd,T));

% The desired angle in the angle constraint
phi=-pi+PM;

% The desired compensator angle
thc=phi-angle(X);

% Real part of design point
sigmad=-real(sd);

% Imaginary part of design point
wd=imag(sd);

% theta p is the angle of compensator denominator
thp=atan(T*wd/(1-T*sigmad));

% theta z is the angle of compensator numerator
thz=thc+thp;

% Evaluation of compensator zero
z=sigmad+wd/tan(thz);

% The pd compensator design of unified system
numc=[1 z];
denc=[T 1];

% Evaluation of pd compensator of unified system at the design point
Y=(polyval(numc,sd,T))/(polyval(denc,sd,T));

% The control gain
k=1/(abs(X)*abs(Y));

% open-loop transfer function of unified system after compensator design
numol=conv(numc,num);
denol=conv(denc,den);

% check dimension of open-loop system
[numol,denol]=tfchk(numol,denol);

% closed-loop transfer function of unified system after compensator design
numcl=k*numol;
dencl=denol+numcl;

if T==0

% Open-loop transfer function of unified system
Gop=tf(numol, denol);
else

% Convert open-loop transfer function of unified system to z domain
[numoz, denoz] = del2z(numol, denol, T);

% the open-loop transfer function of unified system
Gop=tf(numoz, denoz, T);
end

% the closed-loop transfer function of unified system
Gcl=minreal(k*Gop/(1+k*Gop));

% plot the step response of unified system
subplot(2,1,1)
step(numcl, dencl)
title('Step Response of PD Compensator')

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

% plot the root locus response of unified system
subplot(2,1,2)
if PM==0
    rlocus(numol, denol)
title('Root Locus Response of PD Compensator')

    % Plot the closed-loop poles and zeros
    hold on
    pol=roots(dencl);
    zer=roots(numcl);
    plot(real(pol), imag(pol), '^', real(zer), imag(zer), 's')
    hold off
end

% plot the Bode response of the unified system
else
    bode(k*numol, denol)
title('Bode Response of PD Compensator')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
• **PD Compensator Design Script Files**

%pd compensator for continuous time domain

%Open-loop transfer function
Gp=tf([1],[1 1 0]);
[num,den]=tfdata(Gp,'v');

%Design point in continuous time domain
sd=-1+j*0.5;

% Sampling time period in continuous time domain
T=0;

% Phase margin in continuous time root locus
PM=0;

% plot the step response and the root locus in continuous time domain
figure (1)
% call pd function to see all responses of all variables in continuous time domain
[kc,numc,denc,numcl,denc1]=pd(num,den,T,sd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%delta
%pd compensator for delta operator in gama domain

% sampling time period in delta operator
T=0.01;

%Convert open-loop transfer function in continuous time domain to delta operator
[dnum,dden]=c2del(num,den,T);

%Design point in delta operator
dsd=(exp(sd*T)-1)/T;

% plot the step response and the root locus in gama domain
figure(2)
% call pd function to see all responses of all variables in gama domain
[dk,dnumc,ddenc,dnumcl,ddencl]=pd(dnum,dden,T,dsd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%frequency
% pd compensator for frequency response in continuous time domain

%sampling period in frequency response in continuous time domain
T=0;
    sd=-2.5+j*0.4
%real and imaginary part of design point
sigmad=-real(sd);
wd=imag(sd);

%Determine the phase margin and gain cross over frequency
 teta=atan(-wd/sigmad);
zeta=cos(teta);
wn=sigmad/zeta;
PM=atan(2*zeta/sqrt(-2*zeta^2+sqrt(1+4*zeta^4)))
    wgc=2*zeta*wn/tan(PM);

%Design point in frequency response of continuous time domain
sd=j*wgc;

% plot the step response and the Bode response in frequency continuous
time
% domain
figure(3)

% call pd function to see all responses of all variables in frequency
% continuous time domain
[kf,numcf,dencf,numclf,dencclf]=pd(num,den,T,sd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%delta frequency

% sampling time period in frequency delta operator
T=0.01;

%Design point in frequency response of delta operator
dsd=(exp(j*wgc*T)-1)/T;

% plot the step response and the Bode response in frequency delta
operator
    figure(4)

% call pd function to see all responses of all variables in delta
% frequency domain
[dkf,dnumcf,ddencf,dnumclf,ddencclf]=pd(dnum,dden,T,dsd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
APPENDIX B

Matlab M-files for Lead design

- **Lead Compensator Design Function**

```matlab
function [k,numc,denc,numcl,dencl]=lead(num,den,T,sd,PM,zl)

%Lead compensator design in unified notation for continuous or discrete
%time systems using root locus or Bode procedures. Continuous time
%systems are represented in the s-domain and discrete time systems are
%represented in the delta domain.
%
% [k,numc,denc,numcl,dencl]=lead(num,den,T,sd,PM,zl) calculates
% the control gain K, the Lead compensator Gc(s), and the closed-loop
% response such that:
%
% 1. sd, is the design point at the transient response specification
%    in root locus design and in Bode design.
%    % unified design point
%    % -----------------------------------------------
%    % | T=0   | T~0 |
%    % -----------------------------------------------
%    % root locus | s0   | (exp(sd*T)-1)/T |
%    % -----------------------------------------------
%    % Bode       | jwgc | (exp(j*wgc*T)-1)/T |
%    % -----------------------------------------------
%
% 2. K= 1/(abs(Gc(sd))*abs(Gp(sd)))
%    where,
%    sd, is the design point
%    Gc(sd), is the lead compensator for the unified system
%    evaluated
%    at the design point
%    Gp(sd), is the open-loop plant system for the unified system
%    evaluated at the design point
%
% 3. the compensator transfer function for the unified system is
%    Gc(gama)= tf(numc,denc)
%
% 4. the closed-loop transfer function for the unified system is
%    Gcl(gama)=tf(numcl,dencl)
%
% 5. the open-loop transfer function for the unified system is
%    Gp(gama)=tf(num,den)
%
% 6. T, is the sampling time for discrete-time systems and 0
%    for continuous-time systems
%
% 7. PM, is the phase margin specification for Bode design and 0
%    for root locus designs
%
% 8. zl, is the lead compensator
```
% Authors: T. Emami and J. M. Watkins
% Date: 02-14-06

%open-loop transfer function
[num,den]=tfchk(num,den);

% Evaluation of open-loop transfer function at the design point in unified transforms theory
X=(polyval(num,sd,T))/(polyval(den,sd,T));

% The desired angle in the angle constraint
phi=-pi+PM;

% The desired compensator angle
thc=phi-angle(X);

% real part of design point
sigmad=-real(sd);

% imaginary part of design point
wd=imag(sd);

% theta p is the angle of compensator denominator
thp=atan(T*wd/(1-T*sigmad));

% theta z is the angle of compensator numerator
thz=thc+thp;

% Evaluation of compensator zero
z=sigmad+wd/tan(thz);

% the pd compensator design of unified system
numcpd=[1 z];
dencpd=[T 1];

% theta z is the angle of lead compensator numerator zero
thz=atan(wd/(zl-sigmad));

% theta p is the angle of lead compensator numerator pole
thp=thz-thc;

% determine the lead compensator pole
pl=sigmad+wd/tan(thp);

% the lead compensator design of unified system
numc=[1 zl];
denc=[1 pl];

% Evaluation of pd compensator of unified system at the design point
Y=(polyval(numc,sd,T))/(polyval(denc,sd,T));
%The control gain
k = 1/(abs(X)*abs(Y));

% open-loop transfer function of unified system after compensator design
numol = conv(numc, num);
denol = conv(denc, den);
[numol, denol] = tfchk(numol, denol);

% closed-loop transfer function of unified system after compensator design
numcl = k*numol;
dencl = denol + numcl;

if T == 0

% Open-loop transfer function of unified system
Gop = tf(numol, denol);
else

% Convert open-loop transfer function of unified system to z domain
[numoz, denoz] = del2z(numol, denol, T);

% The open-loop transfer function of unified system
Gop = tf(numoz, denoz, T);
end

% The closed-loop transfer function of unified system
Gcl = minreal(k*Gop/(1+k*Gop));

% Plot the step response of unified system
subplot(2, 1, 1)
step(numcl, dencl)
title('Step Response of Lead Compensator')

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

% Plot the root locus response of unified system
subplot(2, 1, 2)

if PM == 0
rlocus(numol, denol)
title('Root Locus Response of Lead Compensator')

% Plot the closed-loop poles and zeros
hold on
pol = roots(dencl);
zer = roots(numcl);
plot(real(pol), imag(pol), '^', real(zer), imag(zer), 's');
hold off

% Plot the Bode response of the unified system
else
bode(k*numol, denol)
• **Lead Compensator Design Script Files**

%Lead compensator for continuous time domain

%Open-loop transfer function
Gp=tf([1],[1 1 0]);
[num,den]=tfdata(Gp,'v');

%Design point in continuous time domain
sd=-1+j*1;

% Sampling time period in continuous time domain
T=0;

% Phase margin in continuous time root locus
PM=0;

% choose lead zero to the right side of pd zero
zl=0.85;

% plot the step response and the root locus in continuous time domain
figure(1)
% call lead function to see all responses of all variables in continuous
% time domain
[kc,numc,denc,numcl,dencl]=lead(num,den,T,sd,PM,zl)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 
%%%%
%delta
%lead compensator for delta operator in gama domain

% Sampling time period in delta operator
T=0.01;

%Convert open-loop transfer function in continuous time domain to delta
%operator
[dnum,dden]=c2del(num,den,T);

%Design point in delta operator
dsd=(exp(sd*T)-1)/T;

% plot the step response and the root locus in gama domain
figure(2)
% call leads function to see all responses of all variables in gama domain
% domain
[dk,dnumc,ddenc,dnumcl,ddencl]=lead(dnum,dden,T,dsd,PM,zl)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%frequency
% Lead compensator for frequency response in continuous time domain

% sampling period in frequency response in continuous time domain
T=0;

% real and imaginary part of design point
sigmad=-real(sd);
wd=imag(sd);

% Determine the phase margin and gain cross over frequency
eta=atan(-wd/sigmad);
zeta=cos(eta);
wn=sigmad/zeta;
PM=atan(2*zeta/sqrt(-2*zeta^2+sqrt(1+4*zeta^4)));  
wgc=2*zeta*wn/tan(PM);

% Design point in frequency response of continuous time domain
sd=j*wgc;

% plot the step response and the Bode response in frequency continuous
time
% domain
figure(3)

% call pd function to see all responses of all variables in frequency
% continuous time domain
[kf,numcf,dencf,numclf,denclf]=lead(num,den,T,td,td,PM,zl)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% delta frequency

% Sampling frequency period in frequency delta operator
T=0.01;

% Design point in frequency response of delta operator
dsd=(exp(j*wgc*T)-1)/T;

% plot the step response and the Bode response in frequency delta
operator
figure(4)

% call pd function to see all responses of all variables in delta
% frequency domain
[dkf,dnumcf,ddencf,dnumclf,ddenclf]=lead(dnum,dden,T,dsd,PM,zl)
APPENDIX C

Matlab M-files for PI Design

- **PI Compensator Design Function**

```matlab
function [k,numc,denc,numcl,dencl,yy,xx,zz]=newpi(num,den,T,sd,PM)

%NEWPI (proportional-integral) compensator design in unified
% notation for continuous or discrete time systems using root locus
% or Bode procedures. Continuous time systems are represented in
% the s-domain and discrete time systems are represented in the delta
% domain.
%
% [k,numc,denc,numcl,dencl]=newpi(num,den,T,sd,PM) calculates
% the control gain K, the PI compensator Gc(s), and the closed-loop
% response such that:

% 1. sd, is the design point at the transient response specification
%    in root locus design and in Bode design.
%    unified design point
%    | T=0   | T~0 |
%    --------------------------------------------
%    root locus | s0   | (exp(sd*T)-1)/T |
%    Bode      | jwgc | (exp(j*wgc*T)-1)/T |
%    --------------------------------------------
%
% 2. K= 1/(abs(Gc(sd))*abs(Gp(sd)))
% where,
% sd, is the design point
% Gc(sd), is the PI compensator for the unified system evaluated
% at the design point
% Gp(sd), is the open-loop plant system for the unified system
% evaluated at the design point

% 3. the compensator transfer function for the unified system is
% Gc(gama)= tf(numc,denc)

% 4. the closed-loop transfer function for the unified system is
% Gcl(gama)=tf(numcl,dencl)

% 5. the open-loop transfer function for the unified system is
% Gp(gama)=tf(num,den)

% 6. T, is the sampling time for discrete-time systems and 0
%    for continuous-time systems

% 7. PM, is the phase margin specification for Bode designs and 0
%    for root locus designs

% Authors: T. Emami and J. M. Watkins
% Date: 02-14-06
```
%open-loop transfer function
[num, den] = tfchk(num, den);

% Evaluation of open-loop transfer function at the design point in unified transforms theory
X = (polyval(num, sd, T)) / (polyval(den, sd, T));

% The desired angle in the angle constraint
phi = -pi + PM;

% The desired compensator angle
thc = phi - angle(X);

% real part of design point
sigmad = -real(sd);

% imaginary part of design point
wd = imag(sd);

% theta p is the angle of compensator denominator
thp = atan(-wd / sigmad);

% theta z is the angle of compensator numerator
thz = thc + thp;

% Evaluation of compensator zero
z = sigmad + wd / tan(thz);

% the PI compensator design of unified system
numc = [1 z];
denc = [1 0];

% Evaluation of pi compensator of unified system at the design point
Y = (polyval(numc, sd, T)) / (polyval(denc, sd, T));

% The control gain
k = 1 / (abs(X) * abs(Y));

% open-loop transfer function of unified system after compensator design
numol = conv(numc, num);
denol = conv(denc, den);
[numol, denol] = tfchk(numol, denol);

% closed-loop transfer function of unified system after compensator design
numcl = k * numol;
dencl = denol + numcl;

if T == 0
%Open-loop transfer function of unified system
Gop=tf(numol,denol);
else

% Convert open-loop transfer function of unified system to z domain
[numoz,denoz]=del2z(numol,denol,T);

%the open-loop transfer function of unified system
Gop=tf(numoz,denoz,T);
end

%the closed-loop transfer function of unified system
Gcl=minreal(k*Gop/(1+k*Gop));

%plot the step response of unified system
subplot(2,1,1)
step(numcl,dencl)
title('Step Response of PI Compensator')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%plot the root locus response of unified system
subplot(2,1,2)
if PM==0
    rlocus(numol,denol)
title('Root Locus Response of PI Compensator')

    % Plot the closed-loop poles and zeros
    hold on
    pol=roots(dencl);
    zer=roots(numcl);
    plot(real(pol),imag(pol),'^',real(zer),imag(zer),'s')
    hold off

%plot the Bode response of the unified system
else
    bode(k*numol,denol)
title('Bode Response of PI Compensator')
end

• **PI Compensator Design Script Files**

%PI compensator for continuous time domain

%Open-loop transfer function
Gp=tf([1],[1 1 0]);
[num,den]=tfdata(Gp,'v');

%Design point in continuous time domain
% sd=-1+j*0.5;
sd=-0.36+j*0.66
% Sampling time period in continuous time domain
T=0;

% Phase margin in continuous time root locus
PM=0;

% plot the step response and the root locus in continuous time domain
figure(1)

% call newpi function to see all responses of all variables in continuous
% time domain
[kc,numc,denc,numcl,denc1]=newpi(num,den,T,sd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%delta
%pd compensator for delta operator in gama domain

% sampling time period in delta operator
T=0.01;

%Convert open-loop transfer function in continuous time domain to delta
%operator
[dnum,dden]=c2del(num,den,T);

%Design point in delta operator
dsd=(exp(sd*T)-1)/T;

% plot the step response and the root locus in gama domain
figure(2)

% call pd function to see all responses of all variables in gama domain
% domain
[dk,dnumc,ddenc,dnumcl,ddencl]=newpi(dnum,dden,T,dsd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%frequency
% pd compensator for frequency response in continuous time domain

%sampling period in frequency response in continuous time domain
T=0;
sd=-.45+j*1.2
%real and imaginary part of design point
sigmad=-real(sd);
wd=imag(sd);

%Determine the phase margin and gain cross over frequency
teta=atan(-wd/sigmad);
zeta=cos(teta);
wn=sigmad/zeta;
PM=atan(2*zeta/sqrt(-2*zeta^2+sqrt(1+4*zeta^4)));
wgc=2*zeta*wn/tan(PM);

% Design point in frequency response of continuous time domain
sd=j*wgc;

% plot the step response and the Bode response in frequency continuous
time
% domain
figure(3)

% call newpi function to see all responses of all variables in
frequency
% continuous time domain
[kf,numcf,dencf,numclf,dencclf]=newpi(num,den,T,sd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% delta frequency

% sampling time period in frequency delta operator
T=0.01;

% Design point in frequency response of delta operator
dsd=(exp(j*wgc*T)-1)/T;

% plot the step response and the Bode response in frequency delta
operator
figure(4)

% call pd function to see all responses of all variables in delta
% frequency domain
[dkf,dnumcf,ddencf,dnumclf,ddencclf]=newpi(dnum,dden,T,dsd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
APPENDIX D

Matlab M-files for PID Design,
with Co-Located Zeros

- PID Compensator Design Function

```matlab
function [k,numc,denc,numcl,dencl]=pid(num,den,T,sd,PM)

%PID (proportional-Integral-derivative) compensator design in unified
% notation for continuous or discrete time systems using root locus
% or Bode procedures, where two zero are at the same location.
% Continuous time systems are represented in
% the s-domain and discrete time systems are represented in the delta
% domain.

[k,numc,denc,numcl,dencl]=pid(num,den,T,sd,PM) calculates
the control gain K, the PID compensator Gc(s), and the closed-loop
response such that:

1. sd, is the design point at the transient response specification
   in root locus design and in Bode design.

2. K= 1/(abs(Gc(sd))*abs(Gp(sd)))
   where,
   sd, is the design point
   Gc(sd), is the pid compensator for the unified system evaluated
   at the design point
   Gp(sd), is the open-loop plant system for the unified system
   evaluated at the design point

3. the compensator transfer function for the unified system is
   Gc(gama)= tf(numc,denc)

4. the closed-loop transfer function for the unified system is
   Gcl(gama)=tf(numcl,dencl)

5. the open-loop transfer function for the unified system is
   Gp(gama)=tf(num,den)

6. T, is the sampling time for discrete-time systems and 0
   for continuous-time systems

7. PM, is the phase margin specification for Bode designs and 0
   for root locus designs

% Authors: T. Emami and J. M. Watkins
```

% Date: 02-14-06

%open-loop transfer function
[num, den] = tfchk(num, den);

% Evaluation of open-loop transfer function at the design point in unified transform theory
X = (polyval(num, sd, T)) / (polyval(den, sd, T));

% The desired angle in the angle constraint
phi = -pi + PM;

% The desired compensator angle
thc = phi - angle(X);

% Real part of design point
sigmad = -real(sd);

% Imaginary part of design point
wd = imag(sd);

% Theta p the angle of denominator of PID compensator
thp = atan2((wd * (1 - 2 * T * sigmad)), (-sigmad + T * (sigmad^2 - wd^2)));

% Theta z is the angle of one compensator zero
thz = (thc + thp) / 2;

% Evaluation of second compensator zero
z = (sigmad + wd / tan(thz))

% PID compensator design where two zero are equal
numc = [1 2*z z^2];
denc = [T 1 0];

% Evaluation of pid compensator of unified system at the design point
Y = (polyval(numc, sd, T)) / (polyval(denc, sd, T));

% The control gain
k = 1 / (abs(X) * abs(Y));

%open-loop transfer function of unified system after compensator design
numol = conv(numc, num);
denol = conv(denc, den);
[numol, denol] = tfchk(numol, denol);

% closed-loop transfer function of unified system after compensator design
numcl = k * numol;
dencl = denol + numcl;
if T==0
    %Open-loop transfer function of unified system
    Gop=tf(numol,denol);
else
    % Convert open-loop transfer function of unified system to z domain
    [numoz,denoz]=del2z(numol,denol,T);
    %the open-loop transfer function of unified system
    Gop=tf(numoz,denoz,T);
end
%the closed-loop transfer function of unified system
Gcl=minreal(k*Gop/(1+k*Gop));

%plot the step response of unified system
subplot(2,1,1)
step(numcl,dencl)
title('Step Response of PID Compensator')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%plot the root locus response of unified system
subplot(2,1,2)
if PM==0
    rlocus(numol,denol)
title('Root Locus Response of PID Compensator')
    % Plot the closed-loop poles and zeros
    hold on
    pol=roots(dencl);
    zer=roots(numcl);
    plot(real(pol),imag(pol),'^',real(zer),imag(zer),'s')
    hold off
else
    bode(k*numol,denol)
title('Bode Response of PID Compensator')
end

---------------------------------------------------------------------------------

• PID Compensator Design Script Files

%pid compensator for continuous time domain

%Open-loop transfer function
Gp=tf([1],[1 1 0]);
[num,den]=tfdata(Gp,'v');

%Design point in continuous time domain
% sd=-1+j*1
sd=-0.37+j*0.1

% Phase margin in continuous time root locus
PM=0;
T=0;
% plot the step response and the root locus in continuous time domain
figure(1)
% call pid function to see all responses of all variables in continuous
% time domain
[kc,numc,denc,numcl,dencl]=pid(num,den,T,sd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%delta
%pd compensator for delta operator in gama domain
% Sampling time period in delta operator
T=0.01;

%Convert open-loop transfer function in continuous time domain to
delta
%operator
[dnum,dden]=c2del(num,den,T);

%Design point in delta operator
dsd=(exp(sd*T)-1)/T;

% plot the step response and the root locus in gama domain
figure(2)
% call pid function to see all responses of all variables in gama
domain
% domain
[dk,dnumc,ddenc,dnumcl,ddencl]=pid(dnum,dden,T,dsd,PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%frequency
% pd compensator for frequency response in continuous time domain

%sampling period in frequency response in continuous time domain
T=0;
sd=-8+j*0.5
%real and imaginary part of design point
sigmad=-real(sd);
wd=imag(sd);

%Determine the phase margin and gain cross over frequency
teta=atan(-wd/sigmad);
zeta = cos(teta);
wn = sigmad/zeta;
PM = atan(2*zeta/sqrt(-2*zeta^2 + sqrt(1 + 4*zeta^4)))
wgc = 2*zeta*wn/tan(PM)

% Design point in frequency response of continuous time domain
sd = j*wgc;

% plot the step response and the Bode response in frequency continuous
time
% domain
figure(3)

% call pid function to see all responses of all variables in frequency
% continuous time domain
[kf, numcf, denpf, numclf, denclf] = pid(num, den, T, sd, PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% delta frequency

% Sampling time period in frequency delta operator
T = 0.01;

% Design point in frequency response of delta operator
dsd = (exp(j*wgc*T) - 1)/T

% plot the step response and the Bode response in frequency delta
operator
figure(4)

% call pid function to see all responses of all variables in delta
% frequency domain
[dkf, dnumcf, ddencf, dnumclf, ddencclf] = pid(dnum, dden, T, dsd, PM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
APPENDIX E

Matlab M-files for PID Design,
with Non-located Zeros

- PID Compensator Design Function

```matlab
function [k,numc,denc,numcl,dencl]=pid2(num,den,T,sd,PM,z1)
%
%PID2 (proportional-integral-derivative) compensator design in unified
%notation for continuous or discrete time systems using root locus
%or Bode procedures, where the two zero are not equal. Continuous
%time systems are represented in
%the s-domain and discrete time systems are represented in the delta
%domain.

% [k,numc,denc,numcl,dencl]=pid2(num,den,T,sd,PM) calculates
%the control gain K, the PID compensator Gc(s), and the closed-loop
%response such that:

% 1. sd, is the design point at the transient response specification
%    in root locus design and in Bode design.

% unified design point
%---------------------------------------------------------------------
% | T=0 | T~=0 |
% root locus | s0 | (exp(sd*T)-1)/T |
% Bode | jwgc | (exp(j*wgc*T)-1)/T |
%---------------------------------------------------------------------

% 2. K= 1/(abs(Gc(sd))*abs(Gp(sd)))
where,
sd, is the design point
Gc(sd), is the pid compensator for the unified system evaluated
at the design point
Gp(sd), is the open-loop plant system for the unified system
evaluated at the design point

% 3. the compensator transfer function for the unified system is
Gc(gama)= tf(numc,denc)

% 4. the closed-loop transfer function for the unified system is
Gcl(gama)=tf(numcl,dencl)

% 5. the open-loop transfer function for the unified system is
Gp(gama)=tf(num,den)

% 6. T, is the sampling time for discrete-time systems and 0
for continuous-time systems

% 7. PM, is the phase margin specification for Bode designs and 0
for root locus designs
```
% Authors: T. Emami and J. M. Watkins
% Date: 02-14-06

% open-loop transfer function
[num, den] = tfchk(num, den);

% Evaluation of open-loop transfer function at the design point in unified transform theory
X = (polyval(num, sd, T)) / (polyval(den, sd, T));

% The desired angle in the angle constraint
phi = -pi + PM;

% The desired compensator angle
thc = phi - angle(X);

% Real part of design point
sigmad = -real(sd);

% Imaginary part of design point
wd = imag(sd);

% Theta p the angle of denominator of PID compensator
thp = atan2(T * wd, (1 - T * sigmad)) + atan2(wd, -sigmad);

% Theta z1 is the angle of one compensator zero
thz1 = atan2(wd, (z1 - sigmad));

% Theta z2 is the angle of second compensator zero
thz2 = thc + thp - thz1;

% Evaluation of second compensator zero
z2 = sigmad + wd / tan(thz2);

% PID2 compensator design where two zero are different
numc = [1 z1 + z2 z1 * z2];
denc = [T 1 0];

% Evaluation of pid compensator of unified system at the design point
Y = (polyval(numc, sd, T)) / (polyval(denc, sd, T));

% The control gain
k = 1 / (abs(X) * abs(Y));

% open-loop transfer function of unified system after compensator design
numol = conv(numc, num);
denol = conv(denc, den);
[numol, denol] = tfchk(numol, denol);
% closed-loop transfer function of unified system after compensator design
numcl=k*numol;
dencl=denol+numcl;

if T==0

  %Open-loop transfer function of unified system
  Gop=tf(numol,denol);
else

  % Convert open-loop transfer function of unified system to z domain
  [numoz,denoz]=del2z(numol,denol,T);

  %the open-loop transfer function of unified system
  Gop=tf(numoz,denoz,T);
end

%the closed-loop transfer function of unified system
Gcl=minreal(k*Gop/(1+k*Gop));

%plot the step response of unified system
subplot(2,1,1)
step(numcl,dencl)
title('Step Response of PID Compensator')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%plot the root locus response of unified system
subplot(2,1,2)
if PM==0
  rlocus(numol,denol)
title('Root Locus Response of PID Compensator')

  % Plot the closed-loop poles and zeros
  hold on
  pol=roots(dencl);
  zer=roots(numcl);
  plot(real(pol),imag(pol),'^',real(zer),imag(zer),'s');
  hold off

  %plot the Bode response of the unified system
else
  bode(k*numol,denol)
  title('Bode Response of PID Compensator')
end
• **PID Compensator Design Script Files**

```
%pid2 compensator for continuous time domain

%Open-loop transfer function
Gp=tf([1],[1 1 0]);
[num,den]=tfdata(Gp,'v');

%Design point in continuous time domain
sd=-2+j*0.1
sd=-1+j*1.65

% Sampling time period in continuous time domain
T=0;

% Phase margin in continuous time root locus
PM=0;

%choose one of the pid zero value to cancel one of the open-loop pole
z1=1

% plot the step response and the root locus in continuous time domain
figure(1)
% call pid2 function to see all responses of all variables in
continuous % time domain
[kc,numc,denc,numcl,dencl]=pid2(num,den,T,sd,PM,z1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%delta
%pid compensator for delta operator in gama domain

% Sampling time period in delta operator
T=0.01;

%choose one of the pid zero value to cancel one of the open-loop pole
z1=0.9950

%Convert open-loop transfer function in continuous time domain to delta
%operator
[dnum,dden]=c2del(num,den,T);

%Design point in delta operator
dsd=(exp(sd*T)-1)/T

% plot the step response and the root locus in gama domain
figure(2)
% call pid2 function to see all responses of all variables in gama domain
% domain
```
[dk, dnumc, ddenc, dnumcl, ddenc1] = pid2(dnum, dden, T, dsd, PM, z1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%frequency
% pid compensator for frequency response in continuous time domain

T = 0;

% choose one of the pid zero value to cancel one of the open-loop pole
z1 = 1
sd = -8 + j*0.1
% real and imaginary part of design point
sigmad = -real(sd);
wd = imag(sd);

determines the phase margin and gain crossover frequency
\[ \text{teta} = \tan(\text{-wd/sigmad}); \]
\[ \text{zeta} = \cos(\text{teta}); \]
\[ \text{wn} = \text{sigmad/}\text{zeta}; \]
\[ \text{PM} = \tan(2*\text{zeta}/\sqrt{-2*\text{zeta}^2+\sqrt{1+4*\text{zeta}^4}})); \]
\[ \text{wgc} = 2*\text{zeta}^*\text{wn}/\tan(\text{PM}); \]

% Design point in frequency response of continuous time domain
sd = j*wgc;

% plot the step response and the Bode response in frequency continuous
time
figure(3)

% call pid2 function to see all responses of all variables in frequency
time domain
[kf, numcf, dencf, numclf, denclf] = pid2(num, den, T, sd, PM, z1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%delta frequency

T = 0.01;

% choose one of the pid zero value to cancel one of the open-loop pole
z1 = 0.9950

% Design point in frequency response of delta operator
\[ \text{dsd} = (\exp(j*\text{wgc}^*\text{T}) - 1)/\text{T}; \]

% plot the step response and the Bode response in frequency delta
operator
figure(4)

% call pid2 function to see all responses of all variables in delta
frequency domain
[dkf, dnumcf, ddencf, dnumclf, ddenc1f] = pid2(dnum, dden, T, dsd, PM, z1)
APPENDIX F

Matlab M-files for PI-lead Design,
with Co-Located Zeros

- PI-lead Compensator Design Function

```matlab
function [k,numc,denc,numcl,dencl]=pilead(num,den,T,sd,PM,zl)

%PI-Lead (proportional-Integral-lead) compensator design in unified
notation for continuous or discrete time systems using root locus
or Bode procedures, where two zero are at the same location.
Continuous time systems are represented in the s-domain and
discrete time systems are represented in the delta domain.

[k,numc,denc,numcl,dencl]=pilead(num,den,T,sd,PM,zl) calculates
the control gain K, the PI-lead compensator Gc(s), and the closed-loop
response such that:

1. sd, is the design point at the transient response specification
   in root locus design and in Bode design.

   unified design point
   -----------------------------------------------
   | T=0 | T~=0 |
   -----------------------------------------------
   root locus | s0 | (exp(sd*T)-1)/T |
   -----------------------------------------------
   Bode | jwgc | (exp(j*wgc*T)-1)/T |
   -----------------------------------------------

2. K= 1/(abs(Gc(sd))*abs(Gp(sd)))
   where,
   sd, is the design point
   Gc(sd), is the pi-lead compensator for the unified system
   evaluated
   at the design point
   Gp(sd), is the open-loop plant system for the unified system
   evaluated at the design point

3. the compensator transfer function for the unified system is
   Gc(gama)= tf(numc,denc)

4. the closed-loop transfer function for the unified system is
   Gcl(gama)=tf(numcl,dencl)

5. the open-loop transfer function for the unified system is
   Gp(gama)=tf(num,den)

6. T, is the sampling time for discrete-time systems and 0
   for continuous-time systems

7. PM, is the phase margin specification for Bode designs and 0
   for root locus designs
```
8. z1, is the pi-lead zero location

Authors: T. Emami and J. M. Watkins
Date: 02-14-06

%open-loop transfer function
[num,den]=tfchk(num,den);

% Evaluation of open-loop transfer function at the design point in unified transform theory
X=(polyval(num,sd,T))/(polyval(den,sd,T));

% The desired angle in the angle constraint
phi=-pi+PM;

% The desired compensator angle
thc=phi-angle(X);

% real part of design point
sigmad=-real(sd);

% imaginary part of design point
wd=imag(sd);

% Determine the location of pd compensator zero
z=sigmad+wd/tan(thc);

% the angle of PI-lead compensator zero
thz1=atan(wd/(z1-sigmad));

% the angle of PI-lead compensator pole
thp1=atan(-wd/sigmad);

% the angle of PI-lead compensator pole
thp2=2*thz1-thc-thp1;

% the pole value of PI-lead compensator
p2=sigmad+wd/tan(thp2)
a=2*z1;
b=z1^2;

% the PI-lead compensator transfer function
numc=[1 a b];
denc=[1 p2 0];

% Evaluation of PI-lead compensator of unified system at the design point
Y=(polyval(numc,sd,T))/(polyval(denc,sd,T));
% The control gain
k = 1/(abs(X) * abs(Y));

% open-loop transfer function of unified system after compensator design
numol = conv(numc, num);
denol = conv(denc, den);
[numol, denol] = tfchk(numol, denol);

% closed-loop transfer function of unified system after compensator design
numcl = k * numol;
dencl = denol + numcl;

if T == 0

% Open-loop transfer function of unified system
Gop = tf(numol, denol);
else

% Convert open-loop transfer function of unified system to z domain
[numoz, denoz] = del2z(numol, denol, T);

% the open-loop transfer function of unified system
Gop = tf(numoz, denoz, T);
end

% the closed-loop transfer function of unified system
Gcl = minreal(k * Gop / (1 + k * Gop));

% plot the step response of unified system
subplot(2, 1, 1)
step(numcl, dencl)
title('Step Response of PI-Lead Compensator')

% plot the root locus response of unified system
subplot(2, 1, 2)
if PM == 0
rlocus(numol, denol)
title('Root Locus Response of PI-Lead Compensator')

% Plot the closed-loop poles and zeros
hold on
pol = roots(dencl)
zer = roots(numcl)
plot(real(pol), imag(pol), '^', real(zer), imag(zer), 's')
hold off
% plot the Bode response of the unified system
else
bode(k*numol,denol)
title('Bode Response of PI-Lead Compensator')
end

- **PI-lead Compensator Design Script Files**

%pi-lead compensator for continuous time domain

%Open-loop transfer function
Gp=tf([1],[1 1 0]);
[num,den]=tfdata(Gp,'v');

%Design point in continuous time domain
sd=-1+j*1;
sd=-8+j*8;

% Sampling time period in continuous time domain
T=0;

% Phase margin in continuous time root locus
PM=0;

%choose the zero value for PI-lead compensator to the right side of pid
%zero
z1=0.25

% plot the step response and the root locus in continuous time domain
figure(1)

% call pilead function to see all responses of all variables in continuous
% time domain
[kc,numc,denc,numcl,dencl]=pilead(num,den,T,sd,PM,z1)

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

%delta
%pi-lead compensator for delta operator in gama domain

% Sampling time period in delta operator
T=0.01;

%Convert open-loop transfer function in cotinuous time domain to delta
%operator
[dnum,dden]=c2del(num,den,T);

%Design point in delta operator
dsd=(exp(sd*T)-1)/T

% plot the step response and the root locus in gama domain
figure(2)
% call pilead function to see all responses of all variables in gama domain
% domain
[d, dnum, ddenc, dnumcl, ddencl] = pilead(dnum, dden, T, dsd, PM, z1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% frequency
% pi-lead compensator for frequency response in continuous time domain

% sampling period in frequency response in continuous time domain
T = 0;
% sd = -6 + j * 0.1;
% real and imaginary part of design point
sigmad = -real(sd);
wd = imag(sd);

% Determine the phase margin and gain cross over frequency
theta = atan(-wd / sigmad);
zh = cos(theta);
wn = sigmad / zh;
PM = atan(2 * zh / sqrt(-2 * zh^2 + sqrt(1 + 4 * zh^4)))
wgc = 2 * zh * wn / tan(PM)

% Design point in frequency response of continuous time domain
sd = j * wgc;

% plot the step response and the Bode response in frequency continuous
time
figure(3)
% call pilead function to see all responses of all variables in frequency
time
[kf, numcf, dencf, numclf, ddenclf] = pilead(num, den, T, sd, PM, z1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% delta frequency

% Sampling time period in frequency delta operator
T = 0.01;

% Design point in frequency response of delta operator
dsd = (exp(j * wgc * T) - 1) / T

% plot the step response and the Bode response in frequency delta
operator
figure(4)
% call pilead function to see all responses of all variables in delta
time
[dkf, dnumcf, ddencf, dnumclf, ddenclf] = pilead(dnum, dden, T, dsd, PM, z1)
APPENDIX G

Matlab M-files for PI-lead Design,
with Non-located Zeros

- **PI-lead Compensator Design Function**

```matlab
function [k,numc,denc,numcl,dencl]=pilead2(num,den,T,sd,PM,z1,z2)

% PI-Lead (proportional-Integral-lead) compensator design in unified
% notation for continuous or discrete time systems using root locus
% or Bode procedures, where two zero are not equal.
% Continuous time systems are represented in the s-domain and
% discrete time systems are represented in the delta domain.

[k,numc,denc,numcl,dencl]=pilead2(num,den,T,sd,PM,z1,z2) calculates
the control gain K, the PI-lead compensator Gc(s), and the closed-
loop
response such that:

1. sd, is the design point at the transient response specification
   in root locus design and in Bode design.

   unified design point
   ---------------------------
   | T=0   | T~=0 |
   ---------------------------
   root locus | s0   | (exp(sd*T)-1)/T |
   Bode       | jwgc | (exp(j*wgc*T)-1)/T |

2. K= 1/(abs(Gc(sd))*abs(Gp(sd)))
where,
   sd, is the design point
   Gc(sd), is the pi-lead compensator for the unified system
evaluated
   at the design point
   Gp(sd), is the open-loop plant system for the unified system
evaluated at the design point

3. the compensator transfer function for the unified system is
   Gc(gama)= tf(numc,denc)

4. the closed-loop transfer function for the unified system is
   Gcl(gama)=tf(numcl,dencl)

5. the open-loop transfer function for the unified system is
   Gp(gama)=tf(num,den)

6. T, is the sampling time for discrete-time systems and 0
   for continuous-time systems

7. PM, is the phase margin specification for Bode designs and 0
   for root locus designs
```
% 8. z1, one of the polead zero at the location of plant pole
% 9. z2, another of the polead zero to the right side of pid zero

% Authors: T. Emami and J. M. Watkins
% Date: 02-14-06

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%open-loop transfer function
[num,den]=tfchk(num,den);

% Evaluation of open-loop transfer function at the design point in
% unified transform theory
X=(polyval(num,sd,T))/(polyval(den,sd,T));

% The desired angle in the angle constraint
phi=-pi+PM;

% The desired compensator angle
thc=phi-angle(X)

% real part of design point
sigmad=-real(sd);

% imaginary part of design point
wd=imag(sd);

% theta z1 is the angle of one of compensator zero
thz1=atan(wd/(z1-sigmad));

% Determine the location of pd compensator zero
z=sigmad+wd/tan(thc);

% theta z2 is the angle of second compensator zero
thz2=atan(wd/(z2-sigmad));

% the angle of compensator pole
thp1=atan(-wd/sigmad);

% the angle of compensator pole
thp2=thz1+thz2-thc-thp1;

% the pole value of compensator
p2=sigmad+wd/tan(thp2)
a=z1+z2;
b=z1*z2;

% the PI-lead compensator transfer function
numc=[1 a b];
denc=[1 p2 0];
% Evaluation of pi-lead2 compensator of unified system at the design point
Y = (polyval(numc, sd, T))/(polyval(denc, sd, T));

% The control gain
k = 1/(abs(X) * abs(Y));

% open-loop transfer function of unified system after compensator design
numol = conv(numc, num);
denol = conv(denc, den);
[numol, denol] = tfchk(numol, denol);

% closed-loop transfer function of unified system after compensator design
numcl = k * numol;
dencl = denol + numcl;

if T == 0
    % Open-loop transfer function of unified system
    Gop = tf(numol, denol);
else
    % Convert open-loop transfer function of unified system to z domain
    [numoz, denoz] = del2z(numol, denol, T);
    % the open-loop transfer function of unified system
    Gop = tf(numoz, denoz, T);
end

% the closed-loop transfer function of unified system
Gcl = minreal(k * Gop / (1 + k * Gop));

% plot the step response of unified system
subplot(2, 1, 1)
step(numcl, dencl)
title('Step Response of PI-Lead Compensator')

% plot the root locus response of unified system
subplot(2, 1, 2)
if PM == 0
    rlocus(numol, denol)
title('Root Locus Response of PI-Lead Compensator')
end

    % Plot the closed-loop poles and zeros
hold on
pol = roots(dencl)
zer = roots(numcl)
plot(real(pol), imag(pol), '^', real(zer), imag(zer), 's')
hold off

% plot the Bode response of the unified system
else
    bode(k*numol,denol)
    title('Bode Response of PI-Lead Compensator')
end

- **PI-lead Compensator Design Script Files**

% Open-loop transfer function
Gp=tf([1],[1 1 0]);
[num,den]=tfdata(Gp,'v');

% Design point in continuous time domain
sd=-1+j*1;
sd=-8+j*5

% Sampling time period in continuous time domain
T=0;

% Phase margin in continuous time root locus
PM=0;

% Choose one of the pid zero value to cancel one of the open-loop pole
z1=1;

% Choose another zero to right of pid compensator
z2=0.155

% Plot the step response and the root locus in continuous time domain
figure(1)

% Call pilead2 function to see all responses of all variables in continuous
time domain
[kc,numc,denc,numcl,denc1]=pilead2(num,den,T,sd,PM,z1,z2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% delta
% PI-lead compensator for delta operator in gama domain

    % Sampling time period in delta operator
    T=0.01;

    % Choose one of the pi-lead zero value to cancel one of the open-loop pole
    z1=0.9950;
%Convert open-loop transfer function in continuous time domain to
delta
%operator
[dnum,dden]=c2del(num,den,T);

%Design point in delta operator
dsd=(exp(sd*T)-1)/T

% plot the step response and the root locus in gama domain
figure(2)
% call pi-lead function to see all responses of all variables in gama
domain
%   domain
[dk,dnumc,ddenc,dnumcl,ddencl]=pilead2(dnum,dden,T,dsd,PM,z1,z2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%frequency
% pi-lead compensator for frequency response in continuous time domain
%sampling period in frequency response in continuous time domain
T=0;
%choose one of the pi-lead zero value to cancel one of the open-loop
pole
z1=1;

%real and imaginary part of design point
sigmad=-real(sd);
wd=imag(sd);

%Determine the phase margin and gain crossover frequency
 teta=atan(-wd/sigmad);
zeta=cos(teta);
wn=sigmad/zeta;
PM=atan(2*zeta/sqrt(-2*zeta^2+sqrt(1+4*zeta^4)))
wgc=2*zeta*wn/tan(PM)

%Design point in frequency response of continuous time domain
sd=j*wgc;

% plot the step response and the Bode response in frequency continuous
time
% domain
figure(3)
% call pilead function to see all responses of all variables in
frequency
% continuous time domain
[kf,numcf,dencf,numclf,denclf]=pilead2(num,den,T,sd,PM,z1,z2)
% delta frequency

% Sampling time period in frequency delta operator
T=0.01;

% choose one of the pid zero value to cancel one of the open-loop pole
z1=0.9950;

% Design point in frequency response of delta operator
ds=(exp(j*wgc*T)-1)/T

% plot the step response and the Bode response in frequency delta operator
figure(4)

% call pilead function to see all responses of all variables in delta frequency domain
[dkf,dnumcf,ddencf,dnumclf,ddencclf]=pilead2(dnum,dden,T,ds,PM,z1,z2)