

Summer 2011

PID Tuning of Plants With Time Delay Using Root Locus

Greg Baker
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Baker, Greg, "PID Tuning of Plants With Time Delay Using Root Locus" (2011). *Master's Theses*. 4036.
DOI: <https://doi.org/10.31979/etd.hqru-y8br>
https://scholarworks.sjsu.edu/etd_theses/4036

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

PID-TUNING OF PLANTS WITH TIME DELAY USING ROOT LOCUS

A Thesis

Presented to

The Faculty of the Department of General Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Greg Baker

August 2011

© 2011

Greg Baker

ALL RIGHTS RESERVED

The Designated Thesis Committee Approves the Thesis Titled

PID-TUNING OF PLANTS WITH TIME DELAY USING ROOT LOCUS

by

Greg Baker

APPROVED FOR THE DEPARTMENT OF GENERAL ENGINEERING

SAN JOSÉ STATE UNIVERSITY

August 2011

Dr. Peter Reischl	Department of Electrical Engineering
Dr. Ping Hsu	Department of Electrical Engineering
Dr. Julio Garcia	Department of Aviation and Technology

ABSTRACT

PID-TUNING OF PLANTS WITH TIME DELAY USING ROOT LOCUS

by Greg Baker

This thesis research uses closed-loop pole analysis to study the dynamic behavior of proportional-integral-derivative (PID) controlled feedback systems with time delay. A conventional tool for drawing root loci, the MATLAB function `rlocus()` cannot draw root loci for systems with time delay, and so another numerical method was devised to examine the appearance and behavior of root loci in systems with time delay.

Approximating the transfer function of time delay can lead to a mismatch between a predicted and actual response. Such a mismatch is avoided with the numerical method developed here. The method looks at the angle and magnitude conditions of the closed-loop characteristic equation to identify the true positions of closed-loop poles, their associated compensation gains, and the gain that makes a time-delayed system become marginally stable. Predictions for system response made with the numerical method are verified with a mathematical analysis and cross-checked against known results.

This research generates tuning coefficients for proportional-integral (PI) control of a first-order plant with time delay and PID control of a second-order plant with time delay. The research has applications to industrial processes, such as temperature-control loops.

ACKNOWLEDGEMENTS

This thesis was produced with the help of several individuals. First, the author is indebted to the inspirational instruction of Dr. Peter Reischl, and the willingness of Mr. Owen Hensinger to share his experience in process control, the source of the overshoot-reduction technique suggested in the text. Sincere appreciation is expressed for the guidance and counsel of Dr. Ping Hsu and Dr. Julio Garcia. Lifelong thanks are due for the patience and support of my wife Adrienne Harrell.

TABLE OF CONTENTS

1.0 Introduction	1
The Problems With Time Delay	3
PID Compensation	10
Review of PID-Tuning Approaches for Systems With Time Delay	13
PID-Tuning Approach for Systems With Time Delay Used in This Study	14
2.0 Validation of Numerical Algorithm	17
Proportional Compensation of First- and Second-Order Plants Without Time Delay	17
3.0 Results	26
Proportional Compensation of a First-Order Plant With Time Delay	26
Proportional-Integral (PI) Compensation of a First-Order Plant Without Time Delay	33
Proportional-Integral (PI) Tuning Strategy With and Without Time Delay	34
Proportional-Integral (PI) Compensation of a First-Order Plant With Time Delay	39
Proportional-Integral (PI) Coefficients for a First-Order Plant With Time Delay	44
Proportional-Integral-Derivative (PID) Compensation of a Second-Order Plant Without Time Delay	46
Proportional-Integral-Derivative (PID) Compensation of a Second-Order Plant With Time Delay	48
Proportional-Integral-Derivative (PID) Coefficients for a Second-Order Plant With Time Delay	49

4.0 Conclusion	56
References	59
Appendices	62
Appendix A: Laplace Transform	62
Appendix B: Inverse Laplace Transform and Residue Theorem	63
Appendix C: Tools	65
Appendix D: Identifying the Poles and Zeros of a PID Compensator	70
Appendix E: Numerical Computation of Root Locus With and Without Time Delay	73
Appendix F: Root Loci for Simple Plants Drawn Using Approximated Time Delay	80
Appendix G: Root Loci for Simple Plants Drawn Using True Time Delay	84
Appendix H: Compensation Gain K_p Yielding 5% Overshoot of Set Point for a First-Order Plant With Time Delay	105
Appendix I: Ziegler-Nichols PID Tuning	108
Appendix J: Determination of Break-Away and Reentry Points of Loci in Systems With Time Delay	111

LIST OF FIGURES

Figure		Page
1	Open-Loop Response of a First-Order Plant Without Time Delay	4
2	Closed-Loop Response of a First-Order Plant Without Time Delay	4
3	Open-Loop Response of a Pure Time-Delay Plant	5
4	Closed-Loop Response of a Pure Time-Delay Plant	6
5	Comparison of Approximated Time Delay and True Time-Delay Root Loci	8
6	Typical Feedback Loop With Time Delay	10
7	Constituents of PID Block	11
8	Pole-Zero Maps showing PI and PID Compensators in the S -Plane	16
9	Block Diagram Showing Proportional Compensation of a First-Order Plant Without Time Delay	17
10	Root Loci of First- and Second-Order Plants Without Time Delay Drawn by MATLAB	21
11	Root Loci of First- and Second-Order Plants Without Time Delay Drawn by the Numerical Algorithm	22
12	Block Diagram Showing Proportional Compensation of a First-Order Plant With Time Delay	26
13	Root Loci of a First-Order Plant With Time Delay	27
14	Block Diagram Showing Feedback Loop With Set-Point and Load-Disturbance Inputs	29
15	Root Loci, With Three Highlighted Test Points, for a First-Order Plant With Time Delay	30

16	Simulation of System Output for Highlighted Test Points in Loci	32
17	Pole-Zero Map of a First-Order Plant Without Time Delay and PI Compensator	36
18	Simulations of System Output for Three Test Points on the Root Loci	38
19	Root Loci for Three Test Points of the PI Compensator Zero	40
20	Simulations of System Output for Three PI-Zero Test Locations	41
21	Simulations of System Output for Three PI-Zero Test Locations with Overshoot Reduction Method Applied	43
22	<i>S</i> -Plane Map of a PID Compensator and Second-Order Plant Without Time Delay, and Resulting Root Loci	47
23	Root Loci of a PID-Compensated Second-Order Plant With Time Delay	50
24	Performance of Recommended PID-Tuning Coefficients	54
C1	Impulse Responses for Pole Locations in the <i>S</i> -Plane	67
F1	Comparison of Padé Time-Delay Approximations	82
G1	Root Loci of a Pure Time-Delay Plant	87
G2	Root Loci of a First-Order Plant With Time Delay Showing Marginal Gain	90
G3	Root Loci of a Single-Zero Plant, a Differentiator, With Time Delay	97
G4	Root Loci of a Double-Zero Plant With Time Delay	98
G5	Root Loci of a Single-Pole, First-Order, Plant With Time Delay	99
G6	Root Loci of a Single-Pole, First-Order, Plant With Time	102

Delay (Close-up of Break-Away Point)

G7	Root Loci of a Double-Pole, Second-Order, Plant With Time Delay	103
G8	Root Loci of a PID-Compensated First-Order Plant With Time Delay	104
H1	Compensation Gain Giving 5% Overshoot for First-Order Plant With Time Delay	107
I1	"Process Reaction Curve"	109

LIST OF TABLES

Table		Page
1	Recommended PI-Tuning Coefficients for a First-Order Plant With Time Delay	45
2	Comparison of the PID Double Zero Position That is the Basis for Tuning Recommendations to the Left-Most Position Where Loci Reenter the Real Axis	52
3	Recommended PID-Tuning Coefficients for a Second-Order Plant With Time Delay	53
H1	Compensation Gain Yielding 5% Overshoot for a First-Order Plant With Time Delay	106

1.0 Introduction

In this study, a numerical method is developed for examining the paths of closed-loop poles, *root loci*, in feedback systems with time delay. The paths of closed-loop poles are rarely tracked in these systems because of a mathematical difficulty posed by delay. The numerical method developed here not only averts this mathematical obstacle but enables recommendations to be made for PID-tuning coefficients in systems with time delay, which is also known as latency, transport delay, and dead time. “For a pure dead-time process, whatever happens at the input is repeated at the output θ time units later” (Deshpande & Ash, 1983, p. 10).

In this research a novel application of root locus analysis is developed for producing tuning coefficients for proportional-integral (PI) and proportional-integral-derivative (PID) compensators that give optimal transient response for first-order and second-order plants with time delay. “The time response of a control system consists of two parts, the transient response and the steady-state response. By transient, we mean that which goes from the initial state to the final state” (Ogata, 2002, p. 220). In this study, optimal transient response means rapid, and roughly equivalent, settling times after unit-step inputs in set-point change or load disturbance.

Frequency response analysis assesses closed-loop system stability through open-loop Bode and Nyquist plots (Stefani, Shahian, Savant, & Hostetter, 2002, p. 461), which convey the relationship of a plant's output at steady-state, in terms of magnitude and phase, to a sinusoidal input. Closed-loop pole analysis, on the other hand, characterizes

the nature of closed-loop dynamics by tracking closed-loop pole locations as a system parameter, typically compensation gain, is varied. The decay rate and oscillation frequency of each component of transient response, not easily predictable from time-delay Bode or Nyquist plots, are known once the location of the associated pole, or pair of poles, in the s -plane is identified. The relationship between a pole's location in the s -plane and associated transient response is depicted in Figure C1.

The transfer function of time delay $G_{delay}(s)$ is an exponential function of the complex variable s , and delay θ (Ogata, 2002, p. 379).

$$G_{delay}(s) = e^{-s\theta} \quad (1)$$

The traditional symbol for angle θ is used because time delay introduces a phase angle difference between input and output sinusoids.

The exponential transfer function, as we will see in Appendix G, leads to a system with time delay having a characteristic equation that is transcendental, meaning it can only be expressed by a function with an infinite number of terms. Since the conventional tool for drawing root loci, the MATLAB function `rlocus()`, cannot accommodate time-delay systems, another numerical technique is developed here (Appendix E).

Root loci for systems with time delay can be constructed with several methods, for example, graphically (Ogata, 2002, p. 379), by approximating time delay (Vajta,

2000; Ogata, 2002), or by solving sets of simultaneous non-linear equations (Appendix E). Time-delay approximations, however, can lead to significant differences between predicted and actual responses. Such mismatches, as well as possible instabilities (see Figure 5; Silva, Datta, & Battacharyya, 2001), are avoided by drawing time-delay root loci with the straightforward and robust numerical technique developed in this study. Its predictions of closed-loop dynamic behavior and recommendations for PID coefficients are cross-checked against known results, MATLAB and SIMULINK simulations (Chapter 2), and mathematical derivations (Appendix G).

The Problems With Time Delay

Though a few systems actually benefit from the addition of time delay (Sipahi, Niculescu, Abdallah, & Michiels, 2011), time delay poses two difficulties to the control of simple plants of interest in this study: 1) it is inherently destabilizing, and 2) it is difficult to accommodate mathematically.

Time delay's destabilizing influence is illustrated by comparing the output of a feedback system where the plant is pure time delay to a feedback system where the plant is first order. The open-loop response of a first-order plant is shown in Figure 1, where the plant time constant is 10 s. After a unit-step input, it settles to within 2% of final value after four time constants (40 s.)

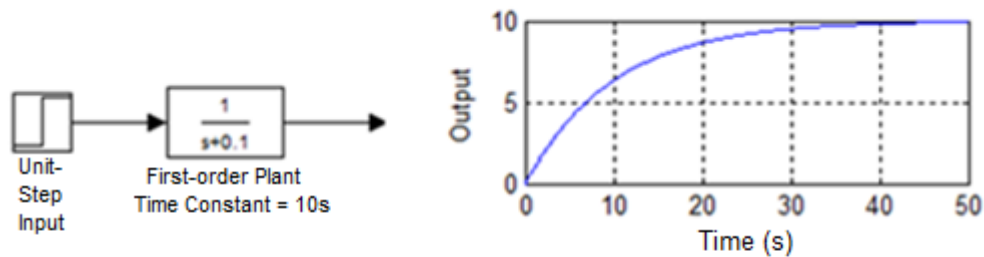


Figure 1. Open-Loop Response of First-Order Plant Without Time Delay

SIMULINK simulation of open-loop output: a unit-step input is applied to a first-order plant with a time constant of 10 s.
Note plant output settles to within 2% of final value in four time constants.

The first-order plant can be accelerated with feedback. With a compensation gain of one, as shown in Figure 2, its output settles within 2% of final steady-state value in only 3.55 s. In Chapter 2 (Figures 10 and 11), it will be shown that the first-order plant's output is always accelerated by increasing compensation gain.

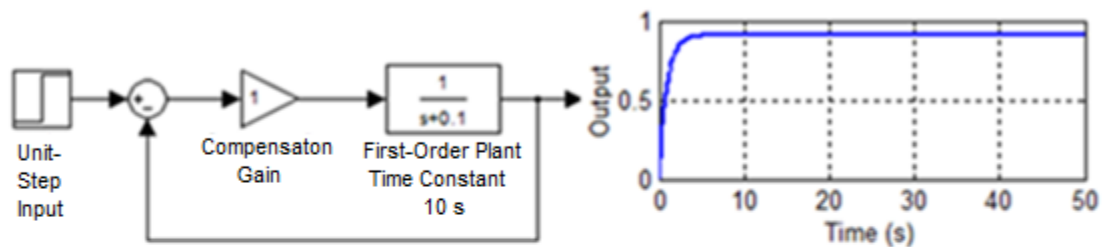


Figure 2. Closed-Loop Response of a First-Order Plant Without Time Delay

SIMULINK simulation of closed-loop output: the first-order plant is placed in a feedback loop under proportional compensation.
Note the first-order plant is accelerated by feedback. Its output settles to within 2% of final value in one tenth the time required of the open-loop plant in Figure 1.

The open-loop response of a pure time-delay plant, on the other hand, is shown in Figure 3, where time delay is one second.

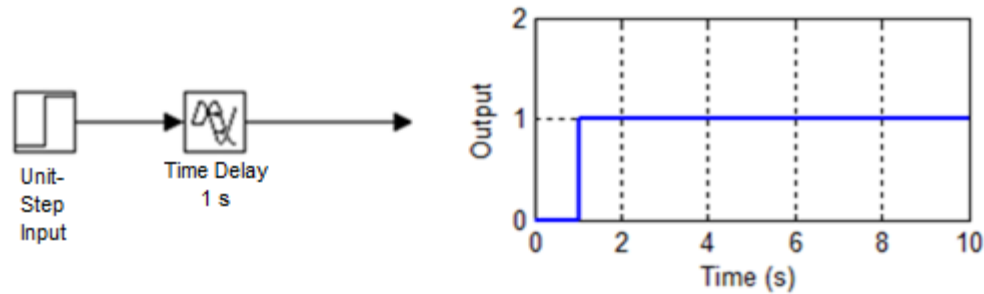


Figure 3. Open-Loop Response of a Pure Time-Delay Plant

SIMULINK simulation of open-loop output: a unit-step input is applied to a plant that is purely a time delay of one second. Note the plant output reaches final value in exactly one second.

When a feedback loop is comprised of a pure time-delay plant compensated at the same gain used for the first-order-plant in Figure 2, the system oscillates and never reaches steady state, as shown in Figure 4. Root loci drawn by the numerical algorithm (shown in Appendix G, Figure G1) are consistent with this time-series response. They show reducing compensation gain below one stabilizes the system, though it will remain oscillatory, and increasing compensation gain above one destabilizes the system. Loci cross the imaginary axis at a gain of one and at vertical positions $i\omega = \pm\pi$. These positions correspond with angular frequencies $\omega = \pm\pi$ radians/sec, which exactly match the oscillating output, shown in Figure 4. Angular frequency ω , measured in radians/sec, is 2π times frequency f in time, which is measured in cycles/sec or Hz. Therefore, since $\omega = 2\pi f$, for an angular frequency of $\omega = \pi$ the associated

frequency in time is $f = \frac{\omega \text{ radians/sec}}{2\pi \text{ radians/cycle}} = \frac{\pi}{2\pi} \text{ cycles/sec} = \frac{1}{2} \text{ Hz}$. A frequency of 1/2 Hz

corresponds with a period of 2 s, and clearly the period of oscillation in Figure 4 is 2 seconds.

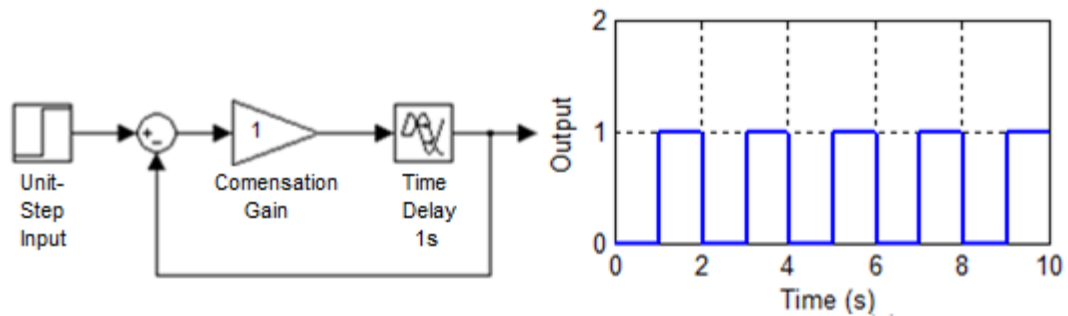


Figure 4. Closed-Loop Response of a Pure Time-Delay Plant

SIMULINK simulation of closed-loop output: the one s. pure time-delay plant is placed in a feedback loop with the same compensation gain as used for the first-order plant in Figure 2.

Note: the time-delay plant is marginally stable, continuously oscillates, and never settles at final value.

The four simulations above show how time delay can have a destabilizing influence by isolating a pure first-order plant, and then a pure time delay plant, in a feedback loop. We saw the simple first-order plant is accelerated by feedback, but the pure time-delay plant oscillates and can become unstable.

The difficulty that time delay poses to the mathematical analysis of feedback systems comes from its exponential transfer function $e^{-s\theta}$, which leads to transcendental characteristic equations (explained in detail in Appendix G). Thus, $G_{delay}(s)$ is usually approximated by a rational polynomial.

A comparison of root loci drawn by the numerical method in Figure 5 demonstrates the variations in predictions of system response that can be expected when using time-delay approximations. First-order Taylor and second-order Padé (see Appendix F; Ogata, 2002, p. 383) time-delay approximations are used to produce these root loci, which depict the closed-loop dynamics of compensated first-order and second-order plants with time delay.

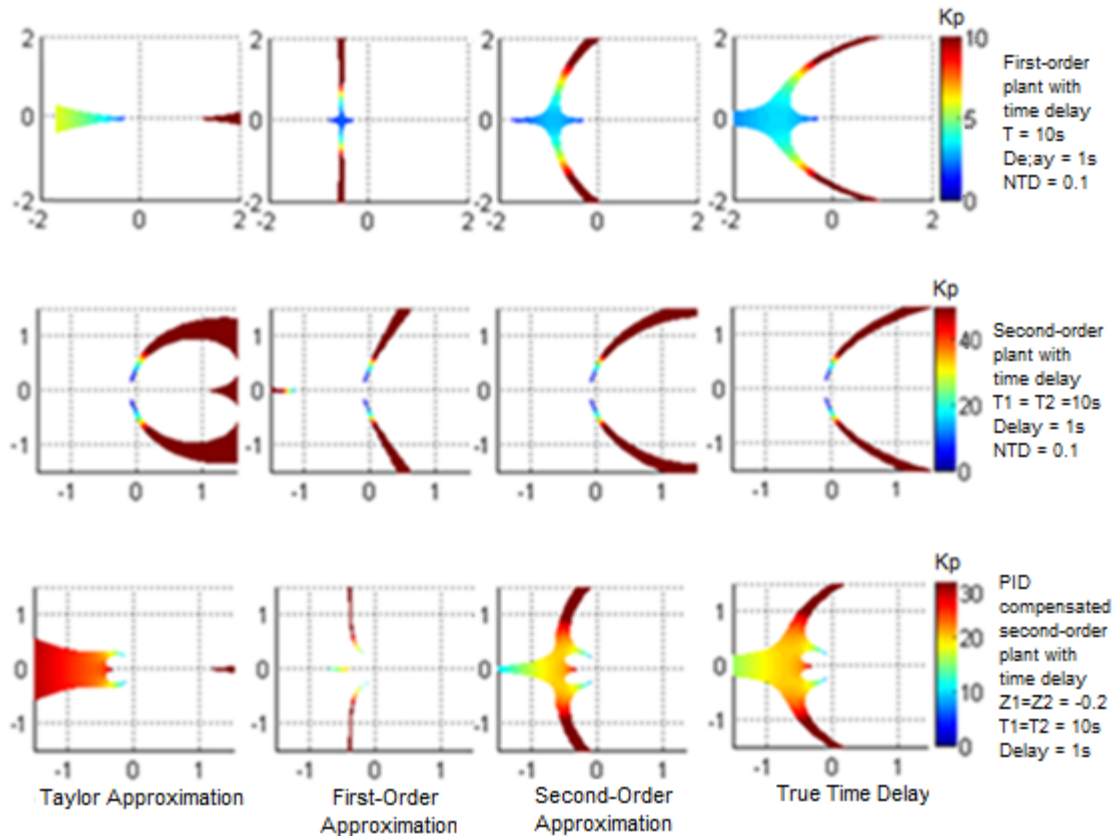


Figure 5. Comparison of Approximated Time Delay and True Time-Delay Root Loci

Comparison of root loci drawn using three types of time-delay approximations, Taylor, first-order, and second-order Padé, to true time-delay root loci for three separate feedback systems: 1) proportionally-compensated first-order plant, 2) proportionally-compensated second-order plant, and 3) a PID-compensated second-order plant.

Note the variation between the predictions of closed-loop dynamics by the root loci based on time-delay approximations and true time-delay root loci. For example, the Taylor approximation predicts the first-order plant is always unstable because closed-loop poles are in the right half-plane for all gains. The Padé approximation is a closer match of true behavior for the first-order plant but overestimates marginal gain to be 21, the true time-delay loci cross the imaginary axis at a gain of 16.

The compensation gain that puts closed-loop poles on the imaginary axis, marginal gain, places the feedback system in an oscillating state. Predictions of marginal gain by the three styles of time-delay approximations shown in Figure 5 clearly vary. The second-order Padé approximation is probably the most accurate, but still gives

optimistic predictions of marginal gain for the first-order plant and the PID-compensated second-order plant.

A side effect of the numerical method is that loci appear wider than they actually are in some regions, and they become invisible in other regions. Loci path widths at a given location are easily thinned or broadened, however, by adjusting a parameter in the numerical algorithm, the decision criterion, as explained in Chapter 2 and Appendix E.

Another way to draw time-delay root loci would be to seek roots of the closed-loop characteristic equation by solving simultaneous non-linear equations. Real and imaginary parts of the characteristic equation would be the simultaneous equations of interest (Appendix E). The MATLAB function `fsolve()`, which requires an initial guess at the solution(s), could be used to find simultaneous solutions. Each call to `fsolve()` would return a value of s that satisfies the closed-loop characteristic equation, and which would be a point on the loci. To completely define all branches comprising the loci throughout a region of interest in the s -plane, `fsolve()` must be called reiteratively with a variety of initial guesses at the solution to cover the region, and a variety of compensation gains to show response as a function of gain; some gaps might still be visible in the loci. The approach used in this paper offers a simpler implementation while identifying all points on the loci within a region of interest. The weakness of the numerical technique is that the widths of the loci can vary.

PID Compensation

The PID compensator is a true workhorse of feedback control. “The majority of control systems in the world are operated by PID controllers” (Silva, Datta, & Battacharyya, 2002, p. 241). A typical application, control of a first-order plant with time delay, is shown in Figure 6.

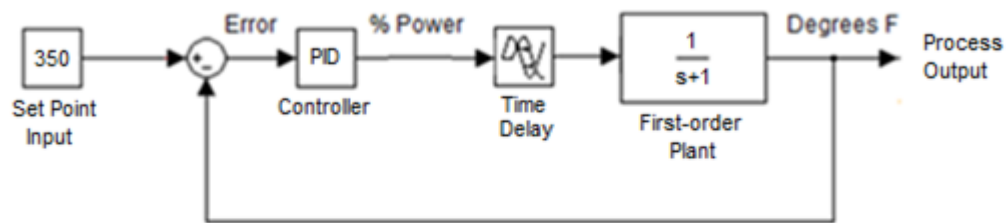


Figure 6. Typical Feedback Loop With Time Delay

Block diagram of a typical feedback loop with a reference (set point) input. A first-order plant is shown for illustration purposes only, the plant may be second order, or higher, in nature.
 Note: the plant transfer function is stated as a function of the complex variable s . In the s -domain transfer functions of neighboring blocks are multiplied together for the combined transfer function.

The simple proportional compensator, used in Figures 1 through 4, will mostly result in a static or steady-state error for plants of interest in this study, such as temperature-control loops (Astrom & Haggund, 1995, p. 64; Ogata, 2002, p. 281). To eliminate steady-state error and accommodate higher-order plants, more complex PI and PID compensators must be used. As described in detail in Chapter 3, the integral (I) in PID eliminates steady-state error, and the derivative (D) improves transient response for high-ordered plants.

The PID compensator's transfer function is a summation of three terms; proportional, integral, and derivative, as shown in Figure 7.

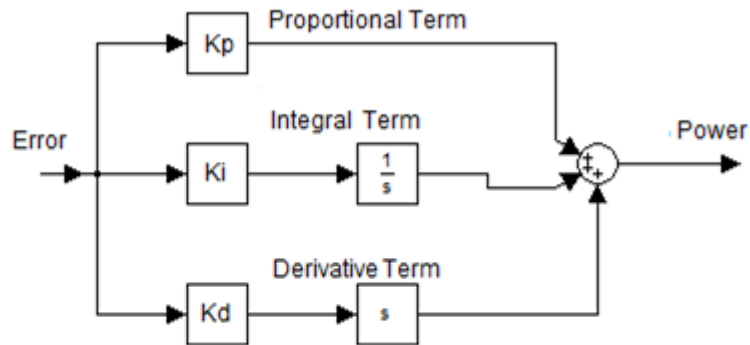


Figure 7. Constituents of PID Block

The output (power) of the PID block in Figure 6 is the sum of proportional, integral, and derivative control terms or actions. Note: setting the coefficients of the three terms (K_p , K_i , and K_d) is equivalent to tuning the controller.

The proportional term produces control action equal to the product of process error, and *compensation gain* k_p .

$$k_p \quad (\text{Proportional Term})$$

The integral term produces control action equal to the continuous summation of process error times, an *integral gain* k_i . Thus integral action can be expressed as a function of the complex variable s .

$$\frac{k_i}{s} \quad (\text{Integral Term})$$

The derivative term produces control action equal to the rate of change of the process times, a *derivative coefficient* k_d . As will be seen in Chapter 3, the derivative stops ringing from occurring in a system composed of a proportionally-compensated second-order plant. The derivative term can be expressed as a function of s .

$$k_d s \quad (\text{Derivative Term})$$

The complete PID transfer function $G_{PID}(s)$ is the sum of all three terms (Silva et al., 2002).

$$G_{PID}(s) = k_p + \frac{k_i}{s} + k_d s, \quad (PID)$$

(2)

The pole and two zeros of a PID compensator are easily identified by rearranging the three terms of $G_{PID}(s)$ over a common denominator.

$$G_{PID}(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_d s^2 + k_p s + k_i}{s} \quad (3)$$

We see $G_{PID}(s)$ has a single *pole* that lies at the origin, where $s = 0$. The roots of its quadratic numerator

$$k_d s^2 + k_p s + k_i = 0 \quad (4)$$

are the two zeros of $G_{PID}(s)$.

When $k_d = 0$ in $G_{PID}(s)$ there is no derivative action, and the compensator has proportional and integral terms only. The transfer function of the PI compensator

$$G_{PI}(s) = \frac{k_p s + k_i}{s} \quad (5)$$

has a pole at the origin, like the PID compensator, and a single zero.

Setting $k_i = k_d = 0$ results in a proportional-only compensator, its transfer function is just *gain* k_p , and it has no poles or zeros.

$$G_{P-only}(s) = k_p \quad (6)$$

Review of PID-Tuning Approaches for Systems With Time Delay

“The process of selecting controller parameters to meet given performance specifications is known as controller tuning” (Ogata, 2002, p. 682). A variety of theoretical approaches have been used to produce PID-tuning formulas for a first-order plant with time delay.

A heuristic time-domain analysis (Hang, Astrom, & Ho, 1991) used set-point weighting to improve Ziegler and Nichols' (1942) original PID-tuning formulas, which were also determined empirically. “Repeated optimizations using a third-order Padé

approximation of time delay produced tuning formulas for discrete values of normalized dead time" (Zhuang & Atherton, 1993, p.217). Barnes, Wang, and Cluett (1993) used open-loop frequency response to design PID controllers by finding the least-squares fit between the desired Nyquist curve and the actual curve. In reviews of the performance and robustness of both PI- and PID-tuning formulas, tuning algorithms optimized for set-point change response were found to have a gain margin of around 6 dB, and those that optimized for load disturbance had margins of around 3.5 dB (Ho, Hang, & Zhou, 1995; Ho, Gan, Tay, & Ang, 1996).

PID-tuning formulas were derived by identifying closed-loop pole positions on the imaginary axis, yielding the system's ultimate gain and period. Dynamics are said to suffer, however, for processes where time delay dominates "due to the existence of many closed-loop poles near the imaginary axis, where the effect of zero addition by the derivative term is insignificant to change the response characteristics" (Mann, Hu, and Gosine, 2001, p.255). In a general review of time-delay systems, Richard (2003) discusses finite dimensional models and robust H₂ and H- ∞ methods.

PID-Tuning Approach for Systems With Time Delay Used in This Study

One outcome of this research is to recommend PI- and PID-tuning coefficients based on closed-loop pole analysis. PI coefficients for first-order plants and PID coefficients for second-order plants are given for six values of normalized time delay (NTD), the ratio of time delay to plant time constant. A two-step approach is used to generate the tuning coefficients for each plant type and NTD.

The first step is to select the most desirable locations for compensator zeros to lie. The rationale for selection of location is discussed in Chapter 3, in the sections on PI and PID-tuning of plants with time delay, where the effect of compensator zero locations on the form of loci is investigated, with three test points for the zeros. The location selection is also discussed in Appendix J, where an alternative mathematical method is used to show how zero locations affect break-away and reentry points. The ultimate goal of the zero-location selection criteria is to achieve the greatest net movement of closed-loop poles to the left.

The second step is to draw the root loci and select the most desirable locations for the dominant closed-loop poles to lie on the loci. These locations determine compensation gain k_p .

The strategy of placing the compensator zero and choosing the point on the loci where closed-loop poles move farthest left seeks the fastest possible closed-loop transient response (see the depiction of the relationship of pole position to the nature of transient response time Figure C1).

Under feedback, open-loop zeros attract closed-loop poles, so compensator zeros will be placed as far to the left as possible. In Chapter 3, it will be shown, using time delay limits, how far to the left a compensator zero can be placed before the plant's closed-loop poles no longer move toward it. Due to time delay, closed-loop poles get to the compensator zeros first. If compensator zeros are too far to the left, plant and

integrator poles move into the right half of the s -plane, which destabilizes the system, rather than moving to the left half of the s -plane, stabilizing and accelerating it.

PI and PID compensators can be represented by mapping their poles and zeros in the s -plane as shown in Figure 8. For more details on mathematically identifying compensator poles and zeros, see Appendix D.

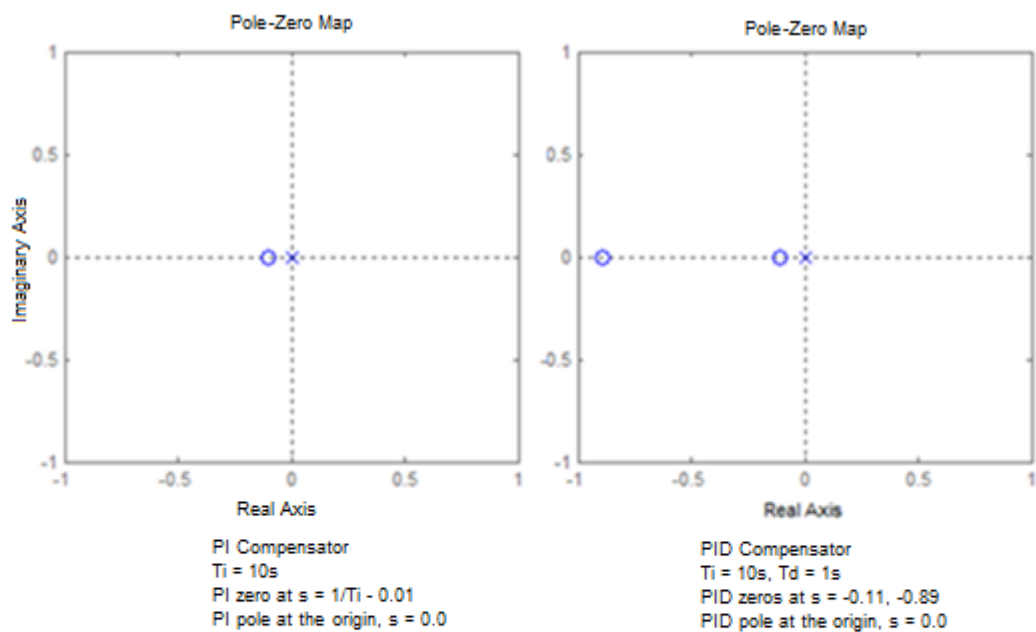


Figure 8. Pole-Zero Maps Showing PI and PID Compensators in the S-Plane

S-plane pole-zero plots of typical PI and PID compensators produced by MATLAB.

PI controllers have one pole and one zero, PID controllers have one pole and two zeros.

Note the same value of T_i is used in both PI and PID controllers above. The PI zero is at $s = -1/T_i$, PID zeros are near $s = -1/T_i$ and $-1/T_d$ since, as described in Appendix D, the ratio of T_i to T_d is relatively large.

2.0 Validation of Numerical Algorithm

In this chapter, the ability of the numerical method developed here to draw root loci for systems with time delay, is tested by comparing its output to known results for two simple plants without time delay.

Proportional Compensation of First- and Second-Order Plants Without Time Delay

A block diagram of a proportionally-compensated first-order plant without time delay is shown in Figure 9.

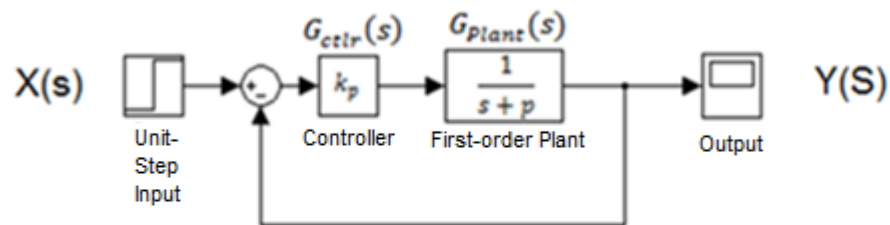


Figure 9. Block Diagram Showing Proportional Compensation of a First-Order Plant Without Time Delay

Note: the closed-loop transfer function, $Y(s)/X(s) = K_p / (s + p + K_p)$, is rational.

Positions of this system's poles, closed-loop poles, are expressed as a function of k_p below. Closed-loop pole paths are then plotted using MATLAB and the numerical method. Root-loci diagrams produced by the numerical method must match those drawn by MATLAB.

The open-loop transfer function $G_{ol}(s)$ of the first-order system without time delay in Figure 9 is:

$$G_{ol}(s) = G_{Ctrl}(s)G_{Plant}(s) = \frac{k_p}{s+p} \quad (7)$$

Distefano, Stuberrud, and Williams (1995, p. 156) describe the canonical form of a system's closed-loop transfer function $G_{cl}(s)$ as:

$$G_{cl}(s) = \frac{Y(s)}{X(s)} = \frac{G_{Ctrl}(s)G_{Plant}(s)}{1 + G_{Ctrl}(s)G_{Plant}(s)} \quad (8)$$

Thus, for the system in Figure 9, the closed-loop transfer function is

$$G_{cl}(s) = \frac{k_p}{s + p + k_p}$$

The closed-loop characteristic equation of the system in Figure 9 is

$$1 + G_{Ctrl}(s)G_{Plant}(s) = 0 \quad (9)$$

Values of s that satisfy the closed-loop characteristic equation are its roots, they make the denominator of the closed-loop transfer function equal to zero so they are closed-loop poles.

The characteristic equation of the system in Figure 9 is

$$s + p + k_p = 0 \quad (10)$$

If the plant is time invariant, the open-loop pole p is constant, so the position of the closed-loop pole is simple to express as a function of k_p .

$$s = -k_p - p \quad (11)$$

If the plant in Figure 9 is second-order, for example, having poles at $s = -0.1$ and $s = -1.0$, the open-loop transfer function is:

$$G_{ol}(s) = G_{Ctrl}(s)G_{Plant}(s) = \frac{k_p}{(s + p_1)(s + p_2)} = \frac{k_p}{(s + 0.1)(s + 1.0)} \quad (12)$$

The closed-loop transfer function is:

$$G_{cl}(s) = \frac{G_{Ctrl}(s)G_{Plant}(s)}{1 + G_{Ctrl}(s)G_{Plant}(s)} = \frac{k_p}{s^2 + (p_1 + p_1)s + p_1p_2 + k_p} \quad (13)$$

Closed-loop pole locations for this second-order system are the roots of its characteristic equation:

$$s^2 + (p_1 + p_1)s + (p_1p_2 + k_p) = 0 \quad (14)$$

Roots of this second-order equation, as well as higher-order equations, can be found with the MATLAB function `roots()`, which takes polynomial coefficients as input and returns roots

```
>>roots([1 p1+p2 p1*p2+Kp])
```

The MATLAB function `rlocus()` takes the uncompensated open-loop system transfer function as input, as described by polynomial coefficients, plots the positions of closed-loop poles, root loci, as k_p varies from zero to infinity. It does this by iteratively varying k_p , and at each value finding the roots of the closed-loop characteristic equation, then plotting them.

Root loci for the closed-loop system shown in Figure 9, which contains a first-order plant with a pole at $s = -0.1$, would be drawn by issuing the MATLAB command:

```
>> rlocus([1], [1 0.1])
```

If the closed-loop system contains a second-order plant, for example with poles at $s = -0.1$ and $s = -1.0$, root loci would be drawn with the MATLAB command:

```
>> rlocus([1], [1 1.1 0.1])
```

These two calls to `rlocus()` generated the two root-loci plots in Figure 10.

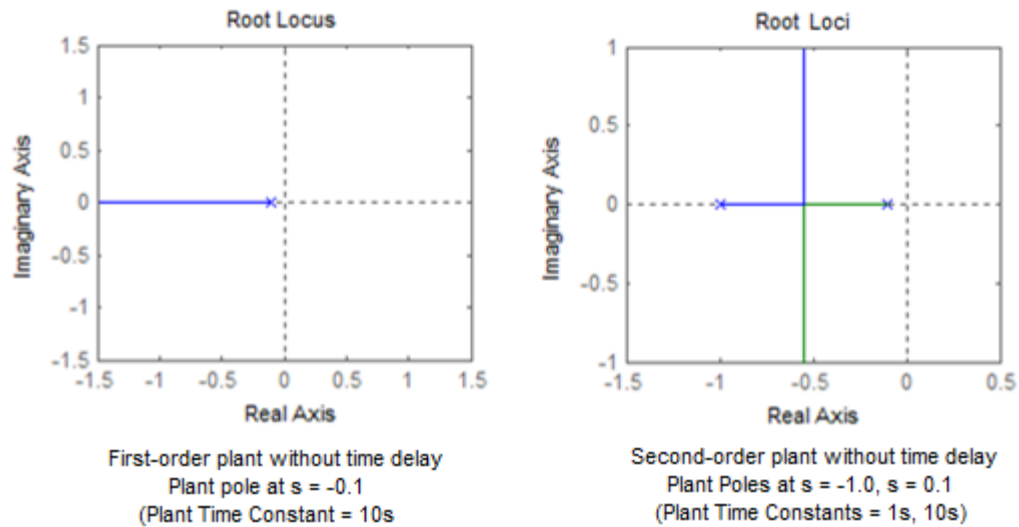


Figure 10. Root Loci of First- and Second-Order Plants Without Time Delay Drawn by MATLAB

Root loci, drawn by the MATLAB function `roloc()`, for a first-order and second-order plant depict the movement of closed-loop poles as compensator gain K_p is varied. Open-loop poles are marked by an X.

Note: first-order plant response accelerates and remains stable as gain increases. Second-order plant response accelerates until closed-loop poles collide with each other, and then break away from the real axis, their imaginary components increase with gain which means the system rings at higher and higher frequencies as gain increases.

To validate the numerical algorithm, which will be used mostly for systems with time delay, the root loci that it draws for systems without time delay will be compared to the root loci drawn by MATLAB in Figure 10.

Loci drawn by the numerical algorithm are shown in Figure 11. Note they differ from the MATLAB plots in that they are shaded and the widths of loci vary. The match between the depictions of closed-loop pole trajectories in Figures 10 and 11, however, is close enough to validate the tool. Further details are discussed in Appendix E.

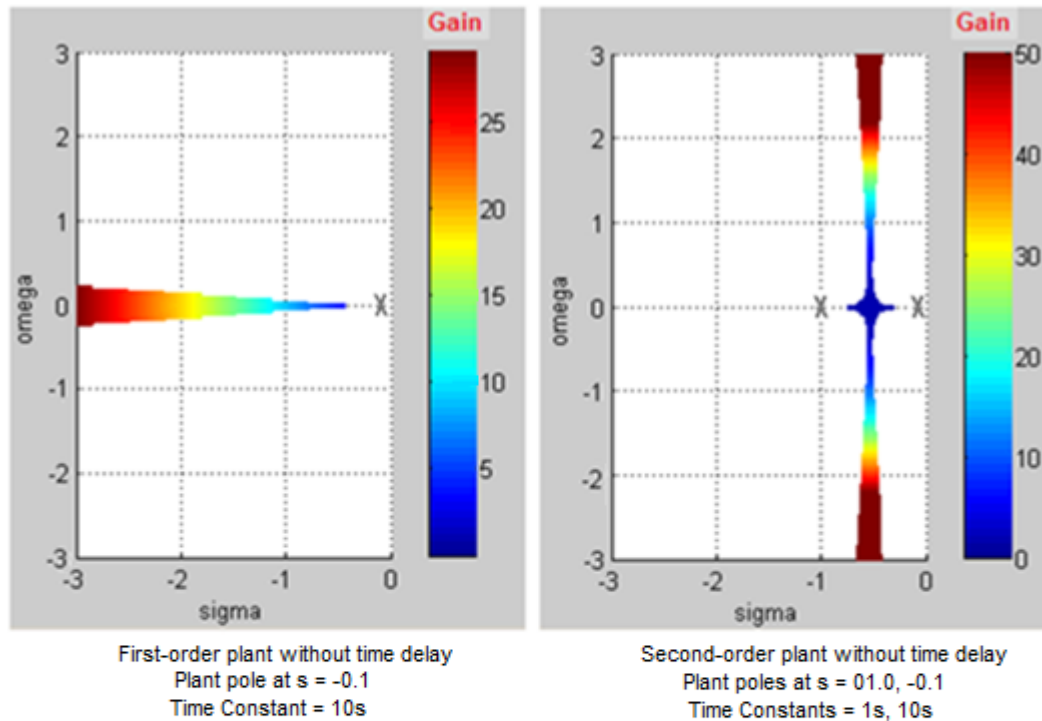


Figure 11. Root Loci of First- and Second-Order Plants Without Time Delay Drawn by the Numerical Algorithm

Movement of closed-loop poles as compensation gain K_p (value indicated by color-coding) varies from 0 to 50. Open-loop poles are marked by an X. Note: these figures serve to validate the numerical method because they match the root loci drawn by MATLAB in Figure 10, showing a first-order plant is accelerated by increasing gain; but a second-order plant can only be accelerated to the point the two closed-loop poles collide, at which point it starts to ring.

Systems with time delay have transcendental closed-loop characteristic equations, so the rules of root locus construction need to be modified (Ogata, 2002, p. 380). The MATLAB function `rlocus()` cannot draw root loci for such systems because their characteristic equations are transcendental (described in Appendix E). Another option for drawing time-delay root loci would be to use the MATLAB function `fsolve()` which finds roots of non-linear equations. In addition to a description of the non-linear equation it

requires an initial guess at the solution as input. Each point in a dense grid of points covering a region of interest in the s -plane would need to be input, and for a wide variety of gains. This method could still leave gaps in the loci.

The method used in this paper, however, is favored for its simplicity and robustness. It analyzes the angle and magnitude conditions of the closed-loop characteristic equation, and is equally effective whether the equation is of polynomial form or transcendental.

The first step is to evaluate the angle condition of the characteristic equation. For the system in Figure 9, the characteristic equation is

$$1 + k_p G_{Plant}(s) = 0$$

This equation is a function of the complex variable s so each side of the equation has a magnitude and phase angle. After moving the one to the right-hand side the phase angle component of each side is expressed

$$\text{Angle}[k_p G_{Plant}(s)] = \text{Angle}[-1] = 180^\circ(1 \pm 2l) \quad l = 0, 1, 2, \dots \quad (15)$$

Closed-loop pole positions are identified by computing $\text{Angle}[G_{Ctrl}(s)G_{Plant}(s)]$ at each point on a grid that spans a region of interest in the s -plane. Locations where the angle condition is satisfied to within a specific criterion $\text{Angle}[G_{Plant}(s)] = \pm 180^\circ \pm \text{Decision Criterion}$ are marked as being on the loci, though their proximity to the actual

loci depends on the decision criterion. Locations may be right on or just very close to the root locus.

Compensation gains at the pole locations are computed from the magnitude condition which comes from taking the magnitude of each side of the characteristic equation, for the system in Figure 9 this gives

$$|k_p G_{Plant}(s)| = |-1| = 1$$

Compensation gain at each pole location is then

$$k_p = \left| \frac{1}{G_{Plant}(s)} \right|$$

and is conveyed through color coding in the plots.

For this study, the decision criterion remains constant throughout any given plot, but varies from plot to plot as appropriate, to keep loci as thin as possible.

The reason loci widths vary within a given plot is because the rate of change of $Angle[G_{Ctrl}(s)G_{Plant}(s)]$ is a function of s , yet the decision criterion remains constant. As a result, some points that are not actual roots look like they are roots because they get color coded.

Figure 10 (drawn by MATLAB) and Figure 11 (drawn by the numerical method developed in this paper) are essentially equivalent depictions of closed-loop system transient response, and so they serve as partial validation of the numerical method. Both depictions show the first-order plant's return to steady state after a transient input is

accelerated with feedback, simply by increasing compensation gain k_p . As k_p increases from zero to infinity, the single closed-loop pole in the system follows a perfectly straight path from the open-loop pole position $s = -0.1$ to its final destination $s = -\infty$, (Figure C1 depicts the relationship between a pole's position, and its resulting impulse response, with an s -plane map of impulse response versus pole location throughout a region surrounding the origin).

The second-order plant's return to steady state after a transient input, on the other hand, is accelerated to a certain point by increasing compensation gain k_p , but then the system starts to ring if k_p increases past that point, as shown in Figures 10 and 11. The plot of the second-order plant in both figures shows two open-loop poles that lie on the real axis. As k_p increases, they approach each other and collide; after colliding the poles depart the real axis in opposite directions. Up to the point when both poles collide, increasing gain accelerates the system. Beyond that point, increasing gain will not accelerate the system, and merely leads to ringing at ever higher frequencies.

3.0 Results

In this chapter, the numerical technique will draw root loci for systems with time delay, and then produce PI- and PID-tuning recommendations for first- and second-order plants with time delay. The method of drawing root loci will be demonstrated on feedback systems without time delay, and then time delay will be brought into the loop.

PI-tuning coefficients will be stated for a first-order plant with time delay, and PID-tuning coefficients will be stated for a second-order plant with time delay, for six values of normalized time delay (NTD), the ratio of time delay to plant time constant.

Proportional Compensation of a First-Order Plant With Time Delay

Next, the numerical method draws root loci for the time-delayed system in Figure 12, a proportionally-compensated first-order plant with time delay.

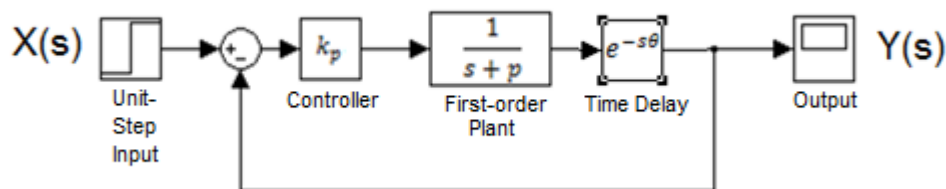


Figure 12. Block Diagram Showing Proportional Compensation of a First-Order Plant With Time Delay

Plant pole is at $s = -p$ (plant time constant = $1/p$), time delay = θ ,
 normalized time delay NTD = $(p)(\theta)$.
 Note: the closed-loop transfer function,
 $Y(s) / X(s) = (K_p \exp(-s \theta)) / (s + p + K_p \exp(-s \theta))$, is transcendental.

Root loci of this system are depicted at two magnification levels in Figure 13.

Note the two highlighted locations on the loci, they are complex conjugates and

correspond with a compensation gain $k_p = 4.8$. These pole locations are 45° from the real axis and, in a purely second-order system, would correlate with a damping coefficient $\zeta = 0.7$, meaning, during recovery from a transient input, the overshoot of the final value is expected to be 5% (Ogata, 1970, p. 238).

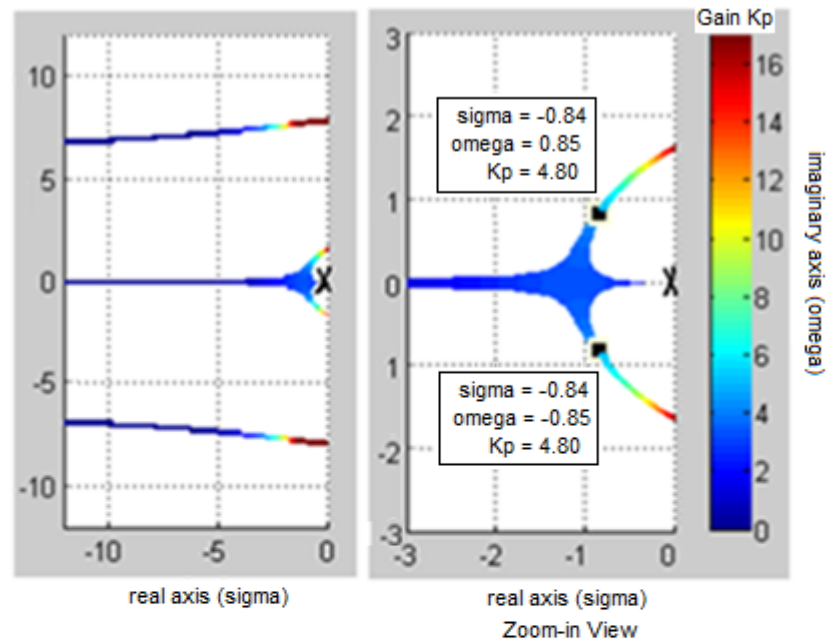


Figure 13. Root Loci of a First-Order Plant With Time Delay

Root loci, drawn by the numerical method developed in this paper, for a first-order plant with a time constant of ten seconds, and one second of time delay. $NTD = 1 / 10 = 0.1$. Movement of closed-loop poles as compensation gain K_p increases from zero to 17, as indicated by color-coding. Open-loop plant pole marked with an X.

Note: two loci, due to the presence of time delay, run roughly parallel to the real axis and approach horizontal asymptotes separated vertically by a distance of $(2\pi / \text{time delay})$. These true time-delay root loci are what the time-delay approximation root loci (Figure 5) try to match. It is also interesting to note the difference between the root loci above and root loci for the same plant without time delay (Figure 11).

By comparing Figure 13 to Figures 10 and 11, we see the difference between the closed-loop dynamics of a first-order plant with time delay and a first-order plant without

time delay. The loci in Figure 13 are consistent with the assertion, proven in Appendix G, that time delay introduces an infinite set of loci to the system. The three loci trajectories shown in Figure 13 are members of that infinite set. Two closed-loop poles due to time delay define loci that run from left to right, roughly parallel but slightly away from the real axis. A third pole due to delay forms a locus with the plant pole. The time-delay pole starts from $s = -\infty$ and travels to the right along the real axis as gain increases, it eventually collides with the plant pole, which moves left from its open-loop position. For this system, as shown in Appendix G (Equation G12), the compensation gain k_m associated with a closed-loop pole crossing the imaginary axis is nearly proportional to the pole's distance from the real axis ω :

$$k_m = \sqrt{1 + (\omega T)^2} \quad \text{G12}$$

A system is marginally stable when a closed-loop pole crosses the imaginary axis and no other poles are in the right-half side of the s -plane. Thus, according to the equation above, in a first-order system with time delay closed-loop poles that are closest to the real axis are dominant.

If the system is purely second order without time delay, applying the gain that places closed-loop poles at the positions highlighted in Figure 13 would result in a 5% overshoot of the final value after a transient input (Distefano et al., 1995, p. 98). However, even though the plant is really first order with time delay, we will see shortly SIMULINK simulations (Figure 16) show its behavior mimics a higher-order plant without time delay. Based on this observation, recommendations for compensation gain,

stated in Appendix H for a first-order plant with time delay, are produced by putting the dominant closed-loop poles at these locations.

All tuning recommendations put forth in this paper are evaluated by measuring the overshoot of final value produced, as well as the time required for the process to settle within 2% of final value, 2% settling time, after unit-step changes in set point and load disturbance. As shown in Figure 14, the load disturbance is introduced immediately downstream of the compensator.

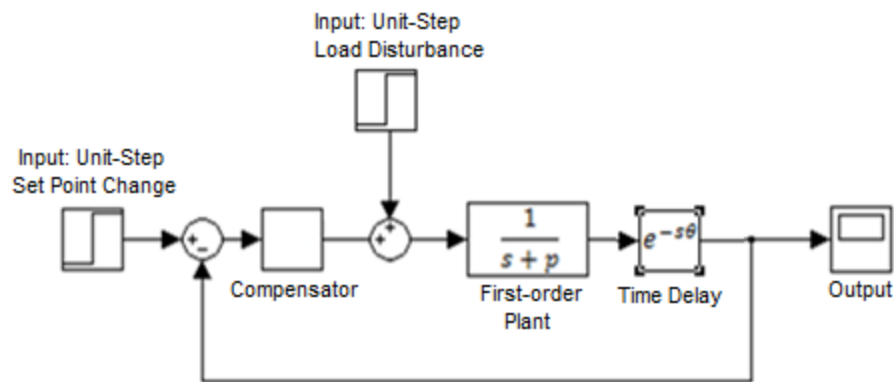


Figure 14. Block Diagram Showing Feedback Loop With Set-Point and Load-Disturbance Inputs

Note: We provide PID-tuning recommendations that seek to optimize 2% settling time after unit-step changes in set point or load disturbance. By optimize we mean rapid and roughly equivalent settling times after set point or load change with minimal overshoot.

Three separate compensation gains, associated with the three highlighted closed-loop pole positions in Figure 15, are used for simulating system output. The three highlighted points correspond with compensation gains of $k_p = 3.3, 4.9,$ and 11.9 . In a purely second-order system, poles at the angular positions, with respect to the origin,

shown in Figure 15, would correlate with damping coefficients of $\gamma = 1.0, 0.707,$ and nearly $0.0,$ respectively (Distefano et al., 1995, p. 98).

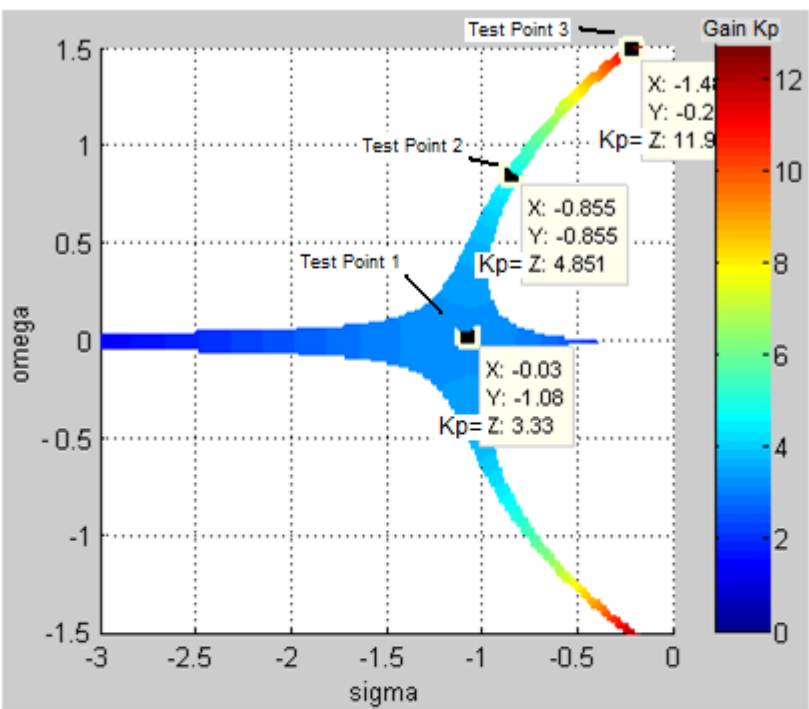


Figure 15. Root Loci, With Three Highlighted Test Points, for a First-Order Plant With Time Delay

Root loci drawn by the numeric algorithm: proportional compensation of a first-order plant with time delay. Plant time constant = 10 s (open-loop plant pole at $s = -0.1$), time delay = 1 s, NTD = 1 s / 10 s = 0.1. Compensation gain Kp is shown as Z. Open-loop plant pole is marked with X. Note: a closed-loop pole due to time-delay originates from the far left extreme of the real axis, then collides with the open-loop plant pole which moves left as gain increases. They both depart the real axis after colliding and move into the right half of the s-plane.

The three SIMULINK simulations in Figure 16 show system response after unit-step changes in set point and load disturbance for the three compensation gains associated with the highlighted pole locations in Figure 15. The time-series response in Figure 16 suggests the dynamic behavior of a first-order plant with time delay is similar to the dynamic behavior of a higher-order plant without time delay. At low gains there is no ringing, at medium gains there is some ringing, and at high gains there is plenty of ringing. Note, in this system, the final steady-state value is not guaranteed to match set point, however, system output happens to reach the desired value because, after the unit-step load disturbance, the input to the plant is exactly the desired output, so once the control output goes to zero the plant output will equal the desired value.

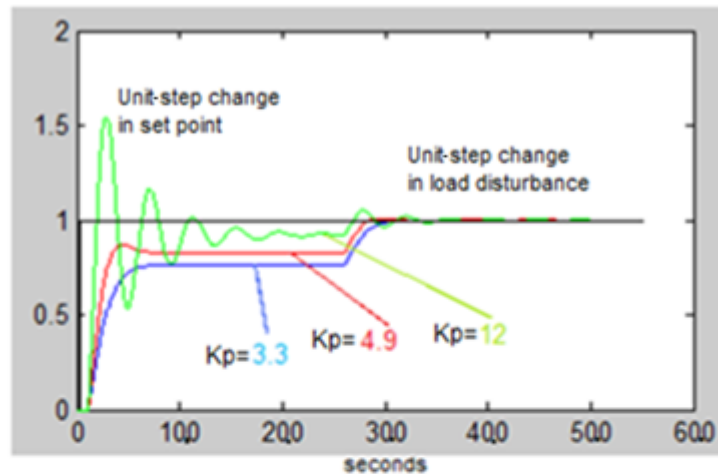


Figure 16. Simulation of System Output for Highlighted Test Points in Loci

Three simulations of closed-loop output, for the system containing a proportionally-compensated first-order plant with time delay ($NTD = 0.1$) depicted in Figures 14 and 15, after a unit-step change in set point and load disturbance. The three values used for compensation are associated with the three highlighted closed-loop pole positions in Figure 15.

Note steady-state error decreases with gain and, as predicted by root loci produced by the numerical method in Figure 15, the system starts to ring. Such dynamic behavior is similar to that of a second-order plant without time delay. Loci in Figure 15 are similar to loci for second-order plant without time delay in Figure 12 since two poles collide and then depart the real axis, except with time-delay loci the poles ultimately move into the right half-plane.

Test Point 1: $k_p = 3.3$. Closed-loop pole positions are on the real axis and have no imaginary component. As expected, the system's output is free of oscillations.

Test Point 2: $k_p = 4.9$. Closed-loop pole positions are 45° from the real axis and would correspond with 5% overshoot in a purely second-order system. Actual overshoot of final steady-state value is about 5%.

Test Point 3: $k_p = 12$. Closed-loop pole positions are close to the imaginary axis. The system is stable, but it is near marginal stability.

Recommendations for compensation gain k_p , for control of a first-order plant with time delay, are tabulated in Appendix H for seven values of normalized time delay (NTD) covering the range $0 < NTD \leq 0.5$. System performance, as measured by 2% settling time after a unit-step change in set point or load disturbance, resulting from the recommended gains is also tabulated. Overshoot, verified through SIMULINK simulations, are within 5%. Recommendations are based on root-loci diagrams, like the one shown in Figure 15, created for each value of NTD.

Proportional-Integral (PI) Compensation of a First-Order Plant Without Time Delay

In the previous example, where a first-order plant is proportionally compensated, steady-state error is apparent (see Figure 16). Steady-state error can be eliminated, however, if a factor of $\frac{1}{s}$, an *integrator*, exists in the open-loop transfer function (Ogata, 2002, p. 847).

When integral control action $\frac{k_i}{s}$ is added to a proportional compensator a proportional-integral (PI) compensator is created. Its transfer function $G_{PI}(s)$ is the sum of proportional and integral terms:

$$G_{PI}(s) = k_p + \frac{k_i}{s} = \frac{k_p s + k_i}{s} \quad (16)$$

The pole of $G_{PI}(s)$ lies at the origin $s = 0$, its zero lies at $s = -\frac{k_i}{k_p}$.

Compensation gain k_p is common to both proportional and integral terms when the integral term is expressed as $\frac{k_p}{sT_i}$, where T_i is integral time (Astrom & Hagglund, 1988, p. 4):

$$G_{PI}(s) = k_p + \frac{k_p}{sT_i} = \frac{k_p\left(s + \frac{1}{T_i}\right)}{s} \quad (17)$$

The zero of $G_{PI}(s)$ is then independent of k_p and lies at $s = -\frac{1}{T_i}$. This form of $G_{PI}(s)$ simplifies our analysis because the zero location is determined by a single parameter T_i , and values of proportional gain k_p are read directly from the root locus diagram and its associated closed-loop pole position immediately identified.

Proportional-Integral (PI) Tuning Strategy With and Without Time Delay

The strategy for tuning plants with time delay is now introduced and applied to plants both with and without time delay. Since a PI compensator's pole must lie at the origin of the s -plane, but its zero can be placed anywhere on the real axis at the discretion of the designer, the compensator is tuned by first placing its zero, then drawing root loci, and finally choosing compensation gain k_p so closed-loop poles are at the most desirable location. This sequence will now be described.

Placement of PI zero. In determining the best place to put the PI zero, we consider two rules restricting movement of closed-loop poles in the s -plane:

- Under feedback, as compensation gain increases from zero to infinity, “the root locus branches start from the open-loop poles and terminate at zeros” (Ogata, 2002, p. 352). Zeros remain fixed in place.
- “If the total number of real poles and real zeros to the right of a test point on the real axis is odd, then that point is on a locus” (Ogata, 2002, p. 352).

To meet the goal of accelerating the plant beyond its open-loop response, by pulling open-loop poles to the left, the PI zero is placed to the left of the plant pole, as shown in Figure 17. In this configuration the two portions of the real axis that will contain loci, according to the second rule stated above, lie between the two open-loop poles and to the left of the PI zero. It will be shown that only in systems without time delay can open-loop plant and integrator poles be pulled to the left of the PI zero, regardless of how far to the left the PID zero is placed. In such systems, transient response can always be accelerated simply by increasing compensation gain.

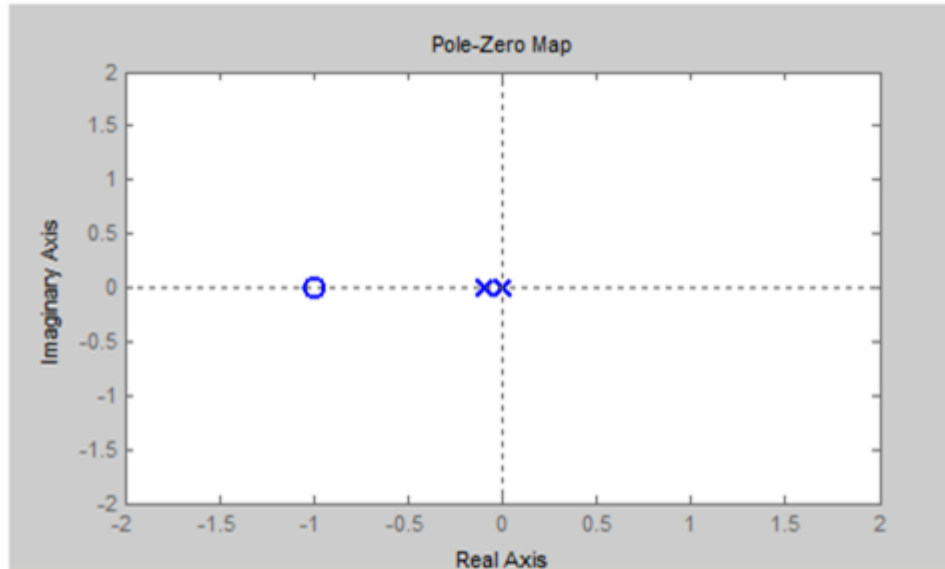


Figure 17. Pole-Zero Map of a First-Order Plant Without Time Delay and PI Compensator

First-order plant time constant = 10 seconds, open-loop plant pole at $s = -0.1$.
PI coefficient $T_i = 1$ second, PI zero at $s = -1.0$.
Note: a PI compensator contributes one pole and one zero to the system.

Drawing root loci. Root loci are next drawn by the numerical algorithm for the open-loop system without time delay depicted in Figure 17. Loci for this system are shown in Figure 18 where, as k_p increases from 0 to 70, the two open-loop poles approach each other on the real axis and collide. Both poles then depart the real axis and head left to reenter the real axis on the left side of the PI zero. The numerical algorithm's drawing agrees with well-known behavior and shows the general shape of loci that can be expected for this type of system, regardless of how far to the left the PI zero is placed.

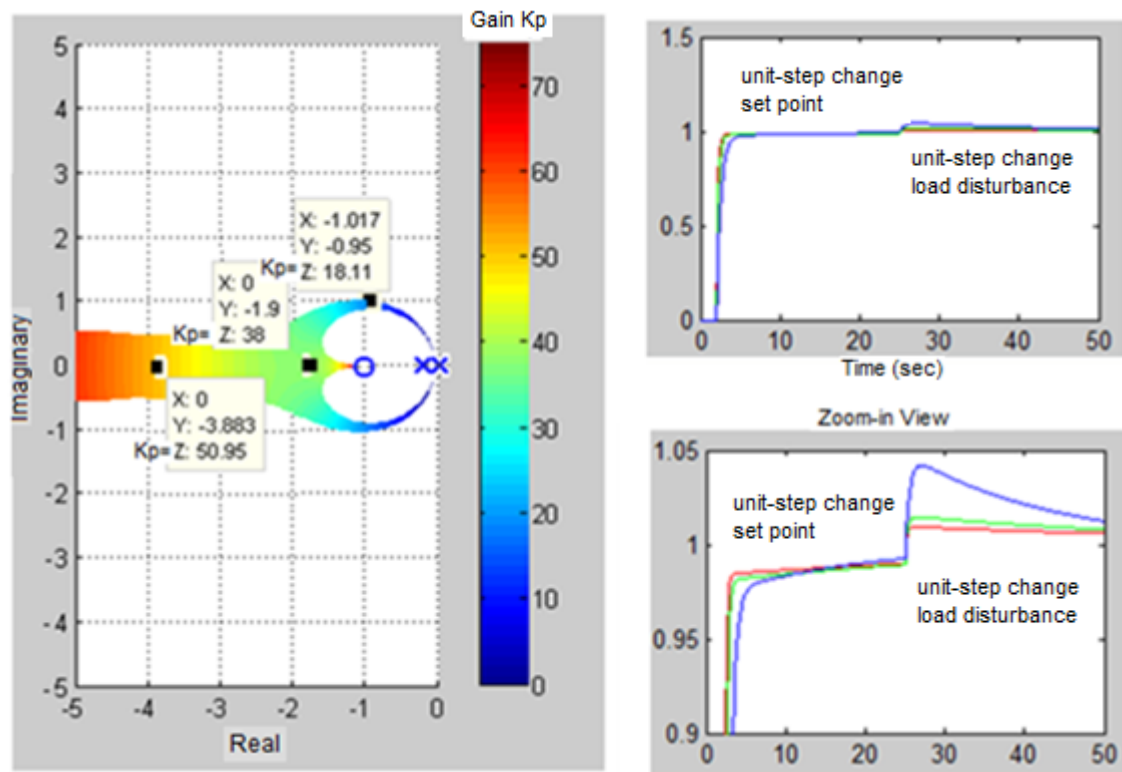


Figure 18. Simulations of System Output for Three Test Points on the Root Loci

Root loci drawn by the numerical algorithm depict closed-loop dynamics of a PI-compensated first-order plant without time delay. Plant time constant = 10 seconds (open-loop plant pole at $s = -0.1$), time delay = 1 second, NTD = 1 second / 10 seconds = 0.1. Compensation gain K_p is shown as Z. Note all highlighted closed-loop pole positions are at least 10 times as far from the origin as the open-loop plant pole.

SIMULINK simulations of system output after unit-step changes in set point and load disturbance. The three PI compensators producing system output are defined by the three highlighted closed-loop pole positions. Note the times required for the closed-loop process to settle within 2% of final value are less than one tenth the time of the open-loop plant.

Three SIMULINK simulations show system output for three different PI compensators, where $T_i = 1$ second. Three compensation gains, $k_p = 18, 38,$ and $51,$ complete the design of the three PI controllers, and are associated with the three closed-loop pole locations highlighted in Figure 18. The simulations show the closed-loop system returns rapidly to steady state after unit-step changes in set point or load

disturbance and two percent settling times are much shorter than in open-loop. Such performance is consistent with the fact that closed-loop poles are relatively far to the left of the open-loop plant pole.

Choosing compensation gain. Once loci are drawn, the compensation gain that results in the most desirable closed-loop pole locations, in terms of system performance, can be chosen. For example, to favor a heavily-damped response closed-loop poles should be close to or on the real axis. To favor less damping, which in some cases leads to faster response (such as in a purely second-order system), closed-loop poles should be off the real axis, but no more than 45° from the real axis.

Proportional-Integral (PI) Compensation of a First-Order Plant With Time Delay

When time delay is introduced to the feedback system previously discussed, a PI-compensated first-order plant, the compensator's zero can no longer be placed anywhere along the real axis and still pull open-loop plant and integrator poles over to its left side. Instead, if the PI zero is placed too far to the left of the origin, a closed-loop pole due to time delay gets to it first. Plant and integrator poles are forced to head into the right-half plane.

Root loci produced by the numerical tool are drawn at two different levels of scale in Figure 19, using three test points for the PI zero. The three test points are located relatively far to the left ($s = -0.5$), just barely to the left ($s = -0.235$), and to the right ($s = -0.1$) of the left-most part of the region that allows open-loop plant and integrator poles to reenter the real axis.

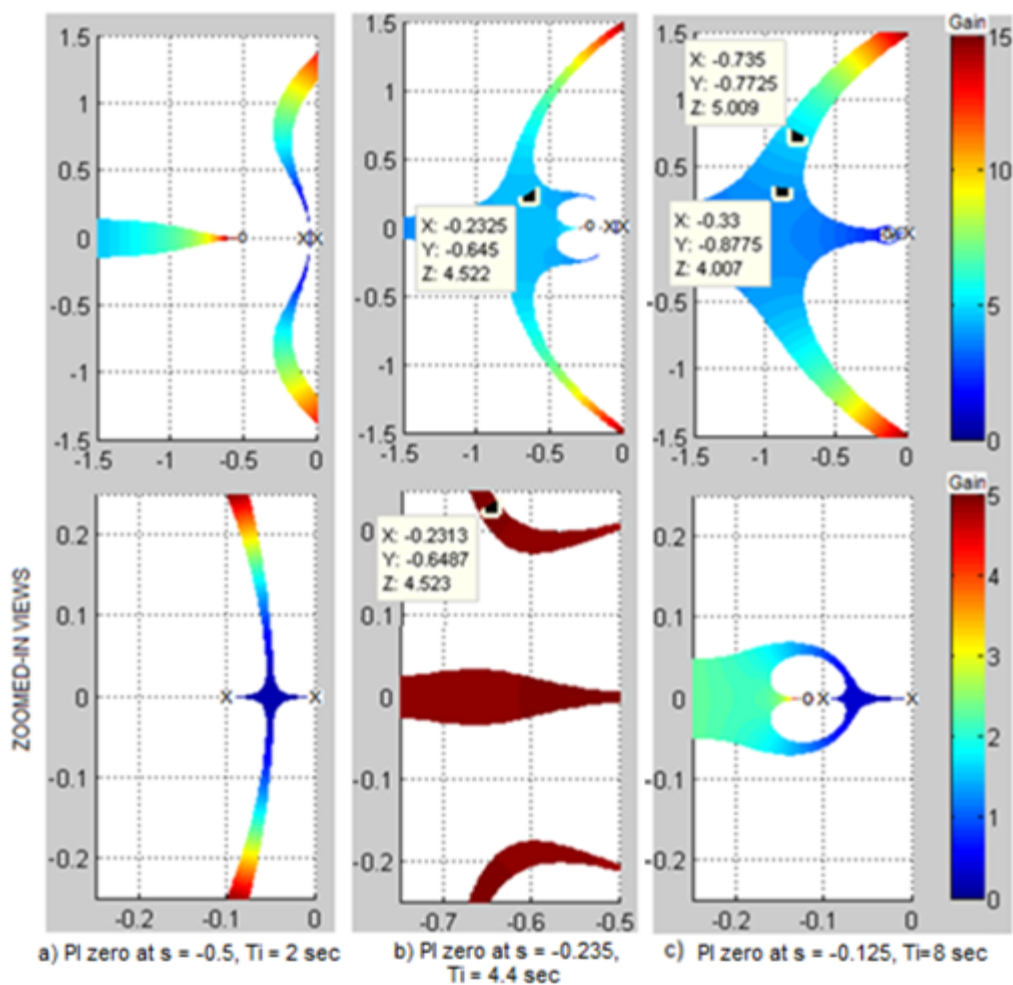


Figure 19. Root Loci for Three Test Points of the PI Compensator Zero

Root loci drawn by the numerical method depict PI compensation of a first-order plant with time delay for three test points of the PI zero. Plant time constant = 10 seconds (open-loop pole at $s = -0.1$), time delay = 1 second, NTD = $1 / 10 = 0.1$. Compensation gain is shown as Z in captions, open-loop poles are marked by X. Note in Figure b) the PI zero is slightly to the left of the region where loci would re-enter the real axis. Since the dominant closed-loop poles are farthest left when the zero is placed as shown in Figure b), this position is used to produce the tuning coefficient recommendations in Table 1.

The connection between the three highlighted closed-loop pole positions in Figure 19, which define three different PI compensators, and associated time responses of the system, is made in Figure 20. System output after unit-step changes in set point and load disturbance is simulated for each controller.

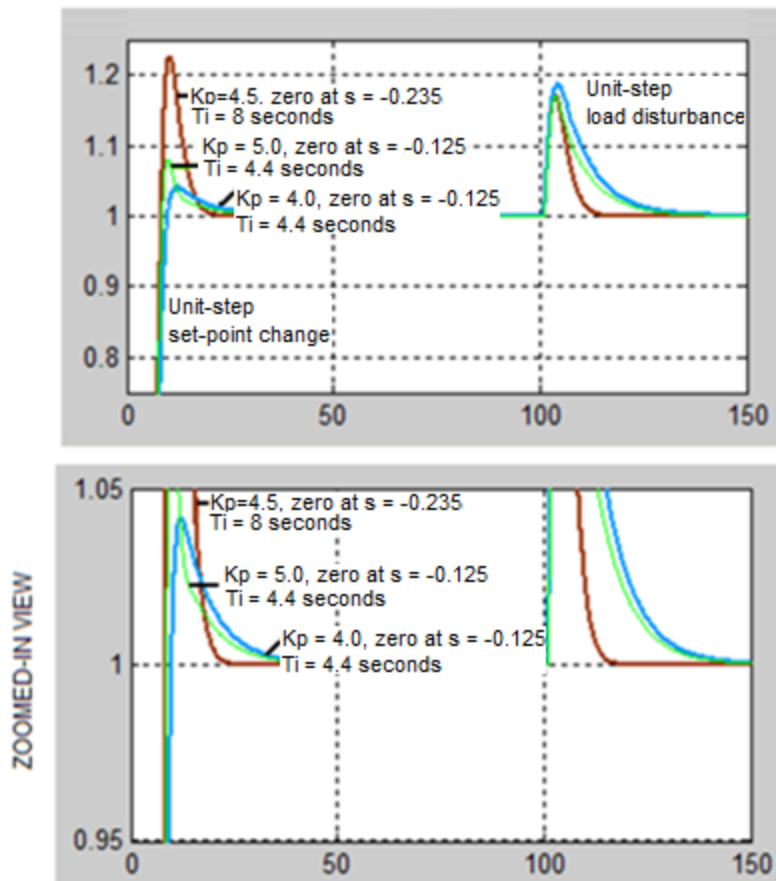


Figure 20. Simulations of System Output for Three PI-Zero Test Locations

Three SIMULINK simulations of system output, using the three PI-compensation gains associated with the closed-loop pole positions highlighted in Figure 19, after unit-step changes in set point and load disturbance. Plant time constant = 10 seconds, time delay = 1 second, NTD = 0.1.

Note, of the choices above, the combination of the zero location $s = -0.235$ and compensation gain value $K_p = 4.5$ that yields the shortest 2% settling time (4.3 seconds) after a load disturbance also results in the most overshoot of the final value after a set-point change.

This study found that closed-loop poles move farthest to the left when the PI compensator zero is placed slightly to the left of the location that allows loci to reenter the real axis (see Figure 19b). As measured by 2% settling time, this zero position gives the fastest recovery after a load disturbance (see Figure 20). During recovery to steady state after a set-point change, however, there is too much overshoot. Overshoot of set point can be eliminated, however, with a technique that leaves load-disturbance response unaltered, as shown in Figure 21 where system output is simulated for the same compensators used in Figure 20, but each compensator is modified to implement this overshoot reduction method.

The overshoot-reduction technique used here linearly decreases the natural rate of integration as the distance between set point and process value grows. Effective integration rate drops to zero when the process is separated from set point by one proportional band = $1/k_p$.

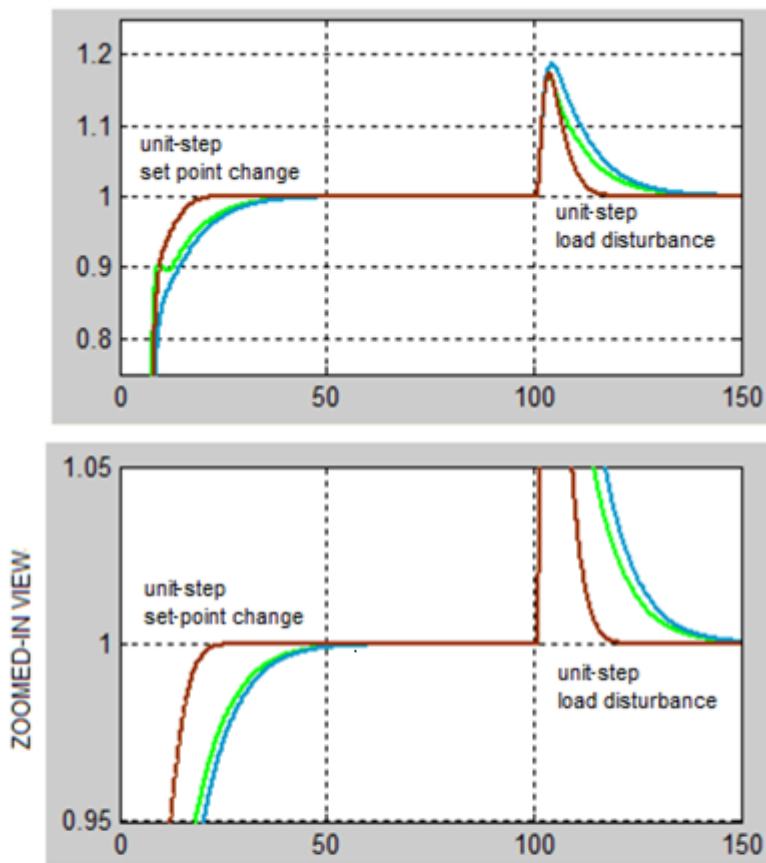


Figure 21. Simulations of System Output for Three PI-Zero Test Locations With Overshoot Reduction Method Applied

SIMULINK simulations of system output for the same closed-loop systems simulated and shown in Figure 20, however, here overshoot inhibition is applied.

Note: Comparison of these simulations to Figure 20 shows overshoot of set point is eliminated without affecting load disturbance response.

Proportional-Integral (PI) Coefficients for a First-Order Plant With Time Delay

Recommendations for PI-tuning parameter sets k_p and T_i , are given in Table 1 for six values of NTD; each set of coefficients is generated for a given NTD, from a root-locus plot similar in form to the one shown in Figure 19b. The design goal is to move closed-loop poles as far to the left of the PI zero as possible.

Table 1

Recommended PI-Tuning Coefficients for a First-Order Plant with Time Delay.

NTD	Recommendation	Result	Recommendation	Result	Result
	PI-Zero Position (Multiples of Open-Loop Plant Pole Position)	Equivalent Ti (% of Open-Loop Plant Time Constant)	k_p	2% Settling Time Unit- Step Change in Set Point (Multiples of Open- Loop Plant Time Constant)	2% Settling Time Unit- Step Change in Load Disturbance (Multiples of Open- Loop Plant Time Constant)
0.05	4.5	22.2	10.0	0.15	0.4
0.10	2.75	36.4	5.0	0.8	0.8
0.20	2.15	46.5	2.5	1.5	1.5
0.30	2.00	50.0	1.8	2.0	1.9
0.40	1.65	60.6	1.3	3.0	3.0
0.50	1.50	66.6	1.0	4.1	4.1

These recommendations meet the design goal of short settling times after unit-step changes in set point and load disturbance that are roughly equivalent when the PI compensator is modified to eliminate overshoot of set point as described in the text. Note shortest settling times occur with smallest normalized time delay, NTD.

Proportional-Integral-Derivative (PID) Compensation of a Second-Order Plant Without Time Delay

A closed-loop feedback system, comprising a second-order plant without time delay, will ring or oscillate at high gains, as previously discussed and depicted with root loci in Figures 10 and 11. Ringing can be eliminated, however, by adding derivative action to the compensator, which adds another zero to the system (see Chapter 1: PID Compensation and Appendix D).

Derivative action allows higher gains to be used on a second-order plant without time delay because it suppresses ringing at high gains, as illustrated by the root loci in Figure 22 where two closed-loop poles depart the real axis but reconnect with it to the left of the PID double zero. Loci will reconnect with the real axis to the left of the compensator double zero, regardless of how far to the left the double zero is placed.

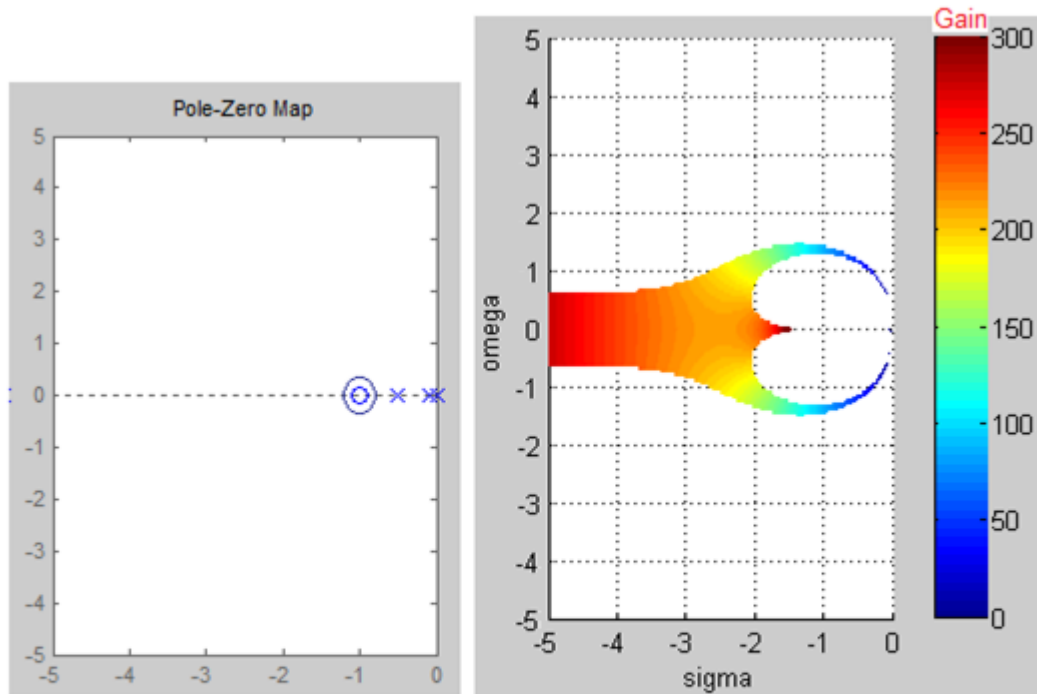


Figure 22. S-Plane Map of a PID Compensator and Second-Order Plant Without Time Delay, and Resulting Root Loci

Note: the PID double zero pulls plant and integrator poles left to re-enter the real axis to the left of the double zero.

PID tuning recommendations will put both PID zeros at the same location, making a double zero because this maximizes their ability to pull closed-loop poles to the left.

Proportional-Integral-Derivative (PID) Compensation of a Second-Order Plant With Time Delay

“When one or two time constants dominate (are much larger than the rest), as is common in many processes, all the smaller time constants work together to produce a lag that very much resembles pure dead time” (Deshpande & Ash, 1981, p. 13). Such a process can be described by a three-parameter double-pole second-order plant model and time delay (Astrom & Hagglund, 1995, p. 19):

$$G(s) = \frac{K}{(1+sT)^2} e^{-s\theta} \quad (18)$$

Recommendations for PID-tuning coefficients will be based on this plant model.

Proportional-Integral-Derivative (PID) Coefficients for a Second-Order Plant With Time Delay

Recommendations for PID-tuning coefficients are based on root-loci diagrams drawn by the numerical tool, which are similar in form to those used for PI-compensator design (Figure 19), and SIMULINK simulations for verification. Figure 23 depicts the dynamic behavior of a second-order plant, modeled by a double pole at $s = -0.10$, with a two-second time delay. The plant is controlled by a PID compensator with a double zero at $s = -0.13$. The double zero is slightly to the left of the region which allows closed-loop poles to reenter the real axis. After colliding and departing the real axis, open-loop plant and integrator poles move to the left, roughly parallel to the real axis as gain continues to increase, before moving away from the real axis and back toward the right-half plane. As was the case for PI-tuning of a first-order plant with time delay, PID-tuning recommendations are generated from root loci with this form because, for a limited range in compensation gain, closed-loop poles move relatively far to the left of their open-loop positions.

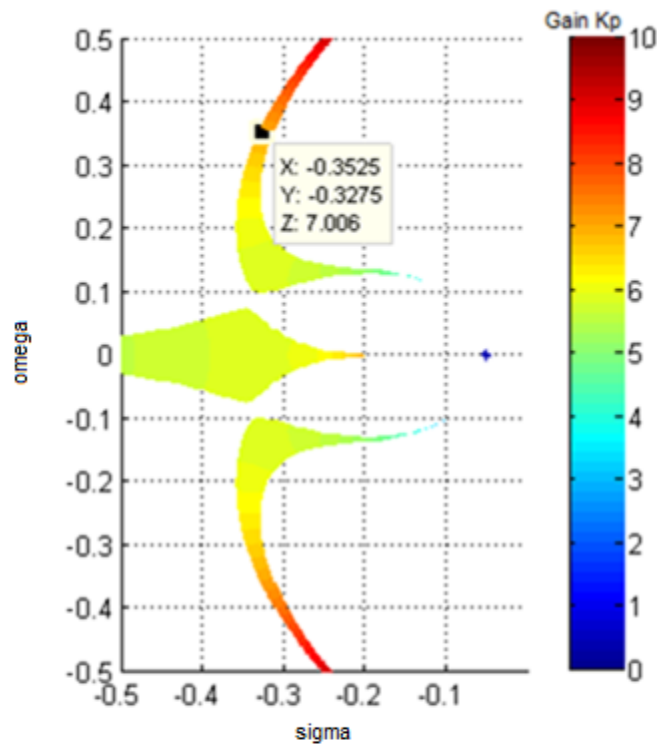


Figure 23. Root Loci of a PID-Compensated Second-Order Plant with Time Delay

Root loci drawn by the numerical method depict the paths of closed-loop poles as PID compensation gain is varied. Plant is second order (with double pole at $s = -0.1$) and has a time delay of two seconds ($NTD = 0.2$). PID double zero is at $s = -0.13$, gain is shown as Z in caption.

Note the PID double zero is slightly to the left of the region on the real axis that allows plant and integrator poles to reenter the real axis, then one would approach and be consumed by the double zero, as gain increases. Instead, a pole due to time delay gets to the double zero first.

For a given second-order plant, the left-most point on the real axis that the PID double zero can be placed, and still draw plant and integrator poles to its left to reenter the real axis, is forced to the right as time delay increases. The relationship between NTD and the maximum distance of separation between the double zero and the open-loop double pole is shown in Table 2. When $NTD \geq 0.5$ the double zero can no longer be placed far enough to the left of the plant's open-loop double pole to allow for reasonable

error in modeling the plant, so PID-tuning recommendations are stated in Table 3 only for the range $0.05 \leq NTD \leq 0.5$. Two transient response performance metrics, overshoot and 2% settling time, for the tuning coefficients are stated in Table 3 and plotted as a function of NTD in Figure 24.

Table 2
Comparison of the PID Double Zero Position That is the Basis for Tuning Coefficient Recommendations to the Left-Most Position Where Loci Reenter the Real Axis

	Finding	Recommendation
NTD	Leftmost position on the real axis the PID double zero can be placed, where plant poles will reenter the real axis (multiples of plant double pole open-loop position)	Position of the PID double zero selected for determining tuning coefficients (multiples of plant double pole open-loop position)
0.05	2.71	3.00
0.10	1.72	2.00
0.20	1.25	1.30
0.30	1.11	1.15
0.40	1.05	1.10
0.50	1.03	1.06

Comparison between two key locations of the PID double: 1) the left-most position on the real axis that permits plant poles to reenter the real axis, and 2) the position used to produce tuning coefficients. Choice of the position for coefficients (listed in Table 3) is based on root loci drawn by the numerical algorithm, matching the form shown in Figure 23. For each value of NTD, SIMULINK simulations were created to verify the design goal is achieved. The goal is rapid return to steady-state conditions, after unit-step changes in set point or load disturbance, by achieving net movement of closed-loop poles to the left. Note: the left-most position the double-zero can be placed and still allow loci to reenter the real axis, moves to the right, toward the open-loop plant double pole, as normalized time delay NTD increases. This effect conveys deterioration in the ability of a PID feedback loop to accelerate the plant as NTD increases

Table 3

Recommended PID-Tuning Coefficients for a Second-Order Plant with Time Delay

NTD	Recommendation	Finding	Recommendation	Finding	Finding
	PID double-zero position	Equivalent T_i, T_d	k_p	2% settling time after a unit-step change in set point	2% settling time after a unit-step change in load disturbance
	(multiples of plant double pole open-loop position)	(multiples of one of the plant's double poles' time constant)		(multiples of one of the plant's double poles' time constant)	(multiples of one of the plant's double poles' time constant)
0.05	3.00	0.67, 0.17	55	0.7	0
0.10	2.00	1.00, 0.25	20	0.9	1.3
0.20	1.30	1.54, 0.38	7	2.6	3.5
0.30	1.15	1.74, 0.43	4	4.7	5.1
0.40	1.10	1.82, 0.45	2.2	6.9	7.4
0.50	1.06	1.89, 0.47	1.7	8.8	8.9

Recommendations for PID-tuning coefficients k_p, T_d and T_i in control of a second-order plant with time delay. Coefficients meet the design goal of rapid, and roughly equivalent 2% settling times, after a unit-step change in set point or load disturbance. Simulations of system response used to generate these settling times used the suggested method of reducing set point overshoot described in the text. Note shortest settling times occur with small NTD.

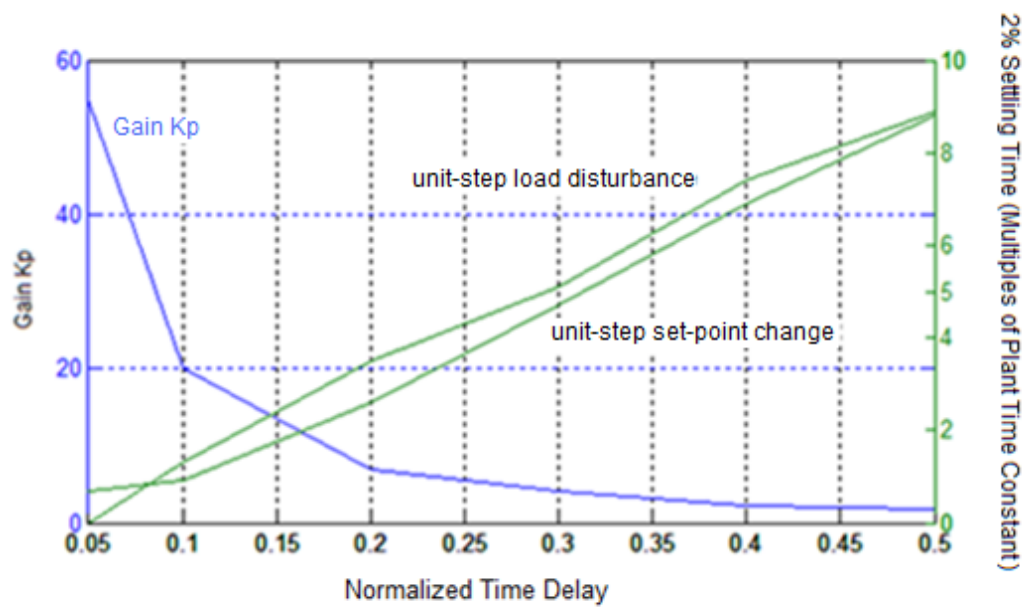


Figure 24. Performance of Recommended PID-Tuning Coefficients

Plot of recommended PID gain as a function of normalized time delay NTD, and expected performance, as measured by 2% settling time, for a second-order double-pole plant. Note, settling time increases linearly with NTD, when NTD = 0.5 closed-loop settling time is almost the same as the open-loop plant which is approximately $4 * (2.1 * T)$, where T is the time constant associated with one of the plant's double poles.

PID coefficient recommendations for optimal load-disturbance response have been found to vary from those giving optimal set-point change response (Zhuang & Atherton, 1993). The tuning recommendations given here optimize both set-point change and load-disturbance settling times, though they intrinsically favor load disturbance and lead to overshoot after a set-point change, by applying an overshoot-reduction method. The natural rate of integration called for by the PID algorithm's integral term is linearly reduced as the distance between set point and process value grows, such that the integration rate reduces to zero when:

$$abs(Set\ Point - Process) \geq \frac{1}{k_p}$$

After modifying the PID algorithm with this overshoot-reduction technique, PID-tuning coefficients shown in Table 3, will give rapid and roughly equivalent settling times after a unit-step change in set point or load disturbance, and with no overshoot.

4.0 Conclusion

Root loci for systems with a variety of polynomial transfer functions are commonly drawn and discussed in textbooks on classical control theory. However, pure polynomial transfer functions cannot exactly express the effect of time delay. Time delay is prevalent in control systems, so it is of interest to see what loci for time-delay systems actually look like. In this study, a comprehensive set of root loci for these systems is exhibited and then used to design PID compensators for first-order and second-order plants with time delay.

Root loci for plants with time delay are drawn by a numerical method developed here. The method avoids the need to approximate time delay and the mismatch between predicted and actual response that sometimes results (see Figure 5). The methodology used here shows:

- How to identify the true positions of closed-loop poles in feedback systems with time delay.
- How to identify marginal gain (Figure G2) in feedback systems with time delay.

Predictions of the numerical method developed here are consistent with mathematical analysis and show:

- In feedback systems with time delay an infinite number of separate and distinct closed-loop pole trajectories will exist. As compensation gain increases from zero, closed-loop poles follow paths that start at the far left

extreme of the real axis, separated vertically by a distance of $\frac{2\pi}{\theta_d}$ where θ_d is time delay, and travel to the right, roughly parallel with the real axis. Some time-delay poles may be consumed by plant zeros, or system poles may be contributed, but ultimately an infinite number of closed-loop poles trend along horizontal asymptotes as gain increases, toward the right extreme of the real axis, at vertical positions $\pm i \frac{\pi}{\theta_d} (2l + 1)$ where $l = 0, 1, 2, \dots$ (see Appendix G, Figure G7).

- In a first-order system with time delay, the two closed-loop poles that cross the imaginary axis closest to the real axis are dominant because they are the first poles to cross into the right-half plane (Appendix G and Figure 13).
- The behavior of a first-order plant with time delay is similar to the behavior of a higher-order plant without time delay. As shown in Figure 16, the first-order plant with time delay begins to ring as compensation gain increases.
- An explanation is given for the limitation in the ability of PI and PID controllers to effectively accelerate open-loop transient response, as NTD increases. There is a restriction on how far to the left of origin a compensator zero can be placed, so that closed-loop poles travel to its left and accelerate the system (see Appendices I and J). As shown in Figure

23, a pole due time delay gets to the compensator zero first, so plant and integrator poles must move into the right half of the s -plane.

The culmination of this research is the generation of PI-tuning coefficients for first-order plants with time delay, and PID-tuning coefficients for second-order plants with time delay. Coefficients are stated for a range in normalized time delay of $0.05 \leq NTD \leq 0.5$. When used with a modification that reduces overshoot of the final value after a set-point change, these coefficients give rapid return to within 2% of steady state after a unit-step change in set point or load disturbance.

References

- Arfken, G. (1970). *Mathematical methods for physicists* (2nd ed.). London: Academic Press.
- Astrom, K. J., & Hagglund, T. (1988). *Automatic tuning of PID controllers*. Research Triangle Park, NC: Instrument Society of America.
- Astrom, K. J., & Hagglund, T. (1995). *PID controllers: Theory, design, and tuning* (2nd ed.). Research Triangle Park, NC: Instrument Society of America.
- Barnes, T. J. D., Wang, L., & Cluett, W. R. (1993, June). A frequency domain design method for PID controllers. *Proceeding of the American Control Conference*, San Francisco, 890–894.
- Beyer, W. H. (Ed.). (1981). *Standard mathematical tables*. Boca Raton, FL: CRC Press.
- Deshpande, P. D., & Ash, R. H. (1981). *Elements of computer process control with advanced applications*. Research Triangle Park, NC: Instrument Society of America.
- Distefano, J. J., Stuberrud, A. R., & Williams, I. J. (1995). *Feedback and control systems* (2nd ed.). United States of: McGraw-Hill.
- Hang, C. C., Astrom, K. J., & Ho, W. K. (1991). Refinement of the Ziegler-Nichols tuning formula. *IEE Proceedings*, 138(2), 111–118.
- Ho, W. K., Gan, O. P., Tay, E. B., & Ang, E. L. (1996). Performance and gain and phase margin of well-known PID tuning formulas. *IEEE Transactions on Control Systems Technology*, 4(4), 473–477.

- Ho, W. K., Hang, C. C., & Zhou, J. H. (1995). Performance and gain and phase margin of well-known PI tuning formulas. *IEEE Transactions on Control Systems Technology*, 3(2), 245–248.
- Mann, G. K. I., Hu, B. -G., & Gosine, R. G. (2001). Time-domain based design and analysis of new PID tuning rules. *IEE Proceedings-Applied Control Theory*, 148(3), 251–261.
- Ogata, K. (1970). *Modern control engineering*. Englewood Cliffs, NJ: Prentice Hall.
- Ogata, K. (2002). *Modern control engineering* (4th ed.). Englewood Cliffs, NJ: Prentice Hall.
- Richard, J-T. (2003). Time-delay systems: an overview of some recent advances and open problems. *Automatica*, 39, 1667–1694.
- Silva, J. S., Datta, A., & Battacharyya, S. P. (2001). Controller design via Pade approximation can lead to instability. *Proceedings of the 40th IEEE Conference on Decision and Control*, 4733–4736.
- Silva, J. S., Datta, A., & Battacharyya, S. P. (2002). New results on the synthesis of PID controllers. *IEEE Transactions on Automatic Control*, 47(2), 241–252.
- Sipahi, R., Niculescu, S., Abdallah, C., & Michiels, W. (2011). Stability and stabilization of systems with time delay: limitations and opportunities. *IEEE Control Systems Magazine*, 31(1), 38–53.
- Stefani, R. T., Shahian, B., Savant, C. J., & Hostetter, G. H. (2002). *Design of feedback systems*. New York, NY: Oxford University Press.

- Vajta, M. (2000). Some remarks on Padé-approximations. *3rd Tempus-Intcom Symposium*, Vezprem, Hungary, 1–6.
- Valkenburg, M. E. (1964). *Network analysis* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.
- Zhuang, M., & Atherton, D. P. (1993, May). Automatic tuning of optimum PID controllers. *IEE Proceedings-D*, 140(3), 216–224.
- Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, 759–765.

Appendices

Appendix A

Laplace Transform

The Laplace integral transform simplifies the process of solving ordinary differential equations, which describe the physical systems, or *plants*, we want to control. Time-based differential equations are converted to polynomial functions of the complex variable $s = \sigma + i\omega$, simplifying analysis of feedback dynamics.

A function in time $f(t)$ is transformed to a function of s (Arfken, 1970, p. 688)

$$F(s) = L[f(t)] = \lim_{a \rightarrow \infty} \int_0^a f(t) e^{-st} dt = \int_0^{\infty} f(t) e^{-st} dt \quad (\text{A1})$$

Consider a simple, first-order plant, its time response $f(t)$ to an impulse input will exponentially decay, with time constant T

$$f(t) = e^{-t/T} \quad (\text{A2})$$

The Laplace transform of the plant $F(s)$, its *transfer function*, is

$$F(s) = L[f(t)] = \int_0^{\infty} e^{-\frac{t}{T}} e^{-st} dt = \int_0^{\infty} e^{-(s+1/T)t} dt = \frac{1}{s + \frac{1}{T}}$$

The plant transfer function has a pole, goes to infinity, at $s = -\frac{1}{T}$.

Appendix B

Inverse Laplace Transform and Residue Theorem

When the Laplace Transform is applied in characterizing the dynamic behavior of a feedback system, the ability to convert back to the time domain is eventually needed.

Transformation of a function of a complex variable, $F(s)$, into a function of time, $f(t)$, is accomplished with the Inverse Laplace Transform $L^{-1}[F(s)]$ (McCollum 1965)

$$L^{-1}[F(s)] = f(t) = \frac{1}{2\pi i} \oint F(s) e^{st} ds \quad (\text{B1})$$

The contour integral must surround a region in the s plane that contains all the poles of $F(s)$.

The residue theorem from complex analysis helps us apply the Inverse Laplace Transform. Residues of a polynomial $\text{Res}(F(s), s_i)$ are the b_k coefficients in its Laurent expansion, they will be calculated below through partial fraction expansion.

The residue theorem

$$\oint F(s) = 2\pi i \sum_{i=1}^N \text{Res}(F(s), s_i) \quad (\text{B2})$$

states the sum of residues within an encircled region is proportional by $2\pi i$ to the contour integral around the region.

As an example, the time-domain response is determined for a first-order plant, with transfer function $G(s) = s/(s + a)$, excited by a unit-step input $R(s) = 1/s$.

The output of the plant $Y(s)$ is the product of its input $R(s)$ and the plant transfer function $G(s)$

$$Y(s) = \frac{1}{s} \frac{1}{(s+a)} \quad (\text{B3})$$

Using the residue theorem, the Inverse Laplace Transform is computed as the sum of the residues of $Y(s)e^{st}$

$$L^{-1}[Y(s)] = L^{-1} \left[\frac{1}{s(s+a)} \right] = f(t) = \lim_{s=0} \left[s \cdot \frac{e^{st}}{s(s+a)} \right] + \lim_{s=-a} \left[(s+a) \cdot \frac{e^{st}}{s(s+a)} \right] = \frac{1}{a} (1 - e^{-at}) \quad (\text{B4})$$

Appendix C

Tools

Frequency analysis. In steady-state frequency analysis, a plant is excited by a sinusoidal signal with a constant frequency and magnitude. After a transient period elapses the system output will oscillate at steady-state and, if the system is linear, it oscillates at the same frequency as the input signal. A difference between the phase and magnitude of the input and output signals, however, will probably be present. The manner in which the plant alters the phase and magnitude of the input signal, as a function of frequency, is an indication of the plant's stability in a closed-loop system. Stability can be determined from gain and phase margins (Ogata, 1970, p. 430) and from “the phase crossover frequency” ω_c (Stefani et al., 2002, p. 465), the frequency at which input and output sinusoids are **180°** out of phase.

Gain margin is the ratio of the magnitudes of input and output signals at ω_c . If gain margin is greater than one the system is stable, when it's equal to one the system will continuously oscillate, being marginally stable. When gain margin is less than one the system is unstable.

“Phase margin is the amount of additional phase lag at the crossover frequency ω_c required to bring the system to the verge of instability” (Ogata, 2002, p. 562). In systems that are not second order “phase and gain margins give only rough estimates of the effective damping ratio of the closed-loop system” (Ibid, p. 565).

The introduction of time delay affects only the phase of the output signal, its magnitude remains unchanged. The transfer function of time-delay (Equation 1) at steady-state, $\sigma = 0$, is

$$G_{delay}(s) = e^{-s\theta} = e^{-(\sigma+i\omega)\theta} = e^{-i\omega\theta},$$

where θ is time delay. Thus, time delay adds a phase lag of $-\omega\theta$, a value which increases with frequency and the actual time delay, as compared to the delay-free system.

S-plane analysis. The transient response of a linear system is comprised by one or more first- or second-order response components that, in a stable system, decay exponentially. Any first- or second-order response can be correlated with a single pole or pair of poles, respectively, in the s-plane as shown below in Figure C1. Plant poles that lie off the real axis must occur in complex conjugate pairs for the plant transfer function coefficients to be real.

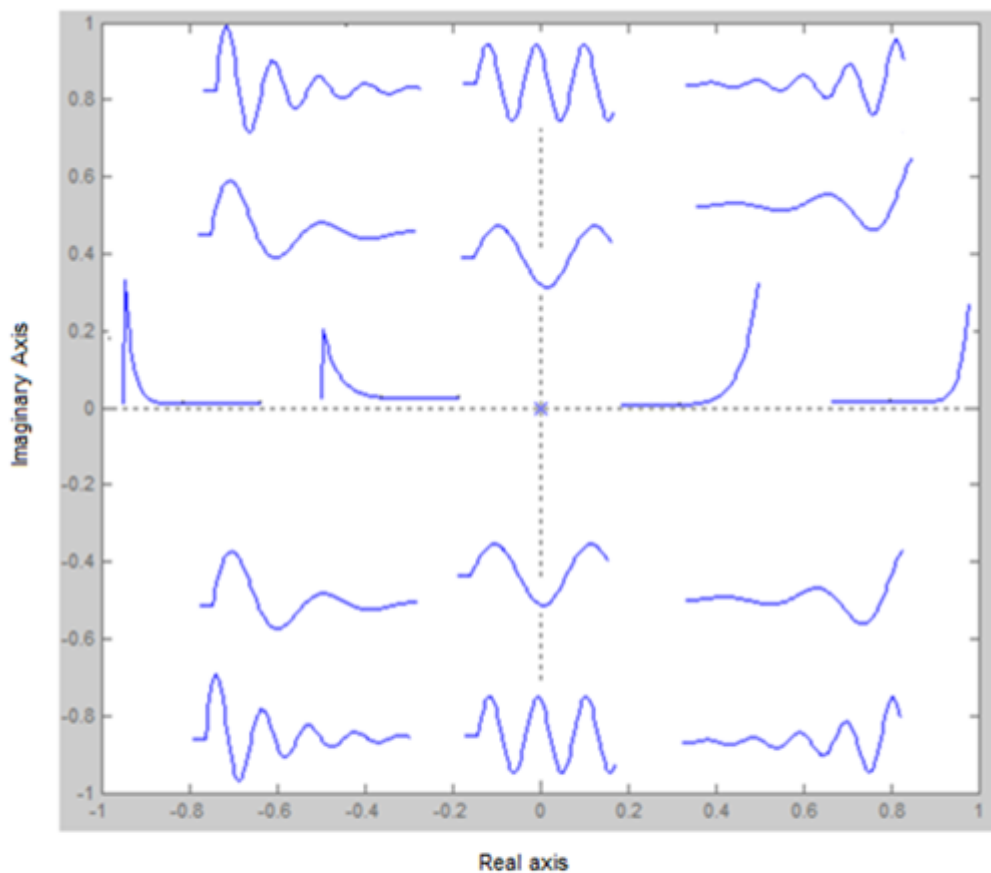


Figure C1. Impulse Responses for Pole Locations in the S-Plane

This map of the s-plane depicts the relationship between a pole's location and its associated impulse response .
 Note: single poles may only occur on the real axis, complex conjugate pole pairs occur off the real axis.

Impulse responses of first- and second-order plants are shown in Figure C1; this visual aid describes the type of time-response that can be expected for a pole's location in the s-plane. Total transient response is the sum of all individual time-responses in a system (Valkenburg, 1964, pp. 280-284). Given a polynomial expression for total system output $Y(s)$, the amplitude of each component of response is determined through partial fraction expansion of $Y(s)$.

The following rules summarize the relationship between impulse response and pole positions in s-plane shown in Figure C1

- Poles in the left half of the s-plane represent stable response
- Poles on the real axis indicate the absence of oscillatory content
- Poles off the real axis indicate the presence of oscillatory content.

Comparison of closed-loop pole analysis to frequency analysis. Closed-loop pole analysis can produce root loci which show the movement of closed-loop poles, and thus describe the dynamic behavior of a plant in feedback, as a parameter, typically compensation gain, varies. Steady-state frequency analysis assesses closed-loop stability by examining the open-loop plant and how it transforms a sinusoidal input to the resulting sinusoidal output, as a function of frequency. The two techniques intersect along the imaginary axis in the s -plane where $\sigma = 0$ and $s = i\omega$. Poles that lie on this axis represent impulse responses that are continuous oscillations (Stefani et al., 2002, p. 465).

In root-locus analysis, if a pole is on the imaginary axis and no poles are in the right half of the s -plane, the system output continuously oscillates. The compensation gain associated with this pole position is the reciprocal of what's referred to as *gain margin* in frequency analysis.

The vertical position of the pole on the imaginary axis is the phase cross-over frequency ω_c , the frequency of excitation at which input and output sinusoids are 180° out of phase.

Appendix D

Identifying the Poles and Zeros of a PID Compensator

The method of using root locus in this paper for tuning PID controllers makes proportional gain k_p a common factor to all three PID terms by expressing the integral term as $\frac{k_p}{T_i s}$, where T_i is *integral time*, and the derivative term as $k_p T_d s$ where T_d is *derivative time* (Astrom & Hagglund, 1995, p. 6)

$$\begin{aligned} G_{PID}(s) &= k_p + \frac{k_i}{s} + k_d s = k_p + \frac{k_p}{T_i s} + k_p T_d s \\ &= k_p T_d \left(s + \frac{1}{T_i T_d s} + \frac{1}{T_d} \right) = k_p T_d \frac{s^2 + \frac{1}{T_d} s + \frac{1}{T_i T_d}}{s} \end{aligned} \quad (D1)$$

A PI compensator's zero location then depends only on T_i , a PID compensator's two zeros' locations, identified below with the quadratic equation, depend only on T_i and T_d .

$$s = \frac{-\frac{1}{T_d} \pm \sqrt{\left(\frac{1}{T_d}\right)^2 - \frac{4}{T_i T_d}}}{2} = -\frac{1}{2T_d} \pm \sqrt{\left(\frac{1}{2T_d}\right)^2 - \frac{1}{T_i T_d}} = -\frac{1}{2T_d} \pm \frac{1}{2T_d} \left[1 - \frac{4T_d}{T_i} \right]^{\frac{1}{2}} \quad (D2)$$

Imaginary zeros. Both PID compensator zeros lie off the real axis when the argument of the square root term in Equation D2 is negative because the square root, and PID zeros, will have imaginary components. The argument of the square root term is

$$\left(\frac{1}{T_d}\right)^2 - \frac{4}{T_i T_d}$$

which is less than zero when

$$T_i < 4T_d$$

Real zeros. Both PID compensator zeros lie on the real axis when the argument of the square root term is greater to or equal to zero

$$\left(\frac{1}{T_d}\right)^2 - \frac{4}{T_i T_d} \geq 0$$

or

$$T_i \geq 4T_d \tag{D3}$$

Both zeros lie at the same position, a repeated or double zero, when the argument of the square root term equals zero; this occurs when

$$T_i = 4T_d \tag{D4}$$

This ratio of T_i to T_d is recommended in Ziegler-Nichols' tuning formulas (Ziegler & Nichols, 1942).

As T_i becomes very large compared to T_d , the pole-zero configuration of a PID compensator approaches that of a PI-only compensator. Expansion of the expression for zero positions resulting from applying the quadratic equation above

$$s = -\frac{1}{2T_d} \pm \frac{1}{2T_d} \left[1 - \frac{4T_d}{T_i} \right]^{\frac{1}{2}} \quad (\text{D5})$$

Applying the binomial series expansion (Beyer, CRC Standard Mathematical Tables, 1981, p. 347) gives

$$s = -\frac{1}{2T_d} \pm \frac{1}{2T_d} \left[1 - \frac{1}{2} \frac{4T_d}{T_i} + \frac{\left(\frac{1}{2}\right)\left(-\frac{1}{2}\right)}{2!} \left(\frac{4T_d}{T_i}\right)^2 - \frac{\left(\frac{1}{2}\right)\left(-\frac{1}{2}\right)\left(-\frac{3}{2}\right)}{2!} \left(\frac{4T_d}{T_i}\right)^3 + \dots \right] \quad (\text{D6})$$

If $T_i \gg T_d$, higher-order terms vanish

$$s = -\frac{1}{2T_d} \pm \frac{1}{2T_d} \left[1 - \frac{1}{2} \frac{4T_d}{T_i} + 0 \right] = -\frac{1}{2T_d} \pm \left[\frac{1}{2T_d} - \frac{1}{T_i} \right] \quad (\text{D7})$$

or

$$s \cong -\frac{1}{T_d}, -\frac{1}{T_i} \quad (\text{D8})$$

Appendix E

Numerical Computation of Root Locus With and Without Time Delay

The numerical method developed in this paper finds poles of feedback systems with time delay by brute-force. The roots of the systems' transcendental equation are found by calculating the value of the open-loop transfer function at each point on a grid of finely spaced points within a region of interest in the s plane.

For example, consider the closed-loop characteristic equation of a proportionally-compensated plant $G_p(s)$ with time delay θ

$$1 + k_p G_p(s) e^{-s\theta} = 0 \quad (\text{E1})$$

where compensation gain is k_p .

Values of s that represent positions of poles in the s plane satisfy the characteristic equation's angle condition

$$\text{Angle}[k_p G_p(s) e^{-s\theta}] = \text{Angle}[-1] \quad (\text{E2})$$

By recognizing that any angle θ equals $\theta + 2\pi$ it is clear how the exponential component contributes an infinite number of poles to the system

$$\text{Angle}[k_p G_p(s) e^{-s\theta}] = (\pi \pm 2\pi l) \text{ radians}, \quad l = 0, 1, 2 \dots \infty \quad (\text{E3})$$

The value of $Angle[G_p(s)e^{-s\theta}]$ is computed at each point on a grid, locations where this value is within a small range of $180^\circ = \pi$ radians, the *decision criterion* or as labeled in the code *phase variation*, are considered to be on the loci.

The characteristic equation's magnitude condition is

$$|k_p G_p(s)e^{-s\theta}| = |-1| = 1 \quad (E4)$$

It gives the value of compensation gain at any location on the loci

$$k_p = \frac{1}{|G_p(s)e^{-s\theta}|} \quad (E5)$$

The open-loop transfer function of a PI compensated first-order plant with time delay $G(s)$ can be written

$$G(s) = k_p \frac{1 + \frac{1}{T_i}}{s} \frac{1}{Ts+1} e^{-s\theta} \quad (E6)$$

where T_i is integral time and T is the plant time constant.

An excerpt from MATLAB code developed in this study shows the open-loop transfer function written as

```
G = exp(-s*TimeDelay)*(1/(TimeConst*s+1))*((s+tiZero))/s);
```

The value of G is calculated at all points on the grid in the region of interest. Locations where $Angle[G]$ is within a specified range around 180° are stored for plotting, they form the loci.

```
if ( (thePhase > (180 - phaseVariation))
    & (thePhase < (180 + phaseVariation)) )
```

Each point is color-coded to match the value of compensation gain k_p which is calculated from the magnitude condition

```
K = 1 / abs(G);
```

Root loci are depicted by plotting the magnitude of k throughout the region of interest in the s -plane which has boundaries at ω_{min} , ω_{max} , σ_{min} , and σ_{max} .

```
surf(omegaMin:omegaIncrement:omegaMax,
     sigmaMin:sigmaIncrement:sigmaMax,
     abs(k));
```

The open-loop transfer function of a PID compensated second-order plant with time delay can be expressed

$$G_{PID}(s) = k_p T_d \frac{s^2 + \frac{1}{T_d} s + \frac{1}{T_i T_d}}{s} \frac{1}{(T_1 s + 1)} \frac{1}{(T_2 + 1)} e^{-s\theta}$$

$$= k_p T_d \frac{(s+TiZero)(s+TdZero)}{s} \frac{1}{(T_1 s+1)} \frac{1}{(T_2+1)} e^{-s\theta} \quad (E7)$$

This form clearly shows the open-loop transfer function's two poles and two zeros. Note $k_p T_d$ is a common factor to all terms. In MATLAB code the open-loop transfer function is written

```
GH = exp(-s * TimeDelay)
    * (1/((TimeConst1 s + 1) ) *(1 / (TimeConst2 s + 1))
    * ((s + tiZero) * (s + tdZero)) / s;
```

Points on the loci are again identified by the angle condition and the decision criterion.

$$\text{Angle}[G_{PID}(s)] = \text{Angle}[-1] = \pi \text{ radians}$$

Then compensation gain k_p at each point on the loci is found from the magnitude of GH

$$1 / \text{abs}(\text{GH})$$

The derivative time T_d is extracted from this quotient as follows to get compensation gain k_p

```
kp(sigmaCounter, omegaCounter) = (tiZero + tdZero)*1/abs(GH);
```


because

$$\frac{1}{T_d} s = (TiZero + TdZero) s \quad (E8)$$

An alternate method of finding roots of transcendental equations. An alternative to the technique developed in this paper would be to use the MATLAB function `fsolve()`, which is designed to find roots of simultaneous non-linear equations. Real and imaginary parts of the closed-loop characteristic equation are both non-linear, `fsolve()` could be used to find their roots and plot root loci.

For example, to find the poles of a closed-loop system comprised purely of time delay, examine its characteristic equation

$$1 + k_p e^{-s\theta} = 0 \quad (E9)$$

Real and imaginary parts are brought out by applying $s = \sigma + i\omega$ and Euler's law

$$e^{i\theta} = \cos \theta + i \sin \theta$$

$$1 + k_p e^{-\sigma\theta} \cos \omega\theta = 0, \quad \text{real part}$$

$$k_p e^{-\sigma\theta} \sin \omega\theta = 0, \quad \text{imaginary part}$$

Both parts are encoded into a single MATLAB .m file function

```

function z=char_equation(s) % pass in initial guess at solution
% s is 2x1 array
% s(1,1) = real(s) = sigma
% s(2,1) = imag(s) = omega

    Kp=0.5;
    theta=1;

    z(1)= 1 + Kp * exp(-s(1,1)) * cos(-s(2,1)*theta);
    z(2) = Kp * exp(-s(1,1)) * sin(-s(2,1)*theta);
end

```

In searching for a value of s that satisfies both real and imaginary parts `char_equation()` is called reiteratively by MATLAB once `fsolve()` is invoked from the command line. An initial guess at the real and imaginary parts of a solution is passed to `fsolve()`, the initial guess is packaged in array format

```
>> s =[0; 3];
```

Then the search for a solution is launched by calling `fsolve()`

```

>> x = fsolve(@char_equation, s)
x =
    -0.6931
     3.1416

```

To draw root locus using `fsolve()` the s -plane would be scanned, as is done with the method developed in this paper, and each point would be used as initial conditions for `fsolve()`. Compensation gain k_p would also need to be varied through an appropriate range, while using each point in the s -plane as an initial condition, to fill-in the loci.

The advantage of the method developed in this paper is it takes fewer steps to determine whether a closed-loop pole exists at a specific location. Once the location of a closed-loop pole is known it is simple to calculate directly its associated compensation gain as shown above in Equation E5.

Appendix F

Root Loci for Simple Plants Drawn Using Approximated Time Delay

“We cannot apply conventional root locus rules to analyze a true time-delay system because the root locus rules require rational transmittance (polynomial ratios) and the true delay $G_D(s)$ is irrational” (Stefani et al., 2002, p. 293). The transfer function of time delay is

$$G_D(s) = e^{-s\theta} \quad (\text{F1})$$

where θ is the delay.

For small time delays, $G_D(s)$ can be approximated (Ogata, 2002, p. 383) by the first two terms in a Taylor series

$$e^{-s\theta} \cong 1 - s\theta \quad \textit{Taylor Approximation} \quad (\text{F2})$$

or a first-order plant

$$e^{-s\theta} \cong \frac{1}{s\theta + 1} \quad \textit{First - Order Approximation} \quad (\text{F3})$$

Padé approximations “approximate delay with a polynomial ratio.” (Stefani et al., 2002, p. 293), a second-order Pade approximation can be derived as follows

$$e^{-ti\theta} = \frac{e^{-\theta s/2}}{e^{\theta s/2}} \cong \frac{1 - \theta s/2}{1 + \theta s/2} = \frac{-(s - 2/\theta)}{s + 2/\theta} \quad \text{Second - Order Pade}$$

(F4)

Padé approximations “are usually superior to Taylor expansions when functions contain poles” (Vajta, 2000). A Padé approximation can be described by a rational polynomial having a numerator of order m , and denominator of order n , written as

$$R_{m,n}(x) = \frac{P_m(x)}{Q_n(x)} \quad \text{(F5)}$$

where the definitions of numerator and denominator are

$$P_m(x) = \sum_{k=0}^m \frac{(m+n-k)! m!}{(m+n)! k! (m-k)!} (-x)^k$$

$$Q_n(x) = \sum_{k=0}^n \frac{(m+n-k)! n!}{(m+n)! k! (n-k)!} (x)^k$$

(F6)

The series of root loci diagrams in Figure F1 all depict closed-loop systems comprised of a pure time-delay plant; where time delay is approximated by a variety of Padé polynomials having up to a fifth-order numerator or denominator.

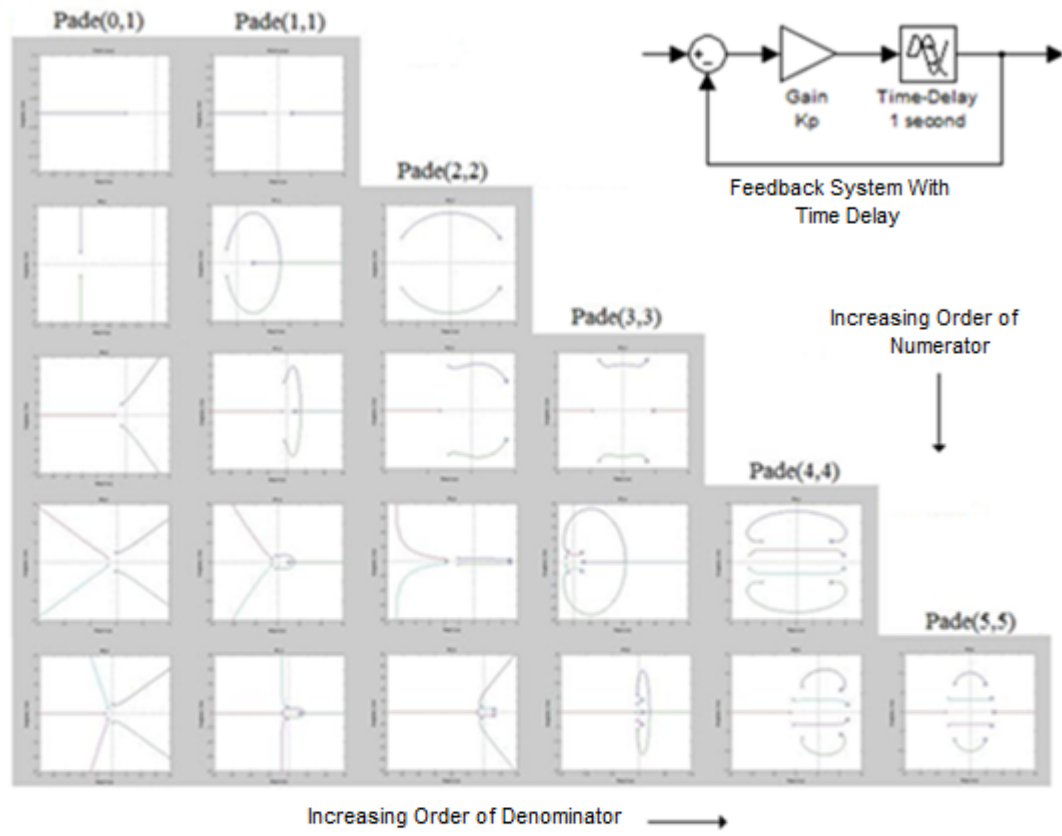


Figure F1. Comparison of Padé Time-Delay Approximations

Comparison of Pade approximations of time delay: each root locus diagram shows movement of closed-loop poles, for a feedback system comprised of a proportionally compensated plant that is purely a one-second time delay. Each plot corresponds to the Pade approximation labeled $P(n, m)$ where n is the numerator order and m is the denominator order.

Note: the variation in system dynamics predicted by the possible time-delay approximations leads to a mix and confusing array of results. The accuracy of our predictions of system response rely on the choice of approximation.

The numerical method developed here enabled the comparison, shown earlier in Figure 5 of root loci that incorporate three different types of time-delay approximation to root loci drawn using true time delay. In the figure, the actual mismatch between true time-delay root loci and time-delay approximation root loci is apparent for the three types of systems shown: proportionally compensated first-order and second-order plants with time delay and a PID-compensated second-order plant with time delay. The variation in predicted from actual response depends on the type of system and the time-delay approximation used and is sometimes significant.

Appendix G

Root Loci for Simple Plants Drawn Using True Time Delay

The methodology developed here for drawing root loci for systems with time delay is based on analysis of the angle and magnitude conditions of a time-delayed system's characteristic equation. The closed-loop characteristic equation comes from setting the denominator of the closed-loop transfer function $G_{cl}(s)$ equal to zero. Values of s that make it equal to zero, its roots, make $G_{cl}(s)$ blowup, and are closed-loop poles.

For the feedback system with time delay shown in Figure 12, the canonical equation for the closed-loop transfer function $G_{cl}(s)$ is

$$G_{cl}(s) = \frac{X(s)}{Y(s)} = \frac{KG(s)e^{-s\theta}}{1+KG(s)e^{-s\theta}} \quad (\text{G1})$$

Setting the denominator of $G_{cl}(s)$ equal to zero gives

$$1 + KG(s)e^{-s\theta} = 0 \quad (\text{G2})$$

Note the characteristic equation is transcendental. Roots of the characteristic equation are poles of the closed-loop transfer function and both magnitude and angle conditions.

The magnitude condition is

$$|KG(s)e^{-s\theta}| = |-1| = 1 \quad (\text{G2})$$

The angle condition is

$$\text{Angle}[KG(s)e^{-s\theta}] = \pi(1 \pm 2l) \text{ radians}, \quad l = 0, 1, 2, \dots \quad (\text{G3})$$

Euler's formula states the complex number $e^{i\theta}$ is the sum of sine and cosine functions (Ogata, 2002, p. 12)

$$e^{i\theta} = (\cos \theta + i \sin \theta) \quad (\text{G3})$$

The value of $e^{i\theta}$ is a complex number, having a magnitude of 1, and a phase angle θ .

Feedback system comprised of a proportionally-compensated pure time-delay plant. If the proportionally-compensated feedback system in Figure 12 contains the time-delay element only, i.e. $\mathbf{G}(s) = \mathbf{1}$, the closed-loop characteristic equation is

$$1 + KG(s) = 1 + Ke^{-s\theta} = 0 \quad (\text{G4})$$

Marginal gain K_m occurs when one or more closed-loop poles lie on the imaginary axis, and there are no poles in the right half-plane. K_m can be determined by evaluating the characteristic equation on the imaginary axis, where $s = i\omega$

$$|K_m e^{-s\theta}|_{\sigma=0} = |K_m e^{-i\omega\theta}| = 1 \quad (\text{G5})$$

From Euler's formula $|e^{i\omega\theta}| = 1$

$$K_m = 1 \quad (\text{G6})$$

Note for this system marginal gain K_m is always one, regardless of the value of time delay.

Vertical positions of closed-loop poles where loci intersect the imaginary axis $i\omega_l$, are given by the angle condition

$$\text{Angle}[K_m e^{-s\theta}]_{\sigma=0} = \text{Angle}[K_m e^{-i\omega_l\theta}] = \text{Angle}[-1] = \pi \quad (\text{G7})$$

which has solutions at

$$\omega_l = \frac{\pi}{\theta} (1 \pm 2l) \quad l = 0, 1, 2 \dots \quad (\text{G8})$$

Note closed-loop poles are separated vertically by a distance that is inversely proportional to time delay. As time delay increases the linear density of closed-loop

poles, along the direction of the imaginary axis, also increases. Also note vertical positions define a set of odd harmonics, and are consistent with system output shown in Figure 4, a square wave with a period of two seconds.

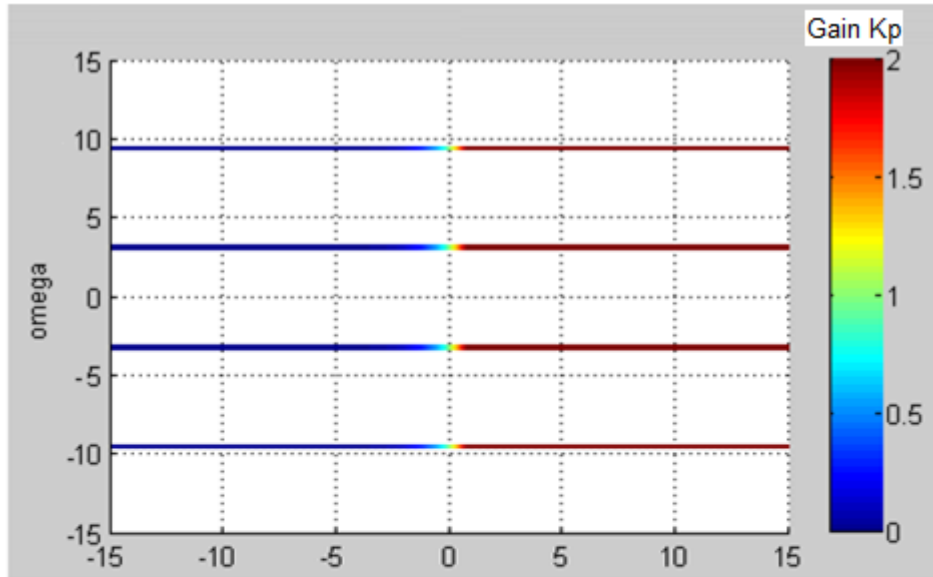


Figure G1. Root Loci of Pure Time-Delay Plant

Root loci drawn by the numerical method developed in this study depict the paths of closed-loop poles for a feedback system comprised of a proportionally-compensated plant that is pure time delay of one second. Note the vertical separation between loci is $(2\pi / \text{delay})$, and all poles cross the imaginary axis at gain $K_p = 1$. This plot accurately represents what the root loci drawn using Pade time-delay approximations (Figure F1) attempt to match.

Feedback system comprised of a first-order plant with time delay. When the feedback system in Figure 12 contains a first-order plant $G(s)$ with time constant T ,

$$G(s) = \frac{1}{Ts+1} \quad (\text{G9})$$

the closed-loop characteristic equation becomes

$$1 + KG(s) = 1 + \frac{K}{Ts+1} e^{-s\theta} = 0 \quad (\text{G10})$$

Marginal gain K_m and the frequency at which the system output will subsequently oscillate are given by applying the magnitude and angle conditions, to identify the poles, on the imaginary axis.

The magnitude condition of Equation G10 becomes

$$\left| \frac{K_m}{Ts+1} e^{-s\theta} \right| = \left| \frac{K_m}{i\omega T+1} e^{-i\omega\theta} \right| = 1 \quad (\text{G11})$$

which yields marginal gain

$$K_m = \sqrt{1 + (\omega T)^2} \quad (\text{G12})$$

To compute the frequency of oscillation ω at marginal gain the angle condition is applied on the imaginary axis

$$\text{Angle} \left[\frac{K_m}{Ts+1} e^{-s\theta} \right] = \text{Angle} \left[\frac{K_m}{i\omega T+1} e^{-i\omega\theta} \right] = \pi(1 \pm 2l) \quad l = 0,1,2 \dots \quad (\text{G13})$$

This yields the following transcendental equation which relates ω to time delay θ and plant time constant T

$$-\omega\theta - \tan^{-1}[\omega T] = \pi(1 \pm 2l) \quad l = 0,1,2 \dots \quad (\text{G14})$$

Unlike the previous example of a system comprised of the pure time-delay plant, where all poles cross the imaginary axis at the same gain, in this system larger compensation gains are required for poles to cross the imaginary axis, the farther they are from the real axis. Note as plant time constant T goes to zero, as expected, points where loci cross the imaginary axis become identical to a pure time-delay system. For large ω , the $\tan^{-1}[\omega T]$ term becomes constant so the vertical spacing between points where loci intersect the imaginary axis approaches $\frac{2\pi}{\theta}$.

Values of ω that satisfy the transcendental equation G14 can be found numerically, as the method developed in this paper does by generating the system's root loci in Figure G2. The frequency of oscillation ω at marginal stability, and the solution of G14, is the vertical position where loci intersect the imaginary axis.

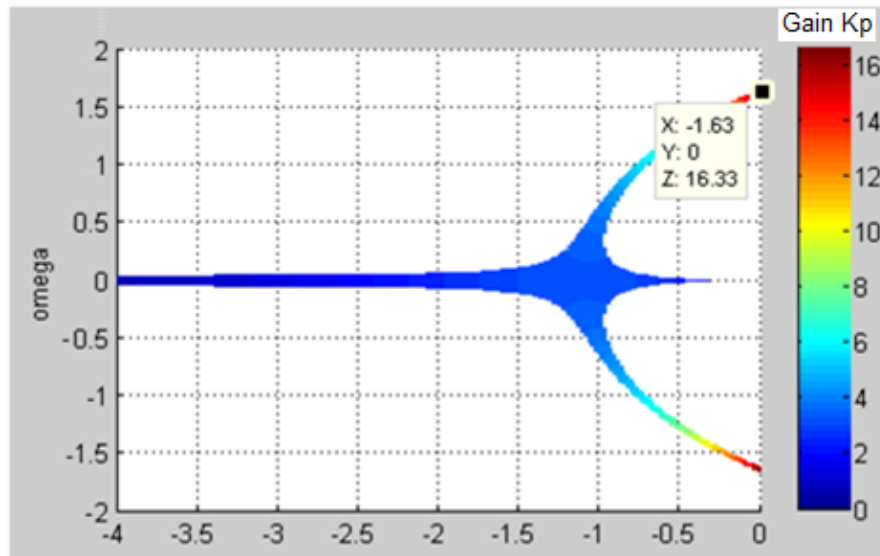


Figure G2. Root Loci of a First-Order Plant With Time Delay Showing Marginal Gain

Root loci drawn, using the numeric method developed in this paper, for a proportionally compensated first-order plant with time delay. Plant time constant $T = 10$ s, plant open-loop pole at $s = -0.1$, time-delay = 1 s, $NTD = 0.1$. Note: marginal gain = $17 = \sqrt{1 + \text{sqr}(w) + \text{sqr}(T)}$, where w is the point of intersection between pole paths and the imaginary axis,

Feedback system comprised of a high-order plant with time delay. The positions and orientations of root loci asymptotes are determined for the feedback system shown in Figure 12 when the plant $G(s)$ is of arbitrarily high-order, having a zero of order n , and a pole of order m .

$$G(s) = \frac{(s+z)^n}{(s+p)^m} \quad (\text{G15})$$

The closed-loop system's characteristic equation is

$$1 + KG(s) = 1 + K \frac{(s+z)^n}{(s+p)^m} e^{-s\theta} = 0 \quad (\text{G16})$$

For small gain, $K \rightarrow 0$. Closed-loop pole positions as $K \rightarrow 0$ are identified by evaluating angle and magnitude conditions of the closed-loop characteristic equation; the magnitude condition is

$$|KG(s)| = \left| K \frac{(s+z)^n}{(s+p)^m} e^{-s\theta} \right| = |-1| = 1 \quad (\text{G17})$$

Applying Euler's equation and a rule of complex arithmetic, the magnitude of the product of two complex numbers is the product of the two numbers' magnitudes, yields

$$K \frac{n|s+z|}{m|s+p|} e^{-\sigma\theta} = 1 \quad (\text{G18})$$

Thus, in regions near the real axis, for $\sigma \rightarrow -\infty$

$$\frac{n|s+z|}{m|s+p|}_{\sigma \rightarrow -\infty} \cong \frac{n|-1|}{m|-1|} = \frac{n}{m} \quad (\text{G19})$$

Substitution into the magnitude condition gives the relationship between σ and K

$$-\sigma\theta = \ln \left[\frac{m}{n} \frac{1}{K} \right] \quad (\text{G20})$$

Thus, as $K \rightarrow 0$, the real part of system pole positions is $\sigma \rightarrow -\infty$.

The angle condition states

$$\text{Angle}\{KG(s)\} = \text{Angle} \left\{ K \frac{(s+z)^n}{(s+p)^m} e^{-s\theta} \right\} = 180^\circ = (\pi \text{ radians})(1 \pm 2l)$$

$$l = 0, 1, 2 \dots \quad (\text{G21})$$

Using a rule of complex arithmetic, the angle of the product of two complex numbers is the sum of the two numbers' angles, gives

$$n \text{ Angle}\{s+z\} - m \text{ Angle}\{s+p\} - \omega\theta = \pi(1 \pm 2l) \quad l = 0, 1, 2 \dots$$

(G22)

On the real axis, $\omega = 0$ and $s = \sigma$, so the angle condition becomes

$$n \text{Angle}\{\sigma + z\} - m \text{Angle}\{\sigma + p\} - \omega\theta = n\pi - m\pi = \pi(1 \pm 2l) \quad l = 0, 1, 2 \dots \quad (\text{G23})$$

Thus, the angle condition is satisfied at $\omega = 0, \sigma \rightarrow -\infty$ if

$$n + m = \#\text{zeros} + \#\text{poles} = \text{odd number} \quad (\text{G24})$$

This solution of the time-delayed system's characteristic equation is, as expected, consistent with a well known rule of root locus construction; regions of the real axis contain loci when there are an odd number of poles plus zeros to the right. This means, if $n + m$ is even, no value of σ satisfies both magnitude and angle condition, i.e. no closed-loop poles exist at the far left extreme of the real axis.

For points that are not on the real axis, but are near the real axis, the small angle approximation can be applied,

$$\text{Angle}\{s + z\}_{\sigma \rightarrow -\infty} = \text{Angle}\{s + p\}_{\sigma \rightarrow -\infty} = \pi \quad (\text{G25})$$

The angle condition becomes

$$n\pi - m\pi - \omega\theta = \pi(1 \pm 2l) \quad l = 0, 1, 2$$

or

$$-\omega\theta = (n - m)\pi + \pi(1 \pm 2l) \quad l = 0,1,2 \quad (\text{G26})$$

Thus, solutions to angle and magnitude conditions have vertical positions

$$\text{for } \sigma = -\infty, \quad \omega = \frac{\pi}{\theta}(1 \pm 2l) \quad l = 0,1,2 \dots \quad n + m \text{ is even} \quad (\text{G27})$$

and

$$\text{for } \sigma = -\infty, \quad \omega = \frac{\pi}{\theta}(2l) \quad l = 0,1,2 \dots \quad n + m \text{ is odd} \quad (\text{G28})$$

For large gain, $K \rightarrow \infty$. Closed-loop pole positions at high gain are identified, as before, by evaluating angle and magnitude conditions. The magnitude condition of this system, Equation G18, once again gives

$$K \frac{n|s+z|}{m|s+p|} e^{-\sigma\theta} = 1 \quad (\text{G29})$$

If $\sigma \rightarrow \infty$

$$\frac{n|s+z|}{m|s+p|}_{\sigma \rightarrow \infty} = \frac{n|1|}{m|1|} = \frac{n}{m} \quad (\text{G30})$$

The magnitude condition simplifies to

$$K \frac{n}{m} e^{-\sigma\theta} = 1 \quad (\text{G31})$$

and is satisfied, as $K \rightarrow \infty$, if $\sigma \rightarrow \infty$.

The angle condition, Equation G21, once again gives

$$n \text{Angle}\{\sigma + z\} - m \text{Angle}\{\sigma + p\} - \omega\theta = \pi(1 \pm 2l) \quad l = 0,1,2$$

Applying the small angle approximation near the real axis

$$\text{Angle}\{\sigma + z\}_{s \rightarrow \infty} = \text{Angle}\{\sigma + p\}_{s \rightarrow \infty} = 0 \text{ radians}$$

simplifies the angle condition

$$n \text{Angle}\{\sigma + z\} - m \text{Angle}\{\sigma + p\} - \omega\theta = -\omega\theta = \pi(1 \pm 2l) \quad l = 0,1,2 \quad (\text{G32})$$

Thus, regardless of whether $m + n$ is even or odd the following set of values for

σ, ω satisfy the characteristic equation as $K \rightarrow \infty$

$$\sigma = \infty, \quad \omega = \frac{\pi}{\theta}(1 \pm 2l) \quad l = 0,1,2 \quad (\text{G33})$$

Other examples of root loci for plants with time delay. The following series of figures provide further examples of root loci for time-delay plants. Figure G3 depicts loci for a single zero, a *differentiator*, compensated with proportional feedback. Closed-loop dynamic behavior of a proportionally-compensated double zero is illustrated with the loci in Figure G4. Figure G5 shows loci and their break-away from the real axis for a single open-loop pole, a first-order plant, with time delay. The position of the break-away point, shown close-up in Figure G6, can be compared to the location derived through a mathematical analysis in the text, equation G37. Figure G7 shows loci for a double-pole, a second-order plant, under proportional compensation. Figure G8 shows loci for a PID-compensated first-order plant with time delay.

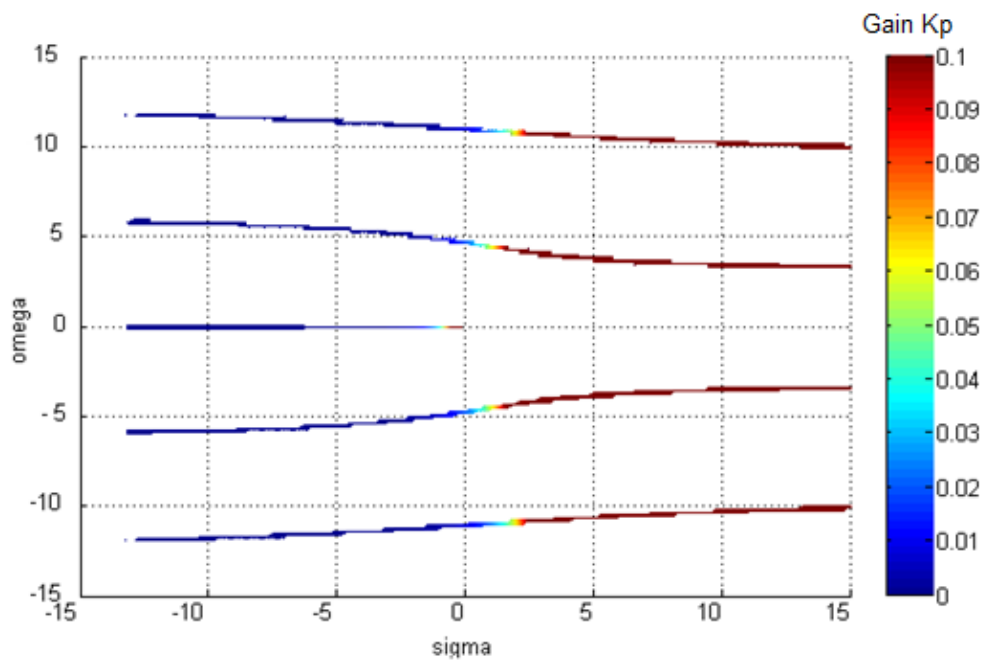


Figure G3. Root Loci of a Single-Zero Plant, a Differentiator, With Time Delay

Root loci, drawn by the numeric tool developed in this paper, for a feedback system comprised of a differentiator, with zero at $s = -0.1$, and time delay of 1s.

Note: a closed-loop pole due to time-delay is consumed by the single plant zero, paths of remaining closed-loop poles curve in such that loci asymptotes are separated vertically by $(2\pi / \text{delay})$ at both extremes of the real-axis.

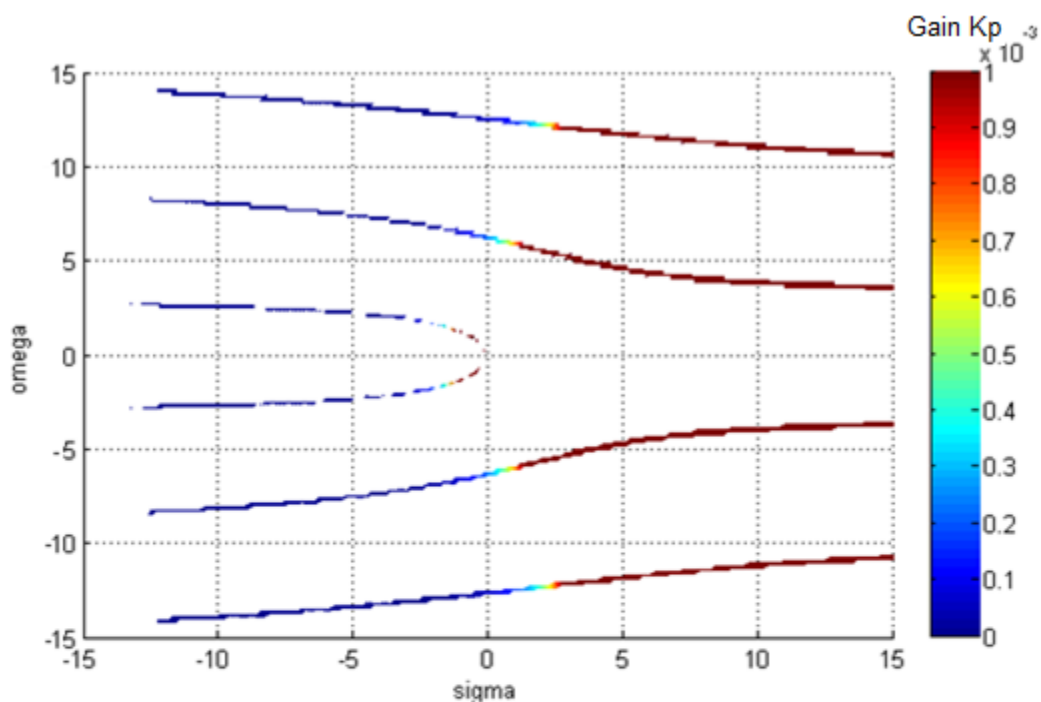


Figure G4. Root Loci of a Double-Zero Plant with Time Delay

Root loci, drawn by the numeric method developed in this paper, for a feedback system comprised of a plant, described by a double zero at $s = -0.1$, and time-delay = 1s.

Note: two closed-loop poles due to delay are consumed by the double zero, the others curve toward the real-axis such that all loci asymptotes, which are horizontal, are vertically separated by $(2\pi / \text{delay})$ at both extremes of the real-axis.

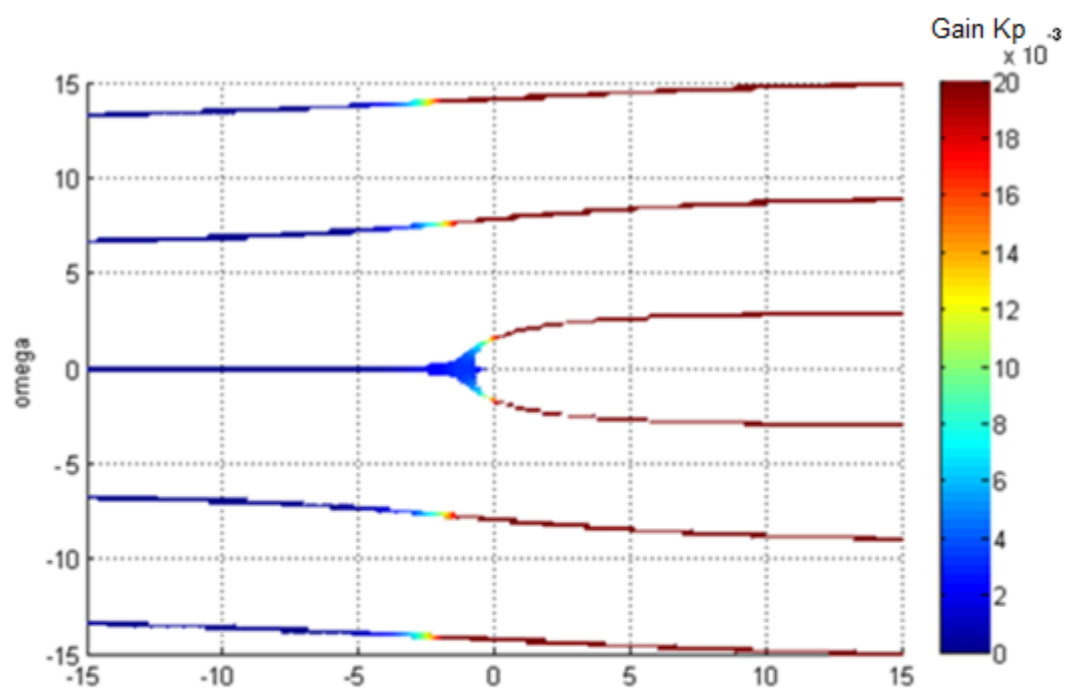


Figure G5. Root Loci of a Single-Pole, First-Order, Plant With Time Delay

Root loci, drawn by the numeric method developed in this paper, show paths of closed-loop poles, and their associated gain, for a feedback system comprised of a first-order plant with time delay. Plant time constant = 10 s (open-loop pole at $s = -0.1$, and delay of 1 s. Note: as gain increases the plant pole and a pole due to time delay collide and break-away from the real-axis. The break-away point moves toward $s = -\infty$ as time-delay goes to zero seconds.

Break-away point for first-order plant with time delay. The point on the real axis where, as compensation gain is increased closed-loop poles collide and then depart the real axis, the *break-away* point, is now determined for a feedback system comprised of a first-order plant with time delay. The plant time constant is $T = 10$ seconds so the open-loop plant pole is located at $s = -\frac{1}{T} = -0.1$, time delay $\theta = 1$ seconds, thus normalized time delay NTD $= \frac{\theta}{T} = 0.1$.

The breakaway point occurs where compensation gain k peaks in value along the real axis. Thus, k must be expressed in terms of position on the real axis σ , then its derivative along the real axis is set to zero, $\frac{dk}{d\sigma} = 0$.

Compensation gain k can be expressed in terms of σ , θ , and T by analyzing the characteristic equation along the real axis, $\omega = 0$

$$1 + kG(s)e^{-s\theta} \Big|_{\omega=0} = 0 \quad (\text{G34})$$

or

$$kG(s) \Big|_{\omega=0} = \frac{ke^{-s\theta}}{Ts + 1} \Big|_{\omega=0} = \frac{ke^{-\sigma\theta}}{T\sigma + 1} = -1$$

so

$$k = -(T\sigma + 1)e^{\sigma\theta} \quad (\text{G35})$$

The derivative of gain along the real axis is

$$\frac{dk}{d\sigma} = -\theta e^{\sigma\theta}(T\sigma + 1) - T e^{\sigma\theta} = (-\theta T\sigma - \theta - T)e^{\sigma\theta} = 0 \quad (\text{G36})$$

The derivative equals zero when $(-\theta T\sigma - \theta - T) = 0$, thus breakaway occurs at

$$-\sigma = \frac{-(\theta+T)}{\theta T} = -\frac{1}{T} - \frac{1}{\theta} = -\frac{1}{10} - \frac{1}{1} = -1.1 \quad (\text{G37})$$

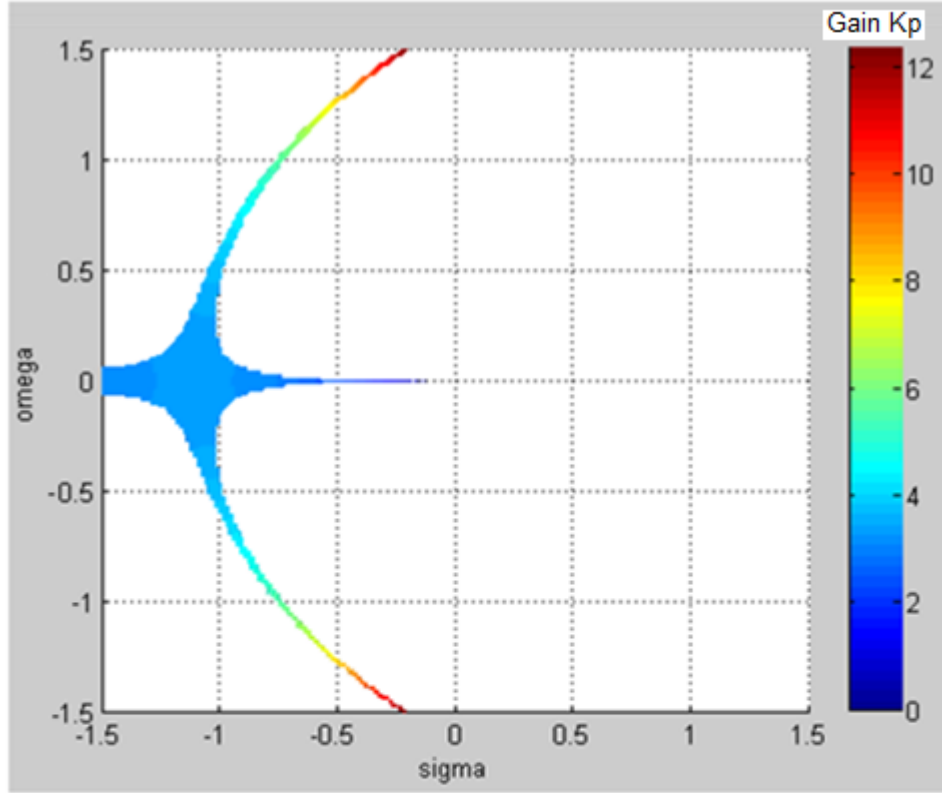


Figure G6. Root Loci of a Single-Pole, First-Order, Plant With Time Delay
(Close-up View of Break-Away Point)

Root loci, drawn by the numeric method developed in this paper, for a feed back system comprised of a first-order plant with time constant = 10 s and time delay = 1 s. NTD = 0.1

Note: the break-away point is at $s = -1.1$, matching the calculated value, this supports the accuracy of the numerical method of drawing root loci developed here.

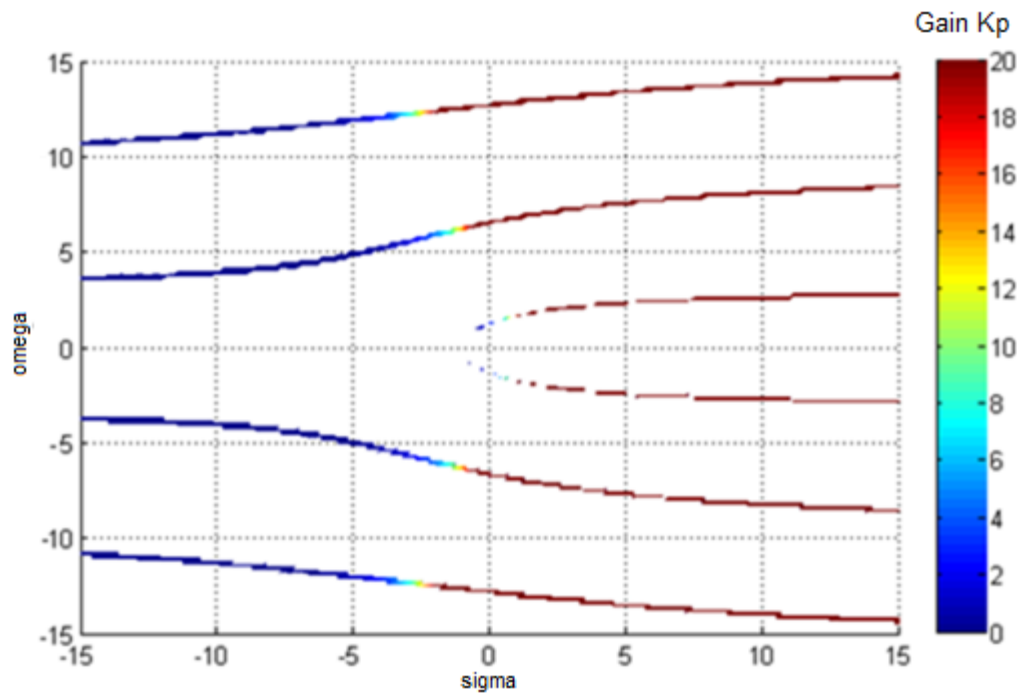


Figure G7. Root Loci of a Double-Pole, Second-Order, Plant With Time Delay

Root loci, drawn by the numerical method developed in this study, for a feedback system comprised of a second-order plant (with double pole at $s = -0.1$) and time delay of one second, $NTD = 0.1$.

Note plant poles produce loci that originate at the open-loop poles and terminate at the extreme right-hand side of the real axis, along with all other poles, due to time delay. Poles follow horizontal asymptotes separated vertically by $(2\pi / \text{delay}) = 2\pi$.

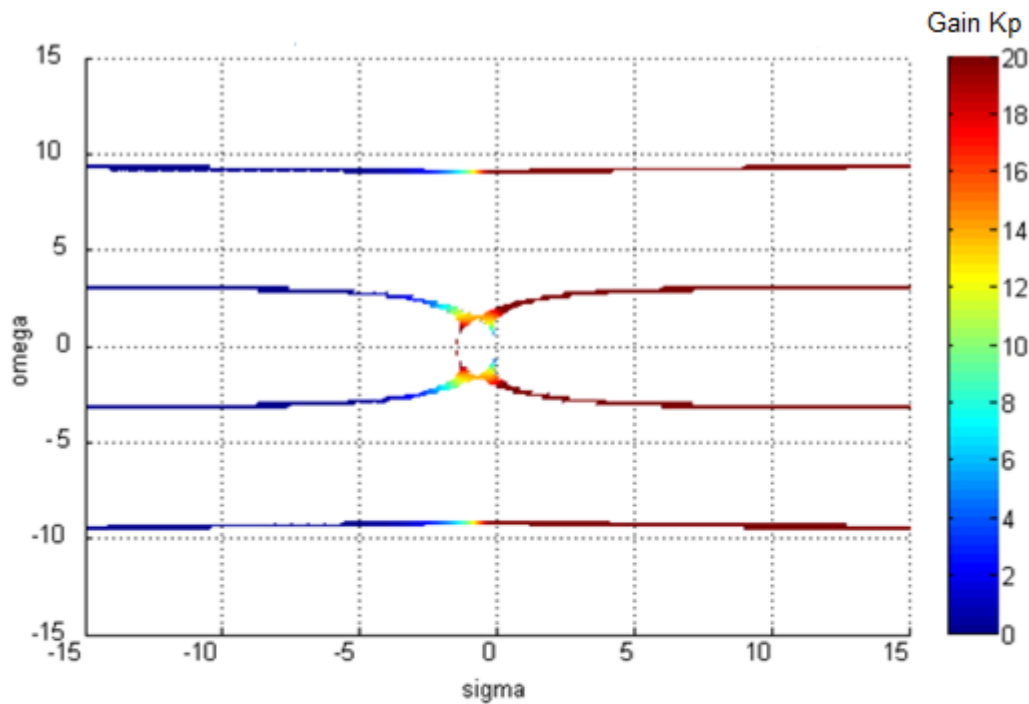


Figure G8. Root Loci of a PID-Compensated First-Order Plant With Time Delay

Root loci, drawn by the numeric method developed in this paper, of a feedback system comprised of a PID controlled first-order plant with time-delay. Plant time constant is 10 s (open-loop pole at $s = -0.1$), time delay = 1 s, compensator double zero is at $s = -1.25$. NTD = 0.1

Note: plant and integrator poles move toward the compensator's double zero, but closed-loop poles due to time delay reach it first. Plant and integrator poles instead head into the right half of the s-plane.

Appendix H

Compensation Gain K_p Yielding 5% Overshoot of Set Point for a First-Order Plant With Time Delay

Compensation gain for a first-order plant with time delay is determined for seven values of normalized time delay. A series of seven root-locus plots, drawn by the numerical method developed in this paper, were used to identify gains as discussed in Chapter 3: Results. Gains are selected such that the dominant closed-loop poles lie at locations which correspond with a damping coefficient of 0.7 (5% overshoot), in a purely second-order system, and are tabulated and plotted below in Table H1 and Figure H1.

Table H1

Compensation Gain Yielding 5% Overshoot for a First-Order Plant With Time Delay

	Recommended	Resulting
Normalized-Time Delay, NTD Time Delay θ Divided by Plant Time Constant T	Compensation Gain k_p (For 5% Overshoot)	2% Settling Time (Multiples of Open-Loop Settling Time)
0.05	9.90	0.015
0.10	4.85	0.029
0.15	3.16	0.0425
0.20	2.32	0.054
0.30	1.49	0.075
0.40	1.08	0.093
0.50	0.82	0.106

Note: Compensation gain for a first-order plant with time delay, resulting in 5% overshoot of final value, stated as a function of normalized time delay NTD.

Settling times are referenced to 2% open-loop settling time, assumed to be 4 plant time constants.

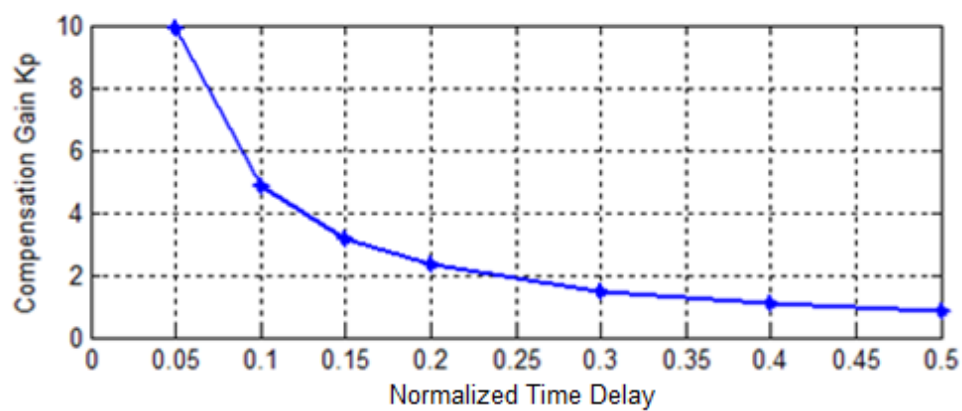


Figure H1. Compensation Gain Giving 5% Overshoot
for First-Order Plant With Time Delay

Graph of values listed in Table H1: compensation gain K_p for a first-order plant (resulting in 2% overshoot) versus normalized time delay = (plant time constant) / (time delay).

Note: no more than 5% of set point will occur after a unit-step change in set point or load disturbance.

Appendix I

Ziegler-Nichols PID Tuning

First-order plant with time delay. Ziegler-Nichols' step-response PID-tuning formulas state (Ziegler & Nichols, 1942)

$$K_p = \frac{1.2}{LR} \quad (11)$$

$$T_i = 2L \quad (12)$$

$$T_d = 0.5L \quad (13)$$

The inputs to these formula, R and L , can perfectly characterize the time constant T , and time delay θ , of a first-order plant with time delay, but here they are used to characterize and tune a second-order process as shown in Figure I1, which is an excerpt from their paper.

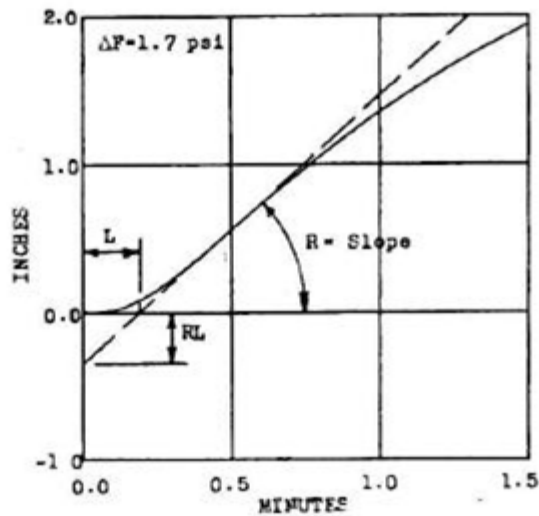


Figure 11. "Process Reaction Curve"

This diagram shows how to characterize a process for PID tuning using the Ziegler-Nichols step-response method (1942).

Note the process shown is actually second order, though metrics used in the tuning formulas, and characterization above, fit a first-order process.

The transfer function of a first-order plant with time constant T , and time delay θ , has the form

$$\frac{e^{-s\theta}}{Ts+1} \quad (14)$$

When Ziegler-Nichols tuning rules are applied to this plant, we show below the PID compensator has a double zero that lies at $s = -\frac{1}{\theta}$, a position that depends on time delay alone.

The zeros of a Ziegler-Nichols-tuned PID compensator are identified by using by their tuning formulas to substitute time delay and plant time constant for T_i , and T_d in the numerator of the PID transfer function $G_{PID}(s)$

$$\begin{aligned} \text{Numerator}[G_{PID}(s)] &= K_p \left(T_d s^2 + s + \frac{1}{T_i} \right) = K_p T_d \left(s^2 + \frac{1}{T_d} s + \frac{1}{T_i T_d} \right) = 0 \\ &= \frac{(1.2)T}{\theta} \frac{\theta}{2} \left(s^2 + \frac{2}{\theta} s + \frac{1}{\theta^2} \right) = 0.6T \left(s + \frac{1}{\theta} \right)^2 \quad (15) \end{aligned}$$

Thus, Ziegler-Nichols PID tuning for a first-order plant with time delay places the compensator's double zero at $s = -\frac{1}{L} = -\frac{1}{\theta}$.

Appendix J

Determination of Break-Away and Reentry Points of Loci in Systems with Time Delay

PI compensation of a first-order plant with time delay. The points where root loci of a PI-compensated first-order plant break-away from or reconnect with the real axis can be identified by analyzing the closed-loop system's characteristic equation

$$1 + G_{PI}(s) G_{FO}(s) e^{-s\theta} = 1 + k_p \frac{s+z}{s} \frac{1}{s+p} e^{-s\theta} = 0 \quad (\text{J1})$$

Applying the constraint that these points are on the real axis, $\omega = 0$, yields an expression for compensation gain k_p in terms of position σ

$$k_{p\omega=0} = -e^{s\theta} \frac{s^2+sp}{s+z} \Big|_{\omega=0} = -e^{\sigma\theta} \frac{\sigma^2+\sigma p}{\sigma+z} \quad (\text{J2})$$

Break-away and reentry points coincide with maxima or minima, respectively, in the value of gain k_p on the real axis, or $\frac{dk_p}{d\sigma} = 0$. From Equation J2 above

$$\frac{dk_p}{d\sigma} = \frac{d}{d\sigma} [e^{\sigma\theta} f(\sigma)] = 0 \quad (\text{J3})$$

where

$$f(\sigma) = \frac{\sigma^2 + \sigma p}{\sigma + z} \quad (\text{J4})$$

Thus

$$\frac{dk_p}{d\sigma} = e^{\sigma\theta} \left[\theta f(\sigma) + \frac{d}{d\sigma} f(\sigma) \right] = 0 \quad (\text{J5})$$

Substituting the derivative of Equation J4

$$\frac{d}{d\sigma} f(\sigma) = \frac{2\sigma + p}{\sigma + z} - \frac{\sigma^2 + \sigma p}{(\sigma + z)^2} \quad (\text{J6})$$

into Equation J5 results in a cubic polynomial

$$\theta\sigma^3 + (\theta z + \theta p + 1)\sigma^2 + (\theta p z + 2z)\sigma + p z = 0 \quad (\text{J7})$$

Roots of this equation are the break-away and reentry points. The MATLAB script shown below was written to generate a list of break-away and reentry points for a variety of PI zero positions. A list of break-away and reentry points is generated and shown below for a system where time delay = 1 s, the open-loop plant pole lies at $s = -0.1$, and where the PI zero position, s_z , is moved throughout the range $-1.5 \leq s_z \leq 0.0$.

```

% FO+PI break away points
clear;
theta=1; % time-delay

p =0.1; % plant pole

z = 0.125; % PI controller zero

element = [0 0 0 0 ];
rArray(1,:) = element;
rArrayIndex=0;

for z=0.00:.01:1.5
    rArrayIndex = rArrayIndex+1;

    ThirdOrder = theta;
    SecondOrder = (theta*(p+z)+1);
    FirstOrder = theta*p*z+2*z;
    ZerothOrder = p*z;

    % compute zeros of dk/d(sigma) polynomial
    rArray(rArrayIndex, 1)=z;
    b = roots([ ThirdOrder SecondOrder FirstOrder ZerothOrder]);
    rArray(rArrayIndex, 2)=b(1,1);
    rArray(rArrayIndex, 3)=b(1,2);
    rArray(rArrayIndex, 4)=b(1,3);

end
strelement ={'PI Zero', 'root#1', 'root#2', 'root#3'};
strelement
rArray

```

Sample Output of MATLAB Script

Time-Delay = 1s
Plant Pole at $s = -0.1$
PI Zero Range: 0.0 to -0.5

PI Zero	root#1	root#2	'root#3'
0	0	0	-1.1000
0.0100	-1.0916	-0.0092 - 0.0288i	-0.0092 + 0.0288i
0.0200	-1.0829	-0.0185 - 0.0388i	-0.0185 + 0.0388i
0.0300	-1.0739	-0.0280 - 0.0448i	-0.0280 + 0.0448i
0.0400	-1.0646	-0.0377 - 0.0483i	-0.0377 + 0.0483i
0.0500	-1.0550	-0.0475 - 0.0498i	-0.0475 + 0.0498i
0.0600	-1.0449	-0.0575 - 0.0493i	-0.0575 + 0.0493i
0.0700	-1.0344	-0.0678 - 0.0466i	-0.0678 + 0.0466i
0.0800	-1.0235	-0.0783 - 0.0411i	-0.0783 + 0.0411i
0.0900	-1.0120	-0.0890 - 0.0312i	-0.0890 + 0.0312i
0.1000	-1.0000	-0.1000	-0.1000
0.1100	-0.9873	-0.1468	-0.0759
0.1200	-0.9739	-0.1762	-0.0699
0.1300	-0.9596	-0.2039	-0.0664
0.1400	-0.9444	-0.2316	-0.0640

0.1500	-0.9280	-0.2598	-0.0622
0.1600	-0.9101	-0.2890	-0.0608
0.1700	-0.8906	-0.3197	-0.0597
0.1800	-0.8687	-0.3525	-0.0588
0.1900	-0.8438	-0.3881	-0.0580
0.2000	-0.8145	-0.4282	-0.0574
0.2100	-0.7776	-0.4756	-0.0568
0.2200	-0.7234	-0.5404	-0.0563
0.2300	-0.6371 - 0.0773i	-0.6371 + 0.0773i	-0.0558
0.2400	-0.6423 - 0.1423i	-0.6423 + 0.1423i	-0.0555
0.2500	-0.6474 - 0.1856i	-0.6474 + 0.1856i	-0.0551
0.2600	-0.6526 - 0.2204i	-0.6526 + 0.2204i	-0.0548
0.2700	-0.6577 - 0.2503i	-0.6577 + 0.2503i	-0.0545
0.2800	-0.6629 - 0.2769i	-0.6629 + 0.2769i	-0.0543
0.2900	-0.6680 - 0.3010i	-0.6680 + 0.3010i	-0.0540
0.3000	-0.6731 - 0.3233i	-0.6731 + 0.3233i	-0.0538
0.3100	-0.6782 - 0.3440i	-0.6782 + 0.3440i	-0.0536
0.3200	-0.6833 - 0.3635i	-0.6833 + 0.3635i	-0.0534
0.3300	-0.6884 - 0.3819i	-0.6884 + 0.3819i	-0.0533
0.3400	-0.6935 - 0.3994i	-0.6935 + 0.3994i	-0.0531
0.3500	-0.6985 - 0.4160i	-0.6985 + 0.4160i	-0.0529
0.3600	-0.7036 - 0.4320i	-0.7036 + 0.4320i	-0.0528
0.3700	-0.7087 - 0.4474i	-0.7087 + 0.4474i	-0.0527
0.3800	-0.7137 - 0.4621i	-0.7137 + 0.4621i	-0.0526
0.3900	-0.7188 - 0.4764i	-0.7188 + 0.4764i	-0.0524
0.4000	-0.7238 - 0.4902i	-0.7238 + 0.4902i	-0.0523
0.4100	-0.7289 - 0.5036i	-0.7289 + 0.5036i	-0.0522
0.4200	-0.7339 - 0.5165i	-0.7339 + 0.5165i	-0.0521
0.4300	-0.7390 - 0.5291i	-0.7390 + 0.5291i	-0.0521
0.4400	-0.7440 - 0.5414i	-0.7440 + 0.5414i	-0.0520
0.4500	-0.7491 - 0.5533i	-0.7491 + 0.5533i	-0.0519
0.4600	-0.7541 - 0.5650i	-0.7541 + 0.5650i	-0.0518
0.4700	-0.7591 - 0.5763i	-0.7591 + 0.5763i	-0.0517
0.4800	-0.7642 - 0.5874i	-0.7642 + 0.5874i	-0.0517
0.4900	-0.7692 - 0.5983i	-0.7692 + 0.5983i	-0.0516
0.5000	-0.7742 - 0.6089i	-0.7742 + 0.6089i	-0.0515

PID compensation of a second-order plant with time delay. Break-away and reentry points of root loci depicting a PID-compensated second-order plant with time delay are also found by analyzing the closed-loop system's characteristic equation

$$1 + G_{PID}(s) G_{SO}(s)e^{-s\theta} = 1 + k \frac{(s+z_1)(s+z_2)}{s} \frac{1}{(s+p)^2} e^{-s\theta} = 0 \quad (J8)$$

Since these points are on the real axis, where $\omega = 0$, gain k is expressed as a function of position σ on the real axis

$$k = -e^{s\theta} \frac{s(s+p)^2}{(s+z_1)(s+z_2)} \Big|_{\omega=0} = -e^{\sigma\theta} \frac{\sigma^3 + 2p\sigma^2 + p^2\sigma}{\sigma^2 + (z_1+z_2)^2 + z_1z_2} \quad (J9)$$

As was the case for a PI-compensated first-order plant with time delay, minima and maxima are determined by finding where the derivative of k with respect to position on the real axis is zero

$$\frac{dk}{d\sigma} = \frac{d}{d\sigma} [e^{\sigma\theta} f(\sigma)] = 0 \quad (J10)$$

where

$$f(\sigma) = \frac{\sigma^3 + 2p\sigma^2 + p^2\sigma}{\sigma^2 + (z_1+z_2)^2 + z_1z_2} \quad (J11)$$

Using

$$\frac{dk}{d\sigma} = e^{\sigma\theta} \left[\theta f(\sigma) + \frac{d}{d\sigma} f(\sigma) \right] = 0$$

and

$$\frac{d}{d\sigma} f(\sigma) = \frac{3\sigma^2 + 4p\sigma + p^2}{\sigma^2 + (z_1 + z_2)^2 + z_1 z_2} - \frac{2\sigma + (z_1 + z_2)}{(\sigma^2 + (z_1 + z_2)^2 + z_1 z_2)^2} (\sigma^3 + 2p\sigma^2 + p^2\sigma) \quad (\text{J12})$$

yields the fifth order polynomial of σ that determines where break-away or reconnection points lie

$$\begin{aligned} \frac{dk_p}{d\sigma} = & \theta\sigma^5 \\ & + (\theta(z_1 + z_2 + 2p) + 1)\sigma^4 \\ & + (\theta[z_1 z_2 + 2p(z_1 + z_2) + p^2] + 3(z_1 + z_2))\sigma^3 \\ & + (\theta[2pz_1 z_2 + p^2(z_1 + z_2)] + 3z_1 z_2 + 2p(z_1 + z_2) - p^2)\sigma^2 \\ & + ([\theta p + 4]pz_1 z_2)\sigma \end{aligned}$$

$$+p^2z_1z_2=0 \quad (\text{J13})$$

The MATLAB script shown below generates a list of break-away and reentry points for a range of PID double-zero positions. The list of break-away and reentry points is shown for a system where time delay = 1 s, the double plant pole is at $s = -0.1$, and where the PID double-zero position, s_{d-z} , is moved throughout the range $-0.5 \leq s_{d-z} \leq 0.0$.

```

%%%%%%%%%%
% MATLAB SCRIPT
%
% Calculate Second-Order Plant with PID compensation
% root locus break-away/reconnection points
%
%
% time-delay
%
theta = 1

%
% plant double-pole position
%
Zp = 0.1

element = [0 0 0 0 0 0];
rArray(1,:) = element;
rootArrayIndex=0;

%
% Generate zeros of polynomial for range in PID double-zero positions
%
for z=0.00:.01:0.5

    %
    % PID double-zero positions
    %
    Z1 = z;
    Z2 = z;

    %
    % dKp/dsigma polynomial coefficients
    %
    FifthOrder = theta;

    FourthOrder = theta*(Z1+Z2) + theta*2*Zp +1;

    ThirdOrder = theta*Z1*Z2 + 2*theta*Zp*(Z1+Z2) + theta*Zp*Zp +
3*(Z1+Z2) - (Z1+Z2);

    SecondOrder = theta*2*Zp*Z1*Z2 + theta*Zp*Zp*(Z1+Z2) + 3*Z1*Z2 +
4*Zp*(Z1+Z2) + Zp^2 - 2*Zp*Zp - 2*Zp*(Z1+Z2);

    FirstOrder = theta*Zp*Zp*Z1*Z2 + 4*Zp*Z1*Z2 +(Zp^2)*(Z1+Z2) -
(Z1+Z2)*Zp*Zp;

    ZerothOrder = Zp*Zp*Z1*Z2;

    %

```

```
% Calculate and store roots of polynomial
%
rootArray(rArrayIndex, 1)=z;
b= roots([FifthOrder FourthOrder ThirdOrder SecondOrder FirstOrder
ZerothOrder]);

rootArrayIndex = rootArrayIndex+1;
rootArray(rArrayIndex, 2) = b(1,1);
rootArray(rArrayIndex, 3) = b(2,1);
rootArray(rArrayIndex, 4) = b(3,1);
rootArray(rArrayIndex, 5) = b(4,1);
rootArray(rArrayIndex, 6) = b(5,1);
end

%
% Display roots of (dK/dSigma) for the range of PID double-zero
positions
%
strelement ={'PID Double Zero', 'root#1', 'root#2', 'root#3',
'root#4', 'root#5'};
strelement
rootArray
```

Sample Output of MATLAB Script
 Time-Delay = 1s
 Plant Double-Pole at $s = -0.1$
 PID Double Zero Range: 0.0 to -0.5

PID	root#1	root#2	'root#3'	'root#4'	'root#5'
Double Zero					
0	0	0	-1.1844	-0.1000	0.0844
0.0100	-1.1697	-0.1000	0.0359	0.0238	-0.0100
0.0200	-1.1544	-0.1000	0.0172 + 0.0379i	0.0172 - 0.0379i	-0.0200
0.0300	-1.1385	-0.1000	0.0042 + 0.0512i	0.0042 - 0.0512i	-0.0300
0.0400	-1.1218	-0.1000	-0.0091 + 0.0590i	-0.0091 - 0.0590i	-0.0400
0.0500	-1.1043	-0.1000	-0.0229 + 0.0633i	-0.0229 - 0.0633i	-0.0500
0.0600	-1.0859	-0.0371 + 0.0644i	-0.0371 - 0.0644i	-0.1000	-0.0600
0.0700	-1.0664	-0.0518 + 0.0623i	-0.0518 - 0.0623i	-0.1000	-0.0700
0.0800	-1.0458	-0.0671 + 0.0561i	-0.0671 - 0.0561i	-0.1000	-0.0800
0.0900	-1.0237	-0.0831 + 0.0434i	-0.0831 - 0.0434i	-0.1000	-0.0900
0.1000	-1.0000	-0.1000	-0.1000 + 0.0000i	-0.1000 - 0.0000i	-0.1000
0.1100	-0.9742	-0.1690	-0.1100	-0.1000	-0.0668
0.1200	-0.9458	-0.2152	-0.1200	-0.1000	-0.0590
0.1300	-0.9141	-0.2615	-0.1300	-0.1000	-0.0544
0.1400	-0.8776	-0.3111	-0.1400	-0.1000	-0.0513
0.1500	-0.8338	-0.3672	-0.1500	-0.1000	-0.0490
0.1600	-0.7766	-0.4361	-0.1600	-0.1000	-0.0472
0.1700	-0.6734	-0.5508	-0.1700	-0.1000	-0.0458
0.1800	-0.6177 + 0.1459i	-0.6177 - 0.1459i	-0.1800	-0.1000	-0.0447
0.1900	-0.6231 + 0.2150i	-0.6231 - 0.2150i	-0.1900	-0.1000	-0.0437
0.2000	-0.6285 + 0.2664i	-0.6285 - 0.2664i	-0.2000	-0.1000	-0.0429
0.2100	-0.6339 + 0.3093i	-0.6339 - 0.3093i	-0.2100	-0.1000	-0.0422
0.2200	-0.6392 + 0.3468i	-0.6392 - 0.3468i	-0.2200	-0.1000	-0.0416
0.2300	-0.6445 + 0.3804i	-0.6445 - 0.3804i	-0.2300	-0.1000	-0.0411
0.2400	-0.6497 + 0.4113i	-0.6497 - 0.4113i	-0.2400	-0.1000	-0.0406
0.2500	-0.6549 + 0.4399i	-0.6549 - 0.4399i	-0.2500	-0.1000	-0.0402
0.2600	-0.6601 + 0.4666i	-0.6601 - 0.4666i	-0.2600	-0.1000	-0.0398
0.2700	-0.6653 + 0.4919i	-0.6653 - 0.4919i	-0.2700	-0.1000	-0.0394
0.2800	-0.6704 + 0.5158i	-0.6704 - 0.5158i	-0.2800	-0.1000	-0.0391
0.2900	-0.6756 + 0.5386i	-0.6756 - 0.5386i	-0.2900	-0.1000	-0.0388
0.3000	-0.6807 + 0.5605i	-0.6807 - 0.5605i	-0.3000	-0.1000	-0.0386
0.3100	-0.6858 + 0.5814i	-0.6858 - 0.5814i	-0.3100	-0.1000	-0.0383
0.3200	-0.6909 + 0.6016i	-0.6909 - 0.6016i	-0.3200	-0.1000	-0.0381
0.3300	-0.6960 + 0.6211i	-0.6960 - 0.6211i	-0.3300	-0.1000	-0.0379
0.3400	-0.7011 + 0.6399i	-0.7011 - 0.6399i	-0.3400	-0.1000	-0.0377
0.3500	-0.7062 + 0.6582i	-0.7062 - 0.6582i	-0.3500	-0.1000	-0.0376
0.3600	-0.7113 + 0.6759i	-0.7113 - 0.6759i	-0.3600	-0.1000	-0.0374

0.3700	-0.7164 + 0.6931i	-0.7164 - 0.6931i	-0.3700	-0.1000	-0.0372
0.3800	-0.7215 + 0.7099i	-0.7215 - 0.7099i	-0.3800	-0.1000	-0.0371
0.3900	-0.7265 + 0.7263i	-0.7265 - 0.7263i	-0.3900	-0.1000	-0.0370
0.4000	-0.7316 + 0.7422i	-0.7316 - 0.7422i	-0.4000	-0.1000	-0.0368
0.4100	-0.7366 + 0.7578i	-0.7366 - 0.7578i	-0.4100	-0.1000	-0.0367
0.4200	-0.7417 + 0.7730i	-0.7417 - 0.7730i	-0.4200	-0.1000	-0.0366
0.4300	-0.7468 + 0.7879i	-0.7468 - 0.7879i	-0.4300	-0.1000	-0.0365
0.4400	-0.7518 + 0.8025i	-0.7518 - 0.8025i	-0.4400	-0.1000	-0.0364
0.4500	-0.7569 + 0.8168i	-0.7569 - 0.8168i	-0.4500	-0.1000	-0.0363
0.4600	-0.7619 + 0.8309i	-0.7619 - 0.8309i	-0.4600	-0.1000	-0.0362
0.4700	-0.7669 + 0.8446i	-0.7669 - 0.8446i	-0.4700	-0.1000	-0.0361
0.4800	-0.7720 + 0.8581i	-0.7720 - 0.8581i	-0.4800	-0.1000	-0.0360
0.4900	-0.7770 + 0.8714i	-0.7770 - 0.8714i	-0.4900	-0.1000	-0.0359
0.5000	-0.7821 + 0.8845i	-0.7821 - 0.8845i	-0.5000	-0.1000	-0.0359